

# **Geração de Malhas por Refinamento de Delaunay**



# Publicações Matemáticas

## **Geração de Malhas por Refinamento de Delaunay**

Marcelo Siqueira  
UFRN

Afonso Paiva  
USP - São Carlos

Paulo Pagliosa  
UFMS



30<sup>o</sup> Colóquio Brasileiro de Matemática

Copyright © 2015 by Marcelo Siqueira, Afonso Paiva e Paulo Pagliosa

Impresso no Brasil / Printed in Brazil

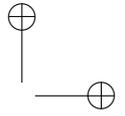
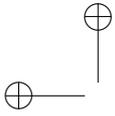
Capa: Noni Geiger / Sérgio R. Vaz

### **30<sup>o</sup> Colóquio Brasileiro de Matemática**

- Aplicações Matemáticas em Engenharia de Produção - Leonardo J. Lustosa e Fernanda M. P. Raupp
- Boltzmann-type Equations and their Applications - Ricardo Alonso
- Dissipative Forces in Celestial Mechanics - Sylvio Ferraz-Mello, Clodoaldo Grotta-Ragazzo e Lucas Ruiz dos Santos
- Economic Models and Mean-Field Games Theory - Diogo A. Gomes, Levon Nurbekyan and Edgard A. Pimentel
- Generic Linear Recurrent Sequences and Related Topics - Letterio Gatto
- **Geração de Malhas por Refinamento de Delaunay - Marcelo Siqueira, Afonso Paiva e Paulo Pagliosa**
- Global and Local Aspects of Levi-flat Hypersurfaces - Arturo Fernández Pérez e Jiri Lebl
- Introdução às Curvas Elípticas e Aplicações - Parham Salehyan
- Métodos de Descida em Otimização Multiobjetivo - B. F. Svaiter e L. M. Graña Drummond
- Modern Theory of Nonlinear Elliptic PDE - Boyan Slavchev Sirakov
- Novel Regularization Methods for Ill-posed Problems in Hilbert and Banach Spaces - Ismael R. Bleyer e Antonio Leitão
- Probabilistic and Statistical Tools for Modeling Time Series - Paul Doukhan
- Tópicos da Teoria dos Jogos em Computação - O. Lee, F. K. Miyazawa, R. C. S. Schouery e E. C. Xavier
- Topics in Spectral Theory - Carlos Tomei

**Distribuição:** IMPA  
Estrada Dona Castorina, 110  
22460-320 Rio de Janeiro, RJ  
E-mail: [ddic@impa.br](mailto:ddic@impa.br)  
<http://www.impa.br>

ISBN: 978-85-244-0406-1



À minha saudosa avó, Mailde (*in memoriam*).  
Aos meus queridos pais, Roberto e Dilma.  
À minha amada esposa, Poliana, e aos meus adoráveis filhos, Sofia  
e Daniel.

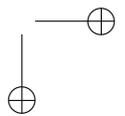
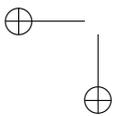
Marcelo Siqueira

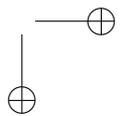
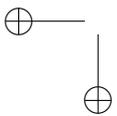
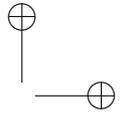
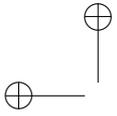
À minha esposa, Lucilly, por todo seu amor, apoio e companheirismo.

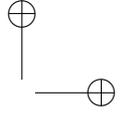
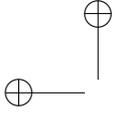
Afonso Paiva

À minha família, pelo apoio, compreensão e paciência.

Paulo Pagliosa

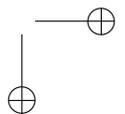
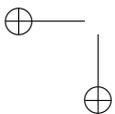




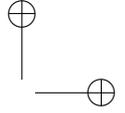
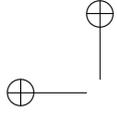


# Conteúdo

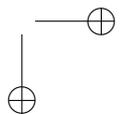
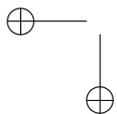
<b>Sobre os autores</b>	<b>vii</b>
<b>Prefácio</b>	<b>ix</b>
<b>Agradecimentos</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Malhas e geração de malhas . . . . .	1
1.2 Refinamento de Delaunay . . . . .	4
1.3 Sobre o livro . . . . .	5
1.4 Organização . . . . .	6
<b>2 Geometria afim</b>	<b>9</b>
2.1 Considerações iniciais . . . . .	9
2.2 Pontos <i>versus</i> vetores . . . . .	10
2.3 Espaço afim . . . . .	15
2.4 Subespaço afim . . . . .	19
2.5 Independência afim e referencial afim . . . . .	23
2.6 Mapa afim . . . . .	26
2.7 Forma (ou funcional) afim . . . . .	32
2.8 Espaço afim euclidiano real . . . . .	35
2.9 Exercícios . . . . .	37
2.10 Notas bibliográficas . . . . .	41
<b>3 Convexidade</b>	<b>43</b>
3.1 Considerações iniciais . . . . .	43
3.2 Conjuntos convexos . . . . .	44

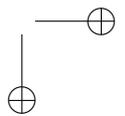
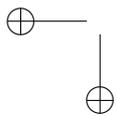
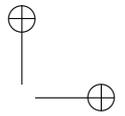
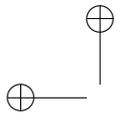


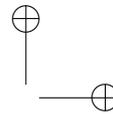
3.3	Fecho convexo . . . . .	46
3.4	Graham <i>scan</i> . . . . .	52
3.4.1	Corretude . . . . .	56
3.4.2	Complexidade . . . . .	61
3.5	Cota inferior . . . . .	64
3.6	Exercícios . . . . .	69
3.7	Notas bibliográficas . . . . .	80
<b>4</b>	<b>Triangulações</b>	<b>81</b>
4.1	Considerações iniciais . . . . .	81
4.2	O que é triangulação? . . . . .	82
4.3	O mapa de elevação . . . . .	85
4.4	A triangulação de Delaunay . . . . .	96
4.5	Propriedades . . . . .	112
4.6	O algoritmo incremental . . . . .	137
4.6.1	Os pontos especiais e a triangulação $\mathcal{TD}_0$ . . .	138
4.6.2	A inserção de pontos . . . . .	139
4.6.3	A remoção dos pontos especiais . . . . .	151
4.6.4	Complexidade . . . . .	158
4.7	Cota inferior . . . . .	170
4.8	Aspectos de implementação . . . . .	172
4.9	Exercícios . . . . .	173
4.10	Notas bibliográficas . . . . .	187
<b>5</b>	<b>Geração de malhas</b>	<b>189</b>
5.1	Considerações iniciais . . . . .	189
5.2	Critérios de otimalidade . . . . .	190
5.3	Refinamento de Delaunay . . . . .	194
5.3.1	Entrada e saída . . . . .	194
5.3.2	O algoritmo de Ruppert . . . . .	198
5.3.3	Implementação . . . . .	206
5.3.4	Complexidade . . . . .	208
5.3.5	A triangulação de Delaunay restrita . . . . .	209
5.4	Término . . . . .	213
5.5	Otimalidade . . . . .	225
5.6	Ângulos pequenos . . . . .	230
5.7	Exercícios . . . . .	237
5.8	Notas bibliográficas . . . . .	238



<b>6</b>	<b>Aplicação</b>	<b>241</b>
6.1	Considerações iniciais . . . . .	241
6.2	Modelo matemático do problema . . . . .	244
6.3	Formulação do MEF . . . . .	253
6.4	Esquema computacional . . . . .	263
6.5	Elemento finito CST . . . . .	271
6.6	Aspectos de implementação . . . . .	274
6.7	Notas bibliográficas . . . . .	287
<b>A</b>	<b>Predicados geométricos</b>	<b>305</b>
A.1	Considerações iniciais . . . . .	305
A.2	Orientação . . . . .	306
A.3	Teste de orientação . . . . .	309
A.4	Os predicados <i>left</i> , <i>lefton</i> e <i>collinear</i> . . . . .	313
A.5	O predicado <i>classify</i> . . . . .	314
A.6	O predicado <i>incircle</i> . . . . .	317
A.7	Notas bibliográficas . . . . .	320





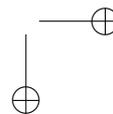
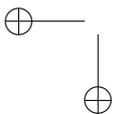


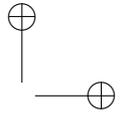
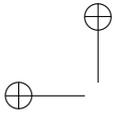
## Sobre os autores

**Marcelo Siqueira** graduou-se em Ciências da Computação em 1992 pela Universidade Federal do Rio Grande do Norte (UFRN), tornou-se mestre em Ciências da Computação e Matemática Computacional em 1994 pela Universidade de São Paulo (USP) e concluiu doutorado em Ciência da Computação em 2006 pela *University of Pennsylvania*. Atualmente é Professor Associado do Departamento de Matemática (DMAT) da UFRN. Seus principais interesses de pesquisa incluem geometria computacional (com ênfase em geração de malhas), modelagem de curvas e superfícies e topologia digital de imagens.

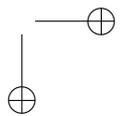
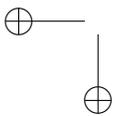
**Afonso Paiva** obteve o título de Bacharel em Matemática em 2000 pela Universidade Federal de Uberlândia (UFU), tornou-se mestre em Matemática em 2003 pela Associação Instituto Nacional de Matemática Pura e Aplicada (IMPA) e concluiu doutorado em Matemática em 2007 pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Atualmente é Professor Doutor 1 do Departamento de Matemática Aplicada e Estatística (SME) do Instituto de Ciências Matemáticas e Computação (ICMC) da Universidade de São Paulo (USP). Suas principais áreas de interesse são modelagem computacional de fenômenos físicos, modelagem de curvas e superfícies, dinâmica dos fluidos computacional e métodos numéricos auto-validados.

**Paulo Pagliosa** graduou-se em Engenharia Civil em 1987 pela Universidade Federal de Santa Maria (UFSM), concluiu doutorado em Engenharia Civil (Engenharia de Estruturas) em 1998 pela Escola de Engenharia de São Carlos (EESC) da Universidade de São Paulo





(USP) e realizou estágio pós-doutorado em 2012 no Instituto de Ciências Matemáticas e Computação (ICMC) da USP. Atualmente é Professor Associado na Faculdade de Computação (FACOM) da Universidade Federal de Mato Grosso do Sul (UFMS). Suas principais áreas de pesquisa são visualização e simulação, com ênfase em GPGPU, mecânica computacional, física para jogos digitais e computação gráfica.

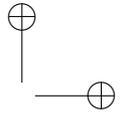
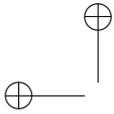


# Prefácio

A ideia de escrever um livro sobre geração de malhas e, em particular, sobre geração de malhas usando refinamento de Delaunay, foi uma consequência natural da pesquisa desenvolvida por nós, autores, no âmbito do projeto “Caracterização de Aquíferos usando Métodos Computacionais”, que é financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e coordenado por nós mesmos.

Como parte das atividades propostas no referido projeto, os autores estão desenvolvendo uma biblioteca de classes escrita na linguagem C++ para gerar malhas 2D e 3D a partir de imagens digitais binárias 2D e 3D de rochas coletadas em aquíferos. O conteúdo deste livro corresponde à parte inicial do trabalho dos autores e tem foco nos fundamentos teóricos de um método para gerar malhas de triângulos de regiões poligonais do plano: o refinamento de Delaunay. Um aspecto que o distingue dos demais métodos de geração de malhas é a sua sólida fundamentação matemática. Esta fundamentação proporcionou o desenvolvimento de algoritmos de geração de malhas 2D e 3D capazes de fornecer cotas justas para os valores de alguns dos mais importantes parâmetros de qualidade de malha em aplicações práticas, tais como quantidade e razão de aspecto de triângulos das malhas.

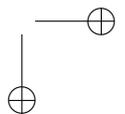
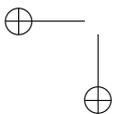
Embora vários livros tenham sido escritos sobre geração de malhas [55, 42, 131, 64, 50, 77, 23, 78], alguns dos quais com foco no método de refinamento de Delaunay, além de artigos que fornecem uma excelente visão geral da área e do estado da arte [9, 12, 129, 122], nós, autores, desconhecemos a existência de livros sobre geração de malhas escritos em português ou até mesmo traduzidos para o por-

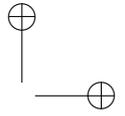
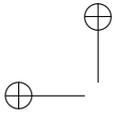


tuguês. Então, acreditamos que este livro possa servir como uma primeira introdução ao tema escrita em português. É importante ressaltar, no entanto, que o objetivo do livro não é o de proporcionar uma visão geral e completa sobre geração de malhas, mas sim de descrever um único algoritmo para gerar malhas de triângulos para regiões poligonais do plano euclidiano. Além disso, a descrição tem foco nos fundamentos matemáticos utilizados para demonstrar a corretude, a complexidade e algumas das propriedades de otimalidade do algoritmo. Os aspectos de implementação não são abordados no livro, mas serão discutidos durante o minicurso homônimo do 30<sup>o</sup> CBM.

Nós não incluímos no livro nenhum resultado original advindo de nossas pesquisas em geração de malhas. No entanto, destacamos dois aspectos importantes do livro. Primeiro, o tópico “triangulação de Delaunay” é tratado de forma autossuficiente; ou seja, nós definimos esta triangulação sem antes definir o diagrama de Voronoi. Embora a exposição possa parecer menos ortodoxa, acreditamos que a triangulação de Delaunay, no contexto de geração de malhas, tenha “vida própria” e que o diagrama de Voronoi deva ser introduzido apenas se ele for utilizado na descrição do tema “geração de malhas”, o que não é o caso aqui. Segundo, escrevemos um capítulo sobre uma aplicação para as malhas geradas pelo algoritmo estudado no livro. Este capítulo fornece ao leitor a oportunidade de conhecer um uso prático para malhas e também para perceber a importância dos critérios de qualidade de malhas no contexto de uma de suas aplicações.

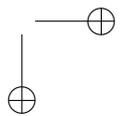
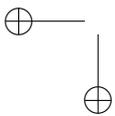
Finalmente, ressaltamos que os fundamentos matemáticos do algoritmo estudado neste livro são basicamente os mesmos vistos em uma disciplina básica sobre “geometria computacional”. O conteúdo desta disciplina é extremamente divertido para aqueles com maior inclinação por matemática (discreta) e algoritmos. Em 1991, no 16<sup>o</sup> CBM, dois pesquisadores do IMPA, Luiz Henrique de Figueiredo e Paulo César Pinto Carvalho, agradeceram-nos com um livro e minicurso sobre Geometria Computacional [32]. Tanto o minicurso quanto o livro exerceram uma influência sobre a decisão de alguns dos autores em fazer pesquisa em temas em que fazemos uso de geometria (computacional, discreta, diferencial, etc). Vinte e quatro anos depois, tivemos a oportunidade de escrever este livro para o 30<sup>o</sup>

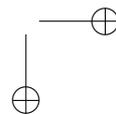
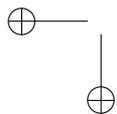
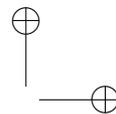
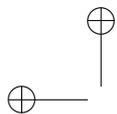




CBM, o qual trata de um tema cuja fundamentação matemática é praticamente a mesma encontrada em [32]. Sendo assim, esperamos que o nosso livro, juntamente com o nosso minicurso, despertem em você, leitor, o interesse por estudos avançados na área de Geração de Malhas.

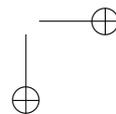
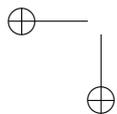
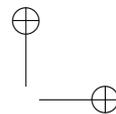
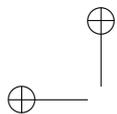
Marcelo Siqueira  
Afonso Paiva  
Paulo Pagliosa

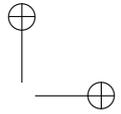
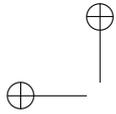




# Agradecimentos

Gostaríamos de agradecer às instituições de ensino às quais estamos vinculados, em particular, aos departamentos e faculdades nos quais estamos lotados, o Departamento de Matemática da UFRN, o Departamento de Matemática Aplicada e Estatística do ICMC-USP e a Faculdade de Computação da UFMS, pelas condições de trabalho que nos foram oferecidas e que nos permitiram concluir a escrita deste livro. Gostaríamos de agradecer ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo financiamento do projeto “Caracterização de Aquíferos Usando Métodos Computacionais”, Processo Nº 486951/2012-0, do qual nasceu a ideia de escrever este livro. O projeto se originou a partir de uma aplicação apresentada aos autores pelo professor Leandson Lucena (DGEF-UFRN). A equipe do projeto conta também com o professor Bruno Motta de Carvalho (DIMAp-UFRN), que é responsável por segmentar as imagens de amostras de rochas a partir das quais os autores geram as malhas da aplicação. Agradecemos a ambos pela oportunidade de trabalho em equipe e pelo constante apoio. Um agradecimento especial às nossas famílias pelo constante apoio e compreensão da nossa ausência, principalmente, nos dois últimos meses do prazo de entrega deste livro. Finalmente, agradecemos à Comissão Organizadora e aos revisores *ad hoc* do 30º do Colóquio Brasileiro de Matemática (CBM) por terem acatado a nossa proposta de curso e pela organização do evento.





# Capítulo 1

## Introdução

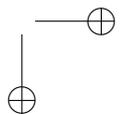
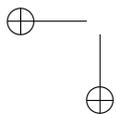
### 1.1 Malhas e geração de malhas

Uma *malha* é uma subdivisão de um domínio geométrico em formas geométricas menores e mais simples, denominadas *elementos*, tais como triângulos e quadrados, se o domínio for planar — 2D, e tetraedros e hexaedros, se o domínio for espacial — 3D (veja a Figura 1.1). Malhas são utilizadas por aplicações de diversas áreas do conhecimento. Em geografia e cartografia, malhas são utilizadas para interpolar mapas de altura em terrenos, na confecção de mapas, na geração de curvas de nível e na visualização topográfica [132, 130]. Em computação gráfica, objetos gráficos são, em geral, convertidos para uma malha antes de serem submetidos à operação de “renderização” e malhas poligonais têm se tornado o padrão *de facto* na representação de objetos gráficos [17]. Em geral, malhas são necessárias em aplicações que se utilizam de interpolação ou aproximação multivariada [71].

De particular interesse para muitas aplicações das diversas engenharias é o fato da geração de uma malha do domínio do problema ser um pré-requisito para o uso do Método dos Elementos Finitos (MEF) clássico<sup>1</sup>, que é um dos métodos numéricos mais potentes e populares

---

<sup>1</sup>O termo “clássico” é usado aqui para distinguir a formulação do MEF baseada em malhas daquela que não utiliza malhas (*meshless*). A formulação *meshless*



para a solução de equações diferenciais parciais na forma variacional (veja o Capítulo 6). A acurácia e precisão da solução numérica dessas equações, pelo MEF, aliadas à eficiência com a qual a solução é obtida, são altamente dependentes de parâmetros de *qualidade* da malha, tais como quantidade e forma dos elementos [5, 81, 39, 121], além de regularidade, direcionalidade e adaptatividade da própria malha [142, 129, 12].

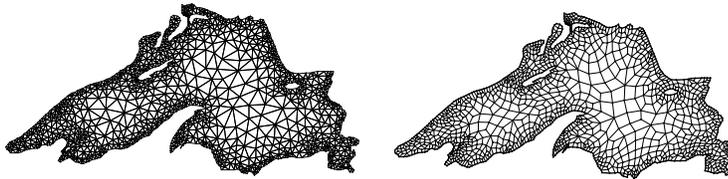
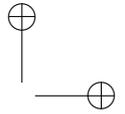
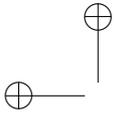


Figura 1.1: Uma malha de triângulos (esquerda) e uma malha de quadriláteros de uma aproximação poligonal de uma imagem do *Lake Superior*.

A grande maioria dos algoritmos de geração malhas descritos na literatura assume que o domínio,  $\Omega$ , para o qual uma malha deve ser criada é um conjunto compacto de  $\mathbb{E}^d$  cuja fronteira é definida por um *complexo politopal* [138]. Quando  $d = 2$ , o domínio  $\Omega$  é simplesmente uma região poligonal de  $\mathbb{E}^2$  (com ou sem buracos). Quando  $d = 3$ , o domínio  $\Omega$  é simplesmente uma região poliedral de  $\mathbb{E}^3$  (com ou sem buracos). A fronteira de  $\Omega$  também pode ser descrita por uma curva (resp. superfície) suave (por partes), fechada e limitada se  $\Omega \subset \mathbb{E}^2$  (resp.  $\mathbb{E}^3$ ).

Nos últimos 30 anos, malhas de triângulos de regiões poligonais (convexas e não convexas), ou limitadas por curvas suaves, em  $\mathbb{E}^2$  têm sido detalhadamente investigadas pela comunidade científica e suas propriedades teóricas são atualmente bem compreendidas [9, 122]. Isto resultou no desenvolvimento de algoritmos de geração de malhas de triângulos capazes de gerar malhas cujos valores de parâmetros, tais como quantidade e forma dos triângulos e adaptatividade da malha, são comprovadamente ótimos ou quase ótimos [6, 82, 11, 113,

apresenta várias limitações em relação à clássica.

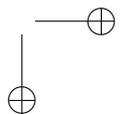
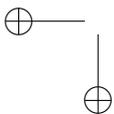


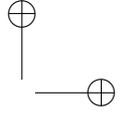
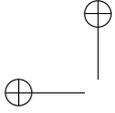
120, 15, 84, 102, 86, 87, 136, 133]. O termo *comprovadamente* é usado neste livro para qualificar uma propriedade (matemática) que pode ser verificada por uma prova.

Em contraste com o estado-da-arte para malhas de triângulos, as propriedades teóricas de malhas de quadriláteros para regiões planares ainda não são bem compreendidas. Apesar da existência de algoritmos de geração de malhas de quadriláteros que otimizam algum parâmetro da malha [46, 92, 10, 106, 75, 115, 103, 3, 134, 43], não se conhece um algoritmo para gerar malhas de quadriláteros que, comprovadamente e *simultaneamente*, otimize vários parâmetros da malha.

Em se tratando de uma região poliedral (convexa ou não convexa) em  $\mathbb{E}^3$ , ou limitada por superfícies suaves (por partes), o problema de gerar uma malha comprovadamente boa oferece alguns bons desafios de pesquisa. Nos últimos 25 anos foram desenvolvidos vários algoritmos para geração de malhas de tetraedros com garantias teóricas para a qualidade da malha gerada [36, 88, 85, 119, 94, 27, 100, 38, 20, 22, 98, 125, 70, 111, 37, 21, 112, 124]. Muitos desses algoritmos podem ser vistos como extensões do caso bidimensional (isto é, malhas de triângulos) para o caso tridimensional (isto é, malhas de tetraedros). Infelizmente, algumas garantias teóricas oferecidas pelos algoritmos para malhas de triângulos não foram estendidas, com o mesmo êxito, para malhas de tetraedros e é aí que reside uma boa fonte de problemas.

Quando se deseja gerar uma malha de hexaedros para uma região poliedral (convexa ou não convexa), muito pouco é conhecido em termos de garantias teóricas para parâmetros de qualidade da malha. A maioria dos esforços de pesquisa que resultou em algoritmos com alguma garantia teórica se limita aos aspectos topológicos da malhas; isto é, tais algoritmos não definem uma geometria para a malha. O trabalho recente de Jeff Erickson [44] representa um avanço significativo em termos de garantias teóricas para malhas de hexaedros. Por outro lado, há uma enorme quantidade de algoritmos para geração de malhas de hexaedros que se baseia em heurísticas e que apresenta bons resultados experimentais para uma larga classe de domínios espaciais [69].





## 1.2 Refinamento de Delaunay

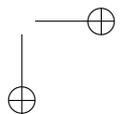
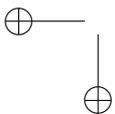
A vasta maioria dos algoritmos capazes de produzir malhas de triângulos ou tetraedros comprovadamente boas é baseada na técnica conhecida por *refinamento de Delaunay* [23]. Esta técnica foi desenvolvida, originalmente, para gerar malhas de triângulos de regiões poligonais,  $\Omega$ , em  $\mathbb{E}^2$ . Os algoritmos baseados em refinamento de Delaunay constroem uma *triangulação de Delaunay*,  $\mathcal{TD}(P)$ , do conjunto,  $P$ , de vértices da curva poligonal de fronteira de  $\Omega$  [4]. Em seguida, a triangulação  $\mathcal{TD}(P)$  é *refinada* para que dois objetivos sejam atingidos: (a) a fronteira de  $\Omega$  seja totalmente coberta por uma união de arestas de  $\mathcal{TD}(P)$  e (b) todos os triângulos de  $\mathcal{TD}(P)$  (no interior de  $\Omega$ ) passem em um teste de qualidade de forma geométrica. Por refinar  $\mathcal{TD}(P)$ , entenda a inserção de vértices extras (de *Steiner*) em  $\mathcal{TD}(P)$ .

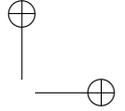
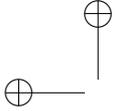
Podemos mostrar que, sob determinadas condições, o processo de refinamento de Delaunay sempre termina e produz uma malha de saída que cumpre os objetivos (a) e (b) [113, 120, 84, 110, 109, 108] (Capítulo 5). Por definição, o domínio coberto por  $\mathcal{TD}(P)$  é o fecho convexo,  $FC(P)$ , do conjunto,  $P$ , de vértices de  $\Omega$ . Quando  $\Omega$  é um subconjunto próprio de  $FC(P)$ , os algoritmos baseados em refinamento de Delaunay removem de  $\mathcal{TD}(P)$  os triângulos no exterior de  $\Omega$ .

Um outro aspecto importante do problema de geração de malhas diz respeito à quantidade de elementos da malha, ou seja, o *tamanho* da malha. Jim Ruppert [113] definiu uma função, denominada *local feature size*, ou simplesmente *lfs*, que, de certa forma, captura a noção de complexidade de espaço de um domínio geométrico. Usando esta função, Ruppert definiu um critério para avaliar otimalidade de tamanho de uma malha; a saber, se  $\Omega \subset \mathbb{E}^d$  é um conjunto compacto que representa o domínio do problema em um espaço (afim) euclidiano  $d$ -dimensional, então uma malha para  $\Omega$  possui *tamanho ótimo* se

$$m \in \Theta \left( \int_{x \in \Omega} \frac{1}{lfs(x)^d} dx \right), \quad (1.1)$$

em que  $m$  é o número de vértices da malha. Usando este critério de otimalidade de tamanho, podemos provar que os algoritmos para geração de malhas via refinamento de Delaunay produzem malhas de





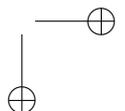
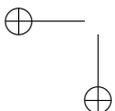
tamanho ótimo quando  $d = 2$ . Para  $d \geq 3$ , os algoritmos baseados em refinamento de Delaunay e disponíveis na literatura não desfrutam da mesma garantia [23]. O algoritmo desenvolvido por Mitchell e Vavasis [88], que *não* se baseia em refinamento de Delaunay, produz malhas de tamanho ótimo, mas apresenta outras limitações práticas, tais como o alinhamento de elementos com eixos coordenados (anisotropia).

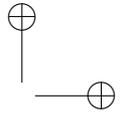
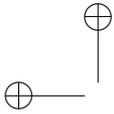
### 1.3 Sobre o livro

O objetivo deste livro é justamente introduzir os aspectos teóricos e algorítmicos da geração de malhas de regiões poligonais em  $\mathbb{E}^2$  usando refinamento de Delaunay. Em particular, estudaremos o algoritmo de Ruppert para geração de malhas de triângulos de regiões poligonais em  $\mathbb{E}^2$  [113], mas com as modificações mais recentes que garantem o término dele [84, 121, 110]. Com o intuito de ressaltar a importância de malhas em diversas aplicações, dedicamos um capítulo do livro (Capítulo 6) a uma aplicação das malhas geradas pelo algoritmo de Ruppert. Em particular, uma aplicação em um problema de análise elastostática 2D usando o Método dos Elementos Finitos (MEF).

Como o livro foi escrito para alunos de graduação e sem necessariamente possuir uma formação específica (por exemplo, alunos de Matemática ou de Ciência da Computação), há uma forte ênfase na fundamentação teórica que alicerça os principais aspectos do algoritmo de Ruppert. É preciso entender que este algoritmo é acompanhado de garantias teóricas, ou seja, propriedades, que podem ser demonstradas por uma prova matemática. Foi realizado um enorme esforço para explicar todos os elementos básicos de Matemática necessários para entender os detalhes de todas as demonstrações do livro.

De maneira geral, acreditamos que os únicos pré-requisitos necessários para ler e entender, por completo, o conteúdo do livro sejam conhecimentos de álgebra linear, cálculo diferencial e integral (em uma e várias variáveis), probabilidade, matemática discreta (métodos de prova, combinatória e grafos), algoritmos e estruturas de dados básicas.

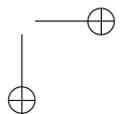
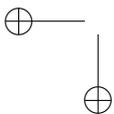


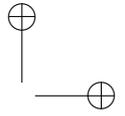
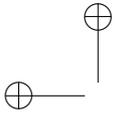


## 1.4 Organização

Este livro contém seis capítulos e um apêndice, como descrito a seguir:

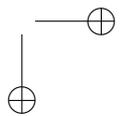
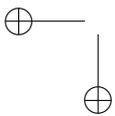
- O Capítulo 2 introduz conceitos elementares da geometria afim, tais como espaço afim, combinações afins, independência afim e mapas afins. O objetivo é introduzir o leitor ao ambiente (espaço) no qual os objetos estudados no livro (as malhas) residem.
- O Capítulo 3 apresenta o importante conceito de fecho convexo de um conjunto de pontos em  $\mathbb{E}^d$  e um algoritmo para calcular o fecho convexo de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$ . Uma simples modificação deste algoritmo, descrita no capítulo seguinte, permite-nos gerar uma triangulação inicial de  $P$ , a qual pode ser utilizada por outro algoritmo descrito no mesmo capítulo para produzir a triangulação de Delaunay de  $P$ .
- O Capítulo 4 é dedicado à triangulação de Delaunay. Ele aborda tanto os aspectos teóricos quanto os aspectos algorítmicos. Uma ênfase maior é dada ao algoritmo incremental de construção da triangulação de Delaunay de um subconjunto,  $P$ , de pontos de  $\mathbb{E}^2$ , já que o algoritmo de Ruppert é uma espécie de extensão dele.
- O Capítulo 5 descreve o algoritmo de Ruppert para geração de malhas de triângulos de regiões poligonais em  $\mathbb{E}^2$ . Este algoritmo é o principal objeto de estudo do livro e faz uso do algoritmo incremental, descrito no Capítulo 4, para construir a triangulação de Delaunay de um subconjunto,  $P$ , de pontos de  $\mathbb{E}^2$ .
- O Capítulo 6 descreve uma aplicação, para as malhas geradas pelo algoritmo de Ruppert, a um problema de análise elastostática 2D usando o método dos elementos finitos (MEF). Primeiro, o capítulo introduz os conceitos elementares de tensão, deformação e elasticidade necessários à compreensão do modelo matemático do problema. Em seguida, apresenta resumi-

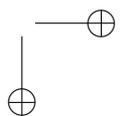
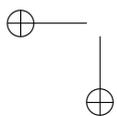
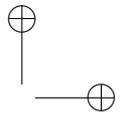
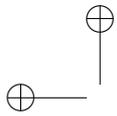


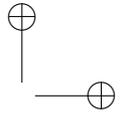
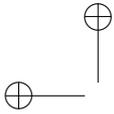


damente a formulação do MEF, baseada em uma forma fraca de resíduos ponderados, bem como o esquema computacional do método e a derivação do elemento finito usado na aplicação. Por fim, o capítulo discute aspectos de implementação e mostra resultados de análise obtidos com a aplicação.

- O Apêndice A descreve os predicados geométricos utilizados pelo algoritmo incremental para gerar triangulações de Delaunay.





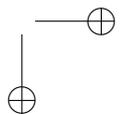
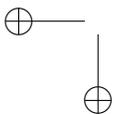


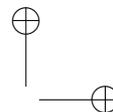
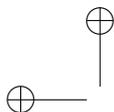
## Capítulo 2

# Geometria afim

### 2.1 Considerações iniciais

Este capítulo é uma breve introdução aos elementos básicos da Geometria Afim e tem como principal objetivo definir, formalmente, o “ambiente” no qual residem os objetos estudados no restante do livro. A Seção 2.2 motiva o estudo de espaços afins através de uma discussão informal sobre a diferenciação entre pontos e vetores. A Seção 2.3 apresenta a definição de espaço afim, que nada mais é do que um espaço no qual os elementos são pontos e vetores. Este é o espaço adequado para definirmos os objetos estudados neste texto. As Seções 2.4-2.7 apresentam, cada uma, a definição de um elemento da Geometria Afim análogo a um elemento da Álgebra Linear. A Seção 2.8 apresenta a definição de espaço (afim) euclidiano real, que é o espaço afim utilizado no restante deste livro. A Seção 2.9 oferece vários exercícios sobre os tópicos abordados neste capítulo. Os exercícios exploram os aspectos teóricos e aplicados do conteúdo do capítulo. Assim como no final dos capítulos seguintes, encerramos o presente comentando as referências bibliográficas utilizadas (a Seção 2.10).





## 2.2 Pontos *versus* vetores

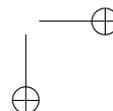
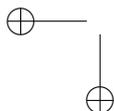
Apesar da familiaridade que temos com os espaços *vetoriais* euclidianos sobre  $\mathbb{R}$ , eles não formam o “ambiente” matemático apropriado para estudarmos malhas de triângulos. De fato, tais malhas são compostas por vértices, segmentos de reta e triângulos. Esses objetos são, por sua vez, compostos por *pontos* e pontos não são *vetores*. Vetores são representados geometricamente por segmentos de reta orientados, os quais possuem direção e comprimento, mas não possuem posição fixa; pontos possuem uma posição fixa, mas não possuem direção nem comprimento. Pontos podem ser movidos via translação, mas translações não afetam vetores — *já que translações preservam a direção e o comprimento deles*. Vetores podem ser somados e multiplicados por um escalar, mas como podemos somar dois pontos?

Para responder a pergunta acima, vamos restringir nossa atenção a pontos do plano  $\Pi$  da geometria euclidiana plana. Como de costume, fixamos um sistema de eixos ortogonais,  $Oxy$ , em  $\Pi$  e definimos uma correspondência biunívoca entre os pares ordenados de  $\mathbb{R}^2$  e os pontos do plano. Sejam  $a$  e  $b$  dois pontos quaisquer de  $\Pi$  tais que  $a$ ,  $b$  e  $O$  são distintos dois a dois. É possível definir a soma  $a + b$  dos pontos  $a$  e  $b$  como sendo o ponto  $c$  da extremidade do vetor

$$\vec{Oc} = \vec{Oa} + \vec{Ob},$$

como mostra a Figura 2.1. O que acabamos de fazer foi simplesmente definir a soma de pontos como sendo a soma de vetores. Esta forma de definir soma de pontos implica em definir as coordenadas do ponto  $c$  como sendo a soma das respectivas coordenadas dos pontos  $a$  e  $b$ . Obviamente, todas as coordenadas são definidas com relação a  $Oxy$ .

O problema com a definição de soma de pontos dada acima é que o ponto resultante é dependente do sistema de eixos ortogonais fixado. De fato, se mudarmos a origem, digamos  $O = a$ , obteremos  $c = a + b = b$  e se tomarmos  $O$  como sendo o ponto médio do segmento  $\overline{ab}$ , obteremos  $c = a + b = O$ . Essas observações indicam que a soma de pontos, como definida acima, contrasta com a soma de vetores, na qual o resultado é um vetor *que não depende do sistema de eixos ortogonais fixado*. De fato, vetores, ao contrário de pontos,



não possuem posição fixa e são determinados apenas por magnitude e direção.

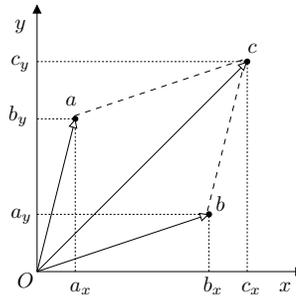


Figura 2.1: Tentativa de definição de soma de dois pontos em  $\mathbb{R}^2$ .

Há, entretanto, uma forma de definirmos a soma de dois pontos que não depende do sistema de eixos ortogonais escolhido. As coordenadas de um ponto são reveladas apenas após tal sistema ser fixado. Para definir a nova operação de soma entre dois pontos, definimos primeiro uma operação de soma entre um ponto e um vetor. Sejam  $p$  um ponto e  $\mathbf{v}$  um vetor em  $\Pi$ . Define-se a operação de *soma entre  $p$  e  $\mathbf{v}$* , denotada por  $p + \mathbf{v}$ , como sendo o (único) ponto  $q$  de  $\Pi$  tal que  $\vec{pq} = \mathbf{v}$  (ver Figura 2.2). Da mesma forma que o ponto  $q$  é único, há um único vetor  $\mathbf{v}$  em  $\Pi$  tal que  $q = p + \mathbf{v}$ , para quaisquer dois pontos  $p$  e  $q$  em  $\Pi$ . Neste contexto, o vetor  $\mathbf{v}$  é visto como uma translação que move o ponto  $p$  para a posição do ponto  $q$ . A operação de soma entre ponto e vetor goza das duas seguintes propriedades:

$$p + \mathbf{0} = p$$

e

$$(p + \mathbf{v}) + \mathbf{u} = p + (\mathbf{v} + \mathbf{u}),$$

em que  $\mathbf{0}$  é o vetor nulo e  $\mathbf{v}$  e  $\mathbf{u}$  são vetores quaisquer (ver Figura 2.2).

Dados dois pontos,  $p$  e  $q$  em  $\Pi$ , e dois números reais,  $\lambda$  e  $\mu$ , tais que  $\lambda + \mu = 1$ , vamos definir a soma ponderada,  $\lambda \cdot p + \mu \cdot q$ , de  $p$  e  $q$  que é conhecida por *combinação afim (ou baricêntrica) de  $p$  e  $q$  com pesos  $\lambda$  e  $\mu$* . A combinação afim  $\lambda \cdot p + \mu \cdot q$  tem um significado

bastante intuitivo, evidenciado pela manipulação algébrica dada a seguir:

$$\lambda \cdot p + \mu \cdot q = (1 - \mu) \cdot p + \mu \cdot q = p + \mu \cdot (q - p).$$

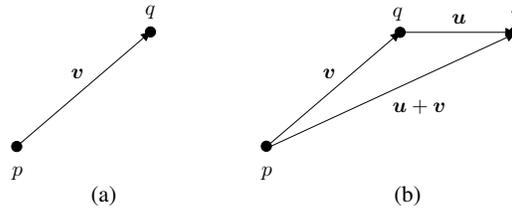


Figura 2.2: Pontos  $q = p + \mathbf{v}$  e  $r = (p + \mathbf{v}) + \mathbf{u} = p + (\mathbf{v} + \mathbf{u})$ .

Obviamente, não sabemos o que é multiplicar um escalar por um ponto e muito menos o que significa subtrair um ponto de outro. Mas, se interpretarmos  $(q-p)$  como sendo o vetor,  $\mathbf{v}$ , associado ao segmento de reta orientado  $\overrightarrow{pq}$ , então  $\lambda \cdot p + \mu \cdot q$  tem o mesmo significado de  $p + \mu \cdot \mathbf{v}$ , que é um ponto bem definido pela operação de soma entre ponto e vetor. Para tornar mais explícito o significado de  $\lambda \cdot p + \mu \cdot q$ , denotamos  $\mathbf{v}$  por  $\overrightarrow{pq}$  e escrevemos  $p + \mu \cdot \overrightarrow{pq}$  em vez de

$$p + \mu \cdot \mathbf{v}.$$

Com o significado que atribuímos a  $\lambda \cdot p + \mu \cdot q$ , com  $\lambda + \mu = 1$ , o ponto médio,  $m$ , do segmento de reta  $\overline{pq}$  pode ser definido pela soma

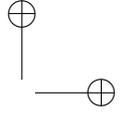
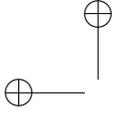
$$\frac{1}{2} \cdot p + \frac{1}{2} \cdot q = p + \frac{1}{2} \cdot \overrightarrow{pq},$$

o que faz sentido, pois o comprimento do vetor  $\overrightarrow{pq}$  é aquele do segmento de reta  $\overline{pq}$ . Logo, a metade deste comprimento é igual ao comprimento do vetor

$$\frac{1}{2} \cdot \overrightarrow{pq}.$$

Consequentemente, a “translação” de  $p$  pelo vetor  $\frac{1}{2} \cdot \overrightarrow{pq}$  é o ponto médio do segmento  $\overline{pq}$ . De forma análoga, podemos dizer que o ponto

$$p + \frac{1}{2} \cdot \overrightarrow{pq}$$



divide o segmento  $\overline{pq}$  na razão  $(1/2) : (1/2)$ . Pela mesma lógica, o ponto

$$\frac{1}{3} \cdot p + \frac{2}{3} \cdot q = p + \frac{2}{3} \cdot \overrightarrow{pq}$$

divide  $\overline{pq}$  na razão  $(1/3) : (2/3)$ . Isto significa que a distância do ponto acima ao ponto  $p$  é duas vezes maior que a distância dele ao ponto  $q$ .

É importante ressaltar que a expressão  $\lambda \cdot p + \mu \cdot q$  só faz sentido porque  $\lambda + \mu = 1$ . Aliás, usando a mesma restrição sobre os pesos, podemos definir a soma de  $n + 1$  pontos quaisquer em  $\Pi$ , com  $n \in \mathbb{N}$ . Sejam  $p_0, \dots, p_n$   $n + 1$  pontos quaisquer em  $\Pi$ , e sejam  $\alpha_0, \dots, \alpha_n$  números reais tais que  $\sum_{i=0}^n \alpha_i = 1$ . Então, a *combinação afim*

$$\sum_{i=0}^n \alpha_i \cdot p_i$$

é o ponto

$$p_0 + \left( \sum_{i=1}^n \alpha_i \cdot \overrightarrow{p_0 p_i} \right).$$

O termo

$$\sum_{i=1}^n \alpha_i \cdot \overrightarrow{p_0 p_i}$$

é uma combinação linear dos vetores  $\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}$ . Por exemplo, se  $p_0, p_1$  e  $p_2$  são três pontos não colineares em  $\Pi$ , então temos que

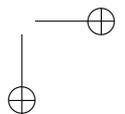
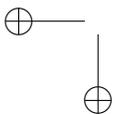
$$\frac{1}{3} \cdot p_0 + \frac{1}{3} \cdot p_1 + \frac{1}{3} \cdot p_2$$

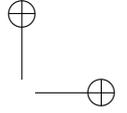
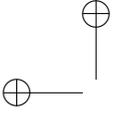
é o *baricentro* do triângulo definido por  $p_0, p_1$  e  $p_2$ . De fato, os três pontos

$$m_{01} = \frac{1}{2} \cdot p_0 + \frac{1}{2} \cdot p_1,$$

$$m_{12} = \frac{1}{2} \cdot p_1 + \frac{1}{2} \cdot p_2,$$

$$m_{02} = \frac{1}{2} \cdot p_0 + \frac{1}{2} \cdot p_2$$





são os pontos médios dos segmentos  $\overline{p_0p_1}$ ,  $\overline{p_1p_2}$  e  $\overline{p_0p_2}$  (ver Figura 2.3) e os segmentos  $\overline{p_0m_{12}}$ ,  $\overline{p_1m_{02}}$  e  $\overline{p_2m_{01}}$  são as *medianas* do triângulo. Sabemos que o baricentro é o ponto de interseção das medianas. Mas,

$$\frac{1}{3} \cdot p_0 + \frac{2}{3} \cdot m_{12} = \frac{1}{3} \cdot p_0 + \frac{2}{3} \cdot \left( \frac{1}{2} \cdot p_1 + \frac{1}{2} \cdot p_2 \right) = \frac{1}{3} \cdot p_0 + \frac{1}{3} \cdot p_1 + \frac{1}{3} \cdot p_2.$$

Analogamente,

$$\frac{1}{3} \cdot p_1 + \frac{2}{3} \cdot m_{02} = \frac{1}{3} \cdot p_2 + \frac{2}{3} \cdot m_{01} = \frac{1}{3} \cdot p_0 + \frac{1}{3} \cdot p_1 + \frac{1}{3} \cdot p_2.$$

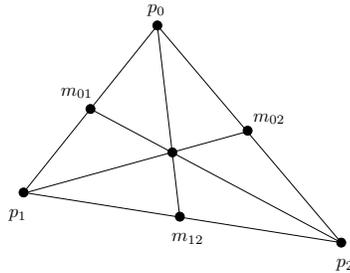
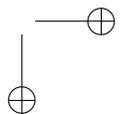
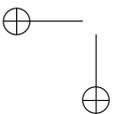


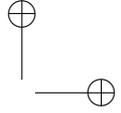
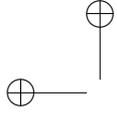
Figura 2.3: Construção do baricentro do triângulo  $\triangle(p_0p_1p_2)$ .

Logo, a combinação afim  $\frac{1}{3} \cdot p_0 + \frac{1}{3} \cdot p_1 + \frac{1}{3} \cdot p_2$  é mesmo o baricentro do triângulo com vértices em  $p_0$ ,  $p_1$  e  $p_2$ . De forma geral, se os coeficientes  $\lambda$ ,  $\mu$  e  $\nu$  são não negativos e a soma deles é igual a 1, então

$$\lambda \cdot p_0 + \mu \cdot p_1 + \nu \cdot p_2 \tag{2.1}$$

é um ponto do triângulo com vértices em  $p_0$ ,  $p_1$  e  $p_2$ . A recíproca também é verdadeira: qualquer ponto do triângulo com vértices em  $p_0$ ,  $p_1$  e  $p_2$  é dado pela Equação (2.1) com  $\lambda, \mu, \nu \geq 0$  e  $\lambda + \mu + \nu = 1$ . Além disso, os escalares  $\lambda$ ,  $\mu$  e  $\nu$  são únicos para cada ponto do triângulo. Por outro lado, se permitirmos que um ou mais de  $\lambda$ ,  $\mu$  e  $\nu$  sejam negativos, a Equação (2.1) pode gerar todos os pontos do plano.





## 2.3 Espaço afim

Agora, estamos devidamente preparados para definir o “ambiente” adequado para lidar com pontos e vetores. A este ambiente, daremos o nome de espaço afim euclidiano real bidimensional. De maneira informal, ele pode ser visto como dois espaços em um só: um espaço vetorial bidimensional sobre  $\mathbb{R}$  e um espaço formado por pontos associados a pares ordenados de  $\mathbb{R}^2$ . Para definir o espaço afim euclidiano real bidimensional, necessitamos primeiro definir espaço jafim.

**Definição 2.1.** Um espaço afim é o conjunto vazio ou uma tripla,

$$\langle E, V_E, + \rangle,$$

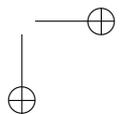
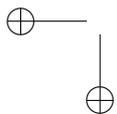
onde  $E$  é um conjunto não vazio de pontos,  $V_E$  é um espaço vetorial (de translações ou vetores livres) e  $+ : E \times V_E \rightarrow E$  é uma função, denominada aqui de ação transitiva, que goza das propriedades a seguir:

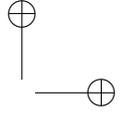
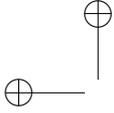
- (1)  $p + \mathbf{0} = p$ , para todo  $p \in E$ .
- (2)  $(p + \mathbf{u}) + \mathbf{w} = p + (\mathbf{u} + \mathbf{w})$ , para todo  $p \in E$  e  $\mathbf{u}, \mathbf{w} \in V_E$ .
- (3) Para quaisquer  $p, q \in E$ , há um único  $\mathbf{v}$  em  $V_E$  tal que  $p + \mathbf{v} = q$ .

A dimensão de  $\langle E, V_E, + \rangle$ , denotada simplesmente por  $\dim(E)$ , é igual à dimensão do espaço vetorial  $V_E$ . O espaço vetorial,  $V_E$ , é chamado de espaço vetorial associado a  $E$ . Por simplicidade de notação, denotamos  $\langle E, V_E, + \rangle$ , ocasionalmente, por  $\langle E, V_E \rangle$  ou simplesmente por  $E$ .

**Observação 2.1.** Assumimos que  $\dim(E)$  é finita para todo espaço  $E$ .

A Definição 2.1 não trata o conjunto  $E$  de forma particular. Mas, neste texto, restringimos nossa atenção ao caso em que tanto  $E$  quanto  $V_E$  são o conjunto  $\mathbb{R}^d$  de  $d$ -uplas de números reais. A fim de distinguir o duplo papel das  $d$ -uplas de  $\mathbb{R}^d$  (isto é, ponto e vetor), usamos a notação de vetor linha para pontos e a notação de vetor coluna para vetores. Desta forma, a ação  $+$  do espaço vetorial  $\mathbb{R}^d$





sobre o conjunto  $\mathbb{R}^d$  (visto como um conjunto de pontos) é definida como

$$(a_1, \dots, a_d) + \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} = (a_1 + u_1, \dots, a_d + u_d).$$

O espaço afim  $\langle \mathbb{R}^d, \mathbb{R}^d, + \rangle$  é denotado por  $\mathbb{A}^d$  e chamado *espaço afim real de dimensão  $d$* . De forma geral, para qualquer corpo  $\mathbb{K}$ , o espaço afim  $\langle \mathbb{K}^d, \mathbb{K}^d, + \rangle$ , com  $+ : \mathbb{K}^d \times \mathbb{K}^d \rightarrow \mathbb{K}^d$ , é denotado por  $\mathbb{A}_{\mathbb{K}}^d$ , com  $\mathbb{A}_{\mathbb{R}}^d = \mathbb{A}^d$ .

**Exemplo 2.1.** *Seja  $L$  o subconjunto de  $\mathbb{A}^2$  consistindo de todos os pontos  $(x, y)$  satisfazendo a equação  $x + y - 1 = 0$ . O conjunto  $L$  é a reta com coeficiente angular igual a  $-1$  e que passa pelos pontos  $(1, 0)$  e  $(0, 1)$  (ver Figura 2.4). A reta  $L$  pode ser transformada em um espaço afim,  $\langle L, \mathbb{R}, + \rangle$  se definirmos a ação  $+ : L \times \mathbb{R} \rightarrow L$  tal que, para qualquer ponto  $(x, 1-x)$  de  $L$  e qualquer vetor  $\mathbf{v} \in \mathbb{R}$ , temos que*

$$(x, 1-x) + \mathbf{v} = (x + \mathbf{v}, 1-x - \mathbf{v}).$$

*Para mostrar que  $L$  é um espaço afim, devemos de provar que  $+$  goza das propriedades (1)-(3) da Definição 2.1. Esta tarefa é deixada como exercício.*

Seja  $E$  um espaço afim. Dados  $a, b$  e  $c$  em  $E$ , sabemos que

$$c = a + \vec{ac}, \quad b = a + \vec{ab} \quad \text{e} \quad c = b + \vec{bc}.$$

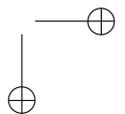
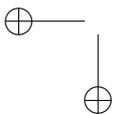
Usando a propriedade (2) da Definição 2.1, obtemos

$$c = b + \vec{bc} = (a + \vec{ab}) + \vec{bc} = a + (\vec{ab} + \vec{bc}).$$

Por sua vez, a propriedade (3) da Definição 2.1 nos diz que

$$\vec{ac} = \vec{ab} + \vec{bc}. \tag{2.2}$$

A igualdade na Equação (2.2) é conhecida como *identidade de Chasles*. Esta igualdade nos permite provar o resultado estabelecido pelo Lema 2.1:



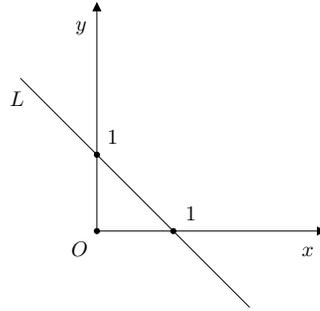


Figura 2.4: A linha  $x + y - 1 = 0$  em  $\mathbb{A}^2$  é um espaço afim.

**Lema 2.1.** *Dado um espaço afim  $E$ , sejam  $(p_i)_{i=1}^n$  uma lista de pontos de  $E$  e  $(\alpha_i)_{i=1}^n$  uma lista de escalares. Se  $\sum_{i=1}^n \alpha_i = 1$ , então temos que*

$$a + \sum_{i=1}^n \alpha_i \cdot \vec{ap_i} = b + \sum_{i=1}^n \alpha_i \cdot \vec{bp_i},$$

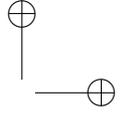
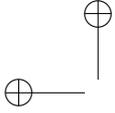
em que  $a$  e  $b$  são dois pontos quaisquer de  $E$ .

*Demonstração.* Da identidade de Chasles, temos que

$$\begin{aligned} a + \sum_{i=1}^n \alpha_i \cdot \vec{ap_i} &= a + \sum_{i=1}^n \alpha_i \cdot (\vec{ab} + \vec{bp_i}) \\ &= a + \sum_{i=1}^n \alpha_i \cdot \vec{ab} + \sum_{i=1}^n \alpha_i \cdot \vec{bp_i} \\ &= a + \vec{ab} + \sum_{i=1}^n \alpha_i \cdot \vec{bp_i} \\ &= b + \sum_{i=1}^n \alpha_i \cdot \vec{bp_i} \end{aligned}$$

pois

$$\sum_{i=1}^n \alpha_i = 1 \quad \text{e} \quad b = a + \vec{ab}.$$



□

A importância do Lema 2.1 reside no fato dele nos garantir que o ponto

$$a + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{ap_i}$$

é independente da escolha da “origem”  $a$ , o que faz com que pontos sejam definidos da mesma maneira que vetores (ou seja, independentes de coordenadas). Esta propriedade motiva a seguinte definição:

**Definição 2.2.** *Seja  $E$  um espaço afim. Então, para qualquer lista,  $(p_i)_{i=1}^n$ , de pontos de  $E$ , para qualquer lista,  $(\alpha_i)_{i=1}^n$ , de escalares em  $\mathbb{R}$  tal que  $\sum_{i=1}^n \alpha_i = 1$  e para qualquer ponto  $a \in E$ , o ponto definido por*

$$a + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{ap_i},$$

que é independente da escolha de  $a$ , como estabelecido pelo Lema 2.1, é chamado de *baricentro* (ou combinação baricêntrica ou combinação afim) dos pontos  $p_i$  associados aos pesos  $\alpha_i$ , e é denotado por

$$\sum_{i=1}^n \alpha_i \cdot p_i.$$

Sejam  $p_0, p_1$  e  $p_2$  três pontos quaisquer de  $\mathbb{A}^2$  e sejam  $\alpha_0, \alpha_1$  e  $\alpha_2$  três números reais tais que  $\alpha_0 + \alpha_1 + \alpha_2 = 1$ . De acordo com a Definição 2.2, o baricentro de  $p_0, p_1$  e  $p_2$  com pesos  $\alpha_0, \alpha_1$  e  $\alpha_2$  é tal que

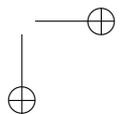
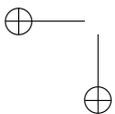
$$\sum_{i=0}^2 \alpha_i \cdot p_i = a + \sum_{i=0}^2 \alpha_i \cdot \overrightarrow{ap_i},$$

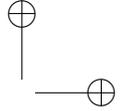
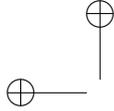
em que  $a$  é qualquer ponto de  $\mathbb{A}^2$ . Tomando  $a = p_0$ , obtemos

$$\sum_{i=0}^2 \alpha_i \cdot p_i = p_0 + \sum_{i=0}^2 \alpha_i \cdot \overrightarrow{p_0 p_i} = p_0 + \sum_{i=1}^2 \alpha_i \cdot \overrightarrow{p_0 p_i},$$

pois

$$\overrightarrow{p_0 p_0} = \mathbf{0}.$$





Analogamente, poderíamos ter  $a = p_1$  ou  $a = p_2$ . Pelo Lema 2.1,

$$\sum_{i=0}^2 \alpha_i \cdot p_i = p_0 + \sum_{i=1}^2 \alpha_i \cdot \overrightarrow{p_0 p_i},$$

$$\sum_{i=0}^2 \alpha_i \cdot p_i = p_1 + \sum_{i \in \{0,2\}} \alpha_i \cdot \overrightarrow{p_1 p_i}$$

e

$$\sum_{i=0}^2 \alpha_i \cdot p_i = p_2 + \sum_{i=0}^1 \alpha_i \cdot \overrightarrow{p_2 p_i}.$$

Quando lidamos com baricentros, é conveniente utilizar a notação de *ponto ponderado*: um par  $(p, \alpha)$ , em que  $p \in E$  é um ponto e  $\alpha$ , um escalar. Então, dada a lista  $((p_i, \alpha_i))_{i=1}^n$ , com  $\sum_{i=1}^n \alpha_i = 1$ , dizemos que  $\sum_{i=1}^n \alpha_i \cdot p_i$  é o *baricentro da lista  $((p_i, \alpha_i))_{i=1}^n$  de pontos ponderados*. De agora em diante, usaremos a definição de combinação afim para introduzir, nos espaços afins, noções da Álgebra Linear análogas a subespaço vetorial, independência linear, base e forma linear.

## 2.4 Subespaço afim

**Definição 2.3.** Dado um espaço afim,  $\langle E, V_E, + \rangle$ , um subconjunto  $F$  de  $E$  é dito um subespaço afim (de  $\langle E, V_E, + \rangle$ ) se, para toda lista de pontos ponderados  $((P_i, \alpha_i))_{i=1}^n$  em  $F$  tal que  $\sum_{i=1}^n \alpha_i = 1$ , temos que

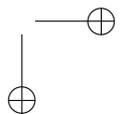
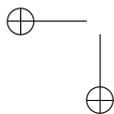
$$\sum_{i=1}^n \alpha_i \cdot p_i \in F.$$

**Exemplo 2.2.** Considere o subconjunto  $U$  de  $\mathbb{R}^2$  definido por

$$U = \{(x, y) \in \mathbb{R}^2 \mid ax + by = c\},$$

isto é, o conjunto de soluções da equação

$$ax + by = c,$$



onde  $a, b, c \in \mathbb{R}$ , com  $a \neq 0$  e  $b \neq 0$ . Dados  $n$  pontos quaisquer,  $((x_i, y_i))_{i=1}^n$ , de  $U$  e  $n$  escalares quaisquer,  $(\alpha_i)_{i=1}^n$ , tais que  $\sum_{i=1}^n \alpha_i = 1$ , queremos mostrar que o baricentro de  $((x_i, y_i), \alpha_i)_{i=1}^n$ ,

$$\sum_{i=1}^n \alpha_i \cdot (x_i, y_i),$$

pertence a  $U$ . De fato, se  $(x_i, y_i) \in U$ , então  $ax_i + by_i = c$ . Se multiplicarmos os dois lados de cada equação  $ax_i + by_i = c$  por  $\alpha_i$  e adicionarmos as  $n$  equações resultantes, obtemos a seguinte igualdade

$$\sum_{i=1}^n (\alpha_i ax_i + \alpha_i by_i) = \sum_{i=1}^n \alpha_i c.$$

Como

$$\sum_{i=1}^n \alpha_i = 1,$$

obtemos

$$a \left( \sum_{i=1}^n \alpha_i x_i \right) + b \left( \sum_{i=1}^n \alpha_i y_i \right) = \left( \sum_{i=1}^n \alpha_i \right) c = c,$$

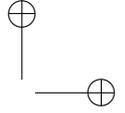
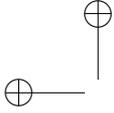
o que mostra que o baricentro de  $((x_i, y_i), \alpha_i)_{i=1}^n$  está em  $U$ ; isto é,

$$\left( \sum_{i=1}^n \alpha_i x_i, \sum_{i=1}^n \alpha_i y_i \right) = \sum_{i=1}^n \alpha_i \cdot (x_i, y_i) \in U.$$

Logo, o subconjunto  $U$ , que é uma reta, é um subespaço afim de  $\mathbb{A}^2$ .

O subespaço  $U$  do Exemplo 2.2 está intimamente relacionado com o subconjunto de  $\mathbb{R}^2$  definido por  $\mathbf{U} = \{(x, y) \in \mathbb{R}^2 \mid ax + by = 0\}$ , que é o conjunto de soluções da equação  $ax + by = 0$ . O conjunto  $\mathbf{U}$  é, na verdade, um subespaço (vetorial) de  $\mathbb{R}^2$  e consiste de uma reta. Esta reta pode ser identificada com a reta que passa pela origem de  $\mathbb{A}^2$  e é paralela à reta  $U$  de equação  $ax + by = c$ . Além disso, temos que

$$U = (x_0, y_0) + \mathbf{U} = \{(x_0 + u_1, y_0 + u_2) \mid (u_1, u_2) \in \mathbf{U}\},$$



em que  $(x_0, y_0)$  é um ponto qualquer de  $U$ . Este exemplo ilustra uma situação geral: todo subespaço afim  $F$  de  $\langle E, V_E, + \rangle$  pode ser caracterizado em termos de um subespaço de  $V_E$ . O subespaço  $V_F$  de  $V_E$  associado com o subespaço afim  $F$  de  $E$  é denominado *direção*. O Lema 2.2 dado a seguir estabelece e prova esta caracterização. A dimensão do subespaço  $F$  é a dimensão de  $V_F$ . Um subespaço de dimensão 1 é chamado de *reta* e um subespaço de dimensão 2, de *plano*. Finalmente, se  $F$  tem codimensão 1, então  $F$  é dito um *hiperplano*.

**Lema 2.2.** *Seja  $\langle E, V_E, + \rangle$  um espaço afim. Então,*

1. *Um subconjunto,  $V$ , não vazio de  $E$  é um subespaço afim se, e somente se, para todo ponto  $a \in V$ , o conjunto  $V_a$  definido como*

$$V_a = \{\vec{ax} \mid x \in V\}$$

*é um subespaço de  $V_E$ . Consequentemente,  $V = a + V_a$ . Além disso,*

$$\mathbf{V} = \{\vec{xy} \mid x, y \in V\}$$

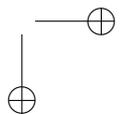
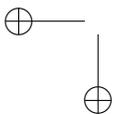
*é um subespaço de  $V_E$  e  $V_a = \mathbf{V}$ , para todo  $a \in E$ . Logo, temos*

$$V = a + \mathbf{V}.$$

2. *Para qualquer subespaço  $\mathbf{V}$  de  $V_E$ , para qualquer  $a \in E$ , o conjunto  $V = a + \mathbf{V}$  é um subespaço afim de  $E$  cuja direção é  $\mathbf{V}$ .*

*Demonstração.* Considere a afirmação 1. Suponha que  $V$  é um subespaço afim não vazio de  $E$ . Seja  $A$  um ponto qualquer de  $V$ . Vamos mostrar primeiramente que o subconjunto  $V_a$  de  $V_E$  é um subespaço de  $V_E$ . Como  $a \in V$ , tem-se, por definição de  $V_a$ , que  $\mathbf{0} = \vec{aa} \in V_a$ . Dada qualquer lista  $((p_1, \alpha_1), \dots, (p_n, \alpha_n))$  de  $n$  pontos ponderados em  $V$ , afirmamos que o vetor  $\sum_{i=1}^n \alpha_i \cdot \vec{ap_i}$  está em  $V_a$ . De fato, o ponto

$$x = \sum_{i=1}^n \alpha_i \cdot p_i + \left(1 - \sum_{i=1}^n \alpha_i\right) \cdot a$$



é o baricentro da lista  $((p_1, \alpha_1), \dots, (p_n, \alpha_n), (a, 1 - \sum_{i=1}^n \alpha_i))$  de pontos ponderados e, como  $V$  é um espaço afim, o ponto  $x$  está em  $V$ . Mas,

$$x = a + \vec{ax}$$

e

$$\begin{aligned} x &= \sum_{i=1}^n \alpha_i \cdot p_i + \left(1 - \sum_{i=1}^n \alpha_i\right) \cdot a \\ &= a + \sum_{i=1}^n \alpha_i \cdot \vec{ap}_i + \left(1 - \sum_{i=1}^n \alpha_i\right) \cdot \vec{aa} \\ &= a + \sum_{i=1}^n \alpha_i \cdot \vec{ap}_i. \end{aligned}$$

Logo,

$$\vec{ax} = \sum_{i=1}^n \alpha_i \cdot \vec{ap}_i,$$

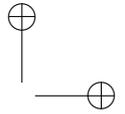
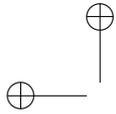
pois  $\vec{ax}$  é único (ver Definição 2.1(3)). Como  $\vec{ax} = \sum_{i=1}^n \alpha_i \cdot \vec{ap}_i$  pertence a  $V_a$ , podemos concluir que  $V_a$  é um subespaço de  $V_E$ . Para provar a recíproca, assumamos que  $V_a$  é um subespaço de  $V_E$  e considere a lista  $((p_1, \beta_1), \dots, (p_n, \beta_n))$  de pontos ponderados, com  $\sum_{i=1}^n \beta_i = 1$ , em que  $p_i \in V$  para todo  $i = 1, \dots, n$ . Do Lema 2.1, temos

$$x = \sum_{i=1}^n \beta_i \cdot p_i = a + \sum_{i=1}^n \beta_i \cdot \vec{ap}_i.$$

Mas,

$$x = a + \vec{ax},$$

com  $\vec{ax} \in V_E$ . Pela unicidade de  $\vec{ax}$ , temos que  $\vec{ax} = \sum_{i=1}^n \beta_i \cdot \vec{ap}_i$  e, como  $\vec{ap}_i \in V_a$ , podemos concluir que  $\vec{ax} \in V_a$ . Consequentemente, o ponto  $x$  está em  $V$  e, portanto,  $V$  é um subespaço afim não vazio de  $E$ . Por definição,  $V_a \subseteq V$ . Reciprocamente, sejam  $x$  e  $y$  dois pontos quaisquer de  $V$ . Então,  $\vec{xy} = \vec{xa} + \vec{ay} = -\vec{ax} + \vec{ay}$ . Como  $\vec{ax}, \vec{ay} \in V_a$  e  $V_a$  é um subespaço, temos que  $\vec{xy} \in V_a$ . Logo,  $V \subseteq V_a$  e, portanto,  $V_a = V$ , para qualquer escolha de  $a$  em  $V$ .



Para provar a afirmação 2, note que se  $V = a + \mathbf{V}$ , em que  $\mathbf{V}$  é um subespaço de  $V_E$ , então, para qualquer lista de pontos ponderados,  $((a + \mathbf{v}_i, \alpha_i))_{i=1}^n$ , com  $\mathbf{v}_i \in \mathbf{V}$  e  $\sum_{i=1}^n \alpha_i = 1$ , o baricentro  $x$  dado por

$$x = \sum_{i=1}^n \alpha_i \cdot (a + \mathbf{v}_i) = a + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{a(a + \mathbf{v}_i)} = a + \sum_{i=1}^n \alpha_i \cdot \mathbf{v}_i$$

está em  $V$ , pois  $\mathbf{V}$  é um subespaço de  $V_E$  e, portanto,  $\sum_{i=1}^n \alpha_i \cdot \mathbf{v}_i \in \mathbf{V}$ .  $\square$

## 2.5 Independência afim e referencial afim

**Definição 2.4.** Dado um espaço afim,  $\langle E, V_E, + \rangle$ , um conjunto de pontos,  $\{p_0, \dots, p_n\}$ , de  $E$  é dito ser afimemente independente se o conjunto

$$\{\overrightarrow{p_i p_j} \mid j \in \{0, \dots, n\} - \{i\}\},$$

consistindo de  $n$  vetores de  $V_E$ , for linearmente independente (LI) para algum  $i \in \{0, \dots, n\}$ . Caso contrário, ele é dito afimemente dependente.

A Figura 2.5 contém um subconjunto afimemente independente de três pontos,  $p_0, p_1$  e  $p_2$ , de  $\mathbb{A}^2$  e um subconjunto afimemente dependente de três pontos,  $q_0, q_1$  e  $q_2$ , de  $\mathbb{A}^2$ . Note que  $q_0, q_1$  e  $q_2$  são colineares.

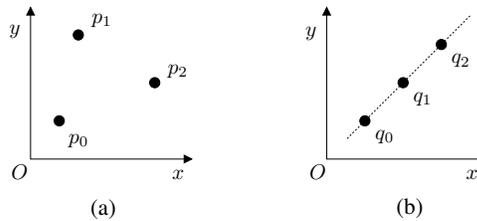
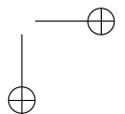
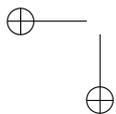


Figura 2.5: (a) Um conjunto de pontos afimemente independente e (b) um conjunto de pontos afimemente dependente, ambos com três pontos.

O lema a seguir caracteriza conjuntos afimemente independentes:



**Lema 2.3.** *Dado um espaço afim,  $E$ , seja  $(p_i)_{i=0}^n$  uma lista de  $n+1$  pontos em  $E$ . Seja  $x$  um ponto de  $E$  tal que  $x = \sum_{i=0}^n \alpha_i \cdot p_i$ , em que  $\sum_{i=0}^n \alpha_i = 1$ . Então, a lista  $(\alpha_i)_{i=0}^n$ , tal que  $x = \sum_{i=0}^n \alpha_i \cdot p_i$ , é única se, e somente se, os vetores em  $(\overrightarrow{p_0 p_i})_{i=1}^n$  formam um conjunto LI.*

*Demonstração.* Sabemos que

$$x = \sum_{i=0}^n \alpha_i \cdot p_i \quad \text{se, e somente se,} \quad \overrightarrow{p_0 x} = \sum_{i=1}^n \alpha_i \cdot \overrightarrow{p_0 p_i},$$

em que  $\sum_{i=0}^n \alpha_i = 1$ . Mas, é bem sabido (da Álgebra Linear) que a lista  $(\alpha_1, \dots, \alpha_n)$ , tal que  $\overrightarrow{p_0 x} = \sum_{i=1}^n \alpha_i \cdot \overrightarrow{p_0 p_i}$ , é única se, e somente se, os vetores em  $(\overrightarrow{p_0 p_i})_{i=1}^n$  formam um conjunto LI. Portanto, se os vetores em  $(\overrightarrow{p_0 p_i})_{i=1}^n$  formam um conjunto LI, a lista  $(\alpha_1, \dots, \alpha_n)$  é única e, dado que  $\alpha_0 = 1 - \sum_{i=1}^n \alpha_i$ , o número real  $\alpha_0$  também é único. Reciprocamente, a unicidade de  $(\alpha_i)_{i=0}^n$  tal que  $x = \sum_{i=0}^n \alpha_i \cdot p_i$  implica a unicidade da lista de vetores  $(\overrightarrow{p_0 p_i})_{i=1}^n$  tal que  $\overrightarrow{p_0 x} = \sum_{i=1}^n \alpha_i \cdot \overrightarrow{p_0 p_i}$ . Logo, os vetores em  $(\overrightarrow{p_0 p_i})_{i=1}^n$  formam um conjunto LI.  $\square$

O lema abaixo caracteriza conjuntos afimemente dependentes:

**Lema 2.4.** *Dado um espaço afim,  $\langle E, V_E, + \rangle$ , seja  $(p_i)_{i=0}^n$  uma lista de  $n+1$  pontos em  $E$ . Os pontos em  $(p_i)_{i=0}^n$  formam um conjunto afimemente dependente se, e somente se, houver uma lista  $(\alpha_i)_{i=0}^n$  de escalares, com  $\alpha_j \neq 0$  para algum  $j \in \{0, 1, \dots, n\}$  e  $\sum_{i=0}^n \alpha_i = 0$ , tal que*

$$\sum_{i=0}^n \alpha_i \cdot \overrightarrow{x p_i} = \mathbf{0},$$

para todo  $x \in E$ .

*Demonstração.* Pelo Lema 2.3, a lista  $(p_i)_{i=0}^n$  é afimemente dependente se, e somente se, os vetores da lista  $(\overrightarrow{p_i p_j})_{j=0, j \neq i}^n$  formam um conjunto linearmente dependente (LD), para algum  $i \in \{0, 1, \dots, n\}$ . Mas, para qualquer  $k \in \{0, 1, \dots, n\}$ , o conjunto formado pelos vetores em  $(\overrightarrow{p_k p_j})_{j=0, j \neq k}^n$  é LD se, e somente se, houver uma lista  $(\alpha_j)_{j=0, j \neq k}^n$  de escalares em  $\mathbb{R}$ , com  $\alpha_j \neq 0$  para algum  $j \in \{0, 1, \dots, n\} - \{k\}$ , tal

que

$$\sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j} = \mathbf{0}.$$

Logo,

$$\begin{aligned} \sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j} &= \sum_{j=0, j \neq i}^n \alpha_j \cdot (\overline{x p_j} - \overline{x p_k}) \\ &= \sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{x p_j} - \left( \sum_{j=0, j \neq k}^n \alpha_j \right) \cdot \overline{x p_k}, \end{aligned}$$

para qualquer ponto  $x$  em  $E$ . Fazendo  $\alpha_k = -(\sum_{j=0, j \neq k}^n \alpha_j)$ , temos

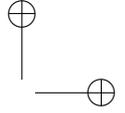
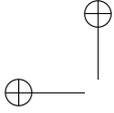
$$\sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j} = \sum_{i=0}^n \alpha_i \cdot \overline{x p_i},$$

com  $\sum_{i=0}^n \alpha_i = 0$  e  $\alpha_j \neq 0$  para algum  $j \in \{0, 1, \dots, n\}$ . Logo, se o conjunto formado pelos vetores em  $(\overline{p_k p_j})_{j=0, j \neq k}^n$  é LD, então existe  $j \in \{0, 1, \dots, n\} - \{k\}$  tal que  $\alpha_j \neq 0$  e  $\sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j} = \mathbf{0}$ , o que implica que  $\sum_{i=0}^n \alpha_i \cdot \overline{x p_i} = \mathbf{0}$ . Reciprocamente, suponha que haja uma lista  $(\alpha_i)_{i=0}^n$ , com  $\alpha_j \neq 0$  para algum  $j \in \{0, 1, \dots, n\}$  e  $\sum_{i=0}^n \alpha_i = 0$ , tal que  $\sum_{i=0}^n \alpha_i \cdot \overline{x p_i} = \mathbf{0}$ , para todo  $x$  em  $E$ . Como  $\sum_{i=0}^n \alpha_i = 0$  e  $\alpha_j \neq 0$ , deve haver um  $k$  em  $\{0, 1, \dots, n\} - \{j\}$ . Daí, podemos substituir  $x$  por  $p_k$  em  $\sum_{i=0}^n \alpha_i \cdot \overline{x p_i}$  de modo que  $\sum_{i=0}^n \alpha_i \cdot \overline{x p_i} = \sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j}$ . Logo,  $\sum_{j=0, j \neq k}^n \alpha_j \cdot \overline{p_k p_j} = \mathbf{0}$ .  $\square$

O Lema 2.3 sugere a noção de *referencial afim*:

**Definição 2.5.** *Dado um espaço afim,  $\langle E, V_E, + \rangle$ , um referencial afim com origem  $p_0$  é uma lista  $(p_i)_{i=0}^n$  de  $n + 1$  pontos em  $E$  tais que os vetores  $\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}$  formam uma base do espaço vetorial  $V_E$  associado a  $E$ . O par  $(p_0, (\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}))$  é também chamado de referencial afim com origem  $p_0$ . Então, todo ponto  $x \in E$  é dado por*

$$x = p_0 + x_1 \cdot \overrightarrow{p_0 p_1} + \dots + x_n \cdot \overrightarrow{p_0 p_n},$$



em que  $(x_i)_{i=1}^n$  é uma lista única de escalares que são denominados coordenadas de  $x$  com relação ao referencial afim  $(p_0, (\overrightarrow{p_0p_1}, \dots, \overrightarrow{p_0p_n}))$ . Além do mais, todo ponto  $x \in E$  pode também ser dado por

$$x = \alpha_0 \cdot p_0 + \dots + \alpha_n \cdot p_n,$$

em que  $(\alpha_i)_{i=0}^n$  é uma lista única de escalares satisfazendo  $\sum_{i=0}^n \alpha_i = 1$ . Denominamos esses escalares de coordenadas baricêntricas de  $x$  com relação ao referencial afim  $(p_0, p_1, \dots, p_n)$ . As coordenadas  $(x_i)_{i=1}^n$  e as coordenadas baricêntricas  $(\alpha_i)_{i=0}^n$  de  $x$  estão relacionadas pelas equações  $\alpha_0 = 1 - \sum_{i=1}^n x_i$  e  $\alpha_i = x_i$ , para todo  $i = 1, \dots, n$ .

A Figura 2.6 ilustra quatro referenciais afins.

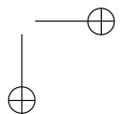
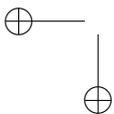
O primeiro referencial afim ilustrado pela Figura 2.6 consiste do ponto  $p_0$  no espaço afim  $\langle \{p_0\}, \{\mathbf{0}\}, + \rangle$ ; o segundo referencial afim consiste de dois pontos distintos em  $\mathbb{A}$ ; o terceiro, de três pontos não colineares em  $\mathbb{A}^2$ ; e o quarto, de quatro pontos não coplanares em  $\mathbb{A}^3$ .

Uma pergunta de interesse prático é: dados um referencial afim, digamos  $(p_0, p_1, p_2)$ , com origem  $p_0$  em  $\mathbb{A}^2$  e um ponto  $q \in \mathbb{A}^2$  qualquer, quais são as coordenadas baricêntricas de  $q$  com relação a  $(p_0, p_1, p_2)$ ? Para determinar tais coordenadas baricêntricas, suponha que as coordenadas cartesianas de  $p_0, p_1$  e  $p_2$  e  $q$  sejam fornecidas com relação a um sistema de eixos ortogonais em  $\mathbb{A}^2$  qualquer:  $p_0 = (x_0, y_0)$ ,  $p_1 = (x_1, y_1)$ ,  $p_2 = (x_2, y_2)$  e  $q = (x, y)$ . Logo,

$$\begin{cases} x_0 \cdot \alpha_0 + x_1 \cdot \alpha_1 + x_2 \cdot \alpha_2 = x \\ y_0 \cdot \alpha_0 + y_1 \cdot \alpha_1 + y_2 \cdot \alpha_2 = y \\ \alpha_0 + \alpha_1 + \alpha_2 = 1 \end{cases}$$

em que  $(\alpha_0, \alpha_1, \alpha_2)$  são as coordenadas baricêntricas de  $q$  com relação ao referencial  $(p_0, p_1, p_2)$ . As três equações acima formam um sistema com três equações afins em três incógnitas,  $\alpha_0, \alpha_1$  e  $\alpha_2$ , o qual possui uma única solução, pois  $(p_0, p_1, p_2)$  é um referencial em  $\mathbb{A}^2$ .

## 2.6 Mapa afim



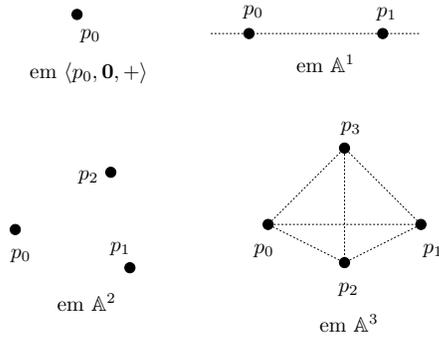


Figura 2.6: Referenciais afins em  $\langle \{p_0\}, \{\mathbf{0}\}, + \rangle$ ,  $\mathbb{A}$ ,  $\mathbb{A}^2$  e  $\mathbb{A}^3$ .

**Definição 2.6.** *Dados dois espaços afins,  $\langle E, V_E, + \rangle$  e  $\langle E', V_{E'}, +' \rangle$ , uma função  $f : E \rightarrow E'$  é um mapa afim se, e somente se, para toda lista  $((p_i, \alpha_i))_{i=1}^n$  de pontos ponderados em  $E$  tal que  $\sum_{i=1}^n \alpha_i = 1$ , temos*

$$f \left( \sum_{i=1}^n \alpha_i \cdot p_i \right) = \sum_{i=1}^n \alpha_i \cdot f(p_i).$$

*Em outras palavras, a função  $f$  preserva combinações baricêntricas.*

**Exemplo 2.3.** *Seja*

$$f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$$

*tal que*

$$f(p) = f((x, y)) = (x + 2y + 3, y + 1),$$

*para todo ponto  $p \in \mathbb{A}^2$ , em que  $(x, y)$  são as coordenadas de  $p$  com relação a um referencial afim fixado,  $(O, \mathbf{i}, \mathbf{j})$ ; isto é,  $p = O + x \cdot \mathbf{i} + y \cdot \mathbf{j}$ . É fácil mostrar que  $f$  é um mapa afim. Para tal, devemos verificar que, para toda lista*

$$((p_i, \alpha_i))_{i=1}^n$$

*de pontos ponderados em  $\mathbb{A}^2$ , com  $\sum_{i=1}^n \alpha_i = 1$ , temos que*

$$f \left( \sum_{i=1}^n \alpha_i \cdot p_i \right) = \sum_{i=1}^n \alpha_i \cdot f(p_i).$$

Mapas afins podem ser obtidos a partir de mapas lineares como segue:

**Lema 2.5.** *Sejam  $\langle E, V_E, + \rangle$  e  $\langle E', V_{E'}, +' \rangle$  dois espaços afins. Dados um ponto  $a$  em  $E$ , um ponto  $b$  em  $E'$  e qualquer mapa linear  $h : V_E \rightarrow V_{E'}$ , o mapa  $f : E \rightarrow E'$  definido abaixo para todo  $\mathbf{v} \in V_E$  é afim:*

$$f(a + \mathbf{v}) = b +' h(\mathbf{v}).$$

*Demonstração.* Seja  $(\alpha_i)_{i=1}^n$ , com  $\sum_{i=1}^n \alpha_i = 1$ , uma lista qualquer de  $n$  escalares e seja  $(\mathbf{v}_i)_{i=1}^n$  uma lista qualquer de  $n$  vetores em  $V_E$ . Como

$$\sum_{i=1}^n \alpha_i \cdot (a + \mathbf{v}_i) = a + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{a(a + \mathbf{v}_i)} = a + \sum_{i=1}^n \alpha_i \cdot \mathbf{v}_i$$

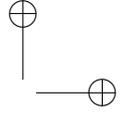
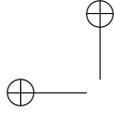
e

$$\sum_{i=1}^n \alpha_i \cdot (b +' h(\mathbf{v}_i)) = b +' \sum_{i=1}^n \alpha_i \cdot \overrightarrow{b(b +' h(\mathbf{v}_i))} = b +' \sum_{i=1}^n \alpha_i \cdot h(\mathbf{v}_i),$$

temos

$$\begin{aligned} f\left(\sum_{i=1}^n \alpha_i \cdot (a + \mathbf{v}_i)\right) &= f\left(a + \sum_{i=1}^n \alpha_i \cdot \mathbf{v}_i\right) \\ &= b +' h\left(\sum_{i=1}^n \alpha_i \cdot \mathbf{v}_i\right) \\ &= b +' \sum_{i=1}^n \alpha_i \cdot h(\mathbf{v}_i) \\ &= \sum_{i=1}^n \alpha_i \cdot (b +' h(\mathbf{v}_i)) \\ &= \sum_{i=1}^n \alpha_i \cdot f(a + \mathbf{v}_i) \end{aligned}$$

□



**Exemplo 2.4.** O mapa afim  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  do Exemplo 2.3 é tal que

$$f\left(O + \begin{bmatrix} x \\ y \end{bmatrix}\right) = (3, 1) + \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Usando a notação do Lema 2.5, temos  $a = O$ ,  $b = (3, 1)$  e  $h$  dada por

$$h(x, y) = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

em que  $\langle x, y \rangle$  são as coordenadas do vetor  $\mathbf{v} \in \mathbb{R}^2$  com relação à base  $(\mathbf{i}, \mathbf{j})$ .

Reciprocamente, todo mapa afim é determinado pela imagem de qualquer ponto e um mapa linear. O mapa linear é único e independe do ponto.

**Lema 2.6.** Sejam  $\langle E, V_E, + \rangle$  e  $\langle E', V_{E'}, +' \rangle$  dois espaços afins. Dado um mapa afim  $f : E \rightarrow E'$ , existe um único mapa linear  $\mathbf{f} : V_E \rightarrow V_{E'}$  tal que, para todo  $a \in E$  e para todo  $\mathbf{v} \in V_E$ , temos

$$f(a + \mathbf{v}) = f(a) +' \mathbf{f}(\mathbf{v}).$$

*Demonstração.* Seja  $a$  um ponto qualquer de  $E$ . Defina o mapa  $\mathbf{f} : V_E \rightarrow V_{E'}$  tal que, para todo  $\mathbf{v} \in V_E$ , temos  $\mathbf{f}(\mathbf{v}) = \overrightarrow{f(a)f(a + \mathbf{v})}$ . Vamos mostrar que  $\mathbf{f}$  é linear. Primeiro, observe que  $a + \alpha \cdot \mathbf{v}$  é tal que

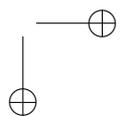
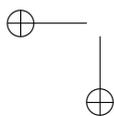
$$\begin{aligned} a + \alpha \cdot \mathbf{v} &= a + \alpha \cdot \overrightarrow{a(a + \mathbf{v})} \\ &= a + \alpha \cdot \overrightarrow{a(a + \mathbf{v})} + (1 - \alpha) \cdot \overrightarrow{aa}, \end{aligned}$$

para todo escalar  $\alpha$  no corpo sobre o qual  $V_E$  é definido. Daí, temos que

$$a + \alpha \cdot \mathbf{v} = \alpha \cdot (a + \mathbf{v}) + (1 - \alpha) \cdot a,$$

Como  $f$  é um mapa afim, as igualdades acima nos permitem concluir que

$$\begin{aligned} f(a + \alpha \cdot \mathbf{v}) &= f(\alpha \cdot (a + \mathbf{v}) + (1 - \alpha) \cdot a) \\ &= \alpha \cdot f(a + \mathbf{v}) +' (1 - \alpha) \cdot f(a) +' (1 - \alpha) \cdot \overrightarrow{f(a)f(a)} \\ &= f(a) +' \alpha \cdot \overrightarrow{f(a)f(a + \mathbf{v})}. \end{aligned}$$



Logo,

$$\overrightarrow{f(a)f(a + \alpha \cdot \mathbf{v})} = \alpha \cdot \overrightarrow{f(a)f(a + \mathbf{v})}.$$

Daí,

$$\mathbf{f}(\alpha \cdot \mathbf{v}) = \alpha \cdot \mathbf{f}(\mathbf{v}).$$

Agora, se  $\mathbf{u}$  é um outro vetor qualquer de  $V_E$ , então temos que

$$a + \mathbf{u} + \mathbf{v} = (a + \mathbf{u}) + (a + \mathbf{v}) - 1 \cdot a,$$

pois

$$a + \mathbf{u} + \mathbf{v} = a + \overrightarrow{a(a + \mathbf{u})} + \overrightarrow{a(a + \mathbf{v})} - \overrightarrow{aa}.$$

Como  $f$  é um mapa afim, as igualdades acima nos permitem concluir que

$$f(a + \mathbf{u} + \mathbf{v}) = f(a + \mathbf{u}) +' f(a + \mathbf{v}) - f(a),$$

o que implica que

$$\begin{aligned} \overrightarrow{f(a)f(a + \mathbf{u} + \mathbf{v})} &= \overrightarrow{f(a)f(a + \mathbf{u})} +' \overrightarrow{f(a)f(a + \mathbf{v})} - \overrightarrow{f(a)f(a)} \\ &= \overrightarrow{f(a)f(a + \mathbf{u})} +' \overrightarrow{f(a)f(a + \mathbf{v})}. \end{aligned}$$

Logo,

$$\mathbf{f}(\mathbf{u} + \mathbf{v}) = \mathbf{f}(\mathbf{u}) +' \mathbf{f}(\mathbf{v}),$$

e, portanto, o mapa  $\mathbf{f}$  é linear. Seja  $b$  é qualquer outro ponto em  $E$ . Como

$$b + \mathbf{v} = a + \overrightarrow{ab} + \mathbf{v} = a + \overrightarrow{a(a + \mathbf{v})} - \overrightarrow{aa} + \overrightarrow{ab},$$

temos

$$b + \mathbf{v} = (a + \mathbf{v}) - a + b,$$

e, como  $f$  é afim,

$$f(b + \mathbf{v}) = f(a + \mathbf{v}) -' f(a) +' f(b),$$

o que implica que

$$\begin{aligned} \overrightarrow{f(b)f(b + \mathbf{v})} &= \overrightarrow{f(b)f(a + \mathbf{v})} -' \overrightarrow{f(b)f(a)} +' \overrightarrow{f(b)f(b)} \\ &= \overrightarrow{f(a)f(b)} +' \overrightarrow{f(b)f(a + \mathbf{v})} \\ &= \overrightarrow{f(a)f(a + \mathbf{v})}. \end{aligned}$$

Logo, a definição de  $\mathbf{f}$  não depende da escolha do ponto  $a$ . Finalmente, o fato que  $\mathbf{f}$  é único é óbvio, pois se  $f(a + \mathbf{v}) = f(a) + \mathbf{f}(\mathbf{v})$ , então  $\mathbf{f}(\mathbf{v}) = \overrightarrow{f(a)f(a + \mathbf{v})}$  é o único (ver condição 3 da Definição 2.1).  $\square$

O mapa linear  $\mathbf{f}$  do Lema 2.6 é denominado *mapa linear associado com o mapa afim  $f$* . Um mapa afim,  $f : E \rightarrow E$ , do qual o mapa linear associado,  $\mathbf{f} : V_E \rightarrow V_E$ , é a identidade é denominado *translação*. Usando referenciais afins e o Lema 2.6 é possível representar mapas afins em termos de matrizes, como no Exemplo 2.4. Sejam  $\langle E, V_E, + \rangle$  um espaço afim,  $f : E \rightarrow E$  um mapa afim e  $(p_0, p_1, \dots, p_n)$  um referencial afim em  $E$ . Então, para todo  $\mathbf{v} \in V_E$ , temos

$$f(p_0 + \mathbf{x}) = f(p_0) + \mathbf{f}(\mathbf{v}),$$

em que  $\mathbf{f} : V_E \rightarrow V_E$  é um mapa linear. Seja  $\mathbf{a} = [\alpha_{i,j}]$  a matriz  $n \times n$  do mapa linear  $\mathbf{f}$  com relação à base  $(\overrightarrow{p_0p_1}, \dots, \overrightarrow{p_0p_n})$ . Com relação a esta base, assumamos que  $\overrightarrow{p_0f(p_0)}$ ,  $\overrightarrow{p_0f(p_0 + \mathbf{v})}$  e  $\mathbf{v}$  são dados por

$$\begin{aligned} \overrightarrow{p_0f(p_0)} &= \beta_1 \cdot \overrightarrow{p_0p_1} + \dots + \beta_n \cdot \overrightarrow{p_0p_n}, \\ \overrightarrow{p_0f(p_0 + \mathbf{v})} &= \alpha_1 \cdot \overrightarrow{p_0p_1} + \dots + \alpha_n \cdot \overrightarrow{p_0p_n}, \\ \mathbf{v} &= \nu_1 \cdot \overrightarrow{p_0p_1} + \dots + \nu_n \cdot \overrightarrow{p_0p_n}. \end{aligned}$$

Como  $f(p_0 + \mathbf{v}) = f(p_0) + \mathbf{f}(\mathbf{v})$  implica que  $\overrightarrow{p_0 + p_0f(p_0 + \mathbf{v})} = \overrightarrow{(p_0 + p_0f(p_0)) + \mathbf{f}(\mathbf{v})}$  e, portanto, que  $\overrightarrow{p_0f(p_0 + \mathbf{v})} = \overrightarrow{p_0f(p_0)} + \mathbf{f}(\mathbf{v})$ , temos

$$\begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} + \mathbf{A} \cdot \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_n \end{bmatrix}$$

e, portanto, o mapa  $f$  pode ser escrito na forma matricial  $\mathbf{y} = \mathbf{Ax} + \mathbf{b}$ , em que  $\mathbf{y}$ ,  $\mathbf{x}$  e  $\mathbf{b}$  são vetores colunas com as coordenadas de  $\overrightarrow{p_0f(p_0 + \mathbf{v})}$ ,  $\mathbf{v}$  e  $\overrightarrow{p_0f(p_0)}$ , respectivamente, com relação à base  $(\overrightarrow{p_0p_1}, \dots, \overrightarrow{p_0p_n})$  de  $V_E$ . O Problema 2.8 pede para você mostrar que o mesmo resultado vale para o caso geral em que  $f$  é um mapa entre dois espaços afins distintos, digamos  $\langle E, V_E, + \rangle$  e  $\langle F, V_F, +' \rangle$ . Neste caso, devemos escolher um referencial afim  $(p_0, p_1, \dots, p_n)$  em  $E$  e outro,  $(q_0, q_1, \dots, q_m)$ , em  $F$ . A matriz  $\mathbf{A}$  é a matriz do mapa linear  $\mathbf{f}$  associado a  $f$  com relação a duas bases: uma de  $V_E$  e outra de  $V_F$ .

## 2.7 Forma (ou funcional) afim

No Exemplo 2.2, observamos que o conjunto  $U$  de soluções de uma equação da forma  $ax + by = c$  é um subespaço afim de  $\mathbb{A}^2$  de dimensão 1 — de fato, uma reta (desde que  $a$  e  $b$  não sejam ambos iguais a zero). É fácil mostrar que o conjunto  $P$  de soluções de uma equação da forma  $ax + by + cz = d$  é um subespaço afim de  $\mathbb{A}^3$  de dimensão 2 — de fato, um plano (desde que  $a$ ,  $b$  e  $c$  não sejam todos iguais a zero). Em geral, o conjunto  $H$  de soluções de uma equação da forma

$$\alpha_1 x_1 + \cdots + \alpha_d x_d = \mu \tag{2.3}$$

é um subespaço afim de  $\mathbb{A}^d$ , e se  $\alpha_1, \dots, \alpha_d$  não são todos iguais a zero, este subespaço possui dimensão  $d-1$  e é denominado *hiperplano*. Nós podemos interpretar a Equação (2.3) em termos do mapa  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  tal que

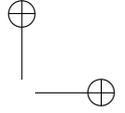
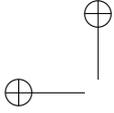
$$f(x_1, \dots, x_d) = \alpha_1 x_1 + \cdots + \alpha_d x_d - \mu, \tag{2.4}$$

para todo  $(x_1, \dots, x_d) \in \mathbb{R}^d$ . É possível mostrar que  $f$  é afim e que o conjunto  $H$  de soluções da Equação (2.3) é o conjunto de nível zero, ou *núcleo*, de  $f$ , pois  $H = f^{-1}(0) = \{X \in \mathbb{A}^d \mid f(x_1, \dots, x_d) = 0\}$ , em que  $(x_1, \dots, x_d)$  são as coordenadas de  $X$  com relação a um referencial afim.

As observações sobre  $H$  realizadas acima nos dizem que mapas afins  $f : E \rightarrow \mathbb{R}$  de um espaço afim  $E$  em  $\mathbb{R}$  são interessantes. Tais mapas são denominados *formas afins*. Em contraste com as formas lineares,  $\mathbf{f} : V \rightarrow \mathbb{R}$ , para as quais o núcleo,  $Nuc(\mathbf{f})$ , não pode ser o conjunto vazio (pois,  $Nuc(\mathbf{f})$  sempre contém o vetor nulo,  $\mathbf{0}$ ), é possível termos  $f^{-1}(0) = \emptyset$  para uma forma afim  $f$ . Dado um mapa afim  $f : E \rightarrow \mathbb{R}$ , denotamos o núcleo de  $f$ ,  $f^{-1}(0)$ , por  $Nuc(f)$ . Pelo que vimos, um hiperplano é um subespaço afim de codimensão 1. A relação entre hiperplanos e formas afins é descrita pelo lema dado a seguir:

**Lema 2.7.** *Seja  $\langle E, V_E, + \rangle$  um espaço afim. Então, as seguintes afirmações são válidas:*

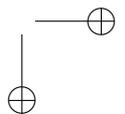
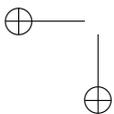
1. *Seja  $f : E \rightarrow \mathbb{R}$  uma forma afim qualquer tal que  $f$  não é uma função constante. Então, o núcleo,  $Nuc(f)$ , de  $f$  é um hiperplano.*



2. Para qualquer hiperplano  $H$  em  $E$ , existe uma forma afim,  $f : E \rightarrow \mathbb{R}$ , tal que  $f$  não é constante e  $H = \text{Nuc}(f)$ . Além disso, se  $g : E \rightarrow \mathbb{R}$  é qualquer outra forma afim tal que  $H = \text{Nuc}(g)$ , então existe um número real  $\alpha$ , com  $\alpha \neq 0$ , tal que  $g$  é a função  $\alpha f$ .
3. Dados qualquer hiperplano  $H$  em  $E$  e qualquer forma afim  $f : E \rightarrow \mathbb{R}$  tal que  $H = \text{Nuc}(f)$ , todo hiperplano  $H'$  paralelo a  $H$  é tal que  $H' = \text{Nuc}(g)$ , em que  $g : E \rightarrow \mathbb{R}$  é uma forma afim dada por  $g(a) = f(a) - \alpha$ , para todo ponto  $a$  em  $E$  e algum  $\alpha$  em  $\mathbb{R}$ .

A prova que damos a seguir para o Lema 2.7 depende de alguns resultados clássicos sobre formas lineares estudados em Álgebra Linear (ver Capítulo 4 do livro [26], por exemplo). Em particular, se  $\mathbf{E}$  é um espaço vetorial sobre um corpo  $\mathbb{K}$ , então as seguintes três afirmações são válidas: (a) se  $\mathbf{f} : \mathbf{E} \rightarrow \mathbb{K}$  é uma forma linear não nula, então o núcleo  $\text{Nuc}(\mathbf{f})$  de  $\mathbf{f}$  é um hiperplano em  $\mathbf{E}$ ; (b) para qualquer hiperplano  $\mathbf{H}$  em  $\mathbf{E}$ , existe uma forma linear não nula,  $\mathbf{f} : \mathbf{E} \rightarrow \mathbb{K}$ , tal que  $\mathbf{H} = \text{Nuc}(\mathbf{f})$ ; (c) dados qualquer hiperplano  $\mathbf{H}$  em  $\mathbf{E}$  e qualquer forma linear não nula  $\mathbf{f} : \mathbf{E} \rightarrow \mathbb{K}$  tal que  $\mathbf{H} = \text{Nuc}(\mathbf{f})$ , para toda forma linear  $\mathbf{g} : \mathbf{E} \rightarrow \mathbb{K}$ , temos que  $\mathbf{H} = \text{Nuc}(\mathbf{g})$  se, e somente se,  $\mathbf{g} = \alpha \mathbf{f}$ , para algum  $\alpha \in \mathbb{K}$ , com  $\alpha \neq 0$ . Agora, vamos à prova do Lema 2.7:

*Demonstração.* (1) Pelo Lema 2.6, há um único mapa linear,  $\mathbf{f} : V_E \rightarrow \mathbb{R}$ , tal que  $f(a + \mathbf{v}) = f(a) + \mathbf{f}(\mathbf{v})$ , para qualquer escolha de  $a$  em  $E$ . Logo, se  $f : E \rightarrow \mathbb{R}$  não é uma função constante, o mapa linear,  $\mathbf{f}$ , associado a  $f$  não pode ser uma função constante. Como  $\mathbf{f}$  é uma forma linear, o núcleo,  $\text{Nuc}(\mathbf{f})$ , de  $\mathbf{f}$  é um hiperplano  $\mathbf{H}$  em  $V_E$ . Consequentemente, o conjunto imagem,  $\text{Im}(\mathbf{f})$ , de  $\mathbf{f}$  tem dimensão 1 e, portanto,  $\mathbf{f}$  é sobrejetora. Como  $f(a + \mathbf{v}) = f(a) + \mathbf{f}(\mathbf{v})$ , a função  $f$  também é sobrejetora. Então, existe um ponto  $x \in E$  tal que  $f(x) = 0$ . Tomando  $a = x$  no Lema 2.6, temos que  $f(x + \mathbf{v}) = f(x) + \mathbf{f}(\mathbf{v}) = \mathbf{f}(\mathbf{v})$ , para todo  $\mathbf{v} \in V_E$ . Logo, podemos concluir que  $f(x + \mathbf{v}) = 0$  se, e somente se,  $\mathbf{f}(\mathbf{v}) = 0$ . Mas, como  $\mathbf{H} = \text{Nuc}(\mathbf{f})$ , o conjunto de pontos  $f^{-1}(0)$ , que é igual a  $x + \mathbf{H}$ , é um hiperplano  $H$  em  $E$ .



(2) Se  $H$  é um hiperplano (afim) de  $V_E$ , então o Lema 2.2 nos diz que  $H = a + \mathbf{H}$ , para algum  $a \in E$ , em que  $\mathbf{H}$  é um hiperplano em  $V_E$ . Como mencionado anteriormente,  $\mathbf{H} = Nuc(\mathbf{f})$ , para alguma forma linear,  $\mathbf{f} : V_E \rightarrow \mathbb{R}$ , não nula. Vamos definir  $f : E \rightarrow \mathbb{R}$  como  $f(a + \mathbf{v}) = \mathbf{f}(\mathbf{v})$ , para todo  $\mathbf{v} \in V_E$ . Observe que  $H = f^{-1}(0)$  e que  $f$  não é uma função constante, pois  $\mathbf{f}$  não é nula. Suponha que  $g : E \rightarrow \mathbb{R}$  é outra forma afim tal que  $H = Nuc(g)$ . O Lema 2.6 nos diz que  $g(a + \mathbf{v}) = g(a) + \mathbf{g}(\mathbf{v})$ , para todo  $\mathbf{v} \in V_E$ , em que  $\mathbf{g} : V_E \rightarrow \mathbb{R}$  é uma forma linear única. Como  $a \in H$  e  $H = g^{-1}(0)$ , podemos concluir que  $g(a) = 0$  e, portanto,  $g(a + \mathbf{v}) = \mathbf{g}(\mathbf{v})$ . Como  $H = Nuc(g)$  e  $H = a + \mathbf{H}$ , o mapa linear  $\mathbf{g}$  não pode ser constante e  $\mathbf{H} = Nuc(\mathbf{g})$ . Note que  $V_E = \mathbf{H} \oplus \mathbf{F}$ , em que  $\mathbf{F} = \{\lambda \cdot \mathbf{v}_0 \mid \lambda \in \mathbb{R}\}$  para algum vetor  $\mathbf{v}_0 \in V_E$ , com  $\mathbf{v}_0 \notin \mathbf{H}$ . Note também que  $\mathbf{f}(\mathbf{v}_0) \neq 0$  e  $\mathbf{g}(\mathbf{v}_0) \neq 0$ . Então, considere a forma linear,  $h : E \rightarrow \mathbb{R}$ , definida como

$$h(a + \mathbf{v}) = g(a + \mathbf{v}) - \left( \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)} \right) \cdot f(a + \mathbf{v}) = \mathbf{g}(\mathbf{v}) - \left( \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)} \right) \cdot \mathbf{f}(\mathbf{v}),$$

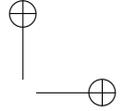
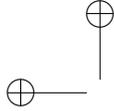
para todo  $\mathbf{v} \in V_E$ . Observe que  $h(a + \mathbf{w}) = 0$ , para todo  $\mathbf{w} \in \mathbf{H}$ , pois  $g(a + \mathbf{w}) = f(a + \mathbf{w}) = 0$ . Por sua vez, se  $\mathbf{w} \in \mathbf{F}$ , então temos que

$$\begin{aligned} h(a + \mathbf{w}) &= \mathbf{g}(\mathbf{w}) - \left( \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)} \right) \cdot \mathbf{f}(\mathbf{w}) \\ &= \mathbf{g}(\lambda \cdot \mathbf{v}_0) - \left( \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)} \right) \cdot \mathbf{f}(\lambda \cdot \mathbf{v}_0) \\ &= \lambda \cdot \mathbf{g}(\mathbf{v}_0) - \lambda \cdot \left( \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)} \right) \cdot \mathbf{f}(\mathbf{v}_0) \\ &= \lambda \cdot \mathbf{g}(\mathbf{v}_0) - \lambda \cdot \mathbf{g}(\mathbf{v}_0) \\ &= 0, \end{aligned}$$

em que  $\mathbf{w} = \lambda \cdot \mathbf{v}_0$ , para algum  $\lambda \in \mathbb{R}$ . Logo,  $h(a + \mathbf{v}) = 0$  para todo  $\mathbf{v} \in V_E$  e, portanto,  $g(a + \mathbf{v}) = \alpha \cdot f(a + \mathbf{v})$ , para todo  $\mathbf{v} \in V_E$ , em que

$$\alpha = \frac{\mathbf{g}(\mathbf{v}_0)}{\mathbf{f}(\mathbf{v}_0)}.$$

(3) Da afirmação (2), sabemos que existe uma forma afim não constante,  $h : E \rightarrow \mathbb{R}$ , tal que  $H' = Nuc(h)$ . Como  $H$  e  $H'$  são



paralelos, o Lema 2.2 nos diz que  $H = a + \mathbf{H}$  e  $H' = b + \mathbf{H}$ , em que  $a \in H$ ,  $b \in H'$  e  $\mathbf{H}$  é um hiperplano em  $V_E$ . Pelo Lema 2.6, temos que  $f(a + \mathbf{v}) = \mathbf{f}(\mathbf{v})$  e  $h(b + \mathbf{v}) = \mathbf{h}(\mathbf{v})$ , para todo  $\mathbf{v} \in V_E$ , em que  $\mathbf{f} : V_E \rightarrow \mathbb{R}$  e  $\mathbf{h} : V_E \rightarrow \mathbb{R}$  são formas lineares únicas. Como  $f(a + \mathbf{v}) = h(b + \mathbf{v}) = 0$  se, e somente se, o vetor  $\mathbf{v}$  pertence ao hiperplano  $\mathbf{H}$ , temos que  $\mathbf{f}(\mathbf{v}) = \mathbf{h}(\mathbf{v}) = 0$  se, e somente se,  $\mathbf{v} \in \mathbf{H}$ . Logo,  $Nuc(\mathbf{f}) = Nuc(\mathbf{h}) = \mathbf{H}$  e, portanto, podemos concluir que  $\mathbf{h} = \lambda \mathbf{f}$ , para algum  $\lambda \in \mathbb{R}$ , com  $\lambda \neq 0$ . Da Definição 2.1(3), sabemos que há um único vetor,  $\mathbf{w} \in V_E$ , tal que  $b + \mathbf{w} = a$ . Então, para todo  $\mathbf{v} \in V_E$ ,

$$\begin{aligned} h(a + \mathbf{v}) &= h(b + \mathbf{w} + \mathbf{v}) \\ &= \mathbf{h}(\mathbf{w} + \mathbf{v}) \\ &= \lambda \cdot \mathbf{f}(\mathbf{w} + \mathbf{v}) \\ &= \lambda \cdot \mathbf{f}(\mathbf{w}) + \lambda \cdot \mathbf{f}(\mathbf{v}) \\ &= \lambda \cdot \mathbf{f}(\mathbf{w}) + \lambda \cdot f(a + \mathbf{v}). \end{aligned}$$

Logo,

$$\frac{1}{\lambda} \cdot h(a + \mathbf{v}) = \mathbf{f}(\mathbf{w}) + f(a + \mathbf{v}) = f(a + \mathbf{v}) - \alpha,$$

com  $\alpha = -\mathbf{f}(\mathbf{w})$ . Seja  $g : E \rightarrow \mathbb{R}$  a forma afim tal que  $g(x) = \frac{1}{\lambda}h(x)$ , para todo  $x \in E$ . Como  $g(x) = 0$  se, e somente se,  $h(x) = 0$ , temos que

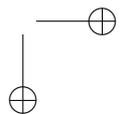
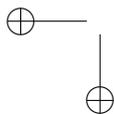
$$Nuc(g) = H' \quad \text{e} \quad g(x) = f(x) - \alpha.$$

□

## 2.8 Espaço afim euclidiano real

De agora em diante, restringiremos nossa atenção ao espaço afim euclidiano  $d$ -dimensional. Na maioria das vezes, consideraremos  $d = 2$ . O espaço afim euclidiano real de dimensão  $d$  nada mais é do que o espaço  $\mathbb{A}^d$  dotado de um produto interno específico: o produto escalar.

**Definição 2.7.** *Um espaço afim,  $\langle E, V_E, + \rangle$ , é dito um espaço afim euclidiano sempre que  $V_E$  é dotado de um produto interno. Um espaço*



afim euclidiano de interesse é o  $\mathbb{A}^d = \langle \mathbb{R}^d, \mathbb{R}^d, + \rangle$ , quando o espaço vetorial  $\mathbb{R}^d$  é dotado do produto escalar. Denotamos este espaço por  $\mathbb{E}^d$ .

Lembre-se de que um produto interno sobre um espaço  $\mathbb{K}$ -vetorial,  $\mathbf{V}$ , em que  $\mathbb{K} = \mathbb{R}$  ou  $\mathbb{K} = \mathbb{C}$ , é uma função,  $\langle \cdot, \cdot \rangle : \mathbf{V} \times \mathbf{V} \rightarrow \mathbb{K}$ , tal que

$$(P1) \quad \langle \mathbf{u} + \mathbf{v}, \mathbf{w} \rangle = \langle \mathbf{u}, \mathbf{w} \rangle + \langle \mathbf{v}, \mathbf{w} \rangle, \text{ para todo } \mathbf{v}, \mathbf{u}, \mathbf{w} \in \mathbf{V}.$$

$$(P2) \quad \langle \lambda \mathbf{u}, \mathbf{v} \rangle = \lambda \langle \mathbf{u}, \mathbf{v} \rangle, \text{ para todo } \lambda \in \mathbb{K} \text{ e } \mathbf{u}, \mathbf{v} \in \mathbf{V}.$$

$$(P3) \quad \langle \mathbf{u}, \mathbf{v} \rangle = \overline{\langle \mathbf{v}, \mathbf{u} \rangle}, \text{ para todo } \mathbf{u}, \mathbf{v} \in \mathbf{V}.$$

$$(P4) \quad \langle \mathbf{u}, \mathbf{v} \rangle > 0 \text{ sempre que } \mathbf{u} \neq \mathbf{0}.$$

Se o  $\mathbb{K}$ -espaço vetorial  $\mathbf{V}$  é  $\mathbb{K}^d$ , o *produto interno canônico* é dado por

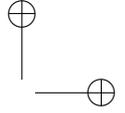
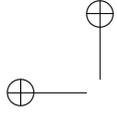
$$\langle \mathbf{u}, \mathbf{v} \rangle = \left\langle \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}, \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \right\rangle = u_1 \bar{v}_1 + \cdots + u_d \bar{v}_d = \sum_{i=1}^d u_i \bar{v}_i.$$

Se  $\mathbb{K} = \mathbb{R}$ , então temos que o conjugado,  $\bar{v}_i$ , de  $v_i$  é o próprio  $v_i$  e, portanto, podemos escrever o produto interno canônico sobre  $\mathbb{R}^d$  como

$$\langle \mathbf{u}, \mathbf{v} \rangle = \left\langle \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}, \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \right\rangle = u_1 v_1 + \cdots + u_d v_d = \sum_{i=1}^d u_i v_i.$$

O produto interno canônico sobre  $\mathbb{R}^d$  de dois vetores,  $\mathbf{u}$  e  $\mathbf{v}$ , é comumente denominado de *produto escalar* e é denotado como  $\mathbf{u} \cdot \mathbf{v}$ . O produto escalar nos permite realizar operações métricas no espaço vetorial euclidiano,  $\mathbb{R}^d$ , tais como calcular o comprimento de um vetor e o ângulo definido por dois vetores. Em particular, se  $\mathbf{u} \in \mathbb{R}^d$ , então temos que

$$\|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}},$$



em que  $\|\mathbf{u}\|$  denota é o comprimento de  $\mathbf{u}$ . Por sua vez, se  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$  são tais que  $\mathbf{u}, \mathbf{v} \neq \mathbf{0}$ , o ângulo  $\theta$  definido por  $\mathbf{u}$  e  $\mathbf{v}$  é obtido pela expressão

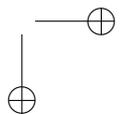
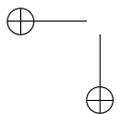
$$\theta = \cos^{-1} \left( \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right).$$

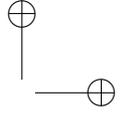
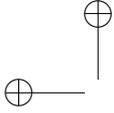
## 2.9 Exercícios

- 2.1 Considere o triângulo  $\triangle(abc)$  em  $\mathbb{A}^2$  em que  $a, b$  e  $c$  são pontos de  $\mathbb{A}^2$  com coordenadas cartesianas  $(3, 1)$ ,  $(5, 4)$  e  $(-2, 3)$ , respectivamente, com relação a um sistema de eixos ortogonais,  $Oxy$ , em  $\mathbb{A}^2$ . Então, se as coordenadas de um ponto  $p$ , com relação a  $Oxy$ , são  $(4, 3)$ , quais são as coordenadas baricênticas deste mesmo ponto com relação ao referencial afim  $(a, b, c)$  de  $\mathbb{A}^2$ ?
- 2.2 Considere o triângulo  $\triangle(abc)$  em  $\mathbb{A}^2$ . As linhas que passam pelos pares de pontos  $(a, b)$ ,  $(b, c)$  e  $(c, a)$  definem sete regiões do plano cujos os interiores são disjuntos. Caracterize as coordenadas baricênticas, com relação ao referencial afim  $(a, b, c)$ , dos pontos que estão no interior de cada um dessas regiões. Isto é, qual coordenada é positiva ou negativa ou zero? E o que dizer das coordenadas baricênticas dos pontos sobre as três linhas?
- 2.3 Dados quaisquer três pontos  $a, b$  e  $c$  em  $\mathbb{A}^2$  cujas coordenadas com relação ao referencial afim *padrão*,  $(O, \mathbf{e}_1, \mathbf{e}_2)$ , são  $(a_1, a_2)$ ,  $(b_1, b_2)$  e  $(c_1, c_2)$ , prove que  $a, b$  e  $c$  são colineares se, e somente se,

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ 1 & 1 & 1 \end{vmatrix} = 0.$$

- 2.4 Sejam  $(a_0, a_1, a_2)$ ,  $(b_0, b_1, b_2)$  e  $(c_0, c_1, c_2)$  as coordenadas *baricênticas* dos pontos  $a, b$  e  $c$  de  $\mathbb{A}^2$  com relação ao referencial afim padrão, isto é, o referencial formado pelos pontos de coordenadas cartesianas  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$  com relação ao referencial afim padrão,  $(O, \mathbf{e}_1, \mathbf{e}_2)$ . Então, prove que  $a, b$  e  $c$





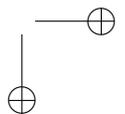
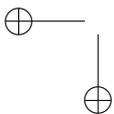
são colineares se, e somente se, o seguinte determinante é igual a zero:

$$\begin{vmatrix} a_0 & b_0 & c_0 \\ a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \end{vmatrix}.$$

- 2.5 Mostre que a função definida no Exemplo 2.3 é um mapa afim. Para tal, utilize o fato que  $\sum_{i=1}^n \alpha_i \cdot p_i = O + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{Op_i}$ . Em seguida, observe que  $\overrightarrow{Op_i} = x_i \cdot \mathbf{i} + y_i \cdot \mathbf{j}$ , para todo  $i = 1, \dots, n$ , em que  $(x_i, y_i)$  são as coordenadas do ponto  $p_i$  com relação ao referencial afim  $(O, \mathbf{i}, \mathbf{j})$ . Daí, calcule  $f(\sum_{i=1}^n \alpha_i \cdot p_i) = f(O + \sum_{i=1}^n \alpha_i \cdot \overrightarrow{Op_i})$  e conclua que o resultado obtido é igual a

$$\sum_{i=1}^n \alpha_i \cdot f(p_i).$$

- 2.6 No Problema 2.5, suponha que os vetores  $\mathbf{i}$  e  $\mathbf{j}$  possuam coordenadas  $\langle 1, 0 \rangle$  e  $\langle 0, 1 \rangle$ , respectivamente, com relação à base canônica,  $(\mathbf{e}_1, \mathbf{e}_2)$ , de  $\mathbb{R}^2$  (isto é, suponha que  $\mathbf{i} = \mathbf{e}_1$  e  $\mathbf{j} = \mathbf{e}_2$ ). Considere o quadrado em  $\mathbb{A}^2$  definido pelos pontos de coordenadas  $(0, 0)$ ,  $(1, 0)$ ,  $(1, 1)$  e  $(0, 1)$  com relação ao referencial afim  $(O, \mathbf{i}, \mathbf{j})$ . Qual é o objeto geométrico obtido quando aplicamos a função  $f$  em todos os pontos deste quadrado? Utilizando a sua resposta para a pergunta anterior, caracterize a função  $f$  em termos da(s) ação(ões) por ela causada(s). Por exemplo,  $f$  é uma rotação? Translação? Mudança de magnitude? Cisalhamento?
- 2.7 O *centroide* de um triângulo  $\Delta(abc)$  em  $\mathbb{A}^2$  é o baricentro dos pontos ponderados  $(a, \frac{1}{3})$ ,  $(b, \frac{1}{3})$  e  $(c, \frac{1}{3})$ . Se um mapa afim leva os vértices de um triângulo  $\Delta_1 = \{(0, 0), (6, 0), (0, 9)\}$  nos vértices do triângulo  $\Delta_2 = \{(1, 1), (5, 4), (3, 1)\}$ , ele também leva o centroide de  $\Delta_1$  no centroide de  $\Delta_2$ ? Justifique a sua resposta. Você pode assumir que as coordenadas dos vértices de  $\Delta_1$  e  $\Delta_2$  são dadas com relação ao referencial afim padrão,  $(O, \mathbf{e}_1, \mathbf{e}_2)$ .
- 2.8 Sejam  $\langle E, V_E, + \rangle$  e  $\langle F, V_F, +' \rangle$  dois espaços afins quaisquer. Seja  $f : E \rightarrow F$  um mapa afim. Então, mostre que  $f$  pode



ser escrito na forma matricial  $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$ , em que  $\mathbf{y}$  e  $\mathbf{b}$  representam coordenadas de dois vetores em  $V_F$ ,  $\mathbf{x}$  representa coordenadas de um vetor em  $V_E$  e  $\mathbf{A}$  é a matriz de uma transformação linear com relação a uma base de  $V_E$  e outra de  $V_F$ .

2.9 Determine a forma matricial do mapa afim  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  que leva os pontos  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$  de  $\mathbb{A}^2$  nos pontos  $(3, 2)$ ,  $(5, 8)$  e  $(7, 3)$  de  $\mathbb{A}^2$ , respectivamente. Assuma que as coordenadas dos pontos são dadas com relação ao referencial afim padrão,  $(O, \mathbf{e}_1, \mathbf{e}_2)$ . Para tal, proceda de acordo com os seguintes passos:

- (a) Como os pontos  $p = (0, 0)$ ,  $q = (1, 0)$  e  $r = (0, 1)$  não são colineares, eles formam um referencial afim em  $\mathbb{A}^2$  e, portanto, você pode definir uma base,  $(\vec{pq}, \vec{pr})$ , em  $\mathbb{R}^2$ . Os pontos  $s = (3, 2)$ ,  $t = (5, 8)$  e  $u = (7, 3)$  também não são colineares e, portanto,  $(\vec{st}, \vec{su})$ , é uma base em  $\mathbb{R}^2$ . Finalmente, determine a matriz,  $\mathbf{A}$ , do mapa linear que leva os vetores  $\vec{pq}$  e  $\vec{pr}$  nos vetores  $\vec{st}$  e  $\vec{su}$ , respectivamente. Observe que esta matriz é dada com relação a uma mesma base, a base canônica  $(\mathbf{e}_1, \mathbf{e}_2)$  de  $\mathbb{R}^2$ , em vez de ser definida com relação às duas bases,  $(\vec{pq}, \vec{pr})$  e  $(\vec{st}, \vec{su})$ , de  $\mathbb{R}^2$ .
- (b) Calcule as coordenadas  $\langle b_1, b_2 \rangle$  do vetor  $\vec{Os}$  com relação a  $(O, \mathbf{e}_1, \mathbf{e}_2)$ . As coordenadas  $\langle b_1, b_2 \rangle$  definem o vetor coluna  $\mathbf{b}$ .
- (c) Escreva o mapa afim  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  na forma  $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ .

2.10 Seja  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  um mapa afim cuja forma matricial, com relação ao referencial afim fixado, é  $f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}$ . Então, mostre que  $f$  é invertível e que a forma matricial da inversa,  $f^{-1} : \mathbb{A}^2 \rightarrow \mathbb{A}^2$ , de  $f$ , com relação ao mesmo referencial, é igual a

$$f^{-1}(\mathbf{x}) = \mathbf{A}^{-1}\mathbf{x} - \mathbf{A}^{-1}\mathbf{b}.$$

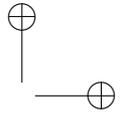
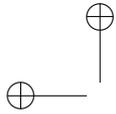
2.11 No Problema 2.9, o passo (a) é trivial porque as coordenadas de  $p$ ,  $q$  e  $r$  são  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$ , respectivamente. Suponha agora que as coordenadas de  $p$ ,  $q$  e  $r$  são  $(2, 3)$ ,  $(1, 6)$  e  $(3, -1)$ , respectivamente, e que as coordenadas de  $s$ ,  $t$  e  $u$  são  $(1, -2)$ ,  $(2, 1)$  e  $(-3, 5)$ , respectivamente. A sua tarefa é determinar a

forma matricial,  $\mathbf{Ax} + \mathbf{b}$ , do mapa afim,  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$ , que leva os pontos  $p, q$  e  $r$  nos pontos  $s, t$  e  $u$ , respectivamente. *Observe que tanto as coordenadas de  $\mathbf{x}$  quanto aquelas de  $\mathbf{Ax} + \mathbf{b}$  são dadas com relação à base canônica,  $(\mathbf{e}_1, \mathbf{e}_2)$ , de  $\mathbb{R}^2$ .* Para determinar  $f$ , calcule dois mapas afins,  $f_1 : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  e  $f_2 : \mathbb{A}^2 \rightarrow \mathbb{A}^2$ , tais que  $f_1$  leva os pontos de coordenadas  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$  nos pontos  $p, q$  e  $r$ , enquanto  $f_2$  leva os pontos de coordenadas  $(0, 0)$ ,  $(1, 0)$  e  $(0, 1)$  nos pontos  $s, t$  e  $u$ , respectivamente. Em seguida, calcule as formas matriciais de  $f_1$  e  $f_2$ . Finalmente, defina  $f$  como sendo  $f_2 \circ f_1^{-1}$  e utilize o enunciado do Problema 2.10 para obter a forma matricial de  $f$  a partir das formas matriciais dos mapas  $f_1$  e  $f_2$ .

- 2.12 Sejam  $(p, q, r)$  e  $(p', q', r')$  duas listas de três pontos não colineares em  $\mathbb{A}^2$ . Então, mostre que existe um mapa afim,  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$ , que leva os pontos  $p, q$  e  $r$  nos pontos  $p', q'$  e  $r'$ , respectivamente. Em seguida, mostre que o mapa  $f$  é único. O que ocorre se a condição de não colinearidade de  $p, q$  e  $r$  for falsa?
- 2.13 Seja  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$  um referencial afim em  $\mathbb{A}^3$  e seja  $(O', \mathbf{i}', \mathbf{j}')$  um referencial afim em  $\mathbb{A}^3$ . Seja  $\pi : \mathbb{A}^3 \rightarrow \mathbb{A}^2$  a função *projeção* tal que para todo ponto  $p \in \mathbb{A}^3$ , com  $p = O + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ , temos que  $\pi(p) = \pi(O + x\mathbf{i} + y\mathbf{j} + z\mathbf{k}) = O' + x\mathbf{i}' + y\mathbf{j}' = q$ ; ou seja,  $q$  é o ponto em  $\mathbb{A}^2$  cujas coordenadas  $(x, y)$  em relação a  $(O', \mathbf{i}', \mathbf{j}')$  são iguais à primeira e à segunda coordenadas de  $p$  em relação a  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ . Mostre que a função  $\pi$  é uma função afim.
- 2.14 Seja  $F$  um conjunto não vazio de pontos em  $\mathbb{A}^d$ . O *fecho afim*,  $FA(F)$ , de  $F$  é definido como o conjunto de todas as combinações afins de pontos de  $F$ ; isto é,  $FA(F)$  é tal que  $p \in FA(F)$  se, e somente se,

$$p = \sum_{i=1}^n \alpha_i \cdot p_i, \quad \sum_{i=1}^n \alpha_i = 1,$$

para alguma lista,  $((p_i, \alpha_i))_{i=1}^n$ , de  $n$  pontos ponderados de  $F$ , com  $n \in \mathbb{Z}$  e  $n \geq 1$ . Prove que  $FA(F)$  é um subespaço afim de  $\mathbb{A}^d$ .



- 2.15 Seja  $f : \mathbb{A}^2 \rightarrow \mathbb{A}^2$  um mapa afim. Então, mostre que  $f$  mapeia retas em  $\mathbb{A}^2$  em retas em  $\mathbb{A}^2$ . Em seguida, mostre que  $f$  mapeia retas paralelas em  $\mathbb{A}^2$  em retas paralelas em  $\mathbb{A}^2$ . Finalmente, se  $a, b$  e  $c, d$  são dois pares de pontos distintos em  $\mathbb{A}^2$ , então

$$\frac{|\overline{ab}|}{|\overline{cd}|} = \frac{|f(a)f(b)|}{|f(c)f(d)|},$$

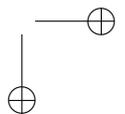
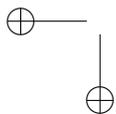
em que  $|xy|$  denota o comprimento do segmento de reta  $\overline{xy}$ .

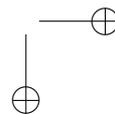
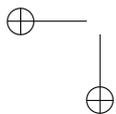
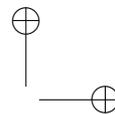
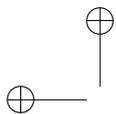
- 2.16 Seja  $\triangle(abc)$  um triângulo em  $\mathbb{A}^2$ . Seja  $x$  um ponto qualquer de  $\mathbb{A}^2$  tal que  $x$  não pertence a nenhuma das três retas definidas por quaisquer dois de  $a, b$  e  $c$ . Mostre que se a reta por  $a$  e  $x$  intersecta a reta por  $b$  e  $c$  no ponto  $p$ , a reta por  $b$  e  $x$  intersecta a reta por  $a$  e  $c$  no ponto  $q$  e a reta por  $c$  e  $x$  intersecta a reta por  $a$  e  $b$  no ponto  $r$ , então vale a seguinte igualdade:

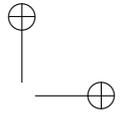
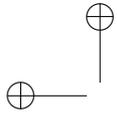
$$\frac{|\overline{ax}|}{|\overline{rb}|} \cdot \frac{|\overline{bp}|}{|\overline{pc}|} \cdot \frac{|\overline{cq}|}{|\overline{qa}|} = 1.$$

## 2.10 Notas bibliográficas

A introdução aos elementos da Geometria Afim apresentada neste capítulo foi inteiramente baseada no Capítulo 2 do livro de Jean Gallier [51].





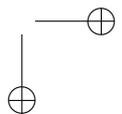
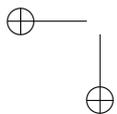


## Capítulo 3

# Convexidade

### 3.1 Considerações iniciais

O domínio da triangulação de Delaunay de um conjunto de pontos em  $\mathbb{E}^2$  é um *conjunto convexo* que é igual ao *fecho convexo* do conjunto de pontos. Neste capítulo, definimos conjuntos convexos em  $\mathbb{E}^d$ , fecho convexo de um conjunto de pontos em  $\mathbb{E}^d$  e damos um algoritmo para calcular o fecho convexo de um conjunto *finito* de pontos em  $\mathbb{E}^2$ . Na Seção 3.2, introduzimos a definição de conjunto convexo em  $\mathbb{E}^d$  e estudamos algumas propriedades. Na Seção Seção 3.3 introduzimos a definição de fecho convexo de um subconjunto (finito ou infinito) de pontos de  $\mathbb{E}^d$  e estabelecemos alguns resultados relevantes para os assuntos abordados nos capítulos seguintes. Na Seção Seção 3.4, damos um algoritmo para calcular o fecho convexo de um subconjunto não vazio e finito de pontos de  $\mathbb{E}^2$ . Uma simples modificação deste algoritmo nos permite calcular uma triangulação do mesmo conjunto de pontos, como veremos no Capítulo 4. Na Seção 3.4.2, introduzimos a análise de complexidade de pior caso de um algoritmo e a aplicamos ao algoritmo estudado na Seção 3.4. A Seção 3.5 discute, brevemente, o que entendemos por cota inferior de um problema. Em seguida, uma cota inferior para o problema de se calcular o fecho convexo de um subconjunto não vazio e finito de pontos de  $\mathbb{E}^2$  é derivada. A Seção 3.6 propõe exercícios sobre o conteúdo do capítulo. Por último, temos a Seção 3.7, que tece comentários



sobre as principais referências bibliográficas acerca do conteúdo do capítulo.

### 3.2 Conjuntos convexos

**Definição 3.1.** Um subconjunto  $F$  de  $\mathbb{E}^d$  é dito convexo se, e somente se, para quaisquer dois pontos,  $a$  e  $b$ , de  $\mathbb{E}^n$ , temos que  $c \in F$  para todo  $c = (1 - \alpha) \cdot a + \alpha \cdot b$ , onde  $\alpha \in \mathbb{R}$  e  $0 \leq \alpha \leq 1$ . O conjunto

$$\{c \in \mathbb{E}^n \mid c = (1 - \alpha) \cdot a + \alpha \cdot b, \alpha \in \mathbb{R}, 0 \leq \alpha \leq 1\},$$

que corresponde ao segmento de reta em  $\mathbb{E}^d$  definido por  $a$  e  $b$ , é denotado por  $[a, b]$ . Logo,  $F$  é convexo se, e somente se,  $[a, b] \subseteq F$ , para quaisquer dois pontos  $a$  e  $b$  (não necessariamente distintos) em  $F$ . O conjunto vazio,  $F = \emptyset$ , é trivialmente convexo, o conjunto unitário,  $F = \{a\}$ , é convexo e o próprio  $F = \mathbb{E}^d$  é claramente convexo.

A Figura 3.1 mostra um conjunto não convexo e outro convexo.

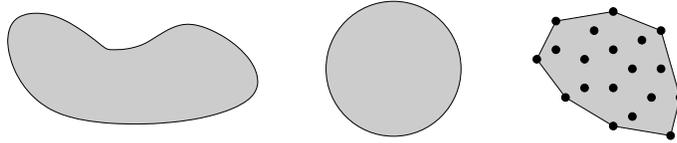
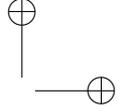
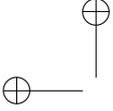


Figura 3.1: Um conjunto não convexo (esquerda), um conjunto convexo (meio) e o fecho convexo de um conjunto de pontos em  $\mathbb{E}^2$  (direita).

Uma propriedade importante dos conjuntos convexos é a que estabelece que a combinação convexa de qualquer subconjunto de pontos de um conjunto convexo qualquer resulta em um ponto do próprio conjunto. Uma combinação afim de pontos em  $\mathbb{E}^d$  é dita *convexa* se, e somente se, todos os pesos envolvidos na combinação são não negativos.

**Lema 3.1.** Seja  $F \subseteq \mathbb{E}^d$  um conjunto convexo. Então, dada qualquer lista  $(p_i)_{i=1}^n$  de  $n$  pontos de  $F$  e qualquer lista  $(\alpha_i)_{i=1}^n$  de  $n$



escalares em  $\mathbb{R}$  tal que  $\sum_{i=1}^n \alpha_i = 1$  e  $\alpha_i \geq 0$ , para todo  $i = 1, \dots, n$ , temos que

$$\sum_{i=1}^n \alpha_i \cdot p_i \in F. \quad (3.1)$$

*Demonstração.* Vamos proceder por indução em  $n$ . O caso base é  $n = 1$  e  $n = 2$ . Quando  $n = 1$ , a afirmação é trivialmente válida. Quando  $n = 2$ , temos que  $[p_1, p_2] \subseteq F$ , pois  $F$  é convexo, por hipótese. Suponha que nossa afirmação seja verdadeira para  $n = k$ , em que  $k$  é um inteiro arbitrário (mas, fixo), com  $k \geq 2$ . Então, considere o caso  $n = k + 1$  (passo indutivo). Quando  $n = k + 1$ , a Equação (3.1) é tal que

$$\sum_{i=1}^n \alpha_i \cdot p_i = \sum_{i=1}^{k+1} \alpha_i \cdot p_i.$$

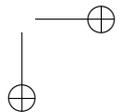
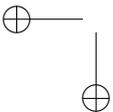
Mas,

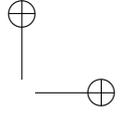
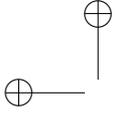
$$\begin{aligned} \sum_{i=1}^{k+1} \alpha_i \cdot p_i &= \left( \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k \alpha_i} \right) \left( \sum_{i=1}^{k+1} \alpha_i \cdot p_i \right) \\ &= \sum_{i=1}^{k+1} \left( \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k \alpha_i} \right) \alpha_i \cdot p_i \\ &= \left( \sum_{i=1}^k \left( \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k \alpha_i} \right) \alpha_i \cdot p_i \right) + \left( \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k \alpha_i} \right) \alpha_{k+1} \cdot p_{k+1} \\ &= \left( \sum_{i=1}^k \left( \frac{\sum_{i=1}^k \alpha_i}{\sum_{i=1}^k \alpha_i} \right) \alpha_i \cdot p_i \right) + \alpha_{k+1} \cdot p_{k+1} \\ &= \left( \sum_{i=1}^k \alpha_i \right) \left( \sum_{i=1}^k \left( \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} \right) \cdot p_i \right) + \alpha_{k+1} \cdot p_{k+1}. \end{aligned}$$

Note que

$$p = \sum_{i=1}^k \left( \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} \right) \cdot p_i$$

é uma combinação convexa de  $p_1, \dots, p_k$ , pois  $\alpha_1, \dots, \alpha_k \geq 0$ , por hipótese, e  $\sum_{i=1}^k \left( \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} \right) = 1$ . Logo, pela hipótese de indução,





temos que  $p \in F$ . Além disso, pelo caso base, podemos concluir que

$$\begin{aligned} \sum_{i=1}^{k+1} \alpha_i \cdot p_i &= \left( \sum_{i=1}^k \alpha_i \right) \left( \sum_{i=1}^k \left( \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} \right) \cdot p_i \right) + \alpha_{k+1} \cdot p_{k+1} \\ &= \left( \sum_{i=1}^k \alpha_i \right) \cdot p + \alpha_{k+1} \cdot p_{k+1} \end{aligned}$$

é um ponto de  $F$ , pois este ponto é a soma convexa de dois pontos de  $F$ .  $\square$

Outra propriedade importante é a dada no lema a seguir:

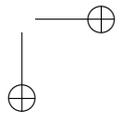
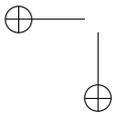
**Lema 3.2.** *A interseção de uma coleção (finita ou infinita) de subconjuntos convexos de  $\mathbb{E}^d$  é ela própria um subconjunto convexo de  $\mathbb{E}^d$ .*

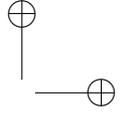
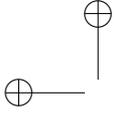
*Demonstração.* Exercício para o leitor (ver Problema 3.1).  $\square$

### 3.3 Fecho convexo

Seja  $A$  qualquer subconjunto de  $\mathbb{E}^d$ . De acordo com o Lema 3.2, a interseção de *todos* os subconjuntos convexos de  $\mathbb{E}^d$  que contêm  $A$  é um conjunto convexo, digamos  $C$ , de  $\mathbb{E}^d$ . Observe que  $C$  deve ser o “menor” conjunto convexo que contém  $A$ , pois se houvesse um outro conjunto menor, digamos  $D$ , então o conjunto  $C - D$  não seria vazio e seus elementos não estariam na interseção de todos os subconjuntos convexos que contêm  $A$ , o que implica que  $C$  não é o resultado desta interseção (uma contradição!). Este menor subconjunto convexo contendo  $A$  é denominado *fecho convexo* de  $A$  e denotado por  $FC(A)$ . Quando  $A$  é convexo, o fecho convexo,  $FC(A)$ , de  $A$  é o próprio  $A$ . O seguinte lema nos fornece uma caracterização de fecho convexo (em termos de combinações afins) de um subconjunto (finito ou infinito) de  $\mathbb{E}^d$ :

**Lema 3.3.** *Seja  $I$  um subconjunto finito ou infinito dos naturais,  $\mathbb{N}$ . Então, para qualquer lista  $(p_i)_{i \in I}$  de pontos de  $\mathbb{E}^d$ , o conjunto  $F$  tal*





que

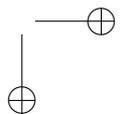
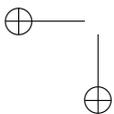
$$F = \left\{ p \in \mathbb{E}^d \mid p = \sum_{i \in I} \alpha_i \cdot p_i, \text{ com } \sum_{i \in I} \alpha_i = 1 \text{ e } \alpha_i \geq 0, \forall i \in I \right\}$$

é o fecho convexo dos pontos em  $(p_i)_{i \in I}$ ; isto é, o conjunto de todas as combinações convexas dos pontos em  $(p_i)_{i \in I}$  é o fecho convexo deles.

*Demonstração.* Se  $I$  é o conjunto vazio, então  $(p_i)_{i \in I}$  e  $F$  são ambos iguais ao conjunto vazio, que é convexo e cujo fecho convexo é ele próprio. De forma análoga, se  $I$  é o conjunto unitário, então  $(p_i)_{i \in I}$  e  $F$  são iguais e, portanto, nossa afirmação também é verdadeira. Então, vamos assumir que  $I$  possui pelo menos dois pontos. Por definição, o fecho convexo de  $(p_i)_{i \in I}$  contém todos os pontos em  $(p_i)_{i \in I}$ . Como o fecho convexo é um conjunto convexo, ele também deve conter todos os pontos resultantes da combinação convexa de dois pontos de  $(p_i)_{i \in I}$ . Mas, com um argumento idêntico àquele usado na prova do Lema 3.1, podemos mostrar que qualquer combinação convexa dos pontos em  $(p_i)_{i \in I}$  pode ser obtida por combinações convexas sucessivas de dois pontos de  $(p_i)_{i \in I}$ . Logo, o fecho convexo de  $(p_i)_{i \in I}$  deve conter os pontos de  $F$ . Mas, como fecho convexo de  $(p_i)_{i \in I}$  é o menor conjunto que contém  $(p_i)_{i \in I}$ , ele tem de ser igual ao conjunto  $F$ .  $\square$

A Figura 3.1 ilustra a noção de fecho convexo do Lema 3.3.

A prova por indução que demos para o Lema 3.3 nos permite concluir que, mesmo quando  $I$  é infinito, qualquer ponto do fecho convexo de  $(p_i)_{i \in I}$  pode ser obtido por uma combinação convexa de um número *finito* de pontos de  $(p_i)_{i \in I}$ . Mas, este número finito pode ser grande quando comparado à dimensão  $d$  do espaço euclidiano,  $\mathbb{E}^d$ , no qual os pontos residem. Logo, é natural perguntar se existe um subconjunto finito e *pequeno* de pontos em  $(p_i)_{i \in I}$  que gera todos os pontos do fecho convexo de  $(p_i)_{i \in I}$ . Uma resposta para esta pergunta é importante se desejarmos, por exemplo, desenvolver um algoritmo para calcular o fecho convexo de  $(p_i)_{i \in I}$ . Um teorema do matemático alemão Constantin Carathéodory (1873-1950) nos dá uma resposta afirmativa.



**Teorema 3.4** (Teorema de Carathéodory). *Seja  $S$  um subconjunto não vazio de pontos de  $\mathbb{E}^d$ . Então, o fecho convexo,  $FC(S)$ , de  $S$  é o conjunto de combinações convexas de subconjuntos de  $d + 1$  pontos de  $S$ .*

*Demonstração.* Seja  $S = (p_i)_{i \in L}$  um conjunto de pontos de  $\mathbb{E}^d$ , em que  $L \subseteq \mathbb{N}$  e  $L \neq \emptyset$ . A prova por indução que demos para o Lema 3.3 nos permite concluir que, para qualquer ponto  $p \in FC(S)$ , existe um conjunto *finito*  $I$ , com  $I \subseteq L$ , tal que  $p = \sum_{i \in I} \alpha_i \cdot p_i$ , em que  $(\alpha_i)_{i \in I}$  é uma lista de escalares não negativos e  $\sum_{i \in I} \alpha_i = 1$ . Nós vamos mostrar que  $|I| = d + 1$ . A prova é por absurdo. Assuma que o teorema é falso. Então, há um ponto  $b$  em  $FC(S)$  tal que  $b$  pode ser expresso como uma combinação convexa,  $b = \sum_{i \in I} \alpha_i \cdot p_i$ , em que  $I \subseteq L$  é um conjunto finito de cardinalidade  $|I| = q$ , com  $q \geq d + 2$ , mas não pode ser expresso por uma combinação convexa,  $\sum_{j \in J} \mu_j \cdot p_j$ , de  $|J|$  pontos de  $FC(S)$ , em que  $J \subseteq L$  e  $|J| < q$ . Por hipótese,

$$b = \alpha_1 \cdot p_1 + \dots + \alpha_q \cdot p_q,$$

em que  $\alpha_1 + \dots + \alpha_q = 1$  e  $\alpha_i > 0$ , com  $1 \leq i \leq q$ . Tome qualquer origem  $O$  em  $\mathbb{E}^d$ . Como há  $q > d + 1$  pontos,  $p_1, \dots, p_q$ , estes pontos formam um conjunto afimemente dependente e, portanto, pelo Lema 2.4, há uma lista  $(\mu_1, \dots, \mu_q)$  de escalares, nem todos iguais a zero, tal que

$$\mu_1 + \dots + \mu_q = 0 \quad \text{e} \quad \sum_{i=1}^q \mu_i \cdot \overrightarrow{Op_i} = \mathbf{0}.$$

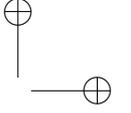
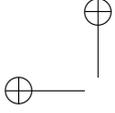
Defina

$$T = \{t \in \mathbb{R} \mid \alpha_i + t \cdot \mu_i \geq 0, \mu_i \neq 0, 1 \leq i \leq q\} \subset \mathbb{R}.$$

O conjunto  $T$  não é vazio, pois ele contém 0 (fazendo  $t = -\alpha_i/\mu_i$ ). Como  $\sum_{i=1}^q \mu_i = 0$  e  $\mu_j \neq 0$  para algum  $j \in \{1, \dots, q\}$ , existem  $\mu_h$  e  $\mu_k$  tais que  $\mu_h < 0$  e  $\mu_k > 0$ , o que implica que  $T = [\beta, \gamma]$ , em que

$$\beta = \max_{1 \leq i \leq q} \left\{ -\frac{\alpha_i}{\mu_i} \mid \mu_i > 0 \right\} \quad \text{e} \quad \gamma = \min_{1 \leq i \leq q} \left\{ -\frac{\alpha_i}{\mu_i} \mid \mu_i < 0 \right\}.$$

Note que  $\beta < 0 < \gamma$ , pois  $\alpha_i > 0$  para todo  $i = 1, \dots, q$ . Afirmamos que há um  $j$ , com  $1 \leq j \leq q$ , tal que  $\alpha_j + \beta\mu_j = 0$ . Caso contrário,



teríamos  $\alpha_i + \beta \cdot \mu_i > 0$ , para todo  $i = 1, \dots, q$ . Mas, isto nos permitiria escolher um número real positivo,  $\epsilon$ , suficientemente pequeno de tal forma que  $\alpha_i + (\beta - \epsilon)\mu_i > 0$  para todo  $i = 1, \dots, q$ . De fato, como  $\alpha_i + \beta\mu_i > 0$  para todo  $i = 1, \dots, q$  (e há algum  $\mu_i > 0$ ), poderíamos escolher  $\epsilon > 0$  tal que  $\epsilon < \min_{1 \leq i \leq q} \{(\alpha_i + \beta\mu_i)/\mu_i \mid \mu_i > 0\}$ . Mas, neste caso,  $\beta - \epsilon \in T$ , com  $\beta - \epsilon < \beta$ : uma contradição! Logo, há um índice  $j$  tal que  $\alpha_j + \beta\mu_j = 0$ . Como  $\sum_{i=1}^q \mu_i \cdot \overrightarrow{Op_i} = \mathbf{0}$ , podemos escrever

$$\begin{aligned} b &= \sum_{i=1}^q \alpha_i \cdot p_i = O + \sum_{i=1}^q \alpha_i \cdot \overrightarrow{Op_i} \\ &= O + \sum_{i=1}^q \alpha_i \cdot \overrightarrow{Op_i} + \beta \left( \sum_{i=1}^q \mu_i \cdot \overrightarrow{Op_i} \right) \\ &= O + \sum_{i=1}^q (\alpha_i + \beta\mu_i) \cdot \overrightarrow{Op_i} \\ &= \sum_{i=1}^q (\alpha_i + \beta\mu_i) \cdot p_i. \end{aligned}$$

Como  $\alpha_j + \beta\mu_j = 0$ ,

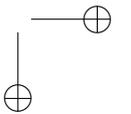
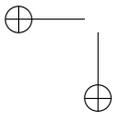
$$b = \sum_{i=1, i \neq j}^q (\alpha_i + \beta\mu_i) \cdot p_i. \quad (3.2)$$

Além disso, como  $\sum_{i=1}^q \mu_i = 0$ ,  $\sum_{i=1}^q \alpha_i = 1$  e  $\alpha_j + \beta\mu_j = 0$ , temos que

$$\sum_{i=1, i \neq j}^q \alpha_i + \beta\mu_i = 1.$$

Mas,  $\alpha_i + \beta\mu_i \geq 0$  para todo  $i = 1, \dots, q$ . Logo, a combinação afim na Equação (3.2) é, na verdade, uma combinação convexa de  $q - 1$  pontos de  $\mathbb{E}^d$ . Esta afirmação contradiz a hipótese que o ponto  $b$  não pode ser expresso como uma combinação convexa de menos de  $q$  pontos.  $\square$

O Teorema 3.4 nos diz, por exemplo, que cada ponto do fecho convexo,  $FC(F)$ , de um subconjunto,  $F$ , de 3 ou mais pontos de  $\mathbb{E}^2$



é dado pela combinação convexa de 3 pontos de  $F$ , como ilustrado na Figura 3.2. Observe que o teorema implica que  $FC(F)$  é uma união de triângulos cujos vértices são pontos de  $F$ . Esta observação nos permite desenvolver um algoritmo simples, mas extremamente ineficiente, para calcular o fecho convexo de um subconjunto *finito* de pontos em  $\mathbb{E}^2$ . Outras noções, tais como semiplanos e pontos extremos, que discutiremos logo a seguir, levam-nos a algoritmos mais eficientes.

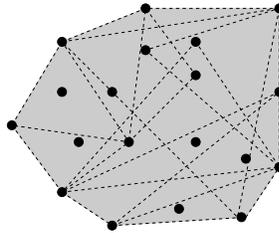


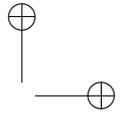
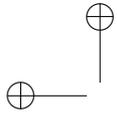
Figura 3.2: Ilustração do Teorema de Caratheódory.

Dado qualquer hiperplano (afim)  $H$  em  $\mathbb{E}^d$ , se  $f : \mathbb{E}^d \rightarrow \mathbb{R}$  é qualquer forma afim (não constante) definindo  $H$  (isto é,  $H = Nuc(f)$ ), podemos definir dois subconjuntos,  $H_+(f)$  e  $H_-(f)$ , de  $\mathbb{E}^d$  como segue:

$$H_+(f) = \{p \in \mathbb{E}^d \mid f(p) \geq 0\} \quad \text{e} \quad H_-(f) = \{p \in \mathbb{E}^d \mid f(p) \leq 0\}.$$

Estes conjuntos são chamados *semiespaços (fechados) associados a  $f$* .

Note que o par  $H_+(f)$  e  $H_-(f)$  depende apenas do hiperplano  $H$ . Por esta razão, nós também dizemos que  $H_+(f)$  e  $H_-(f)$  são os *semiespaços (fechados) associados a  $H$* . Por sua vez, a função  $f$  escolhida para definir  $H$  determina qual dos dois semiespaços é  $H_+(f)$  (resp.  $H_-(f)$ ). Observe que  $H_+(f) \cup H_-(f) = \mathbb{E}^d$  e  $H_+(f) \cap H_-(f) = H$ . Observe também que  $H_+(f)$  e  $H_-(f)$  são convexos. O seguinte teorema é uma versão simplificada do clássico *Main Theorem* para politopos, que caracteriza o fecho convexo de um subconjunto *finito* de pontos em  $\mathbb{E}^d$  em termos da interseção de semiespaços fechados em  $\mathbb{E}^d$ :



**Teorema 3.5.** *Seja  $S$  um subconjunto finito de pontos de  $\mathbb{E}^d$ . Então, o fecho convexo,  $FC(S)$ , de  $S$  é igual à interseção de semiespaços fechados de  $\mathbb{E}^d$  tal que cada semiespaço contém todos os pontos de  $S$ .*

*Demonstração.* Consulte a obra de Günter Ziegler ([138]) para uma prova.  $\square$

A Figura 3.3 ilustra o Teorema 3.3 em  $\mathbb{E}^2$ .

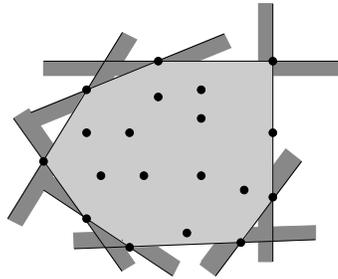
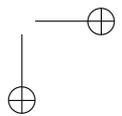
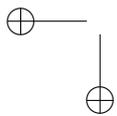
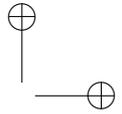
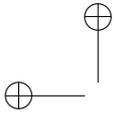


Figura 3.3: Ilustração do *Main Theorem* em  $\mathbb{E}^2$ .

O *Main Theorem* nos diz que  $FC(S)$  é um *politopo convexo*. Em  $\mathbb{E}^2$ , um politopo convexo, quando *limitado*, é simplesmente um polígono convexo. Lembre-se de que um subconjunto de  $\mathbb{E}^2$  é limitado se, e somente se, existe um retângulo de dimensões finitas que o contém. Como  $S$  é um conjunto *finito* de pontos de  $\mathbb{E}^2$ , o politopo convexo,  $FC(S)$ , é sempre limitado e, portanto, um polígono convexo. Além disso, os semiespaços do enunciado do teorema podem se restringir aos semiespaços definidos por hiperplanos que contém pontos de  $S$ . Na Figura 3.3, esses hiperplanos são as retas que contém as arestas do polígono correspondente ao fecho convexo dos pontos mostrados na figura. Essas observações foram cruciais para o surgimento de algoritmos para calcular o fecho convexo de subconjuntos finitos de  $\mathbb{E}^d$ .

Seja  $F$  qualquer subconjunto convexo de  $\mathbb{E}^d$ . Então, um ponto  $p \in F$  é dito *extremo* ou *extremal* se, e somente se,  $p$  não pode ser escrito como uma combinação convexa da forma  $\lambda \cdot a + (1 - \lambda) \cdot b$ , com  $\lambda \in \mathbb{R}$  e  $0 < \lambda < 1$ , para dois pontos quaisquer (e não necessariamente





distintos),  $a$  e  $b$ , de  $F$ . Em outras palavras, o ponto  $p \in F$  é extremo se ele *não* está contido no *interior* de qualquer segmento de reta  $[a, b]$  em  $F$  ou, de forma equivalente, se o conjunto  $F - \{p\}$  ainda é convexo. A noção de ponto extremo é fundamental para o desenvolvimento de algoritmos para calcular o fecho convexo de subconjuntos finitos de  $\mathbb{E}^n$ .

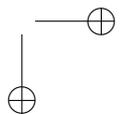
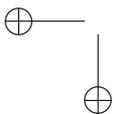
**Teorema 3.6** (Krein e Milman, 1940). *Seja  $S$  qualquer subconjunto convexo de  $\mathbb{E}^d$ . Se  $S$  é um conjunto convexo, compacto e não vazio, então  $S$  é igual ao fecho convexo do conjunto dos pontos extremos de  $S$ .*

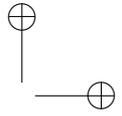
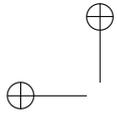
*Demonstração.* Consulte a obra de Jean Gallier [52] para uma prova. □

Se  $F$  é um subconjunto finito de  $\mathbb{E}^d$ , o Teorema 3.6 nos permite concluir que o fecho convexo,  $FC(F)$ , de  $F$  é o conjunto de todas as combinações convexas *dos pontos extremos de  $FC(F)$* , que por sua vez são pontos de  $F$ . Logo, os pontos extremos de  $FC(F)$  formam uma representação finita de  $FC(F)$  (do ponto de vista computacional). Isto sugere que o cálculo de  $FC(F)$  se restrinja a encontrar os pontos de  $F$  que são pontos extremos de  $FC(S)$ . Quando  $d = 2$ , já sabemos que  $FC(F)$  é um polígono convexo — *mais precisamente, quando  $F$  possui pelo menos 3 pontos não colineares*. Então, os pontos extremos de  $FC(F)$  são os vértices do polígono que definem ângulos (internos) menores do que  $\pi$ , pois estes são os únicos pontos do polígono que não estão contidos no interior de nenhum segmento de reta definido por dois pontos distintos do próprio polígono. Note que tais vértices (e também pontos extremos) devem ser obrigatoriamente pontos de  $F$ , pois esta é a única forma deles serem combinações convexas de pontos do polígono! A Figura 3.4 ilustra todas as observações feitas acima.

### 3.4 Graham scan

De agora em diante, o foco da discussão muda para um contexto computacional. Nesta seção, em particular, descrevemos um algoritmo para calcular o fecho convexo,  $FC(S)$ , de um dado subconjunto





finito,  $S$ , qualquer de pontos de  $\mathbb{E}^2$ . Antes de descrever o algoritmo, precisamos definir mais precisamente o que entendemos por “calcular”  $FC(S)$ . Como vimos anteriormente, o conjunto  $FC(S)$  pode ser inteiramente caracterizado pelos seus vértices extremos. Logo, uma representação natural para a saída de qualquer algoritmo para calcular  $FC(S)$  é uma lista ordenada dos vértices extremos de  $FC(S)$ . Se há pelo menos 3 vértices extremos em  $FC(S)$ , então  $FC(S)$  é um polígono convexo e uma possível *ordem* é aquela que obtemos ao visitar os vértices do polígono em um percurso anti-horário por sua fronteira.

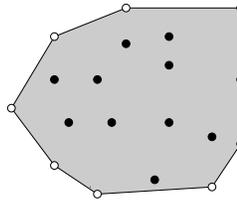


Figura 3.4: O fecho convexo,  $FC(F)$ , de um subconjunto finito,  $F$ , de pontos em  $\mathbb{E}^2$ . Os vértices extremos de  $FC(F)$  são pontos de  $F$  e estão destacados como circunferências em cor preta e interior em cor branca.

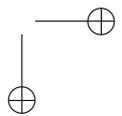
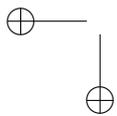
A fronteira do polígono convexo  $FC(S)$  é uma curva poligonal fechada cujos vértices são os vértices extremos de  $FC(S)$  (ver Figura 3.4). Denominamos esta curva poligonal por *fecho* de  $S$ . A saída de qualquer algoritmo para calcular  $FC(S)$  consiste, *por convenção*, dos pontos extremos de  $FC(S)$  na ordem em que eles ocorrem em um percurso anti-horário pelo fecho de  $S$ .

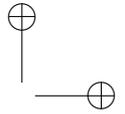
$$(p_i)_{i=1}^n,$$

são dados por suas coordenadas,  $p_i = (x_i, y_i)$ , com relação a um referencial afim  $(O, \mathbf{i}, \mathbf{j})$  em  $\mathbb{E}^2$  tal que  $\mathbf{i}$  e  $\mathbf{j}$  são ortogonais; isto é, tal que

$$\mathbf{i} \cdot \mathbf{j} = 0.$$

O algoritmo que descreveremos a seguir foi desenvolvido pelo matemático estadunidense, Ronald Lewis Graham, e é conhecido como





*Graham scan.* O primeiro passo deste algoritmo consiste em determinar o ponto de  $S$  de menor coordenada  $y$ . Se há mais de um ponto em  $S$  com menor coordenada  $y$ , o algoritmo toma, entre eles, o de maior coordenada  $x$ . Suponha que este ponto é  $p_k$ , para algum  $k \in \{1, \dots, n\}$ . O próximo passo consiste em calcular o ângulo,  $\theta_i$ , definido pelo vetor  $\overrightarrow{p_k p_i}$  e o vetor  $\mathbf{i}$ , para todo  $i \in \{1, \dots, n\} - \{k\}$ . O terceiro passo consiste em rearranjar os pontos em  $S - \{p_k\}$  em ordem não decrescente do valor do ângulo  $\theta_i$ . Seja  $q_1, \dots, q_{n-1}$  a sequência de pontos de  $S - \{p_k\}$  resultante da ordenação. Denote  $p_k$  por  $q_0$ .

A Figura 3.5 ilustra o resultado dos 3 primeiros passos do *Graham scan*. Esses passos compõem a fase de *pré-processamento* do algoritmo. A fase seguinte, denominada fase de *construção do fecho*, determina o fecho de  $S$  e é simplificada pelo resultado da primeira fase.

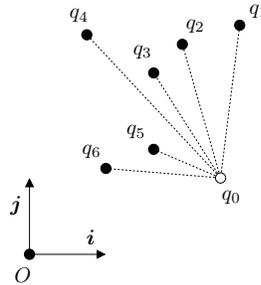
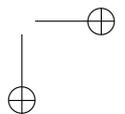
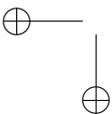
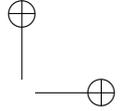
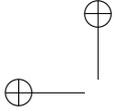


Figura 3.5: Resultado dos três primeiros passos do *Graham scan*.

Denote por  $\beta_i$  o ângulo definido por  $\overrightarrow{q_0 q_i}$  e  $\mathbf{i}$ , para todo  $i = 1, \dots, n - 1$ . Para simplificar a descrição da fase de construção do fecho, vamos assumir que  $n \geq 3$  e que os ângulos  $\beta_i$ 's são distintos dois a dois. Como  $\beta_1 < \beta_j$ , para todo  $j = 2, \dots, n - 1$ , os pontos  $q_2, \dots, q_{n-1}$  estão todos à esquerda — *com relação ao referencial afim*  $(O, \mathbf{i}, \mathbf{j})$  — da reta que passa por  $q_0$  e  $q_1$  e está orientada de  $q_0$  para  $q_1$ . Logo, a partir do Teorema 3.5, podemos concluir que a aresta  $\overline{q_0 q_1}$  pertence ao fecho de  $S$  e, portanto, a fase de construção inicia a composição do fecho de  $S$  pelos pontos  $q_0$  e  $q_1$  e pela aresta  $\overline{q_0 q_1}$ .





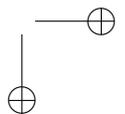
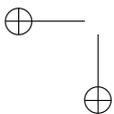
O próximo passo consiste em visitar cada ponto em  $q_2, \dots, q_{n-1}$ , nesta ordem, para que os próximos vértices do fecho de  $S$  sejam determinados. Apenas os vértices do fecho são mantidos pelo algoritmo, pois as arestas podem ser determinadas a partir deles. De forma geral, quando o ponto  $q_i$  é visitado, o algoritmo já conhece os vértices do fecho do conjunto  $S_{i-1}$  dos pontos  $\{q_0, q_1, \dots, q_{i-1}\}$ . Esta afirmação é certamente verdadeira para  $i = 2$ , pois  $S_1 = \{q_0, q_1\}$ , e nós a provaremos para o caso geral mais adiante. Os vértices do fecho de  $S_i$  são calculados na visita a  $q_i$  e a partir dos vértices do fecho de  $S_{i-1}$ . Se  $a$  e  $b$  são os dois últimos pontos do fecho de  $S_{i-1}$ , nesta ordem, então o fecho de  $S_i = S_{i-1} \cup \{q_i\}$  é calculado a partir da tripla  $(a, b, q_i)$ . Refira-se à Figura 3.6 durante a explicação que se segue.

Se  $q_i$  está à esquerda, com relação ao referencial afim  $(O, \mathbf{i}, \mathbf{j})$ , da reta por  $\overline{ab}$  orientada de  $a$  para  $b$ , então o algoritmo define o conjunto de vértices do fecho de  $S_i$  como sendo o conjunto de vértices do fecho de  $S_{i-1}$  acrescido de  $q_i$ . Caso contrário, o algoritmo descarta o vértice  $b$ , denota o vértice  $a$  por  $b$  e denota o vértice que antecede o atual vértice  $a$  no fecho de  $S_{i-1}$  por  $a$ . Depois, o algoritmo considera a tripla  $(a, b, q_i)$  novamente e repete o teste de orientação com relação à reta por  $\overline{ab}$ . Como mostraremos mais adiante, o algoritmo pode, no pior caso, repetir o teste até que  $a$  se torne igual a  $q_0$ . Se  $a = q_0$ , então o ponto  $q_i$  está à esquerda da reta por  $\overline{ab}$  orientada de  $a$  para  $b$ , pois  $b = q_1$  e todos os pontos em  $\{q_2, \dots, q_{n-1}\}$  estão.

O Algoritmo 3.1 fornece o pseudocódigo do Graham *scan*.

O algoritmo utiliza uma pilha,  $\mathcal{P}$ , para armazenar o fecho de  $S_i$ . O teste de orientação para determinar se o ponto  $q_i$  está à esquerda da reta por  $\overline{ab}$ , orientada de  $a$  para  $b$  com relação ao referencial afim  $(O, \mathbf{i}, \mathbf{j})$ , é realizado pela função `LEFT()`. Esta função, assim como os demais predicados geométricos utilizados neste livro, está descrita em detalhes no Apêndice A. A execução de `LEFT(a, b, q_i)` devolve o valor lógico verdadeiro se, e somente se, os pontos  $a$ ,  $b$  e  $q_i$ , nesta ordem, definem uma “curva à esquerda” com relação à orientação definida pelo referencial afim  $(O, \mathbf{i}, \mathbf{j})$ . O cálculo realizado por `LEFT`, no entanto, utiliza apenas as coordenadas dos pontos  $a$ ,  $b$  e  $q_i$ .

Lembre-se de que o pseudocódigo dado em Algoritmo 3.1 supõe que não há três pontos colineares em  $S$ , pois esta condição é suficiente e necessária para que os ângulos definidos pelos vetores  $\overline{p_k p_i}$  e  $\mathbf{i}$  sejam distintos dois a dois, para todo  $i \in \{1, \dots, n\} - \{k\}$ . Uma simples



modificação da linha 3 do pseudocódigo faz com que esta restrição seja ignorada. O Problema 3.15 pede uma descrição do algoritmo modificado.

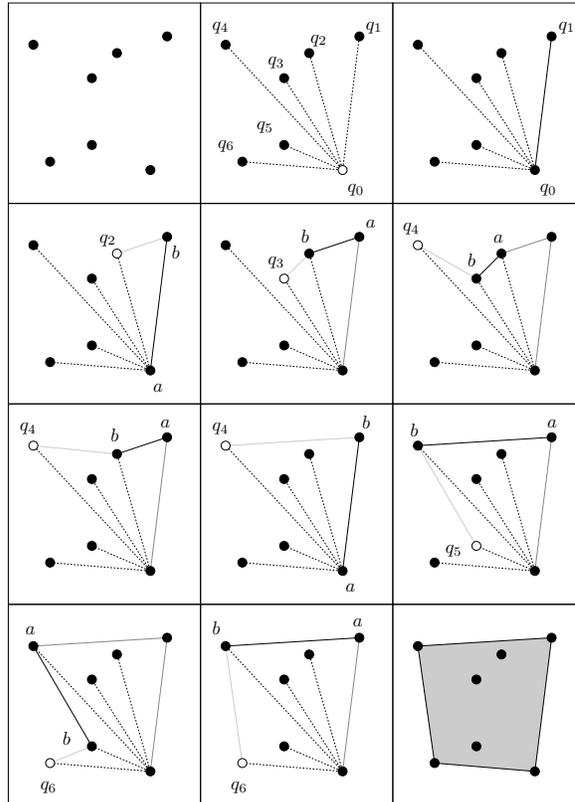
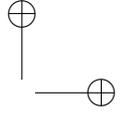
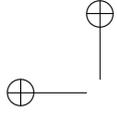


Figura 3.6: Ilustração da execução do *Graham scan*.

### 3.4.1 Corretude

O pseudocódigo em Algoritmo 3.1 está correto, o que significa que se  $S$  é um conjunto qualquer com pelo menos 3 pontos de  $\mathbb{E}^2$



tais que não há, entre eles, três pontos colineares, o algoritmo sempre termina e produz a saída esperada, ou seja, os vértices do fecho de  $S$  em ordem anti-horária. Para provar que a afirmação acima está correta, provaremos primeiro uma propriedade invariante do laço do algoritmo.

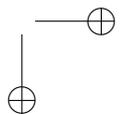
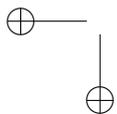
---

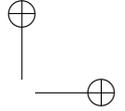
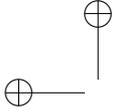
**Algoritmo 3.1** GRAHAMSCAN( $S = (p_i)_{i=1}^n, (O, \mathbf{i}, \mathbf{j})$ )

---

- 1: encontre o ponto  $p_k$  de  $S$  tal que  $y_k \leq y_i$ , para todo  $i \in \{1, \dots, n\} - \{k\}$ , e  $x_k < x_j$  para todo  $j \in \{1, \dots, n\} - \{k\}$  tal que  $y_i = y_k$
  - 2: calcule o ângulo  $\theta_i$  definido por  $\overrightarrow{p_k p_i}$  e  $\mathbf{i}$ ,  $\forall i \in \{1, \dots, n\} - \{k\}$
  - 3: ordene os pontos em  $S - \{p_k\}$  em ordem crescente do valor do ângulo  $\theta_i$  associado com  $p_i$ ; denote sequência ordenada por  $(q_i)_{i=1}^{n-1}$
  - 4:  $\mathcal{P} \leftarrow \emptyset$  { $\mathcal{P}$  é uma pilha vazia}
  - 5: empilhe  $p_k, q_1$  e  $q_2$  em  $\mathcal{P}$ , nesta ordem
  - 6:  $i \leftarrow 3$
  - 7: **enquanto**  $i < n$  **faça**
  - 8:   desempilhe o ponto no topo de  $\mathcal{P}$  e o atribua a  $b$
  - 9:   desempilhe o ponto no topo de  $\mathcal{P}$  e o atribua a  $a$
  - 10:   **se** LEFT( $a, b, q_i$ ) **então**
  - 11:     empilhe  $a, b$  e  $q_i$  em  $\mathcal{P}$ , nesta ordem
  - 12:      $i \leftarrow i + 1$
  - 13:   **senão**
  - 14:     empilhe  $a$  em  $\mathcal{P}$
  - 15:   **fim se**
  - 16: **fim enquanto**
  - 17: **devolva**  $\mathcal{P}$
- 

Observe que o ponto  $p_k$  escolhido na linha 1 de Algoritmo 3.1 é um ponto extremo do fecho convexo,  $FC(S)$ , de  $S$ . Consequentemente, o ponto  $p_k$  tem de pertencer ao fecho de  $S$ . Como não há três pontos colineares em  $S$ , os ângulos calculados na linha 2 são distintos dois a dois. Utilizando qualquer algoritmo correto para ordenar uma sequência de escalares, podemos garantir que a linha 3 produz uma sequência,  $(q_i)_{i=1}^{n-1}$ , de pontos de  $S$  ordenada de tal forma que o ângulo entre  $\overrightarrow{p_k q_j}$  e  $\mathbf{i}$  é sempre maior do que aquele entre  $\overrightarrow{p_k q_{j-1}}$  e  $\mathbf{i}$ , para todo  $j \in \{2, \dots, n-1\} - \{k\}$ . Após a execução da linha 5,





a pilha  $\mathcal{P}$  contém os pontos  $q_0$ ,  $q_1$  e  $q_2$ , nesta ordem, da base para o topo.

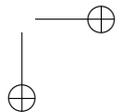
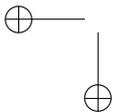
Considere a seguinte afirmação:

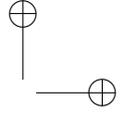
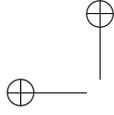
- (I) *Sempre que a linha 7 do algoritmo é atingida, todos os pontos de  $\mathcal{P}$  são pontos extremos de  $FC(S_{i-1})$ , com  $i \in \{1, \dots, n\}$ , na ordem em que eles ocorrem em um percurso anti-horário e a partir de  $q_0$  pelo fecho de  $S_{i-1}$ , em que  $S_{i-1}$  é o conjunto dado por*

$$S_{i-1} = \{q_0, q_1, \dots, q_{i-1}\}.$$

Esta afirmação é dita uma *invariante* do laço **enquanto** das linhas 7 a 16. A nossa prova é por indução no número  $h$  de vezes em que a linha 7 é atingida. O caso base corresponde à primeira vez em que a linha 7 é atingida (isto é,  $h = 1$ ). Neste ponto, temos que  $i = 3$  e  $\mathcal{P}$  contém apenas os pontos,  $q_0$ ,  $q_1$  e  $q_2$ , que foram empilhados em  $\mathcal{P}$  na linha 5, nesta ordem. Como os pontos  $q_0$ ,  $q_1$  e  $q_2$  não são colineares, eles definem um triângulo. Este triângulo corresponde ao fecho convexo,  $FC(S_2)$ , de  $S_2$ . Obviamente, os pontos extremos de  $FC(S_2)$  são os vértices,  $q_0$ ,  $q_1$  e  $q_2$ , do triângulo e, portanto,  $\mathcal{P}$  contém os pontos extremos de  $FC(S_{i-1}) = FC(S_2)$ . Como  $q_0$ ,  $q_1$  e  $q_2$  definem um percurso anti-horário a partir de  $q_0$  pelo fecho de  $S_2$ , e com relação ao referencial  $(O, \mathbf{i}, \mathbf{j})$ , podemos concluir que a afirmação (I) é válida para  $h = 1$ .

Assuma que a afirmação (I) é verdadeira para  $h = m$ , em que  $m$  é um inteiro positivo arbitrário (mas, fixo). Assuma também que quando  $h = m$ , o valor de  $i$  é menor do que  $n$ . Caso contrário, a condição do laço **enquanto** avaliaria para o valor lógico falso quando a linha 7 do algoritmo fosse atingida pela  $m$ -ésima vez e, consequentemente, o corpo do laço não seria executado. Agora, considere o caso em que  $h = m + 1$ ; isto é, o caso em que a linha 7 do algoritmo é atingida pela  $(m + 1)$ -ésima vez. Neste ponto, vamos “retroceder” uma iteração e analisar a execução do corpo do laço logo após a linha 7 do algoritmo ter sido atingida pela  $m$ -ésima vez. Observe que as linhas 8, 9 e 10 são executadas e, em seguida, temos uma situação de exclusão mútua: (1) as linhas 11 e 12 são executadas ou (2) a linha 14 é. O caso (1) ocorre se a condição do comando **se** da linha 10 tiver sido avaliada para o valor lógico verdadeiro. Caso contrário, temos o caso (2).





Pela hipótese de indução, quando a linha 7 do algoritmo é atingida pela  $m$ -ésima vez, todos os pontos em  $\mathcal{P}$  são pontos extremos de  $FC(S_{i-1})$ , em que  $i \leq n - 1$ . Além disso, como  $i > j$  para todo  $q_j \in S_{i-1}$ , o ponto  $q_i$  está à esquerda da reta  $\ell_j$  que passa por  $q_0$  e  $q_j$  e é orientada de  $q_0$  para  $q_j$ , para todo  $j \in \{1, \dots, i - 1\}$ . Logo, podemos concluir que o ponto  $q_i$  tem de ser um ponto extremo de  $FC(S_i)$ .

Se o caso (1) ocorre, então podemos concluir que o predicado  $\text{LEFT}(a, b, q_i)$  devolveu o valor lógico verdadeiro e, portanto, o ponto  $q_i$  está à esquerda da reta  $\ell$  por  $a$  e  $b$  orientada de  $a$  para  $b$  com relação a  $(O, \mathbf{i}, \mathbf{j})$ . Como o ponto  $b$  estava no topo da pilha, a hipótese de indução nos diz também que  $b$  ocorre depois do ponto  $a$  em um percurso anti-horário a partir de  $q_0$ , com relação a  $(O, \mathbf{i}, \mathbf{j})$ , pelo fecho de  $S_{i-1}$ . Logo, não há nenhum ponto de  $\mathcal{P}$  à esquerda da reta  $\ell'$  por  $q_0$  e  $b$  orientada de  $q_0$  para  $b$ . Como  $b = q_j$ , para algum  $j < i$ , o ponto  $q_i$  também está à esquerda da reta  $\ell'$ . Finalmente, pela escolha de  $q_0$ , o ponto  $q_i$  está à esquerda da reta  $\ell''$  (orientada) que passa por  $q_0$  e possui a mesma direção e sentido que  $\mathbf{i}$  (ver Figura 3.7).

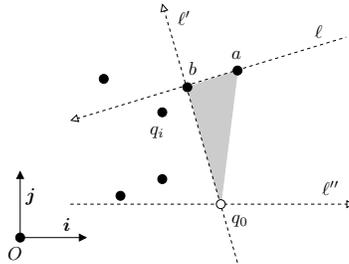
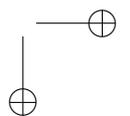
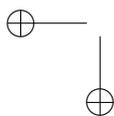
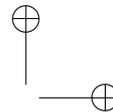
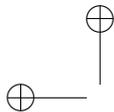


Figura 3.7:  $q_i$  está à esquerda das retas orientadas  $\ell$ ,  $\ell'$  e  $\ell''$ .

As observações do parágrafo anterior nos permitem concluir que  $q_i$  e os pontos em  $\mathcal{P}$  são precisamente os pontos extremos do fecho convexo do conjunto formado por eles. Além disso, como  $S_i - S_{i-1} = \{q_i\}$ , os pontos em  $\mathcal{P}$  e  $q_i$  são pontos extremos de  $FC(S_i)$ . Finalmente, a linha 11 empilha  $a$ ,  $b$  e  $q_i$  em  $\mathcal{P}$ , nesta ordem, e a linha 12 incrementa  $i$  em uma unidade. Logo, quando a linha 7 é atingida pela  $(m + 1)$ -ésima vez, todos os pontos em  $\mathcal{P}$  são pontos extremos de  $FC(S_{i-1})$

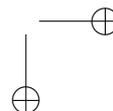
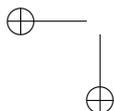


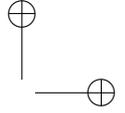
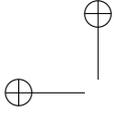


e, portanto, a afirmação (I) é verdadeira para  $h = m + 1$  no caso (1). Se o caso (2) ocorre, então podemos concluir que o predicado  $\text{LEFT}(a, b, q_i)$  devolveu o valor lógico falso e, portanto, o ponto  $q_i$  está à direita da reta  $\ell$ . Isto implica que o ponto  $b$  não é ponto extremo de  $S_i$ , embora seja de  $S_{i-1}$ , por hipótese. A linha 14 empilha o ponto  $a$  em  $\mathcal{P}$ , mas não o ponto  $b$ , de modo que quando a linha 7 é atingida pela  $(m+1)$ -ésima vez, todos os pontos em  $\mathcal{P}$  estavam em  $\mathcal{P}$  quando a linha 7 foi atingida pela  $m$ -ésima vez e na mesma ordem. Finalmente, como o valor da variável  $i$  não é atualizado no caso (2), a afirmação (I) é verdadeira para  $h = m + 1$  em (2) também.

**Teorema 3.7.** *O algoritmo Graham scan está correto.*

*Demonstração.* A corretude do pseudocódigo em Algoritmo 3.1 pode ser estabelecida a partir da veracidade da afirmação (I). Mas, para isso, precisamos mostrar que o laço **enquanto** nas linhas 7 a 16 sempre termina. De fato, podemos utilizar indução no número  $h$  de vezes em que a linha 7 é atingida para provar que *os pontos  $q_0$  e  $q_1$  são o primeiro e segundo ponto de  $\mathcal{P}$  da base para o topo, respectivamente, sempre que a linha 7 é atingida.* Isto implica que o valor lógico devolvido pelo predicado  $\text{LEFT}(a, b, q_i)$  não pode ser o valor lógico falso quando  $a = q_0$  e  $b = q_1$ . Caso contrário, o ponto  $b = q_1$  não estaria em  $\mathcal{P}$  quando a linha 7 fosse atingida logo em seguida. Por sua vez, esta observação garante que a linha 14 não pode ser executada, consecutivamente, por um número infinito de vezes. Além disso, quando  $\text{LEFT}(a, b, q_i)$  devolve o valor lógico verdadeiro, a variável  $i$  é incrementada em uma unidade. Consequentemente, o valor de  $i$  será eventualmente igual a  $n$  e, portanto, o laço **enquanto** sempre termina. Quando a linha 7 for atingida pela última vez (e, portanto, com  $i = n$ ), a afirmação (I) garante que todos os pontos em  $\mathcal{P}$  são pontos extremos de  $FC(S_{n-1})$ . Para concluir esta demonstração, precisamos mostrar que  $\mathcal{P}$  contém *todos* os pontos extremos de  $FC(S_{n-1})$ . Para tal, observe que  $q_0, q_1$  e  $q_2$  são colocados em  $\mathcal{P}$  antes do laço **enquanto** ser executado. Observe também que todo ponto  $q_j$  de  $S$ , com  $j \geq 3$ , é considerado pelo algoritmo na linha 10, pois  $i$  assume todos os valores em  $\{3, \dots, n-1\}$ . Além disso, o valor de  $i$  é incrementado de  $j$  para  $j+1$  se, e somente se, o predicado  $\text{LEFT}(a, b, q_j)$  retorna o valor lógico verdadeiro. Isto implica que todo ponto  $q_j$  é eventualmente inserido em  $\mathcal{P}$ . Afirmamos que se  $q_k$  é ponto extremo



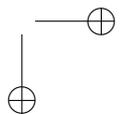
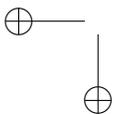


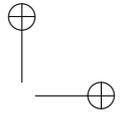
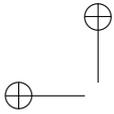
de  $FC(S_{n-1})$ , para todo  $k = 1, \dots, n - 1$ , e  $q_k$  é eventualmente retirado de  $\mathcal{P}$ , então ele é colocado de volta logo em seguida. De fato, se  $q_k$  fosse retirado de  $\mathcal{P}$  sem ser colocado de volta, então alguma chamada a  $\text{LEFT}(a, b, q_i)$  na linha 10, com  $b = q_k$  e  $i > k$ , retornaria o valor lógico falso, de forma que apenas o ponto  $a$  seria colocado de volta em  $\mathcal{P}$  na linha 14. Mas, se  $\text{LEFT}(a, b, q_k)$  retornasse o valor lógico falso, então  $q_k$  estaria à esquerda, com relação ao referencial  $(O, \mathbf{i}, \mathbf{j})$ , da reta por  $a$  e  $q$  orientada de  $a$  para  $q_i$ . Como  $i > k$ , temos que  $a$  não pode ser  $q_0$ . Logo, temos que  $q_k$  pertence ao triângulo  $\Delta(q_0 a q_i)$ , pois  $a$  está abaixo de  $b = q_k$  em  $\mathcal{P}$  e, portanto, a afirmação (I) nos diz que  $a$  precede  $q_k$  em um percurso anti-horário pelo fecho de  $S_{i-1}$ . Consequentemente, o ponto  $q_k$  não seria um ponto extremo de  $FC(S_{n-1})$ , o que contradiz nossa hipótese. Logo, podemos concluir que  $\mathcal{P}$  contém todos os pontos extremos de  $FC(S_{n-1})$ . Como  $S_{n-1} = S$ , o algoritmo produz como saída todos os pontos extremos de  $FC(S)$ .  $\square$

### 3.4.2 Complexidade

Esta seção é dedicada à *análise de complexidade de pior caso* do algoritmo de Graham, como descrito no pseudocódigo em Algoritmo 3.1. Este tipo de análise tem por objetivo obter uma cota superior para o esforço computacional do algoritmo em todas as possíveis entradas de um mesmo *tamanho*. No caso específico do algoritmo de Graham, o tamanho a que nos referimos é o número  $n$  de pontos de  $S$ .

Mais especificamente, a análise de pior caso determina uma cota superior para o valor  $f(n)$ , em que  $f : \mathbb{N} \rightarrow \mathbb{R}$  é uma função que associa ao tamanho  $n$  de uma entrada do algoritmo o maior número de *operações primitivas* que o algoritmo pode executar dentre todas as possíveis entradas de tamanho  $n$ . Uma *operação primitiva* é qualquer operação que tenha custo fixo, embora o custo possa variar de operação para operação. As operações primitivas e seus respectivos custos são especificados pelo *modelo de computação* usado para executar o algoritmo. Logo, para analisarmos um algoritmo, precisamos adotar um modelo de computação. Há vários modelos de computação disponíveis na literatura. A *Máquina de Turing* talvez seja o mais popular entre eles [66]. Porém, não há consenso sobre qual modelo é





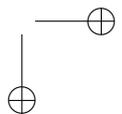
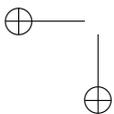
o melhor.

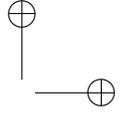
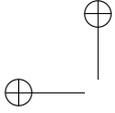
Em se tratando de algoritmos para problemas geométricos, tais como os algoritmos estudados neste livro, o modelo conhecido por *Real Random Access Machine* (ou simplesmente Real RAM) [105] é comum adotado. Neste modelo, as seguintes operações são primitivas e estão disponíveis a um custo unitário: *operações aritméticas* ( $+$ ,  $-$ ,  $\times$ ,  $/$ ), *comparações entre dois números reais* ( $<$ ,  $\leq$ ,  $=$ ,  $\neq$ ,  $\geq$ ,  $>$ ), *endereçamento indireto de memória (endereços inteiros apenas)* e *funções para extrair raízes, trigonométricas, exponencial e logaritmo*. O modelo Real RAM consiste de uma fita de entrada para leitura apenas, uma fita de saída para escrita apenas, um programa e uma memória de acesso aleatório. Não há limite de tamanho para a memória. Todas as computações são efetuadas em um processador. Este modelo é muito próximo de um computador real usado para executar programas em uma linguagem de alto-nível, mas cada posição de memória é capaz de armazenar um único número real de forma *exata*. Por isso, ao contrário dos computadores reais, as operações executadas no Real RAM não são susceptíveis a erros de arredondamento.

A cota superior para  $f(n)$  encontrada pela análise de complexidade de pior caso é, em geral, definida por uma notação especial,  $\mathcal{O}$ . Se  $h : \mathbb{N} \rightarrow \mathbb{R}$  é outra função de  $\mathbb{N}$  em  $\mathbb{R}$ , então escrevemos  $f \in \mathcal{O}(h)$  (lê-se  $f$  é “o” grande de  $h$ ) se existirem constantes  $n_0 \in \mathbb{N}$  e  $c \in \mathbb{R}$ , com  $c > 0$ , tais que  $f(n) \leq c \cdot h(n)$ , para todo  $n > n_0$ . Caso contrário, escrevemos  $f \notin \mathcal{O}(h)$ . Por exemplo, as funções  $f(n) = n^2 + 3n + 4$  e  $g(n) = 2^n + \lg n$  são tais que  $f \in \mathcal{O}(n^2)$  e  $g \in \mathcal{O}(2^n)$ . Temos também que  $f \in \mathcal{O}(n^3)$  e  $f \in \mathcal{O}(2^n)$ , mas  $g \notin \mathcal{O}(n^2)$  e  $g \notin \mathcal{O}(n^3)$ . Dizemos que um algoritmo  $\mathcal{A}$  para resolver um dado problema possui *complexidade (assintótica) de pior caso*  $\mathcal{O}(h)$  se, e somente se,  $f \in \mathcal{O}(h)$ , em que  $f$  e  $h$  são funções de  $\mathbb{N}$  em  $\mathbb{R}$  e  $f$  é a função que associa ao tamanho  $n$  da entrada de  $\mathcal{A}$  o maior número de operações primitivas que  $\mathcal{A}$  pode executar dentre todas as possíveis entradas de tamanho  $n$ . É comum omitirmos o termo “pior caso” e escrevermos apenas *complexidade* quando nos referimos à complexidade de pior caso de  $\mathcal{A}$ .

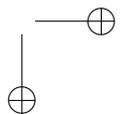
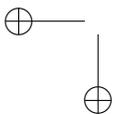
**Teorema 3.8.** *A complexidade do Graham scan é  $\mathcal{O}(n \lg n)$ .*

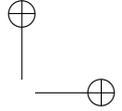
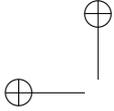
*Demonstração.* A linha 1 encontra o ponto  $p_k$  cuja segunda coorde-





nada possui o menor valor entre todas as segundas coordenadas dos pontos em  $S$ . Havendo empate, o ponto com menor valor na primeira coordenada (entre todos os pontos de menor segunda coordenada) é escolhido. O ponto  $p_k$  pode ser encontrado com, no máximo,  $2n - 2$  comparações entre dois números reais (ver Problema 3.11). Logo, a execução da linha 1 consome  $\mathcal{O}(n)$  operações primitivas. A linha 2 requer um número fixo de operações aritméticas para calcular o valor de  $\theta_i$ , para cada  $i$  em  $\{1, \dots, n\} - \{k\}$ . Como há  $n - 1$  ângulos para serem calculados, a execução da linha 2 também consome  $\mathcal{O}(n)$  operações primitivas. A linha 3 do pseudocódigo em Algoritmo 3.1 realiza a ordenação de um conjunto de  $n$  pares ordenados. Esta tarefa pode ser realizada por determinados algoritmos de ordenação que requerem, no pior caso,  $\mathcal{O}(n \lg n)$  operações primitivas [28]. Um exemplo é o algoritmo de ordenação por intercalação (conhecido em inglês por *mergesort*). As linhas 4, 5 e 6 requerem, cada, um número fixo de operações primitivas e, portanto, as três juntas utilizam  $\mathcal{O}(1)$  operações primitivas. Finalmente, as linhas 7 a 16 consomem, cada uma, um número fixo de operações primitivas, mas a linha 7 é executada  $m + 1$  vezes, enquanto cada uma das linhas 8 a 16 é executada  $m$  vezes, em que  $m$  corresponde ao número de iterações do laço **enquanto**. Afirmamos que  $m$  não pode ser maior do que  $2n - 5$ . De fato, cada iteração do laço retira os pontos  $a$  e  $b$  da pilha e, em seguida, (1) coloca os pontos  $a$ ,  $b$  e  $q_i$  na pilha, nesta ordem, ou (2) coloca apenas o ponto  $a$  de volta na pilha. No caso (1), a variável  $i$  é incrementada após a inserção de  $q_i$  na pilha e, portanto, o ponto  $q_i$  não é mais considerado nas iterações futuras do laço **enquanto**. No caso (2), o ponto  $b$  é um ponto  $q_j$  tal que  $j < i$ . Isto significa que  $b$  jamais retornará à pilha em iterações futuras, pois a variável  $i$  não é jamais decrementada. Daí, podemos concluir que o número  $m$  de iterações do laço não pode exceder  $(n - 3) + (n - 2) = 2n - 5$ , em que  $n - 3$  é o total de pontos considerados pelo algoritmo (isto é, os pontos  $q_i$ 's na chamada  $\text{LEFT}(a, b, q_i)$  da linha 10) e  $(n - 2)$  é o total de pontos que podem ser removidos permanentemente de  $\mathcal{P}$  (lembre-se da prova da afirmação (I) que  $q_0$  e  $q_1$  sempre estarão em  $\mathcal{P}$  quando a linha 7 é atingida). Daí, concluímos que o total de operações primitivas realizadas durante toda execução do laço **enquanto** é  $\mathcal{O}(n)$ . Finalmente, a linha 17 também requer  $\mathcal{O}(n)$  operações primitivas, pois  $\mathcal{P}$  pode possuir até  $n$  pontos. Adicionando todas as cotas supe-





riores que obtivemos acima para as linhas 1 a 17 do algoritmo, isto é,

$$\mathcal{O}(n) + \mathcal{O}(n) + \mathcal{O}(n \lg n) + \mathcal{O}(1) + \mathcal{O}(n) + \mathcal{O}(n),$$

temos que a complexidade do algoritmo Graham scan é  $\mathcal{O}(n \lg n)$ . Isto porque se  $f \in \mathcal{O}(n)$  ou  $f \in \mathcal{O}(1)$ , então  $f \in \mathcal{O}(n \lg n)$ . Além disso, se  $f \in \sum_{h=1}^N \mathcal{O}(n \lg n)$ , em que  $N$  é uma constante, então temos que

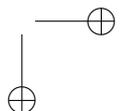
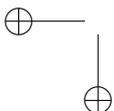
$$f \in \mathcal{O}(n \lg n).$$

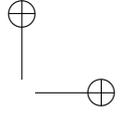
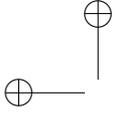
□

### 3.5 Cota inferior

A complexidade de pior caso de um algoritmo  $\mathcal{A}$  para um dado problema  $\mathcal{P}$  nos dá uma ideia do esforço computacional do algoritmo nas entradas que demandam esforço máximo. Uma vez que a complexidade de pior caso de  $\mathcal{A}$  tenha sido determinada, é natural nos perguntarmos se há algum algoritmo para  $\mathcal{P}$  que possua complexidade de pior caso assintoticamente menor do que a de  $\mathcal{A}$ . Uma resposta para esta pergunta não depende do algoritmo  $\mathcal{A}$ , mas sim do problema  $\mathcal{P}$ , pois o que queremos é saber se o problema pode ser resolvido, *no pior caso*, com menos esforço computacional do que aquele realizado por  $\mathcal{A}$ .

Em outras palavras, o que queremos é determinar a *complexidade do problema*  $\mathcal{P}$ , que nada mais é do que o esforço computacional que *todo e qualquer* algoritmo (existente ou não) para  $\mathcal{P}$  tenha *necessariamente* de dispendir para resolver  $\mathcal{P}$  quando uma entrada representativa do pior caso do algoritmo é fornecida. Obviamente, a complexidade de  $\mathcal{P}$  não pode ser assintoticamente maior do que a complexidade de pior caso de um algoritmo que o resolva. No entanto, quando essas complexidades coincidem (assintoticamente), podemos concluir que o algoritmo não pode ser melhorado, pois ele realiza exatamente o esforço mínimo necessário para resolver  $\mathcal{P}$  em entradas representativas do pior caso do algoritmo. Por outro lado, se a complexidade do algoritmo é assintoticamente maior do que a de  $\mathcal{P}$ , então podemos concluir que o algoritmo não é o melhor possível para  $\mathcal{P}$ .





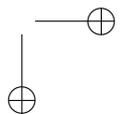
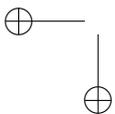
Quando calculamos a complexidade de um problema  $\mathcal{P}$ , determinamos em geral uma *cota inferior*. Mais especificamente, seja  $g : \mathbb{N} \rightarrow \mathbb{R}$  a função que associa ao tamanho  $n$  de uma entrada de  $\mathcal{P}$  o número mínimo de operações primitivas que todo e qualquer algoritmo para  $\mathcal{P}$  deve executar em uma entrada de pior caso (do algoritmo) com tamanho  $n$ . Então, escrevemos  $g \in \Omega(h)$  (lê-se  $g$  é ômega de  $h$ ), em que  $h$  é uma função de  $\mathbb{N}$  em  $\mathbb{R}$ , se existirem constantes  $n_0 \in \mathbb{N}$  e  $c \in \mathbb{R}$ , com  $c > 0$ , tais que  $c \cdot h(n) \leq g(n)$ , para todo  $n > n_0$ . Caso contrário, escrevemos  $g \notin \Omega(h)$ . Por exemplo, se  $g : \mathbb{N} \rightarrow \mathbb{R}$  é tal que  $g(n) = n^2 - 2n$ , então  $g \in \Omega(n)$  e  $g \in \Omega(n^2)$ , mas  $g \notin \Omega(n^3)$ .

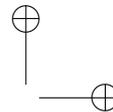
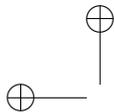
Sejam  $g : \mathbb{N} \rightarrow \mathbb{R}$  e  $f : \mathbb{N} \rightarrow \mathbb{R}$  as funções que representam a complexidade de um problema  $\mathcal{P}$  e a complexidade de um algoritmo  $\mathcal{A}$  que resolve  $\mathcal{P}$ . Então,  $\mathcal{A}$  é dito um algoritmo *ótimo* para  $\mathcal{P}$  se, e somente se, temos que  $g \in \Omega(h)$  e  $f \in \mathcal{O}(h)$ , em que  $h : \mathbb{N} \rightarrow \mathbb{R}$  é alguma função. O Teorema 3.8 estabeleceu que a complexidade do algoritmo Graham *scan* é  $\mathcal{O}(n \lg n)$ . Um pouco mais adiante, provaremos que a complexidade do *problema de calcular o fecho convexo de um conjunto  $S$  de  $n$  pontos em  $\mathbb{E}^2$*  é  $\Omega(n \lg n)$ . Isto nos permitirá concluir que o Graham *scan* é ótimo para o problema do fecho convexo.

Para provar que o Graham *scan* é ótimo, precisamos determinar que a complexidade do problema do fecho convexo é  $\Omega(n \lg n)$ . Para tal, utilizaremos uma técnica conhecida como *redução*, que se baseia em relacionar a complexidade de um problema,  $\mathcal{P}_2$ , com a complexidade de um problema,  $\mathcal{P}_1$ , que de alguma maneira já foi determinada. A relação é do tipo *a complexidade de  $\mathcal{P}_2$  não pode ser menor do que a de  $\mathcal{P}_1$* . Uma *redução de  $\mathcal{P}_1$  para  $\mathcal{P}_2$*  consiste dos três passos dados a seguir:

1. A entrada de  $\mathcal{P}_1$  é convertida em uma entrada de  $\mathcal{P}_2$ .
2. O problema  $\mathcal{P}_2$  é resolvido.
3. A saída de  $\mathcal{P}_2$  é transformada em uma solução correta para  $\mathcal{P}_1$ .

Se os passos 1 e 3, juntos, podem ser executados com  $\mathcal{O}(h(n))$  operações primitivas, em que  $n$  é o tamanho da entrada de  $\mathcal{P}_1$ , então dizemos que  $\mathcal{P}_1$  é  *$\mathcal{O}(h(n))$ -reduzível para  $\mathcal{P}_2$*  e denotamos este fato





por

$$\mathcal{P}_1 \propto_{\mathcal{O}(h(n))} \mathcal{P}_2.$$

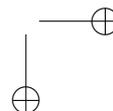
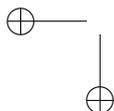
Em geral, a relação de redução não é simétrica. Quando ela é, dizemos que os problemas  $\mathcal{P}_1$  e  $\mathcal{P}_2$  são *equivalentes*. A seguinte proposição estabelece o poder da técnica de redução para os propósitos que temos:

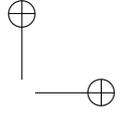
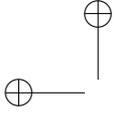
**Teorema 3.9.** *Se o problema  $\mathcal{P}_1$  requer pelo menos  $g(n)$  operações primitivas, no pior caso, para ser resolvido e é  $\mathcal{O}(h(n))$ -reduzível para o problema  $\mathcal{P}_2$  (isto é,  $\mathcal{P}_1 \propto_{\mathcal{O}(h(n))} \mathcal{P}_2$ ), então  $\mathcal{P}_2$  requer pelo menos  $g(n) - \mathcal{O}(h(n))$  operações primitivas, no pior caso, para ser resolvido.*

*Demonstração.* Consulte o livro [54], por exemplo. □

O Teorema 3.9 nos permite obter a cota inferior  $g(n) - \mathcal{O}(h(n))$  para o problema  $\mathcal{P}_2$  a partir de uma cota inferior para  $\mathcal{P}_1$ . Tudo que precisamos fazer é nos certificar de que podemos reduzir  $\mathcal{P}_1$  para  $\mathcal{P}_2$ ; isto é, se conseguimos executar os passos 1, 2 e 3 acima. Esta tarefa nem sempre é fácil, mas a complexidade de vários problemas geométricos tem sido determinada desta forma. Para obter uma cota inferior para o problema do fecho convexo, usamos uma cota inferior estabelecida para o problema da ordenação e reduziremos este para aquele.

Em Teoria da Computação é um fato bem conhecido que o *problema* da ordenação dos  $n$  elementos de um dado conjunto  $N$ , sobre o qual uma ordem total foi definida, requer  $\Omega(n \lg n)$  operações primitivas, no pior caso, para ser resolvido por qualquer algoritmo de ordenação [28]. Esta cota inferior foi obtida com um modelo de computação denominado *árvore de decisão*. Com este modelo é possível mostrar que qualquer algoritmo de ordenação deve realizar  $\Omega(n \lg n)$  comparações entre dois elementos (de um conjunto de entrada representativo do pior caso do algoritmo) para produzir uma sequência ordenada desses elementos. Por outro lado, existem algoritmos baseados em comparações entre pares de elementos do conjunto de entrada, tais como o *heapsort* e o *mergesort*, cujas complexidades são  $\mathcal{O}(n \lg n)$  e, portanto, eles são algoritmos ótimos para o problema da ordenação.





**Teorema 3.10.** *Seja  $F$  um conjunto de  $n$  pontos em  $\mathbb{E}^2$ , com  $n \geq 3$ . Então, qualquer algoritmo que determina os vértices do fecho convexo,  $FC(F)$ , de  $F$  na ordem em que eles ocorrem em um percurso horário ou anti-horário pelo fecho de  $F$  efetua necessariamente  $\Omega(n \lg n)$  operações primitivas sempre que  $F$  é uma entrada de pior caso.*

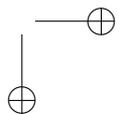
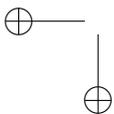
*Demonstração.* Seja  $S = (u_1, \dots, u_n)$  uma sequência de  $n$  números reais positivos. Tal sequência é uma entrada para o problema da ordenação. Esta entrada pode ser convertida em uma entrada para o problema do fecho convexo. De fato, dada  $S$  definimos a sequência  $S' = ((u_i, u_i^2))_{i=1}^n$ . Note que os pontos de  $S'$  residem no gráfico da parábola  $y = u^2$  (ver Figura 3.8). Suponha que utilizemos qualquer algoritmo  $\mathcal{A}$  para calcular o fecho convexo do conjunto  $F$  formado pelos pontos de  $S'$ . Como parábolas são convexas, todo ponto  $(u_i, u_i^2)$  em  $S'$  pertence ao fecho de  $F$  (ver Figura 3.9). Por convenção, sabemos que a saída de todo algoritmo para o problema do fecho convexo é uma sequência formada pelos pontos em  $S'$  na ordem em que eles ocorrem em um percurso anti-horário pelo fecho de  $F$ . O primeiro ponto desta sequência é aquele com menor coordenada  $y$ . Como  $y = u^2$  é uma função estritamente crescente, o ponto de menor coordenada  $y$  também é o ponto de menor coordenada  $x$ . Portanto, se

$$S'' = ((x_j, x_j^2))_{j=1}^n$$

é a sequência de saída de  $\mathcal{A}$ , então devemos ter que  $x_j < x_{j+1}$ , para  $j = 1, \dots, n - 1$ . Finalmente, a partir de  $S''$ , definimos a sequência  $S''' = (x_j)_{j=1}^n$ , que é exatamente a solução da instância do problema da ordenação que tem como entrada a sequência  $S$ . Observe que as conversões de  $S$  para  $S'$  e  $S''$  para  $S'''$  podem ser realizadas com  $\mathcal{O}(n)$  operações primitivas cada. Como o problema da ordenação por comparações de chaves requer  $\Omega(n \lg n)$  operações primitivas para produzir  $S'''$  a partir de  $S$ , o Teorema 3.9 nos diz que o problema do fecho convexo requer, pelo menos,  $n \lg n - \mathcal{O}(n)$  operações primitivas para ser resolvido. Como  $(n \lg n - \mathcal{O}(n)) \in \Omega(n \lg n)$ , podemos concluir que se a complexidade de pior caso de  $\mathcal{A}$  é dada por  $f : \mathbb{N} \rightarrow \mathbb{R}$ , então

$$f(n) \in \Omega(n \lg n).$$

□



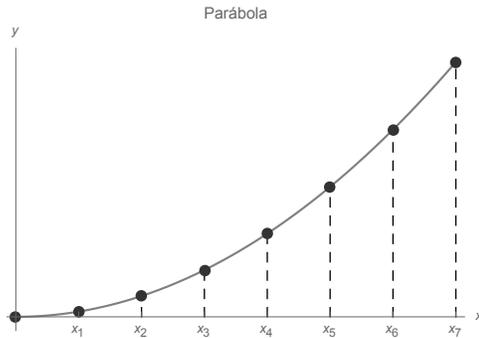


Figura 3.8: A parábola de equação  $y - x^2 = 0$ .

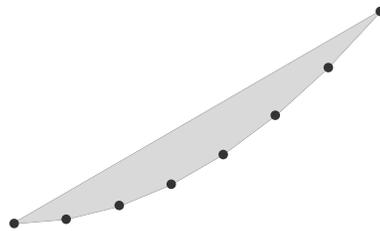
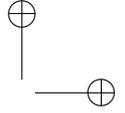
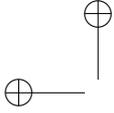


Figura 3.9: O fecho convexo dos pontos na Figura 3.9.

**Corolário 3.11.** *O Graham scan é um algoritmo ótimo.*

O resultado estabelecido pelo Teorema 3.10 usa o fato que os vértices do fecho de  $F$  devem ser produzidos na ordem em que eles ocorrem em um percurso horário ou anti-horário pelo fecho de  $F$ . E se desejássemos obter os vértices do fecho de  $F$  em *qualquer* ordem, ou seja, obter os pontos extremos do fecho de  $F$  em *qualquer* ordem? Franco Preparata e Michael Shamos provaram que a cota inferior do Teorema 3.10 se mantém [105]. Isto é, o problema do fecho convexo não se torna mais fácil se removermos a restrição de produzir pontos em ordem.



## 3.6 Exercícios

3.1 Prove o Lema 3.2.

3.2 Seja  $S$  o subconjunto de  $\mathbb{E}^2$  consistindo dos pontos de coordenadas  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$  e  $(1, 1)$  com relação ao referencial afim padrão de  $\mathbb{E}^2$ . Então, use o Lema 3.1 para mostrar que o fecho convexo,  $FC(S)$ , de  $S$  é o quadrado com vértices nos pontos de  $S$ .

3.3 Seja  $S$  um subconjunto de  $\mathbb{E}^2$  com pelo menos quatro pontos. Mostre que  $S$  pode ser particionado em dois conjuntos,  $A$  e  $B$ , tais que

$$FC(A) \cap FC(B) \neq \emptyset.$$

Lembre-se de que  $\{A, B\}$  é uma partição de  $S$  se, e somente se,

$$A \cup B = S \quad \text{e} \quad A \cap B = \emptyset.$$

3.4 Seja  $S$  um subconjunto não vazio de  $\mathbb{E}^2$  com pelo menos três pontos não colineares. Então, mostre que  $FC(S)$  é o polígono convexo com o menor perímetro e menor área contendo o conjunto  $S$ .

3.5 Seja  $S$  um subconjunto não vazio de  $\mathbb{E}^2$  com pelo menos três pontos não colineares. Então, mostre que pelo menos um dos lados do retângulo de menor área contendo  $S$  cobre uma aresta de  $FC(S)$ .

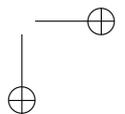
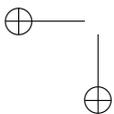
3.6 Seja  $f : \mathbb{E}^d \rightarrow \mathbb{E}^m$  um mapa afim. Então, para quaisquer subconjuntos convexos,  $U$  em  $\mathbb{E}^d$  e  $V$  em  $\mathbb{E}^m$ , mostre que os conjuntos  $f(U) \subseteq \mathbb{E}^m$  e  $f^{-1}(V) \subseteq \mathbb{E}^d$  são convexos. Lembre-se de que

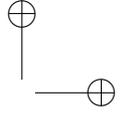
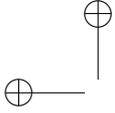
$$f(U) = \{b \in \mathbb{E}^m \mid \exists a \in U, b = f(a)\}$$

é a *imagem direta de  $U$  segundo  $f$*  e que

$$f^{-1}(V) = \{a \in \mathbb{E}^d \mid \exists b \in V, b = f(a)\}$$

é a *imagem inversa de  $V$  segundo  $f$* .





- 3.7 Considere o subconjunto  $S$  de  $\mathbb{E}^2$  consistindo dos pontos que pertencem ao ramo direito da hipérbole de equação  $x^2 - y^2 = 1$ ; isto é

$$S = \{(x, y) \in \mathbb{E}^2 \mid x^2 - y^2 \geq 1, x \geq 0\}.$$

Prove que  $S$  é convexo. Qual é o fecho convexo de  $S \cup \{(0, 0)\}$ ? O fecho convexo de um subconjunto fechado de  $\mathbb{E}^d$  é necessariamente um subconjunto fechado de  $\mathbb{E}^d$ ? Justifique sua resposta.

- 3.8 O fecho afim,  $FA(F)$ , de um subconjunto não vazio,  $F$ , de  $\mathbb{E}^d$  é o conjunto de todas as combinações afins de pontos de  $F$ ; isto é, um ponto  $p$  em  $\mathbb{E}^d$  pertence ao conjunto  $FA(F)$  se, e somente se,

$$p = \sum_{i=1}^n \alpha_i \cdot p_i, \quad \sum_{i=1}^n \alpha_i = 1,$$

para alguma lista,  $((p_i, \alpha_i))_{i=1}^n$ , de  $n$  pontos ponderados de  $F$ , com  $n \in \mathbb{Z}$  e  $n \geq 1$ . O conjunto  $FA(F)$  é um subespaço afim de  $\mathbb{E}^d$  (ver Problema 2.14). Prove que  $FA(F)$  é também convexo.

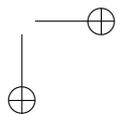
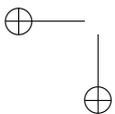
- 3.9 Utilize o Teorema de Carathéodory (Teorema 3.4) para provar a seguinte afirmação: se  $S$  é um subconjunto compacto de  $\mathbb{E}^d$ , então o fecho convexo,  $FC(S)$ , de  $S$  é um subconjunto compacto de  $\mathbb{E}^d$ .

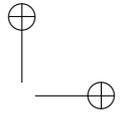
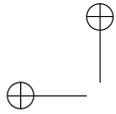
- 3.10 Seja  $S$  um subconjunto não vazio de pontos de  $\mathbb{E}^2$ . Dado um ponto  $p \in S$ , dizemos que  $S$  é um *polígono em forma de estrela com relação ao ponto  $p$*  se, e somente se, o segmento de reta  $\overline{px}$  está contido em  $S$  para todo  $x \in S$ , isto é, se, e somente se,

$$(1 - \lambda) \cdot p + \lambda \cdot x$$

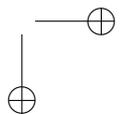
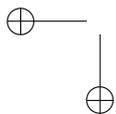
é um ponto de  $S$  para todo  $\lambda \in \mathbb{R}$ , com  $0 \leq \lambda \leq 1$ . Dizemos que  $S$  é um *polígono em forma de estrela* se, e somente se, ele é um polígono em forma de estrela com relação a algum  $p \in S$ . Então,

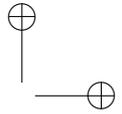
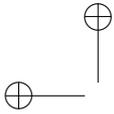
- Prove que todo subconjunto convexo e não vazio de  $\mathbb{E}^2$  é um polígono em forma de estrela.
- Mostre que há subconjuntos de  $\mathbb{E}^2$  que são polígonos em forma de estrela e não são convexos.





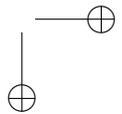
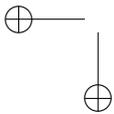
- (c) Mostre que há subconjuntos de  $\mathbb{E}^2$  que são polígonos, mas que não são em forma de estrela.
- (d) Dado um subconjunto de  $S$  de  $\mathbb{E}^2$  tal que  $S$  é um polígono em forma de estrela, denominamos de  $\mathcal{N}(S)$  o conjunto de todos os pontos  $p \in S$  tais que  $S$  é um polígono em forma de estrela com relação a  $p$ . Prove que  $\mathcal{N}(S)$  é convexo. *O conjunto  $\mathcal{N}(S)$  é denominado de núcleo do polígono  $S$ . Informalmente, qual é a principal característica dos pontos de  $\mathcal{N}(S)$ ?*
- 3.11 Escreva um pseudocódigo para as operações da linha 1 do algoritmo Graham *scan* (Algoritmo 3.1); isto é, forneça uma descrição detalhada das operações do Graham *scan* que determinam  $p_k$ .
- 3.12 Escreva um pseudocódigo para as operações da linha 2 do algoritmo Graham *scan* (Algoritmo 3.1); isto é, forneça uma descrição detalhada das operações do Graham *scan* que calculam  $(\theta_i)_{i=1, i \neq k}^n$ .
- 3.13 Escreva um pseudocódigo para as operações da linha 3 do algoritmo Graham *scan* (Algoritmo 3.1); isto é, forneça uma descrição detalhada das operações do Graham *scan* que calculam  $(q_i)_{i=1}^{n-1}$ .
- 3.14 Descreva um conjunto de  $n$  pontos em  $\mathbb{E}^2$  que corresponde a uma entrada de pior caso para o algoritmo Graham *scan*. E quanto a uma entrada de *melhor caso*, isto é, a uma entrada de tamanho  $n$  que faça com que o algoritmo realize o menor número de operações primitivas? Nas duas respostas, a descrição da entrada deve ser feita para qualquer  $n$  (isto é, não fixe  $n!$ ).
- 3.15 Se o conjunto de pontos  $S$ , dado como entrada para o algoritmo Graham *scan*, possuir três pontos colineares, então o algoritmo descrito em Algoritmo 3.1 pode ser facilmente modificado para lidar com este caso geral. Para tal, modifique a linha 3 de tal forma que, se dois ou mais ângulos forem iguais, então os pontos correspondentes ao mesmo ângulo são descartados, exceto





aquele com a maior coordenada  $y$ , ou seja, o mais distante de  $p_k$ . Sendo assim, a lista de pontos produzida pela linha 3 pode conter menos do que  $n - 1$  pontos. Escreva a linha 3 em pseudocódigo e, em seguida, mostre que o algoritmo modificado está correto.

- 3.16 Na discussão sobre a veracidade da propriedade (I) usada na prova de corretude do algoritmo Graham *scan*, observamos que o ponto  $q_i$  está à esquerda das retas orientadas  $\ell$ ,  $\ell'$  e  $\ell''$ , como mostra a Figura 3.7. Se assumirmos que a pilha  $\mathcal{P}$  contém apenas pontos extremos de  $FC(S_{i-1})$ , então por que os pontos em  $R \cup \{q_i\}$ , em que  $R$  é o conjunto de pontos em  $\mathcal{P}$ , são pontos extremos de  $FC(S_i)$ ? Para responder esta pergunta, argumente inicialmente que  $q_i$  é um ponto extremo de  $FC(S_i)$ . Em seguida, observe que se algum ponto em  $R$  não fosse ponto extremo de  $FC(S_i)$ , então haveria um ponto  $a$  do polígono convexo com vértices em  $\mathcal{P}$  tal que algum ponto em  $\mathcal{P}$  é uma combinação convexa de  $q_i$  e  $a$ . Mas, isto não deveria ser possível. Por quê?
- 3.17 A matemática húngara Ester Szekeres (1910-2005) propôs o seguinte problema: *qual é o maior número de pontos de  $\mathbb{E}^2$ , em posição geral, que não contém os vértices de um  $n$ -ágono convexo?* Por *posição geral*, entenda aqui que quaisquer três pontos do conjunto em questão não podem ser colineares. Este problema foi denominado *Problema do Final Feliz*, pois logo após ele ser anunciado pela até então matemática Ester Klein, o matemático húngaro George Szekeres (1911-2005), com o auxílio de Paul Erdős, resolveu-o para  $n = 1, 2, 3, 4, 5$  e os dois se casaram. Para você ter uma ideia melhor do problema, note que se  $n = 3$ , então estamos perguntando qual é o maior número de pontos de  $\mathbb{E}^2$ , em posição geral, que não contém os vértices de um triângulo. Obviamente, este número é 2, pois quaisquer três pontos de  $\mathbb{E}^2$  não colineares constituem os vértices de um triângulo. Quando  $n = 4$ , estamos perguntando qual é o maior número de pontos de  $\mathbb{E}^2$ , em posição geral, que não contém os vértices de um quadrilátero convexo. Szekeres e Erdős mostraram que este número é 4. Quando  $n = 5$ , estamos perguntando qual é o número máximo de pontos de  $\mathbb{E}^2$ , em posição geral, que não contém os vértices de um pentágono convexo. Szekeres



e Erdős mostraram que este número é 8. Podemos notar, portanto, que este número dobra quando  $n$  varia de 3 a 5. Então, podemos conjecturar que a solução do problema é  $2^{n-2}$ . No entanto, até onde saibamos, este problema ainda está em *aberto*. O que conhecemos hoje é que o número máximo,  $g(n)$ , de pontos é tal que  $g(n) \geq 2^{n-2} + 1$  e  $g(n) \leq \binom{2n-5}{n-2} + 2$ , para todo  $n \geq 6$ . Quem sabe você não consegue encontrar um resultado melhor?

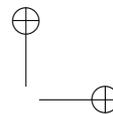
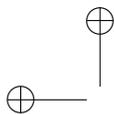
- 3.18 Mostre que a função  $f : \mathbb{N} \rightarrow \mathbb{R}$  dada por  $f(n) = n^2 + 3n + 4$  é tal que  $f \in \mathcal{O}(n^2)$ . Para tal, aplique a definição de notação  $\mathcal{O}$ , isto é, encontre constantes  $n_0 \in \mathbb{N}$  e  $c \in \mathbb{R}$  tais que  $f(n) \leq c \cdot n^2$ ,  $\forall n \geq n_0$ .
- 3.19 Mostre que a função  $g : \mathbb{N} \rightarrow \mathbb{R}$  dada por  $g(n) = n^2 - 2n$  é tal que  $g \in \Omega(n^2)$ . Para tal, aplique a definição de notação  $\Omega$ , isto é, encontre constantes  $n_0 \in \mathbb{N}$  e  $c \in \mathbb{R}$  tais que  $c \cdot n^2 \leq g(n)$ ,  $\forall n \geq n_0$ .
- 3.20 Seja  $f : \mathbb{N} \rightarrow \mathbb{R}$  uma função tal que  $f \in \mathcal{O}(n)$ . Então, mostre que  $f \in \mathcal{O}(n \lg n)$ . Mostre também que nem sempre é o caso que

$$f \in \Omega(n \lg n).$$

- 3.21 Sejam  $f : \mathbb{N} \rightarrow \mathbb{R}$ ,  $g : \mathbb{N} \rightarrow \mathbb{R}$  e  $h : \mathbb{N} \rightarrow \mathbb{R}$  funções tais que  $g(n) = \sum_{i=1}^N f(n)$  e  $f(n) \in \mathcal{O}(h(n))$ , em que  $N$  é um número natural constante (que não depende de  $n$ ). Então, mostre que

$$g \in \mathcal{O}(h(n)).$$

- 3.22 Mostre que se  $S$  é um conjunto de  $n$  pontos escolhidos aleatoriamente a partir de uma distribuição uniforme de pontos em um quadrado unitário em  $\mathbb{E}^2$ , então o número *esperado* de vértices do polígono correspondente ao fecho convexo de  $S$  é da ordem de  $\mathcal{O}(\lg n)$ .
- 3.23 Descreva um algoritmo com complexidade  $\mathcal{O}(n)$  para calcular o fecho convexo,  $FC(S)$ , do conjunto  $S$  de vértices de um dado polígono  $\mathcal{P}$  com  $n$  vértices. Assuma que os vértices de  $\mathcal{P}$  são



dados na ordem em que eles ocorrem em um percurso anti-horário pela fronteira de  $\mathcal{P}$ . A saída do algoritmo consiste dos vértices do fecho de  $S$  na ordem em que eles ocorrem em um percurso anti-horário pelo fecho. *Seja bastante cuidadoso, pois, por diversas vezes, a comunidade científica propôs algoritmos incorretos para o mesmo problema. Portanto, se você encontrar uma “solução” que lhe parece óbvia e cuja complexidade é  $\mathcal{O}(n)$ , desconfie!*

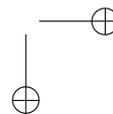
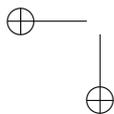
- 3.24 Seja  $S = \{p_1, \dots, p_n\}$  um subconjunto finito com  $n$  pontos de  $\mathbb{E}^2$ . Então, o *diâmetro* de  $S$ , denotado por  $\text{diam}(S)$ , é definido como

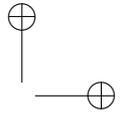
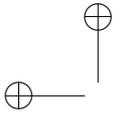
$$\text{diam}(S) = \max_{i,j \in \{1, \dots, n\}} \{d(p_i, p_j)\},$$

isto é, o diâmetro de  $S$  é a maior distância euclidiana entre dois pontos quaisquer de  $S$ . Agora, resolva os seguintes problemas:

- (a) Mostre que o diâmetro do conjunto de  $S$  é igual a distância euclidiana de dois vértices do fecho convexo,  $FC(S)$ , de  $S$ .
- (b) Uma *reta de suporte* para  $S$  é uma reta  $\ell$  que intersecta  $FC(S)$  e tal que todos os pontos de  $S$  estão em um dos dois semiespaços de  $\mathbb{E}^2$  definidos por  $\ell$ . Mostre que  $\text{diam}(S)$  é igual à distância máxima entre retas paralelas de suporte de  $S$ .
- (c) Dois pontos,  $a$  e  $b$ , de  $S$  são ditos *antípodas* se eles admitem retas de suporte paralelas, isto é, se existem retas de suporte paralelas tais que uma delas passa por  $a$  e a outra passa por  $b$ . Desenvolva um algoritmo para enumerar todos os pares antípodas de um dado subconjunto finito,  $S$ , de pontos de  $\mathbb{E}^2$ . Qual é a complexidade de seu algoritmo?

O problema proposto pode ser resolvido por um algoritmo de tempo  $\mathcal{O}(n^2)$ , onde  $n$  é o número de vértices do polígono correspondente ao fecho convexo dos pontos de entrada. Isto porque os pares antípodas do conjunto de entrada devem obrigatoriamente ser vértices do fecho convexo dos pontos de entrada (isto é, pontos extremos). Logo, basta calcular o fecho convexo e verificar, para cada par de vértices do polígono correspondente ao fecho, se

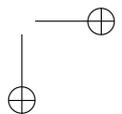
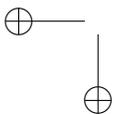


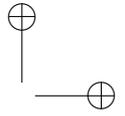


seus componentes são pontos antípodas. Este procedimento utiliza  $\mathcal{O}(n^2)$  operações primitivas, mas podemos calcular o fecho convexo em  $\mathcal{O}(n \log n)$  usando o *Graham scan*. Então, a pergunta é: podemos enumerar os pontos antípodas com  $\mathcal{O}(n \log n)$  operações também? A resposta é *sim!*

Denomine o polígono correspondente ao fecho convexo dos pontos de entrada por  $\mathcal{P}$ . Seja  $v_i$  o vértice de  $\mathcal{P}$  de maior coordenada  $y$ . Se existirem dois vértices com esta mesma coordenada, considere o mais à direita. Seja  $v_j$  o vértice de  $\mathcal{P}$  de menor coordenada  $y$ . Se existirem dois vértices com esta mesma coordenada, considere o mais à esquerda. Note que as retas horizontais,  $\ell_i$  e  $\ell_j$ , passando por  $v_i$  e  $v_j$ , respectivamente, são retas de suportes de  $\mathcal{P}$ . Logo, os vértices  $v_i$  e  $v_j$  são antípodas (ver Figura 3.10). Seja  $\theta_i$  o ângulo exterior a  $\mathcal{P}$  que a reta  $\ell_i$  faz com a reta que passa por  $v_i$  e  $v_{i+1}$ , em que  $v_{i+1}$  é o vértice que sucede  $v_i$  em um percurso horário pela fronteira de  $\mathcal{P}$ . Analogamente, seja  $\theta_j$  o ângulo exterior a  $\mathcal{P}$  que a reta  $\ell_j$  faz com a reta que passa por  $v_j$  e  $v_{j+1}$ , onde  $v_{j+1}$  é o vértice que sucede  $v_j$  em um percurso horário pela fronteira de  $\mathcal{P}$ . Note que se  $\theta_i < \theta_j$ , o vértice  $v_{i+1}$  e o vértice  $v_j$  são antípodas (por quê?). Use esta observação para criar um algoritmo que calcula todos os pares de vértices antípodas de  $\mathcal{P}$  em  $\mathcal{O}(m)$ , em que  $m$  é o número de vértices de  $\mathcal{P}$ . Combine este algoritmo com um algoritmo ótimo para calcular  $\mathcal{P}$  de modo a obter um algoritmo de complexidade  $\mathcal{O}(n \log n)$  para o problema proposto neste item, em que  $n$  é o número de pontos do conjunto  $S$  dado como entrada no problema.

- (d) Usando o algoritmo do item anterior, descreva um algoritmo para calcular o diâmetro,  $\text{diam}(S)$ , de  $S$ . Qual é a complexidade do seu algoritmo? Descreva um conjunto,  $S$ , de pontos que represente uma entrada de pior caso do algoritmo.
- (e) Seja  $S$  um conjunto de  $n$  pontos de  $\mathbb{E}^2$ . O prolífico matemático húngaro Paul Erdős (1923-1996) mostrou, em 1946, que a distância máxima entre dois pontos quaisquer





de  $S$  pode ser atingida por, no máximo,  $n$  pares dos  $\binom{n}{2}$  possíveis pares de pontos de  $S$ . O resultado de Erdős confirma a noção intuitiva de que nem todo par de antípodas fornece a maior distância entre dois pontos de  $S$ . Descreva um subconjunto  $S$  no qual a distância máxima entre dois pontos quaisquer de  $S$  seja atingida por *exatamente*  $n$  pontos?

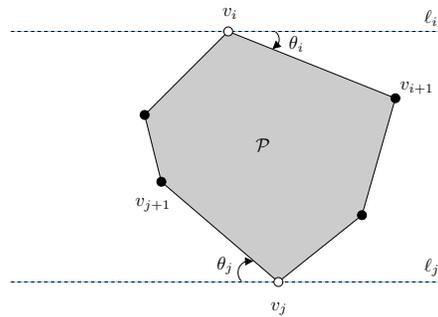
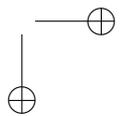
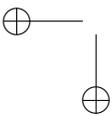
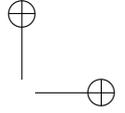
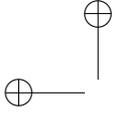


Figura 3.10: Um par  $(v_i, v_j)$  de vértices antípodas.

- 3.25 Grades retangulares de pontos espaçados uniformemente (também conhecidas como *lattices*) são utilizadas em várias aplicações. Por exemplo, imagens digitais bidimensionais são funções definidas em uma grade,  $D = [x_1, x_2] \times [y_1, y_2]$ , de pontos de  $\mathbb{Z}^2 = \mathbb{Z} \times \mathbb{Z}$ . Seja  $S$  um conjunto de  $n$  pontos de um *lattice*. Se as coordenadas dos pontos de  $S$  são inteiros maiores ou iguais a zero e menores ou iguais a  $m$ , descreva um algoritmo para calcular o fecho convexo de  $S$  cuja complexidade é  $\mathcal{O}(n + m)$ . A entrada do algoritmo consiste dos  $n$  pontos de  $S$  e do inteiro  $m$ . A saída do algoritmo deve ser uma sequência com os vértices do fecho de  $S$  em ordem anti-horária. *Existe uma estratégia bastante utilizada em algoritmos para problemas geométricos que consiste no uso de uma **linha de varredura** [30]; isto é, uma linha vertical que é movida horizontalmente enquanto algum processamento é executado pelo algoritmo. Esta estratégia*





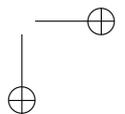
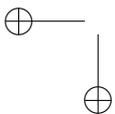
e uma estrutura de dados adequada podem lhe ser úteis neste problema.

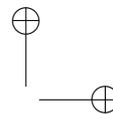
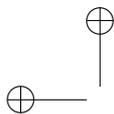
- 3.26 Seja  $\mathcal{P}$  um polígono convexo em  $\mathbb{E}^2$ . Escreva um algoritmo para encontrar o menor retângulo que contém  $\mathcal{P}$ . Note que o menor retângulo não necessariamente possui arestas verticais e horizontais (isto é, não o confunda com a tradicional *bounding box* frequentemente utilizada em aplicações gráficas). A entrada do seu algoritmo é uma sequência com os vértices  $\mathcal{P}$ , digamos  $v_0, \dots, v_{n-1}$ , tal que  $[v_i, v_{(i+1) \bmod n}]$  é uma aresta de  $\mathcal{P}$ , para todo  $i \in \{0, \dots, n-1\}$ , e  $n$  é o número de vértices de  $\mathcal{P}$ . Você pode assumir que  $v_0, \dots, v_{n-1}$  é uma ordenação anti-horária dos vértices de  $\mathcal{P}$ . A saída são as coordenadas dos quatro vértices do menor retângulo que contém  $\mathcal{P}$ . Há duas dicas interessantes para você desenvolver um algoritmo com complexidade  $\mathcal{O}(n)$ . A primeira é que o retângulo sempre possui um lado colinear a uma aresta de  $\mathcal{P}$  (ver Problema 3.5). A segunda é que o problema pode ser resolvido com a mesma estratégia descrita para o algoritmo de tempo linear para encontrar os pares de vértices antípodas de um polígono convexo (ver Problema 3.24). No entanto, desta vez, utilize quatro retas de suporte. Inicialmente, utilize duas retas horizontais e duas retas verticais. Em seguida, rotacione, sucessivamente, essas quatro retas de uma só vez por um determinado ângulo. Qual?

- 3.27 Um polígono  $\mathcal{P}$  em  $\mathbb{E}^2$  é dito *ortogonal* se, e somente se, todos os ângulos internos de  $\mathcal{P}$  medem  $k \cdot (\pi/2)$ , para qualquer  $k \in \mathbb{Z}$ . Informalmente, isto significa que o polígono pode ser rotacionado de tal forma que todas as suas arestas se tornem verticais ou horizontais com relação ao referencial afim ( $O = (0, 0), \mathbf{e}_1, \mathbf{e}_2$ ) em  $\mathbb{E}^2$ .

A Figura 3.11 ilustra um polígono ortogonal.

Um polígono ortogonal  $\mathcal{P}$  (cujas arestas são verticais ou horizontais apenas) é dito *ortogonalmente convexo* se, e somente se, a interseção de  $\mathcal{P}$  com qualquer reta horizontal ou vertical é vazia ou um único segmento de reta. Então, desenvolva um algoritmo que recebe como entrada um polígono ortogonal e decide se este polígono é ortogonalmente convexo. Analise a





complexidade do seu algoritmo e, se puder, forneça uma prova de corretude. Uma boa estratégia para desenvolver um algoritmo cuja corretude seja fácil de provar é pensar em uma caracterização de *ortogonalmente convexo* que ajude a criar um teste para saber se o polígono é ou não ortogonalmente convexo. Por exemplo, se o polígono for ortogonalmente convexo, o que deveria ocorrer quando passamos de uma aresta para outra adjacente em um percurso (anti-)horário pela fronteira do polígono?

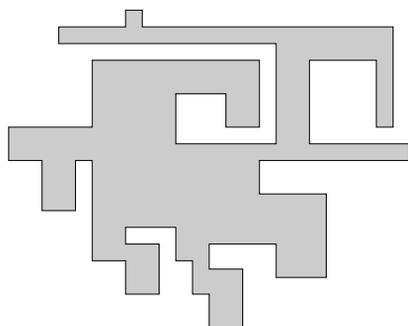
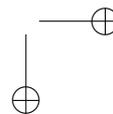
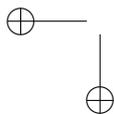


Figura 3.11: Um polígono ortogonal em  $\mathbb{E}^2$ .

3.28 O *fecho convexo ortogonal*,  $FCO(\mathcal{P})$ , de um polígono ortogonal  $\mathcal{P}$  é definido como sendo o menor polígono ortogonalmente convexo que contém  $\mathcal{P}$ . A Figura 3.12 mostra o fecho convexo ortogonal do polígono ortogonal da Figura 3.11. Note que se  $\mathcal{P}$  já for ortogonalmente convexo, então temos que  $FCO(\mathcal{P}) = \mathcal{P}$ . Caso contrário, temos que  $\mathcal{P}$  é um subconjunto próprio de  $FCO(\mathcal{P})$ .

Desenvolva um algoritmo para calcular o fecho convexo ortogonal,  $FCO(\mathcal{P})$ , de um dado polígono ortogonal,  $\mathcal{P}$ . A entrada do algoritmo é uma sequência com os vértices de  $\mathcal{P}$ , digamos  $v_0, \dots, v_{n-1}$ , tal que  $[v_i, v_{(i+1) \bmod n}]$  é uma aresta de  $\mathcal{P}$ , para todo  $i \in \{0, \dots, n-1\}$ , onde  $n$  é o número de vértices de  $\mathcal{P}$ . Você pode assumir que a sequência  $v_0, \dots, v_{n-1}$  é uma ordenação anti-horária dos vértices de  $\mathcal{P}$ . A saída do seu algo-



ritmo deve ser uma sequência ordenada, digamos  $u_0, \dots, u_{m-1}$ , dos vértices de  $FCO(\mathcal{P})$  tal que  $[u_i, u_{(i+1) \bmod m}]$  é uma aresta de  $FCO(\mathcal{P})$ , para todo  $i \in \{0, \dots, m-1\}$ , onde  $m$  é o número de vértices de  $FCO(\mathcal{P})$ . *O seu algoritmo não deve assumir que três vértices consecutivos de  $\mathcal{P}$  não estão em uma mesma reta vertical ou horizontal.* Qual é a complexidade de seu algoritmo?

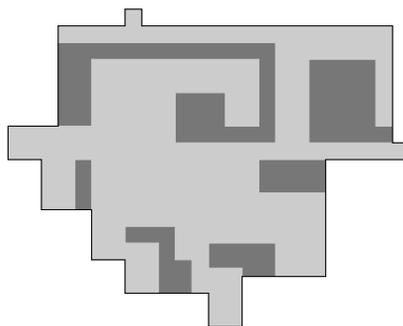


Figura 3.12: Fecho convexo ortogonal do polígono da Figura 3.11.

3.29 O Teorema de Carathéodory (Teorema 3.4) sugere naturalmente um algoritmo para calcular os pontos extremos do fecho convexo,  $FC(S)$ , de um conjunto  $S$  de  $n$  pontos de  $\mathbb{E}^2$ . A ideia é determinar, para cada tripla de pontos não colineares de  $S$ , se os demais pontos pertencem ao triângulo com vértices nos pontos da tripla. Se um ponto  $p$ , que não é um ponto da tripla, pertence ao triângulo com vértices nos pontos da tripla, então  $p$  não pode ser um ponto extremo de  $FC(S)$ . Por quê? Com base nisso, escreva o pseudocódigo do algoritmo e analise sua complexidade.

3.30 O Teorema 3.5 também sugere, de forma natural, um outro algoritmo para calcular os pontos extremos e as arestas do fecho convexo,  $FC(S)$ , de um conjunto  $S$  de  $n$  pontos de  $\mathbb{E}^2$ . A ideia é determinar, para cada par de pontos,  $p$  e  $q$ , de  $S$  se os demais pontos de  $S$  estão todos à esquerda ou todos à direita da reta por  $p$  e  $q$  orientada de  $p$  para  $q$ . Se sim, podemos concluir que

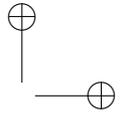
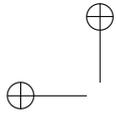
$p$  e  $q$  são pontos extremos de  $FC(S)$  e que  $\overline{pq}$  é uma aresta de  $FC(S)$ . Por quê? Caso contrário, podemos concluir que  $\overline{pq}$  não é uma aresta de  $FC(S)$  e que um dos dois pontos,  $p$  ou  $q$ , não é ponto extremo de  $FC(S)$ . Por quê? Com base nessas observações, escreva o pseudocódigo do algoritmo e analise sua complexidade.

- 3.31 Desenvolva um algoritmo para determinar se um dado ponto  $p$  de um dado conjunto  $S$  de  $n$  pontos em  $\mathbb{E}^2$  é um ponto extremo de  $FC(S)$ . É possível fornecer um algoritmo cuja complexidade é  $\mathcal{O}(n)$ .

### 3.7 Notas bibliográficas

A Seção 3.2 e a Seção 3.3 foram inteiramente baseadas no Capítulo 3 do livro [52]. O material da Seção 3.4 pode ser encontrado em diversos textos clássicos e introdutórios sobre Geometria Computacional, tais como [105, 32, 97, 30, 34] (para citar uns poucos). Em particular, encorajamos o leitor menos familiarizado com “análise de algoritmos” a ler uma breve introdução ao assunto no Capítulo 1 do livro escrito por Luiz Henrique de Figueiredo e Paulo César Pinto Carvalho para o 16º Colóquio Brasileiro de Matemática (CBM) [32]. Uma das razões para recomendarmos esta introdução é que ela foi escrita no contexto de problemas geométricos. O algoritmo de Graham foi originalmente descrito em um artigo de 1972 [56]. Os problemas 3.2-3.5 foram retirados do livro [34], enquanto os problemas 3.6-3.10 foram retirados do livro [52]. Um estudo detalhado do Problema do Final Feliz (Problema 3.17) pode ser encontrado em [90]. Os problemas 3.22-3.24 foram elaborados a partir do Capítulo 4 do livro de Preparata e Shamos [105]. Finalmente, os problemas 3.27 e 3.28 foram retirados de [97]. Uma rica discussão sobre o Problema 3.26 e problemas relacionados pode ser encontrada na dissertação de mestrado de Hormoz Pirzadeh [104] e no sítio mantido por ele na Universidade de *McGill*<sup>1</sup>.

<sup>1</sup><http://cgm.cs.mcgill.ca/~orm/rotcal.html> (último acesso: 07/03/2015)

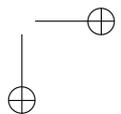
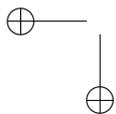


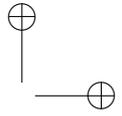
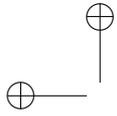
# Capítulo 4

## Triangulações

### 4.1 Considerações iniciais

Este capítulo é dedicado à definição, estudo de propriedades e construção da *triangulação de Delaunay*. Esta triangulação é um dos objetos de estudo mais importantes deste livro, pois as malhas geradas pelo algoritmo de geração de malhas que descrevemos no Capítulo 5 são, na verdade, triangulações de Delaunay acompanhadas de propriedades adicionais relacionadas à qualidade das respectivas malhas. A Seção 4.2 fornece uma definição de triangulação de um subconjunto finito e não vazio de pontos de  $\mathbb{E}^d$ . A Seção 4.3 introduz o mapa de elevação (do inglês, *lifting map*) ao parabolóide, que é utilizado para provar, na Seção 4.4, a existência da triangulação de Delaunay de um subconjunto finito e não vazio de pontos de  $\mathbb{E}^2$ . A Seção 4.5 introduz e demonstra uma série de propriedades da triangulação de Delaunay. A Seção 4.6 mostra um algoritmo eficiente, incremental e *aleatorizado* para calcular a triangulação de Delaunay. Este algoritmo é a base do algoritmo de geração de malhas apresentado no Capítulo 5. A complexidade do tempo de execução *esperado* do algoritmo também é analisada. A Seção 4.7 deriva uma cota inferior para o problema de calcular uma triangulação de Delaunay de subconjunto finito e não vazio de pontos de  $\mathbb{E}^2$ . A Seção 4.9 fornece uma vasta gama de exercícios para o leitor acerca do conteúdo do capítulo. A Seção 4.10 encerra o capítulo com comentários sobre as





principais referências bibliográficas que adotamos para a escrita dele.

## 4.2 O que é triangulação?

Como visto no Capítulo 2, o maior número de pontos de qualquer conjunto afimemente independente de pontos de  $\mathbb{E}^d$  é  $d + 1$ . Seja  $P = \{p_0, \dots, p_k\}$  um conjunto afimemente independente de pontos de  $\mathbb{E}^d$ , com  $k \in \mathbb{Z}$  e  $0 \leq k \leq d$ . Definimos o *simplexo*,  $\sigma$ , gerado pelos pontos  $p_0, \dots, p_k$  como sendo o fecho convexo,  $FC(P)$ , de  $P$  e denotamos  $\sigma$  por  $[p_0, \dots, p_k]$ . Os pontos  $p_0, \dots, p_k$  são os *vértices* de  $\sigma$ . A *dimensão*,  $\dim(\sigma)$ , de  $\sigma$  é  $k$  e  $\sigma$  é denominado um  $k$ -simplexo. Em  $\mathbb{E}^d$ , há somente simplexos de dimensão  $0, 1, \dots, d$ . Um 0-simplexo é um ponto, um 1-simplexo é um segmento de reta, um 2-simplexo é um triângulo, um 3-simplexo é um tetraedro e assim por diante (ver Figura 4.1).

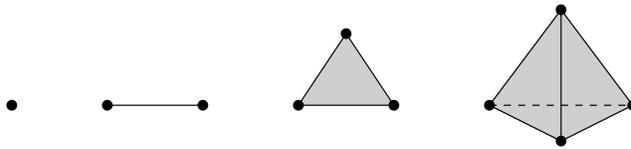


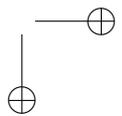
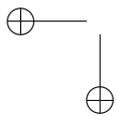
Figura 4.1: Exemplo de 0-, 1-, 2- e 3-simplexos.

O fecho convexo de qualquer subconjunto (próprio) não vazio do conjunto de vértices de um simplexo  $\sigma$  também é um simplexo (ver Problema 4.1). Este simplexo é denominado uma *face* (própria) de  $\sigma$ .

**Definição 4.1.** Um *complexo simplicial*,  $\mathcal{K}$ , em  $\mathbb{E}^d$  é um conjunto não vazio e finito de simplexos em  $\mathbb{E}^d$  que goza das propriedades a-baixo:

- (1) se  $\sigma \in \mathcal{K}$  e  $\tau \preceq \sigma$  então  $\tau \in \mathcal{K}$  e
- (2) se  $\sigma \cap \tau \neq \emptyset$  então  $\sigma \cap \tau \preceq \sigma$  e  $\sigma \cap \tau \preceq \tau$ , para todo  $\sigma, \tau \in \mathcal{K}$ ,

em que  $a \preceq b$  denota “ $a$  é uma face (não necessariamente própria) de  $b$ ”. A *dimensão*,  $\dim(\mathcal{K})$ , de  $\mathcal{K}$  é o maior valor entre as dimensões de



*todos os simplexes de  $\mathcal{K}$ . Um complexo simplicial de dimensão  $d$  (ou  $d$ -dimensional) é chamado, simplesmente, de  $d$ -complexo simplicial.*

A Figura 4.2 mostra três conjuntos de simplexes em  $\mathbb{E}^2$ . O conjunto mais à esquerda viola a propriedade (1) da Definição 4.1, pois um vértice e uma aresta de um dos triângulos não faz parte do conjunto. O conjunto do meio viola a propriedade (2) da Definição 4.1, pois o conjunto interseção de dois triângulos não é formado por uma ou mais faces comuns aos dois. Já o conjunto mais à direita goza das propriedades (1) e (2) e, portanto, é um complexo simplicial em  $\mathbb{E}^2$ .

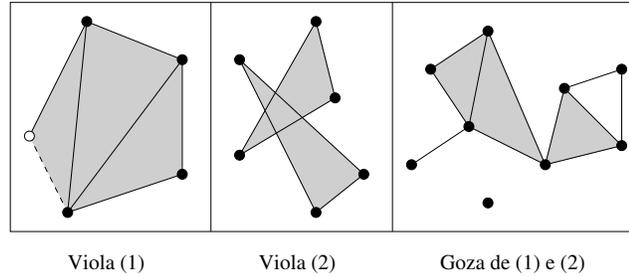
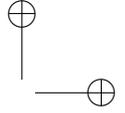
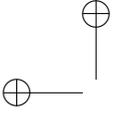


Figura 4.2: Apenas o conjunto de simplexes à direita é um complexo simplicial em  $\mathbb{E}^2$ . Os demais violam as propriedades (1) ou (2) da Definição 4.1.

É importante observar que um complexo simplicial é um objeto “discreto”, ou seja, um conjunto finito de elementos (pontos, segmentos de reta, triângulos, tetraedros, e assim por diante), enquanto um simplexo é um objeto “contínuo”, isto é, um conjunto convexo em  $\mathbb{E}^d$ . O conjunto (de pontos) consistindo da união de todos os simplexes de um complexo simplicial,  $\mathcal{K}$ , é denominado de *espaço subjacente* de  $\mathcal{K}$  e denotado por  $|\mathcal{K}|$ . A Figura 4.3 ilustra o espaço subjacente do único complexo simplicial dentre os três conjuntos de simplexes da Figura 4.2.

Uma triangulação de um conjunto de pontos é um complexo simplicial:

**Definição 4.2.** *Seja  $P$  um conjunto não vazio e finito de pontos em  $\mathbb{E}^d$ . Uma triangulação de  $P$  é um complexo simplicial, denotado*



por  $\mathcal{T}(P)$ , tal que todos os vértices de  $\mathcal{T}(P)$  estão em  $P$  e o espaço subjacente,  $|\mathcal{T}(P)|$ , de  $\mathcal{T}(P)$  é exatamente o fecho convexo,  $FC(P)$ , de  $P$ .

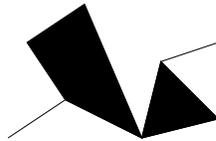
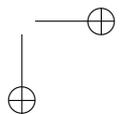
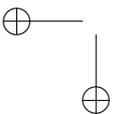


Figura 4.3: Espaço subjacente do complexo simplicial da Figura 4.2.

A Figura 4.4 ilustra duas triangulações de um mesmo conjunto de cinco pontos de  $\mathbb{E}^2$ . Observe que a Definição 4.2 não assume que o fecho afim de  $P$  possui dimensão  $d$ , em que o fecho afim de  $P$  é o subespaço afim que consiste de todos os pontos definidos por combinações afins (e, portanto, não apenas as convexas!) dos pontos de  $P$ . Isto implica que  $\mathcal{T}(P)$  pode não conter nenhum  $d$ -simplexo. A Figura 4.5 ilustra uma triangulação em  $\mathbb{E}^2$  que não possui triângulos. Observe também que nem todos os pontos de  $P$  necessitam ser vértices de  $\mathcal{T}(P)$ , exceto aqueles que são pontos extremos de  $FC(P)$ . Sempre que todos os pontos de  $P$  são vértices de  $\mathcal{T}(P)$  dizemos que  $\mathcal{T}(P)$  é uma *triangulação cheia*. Este é o tipo de triangulação que estudaremos neste livro. Portanto, de agora em diante, omitiremos o termo “cheia” ao nos referirmos a qualquer triangulação de pontos, pois assumimos que todas elas são triangulações cheias.

Dado um conjunto  $P$  finito e não vazio de pontos de  $\mathbb{E}^d$ , é natural nos perguntarmos se  $P$  admite uma triangulação, isto é, se existe uma triangulação,  $\mathcal{T}(P)$ , do conjunto de pontos,  $P$ . A Seção 4.4 fornece uma resposta afirmativa para o caso em que  $d = 2$ , já que provaremos a existência de um tipo especial de triangulação de um subconjunto qualquer de pontos de  $\mathbb{E}^2$ : a triangulação de Delaunay. Para tal, necessitamos definir uma função que mapeia pontos em  $\mathbb{E}^2$  para um parabolóide em  $\mathbb{E}^3$ . Esta função é o objeto de estudo da Seção 4.3 que veremos a seguir.



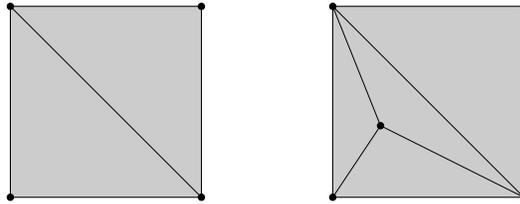
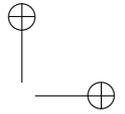


Figura 4.4: Duas triangulações do mesmo conjunto de pontos em  $\mathbb{E}^2$ .

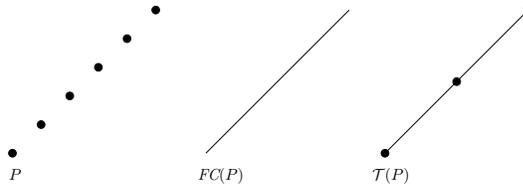


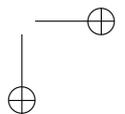
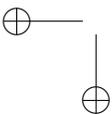
Figura 4.5: Uma triangulação em  $\mathbb{E}^2$  que não possui triângulo.

### 4.3 O mapa de elevação

De agora em diante, vamos assumir que as coordenadas  $(x, y)$  de qualquer ponto em  $\mathbb{E}^2$  (resp.  $\mathbb{E}^3$ ) são dadas com relação a algum referencial afim,  $(O, \mathbf{i}, \mathbf{j})$  (resp.  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ ), em  $\mathbb{E}^2$  (resp.  $\mathbb{E}^3$ ) tal que os vetores  $\mathbf{i}$  e  $\mathbf{j}$  (resp.  $\mathbf{i}, \mathbf{j}$  e  $\mathbf{k}$ ) são (dois a dois) ortogonais<sup>1</sup>. Seja  $\omega : \mathbb{E}^2 \rightarrow \mathbb{R}$  a função definida por  $\omega(a) = \|\vec{Oa}\|^2 = x^2 + y^2$ , para todo ponto  $a$  em  $\mathbb{E}^2$ , em que  $(x, y)$  são as coordenadas de  $a$ . Note que  $\omega$  pode ser vista como uma função altura que eleva o ponto  $a = (x, y)$  em  $\mathbb{E}^2$  ao parabolóide em  $\mathbb{E}^3$  de equação  $z = x^2 + y^2$ . Em outras palavras, o ponto de coordenadas  $(x, y, \omega(x, y)) = (x, y, x^2 + y^2)$  pertence ao gráfico de  $\omega$ , que é um parabolóide. A Figura 4.6 ilustra a definição de  $\omega$ .

A seguinte proposição estabelece uma importante propriedade da função  $\omega$ . Refira-se à Figura 4.7 durante a leitura da prova da pro-

<sup>1</sup>Isto é,  $(O, \mathbf{i}, \mathbf{j})$  (resp.  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ ) define um sistema de coordenadas cartesianas.



posição.

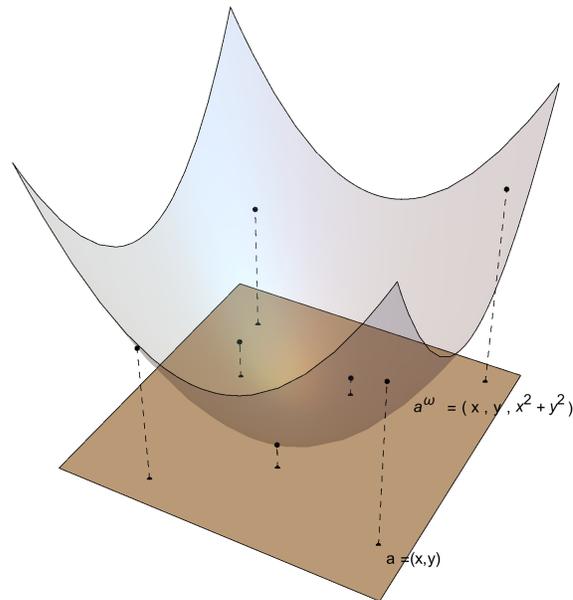


Figura 4.6:  $a = (x, y) \in \mathbb{E}^2$  é levado por  $\omega$  em  $\omega(a) = (x, y, x^2 + y^2) \in \mathbb{E}^3$ .

**Proposição 4.1.** *Seja  $S$  o parabolóide em  $\mathbb{E}^3$  definido pela equação  $z = x^2 + y^2$ . Seja  $H \subset \mathbb{E}^3$  um plano afim não “vertical”, isto é, um plano que não é paralelo ao vetor  $\mathbf{k}$ . Seja  $C$  a projeção ortogonal do conjunto interseção  $H \cap S$  em  $\mathbb{E}^2$ :  $C = \{(x, y) \in \mathbb{E}^2 \mid (x, y, z) \in H \cap S\}$ . Então, o conjunto  $C$  é vazio, unitário ou uma circunferência em  $\mathbb{E}^2$ .*

*Demonstração.* Pelo que vimos na Seção 2.7, o plano  $H$  é dado por uma equação da forma  $\alpha x + \beta y + \gamma z - \mu = 0$ , em que  $\alpha, \beta, \gamma$  e  $\mu$  são constantes reais. Como  $H$  não é “vertical”, a constante  $\gamma$  não pode

ser zero e, portanto, podemos reescrever a equação de  $H$  como segue:

$$z = \mu - \frac{\alpha}{\gamma}x - \frac{\beta}{\gamma}y.$$

Consequentemente, um ponto em  $\mathbb{E}^3$  de coordenadas  $(x, y, z)$  pertence a  $H \cap S$  se, e somente se, as coordenadas  $x$  e  $y$  satisfazem a seguinte equação:

$$x^2 + y^2 = \mu - \frac{\alpha}{\gamma}x - \frac{\beta}{\gamma}y,$$

que é equivalente a

$$\left(x + \frac{\alpha}{2\gamma}\right)^2 + \left(y + \frac{\beta}{2\gamma}\right)^2 = \left(\frac{\alpha}{2\gamma}\right)^2 + \left(\frac{\beta}{2\gamma}\right)^2 + \mu.$$

A equação acima não envolve a coordenada  $z$  e, portanto, ela define um conjunto,  $C$ , em  $\mathbb{E}^2$  que corresponde à projeção ortogonal de  $H \cap S$  em  $\mathbb{E}^2$ . Além disso, a equação não possui solução se o lado direito da igualdade for negativo, possui uma única solução se o lado direito da igualdade for igual a zero e define uma circunferência em  $\mathbb{E}^2$  se o lado direito da igualdade for positivo. Consequentemente, o conjunto  $C$  é vazio, unitário ou uma circunferência em  $\mathbb{E}^2$  e, portanto, a proposição é válida.  $\square$

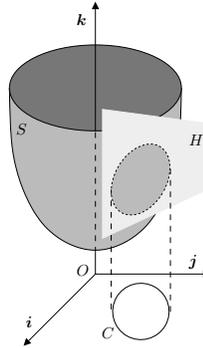
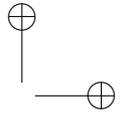
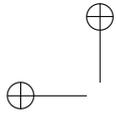


Figura 4.7: Interseção entre um parabolóide e um plano afim.



Seja  $P$  um conjunto não vazio e finito de pontos de  $\mathbb{E}^2$  e seja  $P^\omega$  o conjunto de pontos em  $\mathbb{E}^3$  obtido através da elevação dos pontos de  $P$  ao parabolóide de equação  $z = x^2 + y^2$ ; isto é, o conjunto de pontos

$$P^\omega = \{(x, y, w(x, y)) \in \mathbb{E}^3 \mid (x, y) \in \mathbb{E}^2\} .$$

Denote por  $\mathcal{P}$  o fecho convexo,  $FC(P^\omega)$ , de  $P^\omega$ . Sabemos, do Capítulo 3, que o conjunto  $\mathcal{P}$  é convexo. Além disso, o fato de  $\mathcal{P}$  ser o fecho convexo de um conjunto *finito* de pontos em  $\mathbb{E}^3$  faz dele um *politopo* em  $\mathbb{E}^3$ , o que implica que  $\mathcal{P}$  é dado por um conjunto finito de formas afins.

Mais especificamente, um subconjunto  $\mathcal{R}$  de  $\mathbb{E}^d$  é um politopo em  $\mathbb{E}^d$  se, e somente se, ele é limitado e existe uma lista  $(f_i)_{i=1}^n$  de formas afins de  $\mathbb{E}^d$  em  $\mathbb{R}$  tal que um ponto  $x$  de  $\mathbb{E}^d$  está em  $\mathcal{P}$  se, e somente se,

$$f_i(x) \leq 0 ,$$

para todo  $i = 1, \dots, n$ . Fixado um referencial afim  $(O, \mathbf{i}_1, \dots, \mathbf{i}_d)$  em  $\mathbb{E}^d$ , a forma  $f_i$  pode ser escrita como  $f_i(\mathbf{x}) = \mathbf{A}_i^T \mathbf{x} - b_i$  tal que  $\mathbf{A}_i, \mathbf{x} \in \mathbb{R}^d$  e  $b_i \in \mathbb{R}$ , com  $\mathbf{x} = \vec{Ox}$ . Do Capítulo 2, sabemos que o conjunto de pontos  $\{x \in \mathbb{E}^d \mid f_i(x) = 0\}$  define um hiperplano afim em  $\mathbb{E}^d$  e, do Capítulo 3, sabemos que o conjunto  $H_-(f_i) = \{x \in \mathbb{E}^d \mid f_i(x) \leq 0\}$  define um dos dois semiespaços fechados associados a  $f_i$ . Portanto, temos

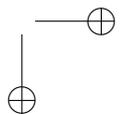
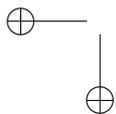
$$\mathcal{R} = \bigcap_{i=1}^n H_-(f_i) ,$$

o que nada mais é do que uma outra forma de escrever o enunciado do Teorema 3.5. Na literatura de Teoria de Poliedros [47] é comum definir  $\mathcal{R}$  por uma única função afim,  $f : \mathbb{E}^d \rightarrow \mathbb{E}^n$ , tal que  $f$  é dada por

$$f(x) = (f_1(x), \dots, f_n(x)) ,$$

para todo  $x \in \mathbb{E}^d$ . Em notação matricial, temos  $f(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{b}$ , para todo  $\mathbf{x} \in \mathbb{R}^d$ , em que  $\mathbf{M} \in \mathbb{R}^{n \times d}$  é a matriz cuja  $i$ -ésima linha é  $\mathbf{A}_i^T$  e  $\mathbf{b} \in \mathbb{R}^n$  é o vetor cuja  $i$ -ésima componente é  $b_i$ , para todo  $i = 1, \dots, n$ . Com um *abuso de notação*, é comum também definirmos  $\mathcal{R}$  por

$$\mathcal{R} = \{x \in \mathbb{E}^d \mid f(x) \leq O\} .$$



A Figura 4.8 ilustra o politopo  $\mathcal{P}$  correspondente a  $FC(P^\omega)$ .

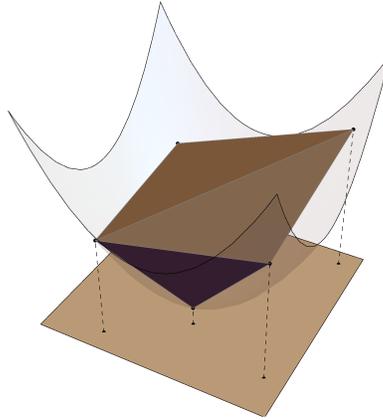


Figura 4.8: O fecho convexo,  $FC(P^\omega)$ , de  $P^\omega$  é um politopo em  $\mathbb{E}^3$ .

Seja  $\mathcal{R}$  um politopo em  $\mathbb{E}^d$ . A *dimensão*,  $\dim(\mathcal{R})$ , de  $\mathcal{R}$  é a dimensão,  $\dim(FA(\mathcal{R}))$ , do subespaço afim correspondente ao *fecho afim*,  $FA(\mathcal{R})$ , de  $\mathcal{R}$  (ver Problema 2.14 e o Problema 3.8). Se  $\dim(\mathcal{R}) = d$ , então dizemos que  $\mathcal{R}$  é um  $d$ -politopo. Um 0-politopo é um ponto, um 1-politopo é um segmento de reta e um 2-politopo é um polígono convexo com  $n$  vértices. Todo simplexo é um politopo e um  $d$ -simplexo pode ser visto como o  $d$ -politopo *mais simples*. Uma *face* de  $\mathcal{R}$  é qualquer conjunto da forma  $\mathcal{R} \cap \{x \in \mathbb{E}^d \mid g(x) = 0\} = \mathcal{R} \cap Nuc(g)$ , em que  $g : \mathbb{E}^d \rightarrow \mathbb{R}$  é uma forma afim tal que  $g(y) \leq 0$ , para todo  $y \in \mathcal{R}$ . A *dimensão*,  $\dim(F)$ , de uma face  $F$  de  $\mathcal{R}$  é a dimensão,  $\dim(FA(F))$ , do fecho afim,  $FA(F)$ , de  $F$ . As faces de  $\mathcal{R}$  de dimensão 0, 1 e  $\dim(\mathcal{R}) - 1$  são ditas *vértices*, *arestas* e *facetas* de  $\mathcal{R}$ , respectivamente. Denotamos por  $vert(\mathcal{R})$  o conjunto de vértices de  $\mathcal{R}$ .

No que segue, apresentaremos alguns fatos simples, mas cruciais, sobre faces de politopos. Para provar um deles, faremos uso do Lema de Farkas (Versão II), que é um resultado clássico da Programação Linear:

**Teorema 4.2** (Lema de Farkas – Versão II). *Dados qualquer matriz  $\mathbf{A} \in \mathbb{R}^{m \times n}$  e qualquer vetor  $\mathbf{b} \in \mathbb{R}^n$ , apenas um dos seguintes itens é válido:*

(a) *O sistema linear*

$$\mathbf{Ax} = \mathbf{b}$$

*tem uma solução  $\mathbf{x} = (x_1, \dots, x_n)^T$  tal que  $x_i \geq 0$ ,  $\forall i = 1, \dots, n$ .*

(b) *Há um vetor  $\mathbf{c} \in \mathbb{R}^n$  tal que  $\mathbf{c}^T \cdot \mathbf{b} < 0$  e  $\mathbf{c}^T \cdot \mathbf{A}_j \geq 0$ , para todo  $j = 1, \dots, n$ , em que  $\mathbf{A}_j$  corresponde à  $j$ -ésima coluna de  $\mathbf{A}$ .*

*Demonstração.* Consulte o Capítulo 7 do livro [52] para uma prova.  $\square$

**Proposição 4.3.** *Seja  $\mathcal{R}$  um polítopo em  $\mathbb{E}^d$ . Então, o polítopo  $\mathcal{R}$  é o fecho convexo,  $FC(\text{vert}(\mathcal{R}))$ , do seu conjunto de vértices,  $\text{vert}(\mathcal{R})$ . Além disso, se  $\mathcal{R}$  pode ser definido como o fecho convexo,  $FC(P)$ , de um conjunto  $P$  de pontos em  $\mathbb{E}^d$ , então os vértices de  $\mathcal{R}$  estão em  $P$ ; isto é,*

$$\text{vert}(\mathcal{R}) \subseteq P.$$

*Demonstração.* Seja  $Q$  um subconjunto não vazio e finito de pontos de  $\mathbb{E}^d$  tal que  $\mathcal{R} = FC(Q)$ . Considere um ponto  $v$  qualquer de  $Q$ . Afirmamos que se  $v$  não pode ser escrito como uma combinação convexa dos pontos em  $Q' = Q - \{v\}$ , então  $v$  é um vértice de  $\mathcal{R}$ ; isto é,  $v \in \text{vert}(\mathcal{R})$ . De fato, seja  $(u_i)_{i=1}^m$  uma lista com os pontos em  $Q'$ . Se  $v$  não pode ser escrito como uma combinação convexa dos pontos em  $Q'$ , então não existem escalares  $\alpha_1, \dots, \alpha_m$  em  $\mathbb{R}$  tais que  $\alpha_i \geq 0$ , para todo  $i = 1, \dots, m$ ,  $\sum_{i=1}^m \alpha_i = 1$  e  $\sum_{i=1}^m \alpha_i \cdot u_i = v$ . Isto é, se fixarmos qualquer referencial afim  $(O, \mathbf{i}_1, \dots, \mathbf{i}_d)$  em  $\mathbb{E}^d$ , a referida condição sobre  $v$  implica que o sistema de equações lineares,

$$\mathbf{A}\alpha = \mathbf{b}, \tag{4.1}$$

com

$$\mathbf{A} = \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ u_{11} & \cdots & u_{i1} & \cdots & u_{m1} \\ u_{12} & \cdots & u_{i2} & \cdots & u_{m2} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{1d} & \cdots & u_{id} & \cdots & u_{md} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix},$$

não admite solução  $\alpha = (\alpha_1 \cdots \alpha_m)^T$ , com  $\alpha_i \geq 0$ , para todo  $i = 1, \dots, m$ , em que  $u_{ij}$  e  $v_j$  são as  $j$ -ésimas coordenadas de  $u_i$  e  $v$ , respectivamente, para todo  $j = 1, \dots, d$ . Logo, pelo Teorema 4.2, temos

$$\mathbf{c}^T \cdot \begin{bmatrix} 1 \\ u_{i1} \\ u_{i2} \\ \vdots \\ u_{id} \end{bmatrix} \geq 0 \quad \text{e} \quad \mathbf{c}^T \cdot \begin{bmatrix} 1 \\ v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix} < 0,$$

para algum  $\mathbf{c} = (c_0, c_1, \dots, c_d)^T \in \mathbb{R}^{d+1}$  e todo  $i = 1, \dots, m$ . Logo, se  $\mathbf{h} = (c_0, h_1, \dots, h_d)^T \in \mathbb{R}^{d+1}$ , com  $h_j = -c_j$ , para  $j = 1, \dots, d$ , então temos:

$$\mathbf{h}^T \cdot \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{id} \end{bmatrix} \leq c_0 \quad \text{e} \quad \mathbf{h}^T \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix} > c_0.$$

As desigualdades acima nos permitem concluir que existe uma forma afim,

$$f : \mathbb{E}^d \rightarrow \mathbb{R},$$

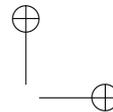
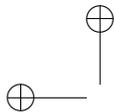
com

$$f(x) = h_1 x_1 + \cdots + h_d x_d - c_0$$

para todo ponto  $x = (x_1, \dots, x_d)$  em  $\mathbb{E}^d$ , tal que  $f(u_i) \leq c_0$ , para todo  $i = 1, \dots, m$ , e  $f(v) > c_0$ . Em outras palavras, há um hiperplano afim,  $f^{-1}(0)$ , que separa  $v$  dos pontos pertencentes a  $Q'$ . Além disso, se

$$\beta = \mathbf{h}^T \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix} \in \mathbb{R},$$

então a forma afim,  $g : \mathbb{E}^d \rightarrow \mathbb{R}$ , com  $g(x) = h_1 x_1 + \cdots + h_d x_d - \beta$ , para todo  $x = (x_1, \dots, x_d)$  em  $\mathbb{E}^d$ , é tal que  $g(v) = 0$  e  $g(u_i) \leq 0$ , para todo  $i = 1, \dots, m$ . Logo,  $v$  é um vértice do politopo  $\mathcal{R}$ . Isto nos diz que todos os pontos extremos de  $FC(Q)$ , que necessariamente pertencem



a  $Q$  (pela definição de ponto extremo), são vértices de  $\mathcal{R}$ . Logo, pelo Teorema 3.6, a afirmação (a) é verdadeira. Por outro lado, nenhum vértice  $w$  de  $\mathcal{R}$  pode, por definição, ser escrito como uma combinação convexa dos pontos em  $\mathcal{R} - \{w\}$ . Logo, todo vértice de  $\mathcal{R}$  é ponto extremo de  $FC(Q)$  e, portanto, nós temos que  $vert(\mathcal{R}) \subseteq Q$ , provando (b).  $\square$

**Proposição 4.4.** *Seja  $\mathcal{R}$  um politopo em  $\mathbb{E}^d$  e  $V = vert(\mathcal{R})$ . Seja  $F$  uma face qualquer de  $\mathcal{R}$ . Então, as seguintes afirmações são válidas:*

- (a)  $F$  é um politopo e  $vert(F) = F \cap V$ .
- (b) Toda interseção de faces de  $\mathcal{R}$  é uma face de  $\mathcal{R}$ .
- (c) As faces de  $F$  são exatamente as faces de  $\mathcal{R}$  contidas em  $F$ .
- (d)  $F = \mathcal{R} \cap FA(F)$ .

*Demonstração.* Por definição de face, sabemos que existe uma forma afim,  $f : \mathbb{E}^d \rightarrow \mathbb{R}$ , tal que  $F = \mathcal{R} \cap H$ , com  $H = Nuc(f)$ , e  $f(x) \leq 0$ , para todo  $x \in \mathcal{R}$ . Por definição, o politopo  $\mathcal{R}$  é o fecho convexo de um conjunto finito de pontos em  $\mathbb{E}^d$ . Pelo Teorema 3.5, o politopo  $\mathcal{R}$  é a interseção de semiespaços fechados contendo todos os seus pontos. Como

$$H = H_-(f) \cap H_+(f) \quad \text{e} \quad \mathcal{R} \subset H_-(f),$$

temos que

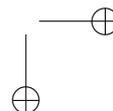
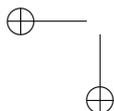
$$F = \mathcal{R} \cap H_+(f).$$

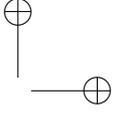
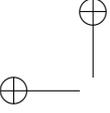
Logo,  $F$  também é uma interseção de semiespaços fechados contendo todos os seus pontos e, portanto, pelo Teorema 3.5,  $F$  é um politopo. Logo, a primeira parte da afirmação (a) está provada. Se  $y$  é um ponto em  $FA(F)$ , então existe uma lista,  $((p_i, \alpha_i))_{i=1}^n$ , de  $n$  pontos ponderados de  $F$  tal que  $y = \sum_{i=1}^n \alpha_i \cdot p_i$  e  $\sum_{i=1}^n \alpha_i = 1$ . Como  $f$  é afim,

$$F(y) = f\left(\sum_{i=1}^n \alpha_i \cdot p_i\right) = \sum_{i=1}^n \alpha_i \cdot f(p_i) = \sum_{i=1}^n \alpha_i \cdot 0 = 0,$$

pois  $p_i \in F$  implica que  $F(p_i) = 0$ , para todo  $i = 1, \dots, n$ . Como  $F(y) = 0$ , temos que  $y \in H$ . Logo, concluímos que  $F \subseteq FA(F) \subseteq H$ . Daí,

$$F = \mathcal{R} \cap H = \mathcal{R} \cap FA(F)$$





e, portanto, a afirmação (d) também é válida. Para provar a segunda parte de (a), note que  $(F \cap V) \subseteq \text{vert}(F)$ . Por outro lado, considere qualquer ponto  $y \in F$ . Então, existem escalares  $\beta_1, \dots, \beta_m$  tais que  $\beta_i \geq 0$ , para todo  $i = 1, \dots, m$ ,  $\sum_{i=1}^m \beta_i = 1$  e  $y = \sum_{i=1}^m \beta_i \cdot v_i$ , com  $m = |V|$  e  $v_i \in V$ . Como  $f$  é afim e o ponto  $y$  pertence a  $F$ , temos que

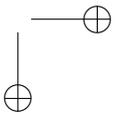
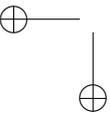
$$0 = f(y) = f\left(\sum_{i=1}^m \beta_i \cdot v_i\right) = \sum_{i=1}^m \beta_i \cdot f(v_i) \leq \sum_{i=1}^m \beta_i \cdot 0 = 0,$$

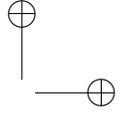
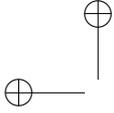
pois  $\beta_i \geq 0$  e  $f(v_i) \leq 0$ . Logo, podemos concluir que  $\beta_i \cdot f(v_i) = 0$ , para todo  $i = 1, \dots, m$ . Isto implica que  $\beta_i = 0$  para todo  $i \in \{1, \dots, m\}$  tal que  $v_i \notin (F \cap V)$ , o que por sua vez implica que  $y \in FC(F \cap V)$ . Logo,  $F = FC(F \cap V)$ . Daí, a Proposição 4.3 nos diz que  $\text{vert}(F) \subseteq (F \cap V)$  e, portanto, a segunda parte da afirmação (a) está provada. Para provar a afirmação (b), considere duas faces quaisquer de  $\mathcal{R}$ , digamos  $F$  e  $G$ , tais que  $F = \mathcal{R} \cap \text{Nuc}(f_F)$  e  $G = \mathcal{R} \cap \text{Nuc}(f_G)$ , em que  $f_F : \mathbb{E}^d \rightarrow \mathbb{R}$  e  $f_G : \mathbb{E}^d \rightarrow \mathbb{R}$  são duas formas afins tais que  $f_F(x) \leq 0$  e  $f_G(x) \leq 0$ , para todo ponto  $x \in \mathcal{R}$ . Seja  $g : \mathbb{E}^d \rightarrow \mathbb{R}$  tal que  $g(x) = f_F(x) + f_G(x)$ , para todo  $x \in \mathcal{R}$ . Observe que  $g(x) = 0$ , para todo  $x \in (F \cap G)$ , e  $g(x) \leq 0$ , para todo  $x \in \mathcal{R}$ . Logo, temos que

$$\mathcal{R} \cap \text{Nuc}(g) = F \cap G$$

e, portanto,  $F \cap G$  é uma face de  $\mathcal{R}$ . Para provar (c), observe que se  $G \subseteq F$  é uma face de  $\mathcal{R}$ , então  $G$  é uma face de  $F$  também. Para a recíproca, suponha que  $F = \mathcal{R} \cap \text{Nuc}(f_F)$  e  $G = F \cap \text{Nuc}(f_G)$ , em que  $f_F : \mathbb{E}^d \rightarrow \mathbb{R}$  e  $f_G : \mathbb{E}^d \rightarrow \mathbb{R}$  são duas formas afins tais que  $f_F(x) \leq 0$  e  $f_G(y) \leq 0$ , para todo ponto  $x \in \mathcal{R}$  e para todo ponto  $y \in F$ . Sejam  $W = \text{vert}(F)$  e  $V' = V - W$ . Nós podemos assumir que  $F \neq \mathcal{R}$  e, portanto,  $V' \neq \emptyset$ . Seja  $g : \mathbb{E}^d \rightarrow \mathbb{R}$  tal que  $g(x) = \lambda \cdot f_F(x) + f_G(x)$ , para todo  $x \in \mathbb{E}^d$ , em que  $\lambda$  é um número real arbitrário. Como  $G \subseteq F$ , temos que  $g(x) \leq 0$ , para todo  $x \in F$ . Além disso, a forma afim  $g$  define  $G$  como uma face de  $F$ , pois  $g(x) = 0$ , para todo  $x \in G$ . Agora, vamos escolher  $\lambda$  tal que

$$\lambda > -\frac{f_G(y)}{f_F(y)},$$





para todo  $y \in V'$ . Como consequência, para qualquer  $z \in V'$ , temos que

$$\begin{aligned} g(z) &= \lambda \cdot f_F(z) + f_G(z) \\ &\leq -\frac{f_G(z)}{f_F(z)} \cdot f_F(z) + f_G(z) \\ &= -f_G(z) + f_G(z) \\ &= 0. \end{aligned}$$

Daí,  $g(x) \leq 0$  para todo ponto  $x \in V$  e, portanto, para todo ponto  $x \in \mathcal{R}$ , o que implica que  $G = \mathcal{R} \cap Nuc(g)$  é uma face de  $\mathcal{R}$  também.  $\square$

Vamos voltar nossa atenção novamente para o politopo  $\mathcal{P}$  correspondente a  $FC(P^\omega)$ . Observe que a dimensão,  $dim(\mathcal{P})$ , de  $\mathcal{P}$  é igual a três se, e somente se,  $P^\omega$  possui um subconjunto afim independente com quatro pontos ou, de forma equivalente, se, e somente se, o conjunto  $P$  possui pelo menos quatro pontos não colineares. Assuma que este seja o caso para  $P$ . Então, as facetas do politopo  $\mathcal{P}$  possuem dimensão 2. O *envelope inferior* de  $\mathcal{P}$  é o conjunto,  $\mathcal{E}(\mathcal{P})$ , de faces de  $\mathcal{P}$  tal que  $F$  pertence a  $\mathcal{E}(\mathcal{P})$  se, e somente se, existe uma forma afim,

$$g_F : \mathbb{E}^3 \rightarrow \mathbb{R},$$

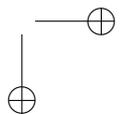
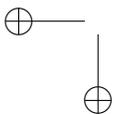
tal que

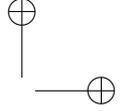
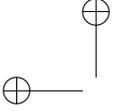
$$\begin{aligned} F &= \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g_F(x) = 0\}, \\ g_F(y) &\leq 0, \quad \forall y \in \mathcal{P} \end{aligned}$$

e

$$g_F(O + \alpha \cdot \mathbf{k}) > 0$$

para todo  $\alpha \in [\beta, -\infty)$  e para alguma constante real  $\beta$ , com  $\beta < 0$ . A primeira e a segunda condições nos dizem que  $F$  é uma face de  $\mathcal{P}$ , enquanto a terceira condição nos diz que  $F$  pode ser vista por um observador posicionado em qualquer ponto  $O + \alpha \cdot \mathbf{k}$  de  $\mathbb{E}^3$ , com  $\alpha \in [\beta, -\infty)$ . Na Seção 4.4, mostraremos que se nenhum subconjunto de quatro pontos de  $P$  reside em uma mesma circunferência, então a projeção de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  define uma triangulação de  $P$ . A nossa prova faz uso das Proposições 4.3 e 4.4 e da Proposição 4.5 enunciada abaixo:

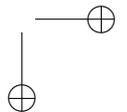
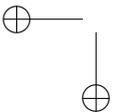




**Proposição 4.5.** *Seja  $P$  um conjunto não vazio e finito de pontos em  $\mathbb{E}^2$ . Seja  $Q \subseteq P$  qualquer subconjunto de  $P$  afimemente independente contendo três pontos de  $\mathbb{E}^2$ . Então, o conjunto  $Q^\omega$  corresponde ao conjunto de vértices de uma faceta do envelope inferior de  $\mathcal{P} = FC(P^\omega)$  se, e somente se, todos os pontos de  $Q$  residem em uma circunferência tal que todos os pontos em  $P - Q$  estão no exterior dela.*

*Demonstração.* Por hipótese, o subconjunto  $Q$  de  $P$  é afimemente independente e possui três pontos de  $\mathbb{E}^2$ . Isto nos diz que há, no máximo, uma única circunferência passando por todos os pontos de  $Q$ . Se tal circunferência existir, dizemos que os pontos de  $Q$  são cocirculares. Assuma que tal circunferência *não* existe. Então, pela Proposição 4.1, os pontos em  $Q^\omega$  não podem corresponder ao conjunto de vértices de uma faceta de  $\mathcal{P}$ . De fato, se os pontos de  $Q^\omega$  fossem vértices de uma faceta de  $\mathcal{P}$ , então eles residiriam na interseção do parabolóide,  $\Gamma = \{(x, y, z) \in \mathbb{E}^3 \mid z = \omega(x, y)\}$ , e o plano afim contendo a faceta. Mas, a projeção ortogonal desta interseção sobre  $\mathbb{E}^2$  é uma circunferência e, portanto, todos os pontos em  $Q$  residiriam em uma mesma circunferência: uma contradição. Reciprocamente, assuma que existe uma circunferência,  $C$ , contendo todos os pontos de  $Q$ . Então, há um plano afim,  $H = Nuc(f)$ , em que  $f : \mathbb{E}^3 \rightarrow \mathbb{R}$  é uma forma afim, contendo todos os pontos de  $Q^\omega$ . A interseção de  $H$  com  $\Gamma$  é a elipse  $C^\omega = \{(x, y, z) \in \mathbb{E}^3 \mid (x, y) \in C \text{ e } z = \omega(x, y)\}$ . A elipse  $C^\omega$  separa  $\Gamma$  em dois conjuntos conexos (ver Figura 4.9):  $\Gamma \cap int(H_+(f))$  e  $\Gamma \cap int(H_-(f))$ , em que  $int(V)$  denota o interior de um subconjunto  $V$  de  $\mathbb{E}^d$ .

Os pontos em  $\Gamma \cap int(H_+(f))$  são projetados ortogonalmente em  $\mathbb{E}^2$  no interior da circunferência  $C$ , enquanto os pontos do conjunto  $\Gamma \cap int(H_-(f))$  são projetados ortogonalmente em  $\mathbb{E}^2$  no exterior de  $C$ . Logo, se não há pontos de  $(P - Q)$  no interior de  $C$ , não haverá pontos de  $P^\omega$  em  $\Gamma \cap int(H_+(f))$ . Daí, os pontos de  $Q^\omega$  pertencem a uma faceta,  $F$ , do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$  tal que  $F$  não contém nenhum ponto em  $(P - Q)^\omega$ , pois  $(P - Q)$  é um subconjunto do exterior de  $C$ . Observe que todos os pontos em  $Q^\omega$  são pontos extremos de  $FC(Q^\omega)$ , pois eles pertencem a uma elipse, isto é,  $C^\omega$ . Mas, pela Proposição 4.4(a), toda faceta de  $\mathcal{P}$  é um politopo. Além disso, os únicos pontos de  $P^\omega$  em  $F$  são aqueles de  $Q^\omega$ . Logo, o fato



de todo ponto de  $Q^\omega$  ser ponto extremo de  $FC(Q^\omega)$  e a Proposição 4.3 implicam que  $vert(F) = Q^\omega$  e, portanto,  $F = FC(Q^\omega)$  é um polígono convexo.  $\square$

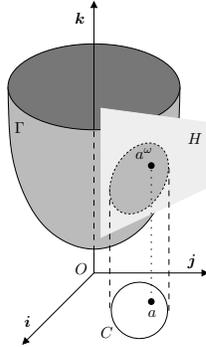


Figura 4.9: O ponto  $a \in \mathbb{E}^2$  é levado por  $\omega$  para  $\Gamma \cap int(H_+(f))$ .

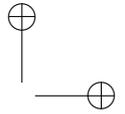
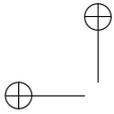
#### 4.4 A triangulação de Delaunay

Seja  $P$  um conjunto finito e não vazio de pontos em  $\mathbb{E}^2$ . Seja  $P^\omega$  o conjunto de pontos em  $\mathbb{E}^3$  correspondente à elevação dos pontos de  $P$  ao parabolóide definido pela função  $\omega$ ; isto é,  $P^\omega = \bigcup_{p \in P} p^\omega$ , em que

$$p^\omega = (x, y, \omega(x, y)), \quad \text{com } p = O + x \cdot \mathbf{i} + y \cdot \mathbf{j}.$$

Seja  $\mathcal{P}$  o fecho convexo,  $FC(P^\omega)$ , de  $P^\omega$ . Se a projeção ortogonal do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$  sobre  $\mathbb{E}^2$  for uma triangulação de  $P$ , então esta triangulação é denominada *triangulação de Delaunay de  $P$*  e é denotada por  $\mathcal{TD}(P)$ . No que segue, estudamos condições necessárias e suficientes para que a *projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  em  $\mathbb{E}^2$  seja uma triangulação de  $P$* , ou seja, para a existência da triangulação  $\mathcal{TD}(P)$ .

Mais especificamente, assumimos três condições sobre  $P$  e mostramos inicialmente que se essas condições forem satisfeitas, então

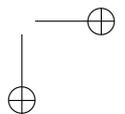
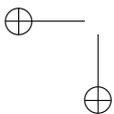


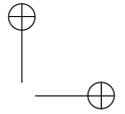
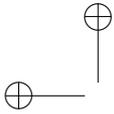
a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é um conjunto de triângulos, arestas e vértices satisfazendo a condição (1) da definição de complexo simplicial (Definição 4.1). Em seguida, mostramos que a condição (2) desta mesma definição também é satisfeita. Isto nos permitirá afirmar que a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é um complexo simplicial em  $\mathbb{E}^2$ . Depois, mostramos que os vértices deste complexo simplicial são exatamente os pontos de  $P$ , provando que a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é uma triangulação e, conseqüentemente, que a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  existe e está bem definida. Finalmente, estabelecemos a unicidade de  $\mathcal{TD}(P)$  e discutimos o que ocorre quando uma ou mais das condições que assumimos for violada.

**Observação 4.1.** *Assuma que o conjunto  $P$  satisfaz três condições: (1)  $P$  possui pelo menos quatro pontos, (2) nenhum subconjunto de quatro pontos de  $P$  reside em uma mesma circunferência e (3) há um subconjunto de  $P$  afimemente independente com três pontos de  $\mathbb{E}^2$ . É comum dizer que  $P$  está em posição geral quando ele satisfaz (2) e (3).*

As condições (1), (2) e (3) implicam a existência de um subconjunto  $Q^\omega$  de  $P^\omega$  afimemente independente em  $\mathbb{E}^3$  e tal que  $Q^\omega$  possui quatro pontos. De fato, a condição (3) nos diz que existe um subconjunto,  $B$ , de  $P$  afimemente independente e contendo três pontos de  $P$ . Defina  $Q = B \cup \{a\}$ , em que  $a$  é qualquer ponto em  $P - B$ . A existência de  $a$  é garantida pela condição (1). A condição (2) garante que os pontos de  $Q$  não estão sobre uma mesma circunferência em  $\mathbb{E}^2$ . A Proposição 4.5 garante que os três pontos em  $B^\omega$  residem em um mesmo plano,  $H$ , em  $\mathbb{E}^3$ . Por sua vez, a Proposição 4.1 nos diz que o ponto  $a^\omega$  não pertence a  $H$ . Caso contrário, os quatro pontos em  $Q$  pertenceriam a uma mesma circunferência em  $\mathbb{E}^2$ . Finalmente, o conjunto  $B^\omega$  também possui um subconjunto afimemente independente de três pontos de  $\mathbb{E}^3$ , já que os pontos de  $B^\omega$  pertencem a uma elipse em  $\mathbb{E}^3$ .

O fato de  $Q^\omega$  ser um conjunto afimemente independente de quatro pontos em  $\mathbb{E}^3$  implica que o fecho afim,  $FA(P^\omega)$ , de  $P^\omega$  é um subespaço de  $\mathbb{E}^3$  de dimensão 3. Logo, o politopo  $\mathcal{P}$ , que é o fecho convexo,  $FC(P^\omega)$ , de  $P^\omega$  possui dimensão 3 e suas facetas, dimensão 2. Além disso, a condição (2) e a Proposição 4.5 nos garantem que todas as facetas do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$  são triângulos. A





projeção ortogonal de qualquer um desses triângulos sobre  $\mathbb{E}^2$  é um triângulo. Caso contrário, a faceta seria paralela ao vetor  $\mathbf{k}$  (isto é, vertical), o que é um absurdo, pois nenhuma faceta de  $\mathcal{E}(\mathcal{P})$  é paralela a  $\mathbf{k}$ .

**Observação 4.2.** *As observações acima nos permitem concluir que a projeção ortogonal das facetas de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é um conjunto de triângulos.*

O próximo passo é mostrar que os triângulos projetados sobre  $\mathbb{E}^2$ , com os seus vértices e arestas, formam um complexo simplicial em  $\mathbb{E}^2$ .

Afirmamos que todos os pontos de  $P^\omega$  são vértices de  $\mathcal{P}$  e pertencem a  $\mathcal{E}(\mathcal{P})$ . De fato, para cada ponto  $a = (x_a, y_a)$  em  $P$ , considere o plano,  $H_a$ , tangente ao parabolóide  $\Gamma$  em  $a^\omega = (x_a, y_a, \omega(x_a, y_a))$ , em que  $\Gamma = \{(x, y, z) \in \mathbb{E}^3 \mid z = \omega(x, y)\}$ . O plano  $H_a$  pode ser definido por

$$H_a = g_a^{-1}(0),$$

em que  $g_a : \mathbb{E}^3 \rightarrow \mathbb{R}$  é a forma afim tal que, para todo ponto  $x$  de  $\mathbb{E}^3$  com coordenadas  $(x, y, z)$  em relação a  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ ,  $g_a(x, y, z)$  é igual a

$$g_a(x, y, z) = 2x_a x + 2y_a y - z - x_a^2 - y_a^2,$$

em que  $(x_a, y_a, z_a)$  são as coordenadas de  $a^\omega$  com relação a  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ . Por definição de plano tangente, a interseção de  $H_a$  com uma vizinhança de  $a^\omega$  em  $\Gamma$  contém apenas o ponto  $a^\omega$ . Logo, pela Proposição 4.1, a interseção  $H_a \cap \Gamma$  tem de ser igual a  $a^\omega$ . Conseqüentemente, todos os pontos de  $\Gamma - \{a^\omega\}$  estão acima ou abaixo de  $H_a$ . No entanto, como  $H_a$  intersecta a reta paralela a  $\mathbf{k}$  e que passa pelo ponto  $O$  em

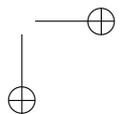
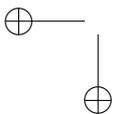
$$(0, 0, -x_a^2 - y_a^2),$$

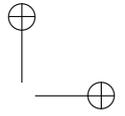
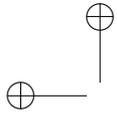
podemos concluir que  $g_a(x) \leq 0$ , para todo ponto  $x \in \mathcal{P}$ . Além disso,

$$g_a(O + \alpha \cdot \mathbf{k}) > 0,$$

para  $\alpha < -(x_a^2 + y_a^2)$ , o que nos permite concluir que  $a^\omega$  pertence a  $\mathcal{E}(\mathcal{P})$ .

Observe que a Proposição 4.3, juntamente com os fatos que  $\mathcal{P} = FC(P^\omega)$  e  $vert(\mathcal{P}) = P^\omega$ , implicam que *todos os vértices dos triângulos resultantes da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  em  $\mathbb{E}^2$  estão em  $P$ .*





Por outro lado, todo vértice de  $\mathcal{P}$  é resultante da interseção de três ou mais facetas de  $\mathcal{P}$ . Afirmamos que uma delas tem de pertencer a  $\mathcal{E}(\mathcal{P})$ .

Seja  $a$  qualquer ponto de  $P$  e considere a Figura 4.10. Sabemos que  $a^\omega$  é um vértice de  $\mathcal{P}$ . Queremos mostrar que existe uma faceta  $F$  de  $\mathcal{P}$  tal que  $F$  pertence a  $\mathcal{E}(\mathcal{P})$  e  $a^\omega$  é vértice de  $F$ . Seja  $b$  um vértice de  $P$  mais próximo de  $a$  e considere o círculo  $C_{ab}$  cuja circunferência passa por  $a$  e  $b$  e cujo centro é o ponto médio,  $M_{ab}$ , de  $\overline{ab}$ . Por escolha de  $b$ , não pode existir nenhum ponto de  $P$  no interior,  $\text{int}(C_{ab})$ , de  $C_{ab}$ .

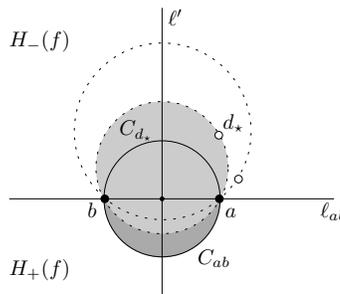
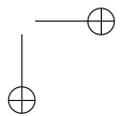
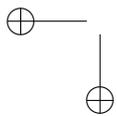
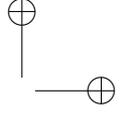
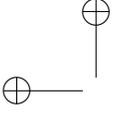


Figura 4.10: Os pontos  $a$ ,  $b$  e  $d_*$  pertencem à circunferência de um círculo em  $\mathbb{E}^2$  cujo interior não contém nenhum ponto do conjunto  $P$ .

Seja  $f : \mathbb{E}^2 \rightarrow \mathbb{R}$  uma forma afim tal que  $f^{-1}(0)$  é a reta  $\ell_{ab}$  que passa por  $a$  e  $b$ . A condição (1) nos diz que há pelo menos mais dois pontos de  $P$  em  $P - \{a, b\}$ , enquanto a condição (3) nos diz que pelo menos um deles não pertence a  $\ell_{ab}$ . Logo, há um ponto de  $P$  em  $\text{int}(H_+(f))$  ou  $\text{int}(H_-(f))$ . Sem perda de generalidade, assumamos que há pelo menos um ponto de  $P$  no semiespaço  $\text{int}(H_-(f))$ . Para cada ponto,  $d$ , de  $P$  em  $\text{int}(H_-(f))$  considere o círculo,  $C_d$ , cuja circunferência contém  $a$ ,  $b$  e  $d$ . O centro do círculo  $C_d$  pertence à reta  $\ell'$  por  $M_{ab}$  e perpendicular a  $\ell_{ab}$ . Além disso, temos também que

$$(\text{int}(C_d) \cap H_+(f)) \subset (\text{int}(C_{ab}) \cap H_+(f))$$





e

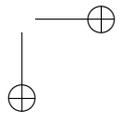
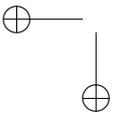
$$d \in (\text{int}(H_-(f)) - \text{int}(C_{ab})).$$

Considere o ponto  $d_*$  de  $P$  em  $\text{int}(H_-(f))$  tal que o círculo  $C_{d_*}$  possui uma circunferência cujo raio é o menor entre todos os círculos,  $C_d$ , definidos por  $a$ ,  $b$  e  $d$ , com  $d \in (P \cap \text{int}(H_-(f)))$ . Observe que o círculo  $C_{d_*}$  é único. Caso contrário, a circunferência de  $C_{d_*}$  conteria quatro pontos de  $P$ , violando a condição (2). Logo, temos que  $\text{int}(C_{d_*}) \cap H_-(f)$  não pode conter nenhum ponto de  $P$ . Mas, como não há ponto de  $P$  no interior,  $\text{int}(C_{ab})$ , de  $C_{ab}$  e nem em  $\text{int}(C_{d_*}) \cap H_-(f)$ , e como  $(C_{d_*} \cap H_+(f)) \subset C_{ab}$ , podemos concluir que o interior de  $C_{d_*}$  não contém nenhum ponto de  $P$ . Logo, pela Proposição 4.5, os pontos  $a^\omega$ ,  $b^\omega$  e  $d_*^\omega$  são vértices de uma faceta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ .

**Observação 4.3.** *Acabamos de provar, portanto, que todo ponto em  $P$  é um vértice de um triângulo resultante da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  e, reciprocamente, que todo vértice de um triângulo resultante da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é um ponto de  $P \subset \mathbb{E}^2$ .*

O próximo passo é mostrar que toda aresta de um triângulo resultante da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  também pertence a  $\mathcal{E}(\mathcal{P})$  e, portanto, também está entre as arestas resultantes da projeção. Logo, a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  contém todos os vértices e arestas dos triângulos projetados em  $\mathbb{E}^2$  e, portanto, esse conjunto de triângulos, vértices e arestas satisfaz a condição (1) da Definição 4.1.

Seja  $\overline{ab}$  o lado de um triângulo resultante da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$ . Sabemos que  $a$  e  $b$  são pontos de  $P$ . Queremos mostrar que o segmento  $\overline{a^\omega b^\omega}$  em  $\mathbb{E}^3$  é uma aresta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . Seja  $f : \mathbb{E}^2 \rightarrow \mathbb{R}$  uma forma afim tal que  $f^{-1}(0)$  é a reta  $\ell_{ab}$  que passa por  $a$  e  $b$  (ver Figura 4.11). Por razões citadas anteriormente, sabemos que há um ponto de  $P$  em  $\text{int}(H_+(f))$  ou  $\text{int}(H_-(f))$ . Sem perda de generalidade, assumamos que há um ponto de  $P$  em  $\text{int}(H_-(f))$ . Dentre todos os pontos de  $P$  em  $\text{int}(H_-(f))$ , denote por  $d$  aquele que define com  $a$  e  $b$  uma circunferência de menor raio. Denote por  $C_-$  o respectivo círculo. Note que  $\text{int}(C_-)$  não contém nenhum ponto de  $P$ .



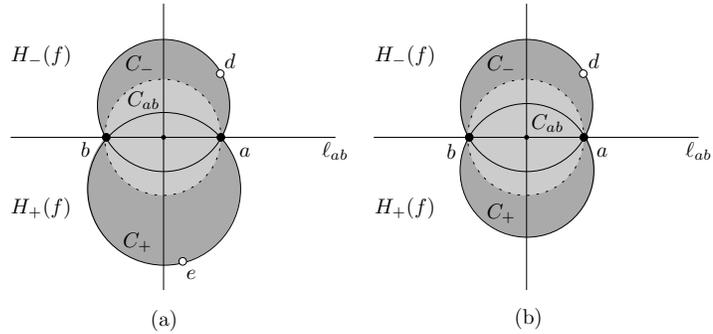


Figura 4.11: (a) Há pontos de  $P$  em  $\text{int}(H_-(f))$  e  $\text{int}(H_+(f))$  e (b) os pontos de  $P$  estão todos em  $H_-(f)$ , em que  $f$  é uma forma afim tal que  $\ell_{ab} = f^{-1}(0)$ . Em ambos os casos, temos  $C_{ab} \subset C_- \cup C_+$ . Não há pontos de  $P$  no interior de  $C_- \cup C_+$  e, logo, nem no interior de  $C_{ab}$ .

Se houver algum ponto de  $P$  em  $\text{int}(H_+(f))$ , denote por  $e$  aquele que define com  $a$  e  $b$  o círculo,  $C_+$ , com a circunferência de menor raio. Se não houver nenhum ponto de  $P$  em  $\text{int}(H_+(f))$ , então  $C_+$  é qualquer círculo cuja circunferência passa por  $a$  e  $b$  e cujo centro pertence a  $\text{int}(H_+(f))$ . Observe que o interior,  $\text{int}(C_+)$ , de  $C_+$  não contém nenhum ponto de  $P$ . Observe também que o círculo,  $C_{ab}$ , cuja circunferência passa por  $a$  e  $b$  e cujo centro é o ponto médio,  $M_{ab}$ , de  $\overline{ab}$  está contido no conjunto união,  $C_- \cup C_+$ , dos círculos  $C_-$  e  $C_+$ .

Seja

$$J = \bigcup_{x \in \partial(C_{ab})} x^\omega,$$

em que  $\partial(C_{ab})$  denota a fronteira (circunferência) do círculo  $C_{ab}$  em  $\mathbb{E}^2$ . Afirmamos que todo ponto de  $J$  pertence a um mesmo plano,  $H_{ab}$ , em  $\mathbb{E}^3$ , tal que um ponto  $x$  de  $\mathbb{E}^3$  está em  $H_{ab}$  se, e somente se,

$$x = \alpha \cdot a^\omega + \beta \cdot b^\omega + \gamma \cdot n^\omega,$$

em que  $n$  é algum ponto de  $\partial(C_{ab})$ , com  $n \neq a, b$ , e  $\alpha, \beta$  e  $\gamma$  são

números reais, com  $\alpha + \beta + \gamma = 1$ . Seja  $u$  um ponto qualquer de  $\partial(C_{ab})$ . Então, existem número reais  $\mu, \nu$  e  $\lambda$  tais que  $\mu + \nu + \lambda = 1$  e

$$u = \mu \cdot a + \nu \cdot b + \lambda \cdot n,$$

pois  $\{a, b, n\}$  é um conjunto afimemente independente em  $\mathbb{E}^2$ . Queremos mostrar que  $u^\omega$  pertence a  $H_{ab}$ . Suponha que  $(x, y), (x_a, y_a), (x_b, y_b)$  e  $(x_n, y_n)$  são as coordenadas de  $u, a, b$  e  $n$ , respectivamente, com relação ao referencial  $(O, \mathbf{i}, \mathbf{j})$  de  $\mathbb{E}^2$ . Por hipótese, as coordenadas de  $u, a, b$  e  $n$  estão relacionadas entre si pelas duas seguintes equações:

$$x = \mu x_a + \nu x_b + \lambda x_n \quad \text{e} \quad y = \mu y_a + \nu y_b + \lambda y_n.$$

Já as coordenadas de  $u^\omega, a^\omega, b^\omega$  e  $n^\omega$ , com relação ao referencial  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$  de  $\mathbb{E}^3$ , são  $(x, y, z), (x_a, y_a, z_a), (x_b, y_b, z_b)$  e  $(x_n, y_n, z_n)$ , respectivamente, em que  $z = x^2 + y^2, z_a = x_a^2 + y_a^2, z_b = x_b^2 + y_b^2$  e  $z_n = x_n^2 + y_n^2$ . Isto significa que o ponto  $u^\omega \in \mathbb{E}^3$  pode ser definido como

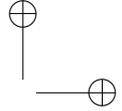
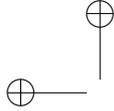
$$u^\omega = \mu \cdot a^\omega + \nu \cdot b^\omega + \lambda \cdot n^\omega$$

se, e somente se,  $z = \mu z_a + \nu z_b + \lambda z_n$  ou, de forma equivalente, se, somente se,  $x^2 + y^2 = \mu(x_a^2 + y_a^2) + \nu(x_b^2 + y_b^2) + \lambda(x_n^2 + y_n^2)$ . Mas, se  $(x_m, y_m)$  são as coordenadas do centro,  $M_{ab}$ , de  $C_{ab}$ , com relação a  $(O, \mathbf{i}, \mathbf{j})$ , e  $r$  é o raio de  $C_{ab}$ , então a expressão  $z - x_m^2 - y_m^2$  é tal que

$$\begin{aligned} z - x_m^2 - y_m^2 &= x^2 + y^2 - x_m^2 - y_m^2 \\ &= (x - x_m)^2 + (y - y_m)^2 + 2xx_m + 2yy_m \\ &= r^2 + 2xx_m + 2yy_m. \end{aligned}$$

Como  $\mu + \nu + \lambda = 1, x = \mu x_a + \nu x_b + \lambda x_n$  e  $y = \mu y_a + \nu y_b + \lambda y_n$ ,

$$\begin{aligned} r^2 + 2xx_m + 2yy_m &= (\mu + \nu + \lambda)r^2 \\ &\quad + 2(\mu x_a + \nu x_b + \lambda x_n)x_m \\ &\quad + 2(\mu y_a + \nu y_b + \lambda y_n)y_m \\ &= \mu r^2 + \mu 2x_a x_m + \mu 2y_a y_m \\ &\quad + \nu r^2 + \nu 2x_b x_m + \nu 2y_b y_m \\ &\quad + \lambda r^2 + \lambda 2x_c x_m + \lambda 2y_c y_m. \end{aligned}$$



Como  $a$ ,  $b$  e  $n$  pertencem a  $\partial(C_{ab})$ , temos que  $r^2 = (x_a - x_m)^2 + (y_a - y_m)^2$ ,  $r^2 = (x_b - x_m)^2 + (y_b - y_m)^2$  e  $r^2 = (x_n - x_m)^2 + (y_n - y_m)^2$ .  
Daí,

$$\begin{aligned}\mu r^2 &= \mu(x_a - x_m)^2 + \mu(y_a - y_m)^2, \\ \nu r^2 &= \nu(x_b - x_m)^2 + \nu(y_b - y_m)^2\end{aligned}$$

e

$$\lambda r^2 = \lambda(x_n - x_m)^2 + \lambda(y_n - y_m)^2.$$

Logo,

$$\begin{aligned}r^2 + 2xx_m + 2yy_m &= \mu x_a^2 + \mu y_a^2 - \mu x_m^2 - \mu y_m^2 \\ &\quad + \nu x_b^2 + \nu y_b^2 - \nu x_m^2 - \nu y_m^2 \\ &\quad + \lambda x_n^2 + \lambda y_n^2 - \lambda x_m^2 - \lambda y_m^2 \\ &= \mu z_a + \nu z_b + \lambda z_n - x_m^2 - y_m^2\end{aligned}$$

e, portanto,

$$z = \mu z_a + \nu z_b + \lambda z_n,$$

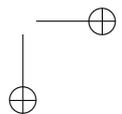
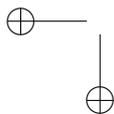
o que implica que *todo* ponto de  $J$  pertence a  $H_{ab}$ . Como  $J$  pertence ao parabolóide  $\Gamma$  e a projeção ortogonal de  $J$  sobre  $\mathbb{E}^2$  é a circunferência  $\partial(C_{ab})$ , a Proposição 4.1 implica que  $H_{ab} \cap \Gamma = J$ . Observe que o plano  $H_{ab}$  contém o segmento  $\overline{a^\omega b^\omega}$ . Observe também que se  $u$  pertencesse ao interior,  $\text{int}(C_{ab})$ , do círculo  $C_{ab}$ , então teríamos que

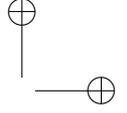
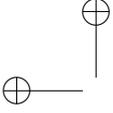
$$\begin{aligned}z - x_m^2 - y_m^2 &= x^2 + y^2 - x_m^2 - y_m^2 \\ &= (x - x_m)^2 + (y - y_m)^2 + 2xx_m + 2yy_m \\ &< r^2 + 2xx_m + 2yy_m.\end{aligned}$$

Por sua vez, se  $u$  pertencesse ao exterior do círculo  $C_{ab}$ , então teríamos que

$$\begin{aligned}z - x_m^2 - y_m^2 &= x^2 + y^2 - x_m^2 - y_m^2 \\ &= (x - x_m)^2 + (y - y_m)^2 + 2xx_m + 2yy_m \\ &> r^2 + 2xx_m + 2yy_m.\end{aligned}$$

Logo, existe uma forma afim,  $g : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $H_{ab} = g^{-1}(0)$  e  $g(x) < 0$ , para todo  $x \in \mathbb{E}^3$  tal que  $x = y^\omega$ , para algum ponto  $y$  do





exterior do círculo  $C_{ab}$ . Como não há ponto de  $P$  no interior de  $C_{ab}$ , podemos concluir que o segmento  $\overline{a^\omega b^\omega} \in H_{ab}$  é uma aresta de  $\mathcal{P}$ . O que resta é provar que  $\overline{a^\omega b^\omega}$  é uma aresta do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$ .

Por ser uma aresta de  $\mathcal{P}$ , a aresta  $\overline{a^\omega b^\omega}$  pertence a exatamente duas facetas de  $\mathcal{P}$ . Uma delas está certamente em  $\mathcal{E}(\mathcal{P})$ , pois a Proposição 4.5 garante que  $a^\omega$ ,  $b^\omega$  e  $d^\omega$  são os vértices de uma faceta em  $\mathcal{E}(\mathcal{P})$ , já que o interior de  $C_-$  não possui nenhum ponto de  $P$  (ver Figura 4.11). Se a circunferência do círculo  $C_+$  contiver um ponto  $e$  de  $P$ , com  $e \neq a, b$ , então a Proposição 4.5 garante que a outra faceta de  $\mathcal{P}$  contendo  $\overline{a^\omega b^\omega}$  tem como vértices  $a^\omega$ ,  $b^\omega$  e  $e^\omega$  e também pertence a  $\mathcal{E}(\mathcal{P})$ . Caso contrário, a outra faceta não pertence a  $\mathcal{E}(\mathcal{P})$ . Denote por  $F_1$  e  $F_2$  as duas facetas de  $\mathcal{P}$  contendo  $\overline{a^\omega b^\omega}$ . Suponha que  $F_1$  pertença a  $\mathcal{E}(\mathcal{P})$ . Agora, considere os casos mutuamente exclusivos a seguir:

- (a)  $F_2 \in \mathcal{E}(\mathcal{P})$  e
- (b)  $F_2 \notin \mathcal{E}(\mathcal{P})$ .

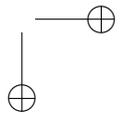
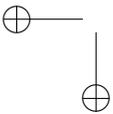
Como  $F_1 \in \mathcal{E}(\mathcal{P})$ , existe uma forma afim,  $g_{F_1} : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $F_1 = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g_{F_1}(x) = 0\}$ ,  $g_{F_1}(y) \leq 0$ , para todo  $y \in \mathcal{P}$ , e  $g_{F_1}(O + \alpha \mathbf{k}) > 0$ , para todo  $\alpha \in [\alpha_1, -\infty)$  e algum  $\alpha_1 \in \mathbb{R}$ , com  $\alpha_1 < 0$ . Se o caso (a) ocorre, então também existe uma forma afim,  $g_{F_2} : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $F_2 = \mathcal{P} \cap \{y \in \mathbb{E}^3 \mid g_{F_2}(y) = 0\}$ ,  $g_{F_2}(y) \leq 0$ , para todo  $y \in \mathcal{P}$ , e  $g_{F_2}(O + \alpha \mathbf{k}) > 0$ , para todo  $\alpha \in [\alpha_2, -\infty)$  e algum  $\alpha_2 \in \mathbb{R}$ , com  $\alpha_2 < 0$ . Sendo assim, defina outra forma afim,  $g_{ab} : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $g_{ab}(x) = g_{F_1}(x) + g_{F_2}(x)$ , para todo  $x \in \mathbb{E}^3$ . Observe que  $g_{ab}(x) = 0$  se, e somente se,  $x = F_1 \cap F_2$ . Observe também que  $g_{ab}(x) \leq 0$ , para todo  $x \in \mathcal{P}$ . Consequentemente, temos que

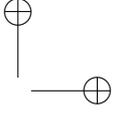
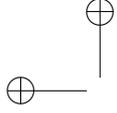
$$\overline{ab} = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g_{ab}(x) = 0\} \quad \text{e} \quad g_{ab}(y) \leq 0,$$

para todo  $y \in \mathcal{P}$ . Finalmente,  $g_{ab}(O + \alpha \mathbf{k}) = g_{F_1}(O + \alpha \mathbf{k}) + g_{F_2}(O + \alpha \mathbf{k}) > 0$ , para todo  $\alpha \in [\beta, -\infty)$ , com  $\beta = \min\{\alpha_1, \alpha_2\}$ . Logo, temos que

$$\overline{a^\omega b^\omega} \in \mathcal{E}(\mathcal{P}).$$

Se o caso (b) ocorre, então considere o plano  $H_2$  paralelo ao vetor  $\mathbf{k}$  e contendo a aresta  $\overline{a^\omega b^\omega}$ . Observe que a projeção ortogonal de  $H_2$





sobre  $\mathbb{E}^2$  é exatamente a reta  $\ell_{ab}$  (ver Figura 4.11). Seja  $g_2 : \mathbb{E}^3 \rightarrow \mathbb{R}$  uma forma afim tal que  $H_2 = g_2^{-1}(0)$ . Observe que a projeção ortogonal de  $\text{int}(H_+(g_2))$  sobre  $\mathbb{E}^2$  é exatamente  $\text{int}(H_+(f))$  ou  $\text{int}(H_-(f))$ , em que  $f : \mathbb{E}^2 \rightarrow \mathbb{R}$  é a forma afim que define a reta  $\ell_{ab}$ . Assuma que a projeção ortogonal de  $\text{int}(H_+(g_2))$  sobre  $\mathbb{E}^2$  é exatamente  $\text{int}(H_+(f))$ . Caso contrário, basta substituir  $g_2$  por  $-g_2$ . Por hipótese, não existe nenhum ponto de  $P$  em  $\text{int}(H_+(f))$  e, portanto, não pode existir nenhum ponto de  $P^\omega$  em  $\text{int}(H_+(g_2))$ . Logo, temos que

$$\mathcal{P} \cap \text{int}(H_+(g_2)) = \emptyset,$$

o que implica que  $g_2(x) \leq 0$ , para todo  $x \in \mathcal{P}$ . Além disso, como  $H_2$  é paralelo a  $\mathbf{k}$ , temos também que  $g_2(O + \alpha\mathbf{k}) = \beta$ , para todo  $\alpha \in \mathbb{R}$  e para alguma constante  $\beta \in \mathbb{R}$ . Assim como no caso (a), definimos a função  $g_{ab} : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $g_{ab}(x) = g_{F_1}(x) + g_2(x)$ , para todo  $x \in \mathbb{E}^3$ .

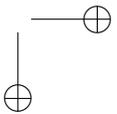
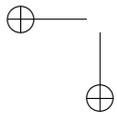
Podemos verificar imediatamente que se  $x \in \mathcal{P}$ , então  $g_{ab}(x) = 0$  se, e somente se,  $x \in \overline{a^\omega b^\omega}$ . De fato, da definição de  $H_2$ , concluímos que

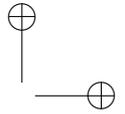
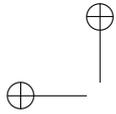
$$\overline{a^\omega b^\omega} \subseteq (F_1 \cap H_2).$$

Além disso, temos que os únicos vértices de  $\mathcal{P}$  em  $F_1 \cap H_2$  são  $a^\omega$  e  $b^\omega$  e que  $g_{ab}(x) = 0$  se, e somente se,  $x = F_1 \cap H_2$ . Daí, concluímos que

$$\overline{a^\omega b^\omega} = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g_{ab}(x) = 0\}.$$

Observe também que  $g_{ab}(y) \leq 0$ , para todo  $y \in \mathcal{P}$ , pois  $g_{ab}(y) = g_{F_1}(y) + g_2(y)$  e  $g_{F_1}(x), g_2(x) \leq 0$ , para todo  $x \in \mathcal{P}$ . Finalmente, por  $H_2$  ser paralelo a  $\mathbf{k}$ , temos que  $g_2(O + \alpha\mathbf{k}) = \beta$ , para todo  $\alpha \in \mathbb{R}$  e para alguma constante  $\beta \in \mathbb{R}$ . Como  $F_1$  está em  $\mathcal{E}(\mathcal{P})$ , existe  $\alpha_1 \in \mathbb{R}$ , com  $\alpha_1 < 0$ , tal que  $g_{F_1}(O + \alpha\mathbf{k}) > 0$ , para todo  $\alpha \in [\alpha_1, -\infty)$ . Por  $g_{F_1}$  ser uma forma afim, temos que  $\nabla g_{F_1}(x) \cdot \mathbf{k} = \lambda z$ , para alguma constante  $\lambda \in \mathbb{R}$  e para todo ponto  $x$  de  $\mathbb{E}^3$  cuja terceira coordenada com relação a  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$  é  $z$ . Logo, a restrição,  $g_{F_1}|_{\ell'} : \mathbb{E} \rightarrow \mathbb{R}$ , de  $g_{F_1}$  à reta  $\ell'$  paralela a  $\mathbf{k}$  por  $O$  é estritamente crescente (decrecente). Como  $F_1 \in \mathcal{E}(\mathcal{P})$ , o plano  $g_{F_1}^{-1}(0)$  contendo  $F_1$  não é paralelo a  $\mathbf{k}$  e, portanto, existe um único ponto  $q = O + \gamma_0\mathbf{k}$  em  $\ell'$ , em que  $\gamma_0$  é uma constante em  $\mathbb{R}$ , tal que  $g_{F_1}(q) = 0$ . Note que  $\gamma_0 \notin [\alpha_1, -\infty)$ . Caso contrário, teríamos  $g_{F_1}(q) > 0$ . Daí, podemos concluir que  $\gamma_0 > \alpha_1$ , o





que implica que  $g_{F_1}(O + \gamma \mathbf{k}) > g_{F_1}(O + \mu \mathbf{k})$  para quaisquer  $\gamma, \mu \in \mathbb{R}$ , com  $\gamma < \mu$ . Além disso, deve existir  $\gamma_1 \in \mathbb{R}$  tal que  $g_{F_1}(O + \gamma_1 \mathbf{k}) > |\beta|$ . Mas, isto implica que  $g_{ab}(O + \alpha \mathbf{k}) = g_{F_1}(O + \alpha \mathbf{k}) + g_2(O + \alpha \mathbf{k}) = g_{F_1}(O + \alpha \mathbf{k}) + \beta > 0$ , para todo  $\alpha \in [\min\{\alpha_1, \gamma_1\}, -\infty)$ . Logo, temos que

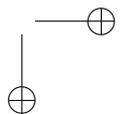
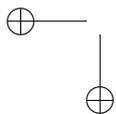
$$\overline{a^\omega b^\omega} \in \mathcal{E}(\mathcal{P}).$$

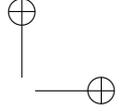
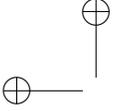
Acabamos de mostrar que se  $a$  e  $b$  são vértices de um triângulo resultante da projeção de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$ , então  $\overline{a^\omega b^\omega}$  é uma aresta do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$ . Logo, as arestas dos triângulos resultantes da projeção ortogonal das facetas de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  são, elas próprias, resultantes da projeção ortogonal de arestas de  $\mathcal{P}$  sobre  $\mathcal{E}(\mathcal{P})$ .

Reciprocamente, se  $L$  é uma aresta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ , então  $L$  deve pertencer a uma faceta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . De fato, como  $L$  está em  $\mathcal{E}(\mathcal{P})$ , ela não pode ser paralela ao vetor  $\mathbf{k}$ . Logo, a projeção ortogonal de  $L$  em  $\mathbb{E}^2$  é um segmento de reta, digamos  $\overline{ab}$ , em  $\mathbb{E}^2$ , com  $a, b \in P$ . Usando os argumentos ilustrados pela Figura 4.11, podemos encontrar um círculo,  $C_-$ , em  $\mathbb{E}^2$  contendo os pontos  $a$  e  $b$  e um terceiro ponto,  $a$ , de  $P$  em sua circunferência e tal que  $\text{int}(C_-)$  não contém nenhum ponto de  $P$ . Pela Proposição 4.5, os pontos  $a^\omega$ ,  $b^\omega$  e  $d^\omega$  são os vértices de uma faceta,  $F$ , de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . Como  $a^\omega$  e  $b^\omega$  são os vértices de  $L$ , temos que  $L$  é uma aresta de  $F$  e, portanto, *toda aresta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$  pertence a uma faceta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ , como queríamos provar.*

**Observação 4.4.** *Com base nos fatos provados até então, podemos afirmar que a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é um conjunto de triângulos, arestas e vértices no qual todo vértice é vértice de um triângulo e toda aresta é aresta de um triângulo. Reciprocamente, todos os vértices e arestas desses triângulos também pertencem ao conjunto de vértices, arestas e triângulos resultantes da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$ . Note que este conjunto é “discreto” e finito.*

A Observação 4.4 nos diz que o conjunto de vértices, arestas e triângulos resultantes da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  satisfaz a condição (1) da Definição 4.1. Vamos provar que ele também satisfaz a condição (2), constituindo-se em um complexo simplicial em  $\mathbb{E}^2$ .





A condição (2) da Definição 4.1 diz que se quaisquer duas faces de um complexo simplicial possuem interseção não vazia, então a interseção é uma face do complexo. Denote por  $\pi : \mathcal{E}(\mathcal{P}) \rightarrow \mathbb{E}^2$  a projeção ortogonal de  $\mathbb{E}^3$  sobre  $\mathbb{E}^2$  restrita ao envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$ . Suponha que dois vértices, arestas ou triângulos,  $\tau_1$  e  $\tau_2$ , da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  são tais que  $(\tau_1 \cap \tau_2) \neq \emptyset$ . Todo ponto  $a$  em  $\tau_1 \cap \tau_2$  é tal que  $\pi^{-1}(a)$  pertence às faces  $\pi^{-1}(\tau_1)$  e  $\pi^{-1}(\tau_2)$  de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . Isto é,

$$\pi^{-1}(\tau_1 \cap \tau_2) \subseteq \pi^{-1}(\tau_1) \cap \pi^{-1}(\tau_2).$$

Por outro lado, pela Proposição 4.4(b), temos que  $\pi^{-1}(\tau_1) \cap \pi^{-1}(\tau_2)$  é uma face,  $F$ , de  $\mathcal{P}$ . Por  $F$  pertencer a  $\pi^{-1}(\tau_1)$  e  $\pi^{-1}(\tau_2)$ , temos que  $F$  é um vértice ou aresta de  $\mathcal{P}$ . Além disso, como  $\pi^{-1}(\tau_1)$  e  $\pi^{-1}(\tau_2)$  estão em  $\mathcal{E}(\mathcal{P})$ , temos que  $F \in \mathcal{E}(\mathcal{P})$ . Pelo que provamos anteriormente, sabemos que  $\pi(F)$  é um vértice ou aresta de  $\tau_1$  e  $\tau_2$ . Logo,

$$\pi(F) = \pi(\pi^{-1}(\tau_1) \cap \pi^{-1}(\tau_2)) \subseteq (\tau_1 \cap \tau_2),$$

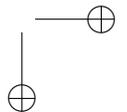
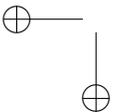
e, portanto,

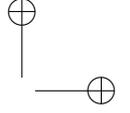
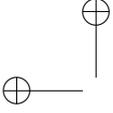
$$\pi(F) = \tau_1 \cap \tau_2.$$

Logo, a interseção não vazia de dois elementos (vértices, arestas ou triângulos) da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é uma face de ambos os elementos. Portanto, a projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  satisfaz a condição (2) da Definição 4.1 e é, de fato, um complexo simplicial.

Daremos um passo adiante e mostraremos que o complexo simplicial,  $\mathcal{K}$ , correspondente ao conjunto de vértices, arestas e triângulos da projeção ortogonal de  $\mathcal{E}(\mathcal{P})$  sobre  $\mathbb{E}^2$  é, na verdade, uma triangulação de  $P$  em  $\mathbb{E}^2$ . Para tal é suficiente mostrar que o espaço subjacente,  $|\mathcal{K}|$ , de  $\mathcal{K}$  é o fecho convexo,  $FC(P)$ , de  $P$ , pois já sabemos que o conjunto de vértices de  $\mathcal{K}$  é exatamente igual ao conjunto  $P$ .

As condições (1) e (3) sobre  $P$  garantem que  $FC(P)$  é um polígono em  $\mathbb{E}^2$ . Vamos considerar um ponto  $x$  no interior de  $FC(P)$ . Precisamos mostrar que existe um triângulo em  $\mathcal{K}$  que contém  $P$ . O Teorema de Carathéodory (ver Teorema 3.4) nos diz que há três vértices, digamos  $a$ ,  $b$  e  $c$ , em  $P$  tais que  $x = \alpha \cdot a + \beta \cdot b + \gamma \cdot c$ , para alguns





$\alpha, \beta, \gamma \in \mathbb{R}$ , com  $\alpha + \beta + \gamma = 1$  e  $\alpha, \beta, \gamma \geq 0$ . Isto é, o ponto  $x$  pertence ao triângulo  $\tau$  de vértices em  $a, b$  e  $c$ . Suponha que  $\tau$  não pertence a  $\mathcal{K}$ . Caso contrário, não teríamos o que provar. Como  $\mathcal{P} = FC(P^\omega)$ ,

$$x_\star = \alpha \cdot a^\omega + \beta \cdot b^\omega + \gamma \cdot c^\omega$$

é um ponto em  $\mathcal{P}$ . Considere a reta  $\ell''$  paralela a  $\mathbf{k}$  por  $x_\star$ . Por  $\mathcal{P}$  ser convexo, a interseção entre  $\ell''$  e  $\mathcal{P}$  é (a) um segmento de reta ou (b) apenas um ponto. Afirmamos que, no caso (a), um dos extremos do segmento está em  $\mathcal{E}(\mathcal{P})$ , Por sua vez, no caso (b), o ponto pertence a  $\mathcal{E}(\mathcal{P})$ .

Considere o caso (a). Sejam  $x'$  e  $x''$  os pontos extremos do segmento correspondente à interseção entre  $\ell''$  e  $\mathcal{P}$ . Suponha que a terceira coordenada,  $x'_3$ , de  $x'$  é menor do que a terceira coordenada,  $x''_3$ , de  $x''$ . Seja  $F$  qualquer face de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . Então, existe uma forma afim,  $g_F : \mathbb{E}^3 \rightarrow \mathbb{R}$ , tal que  $F = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g_F(x) = 0\}$ ,  $g_F(y) \leq 0$ , para todo  $y \in \mathcal{P}$ , e  $g_F(O + \alpha \mathbf{k}) > 0$ , para todo  $\alpha \in \mathbb{R}$ , com  $\alpha \in [\alpha_1, -\infty)$ , em que  $\alpha_1 < 0$ . Vimos antes que para quaisquer  $\gamma, \mu \in \mathbb{R}$ , com  $\gamma < \mu$ , temos  $g_F(O + \gamma \mathbf{k}) > g_F(O + \mu \mathbf{k})$ . Vimos também que  $\nabla g_F(x) \cdot \mathbf{k} = \lambda x_3$ , para alguma constante  $\lambda \in \mathbb{R}$  e para todo  $x \in \mathbb{E}^3$  tal que  $x_3$  é a terceira coordenada de  $x$  com relação ao referencial  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$  de  $\mathbb{E}^3$ . Como  $\nabla g_F$  cresce na direção de crescimento de  $g_F$ , temos que  $\lambda < 0$ . Como  $\ell''$  é paralela a  $\mathbf{k}$  e  $0 < x'_3 < x''_3$ , temos

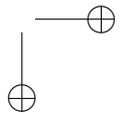
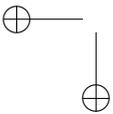
$$g_F(x') > g_F(x'').$$

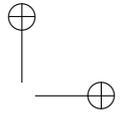
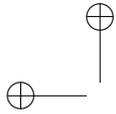
Esta desigualdade implica que  $x'$  é ponto de máximo, em  $\overline{x'x''}$ , para qualquer forma afim que caracteriza uma face de  $\mathcal{P}$  como pertencente a  $\mathcal{E}(\mathcal{P})$ .

Por outro lado, se uma face  $F$  de  $\mathcal{P}$  não pertence a  $\mathcal{E}(\mathcal{P})$ , então não pode existir  $\alpha_1 \in \mathbb{R}$ , com  $\alpha_1 < 0$ , tal que  $g_F(O + \alpha \mathbf{k}) > 0$ , para todo  $\alpha \in [\alpha_1, -\infty)$ . Isto implica que para quaisquer  $\gamma, \mu \in \mathbb{R}$ , com  $\gamma < \mu$ , temos  $g_F(O + \gamma \mathbf{k}) \leq g_F(O + \mu \mathbf{k})$ , havendo igualdade se, e somente se, a face  $F$  pertence a um plano paralelo ao vetor  $\mathbf{k}$ . Daí, temos

$$g_F(x') \leq g_F(x'').$$

A desigualdade acima nos diz que  $x''$  é ponto de máximo, em  $\overline{x'x''}$ , para qualquer forma afim de qualquer face de  $\mathcal{P}$  que não está





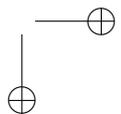
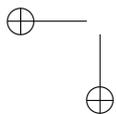
em  $\mathcal{E}(\mathcal{P})$ . Suponha que  $x'$  pertença a uma face  $F$  de  $\mathcal{P}$  que não está em  $\mathcal{E}(\mathcal{P})$ . Então,  $g_F(x') = 0$ . Se  $g_F^{-1}(0)$  não é paralelo a  $\mathbf{k}$ , então  $g_F(x'') > 0$ , o que implica que  $x'' \notin \mathcal{P}$ : *um absurdo!* Logo,  $g_F^{-1}(0)$  deve ser paralelo a  $\mathbf{k}$ , o que implica que  $x'' \in F$ . Mas, neste caso, o ponto  $x = \alpha \cdot a + \beta \cdot b + \gamma \cdot c$  não pode pertencer ao interior de  $FC(P)$ , já que não pode haver pontos de  $P^\omega$  em  $\text{int}(H_-(g_F))$  e em  $\text{int}(H_+(g_F))$ : *um absurdo!* Logo,  $x'$  pertence a uma face de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ .

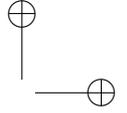
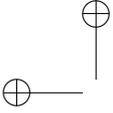
Considere o caso (b). Seja  $x'''$  o ponto correspondente à interseção entre  $\ell''$  e  $\mathcal{P}$ . Afirmamos que  $x'''$  não pertence a uma faceta de  $\mathcal{P}$  em um plano paralelo ao vetor  $\mathbf{k}$ . Caso contrário, a interseção entre  $\ell''$  e  $\mathcal{P}$  seria um segmento de reta. Isto significa que  $x'''$  é um vértice de  $\mathcal{P}$  ou um ponto interior de uma aresta de  $\mathcal{P}$ . Em ambos os casos, existe um plano,  $H$ , por  $x'''$  e paralelo ao vetor  $\mathbf{k}$  tal que todos os vértices de  $\mathcal{P}$  pertencem a apenas um dos semiespaços definidos por  $H$ . Mas, isto implica que a projeção ortogonal,  $x$ , de  $x'''$  sobre  $\mathbb{E}^2$  não é ponto interior de  $FC(P)$ . Logo, o caso (b) pode ser ignorado, pois ele não ocorre!

*Acabamos de provar que todo ponto  $x$  no interior de  $FC(P)$  é um ponto de  $|\mathcal{K}|$ , pois  $x$  é a projeção ortogonal sobre  $\mathbb{E}^2$  de um ponto em uma faceta de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ .* O que nos resta é mostrar que se  $x$  não está no interior de  $FC(P)$ , ou seja, se  $x$  está na fronteira de  $FC(P)$ , então  $x$  também pertence a  $|\mathcal{K}|$ . De fato, se  $x$  é um ponto extremo de  $FC(P)$ , então  $x$  é um ponto de  $P$ . Mas, já provamos que todo ponto de  $P^\omega$  é um vértice de  $\mathcal{P}$  em  $\mathcal{E}(\mathcal{P})$ . Logo, vamos supor que  $x$  pertence ao interior de uma aresta de fronteira de  $FC(P)$ . Sejam  $a$  e  $b$  os extremos da aresta contendo  $x$ . Usando os argumentos ilustrados na Figura 4.11, podemos afirmar que existe um triângulo de  $\mathcal{K}$  que contém  $\overline{ab}$  e, portanto, o ponto  $x$  também está em  $|\mathcal{K}|$ . Isto implica que  $FC(P) \subseteq |\mathcal{K}|$ . Por outro lado, como (i) todo vértice de  $\mathcal{K}$  é um ponto de  $P$ , (ii) toda aresta ou triângulo de  $\mathcal{K}$  é uma combinação convexa dos seus vértices e (iii)  $|\mathcal{K}|$  é a união dos pontos dos vértices, arestas e triângulos de  $\mathcal{K}$ , temos que  $|\mathcal{K}| \subseteq FC(P)$ . Logo, devemos ter

$$|\mathcal{K}| = FC(P).$$

O seguinte teorema enuncia o resultado geral que acabamos de provar:





**Teorema 4.6.** *Seja  $P$  um conjunto finito de pontos de  $\mathbb{E}^2$  tal que (1)  $P$  possui pelo menos quatro pontos, (2) nenhum subconjunto de quatro pontos de  $P$  reside em uma mesma circunferência e (3) há um subconjunto de  $P$  afimemente independente com três pontos de  $\mathbb{E}^2$ . Então, o conjunto de triângulos, arestas e vértices resultantes da projeção ortogonal do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , do politopo  $\mathcal{P} = FC(P^\omega)$ , em que  $P^\omega = \{(x, y, z) \in \mathbb{E}^3 \mid (x, y) \in P \text{ e } z = x^2 + y^2\}$ , é uma triangulação de  $P$ .*

O Teorema 4.6 estabelece a existência de uma triangulação de  $P$  dado que  $P$  satisfaça as condições (1), (2) e (3) do enunciado do teorema.

**Definição 4.3.** *Seja  $P$  um conjunto finito de pontos de  $\mathbb{E}^2$  tal que (1)  $P$  possui pelo menos quatro pontos, (2) nenhum subconjunto de quatro pontos de  $P$  reside em uma mesma circunferência e (3) há um subconjunto de  $P$  afimemente independente com três pontos de  $\mathbb{E}^2$ . Então, a triangulação que consiste dos triângulos, arestas e vértices da projeção ortogonal do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , do politopo  $\mathcal{P} = FC(P^\omega)$  é denominada triangulação de Delaunay de  $P$  e denotada por  $\mathcal{TD}(P)$ .*

A Figura 4.12 mostra uma triangulação de Delaunay em  $\mathbb{E}^2$ .

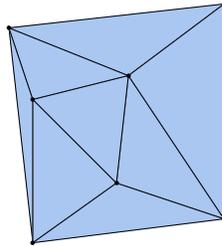
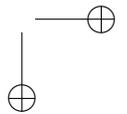
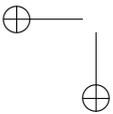
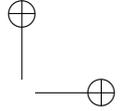
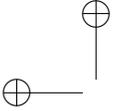


Figura 4.12: Triangulação de Delaunay correspondente à projeção ortogonal do envelope inferior,  $\mathcal{E}(\mathcal{P})$ , do politopo  $\mathcal{P}$  exibido na Figura 4.8.

Observe que a triangulação de Delaunay é única, pois a projeção ortogonal,  $\pi : \mathbb{E}^3 \rightarrow \mathbb{E}^2$ , de  $\mathbb{E}^3$  sobre  $\mathbb{E}^2$ , quando restrita ao envelope



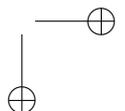
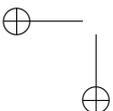


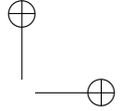
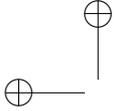
inferior,  $\mathcal{E}(\mathcal{P})$ , do politopo  $\mathcal{P}$  é uma função bijetora. Logo, podemos estabelecer um isomorfismo entre os elementos de  $\mathcal{K}$  e aqueles de  $\mathcal{E}(\mathcal{P})$ . Este isomorfismo garante que se  $[v_0, \dots, v_k]$  é um  $k$ -simplexo em  $\mathcal{E}(\mathcal{P})$  (com  $k = 0, 1, 2$ ), então  $[\pi(v_0), \dots, \pi(v_k)]$  é um  $k$ -simplexo em  $\mathcal{K}$ . Portanto, só pode existir uma maneira de definirmos o conjunto  $\mathcal{K}$ .

Para concluir esta seção, vamos discutir quais das três condições sobre  $P$  do enunciado do Teorema 4.6 são, de fato, necessárias para se definir uma triangulação de  $P$  e que tipo de complexo podemos obter violando uma ou mais dessas condições. Suponha, inicialmente, que o conjunto  $P$  viole a condição (1). Isto é,  $P$  possui menos do que quatro pontos, o que torna a condição (2) trivialmente verdadeira. Se  $P$  satisfaz a condição (3), então  $FC(P^\omega)$  é um único triângulo,  $\tau$ , em  $\mathbb{E}^3$  e a projeção ortogonal de  $\tau$  sobre  $\mathbb{E}^2$  define uma triangulação de  $P$  consistindo do triângulo  $\pi(\tau)$ , das suas três arestas e de seus três vértices (os pontos de  $P$ ). Observe que esta triangulação também é única.

Suponha que  $P$  viole a condição (3) e, possivelmente, a condição (1). Então, os pontos de  $P$  são colineares (se houver mais de um ponto). Logo, a condição (2) é válida e  $FC(P^\omega)$  é um politopo de dimensão 0, 1 ou 2 em  $\mathbb{E}^3$ . A dimensão de  $FC(P^\omega)$  é 0, 1 ou 2 se  $P$  tiver exatamente um, exatamente dois ou mais de dois pontos, respectivamente. Em qualquer um desses casos, podemos mostrar que a projeção ortogonal do envelope inferior,  $\mathcal{E}(FC(P^\omega))$ , de  $FC(P^\omega)$  sobre  $\mathbb{E}^2$  define uma triangulação de  $P$ , que também é única mas não possui triângulos, como aquela na Figura 4.5 (ver Problema 4.2). Observe que  $\mathcal{E}(FC(P^\omega))$  é uma curva poligonal simples e aberta em  $\mathbb{E}^3$ .

As observações dos dois parágrafos anteriores nos dizem que as condições (1) e (3) não são necessárias para definirmos uma triangulação de  $P$ , que também é única. O que dizer da condição (2)? Se  $P$  viola a condição (2), mas satisfaz as condições (1) e (3), então nem todas as facetas de  $FC(P^\omega)$  são triângulos (ver Proposição 4.5). Se uma faceta com mais de três vértices pertence ao envelope inferior,  $\mathcal{E}(FC(P^\omega))$ , de  $FC(P^\omega)$ , então a projeção ortogonal de  $\mathcal{E}(FC(P^\omega))$  em  $\mathbb{E}^2$  contém polígonos (convexos) com mais de três vértices. No entanto, é possível mostrar que o conjunto de vértices, arestas, triângulos e demais polígonos convexos resultante





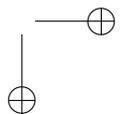
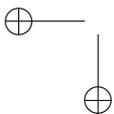
da projeção ortogonal de  $\mathcal{E}(FC(P^\omega))$  em  $\mathbb{E}^2$  é um *complexo politopal* denominado *subdivisão de Delaunay de  $P$* . Um complexo politopal é um conjunto de politopos que satisfazem duas condições equivalentes às condições (1) e (2) da definição de complexo simplicial (Definição 4.1) [138].

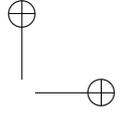
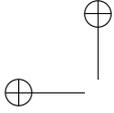
A subdivisão de Delaunay de  $P$  sempre pode ser transformada em uma triangulação de  $P$  se adicionarmos arestas ao polígono da subdivisão que não são triângulos. Esta partição de polígonos em triângulos é sempre possível de ser feita e o problema de encontrar uma partição é conhecido como o *problema da triangulação de polígonos*. Uma descrição de uma solução algorítmica bastante elegante para o problema pode ser encontrada no livro de O’Rourke [97]. O fato dos polígonos da subdivisão de Delaunay de  $P$  serem convexos, no entanto, torna o problema da triangulação quase trivial (ver Problema 4.3). Por outro lado, há, em geral, mais de uma maneira de particionarmos um polígono (convexo ou não) em triângulos. Por isso, a triangulação de  $P$  obtida a partir de uma subdivisão de Delaunay de  $P$  não é única. O Problema 4.11 pede ao leitor para demonstrar que a projeção ortogonal de  $\mathcal{E}(FC(P^\omega))$  em  $\mathbb{E}^2$  é um complexo politopal.

Como apenas a condição (2) é necessária para se obter uma única triangulação de  $P$  a partir da projeção ortogonal de  $\mathcal{E}(FC(P^\omega))$  em  $\mathbb{E}^2$ , chamaremos a triangulação obtida de triangulação de Delaunay de  $P$  também. Quando a condição (2) for violada, chamaremos a projeção ortogonal de  $\mathcal{E}(FC(P^\omega))$  em  $\mathbb{E}^2$  de subdivisão de Delaunay de  $P$  e, *de forma propositalmente equivocada*, também chamaremos a triangulação de  $P$  obtida a partir da subdivisão de Delaunay de  $P$  de triangulação de Delaunay de  $P$ . Este “equivoco” ocorre com frequência na literatura especializada [33], mas deixamos o leitor ciente do fato.

## 4.5 Propriedades

A triangulação de Delaunay,  $\mathcal{TD}(P)$ , de um conjunto finito de pontos,  $P$ , de  $\mathbb{E}^2$  possui muitas propriedades matemáticas, algumas das quais são cruciais para que possamos desenvolver um algoritmo eficiente para *calcular* a triangulação a partir do conjunto  $P$ . Nesta





seção, estudamos as principais propriedades da triangulação de Delaunay. Durante toda a seção, e a menos que façamos uma eventual exceção, assumimos que  $P$  satisfaz as condições (1), (2) e (3) do Teorema 4.6.

A demonstração de algumas propriedades estudadas nesta seção está calcada em resultados básicos da geometria euclidiana plana enunciados abaixo e demonstrados em textos básicos de geometria [8, 53, 96]. Lembre-se de que denominamos por *ângulo inscrito* em uma circunferência o ângulo cujo vértice está sobre a circunferência e cujos lados passam, cada um, por pontos sobre a circunferência (ver Figura 4.13).

**Proposição 4.7.** *Todo ângulo inscrito em uma circunferência tem a metade da medida do ângulo central correspondente ao arco determinado pelos pontos da circunferência pelos quais passam os lados do ângulo.*

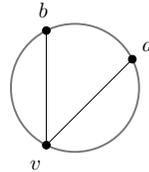


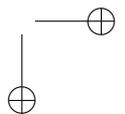
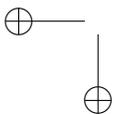
Figura 4.13: O ângulo  $\angle avb$  inscrito na circunferência por  $a$ ,  $b$  e  $v$ .

**Corolário 4.8.** *Todos os ângulos inscritos em uma mesma circunferência e que correspondem exatamente a um mesmo arco possuem a mesma medida. Em particular, todos os ângulos inscritos na circunferência que correspondem a uma semicircunferência são ângulos retos.*

A Figura 4.14 ilustra a Proposição 4.7 e o Corolário 4.8.

A Proposição 4.7 e o Corolário 4.8 nos permitem mostrar o seguinte:

**Corolário 4.9.** *Seja  $C$  uma circunferência em  $\mathbb{E}^2$ . Seja  $\ell$  uma reta orientada que intersecta  $C$  em dois pontos,  $a$  e  $b$ , com  $a, b \in \mathbb{E}^2$  e*



$a \neq b$ . Sejam  $p, q, r$  e  $s$  pontos de  $\mathbb{E}^2$  do mesmo lado de  $\ell$ . Suponha que os pontos  $p$  e  $q$  pertençam a  $C$ , que  $r$  resida no interior de  $C$  e que  $s$  pertença ao exterior de  $C$ . Então, é verdade que

$$\angle arb > \angle apb = \angle aqb > \angle asb.$$

*Demonstração.* A Proposição 4.7 nos diz que  $\angle apb = \angle aqb$ . Como  $r$  pertence ao interior de  $C$ , considere a reta  $\ell'$  por  $a$  e  $r$  e refira-se à Figura 4.15. A reta  $\ell'$  intersecta  $C$  em dois pontos. Um deles é  $a$ . Denomine o outro ponto de  $r'$ . Observe que  $a$  e  $r'$  residem em lados opostos da reta  $\ell''$  orientada de  $b$  para  $r$ . Observe também que  $\angle ar'b + \angle brr' + \angle rbr' = \pi$  e que  $\angle brr' = \pi - \angle arb$ . Daí, temos:

$$\angle ar'b + \angle rbr' = \angle arb.$$

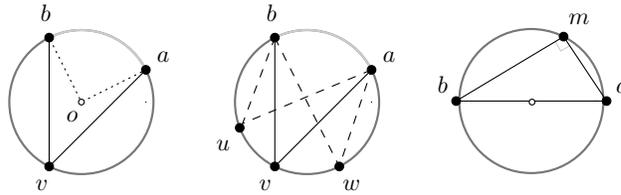


Figura 4.14:  $\angle avb = \frac{1}{2}\angle aob$ ,  $\angle avb = \angle aub$  e  $\angle amb = \frac{\pi}{2}$ .

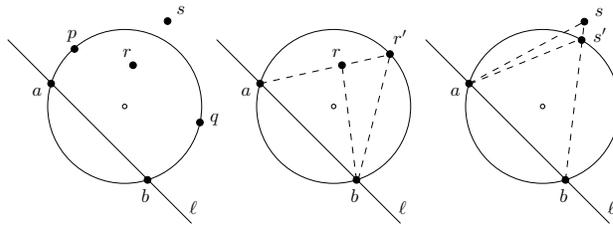
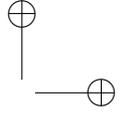
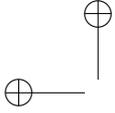


Figura 4.15: Ilustração do argumento da prova do Corolário 4.9.

Como  $\angle rbr' > 0$ , pois, do contrário, teríamos  $r = r'$ , concluímos que

$$\angle ar'b < \angle arb.$$



De forma análoga, considere o segmento de reta  $\overline{bs}$ . Este segmento intersecta  $C$  nos pontos  $b$  e  $s'$ . Observe que os pontos  $b$  e  $s$  residem em lados opostos da reta  $\ell'''$  orientada de  $a$  para  $s'$ . Observe também que  $\angle as's + \angle asb + \angle sas' = \pi$  e que  $\angle as's = \pi - \angle as'b$ . Daí, temos:

$$\angle asb + \angle sas' = \angle as'b.$$

Mas, sabemos que  $\angle sas' > 0$ , pois, do contrário, teríamos  $s = s'$ . Além disso, do Corolário 4.8, sabemos também que  $\angle as'b = \angle apb$ . Logo,

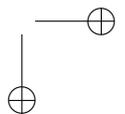
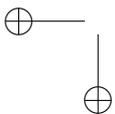
$$\angle asb < \angle apb.$$

□

O lema a seguir nos permite verificar se uma dada triangulação,  $\mathcal{T}(P)$ , de  $P$  é a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ . A prova que apresentamos para o lema se vale de um argumento usado na prova do Teorema 4.6 e dos resultados da geometria euclidiana plana vistos antes.

**Lema 4.10.** *Seja  $\mathcal{T}(P)$  uma triangulação de um conjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$ . Então,  $\mathcal{T}(P)$  é a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  se, e somente se, para cada par de triângulos de  $\mathcal{T}(P)$  incidentes em uma mesma aresta de  $\mathcal{T}(P)$ , a soma dos ângulos internos dos triângulos opostos à aresta comum é menor do que  $\pi$ .*

*Demonstração.* Sejam  $\sigma = [a, b, c]$  e  $\tau = [a, b, d]$  dois triângulos de  $\mathcal{T}(P)$  que compartilham a aresta  $[a, b]$  de  $\mathcal{T}(P)$ , em que  $a, b, c$  e  $d$  são todos vértices de  $\mathcal{T}(P)$  em  $P$ . Seja  $C_\sigma$  o círculo que contém  $a, b$  e  $c$  em sua circunferência e seja  $C_\tau$  o círculo que contém  $a, b$  e  $d$  em sua circunferência; ou seja,  $C_\sigma$  e  $C_\tau$  são os círculos circunscritos aos triângulos  $\sigma$  e  $\tau$ , respectivamente (ver Figura 4.16). Seja  $\ell_{ab}$  a reta por  $a$  e  $b$  orientada de  $a$  para  $b$ . Observe que os pontos  $c$  e  $d$  estão em lados opostos de  $\ell_{ab}$ , pois, do contrário, a interseção de  $\sigma$  e  $\tau$  conteria pontos do interior de ambos e, conseqüentemente, eles não poderiam fazer parte de uma triangulação. Então, suponha que  $C \in \text{int}(H_+(f_{ab}))$  e  $D \in \text{int}(H_-(f_{ab}))$ , em que  $f_{ab} : \mathbb{E}^2 \rightarrow \mathbb{R}$  é qualquer forma afim tal que  $\text{Nuc}(f_{ab}) = \ell_{ab}$ . Com relação a  $C_\sigma$  e  $d$ , há três possíveis cenários mutuamente exclusivos: (a)  $d \in \partial(C_\sigma)$ , (b)  $d \in$



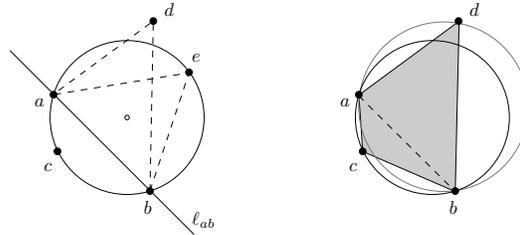
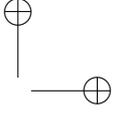
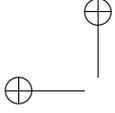


Figura 4.16: Ilustração do argumento da prova do Lema 4.10.

Afirmamos que (c) ocorre se, e somente se, a soma dos ângulos  $\angle acb$  e  $\angle adb$  é menor do que  $\pi$ . De fato, se  $e$  é qualquer ponto sobre o arco  $\widehat{ab}$  de  $\partial(C_\sigma)$  que não contém o ponto  $c$ , então  $\angle acb + \angle aeb = \pi$ , pois a soma dos comprimentos dos arcos correspondentes a  $\angle acb$  e  $\angle aeb$  é  $2\pi$  e cada um desses ângulos mede a metade do comprimento do arco correspondente (ver Proposição 4.7). Por sua vez, o Corolário 4.8 nos diz que  $\angle aeb > \angle adb$  se, e somente se,  $d$  pertence a  $ext(C_\sigma)$ . Logo, temos que  $\angle acb + \angle adb < \pi$  se, e somente se, o ponto  $d$  pertence a  $ext(C_\sigma)$ . Um argumento análogo pode ser usado para concluir que  $\angle acb + \angle adb < \pi$  se, e somente se, o ponto  $C$  pertence ao exterior de  $C_\tau$ . Consequentemente, se, para *todo* par de triângulos de  $\mathcal{T}(P)$  incidentes em uma mesma aresta de  $\mathcal{T}(P)$ , a soma dos ângulos internos dos triângulos opostos à aresta comum é menor do que  $\pi$ , então podemos concluir que não pode haver nenhum ponto de  $P$  no interior da união dos círculos  $C_\sigma$  e  $C_\tau$ . Além disso, os únicos pontos de  $P$  pertencentes a  $\partial(C_\sigma)$  são  $a$ ,  $b$  e  $c$  e os únicos pontos de  $P$  pertencentes à circunferência de  $C_\tau$  são  $a$ ,  $b$  e  $d$ . Logo, pela Proposição 4.5, os subconjuntos  $\{a^\omega, b^\omega, c^\omega\}$  e  $\{a^\omega, b^\omega, d^\omega\}$  de  $\mathbb{E}^3$  são vértices de duas facetas,  $F$  e  $G$ , respectivamente, do envelope inferior,  $\mathcal{E}(P)$ , do politopo  $\mathcal{P} = FC(P^\omega)$ . A aresta  $\overline{a^\omega b^\omega}$ , comum a  $F$  e  $G$  também pertence a  $\mathcal{E}(P)$ . De fato, como  $F, G \in \mathcal{E}(P)$ , existem formas afins,

$$f : \mathbb{E}^3 \rightarrow \mathbb{R} \quad \text{e} \quad g : \mathbb{E}^3 \rightarrow \mathbb{R},$$



tais que

$$F = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid f(x) = 0\} \quad \text{e} \quad G = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid g(x) = 0\},$$

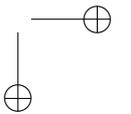
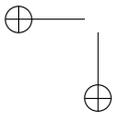
$$f(y), g(y) \leq 0, \quad \forall y \in \mathcal{P}$$

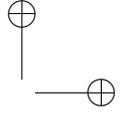
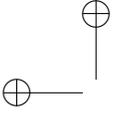
e

$$f(O + \alpha \cdot \mathbf{k}), g(O + \alpha \cdot \mathbf{k}) > 0$$

para todo  $\alpha \in [\beta, -\infty)$  e para alguma constante real  $\beta$ , com  $\beta < 0$ . Seja  $h : \mathbb{E}^3 \rightarrow \mathbb{R}$  a forma afim definida por  $h(x) = f(x) + g(x)$ , para todo  $x \in \mathbb{E}^3$ . Então, temos que  $\overline{a^\omega b^\omega} = \mathcal{P} \cap \{x \in \mathbb{E}^3 \mid h(x) = 0\}$ ,  $h(y) \leq 0$ , para todo  $y \in \mathcal{P}$ , e  $h(O + \alpha \cdot \mathbf{k}) > 0$  para todo  $\alpha \in [\beta, -\infty)$ . Logo, a aresta  $\overline{a^\omega b^\omega}$  pertence a  $\mathcal{E}(P)$ . Pelo que acabamos de provar, tanto a aresta  $[a, b]$  quanto os triângulos  $\sigma$  e  $\tau$  de  $\mathcal{T}(P)$  pertencem à triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ . Logo, toda aresta interna de  $\mathcal{T}(P)$  e todo triângulo de  $\mathcal{T}(P)$  que compartilha uma aresta com outro triângulo de  $\mathcal{T}(P)$  pertencem a  $\mathcal{TD}(P)$ . Mas, todo triângulo de  $\mathcal{T}(P)$  tem de compartilhar pelo menos uma aresta com outro triângulo de  $\mathcal{T}(P)$ ; caso contrário,  $\mathcal{T}(P)$  teria apenas um triângulo, o que viola a condição (1) imposta sobre  $P$ . Além disso, pela definição de triangulação, todo ponto de  $P$  é um vértice de  $\mathcal{T}(P)$ . Por outro lado, se houvesse um triângulo (resp. aresta interior) de  $\mathcal{TD}(P)$  que não estivesse em  $\mathcal{T}(P)$ , este triângulo (resp. aresta interior) teria de intersectar o interior de um triângulo (ou aresta) de  $\mathcal{T}(P)$ , pois  $|\mathcal{T}(P)| = FC(P) = |\mathcal{TD}(P)|$ . Mas, neste caso, esta interseção também ocorreria em  $\mathcal{E}(P)$ , o que não é possível (Proposição 4.4). Logo, podemos concluir que  $\mathcal{T}(P)$  e  $\mathcal{TD}(P)$  são iguais. Reciprocamente, se  $\mathcal{T}(P)$  e  $\mathcal{TD}(P)$  são iguais, então não há nenhum ponto de  $P$  em  $\text{int}(C_\sigma)$  e  $\text{int}(C_\tau)$ . Isto implica que o ponto  $d$  pertence a  $\text{ext}(C_\sigma)$  e o ponto  $c$ , a  $\text{ext}(C_\tau)$ . Daí, temos que  $\angle acb + \angle adb < \pi$ .  $\square$

**Definição 4.4.** *Uma aresta,  $[a, b]$ , em uma triangulação,  $\mathcal{T}(P)$ , de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  é denominada localmente Delaunay se, e somente se, a aresta  $[a, b]$  incide em apenas um triângulo de  $\mathcal{T}(P)$  ou  $[a, b]$  incide sobre dois triângulos de  $\mathcal{T}(P)$  cuja soma das medidas dos ângulos opostos a  $[a, b]$  é menor do que  $\pi$ .*





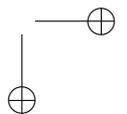
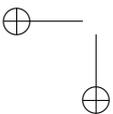
**Observação 4.5.** *Pelo Lema 4.10 e pela Definição 4.4, podemos concluir que uma triangulação,  $\mathcal{T}(P)$ , de  $P$  é a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  se, e somente se, toda aresta de  $\mathcal{T}(P)$  é localmente Delaunay.*

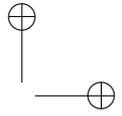
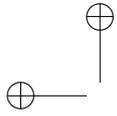
Note que podemos decidir se uma aresta,  $[a, b]$ , de  $\mathcal{T}(P)$ , compartilhada por dois triângulos de  $\mathcal{T}(P)$ , é localmente Delaunay calculando os ângulos  $\angle acb$  e  $\angle adb$  dos triângulos  $[a, b, c]$  e  $[a, b, d]$  incidentes sobre  $[a, b]$  e verificando se  $\angle acb + \angle adb < \pi$ . No entanto, por razões que se tornarão evidentes mais adiante, realizamos esta verificação utilizando o predicado *incircle* discutido no Apêndice A. O lema a seguir estabelece o resultado que nos permite utilizar o predicado:

**Lema 4.11.** *Seja  $\mathcal{T}(P)$  qualquer triangulação de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$ . Sejam  $\sigma = [a, b, c]$  e  $\tau = [a, b, d]$  dois triângulos de  $\mathcal{T}(P)$  incidentes sobre a aresta  $[a, b]$  de  $\mathcal{T}(P)$ , em que  $a, b, c$  e  $d$  são pontos em  $P$ . Seja  $C_\sigma$  o círculo circunscrito a  $\sigma$ . Então, a aresta  $[a, b]$  é localmente Delaunay se, e somente se, o ponto  $d$  pertence ao exterior de  $C_\sigma$ . Além disso, se  $a, b, c$  e  $d$  de  $P$  são vértices de um quadrilátero convexo, mas não pertencem a uma mesma circunferência, então apenas uma de  $[a, b]$  e  $[c, d]$  é localmente Delaunay.*

*Demonstração.* A primeira afirmação segue da prova dada para o Lema 4.10. Então, vamos provar a segunda afirmação. Para tal, assumamos que os pontos  $a, b, c$  e  $d$  de  $P$  definem os vértices de um quadrilátero convexo,  $\square(acbd)$ , mas não pertencem a uma mesma circunferência em  $\mathbb{E}^2$  (ver Figura 4.17). Então, há dois possíveis cenários: (a) o ponto  $d$  pertence ao interior do círculo cuja circunferência passa por  $a, b$  e  $c$  e (b) o ponto  $d$  pertence ao exterior do círculo cuja circunferência passa por  $a, b$  e  $c$ . Do Corolário 4.9, podemos concluir que  $\angle acb + \angle adb > \pi$  e  $\angle acb + \angle adb < \pi$  nos cenários (a) e (b), respectivamente. Logo, a aresta  $[a, b]$  não é localmente Delaunay em (a) e é localmente Delaunay em (b). O fato de  $\square(acbd)$  ser convexo implica que o segmento de reta  $\overline{cd}$  está inteiramente contido nele, ou seja, que  $\overline{cd}$  é uma diagonal do quadrilátero  $\square(acbd)$ . Como

$$\angle acb + \angle adb + \angle cad + \angle cbd = 2\pi,$$





podemos concluir que  $[c, d]$  *não* é localmente Delaunay em (a) e  $[c, d]$  é localmente Delaunay em (b). Portanto, no cenário (a), temos que a aresta  $[a, b]$  é localmente Delaunay, mas a aresta  $[c, d]$  não é e, no cenário (b), temos que  $[a, b]$  não é localmente Delaunay, mas  $[c, d]$  é.  $\square$

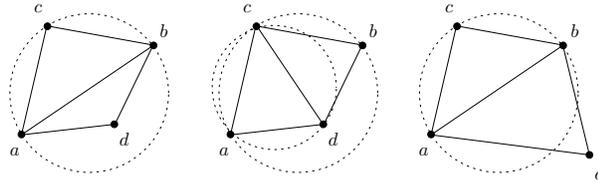
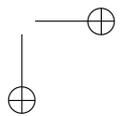
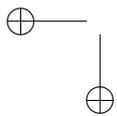
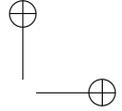
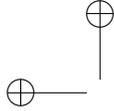


Figura 4.17: Ilustração do argumento da prova do Lema 4.11.

O Lema 4.11 sugere um algoritmo para calcular a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  a partir de *qualquer* triangulação,  $\mathcal{T}(P)$ , de  $P$ . A ideia por trás do algoritmo é bastante simples: consideramos uma aresta por vez de  $\mathcal{T}(P)$ . Se a aresta não for localmente Delaunay, nós a “trocamos” por outra que é garantida ser localmente Delaunay. O resultado é uma outra triangulação de  $P$ . Repetimos este passo até que todas as arestas da triangulação atual sejam localmente Delaunay. Este algoritmo suscita, obviamente, três questões importantes: qual é a aresta que substitui uma aresta que não é localmente Delaunay? Por que sempre podemos encontrar uma aresta localmente Delaunay para substituir uma aresta que não é localmente Delaunay? Será que este algoritmo sempre termina? Em outras palavras, é sempre possível obter uma triangulação em que todas as arestas são localmente Delaunay realizando um número finito de trocas de arestas?

Seja  $[a, b]$  uma aresta de  $\mathcal{T}(P)$  incidente nos triângulos  $[a, b, c]$  e  $[a, b, d]$  de  $\mathcal{T}(P)$ . Da prova do Lema 4.11, podemos concluir que o quadrilátero  $\square(acbd)$  é convexo sempre que  $[a, b]$  *não* é localmente Delaunay. De fato, se  $\square(acbd)$  não fosse convexo, então o ponto  $d$  não poderia pertencer ao círculo cuja circunferência passa pelos pontos  $a$ ,  $b$  e  $c$  (Figura 4.18). Mas, se  $d$  não pertence a este círculo, então a





aresta  $[a, b]$  é localmente Delaunay, o que contradiz nossa hipótese. Como não há, por hipótese, quatro pontos de  $P$  sobre uma mesma circunferência (ver Observação 4.1), o fato de  $\square(acbd)$  ser convexo e o Lema 4.11 implicam que  $[c, d]$  é localmente Delaunay. Por sua vez, a convexidade de  $\square(acbd)$  nos permite realizar uma troca de aresta: substituímos  $[a, b]$  por  $[c, d]$  em  $\mathcal{T}(P)$  e obtemos uma nova triangulação de  $P$ . Logo, sempre é possível substituir uma aresta que não é localmente Delaunay, em qualquer triangulação de  $P$ , por uma que é.

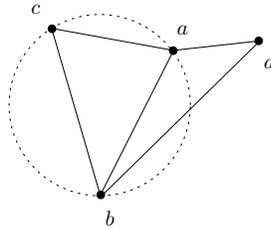
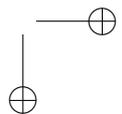
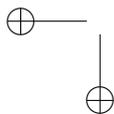


Figura 4.18: O quadrilátero  $\square(acbd)$  não é convexo e possui um ângulo reflexo com vértice em  $a$ . A aresta  $[a, b]$  é localmente Delaunay.

O algoritmo `GERADELAUNAY()` (ver Algoritmo 4.1) testa cada aresta  $[a, b]$  de uma triangulação  $\mathcal{T}(P)$  até encontrar uma aresta que não é localmente Delaunay. Sempre que  $[a, b]$  não for localmente Delaunay, o algoritmo a substitui em  $\mathcal{T}(P)$  pela aresta  $[c, d]$ , em que  $[a, b, c]$  e  $[a, b, d]$  são os dois triângulos de  $\mathcal{T}(P)$  incidentes sobre  $[a, b]$ . Cada troca de aresta gera uma nova triangulação de  $P$ , que o algoritmo continua a denotar por  $\mathcal{T}(P)$ . `GERADELAUNAY()` realiza trocas de arestas até obter uma triangulação em que todas as arestas são localmente Delaunay. O Lema 4.12 garante que o algoritmo termina.

**Lema 4.12.** `GERADELAUNAY()` sempre termina e devolve  $\mathcal{TD}(P)$ .

*Demonstração.* Considere a aresta  $[a, b]$  e os triângulos  $[a, b, c]$  e  $[a, b, d]$  de  $\mathcal{T}(P)$ . Denote os ângulos  $\angle bac$ ,  $\angle bad$ ,  $\angle abc$  e  $\angle abd$  por  $\alpha$ ,  $\beta$ ,  $\gamma$  e  $\rho$ , respectivamente, e refira-se à Figura 4.19. Também denote os



ângulos  $\angle cdb$ ,  $\angle dcb$ ,  $\angle cda$  e  $\angle acd$  por  $\alpha'$ ,  $\beta'$ ,  $\gamma'$  e  $\rho'$ , respectivamente. Do Corolário 4.9, sabemos que  $\alpha < \alpha'$ ,  $\beta < \beta'$  e  $\gamma < \gamma'$ . Além disso,  $\rho + \alpha' = \alpha + \rho'$ , o que implica que  $\rho = \rho' + (\alpha - \alpha') < \rho'$ , pois  $\alpha < \alpha'$ . Essas desigualdades nos dizem que

$$\begin{aligned} \min\{\alpha, \beta, \gamma, \rho, \rho' + \beta', \gamma' + \alpha'\} &= \min\{\alpha, \beta, \gamma, \rho\} \\ &< \min\{\alpha', \beta', \gamma', \rho', \alpha + \beta, \gamma + \beta\}. \end{aligned}$$

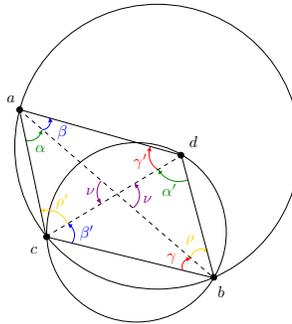
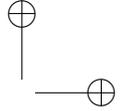
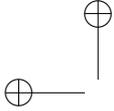


Figura 4.19: Ilustração do argumento da prova do Lema 4.12.

Em outras palavras, cada troca de aresta do algoritmo gera dois triângulos cujo menor ângulo é maior do que o menor ângulo dos dois triângulos removidos. Seja  $m$  o número de triângulos de  $\mathcal{T}(P)$ . Observe que a operação de troca sempre gera uma triangulação com o mesmo número de triângulos da triangulação anterior. Logo, toda triangulação obtida durante a execução de `GERADELAUNAY()` possui exatamente  $m$  triângulos. Seja  $\mathcal{T}$  qualquer triangulação de  $P$  com  $m$  triângulos. Denote por  $A(\mathcal{T})$  qualquer seqüência,  $(\alpha_1, \dots, \alpha_{3m})$ , não decrescente dos ângulos de todos os triângulos de  $\mathcal{T}$ . Se  $\mathcal{T}'$  é qualquer outra triangulação de  $P$  com  $m$  triângulos, então dizemos que  $A(\mathcal{T}') = (\alpha'_1, \dots, \alpha'_{3m})$  é maior do que  $A(\mathcal{T})$  se, e somente se, existir um índice  $i$ , com  $1 \leq i \leq 3m$ , tal que  $\alpha_j = \alpha'_j$  e  $\alpha_i < \alpha'_i$ , para todo  $j = 1, \dots, i - 1$ . Agora, observe que cada operação de troca executada por `GERADELAUNAY()` gera uma triangulação cujo vetor ângulo é maior do que o da triangulação anterior. Como há um número finito



de vértices em  $P$ , há um número finito de triangulações de  $P$  (ver Problema 4.12). Logo, se o algoritmo não terminasse, geraríamos uma sequência infinita de triangulações em que toda triangulação possui um vetor ângulo maior do que o de todas as triangulações anteriores. Mas, isto implica que teríamos uma sequência infinita de triangulações distintas de  $P$ , o que não é possível. Logo, o algoritmo tem de terminar. Por outro lado, o término do algoritmo só pode ocorrer quando a condição do laço **enquanto** da linha 1 for falsa, o que implica que todas as arestas da última triangulação obtida são localmente Delaunay. Logo,  $\mathcal{T}(P) = \mathcal{TD}(P)$  quando o laço **enquanto** termina.  $\square$

---

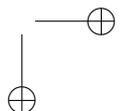
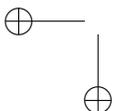
**Algoritmo 4.1** GERADELAUNAY( $\mathcal{T}(P)$ )

---

- 1: **enquanto**  $[a, b] \in \mathcal{T}(P)$  não é localmente Delaunay **faça**
  - 2:    remova  $[a, b]$ ,  $[a, b, c]$  e  $[a, b, d]$  de  $\mathcal{T}(P)$
  - 3:    insira  $[c, d]$ ,  $[a, c, d]$  e  $[b, c, d]$  em  $\mathcal{T}(P)$
  - 4: **fim enquanto**
  - 5: **devolva**  $\mathcal{T}(P)$
- 

Para que possamos utilizar o Algoritmo 4.1, temos de dispor de *alguma* triangulação,  $\mathcal{T}(P)$ , de  $P$ . Felizmente, há uma maneira simples de gerarmos tal triangulação. Para tal, basta fazermos uma pequena modificação no Algoritmo 3.1. A Figura 4.20 exemplifica a execução do algoritmo modificado. Antes do laço **enquanto** do Algoritmo 3.1 ser iniciado, criamos o triângulo de vértices  $p_k$ ,  $q_1$  e  $q_2$ . Quando o vértice  $q_i$  é considerado pela *primeira* vez no laço **enquanto**, adicionamos as arestas  $[q_i, q_0]$  e  $[q_i, b]$  e o triângulo  $[q_0, b, q_i]$  à triangulação. Se  $q_i$  está sendo considerado pela segunda vez em diante, então adicionamos as arestas  $[q_i, b]$  e  $[q_i, c]$  e o triângulo  $[q_i, b, c]$  à triangulação, em que  $c$  é o vértice  $b$  da iteração anterior. Isto é, precisamos guardar, em uma nova variável,  $c$ , o vértice  $b$  da iteração anterior sempre que  $\text{LEFT}(a, b, q_i)$ , na linha 10 do Algoritmo 3.1, avalia para o valor lógico falso. O Problema 4.31 pede ao leitor o pseudocódigo deste algoritmo.

Embora o Algoritmo 4.1 nos permita calcular  $\mathcal{TD}(P)$ , ele não consiste no algoritmo mais eficiente que dispomos para realizar tal tarefa.



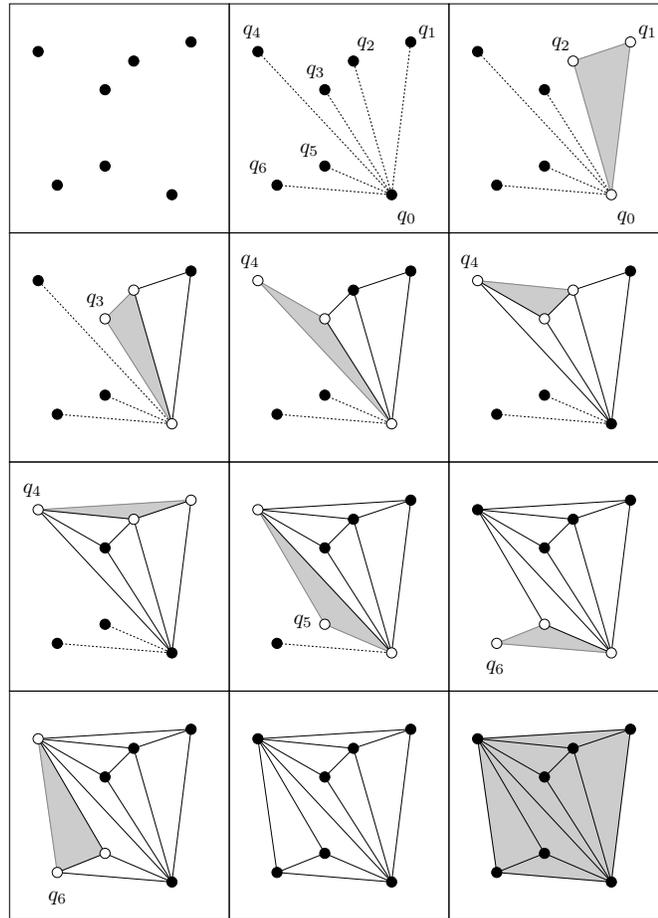


Figura 4.20: Ilustração da execução do Graham *scan* modificado.

Por outro lado, a operação “troca de aresta” é crucial para o desenvolvimento do algoritmo mais eficiente dado na Seção 4.6. Antes de apresentarmos o algoritmo eficiente, vamos analisar a complexidade do Algoritmo 4.1. Considere o conjunto  $P$  consistindo de  $2n$  pontos

de  $\mathbb{E}^2$ :

$$P = \{e_1, \dots, e_n\} \cup \{d_1, \dots, d_n\},$$

tal que os pontos em  $\{e_1, \dots, e_n\}$  estão espaçados uniformemente no segmento de reta  $[-1, 0] \times [-1, 1] \subset \mathbb{E}^2$  e numerados de cima para baixo, enquanto os pontos  $\{d_1, \dots, d_n\}$  estão espaçados uniformemente no segmento de reta  $[1, -1] \times [1, 0] \subset \mathbb{E}^2$  e numerados de baixo para cima (Figura 4.21). Esses dois subconjuntos de  $P$  gozam da seguinte propriedade [48]: para  $i, j \in \{1, \dots, n\}$ , se  $i < j$  então a circunferência  $C_{e_i, e_j, d_k}$ , que passa por  $e_i, e_j$  e  $d_k$ , contém *exatamente* os pontos  $e_l$ , para todo  $l$ , com  $i \leq l \leq j$ , e os pontos  $d_h$ , para todo  $h \geq k$ .

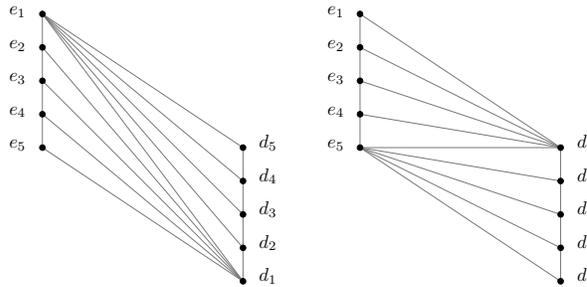
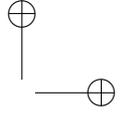
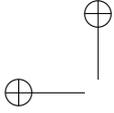


Figura 4.21: Uma triangulação de um subconjunto de 10 pontos de  $\mathbb{E}^2$  (esquerda). A triangulação de Delaunay do mesmo conjunto (direita).

Seja  $\mathcal{T}(P)$  a triangulação de  $P$  consistindo das arestas do fecho convexo,  $FC(P)$ , de  $P$  e das arestas  $[e_i, d_1]$  e  $[e_1, d_i]$ , para todo  $i = 1, \dots, n$ . Os triângulos da forma  $[e_i, e_{i+1}, d_n]$  e  $[d_i, d_{i+1}, e_n]$  estão circunscritos por círculos cujos interiores não possuem nenhum ponto de  $P$ . Logo, a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  consiste das arestas de  $FC(P)$  e das arestas  $[e_i, d_n]$  e  $[e_n, d_i]$ , para todo  $i = 1, \dots, n$ . Nós podemos mostrar que são necessárias  $(n - 1)^2$  trocas de arestas para transformar  $\mathcal{T}(P)$  em  $\mathcal{TD}(P)$ . Para tal, afirmamos que a única troca de aresta possível é a que substitui  $[e_i, d_j]$  por  $[e_{i+1}, d_{j+1}]$  como diagonal do quadrilátero  $\square(e_i, e_{i+1}, d_j, d_{j+1})$ . Esta afirmação pode ser provada por indução usando a hipótese que todas as arestas de  $\mathcal{T}(P)$ , que não pertencem ao fecho de  $P$ , conectam



necessariamente um ponto em  $\{e_1, \dots, e_n\}$  a outro em  $\{d_1, \dots, d_n\}$ . Seja  $\mathcal{T}$  o conjunto união de  $\{\mathcal{T}(P)\}$  com o conjunto de todas as triangulações de  $P$  obtidas por trocas de arestas durante a transformação de  $\mathcal{T}(P)$  em  $\mathcal{TD}(P)$ . Defina a função  $\Phi : \mathcal{T} \rightarrow \mathbb{R}$  tal que, para toda  $T \in \mathcal{T}$ ,

$$\Phi(T) = \sum_{[e_i, d_j] \in T} (i + j).$$

Observe que

$$\begin{aligned} \Phi(\mathcal{TD}(P)) - \Phi(\mathcal{T}(P)) &= 2 \left( \sum_{h=1}^n (h + n) \right) - 2n \\ &\quad - 2 \left( \sum_{h=1}^n (h + 1) \right) - 2 \\ &= 2 \cdot (n - 1)^2. \end{aligned}$$

Como o valor de  $\Phi(T)$  aumenta em duas unidades a cada troca de aresta, podemos concluir que são necessárias  $(n - 1)^2$  trocas de arestas para transformar  $\mathcal{T}(P)$  na triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ .

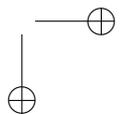
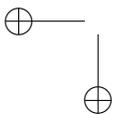
Por outro lado, temos a seguinte cota superior no número de trocas:

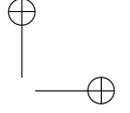
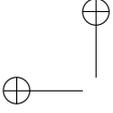
**Lema 4.13.** *O Algoritmo 4.1 faz, no máximo,  $\binom{|P|}{2}$  trocas de arestas.*

*Demonstração.* Lembre-se de que Algoritmo 4.1 troca uma aresta  $\sigma$  por uma aresta  $\tau$  se, e somente se, a aresta  $\sigma$  não é localmente Delaunay na triangulação modificada. Além disso, Lema 4.11 nos garante que  $\tau$  é localmente Delaunay na triangulação modificada. Vamos provar agora que uma vez que  $\sigma$  seja substituída por  $\tau$ , a aresta  $\sigma$  jamais retorna a uma triangulação nas trocas futuras. A partir daí, podemos concluir que o número de trocas é limitado superiormente pelo número máximo de arestas definidas por pares de pontos em  $P$ ; isto é,

$$\binom{|P|}{2}.$$

Para mostrar que  $\sigma$  jamais retorna a uma triangulação nas trocas futuras realizadas por Algoritmo 4.1, voltamos a fazer uso do importante mapa de elevação  $\omega$  da Seção 4.3. A ideia é examinar o que





ocorre com a “imagem” da triangulação de um quadrilátero,  $\square(acbd)$ , convexo por  $\omega$  quando o Algoritmo 4.1 troca uma aresta correspondente a uma diagonal de  $\square(acbd)$ , digamos  $\sigma = [a, b]$ , pela aresta correspondente a outra diagonal, digamos  $\tau = [c, d]$ . Seja  $Q$  o conjunto

$$\{a, b, c, d\} \subset \mathbb{E}^2.$$

Se a aresta  $\sigma$  foi trocada pela aresta  $\tau$ , então a triangulação de Delaunay,  $\mathcal{TD}(Q)$ , de  $Q$  contém a aresta  $\tau$  e os triângulos  $\tau_1 = [a, d, c]$  e  $\tau_2 = [c, d, b]$ , mas não contém a aresta  $\sigma$  nem os triângulos  $\sigma_1 = [a, b, c]$  e  $\sigma_2 = [a, c, d]$ . Sabemos que  $T = FC(Q^\omega)$  é um tetraedro e que  $\mathcal{TD}(Q)$  corresponde à projeção ortogonal do envelope inferior,  $\mathcal{E}(T)$ , de  $T$  sobre  $\mathbb{E}^2$ . Seja  $\pi : \mathcal{E}(\mathcal{P}) \rightarrow \mathbb{E}^2$  a função projeção ortogonal de  $\mathbb{E}^3$  sobre  $\mathbb{E}^2$  restrita ao envelope inferior,  $\mathcal{E}(\mathcal{P})$ , de  $\mathcal{P}$ . Então,  $\pi^{-1}(\tau)$ ,  $\pi^{-1}(\tau_1)$  e  $\pi^{-1}(\tau_2)$  pertencem a  $\mathcal{E}(T)$ , mas  $\pi^{-1}(\sigma)$ ,  $\pi^{-1}(\sigma_1)$  e  $\pi^{-1}(\sigma_2)$  não. Daí, concluímos que existem duas formas afins, digamos

$$g_{\tau_1} : \mathbb{E}^3 \rightarrow \mathbb{R} \quad \text{e} \quad g_{\tau_2} : \mathbb{E}^3 \rightarrow \mathbb{R},$$

tais que

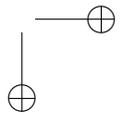
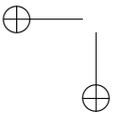
$$\begin{aligned} \pi^{-1}(\tau_1) &= T \cap \{x \in \mathbb{E}^3 \mid g_{\tau_1}(x) = 0\}, \\ \pi^{-1}(\tau_2) &= T \cap \{x \in \mathbb{E}^3 \mid g_{\tau_2}(x) = 0\}, \\ g_{\tau_1}(y), g_{\tau_2}(y) &\leq 0, \quad \forall y \in T, \end{aligned}$$

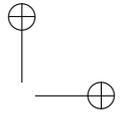
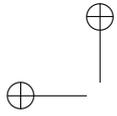
e

$$g_{\tau_1}(O + \alpha \cdot \mathbf{k}), g_{\tau_2}(O + \alpha \cdot \mathbf{k}) > 0,$$

para todo  $\alpha \in [\beta, \infty)$  e para alguma constante  $\beta \in \mathbb{R}$ , com  $\beta < 0$ . Em outras palavras, tanto o plano  $Nuc(g_{\tau_1})$  quanto o plano  $Nuc(g_{\tau_2})$  separam os pontos  $O + \alpha \cdot \mathbf{k}$  de qualquer ponto,  $z$ , interior à aresta  $\pi^{-1}(\sigma)$  ou às facetas  $\pi^{-1}(\tau_3)$  e  $\pi^{-1}(\tau_4)$  de  $T$ ; isto é,  $z \in H_-(g_{\tau_1}), H_-(g_{\tau_2})$  e  $\{O + \alpha \cdot \mathbf{k}\} \subset H_+(g_{\tau_2}), H_+(g_{\tau_2})$ . Intuitivamente, isto significa dizer que os pontos interiores da aresta  $\pi^{-1}(\sigma)$  e das facetas  $\pi^{-1}(\tau_3)$  e  $\pi^{-1}(\tau_4)$  de  $T$  estão “acima” dos planos  $Nuc(g_{\tau_1})$  e  $Nuc(g_{\tau_2})$  para um observador em  $O + \alpha \cdot \mathbf{k}$  de  $\mathbb{E}^3$  olhando na direção de  $\mathbf{k}$  (ver Figura 4.22).

Suponha que o Algoritmo 4.1 troque uma aresta, digamos  $\mu$ , por  $\sigma$  após uma eventual troca de  $\sigma$  por  $\tau$  durante a construção de  $\mathcal{TD}(P)$ . Sem perda de generalidade, assumamos que esta é a primeira vez em que





uma aresta (neste caso,  $\sigma$ ) “reaparece” durante a execução do Algoritmo 4.1. Como  $\square(acbd)$  é convexo, as arestas  $\sigma$  e  $\tau$  se intersectam em um ponto,  $m$ , de  $\mathbb{E}^2$ . Seja  $\Delta_1, \dots, \Delta_k$  a sequência tal que, para todo  $i = 1, \dots, k$ , cada  $\Delta_i$  representa um par de triângulos em que (1) pelo menos um deles contém  $m$ , (2) a aresta compartilhada por eles foi resultante de uma troca de aresta ocorrida após a troca de  $\sigma$  por  $\tau$  e antes da troca de  $\mu$  por  $\sigma$  e (3) os triângulos em  $\Delta_{j+1}$  surgiram a partir da troca de uma das arestas de um triângulo de  $\Delta_j$  contendo  $m$ , para todo  $j = 1, \dots, k-1$ . Note que  $\mu$  é a aresta comum aos dois triângulos em  $\Delta_k$ , pois  $m \in \sigma$  e, portanto,  $m$  pertence a um dos triângulos incidentes em  $\mu$ . Mas, esse é o último par de triângulos que contém  $m$  gerado pelo Algoritmo 4.1 antes da troca de  $\mu$  por  $\sigma$ .

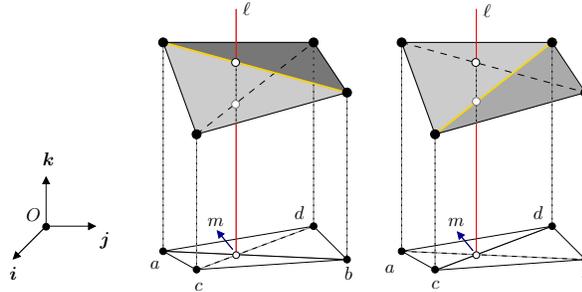


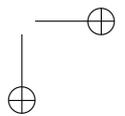
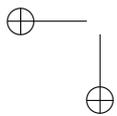
Figura 4.22: As projeções ortogonais sobre  $\mathbb{E}^2$  dos envelopes superior (esquerda) e inferior (direita) de um tetraedro. A triangulação de Delaunay dos vértices do quadrilátero  $\square(acbd)$  corresponde à segunda.

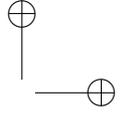
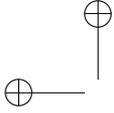
Seja

$$z = (1 - \gamma) \cdot a^\omega + \gamma \cdot c^\omega$$

um ponto interior de  $\pi^{-1}(\sigma)$ , em que  $\gamma \in (0, 1) \subset \mathbb{R}$  é tal que  $m = (1 - \gamma) \cdot a + \gamma \cdot c$ . Então, considere a reta  $\ell$  por  $z$  paralela a  $\mathbf{k}$ . Por construção,  $\ell$  intersecta  $\tau^* = \pi^{-1}(\tau)$  em um ponto  $z_0 \in \mathbb{E}^3$  e  $\rho_i^* = \pi^{-1}(\rho_i)$  em um ponto  $z_i \in \mathbb{E}^3$ , para todo  $i = 1, \dots, k$ , em que  $\rho_i$  é um dos dois triângulos de  $\Delta_i$ . Pela condição (3) acima, temos que

$$g_{\tau_1}(z), g_{\tau_2}(z) < 0 \quad \text{e} \quad g_{\tau_1}(z_0), g_{\tau_2}(z_0) = 0,$$





$$g_{\rho_i}(z_{i-1}) < 0 \quad \text{e} \quad g_{\rho_i}(z_i) = 0,$$

para todo  $i = 1, \dots, k$ , em que  $g_{\rho_i} : \mathbb{E}^3 \rightarrow \mathbb{R}$  é uma forma afim tal que

$$\rho_i^* = T_i \cap \{x \in \mathbb{E}^3 \mid g_{\rho_i}(x) = 0\},$$

$$g_{\rho_i}(y) \leq 0, \quad \forall y \in T_i$$

e

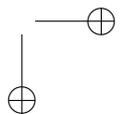
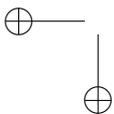
$$g_{\rho_i}(O + \alpha_i \cdot \mathbf{k}) > 0,$$

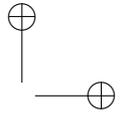
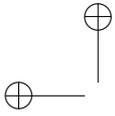
para todo  $\alpha_i \in [\beta_i, \infty)$  e para alguma constante  $\beta_i \in \mathbb{R}$ , com  $\beta_i < 0$ , em que  $T_i$  é o tetraedro definido pela imagem dos vértices dos triângulos em  $\Delta_i$  por  $\omega$ . Isto nos permite concluir que a sequência de pontos  $z, z_0, z_1, \dots, z_k$  está ordenada em ordem decrescente em relação à terceira coordenada dos pontos com relação ao referencial afim  $(O, \mathbf{i}, \mathbf{j}, \mathbf{k})$ . Mas, se  $\mu$  é trocada por  $\sigma$ , então também podemos concluir que a terceira coordenada de  $z$  é menor do que a de  $z_k$ : um absurdo! Logo, a aresta  $\sigma$  não pode reaparecer e nossa afirmação segue.  $\square$

Uma consequência natural do Lema 4.12 é que toda triangulação,  $\mathcal{T}(P)$ , de um subconjunto,  $P$ , de  $\mathbb{E}^2$ , satisfazendo as condições (1)-(3) do Teorema 4.6, pode ser convertida em uma triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  por meio *apenas* da execução de uma sequência de troca de aresta. Além disso, dadas quaisquer duas triangulações,  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$ , de  $P$ , podemos converter  $\mathcal{T}_1(P)$  em  $\mathcal{T}_2(P)$  ou vice-versa por meio de troca de aresta apenas. De fato, sejam  $S_1$  e  $S_2$  duas sequências de trocas de arestas que convertem  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$  em  $\mathcal{TD}(P)$ , respectivamente. Então, se concatenarmos  $S_1$  à sequência reversa de  $S_2$ , obteremos uma sequência que converte  $\mathcal{T}_1(P)$  em  $\mathcal{T}_2(P)$ . Reciprocamente, se concatenarmos  $S_2$  à sequência reversa de  $S_1$ , obteremos uma sequência que converte  $\mathcal{T}_2(P)$  em  $\mathcal{T}_1(P)$ . Daí, temos que:

**Corolário 4.14.** *Sejam  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$  quaisquer duas triangulações de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$ . Então, há uma sequência de troca de aresta que converte  $\mathcal{T}_1(P)$  em  $\mathcal{T}_2(P)$ .*

Há outras maneiras de se transformar uma triangulação  $\mathcal{T}_1(P)$  de  $P$  em outra, digamos  $\mathcal{T}_2(P)$ , por meio de trocas de arestas [33]. Este





fato nos leva naturalmente à seguinte questão: qual é o menor número de trocas de arestas necessárias para converter  $\mathcal{T}_1(P)$  em  $\mathcal{T}_2(P)$ ? Para responder esta pergunta, uma ferramenta conceitual foi definida: o *grafo de troca de aresta* (do inglês, *flip graph*). Neste grafo, todo nó representa uma única triangulação distinta de  $P$  e dois nós estão conectados por uma aresta do grafo se, e somente se, as triangulações representadas por eles podem ser convertidas uma na outra por meio de uma, e somente uma, troca de aresta. A Figura 4.23 exhibe o grafo de troca de aresta para um subconjunto particular de seis pontos de  $\mathbb{E}^2$ .

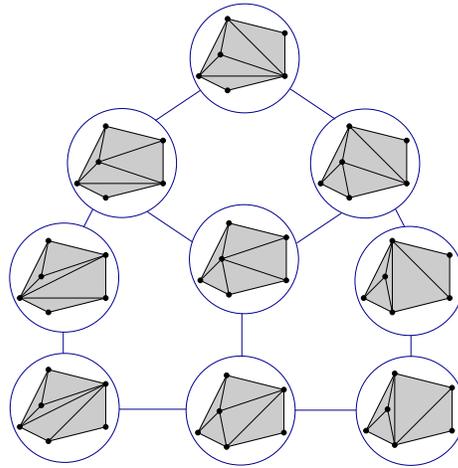
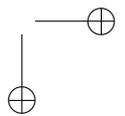
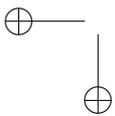
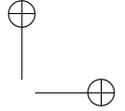
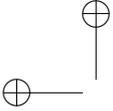


Figura 4.23: O grafo de trocas de arestas de um subconjunto de seis pontos de  $\mathbb{E}^2$ . Este exemplo foi retirado do livro de Satyan e O’Rourke [34].

O grafo de trocas de arestas de um subconjunto,  $P$ , finito e não vazio de  $\mathbb{E}^2$  é uma representação para o espaço de todas as triangulações de  $P$ . Qualquer caminho conectando dois nós do grafo representa uma sequência de trocas de arestas que converte a triangulação representada por um nós na triangulação representada pelo outro nó. Logo, o número mínimo de trocas de arestas necessário para converter uma





triangulação em outra é igual ao comprimento de um *caminho mais curto* no grafo de trocas de arestas conectando os nós correspondentes às duas triangulações. Uma discussão recente sobre resultados e propriedades <sup>2</sup> relacionados ao grafo de arestas pode ser encontrada em [16].

Para concluir esta seção, vamos estudar duas outras propriedades da triangulação de Delaunay. As duas estão relacionadas com aspectos práticos da construção da triangulação. O primeiro diz respeito à violação da condição (2) do Teorema 4.6. Isto é, vamos assumir que quatro pontos do conjunto  $P$  possam residir sobre uma mesma circunferência. O que ocorreria com o Algoritmo 4.1 se ele recebesse como entrada um tal conjunto  $P$ ? Convença-se de que o algoritmo não terminará! Por outro lado, se fizermos uma pequena modificação na definição de *aresta localmente Delaunay*, o mesmo algoritmo passará a funcionar e sempre gerará *uma* triangulação de Delaunay de  $P$  (na verdade, uma subdivisão de Delaunay). A *nova* definição é dada a seguir:

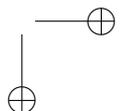
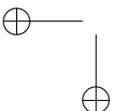
**Definição 4.5.** *Uma aresta,  $[a, b]$ , em uma triangulação,  $\mathcal{T}(P)$ , de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  é denominada localmente Delaunay se, e somente se, a aresta  $[a, b]$  incide em apenas um triângulo de  $\mathcal{T}(P)$  ou  $[a, b]$  incide sobre dois triângulos de  $\mathcal{T}(P)$  cuja soma das medidas dos ângulos opostos à aresta  $[a, b]$  não excede  $\pi$ .*

A Figura 4.24 ilustra a Definição 4.5. O Problema 4.42 pede para mostrar que o Algoritmo 4.1 produz uma triangulação de Delaunay de  $P$ .

Uma propriedade que nos será útil para provar a corretude do algoritmo que estudaremos na Seção 4.6 é a que caracteriza as arestas de uma triangulação de Delaunay de uma forma semelhante àquela dos triângulos de uma triangulação de Delaunay dada no Proposição 4.5.

**Proposição 4.15.** *Seja  $P$  um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  e sejam  $a$  e  $b$  dois pontos quaisquer de  $P$ . Então, a aresta,  $[a, b]$ , definida por  $a$  e  $b$ , pertence à triangulação de Delaunay,*

<sup>2</sup>Ver os exercícios 4.32-4.37 para conhecer algumas das propriedades.



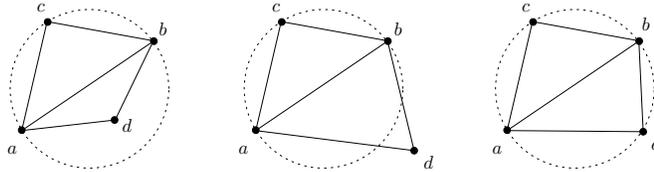


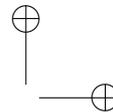
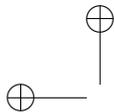
Figura 4.24: A aresta  $[a, b]$  não é localmente Delaunay na configuração mais à esquerda, mas é localmente Delaunay nas configurações central e direita pela nova definição de aresta localmente Delaunay.

$\mathcal{TD}(P)$  de  $P$ , se, e somente se,  $[a, b]$  é corda de um círculo,  $C$ , tal que

$$P \cap \text{int}(C) = \emptyset.$$

*Demonstração.* Suponha que  $[a, b]$  é uma aresta de  $\mathcal{TD}(P)$ . Desejamos mostrar que há um círculo,  $C$ , contendo  $a$  e  $b$  em sua circunferência e cujo interior,  $\text{int}(C)$ , não contém nenhum ponto de  $P$ . Podemos assumir que há, em  $\mathcal{TD}(P)$ , dois triângulos, digamos  $[a, b, c]$  e  $[a, b, d]$ , incidentes em  $[a, b]$ . Caso contrário, nossa afirmação seria trivialmente verdadeira, pois sempre há um triângulo incidente em  $[a, b]$  e se ele for o único, então podemos tomar  $C$  como sendo o círculo circunscrito a este triângulo. Sejam  $C_{abc}$  e  $C_{abd}$  os círculos circunscritos aos triângulos  $[a, b, c]$  e  $[a, b, d]$ , respectivamente. Denote por  $m_{ab}$  o ponto médio da aresta  $[a, b]$  (ver Figura 4.25). O ponto  $m_{ab}$  e os centros,  $q_c$  e  $q_d$ , de  $C_{abc}$  e  $C_{abd}$ , respectivamente, são colineares. Seja  $\ell_{cd}$  a reta orientada de  $q_c$  para  $q_d$  que passa por esses três pontos. O ponto  $q_c$  não pode estar à direita do ponto  $q_d$  em  $\ell_{cd}$  em relação à orientação de  $\ell_{cd}$ . Caso contrário, o ponto  $d$  pertenceria ao interior de  $C_{abc}$ , o que contradiz o fato de  $[a, b, d]$  pertencer a  $\mathcal{TD}(P)$ . Os pontos  $q_c$  e  $q_d$  também não podem coincidir, pois isto implicaria em  $C_{abc} = C_{abd}$  e, portanto, a condição (2) sobre  $P$  seria violada (ver Observação 4.1)<sup>3</sup>. Logo, o ponto  $q_c$  está à esquerda do ponto  $q_d$  em

<sup>3</sup>Se  $q_c = q_d$ , então tomaríamos  $C = C_{abc} = C_{abd}$  e nossa afirmação ainda seria válida!



relação à orientação de  $\ell_{cd}$ . Seja  $q$  qualquer ponto de  $\ell_{cd}$  entre  $c$  e  $d$ . Então, tome  $C$  como o círculo de centro  $q$  e raio igual à distância,  $d(q, a)$ , entre  $q$  e o ponto  $a$ , que é igual a distância,  $d(q, b)$ , entre  $q$  e o ponto  $b$ . Por construção, a aresta  $[a, b]$  é uma corda de  $C$ . Além disso, como  $c$  está à esquerda de  $q_c$  em  $\ell_{cd}$  e como  $d$  está à direita de  $q_d$  em  $\ell_{cd}$ , podemos concluir que  $C \subset C_{abc} \cup C_{abd}$ . Logo, não pode haver nenhum ponto de  $P$  no interior de  $C$ . Reciprocamente, suponha que  $[a, b]$  é corda de um círculo,  $C$ , cujo interior não contém nenhum ponto de  $P$ . Desejamos mostrar que  $[a, b]$  é uma aresta de  $\mathcal{TD}(P)$ . Denote por  $\ell_{ab}$  a reta por  $a$  e  $b$  orientada de  $a$  para  $b$  (ver Figura 4.25). Seja  $g_{ab} : \mathbb{E}^2 \rightarrow \mathbb{R}$  qualquer forma afim tal que  $\ell_{ab} = Nuc(g_{ab})$ . Caso contrário, bastaria tomarmos  $g_{ab} = -g_{ab}$ . Seja  $c$  o ponto de  $P$  em  $int(H_+(g_{ab}))$  que forma com  $a$  e  $b$  um círculo,  $C_{abc}$ , de menor raio entre todos os pontos de  $P$  em  $int(H_+(g_{ab}))$ . Analogamente, seja  $d$  o ponto de  $P$  em  $int(H_-(g_{ab}))$  que forma com  $a$  e  $b$  um círculo,  $C_{abd}$ , de menor raio entre todos os pontos de  $P$  em  $int(H_-(g_{ab}))$ . Se qualquer um de  $c$  e  $d$  não existir, então a aresta  $[a, b]$  pertence à fronteira,  $\partial FC(P)$ , do fecho convexo,  $FC(P)$ , de  $P$ . Mas, como  $P \cap int(C) = \emptyset$ , não pode haver nenhum ponto de  $P$  no interior do segmento  $\overline{ab}$  e, portanto, a aresta  $[a, b]$  tem de pertencer a  $\mathcal{TD}(P)$ . Logo, vamos assumir que  $c$  e  $d$  existam. Por construção, o centro,  $q$ , de  $C$  e os centros,  $q_c$  e  $q_d$ , de  $C_{abc}$  e  $C_{abd}$ , respectivamente, são colineares. Seja  $\ell_{cd}$  a reta orientada de  $q_c$  para  $q_d$  que passa por esses três pontos. Afirmamos que o ponto  $q_c$  não está à direita de  $q$  e o ponto  $q_d$  não está à esquerda de  $q$  em relação à orientação de  $\ell_{cd}$ . Do contrário, o ponto  $c$  ou o ponto  $d$  ou ambos pertenceriam a  $int(C)$ . Logo,

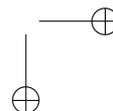
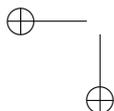
$$C_{abc} \cap H_-(g_{ab}) \subseteq C_{abd} \cap H_-(g_{ab})$$

e

$$C_{abd} \cap H_+(g_{ab}) \subseteq C_{abc} \cap H_+(g_{ab}).$$

Consequentemente, não pode haver nenhum ponto de  $P$  em  $int(C_{abc}) \cup int(C_{abd})$ , o que nos diz que tanto  $[a, b, c]$  quanto  $[a, b, d]$  são triângulos de  $\mathcal{TD}(P)$  e, portanto, a aresta  $[a, b]$  tem de pertencer a  $\mathcal{TD}(P)$ .  $\square$

Todas as propriedades da triangulação de Delaunay que estudamos até então nos permitiram descrever um algoritmo para construir a triangulação e analisar a complexidade deste algoritmo. A



última propriedade que veremos nesta seção é a principal responsável pela preferência pela triangulação de Delaunay em muitas aplicações práticas<sup>4</sup>. Antes de apresentar a propriedade, vamos provar um resultado que é comumente usado para demonstrá-la (refira-se à Figura 4.26).

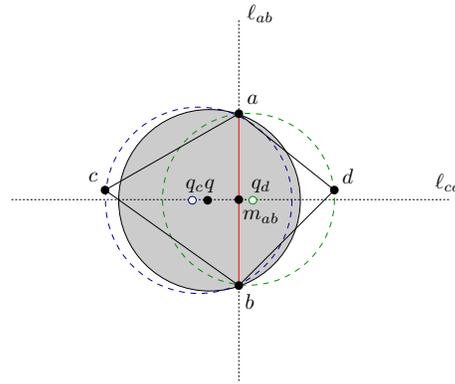


Figura 4.25: Ilustração do argumento da prova do Proposição 4.15.

**Lema 4.16** ([29]). *Dado um quadrilátero convexo,  $\square(acbd)$ , como o ilustrado pela Figura 4.26, tal que o vértice  $a$  pertence ao exterior do círculo, ou reside na circunferência deste mesmo círculo, cuja circunferência passa pelos vértices  $b, c$  e  $d$ , então a desigualdade abaixo é válida:*

$$\max\{R([a, b, d]), R([a, b, c])\} \geq \max\{R([b, c, d]), R([a, c, d])\},$$

em que  $R(\tau)$  denota o raio da circunferência circunscrita ao triângulo  $\tau$ .

*Demonstração.* Afirmamos que um de  $R([a, b, d])$  e  $R([a, b, c])$  é maior do que  $R([b, c, d])$  e o outro é maior do que  $R([a, c, d])$ . Aplicando o Corolário 4.9 ao círculo,  $C_{abd}$ , cuja circunferência passa por  $a, b$  e  $d$

<sup>4</sup>Ver [95] para mais propriedades importantes na prática.

e ao círculo,  $C_{bcd}$ , cuja circunferência passa por  $b$ ,  $c$  e  $d$ , concluímos que

$$\angle bad < \angle bcd, \angle cab < \angle cdb, \angle dba < \angle dca, \angle abc < \angle adc.$$

Sabemos também que

$$\angle dcb + \angle cdb < \pi.$$

Logo,

$$\min\{\angle dcb, \angle cdb\} \leq \frac{\pi}{2}.$$

Suponha, inicialmente, que  $\angle dcb \leq \pi/2$ . Seja  $e$  o ponto médio do segmento  $\overline{bd}$  e sejam  $o_1$  e  $o_2$  os centros de  $C_{abd}$  e  $C_{bcd}$ , respectivamente. Sabemos que  $o_1$  e  $o_2$  pertencem à mediatriz  $\ell$  do segmento  $\overline{bd}$ , que é a reta perpendicular à  $\overline{bd}$  por  $e$ . Logo,  $e$ ,  $o_1$  e  $o_2$  são colineares. Denote por  $a'$  e  $b'$  os pontos em que  $\ell$  intersecta  $C_{abd}$  e  $C_{bcd}$  nos arcos definidos por  $b$  e  $d$  que contêm  $a$  e  $c$  em  $C_{abd}$  e  $C_{bcd}$ , respectivamente.

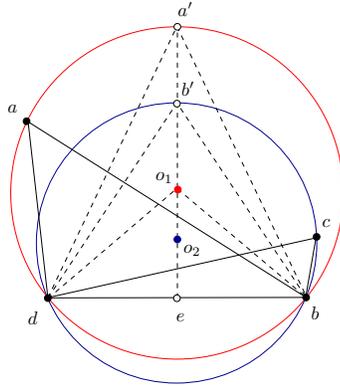
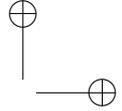
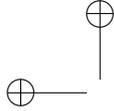


Figura 4.26: Ilustração do argumento da prova do Lema 4.16.

Agora, note que

$$\angle ea'b = \frac{\angle bad}{2} \quad \text{e} \quad \angle eb'b = \frac{\angle bcd}{2},$$



pois

$$\angle bad = \angle ba'd \quad \text{e} \quad \angle bcd = \angle bb'd.$$

Dai,

$$\operatorname{tg}\left(\frac{\angle bcd}{2}\right) = \frac{\|\vec{eb}\|}{\|\vec{eb'}\|} = \frac{\|\vec{eb}\|}{\|\vec{eo_2}\| + \|\vec{o_2b'}\|} = \frac{\|\vec{eb}\|}{\|\vec{eo_2}\| + \|\vec{o_2d}\|}.$$

Como

$$\|\vec{eo_2}\|^2 = \|\vec{o_2b}\|^2 - \|\vec{eb}\|^2,$$

temos

$$\operatorname{tg}\left(\frac{\angle bcd}{2}\right) = \frac{\|\vec{eb}\|}{\|\vec{o_2b}\| + \sqrt{\|\vec{o_2b}\|^2 - \|\vec{eb}\|^2}}.$$

De forma análoga,

$$\operatorname{tg}\left(\frac{\angle bad}{2}\right) = \frac{\|\vec{eb}\|}{\|\vec{o_1b}\| + \sqrt{\|\vec{o_1b}\|^2 - \|\vec{eb}\|^2}}.$$

Como

$$\angle bad < \angle bcd \leq \frac{\pi}{2},$$

obtemos

$$\operatorname{tg}\left(\frac{\angle bad}{2}\right) \leq \operatorname{tg}\left(\frac{\angle bcd}{2}\right),$$

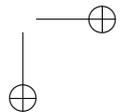
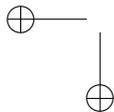
ou seja,

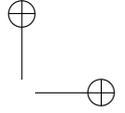
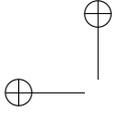
$$\frac{\|\vec{ec}\|}{\|\vec{o_1b}\| + \sqrt{\|\vec{o_1b}\|^2 - \|\vec{eb}\|^2}} \leq \frac{\|\vec{eb}\|}{\|\vec{o_2b}\| + \sqrt{\|\vec{o_2b}\|^2 - \|\vec{eb}\|^2}}$$

e

$$\|\vec{o_1b}\| + \sqrt{\|\vec{o_1b}\|^2 - \|\vec{eb}\|^2} \geq \|\vec{o_2b}\| + \sqrt{\|\vec{o_2b}\|^2 - \|\vec{eb}\|^2}.$$

Como a função  $f : \mathbb{R} \rightarrow \mathbb{R}$ , com  $f(x) = x + \sqrt{x^2 - \alpha}$ , para alguma constante  $\alpha \in \mathbb{R}$ , é monotonicamente crescente em  $x$ , podemos concluir que  $\|\vec{o_2b}\| < \|\vec{o_1b}\|$  e, portanto,  $R([a, b, d]) > R([b, c, d])$ . Suponha, agora, que  $\angle dcb > \pi/2$ . Isto implica que  $\angle cdb \leq \pi/2$ . Daí, podemos utilizar os mesmos argumentos de antes para deduzir que  $R([a, b, c]) > R([b, c, d])$ . Daí, podemos concluir que  $R([a, c, d])$  ou





$R([a, b, c])$  ou ambos são maiores do que  $R([b, c, d])$ . Além disso, como o ponto  $b$  pertence ao exterior de  $C_{acd}$ , podemos repetir os mesmos passos de antes, com  $b$  no papel de  $a$ , para concluir que  $R([a, b, d])$  ou  $R([a, b, c])$  ou ambos são maiores do que  $R([a, c, d])$ . Logo, acabamos de provar que

$$\max\{R([a, b, d]), R([a, b, c])\} \geq \max\{R([b, c, d]), R([a, c, d])\}.$$

□

A triangulação de Delaunay,  $\mathcal{TD}(P)$ , de um subconjunto,  $P$ , finito e não vazio de pontos de  $\mathbb{E}^2$  é ótima, com relação a todas as triangulações de  $P$ , de acordo com os critérios listados no corolário a seguir:

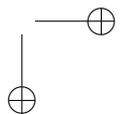
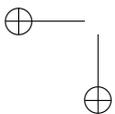
**Corolário 4.17.** *Seja  $P$  um subconjunto qualquer finito e não vazio de pontos de  $\mathbb{E}^2$ . Então, entre todas as triangulações de  $P$ , a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  é ótima com relação aos critérios abaixo:*

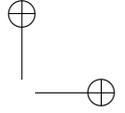
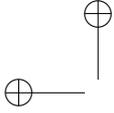
- (a)  $\mathcal{TD}(P)$  maximiza o valor do menor ângulo da triangulação.
- (b)  $\mathcal{TD}(P)$  minimiza o valor do maior raio entre os raios de todas as circunferências circunscritas aos triângulos da triangulação.
- (c) Para todo número real positivo,  $m$ , a função escalar  $J : \mathcal{T} \rightarrow \mathbb{R}$ , com

$$J(\mathcal{T}(P)) = \sum_{\tau \in \mathcal{T}(P)} R(\tau)^m,$$

é minimizada se  $\mathcal{T}(P) = \mathcal{TD}(P)$ , em que  $\tau$  denota um triângulo de  $\mathcal{T}(P)$  e  $\mathcal{T}$ , o conjunto de todas as possíveis triangulações de  $P$ .

*Demonstração.* Para provar (a), suponha que  $\mathcal{T}(P)$  seja qualquer triangulação de  $P$  que maximiza o valor do menor ângulo da triangulação entre todas as triangulações de  $P$ . Se  $\mathcal{T}(P) = \mathcal{TD}(P)$ , então não há mais nada a provar. Logo, suponha que  $\mathcal{T}(P) \neq \mathcal{TD}(P)$  e considere qualquer sequência  $S$  de trocas de arestas que converte  $\mathcal{T}(P)$  em  $\mathcal{TD}(P)$ . Da prova do Lema 4.12, sabemos que cada troca de aresta





substituí seis ângulos da triangulação por seis novos ângulos. No entanto, o menor ângulo dos seis resultantes não pode ser menor do que o menor ângulo entre os seis ângulos substituídos. Daí, podemos concluir que o menor ângulo de  $\mathcal{T}(P)$  não pode ser maior do que o menor ângulo de  $\mathcal{TD}(P)$ , o que implica que  $\mathcal{TD}(P)$  também maximiza o valor do menor ângulo da triangulação. Este mesmo argumento pode ser usado para provar as afirmações (b) e (c). No entanto, devemos utilizar o Lema 4.16 para concluir que os dois triângulos gerados por cada troca de aresta estão circunscritos por circunferências cujos raios não excedem o maior dos raios das circunferências circunscritas aos dois triângulos que foram removidos da triangulação anterior à troca.  $\square$

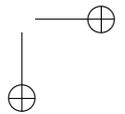
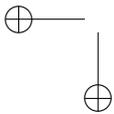
## 4.6 O algoritmo incremental

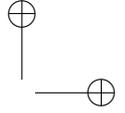
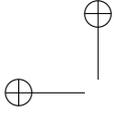
Esta seção apresenta um algoritmo eficiente para calcular a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de um subconjunto,  $P$ , de pontos de  $\mathbb{E}^2$ . O conjunto  $P$  deve satisfazer as condições (1)-(3) do Teorema 4.6. Eventualmente, a condição (2) será descartada e as mudanças no algoritmo, para que ele gere *uma* triangulação de Delaunay de  $P$ , serão discutidas. O algoritmo constrói a triangulação de forma *incremental*; isto é, o algoritmo recebe como entrada o conjunto  $P$  e gera uma sequência,  $p_{-2}, p_{-1}$  e  $p_0, p_1, \dots, p_n$ , tal que  $p_{-2}, p_{-1}$  e  $p_0$  são três pontos especiais cujas coordenadas são especificadas na Subseção 4.6.1,  $n = |P|$  e  $p_i$  é um ponto de  $P$ , para todo  $i = 1, \dots, n$ . Em seguida, o algoritmo constrói uma sequência de triangulações de subconjuntos de  $\mathbb{E}^2$ ,

$$\mathcal{TD}(S_0), \mathcal{TD}(S_1), \dots, \mathcal{TD}(S_n),$$

em que  $S_0 = \{p_{-2}, p_{-1}, p_0\}$  é o conjunto formado pelos pontos especiais e  $S_i = S_{i-1} \cup \{p_i\}$ , para todo  $i = 1, \dots, n$ . Provaremos que a  $j$ -ésima triangulação,  $\mathcal{TD}_j$ , da sequência é a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$ , para todo  $j = 0, \dots, n$ . O algoritmo gera  $\mathcal{TD}(P)$  a partir de  $\mathcal{TD}_n$ , removendo de  $\mathcal{TD}_n$  os vértices correspondentes aos pontos especiais e todas as arestas e triângulos que incidem sobre eles.

O Algoritmo 4.2 consiste de todos os passos descritos acima. As próximas subseções detalham e provam a corretude de cada um desses





passos.

#### 4.6.1 Os pontos especiais e a triangulação $\mathcal{TD}_0$

Dado um subconjunto,  $P$ , de  $n$  pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6, o primeiro passo do algoritmo incremental consiste em gerar uma sequência,  $\mathcal{S}$ , consistindo exatamente dos três pontos especiais,  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ , nesta ordem, e de alguma permutação,  $p_1, \dots, p_n$ , de todos os  $n$  pontos de  $P$ ; isto é,  $\mathcal{S}$  é da forma

$$p_{-2}, p_{-1}, p_0, p_1, \dots, p_n.$$

Assumimos que  $p_1, \dots, p_n$  é qualquer permutação dos pontos de  $P$  e que *toda permutação dos pontos de  $P$  é igualmente provável de ser escolhida pelo algoritmo para gerar a sequência  $\mathcal{S}$* . A segunda hipótese nos permitirá analisar a complexidade de tempo de execução *esperado* do algoritmo.

---

#### Algoritmo 4.2 CALCULATRIDEL( $P$ )

---

- 1: gere uma sequência  $p_1, \dots, p_n$  a partir de uma permutação de  $P$
  - 2: calcule as coordenadas dos pontos especiais  $p_{-2}$ ,  $p_{-1}$  e  $p_0$
  - 3: gere a triangulação  $\mathcal{TD}_0$
  - 4: **para**  $j = 1, \dots, |P|$  **faça**
  - 5:    $\mathcal{TD}_j \leftarrow \text{INSEREPONTO}(p_j, \mathcal{TD}_{j-1})$
  - 6: **fim para**
  - 7: remova  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  de  $\mathcal{TD}_{|P|}$
  - 8: **devolva**  $\mathcal{TD}_{|P|}$
- 

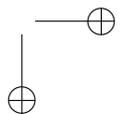
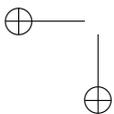
As coordenadas dos três pontos especiais,  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ , são tais que

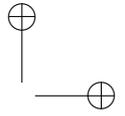
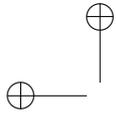
$$p_{-2} = (-M, -M), \quad p_{-1} = (0, M) \quad \text{e} \quad p_0 = (M, 0),$$

em que

$$M = 3 \cdot \max_{i=1}^n \{|x_i|, |y_i|\},$$

sendo  $(x_i, y_i)$  as coordenadas do ponto  $p_i$ , para todo  $i = 1, \dots, n$ . Em outras palavras, as coordenadas de  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ , são definidas de tal





forma que o fecho convexo,  $FC(P)$ , de  $P$  está inteiramente contido no interior do triângulo  $\tau_0 = [p_{-2}, p_{-1}, p_0]$ , como pode ser visto na Figura 4.27.

### 4.6.2 A inserção de pontos

O segundo passo do algoritmo (linhas 3 a 6 do Algoritmo 4.2) constrói uma sequência de triangulações,  $\mathcal{TD}_0, \mathcal{TD}_1, \dots, \mathcal{TD}_n$ , em que  $\mathcal{TD}_0$  é a triangulação de Delaunay,  $\mathcal{TD}(S_0)$ , do conjunto  $S_0 = \{p_{-2}, p_{-1}, p_0\}$  e, para todo  $i = 1, \dots, n$ , a triangulação  $\mathcal{TD}_i$  é a triangulação de Delaunay,  $\mathcal{TD}(S_i)$ , do conjunto  $S_i = S_{i-1} \cup \{p_i\}$ . Observe que  $\mathcal{TD}(0)$  contém apenas um triângulo,  $\tau_0 = [p_{-2}, p_{-1}, p_0]$ . Além disso, todo  $p_i$  pertence ao interior de  $\tau_0$ . Logo, o espaço subjacente,

$$|\mathcal{TD}_i|,$$

de  $\mathcal{TD}_i$ , que é o fecho convexo  $FC(S_i)$  de  $S_i$ , é exatamente o triângulo  $\tau_0$ .

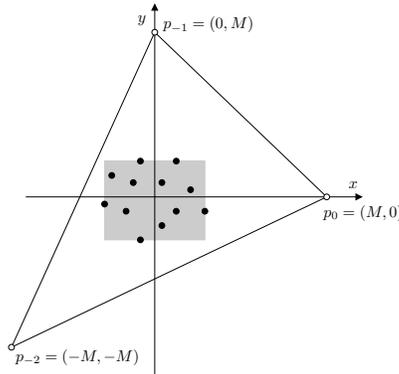
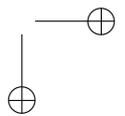
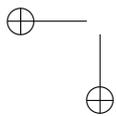
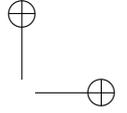
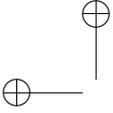


Figura 4.27: O conjunto de pontos  $P \cup \{p_{-2}, p_{-1}, p_0\}$  e o triângulo  $\tau_0$ .

A construção da sequência de triangulações pode ser feita de forma indutiva. Para tal, suponha que  $\mathcal{TD}_{j-1}$  já tenha sido construída e que ela seja igual à triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , de  $S_{j-1}$ . Vamos





mostrar como podemos *inserir* o ponto  $p_j$  em  $\mathcal{TD}_{j-1}$  para gerar a triangulação,  $\mathcal{TD}_j$ , que provaremos ser a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$ . Como  $p_i \in \tau_0$  e  $|\mathcal{TD}_{j-1}| = \tau_0$ , o ponto  $p_i$  tem de pertencer ao interior de algum triângulo ou de alguma aresta de  $\mathcal{TD}_{j-1}$ . A primeira tarefa do segundo passo do algoritmo é encontrar este triângulo ou aresta. Esta tarefa é conhecida por *localização de ponto* e é discutida na Subsecção 4.6.4. Por enquanto, vamos assumir que existe um procedimento do tipo “caixa preta” tal que, dados o ponto  $p_j$  e a triangulação  $\mathcal{TD}_{j-1}$ , o procedimento nos devolve o triângulo ou a aresta de  $\mathcal{TD}_{j-1}$  que contém  $p_j$  em seu interior. A Figura 4.28 ilustra os dois cenários mutuamente excludentes: (a)  $p_j$  está no interior de um triângulo e (b)  $p_j$  está no interior de uma aresta.

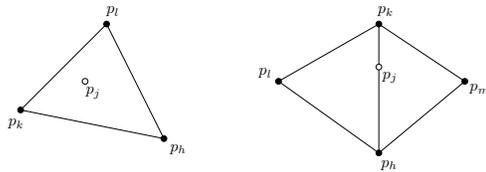
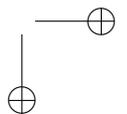
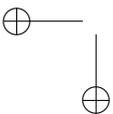
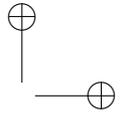
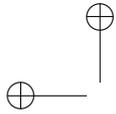


Figura 4.28: O ponto  $p_j$  inserido em  $\mathcal{TD}_{j-1}$  para gerar  $\mathcal{TD}_j$  pertence ao interior de um triângulo de  $\mathcal{TD}_{j-1}$  ou ao interior de uma aresta de  $\mathcal{TD}_{j-1}$ .

**Observação 4.6.** A notação  $\mathcal{TD}_j$  é usada tanto para a triangulação devolvida pela chamada `INSEREPONTO( $p_j, \mathcal{TD}_j$ )`, na linha 5 de Algoritmo 4.2, quanto para as triangulações intermediárias que surgem durante a execução de `INSEREPONTO( $p_j, \mathcal{TD}_j$ )` para gerar a  $\mathcal{TD}_j$  devolvida.

O algoritmo trata os dois casos da Figura 4.28 de maneira distinta. Se  $p_j$  pertence ao interior de um triângulo,  $\tau = [p_h, p_k, p_l]$ , de  $\mathcal{TD}_{j-1}$ , então o triângulo  $\tau$  é removido de  $\mathcal{TD}_{j-1}$ , o que resulta em um complexo simplicial denotado por  $\mathcal{TD}_{j-1} - \tau$ . Em seguida, o ponto  $p_j$ , os triângulos  $[p_h, p_j, p_k]$ ,  $[p_k, p_j, p_l]$  e  $[p_l, p_j, p_h]$  e as arestas  $[p_j, p_h]$ ,  $[p_j, p_k]$  e  $[p_j, p_l]$  são inseridos em  $\mathcal{TD}_{j-1} - \tau$ , gerando a





triangulação,  $\mathcal{TD}_j$ , de  $S_j$  (ver Figura 4.29). Observe que  $\mathcal{TD}_j$  não é necessariamente a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$  neste ponto.

Se  $p_j$  pertence ao interior de uma aresta,  $\mu = [p_h, p_k]$ , de  $\mathcal{TD}_{j-1}$ , então  $\mu$  incide sobre dois triângulos,  $\sigma$  and  $\tau$ , de  $\mathcal{TD}_{j-1}$ , em que  $\sigma = [p_h, p_k, p_l]$  e  $\tau = [p_h, p_k, p_m]$ , pois  $p_j$  não pode pertencer à fronteira de  $\tau_0$ . O algoritmo remove  $\mu$ ,  $\sigma$  e  $\tau$  de  $\mathcal{TD}_{j-1}$ , gerando um complexo simplicial denotado por  $\mathcal{TD}_{j-1} - \{\tau, \sigma, \mu\}$ . Em seguida, o ponto  $p_j$ , os triângulos  $[p_j, p_h, p_l]$ ,  $[p_j, p_l, p_k]$ ,  $[p_j, p_k, p_m]$  e  $[p_j, p_m, p_h]$  e as arestas  $[p_j, p_h]$ ,  $[p_j, p_l]$ ,  $[p_j, p_k]$  e  $[p_j, p_m]$  são inseridos em  $\mathcal{TD}_{j-1} - \{\tau, \sigma, \mu\}$ , gerando a triangulação,  $\mathcal{TD}_j$ , de  $S_j$  (ver Figura 4.30), que não é necessariamente a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$  neste ponto.

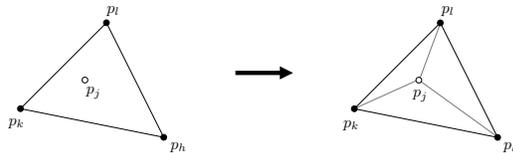


Figura 4.29: A inserção de  $p_j$  no interior do triângulo  $[p_h, p_k, p_l]$ .

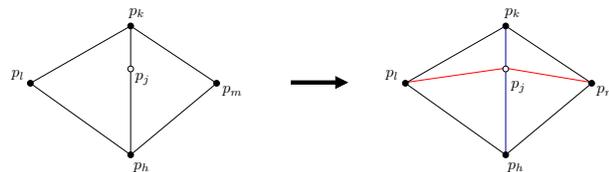
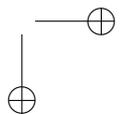
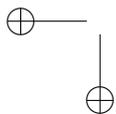
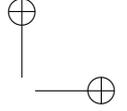
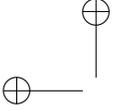


Figura 4.30: A inserção de  $p_j$  no interior da aresta  $[p_h, p_k]$ .

A próxima tarefa do algoritmo consiste em converter  $\mathcal{TD}_j$  na triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$ . Para tal, o algoritmo repetidamente procura por uma aresta de  $\mathcal{TD}_j$  que não é localmente Delaunay. Se uma tal aresta for encontrada, então o algoritmo executa

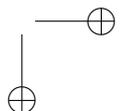
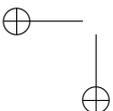


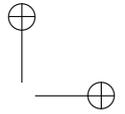


a operação de troca de aresta, introduzida na Seção 4.5, para substituí-la por uma aresta localmente Delaunay. Se nenhuma aresta que não é localmente Delaunay for encontrada, então  $\mathcal{TD}_j$  já é a própria  $\mathcal{TD}(S_j)$  (ver Definição 4.4 e o Lema 4.10) e o segundo passo é encerrado. Mas, como o algoritmo pode determinar, de *forma sistemática e eficiente*, as arestas de  $\mathcal{TD}_j$  que não são localmente Delaunay?

A pergunta acima está relacionada a um aspecto crucial do segundo passo do Algoritmo 4.2, pois uma troca de aresta pode fazer com que uma aresta que era localmente Delaunay *antes da troca* passe a ser não localmente Delaunay *após a troca*. Felizmente, através do estudo de algumas propriedades de  $\mathcal{TD}_j$ , podemos desenvolver uma forma eficiente de determinar todas as arestas afetadas por cada troca de aresta. Primeiro, provaremos que, nos dois cenários ilustrados pela Figura 4.29 e Figura 4.30, as três (resp. quatro) arestas inseridas no complexo simplicial  $\mathcal{TD}_{j-1} - \tau$  (resp.  $\mathcal{TD}_{j-1} - \{\tau, \sigma, \mu\}$ ) para gerar  $\mathcal{TD}_j$  são localmente Delaunay em  $\mathcal{TD}_j$  (Lema 4.18). Segundo, observamos que todas ou alguma(s) das três (resp. quatro) arestas opostas a  $p_j$  em  $\mathcal{TD}_j$ , isto é,  $[p_h, p_k]$ ,  $[p_k, p_l]$  e  $[p_l, p_h]$  (resp.,  $[p_h, p_k]$ ,  $[p_k, p_l]$ ,  $[p_l, p_m]$  e  $[p_m, p_h]$ ), pode(m) não ser localmente Delaunay em  $\mathcal{TD}_j$ , embora todas elas sejam localmente Delaunay em  $\mathcal{TD}_{j-1}$ . Isto porque *assumimos* que  $\mathcal{TD}_{j-1} = \mathcal{TD}(S_{j-1})$ . Porém, em  $\mathcal{TD}_j$ , cada uma delas é incidente em um triângulo que não está em  $\mathcal{TD}_{j-1}$ . Logo, é preciso verificar se a inserção de  $p_j$  em  $\mathcal{TD}_{j-1}$  fez com que tais arestas não sejam mais localmente Delaunay. Daqui em diante, utilizamos o termo *suspeita* para nos referirmos a qualquer aresta de  $\mathcal{TD}_j$  que é testada pelo algoritmo para determinar se a mesma é localmente Delaunay.

Inicialmente, há, no máximo, quatro arestas suspeitas. Todas elas são opostas a  $p_j$  em algum triângulo de  $\mathcal{TD}_j$ . Se  $\nu$  é qualquer uma das arestas suspeitas, então o algoritmo utiliza o predicado *incircle* (ver Apêndice A) para determinar se  $\nu$  é localmente Delaunay. Se for, então o algoritmo descarta  $\nu$  e considera outra aresta suspeita. Caso contrário, o algoritmo executa a operação de troca de aresta (Seção 4.5), substituindo  $\nu$  por uma aresta incidente em  $p_j$  (refira-se à Figura 4.31). A troca de aresta faz com que dois triângulos sejam substituídos por outros dois triângulos em  $\mathcal{TD}_j$  e, portanto, faz com que duas outras arestas de  $\mathcal{TD}_j$  se tornem suspeitas. Cada uma das duas arestas é a aresta oposta a  $p_j$  em um dos dois novos triângulos





de  $\mathcal{TD}_j$ .

**Observação 4.7.** *De forma geral, toda aresta suspeita é oposta ao ponto  $p_j$  e toda aresta que substitui uma aresta suspeita é incidente em  $p_j$ .*

A Observação 4.7 nos sugere um algoritmo recursivo para converter a triangulação  $\mathcal{TD}_j$  em  $\mathcal{TD}(S_j)$  (ver Algoritmo 4.3). A linha 3 remove  $\tau$  de  $\mathcal{TD}_{j-1}$  e insere o ponto  $p_j$ , as arestas  $[p_j, p_h]$ ,  $[p_j, p_k]$  e  $[p_j, p_l]$  e os triângulos  $[p_h, p_k, p_j]$ ,  $[p_k, p_l, p_j]$  e  $[p_l, p_h, p_j]$  no complexo  $\mathcal{TD}_{j-1} - \tau$  para gerar a triangulação  $\mathcal{TD}_j$ . Por sua vez, a linha 10 remove  $\nu$ ,  $\tau$  e  $\sigma$  de  $\mathcal{TD}_{j-1}$  e insere o ponto  $p_j$ , as arestas  $[p_j, p_h]$ ,  $[p_j, p_l]$ ,  $[p_j, p_k]$  e  $[p_j, p_m]$  e os triângulos  $[p_j, p_h, p_l]$ ,  $[p_j, p_l, p_k]$ ,  $[p_j, p_k, p_m]$  em  $\mathcal{TD}_{j-1} - \{\tau, \sigma, \nu\}$  para gerar  $\mathcal{TD}_j$ . Já o procedimento  $\text{TESTATROCA}()$  determina se uma aresta suspeita não é localmente Delaunay e realiza a troca de aresta caso ela não seja. Este procedimento chama a si mesmo duas vezes caso a aresta seja trocada, pois cada troca de aresta faz com que duas arestas se tornem suspeitas e  $\text{TESTATROCA}()$  (ver Algoritmo 4.4) é executado para cada uma delas.

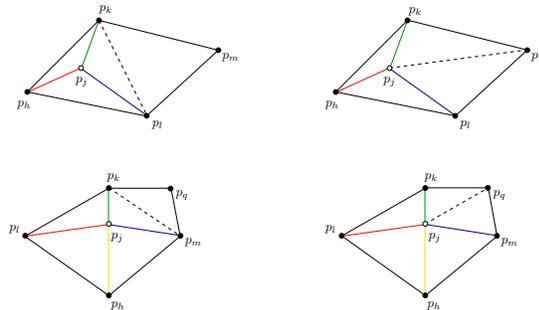
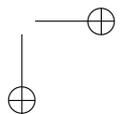
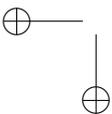
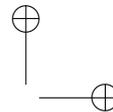
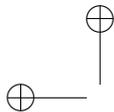


Figura 4.31: Cada troca de aresta substitui uma aresta por por outra e dois triângulos por outros dois em  $\mathcal{TD}_j$ . A nova aresta é sempre incidente no vértice  $p_j$  inserido em  $\mathcal{TD}_{j-1}$  para gerar a triangulação  $\mathcal{TD}_j$ .

A Figura 4.32 exemplifica a execução de uma chamada da forma





TROCAARESTAS( $p, a, b, \mathcal{T}$ ), exibindo apenas o subconjunto de triângulos da triangulação  $\mathcal{T}$  gerados, recursivamente, a partir da troca de  $[a, b]$ . Observe que a linha 1 de TROCAARESTAS() evita que uma aresta pertencente à fronteira,  $\partial(|\mathcal{T}|)$ , do espaço subjacente,  $|\mathcal{T}|$ , de  $\mathcal{T}$ , denominada também de *aresta de fronteira de  $\mathcal{T}$* , seja considerada para o teste de troca de aresta da linha 5, pois tal aresta é incidente em apenas um triângulo de  $\mathcal{T}$ . O que nos resta agora é mostrar que a execução da chamada INSEREPONTO( $p_j, \mathcal{TD}_{j-1}$ ) gera, de fato, a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j = S_{j-1} \cup \{p_j\}$  sempre que  $\mathcal{TD}_{j-1}$  for a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , dos pontos em  $S_{j-1}$ .

---

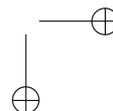
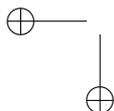
**Algoritmo 4.3** INSEREPONTO( $p, \mathcal{T}$ )

---

- 1: encontre um triângulo  $\tau = [p_h, p_k, p_l]$  de  $\mathcal{T}$  contendo  $p$
  - 2: **se**  $p$  pertence ao interior de  $\tau$  **então**
  - 3:   insira  $p$  em  $\mathcal{T}$  e gere  $\mathcal{T}'$
  - 4:   TESTATROCA( $p, p_h, p_k, \mathcal{T}'$ )
  - 5:   TESTATROCA( $p, p_k, p_l, \mathcal{T}'$ )
  - 6:   TESTATROCA( $p, p_l, p_h, \mathcal{T}'$ )
  - 7: **senão**
  - 8:   seja  $\nu = [p_h, p_k]$  a aresta de  $\tau$  contendo  $p$
  - 9:   seja  $\sigma = [p_h, p_k, p_m]$  o outro triângulo de  $\mathcal{T}$  contendo  $\nu$
  - 10:   insira  $p$  em  $\mathcal{T}$  e gere  $\mathcal{T}'$
  - 11:   TESTATROCA( $p_j, p_h, p_l, \mathcal{T}'$ )
  - 12:   TESTATROCA( $p_j, p_l, p_k, \mathcal{T}'$ )
  - 13:   TESTATROCA( $p_j, p_k, p_m, \mathcal{T}'$ )
  - 14:   TESTATROCA( $p_j, p_m, p_h, \mathcal{T}'$ )
  - 15: **fim se**
  - 16: **devolva**  $\mathcal{T}'$
- 

O lema a seguir nos diz que as arestas criadas inicialmente pelo Algoritmo 4.3 na inserção de  $p_j$  em  $\mathcal{TD}_{j-1}$  são arestas da triangulação  $\mathcal{TD}(S_j)$ :

**Lema 4.18.** *As arestas criadas inicialmente por INSEREPONTO() para inserir o ponto  $p_j$  em  $\mathcal{TD}_{j-1}$  são arestas da triangulação  $\mathcal{TD}(S_j)$  se  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , dos pontos em  $S_{j-1}$ .*



---

**Algoritmo 4.4** TROCAARESTAS( $p, a, b, \mathcal{T}$ )

---

- 1: se  $[a, b]$  é uma aresta da fronteira de  $\mathcal{T}$  então
  - 2:   **devolva**
  - 3: **fim se**
  - 4: seja  $[a, c, b]$  o triângulo que compartilha  $[a, b]$  com  $[a, b, p]$
  - 5: se INCIRCLE( $a, c, b, p$ ) então
  - 6:   troque a aresta  $[a, b]$  pela aresta  $[p, c]$  em  $\mathcal{T}$
  - 7:   TROCAARESTAS( $p, a, c, \mathcal{T}$ )
  - 8:   TROCAARESTAS( $p, c, b, \mathcal{T}$ )
  - 9: **fim se**
  - 10: **devolva**
- 

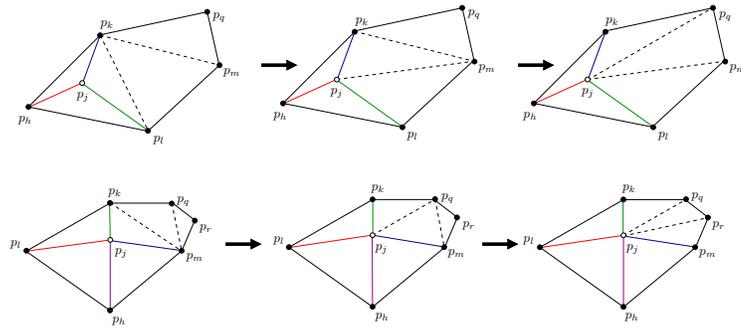


Figura 4.32: Exemplo de uma chamada a TROCAARESTAS().

*Demonstração.* Suponha que  $p_j$  pertença a um triângulo  $[p_h, p_k, p_l]$  de  $\mathcal{TD}_{j-1}$  ( $p_j$  não necessariamente está no interior do triângulo). Seja  $C_{hkl}$  o círculo circunscrito a  $[p_h, p_k, p_l]$ . Como, por hipótese,  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , dos pontos de  $S_{j-1} = \{p_{-2}, p_{-1}, p_0, p_1, \dots, p_{j-1}\}$ , sabemos que o interior,  $int(C_{hkl})$ , de  $C_{hkl}$  contém apenas um ponto de  $S_j = S_{j-1} \cup \{p_j\}$ : o próprio  $p_j$ . Além disso, há exatamente três pontos sobre a circunferência de  $C_{hkl}$ :  $p_h$ ,  $p_k$  e  $p_l$ . Para cada ponto  $b \in \{p_h, p_k, p_l\}$ , considere o círculo,  $C_b$ , que passa por  $p_j$  e é tangente a  $C_{hkl}$  em  $b$ . Por construção, os centros de  $C_{hkl}$ ,  $C_b$  e o ponto  $b$  são colineares. Este fato, aliado ao fato que

$p_j \in \text{int}(C_{hkl})$ , permite-nos concluir que  $C_b \subset C_{hkl}$  (ver Figura 4.33). Logo, o interior,  $\text{int}(C_b)$ , de  $C_b$  e  $S_j$  são conjuntos disjuntos. Pela Proposição 4.15, temos que  $[p_j, b] \in \mathcal{TD}(S_j)$  (ver Figura 4.34).  $\square$

**Observação 4.8.** *Toda aresta de uma triangulação de Delaunay é denominada aresta de Delaunay com relação ao conjunto de vértices da triangulação. De forma análoga, todo triângulo de uma triangulação de Delaunay é denominado triângulo de Delaunay com relação ao conjunto de vértices da triangulação. Qualquer aresta (resp. triângulo) de Delaunay goza da propriedade que garante a existência de um círculo cuja circunferência passa pelos vértices da aresta (resp. triângulo) e cujo interior não contém nenhum vértice da triangulação que a (resp. o) contém. Esta propriedade é chamada de círculo vazio.*

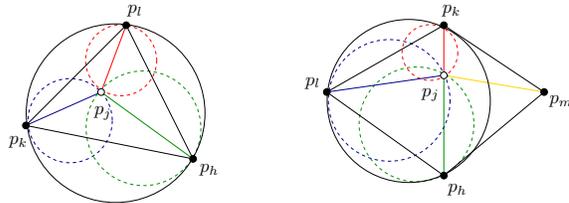


Figura 4.33: As arestas criadas pela linha 3 ou 10 do Algoritmo 4.3 pertencem a círculos cujos interiores não contêm vértices da triangulação.

O próximo lema garante que uma aresta que se tornou incidente em  $p_j$  mediante uma operação de troca de aresta, realizada pela execução do Algoritmo 4.3 para inserir de  $p_j$  em  $\mathcal{TD}_{j-1}$ , pertence a  $\mathcal{TD}(S_j)$ :

**Lema 4.19.** *Toda aresta que se torna incidente a  $p_j$ , mediante uma troca de aresta, durante a execução de  $\text{INSEREPONTO}(p_j, \mathcal{TD}_{j-1})$  para inserir o ponto  $p_j$  em  $\mathcal{TD}_{j-1}$ , pertence à triangulação  $\mathcal{TD}(S_j)$  se  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , dos pontos em  $S_{j-1}$ .*

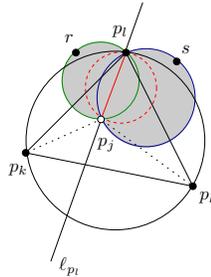


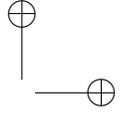
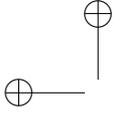
Figura 4.34: Uma aresta que corresponde a uma corda de um círculo cujo interior não possui nenhum ponto de um conjunto  $P$  contendo os extremos da aresta é uma aresta da triangulação de Delaunay de  $P$ .

*Demonstração.* Suponha que a aresta  $[p_h, p_l]$  do quadrilátero (convexo)  $\square(p_j p_h p_k p_l)$  seja substituída pela aresta  $[p_j, p_k]$  durante a execução de  $\text{INSERE PONTO}(p_j, \mathcal{T}\mathcal{D}_{j-1})$ . Se isto ocorreu, então  $p_j$  deve pertencer ao interior,  $\text{int}(C_{hkl})$ , do círculo  $C_{hkl}$  circunscrito ao triângulo de vértices em  $p_h, p_k$  e  $p_l$  de  $S_{j-1}$  (Lema 4.11). Mas, como o triângulo  $[p_h, p_k, p_l]$  pertence à triangulação de Delaunay,  $\mathcal{T}\mathcal{D}(S_{j-1})$ , de  $S_{j-1}$ , o único ponto de  $S_j$  em  $\text{int}(C_{hkl})$  é  $p_j$ . Seja  $C_k$  o círculo tangente a  $C_{hkl}$  em  $p_k$  e cuja circunferência também contém  $p_j$ . Por construção, os centros de  $C_{hkl}$  e  $C_k$  e o ponto  $p_k$  são colineares, o que nos diz que  $C_k \subset C_{hkl}$ . Por sua vez, isto implica que o interior de  $C_k$  não contém nenhum ponto de  $S_j$ . Mas, pela Proposição 4.15, temos que

$$[p_j, p_k] \in \mathcal{T}\mathcal{D}(S_j).$$

□

Observe que  $\text{TROCAARESTAS}()$  realiza uma troca de aresta se, e somente se, a aresta suspeita *não* é localmente Delaunay. Para determinar se uma aresta é localmente Delaunay, o predicado *incircle* é usado. Por definição, se uma aresta não é localmente Delaunay, então ela certamente não é Delaunay, mas a recíproca não é verdadeira; isto é, se uma aresta é localmente Delaunay, então ela não

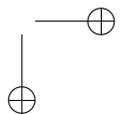
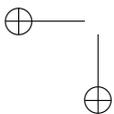


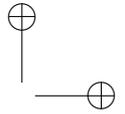
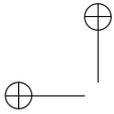
é necessariamente Delaunay (ver Problema 4.39). No entanto, pelo Lema 4.19, se `TROCAARESTAS()` descobre que uma aresta suspeita não é localmente Delaunay e a troca por outra aresta — que sabemos incidir em  $p_j$  e ser *localmente* Delaunay em  $\mathcal{TD}_j$  — também é uma aresta (de Delaunay) em  $\mathcal{TD}(S_j)$ . Por outro lado, se o predicado *incircle* acusar que uma aresta é localmente Delaunay em  $\mathcal{TD}_j$ , ela *não* é substituída. A pergunta natural é: uma aresta que é testada por `TROCAARESTAS()`, mas não é substituída, também é uma aresta de  $\mathcal{TD}(S_j)$ ? A resposta para esta pergunta é dada pelo Lema 4.20 a seguir:

**Lema 4.20.** *Toda aresta suspeita que foi testada durante a execução de `INSERE PONTO`( $p_j, \mathcal{TD}_{j-1}$ ), mas não foi substituída por uma operação de troca de aresta, pertence à triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$  se  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , de  $S_{j-1}$ .*

*Demonstração.* Toda aresta,  $[p_h, p_k]$ , de  $\mathcal{TD}_{j-1}$  que é testada durante a execução de `TROCAARESTAS()` (isto é,  $[p_h, p_k]$  é suspeita) é oposta ao ponto  $p_j$  no triângulo  $[p_j, p_h, p_k]$ . Se  $[p_h, p_k]$  incide apenas sobre um triângulo, isto é,  $[p_j, p_h, p_k]$ , então ela pertence à fronteira do espaço subjacente,  $|\mathcal{TD}_{j-1}|$ , de  $\mathcal{TD}_{j-1}$ , que é o mesmo de  $\mathcal{TD}_j$  e  $\mathcal{TD}(S_j)$ . Logo, a aresta  $[p_h, p_k]$  está em  $\mathcal{TD}(S_j)$ . Por outro lado, se  $[p_h, p_k]$  incide sobre outro triângulo, digamos  $[p_h, p_k, p_m]$ , então afirmamos que  $[p_h, p_k, p_m]$  pertence a  $\mathcal{TD}_{j-1}$ . De fato, todo triângulo que pertence a  $\mathcal{TD}_j$ , mas não pertence a  $\mathcal{TD}_{j-1}$ , é gerado por uma troca de aresta e, portanto, deve conter  $p_j$  como um de seus vértices (ver linha 6 do Algoritmo 4.4). Como  $\mathcal{TD}_{j-1}$  é uma triangulação de Delaunay, há um círculo,  $C_{hkm}$ , cujo interior,  $\text{int}(C_{hkm})$ , não contém nenhum ponto de  $S_{j-1}$ . Afirmamos que  $\text{int}(C_{hkm})$  também não contém nenhum ponto de  $S_j$ , pois, do contrário,  $\text{int}(C_{hkm})$  conteria apenas um ponto:  $p_j$ . Mas, se  $p_j$  pertencesse a  $\text{int}(C_{hkm})$ , a aresta  $[p_h, p_k]$  não seria localmente Delaunay (Lema 4.11) e, portanto, a aresta seria substituída, o que contradiz a hipótese do lema. Logo, o triângulo  $[p_h, p_k, p_m]$  pertence a  $\mathcal{TD}(S_j)$  e, portanto,  $[p_h, p_k]$  também.  $\square$

Os dois próximos lemas garantem o término da execução do algoritmo, que, por sua vez, serve de hipótese para concluir que  $\mathcal{TD}_j =$





$\mathcal{TD}(S_j)$ .

**Lema 4.21.**  $\text{TROCAARESTAS}()$  não testa uma mesma aresta duas vezes durante a execução de  $\text{INSEREPONTO}()$  que gera a triangulação  $\mathcal{TD}_j$ .

*Demonstração.* Seja  $\mu$  uma aresta de  $\mathcal{TD}_{j-1}$  que se torna suspeita durante a execução de  $\text{INSEREPONTO}(p_j, \mathcal{TD}_{j-1})$  e, portanto, é eventualmente testada por  $\text{TROCAARESTAS}()$ . Mas, este é o caso se, e somente se, existe uma aresta  $\nu$  tal que o ponto  $p_j$  e a aresta  $\mu$  estão em lados opostos de uma reta orientada que passa pelos extremos de  $\nu$  (ver Figura 4.35). Mas, como as novas arestas criadas pela operação de troca de aresta são sempre incidentes no ponto  $p_j$ , o cenário descrito acima só pode ocorrer uma única vez, já que nenhuma aresta pode separar  $\mu$  de  $p_j$  após  $\nu$  ser trocada por outra aresta.  $\square$

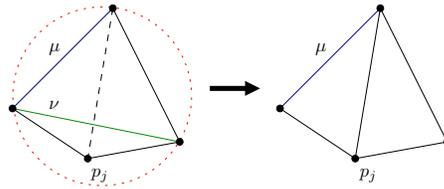
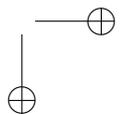
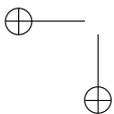
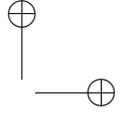
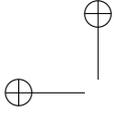


Figura 4.35: A aresta  $\mu$  se torna suspeita após a troca da aresta  $\nu$ , que se deve ao fato de  $\nu$  não ser localmente Delaunay, por outra aresta.

**Lema 4.22.** A execução de  $\text{INSEREPONTO}(p_j, \mathcal{TD}_{j-1})$  sempre termina.

*Demonstração.* Durante a execução de  $\text{INSEREPONTO}(p_j, \mathcal{TD}_{j-1})$ , uma aresta criada pela operação de troca de aresta jamais é testada pelo algoritmo  $\text{TROCAARESTAS}()$ , pois a aresta é incidente em  $p_j$ . Logo, apenas as arestas existentes em  $\mathcal{TD}_{j-1}$  podem se tornar suspeitas e testadas por  $\text{TROCAARESTAS}()$ . Este fato, aliado ao resultado do Lema 4.21 e ao fato que há um número finito de arestas





em  $\mathcal{TD}_{j-1}$ , garantem que o número de chamadas a TROCAARESTAS() é finito e isto é suficiente para garantir a validade da nossa afirmação.  $\square$

Finalmente, podemos provar que  $\mathcal{TD}_j = \mathcal{TD}(S_j)$ :

**Teorema 4.23.** *A execução de INSEREPONTO( $p_j, \mathcal{TD}_{j-1}$ ) devolve  $\mathcal{TD}(S_j)$  sempre que  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , de  $S_{j-1}$ .*

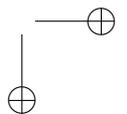
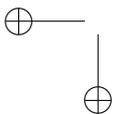
*Demonstração.* Se  $\mathcal{TD}_{j-1}$  é a triangulação de Delaunay,  $\mathcal{TD}(S_{j-1})$ , de  $S_{j-1}$ , então o Lema 4.18 e o Lema 4.19 nos dizem que as arestas que estão em  $\mathcal{TD}_j$ , mas não estão em  $\mathcal{TD}_{j-1}$ , são arestas da triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$ . Por outro lado, qualquer aresta de  $\mathcal{TD}_j$  que também pertence a  $\mathcal{TD}_{j-1}$  é (1) testada e passa no teste ou (2) nunca foi testada, durante a execução de INSEREPONTO( $p_j, \mathcal{TD}_{j-1}$ ). O Lema 4.20 garante que as arestas que se enquadram no caso (1) são arestas de  $\mathcal{TD}(S_j)$ . Afirmamos que as arestas que se enquadram no caso (2) também pertencem a  $\mathcal{TD}(S_j)$ . De fato, para cada uma dessas arestas, os dois triângulos (resp. o único triângulo) incidente(s) nela não mudaram(ou). Logo, como  $\mathcal{TD}_{j-1}$  é uma triangulação de Delaunay, essas arestas continuam sendo localmente Delaunay em  $\mathcal{TD}_j$  e, portanto, todas as arestas de  $\mathcal{TD}_j$  são localmente Delaunay. Daí,

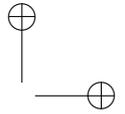
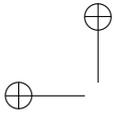
$$\mathcal{TD}_j = \mathcal{TD}(S_j).$$

A conclusão acima é uma consequência direta dos fatos na Observação 4.5.  $\square$

**Corolário 4.24.** *O segundo passo de CALCULATRIDEL( $P$ ) devolve  $\mathcal{TD}(S_{|P|})$ .*

*Demonstração.* O segundo passo de CALCULATRIDEL( $P$ ) consiste das linhas 3 a 6 do Algoritmo 4.2. A triangulação inicial,  $\mathcal{TD}_0$ , consiste apenas de um triângulo,  $\tau_0 = [p_{-2}, p_{-1}, p_0]$ , e, portanto, ela é a triangulação de Delaunay,  $\mathcal{TD}(S_0)$ , de  $S_0 = \{p_{-2}, p_{-1}, p_0\}$ . Daí, usando indução em  $j$  e o Teorema 4.23, podemos concluir que  $\mathcal{TD}_j$  é a triangulação de Delaunay,  $\mathcal{TD}(S_j)$ , de  $S_j$ , para todo  $j = 1, \dots, |P|$ .  $\square$





### 4.6.3 A remoção dos pontos especiais

O terceiro e último passo do Algoritmo 4.2 obtém a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$  a partir de  $\mathcal{TD}_{|P|}$ , que sabemos ser a triangulação de Delaunay,  $\mathcal{TD}(S_{|P|})$ , de  $S_{|P|}$ . Como dito anteriormente, queremos obter  $\mathcal{TD}(P)$  a partir de  $\mathcal{TD}_{|P|}$  através da simples remoção dos três pontos especiais,  $p_0$ ,  $p_{-1}$  e  $p_{-2}$ , de  $\mathcal{TD}_{|P|}$  e de todas as arestas e triângulos incidentes nesses pontos. Denote por  $\mathcal{K}$  o conjunto de vértices, arestas e triângulos que sobraram. É sempre verdade que  $\mathcal{K}$  é igual a  $\mathcal{TD}(P)$ ? A resposta é *não*! Na verdade, este é o caso se, e somente se, todos os vértices, arestas e triângulos de  $\mathcal{TD}(P)$  são vértices, arestas e triângulos de  $\mathcal{TD}_{|P|}$ . Se são, então todas as arestas e triângulos incidentes em  $p_0$ ,  $p_{-1}$  e  $p_{-2}$  estariam “fora” do espaço subjacente,  $|\mathcal{TD}(P)|$ , de  $\mathcal{TD}(P)$ . Caso contrário, temos

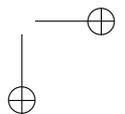
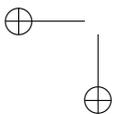
$$\mathcal{K} \neq \mathcal{TD}(P).$$

Considere o caso em que  $\mathcal{K} \neq \mathcal{TD}(P)$ . Afirmamos que há uma aresta de  $\mathcal{TD}(P)$  que pertence à *fronteira*,  $\partial(|\mathcal{TD}(P)|)$ , de  $|\mathcal{TD}(P)|$ , mas que não pertence a  $\mathcal{TD}_{|P|}$ . De fato, se todas as arestas de  $\mathcal{TD}(P)$  que pertencem a  $\partial(|\mathcal{TD}(P)|)$  fossem arestas de  $\mathcal{TD}_{|P|}$ , então todos os triângulos removidos de  $\mathcal{TD}_{|P|}$  estariam fora de  $|\mathcal{TD}(P)|$ . Neste caso, teríamos que  $|\mathcal{K}| = |\mathcal{TD}(P)|$  e, portanto, o conjunto  $\mathcal{K}$  seria uma triangulação de  $P$ . Afirmamos que esta triangulação é a própria  $\mathcal{TD}(P)$ . De fato, se  $\mu$  é qualquer aresta de  $\mathcal{K}$  tal que  $\mu \notin \partial(|\mathcal{TD}(P)|)$ , então  $\mu$  incide sobre dois triângulos de  $\mathcal{K}$  e, portanto, de  $\mathcal{TD}_{|P|}$ . Logo, a aresta  $\mu$  é localmente Delaunay. Mas, pelo Lema 4.11, se toda aresta de  $\mathcal{K}$  é localmente Delaunay, então  $\mathcal{K}$  é a triangulação de Delaunay de  $P$ :

$$\mathcal{K} = \mathcal{TD}(P),$$

uma contradição! Daí, concluímos que a condição *toda aresta de  $\mathcal{TD}(P)$  que pertence a  $\partial(|\mathcal{TD}(P)|)$  deve estar em  $\mathcal{TD}_{|P|}$*  é necessária e suficiente para que  $\mathcal{K}$  seja igual a  $\mathcal{TD}(P)$ . Por um lado, o Algoritmo 4.2 não garante que esta condição seja satisfeita. Por outro lado, a condição pode ser satisfeita com uma modificação do predicado *incircle*, mantendo o Algoritmo 4.2 intacto, de modo que teremos

$$\mathcal{TD}(P) = \mathcal{K}.$$



A modificação do predicado *incircle* a que nos referimos acima vai nos permitir satisfazer uma condição um pouco mais forte, a saber: *toda aresta de  $\mathcal{TD}(P_i)$  que pertence a  $\partial(|\mathcal{TD}(P_i)|)$  está em  $\mathcal{TD}_i$ , para todo  $i = 1, \dots, |P|$ , em que  $P_i$  é o conjunto dos pontos  $\{p_1, \dots, p_i\}$ . Quando  $i = |P|$ , esta condição implica que  $\mathcal{TD}(P) = \mathcal{K}$ . Para descrever a modificação que faremos no predicado *incircle*, considere a chamada  $\text{INCIRCLE}(a, c, b, p)$  na linha 5 de Algoritmo 4.4. Esta chamada devolve o valor lógico verdadeiro se, e somente se, o ponto  $p$  pertence ao interior,  $\text{int}(C_{acb})$ , do círculo,  $C_{acb}$ , cuja circunferência contém os pontos  $a, c$  e  $b$ , em que  $[a, b, p]$  e  $[a, c, b]$  são triângulos da triangulação corrente (ver Figura 4.36). Lembre-se de que se  $p$  pertence a  $\text{int}(C_{acb})$ , então o Algoritmo 4.4 substitui a aresta  $[a, b]$  pela aresta  $[c, p]$ . Caso contrário, a substituição de  $[a, b]$  por  $[c, p]$  não ocorre.*

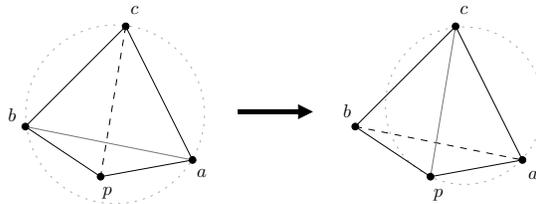
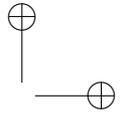
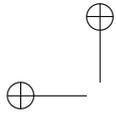


Figura 4.36: O ponto  $p$ , inserido na triangulação corrente, pertence ao interior,  $\text{int}(C_{acb})$ , do círculo,  $C_{acb}$ , cuja circunferência contém os pontos  $a, c$  e  $b$ . Isto faz com que a aresta  $[a, b]$  seja substituída por  $[c, p]$ .

Nós estamos interessados no cenário em que  $p$  é o ponto  $p_j$  e a aresta  $[a, b]$  pertence à fronteira,  $\partial(|\mathcal{TD}(P_{j-1})|)$ , do espaço subjacente,  $|\mathcal{TD}(P_{j-1})|$ , da triangulação  $\mathcal{TD}(P_{j-1})$  na qual  $p$  é inserido. Se  $[a, b]$  também pertence a  $\partial(|\mathcal{TD}(P_j)|)$ , então a aresta  $[a, b]$  *não deveria ser trocada mesmo no caso em que  $p$  pertence a  $\text{int}(C_{acb})$* . Mas, se a aresta  $[a, b]$  não for substituída pela aresta  $[c, p]$  quando  $p$  pertence a  $\text{int}(C_{acb})$ , a triangulação  $\mathcal{TD}_j$  não será Delaunay e, portanto, não será igual a  $\mathcal{TD}_{S_j}$ . Tudo indica que chegamos a uma contradição! Este aparente obstáculo será transposto em breve. No momento, vamos voltar a nossa atenção para a solução de um problema anterior



ao obstáculo: *determinar se  $[a, b]$  pertence à fronteira,  $\partial(|\mathcal{TD}(P_j)|)$ , de  $|\mathcal{TD}(P_j)|$ .*

Para resolver o problema acima, vamos considerar os casos em que (1) nenhum ponto em  $\{a, b, c\}$  é ponto especial, (2) exatamente um ponto em  $\{a, b, c\}$  é ponto especial e (3) exatamente dois pontos em  $\{a, b, c\}$  são pontos especiais. O caso em que todos os pontos em  $\{a, b, c\}$  são especiais nunca ocorre, pois o ponto  $p$  não é especial e deve pertencer ao interior do triângulo,  $\tau_0$ , cujos vértices são os pontos especiais. No caso (1), podemos concluir que  $[a, b]$  não pode pertencer a  $\partial(|\mathcal{TD}(P_j)|)$ , uma vez que  $|\mathcal{TD}(P_j)| = FC(P_j)$ ,  $p, a, b, c \in P_j$  e há um ponto em cada lado da reta por  $a$  e  $b$  orientada de  $a$  para  $b$ . No caso (3), também podemos concluir que  $[a, c]$  não pode pertencer a  $\partial(|\mathcal{TD}(P_j)|)$ , pois ela incide sobre um ponto especial e, portanto, sobre um ponto que não está em  $P_j$ . O mesmo pode ser dito da aresta  $[c, p]$ . O que nos resta é analisar o caso (2). Para tal, consideramos dois sub-casos mutuamente exclusivos: (i)  $c$  é o único ponto especial de  $\{a, b, c\}$  (ii)  $a$  ou  $b$  é o único ponto especial de  $\{a, b, c\}$  (ver Figura 4.37).

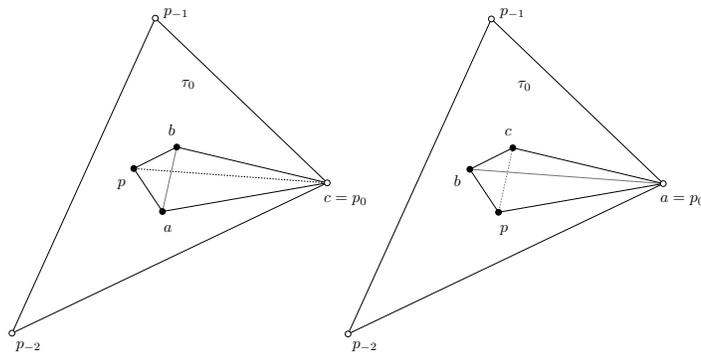
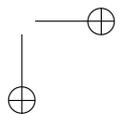
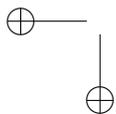
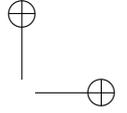
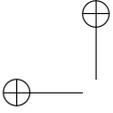


Figura 4.37: O ponto  $c$  é o único ponto especial de  $\{a, b, c\}$  (esquerda, caso 2(i)). O ponto  $a$  é o único ponto especial de  $\{a, b, c\}$  (direita, caso 2(ii)).

Considere o caso 2(i) e suponha que toda aresta da triangulação de Delaunay,  $\mathcal{TD}(P_{j-1})$ , de  $P_{j-1}$  que está em  $\partial(|\mathcal{TD}(P_{j-1})|)$  está

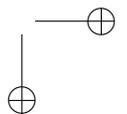
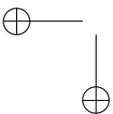


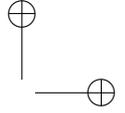
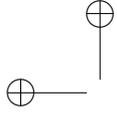


em  $\mathcal{TD}_{j-1}$  *imediatamente antes* da inserção de  $p = p_j$  em  $\mathcal{TD}_{j-1}$ . Se o quadrilátero  $\square(acbp)$  não for convexo, então a aresta  $[a, b]$  é localmente Delaunay e não é substituída por  $[c, p]$ . Caso contrário, a aresta  $[a, b]$  pode não ser localmente Delaunay e, neste caso, ser substituída por  $[c, p]$ . Então, suponha que  $\square(acbp)$  é convexo. Seja  $\ell_{ab}$  a reta por  $a$  e  $b$  orientada de  $a$  para  $b$ . Note que  $p$  e  $c$  estão em lados opostos de  $\ell_{ab}$ . Como o interior de  $\square(acbp)$  não contém nenhum ponto de  $P_j$ , a aresta  $[c, p]$  tem de intersectar  $\partial(|\mathcal{TD}(P_{j-1})|)$  em um ponto interior dela, pois  $|\mathcal{TD}(P_{j-1})| = FC(P_{j-1})$ . Mas, por hipótese,  $\partial(|\mathcal{TD}(P_{j-1})|)$  é uma união de arestas de  $\mathcal{TD}_{j-1}$ , o que nos diz que  $[a, b]$  pertence a  $\partial(|\mathcal{TD}(P_{j-1})|)$ , pois ela é a única aresta de  $\mathcal{TD}_{j-1}$  intersectada por  $[c, p]$ . Isto implica que  $[a, b]$  não pode ser substituída por  $[c, p]$ , *mesmo que ela não seja localmente Delaunay em  $\mathcal{TD}_j$ .*

Considere agora o caso 2(ii) e, sem perda de generalidade, assuma que  $a$  é o único ponto especial entre  $a$ ,  $b$  e  $c$ . Novamente, suponha que toda aresta da triangulação de Delaunay,  $\mathcal{TD}(P_{j-1})$ , de  $P_{j-1}$  que está em  $\partial(|\mathcal{TD}(P_{j-1})|)$  está em  $\mathcal{TD}_{j-1}$  *imediatamente antes* da inserção de  $p = p_j$  em  $\mathcal{TD}_{j-1}$ . Se o quadrilátero  $\square(acbp)$  não for convexo, então a aresta  $[a, b]$  é localmente Delaunay e não é substituída por  $[c, p]$ . Caso contrário, a aresta  $[a, b]$  pode não ser localmente Delaunay e, neste caso, ser substituída por  $[c, p]$ . Então, suponha que  $\square(acbp)$  é convexo. Seja  $\ell_{cp}$  a reta por  $c$  e  $p$  orientada de  $c$  para  $p$ . Note que  $a$  e  $b$  estão em lados opostos de  $\ell_{cp}$ . Como o interior de  $\square(acbp)$  não contém nenhum ponto de  $P_j$ , a aresta  $[a, b]$  tem de intersectar  $\partial(|\mathcal{TD}(P_j)|)$  em um ponto interior dela, pois  $|\mathcal{TD}(P_j)| = FC(P_j)$ . Além disso, este ponto tem de ser o ponto de interseção entre  $[a, b]$  e  $[c, p]$ . Caso contrário, seria o ponto de interseção entre  $[a, b]$  e outra aresta  $\mu$  de  $\mathcal{TD}(P_{j-1})$  em  $\partial(|\mathcal{TD}(P_{j-1})|)$ . Mas, por hipótese, a aresta  $\mu$  deveria pertencer a  $\mathcal{TD}_{j-1}$ , o que é um absurdo, pois duas arestas de uma triangulação não podem se intersectar em um ponto interior de nenhuma delas. Logo,  $[a, b]$  deve ser substituída por  $[c, p]$ , sempre que  $\square(acbp)$  for convexo e *mesmo que ela seja localmente Delaunay em  $\mathcal{TD}_j$ .*

As observações sobre os casos 2(i) e 2(ii) sugerem que *podemos* substituir uma aresta que é localmente Delaunay por uma que não é — *caso 2(i)* — e de não substituir uma aresta que não é localmente Delaunay por uma que é — *casos 2(ii)*. Em princípio, o problema com





esta sugestão é que  $\mathcal{TD}_j$  pode não ser uma triangulação de Delaunay e, portanto, todos os resultados que derivamos para mostrar a corretude do Algoritmo 4.2 não são mais válidos. Felizmente, o cenário não é catastrófico como parece. Mais especificamente, observe que temos a liberdade de preescrever as coordenadas de  $p_0$ ,  $p_{-1}$  e  $p_{-2}$ . Então, vamos supor que as coordenadas de  $p_0$  sejam escolhidas de tal forma que  $p_0$  reside no exterior de todo círculo cuja circunferência é definida por três pontos de  $P$ . De forma análoga, escolhemos as coordenadas de  $p_{-1}$  e  $p_{-2}$  são escolhidas de tal forma que  $p_{-1}$  reside no exterior de todo círculo cuja circunferência é definida por três pontos de  $P \cup \{p_0\}$  e que  $p_{-2}$  reside no exterior de todo círculo cuja circunferência é definida por três pontos de  $P \cup \{p_0, p_{-1}\}$ . Afirmamos que  $\mathcal{TD}(S_j)$  contém todas as arestas e triângulos de  $\mathcal{TD}(P_j)$ . Além disso, afirmamos que a modificação do predicado *incircle* no Algoritmo 4.5 nos permite calcular uma triangulação,  $\mathcal{TD}_j$ , que é exatamente “igual” a  $\mathcal{TD}(S_j)$ , exceto pelo valor real das coordenadas atribuídas aos pontos  $p_0$ ,  $p_{-1}$  e  $p_{-2}$ . Isto porque queremos evitar o cálculo das coordenadas  $p_0$ ,  $p_{-1}$  e  $p_{-2}$  que satisfaçam as condições acima.

---

**Algoritmo 4.5** INCIRCLEMOD( $a, c, b, p$ )

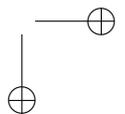
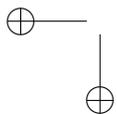
---

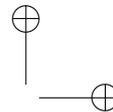
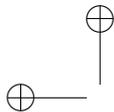
```

1: se nenhum de  $a$ ,  $c$  e  $b$  é especial então
2:   devolva INCIRCLE( $a, c, b, p$ )
3: fim se
4: se exatamente um de  $a$ ,  $b$  e  $c$  é especial então
5:   se  $c$  é especial então
6:     devolva falso {apenas  $c$  é especial}
7:   senão
8:     {exatamente um de  $a$  e  $b$  é especial e  $c$  não é}
9:      $d1 \leftarrow \text{LEFT}(c, b, p)$ 
10:     $d2 \leftarrow \text{LEFTON}(c, a, p)$ 
11:    {verdadeiro se, e somente se,  $\square(a, c, b, p)$  é convexo}
12:    devolva  $d1$  e ( not  $d2$  )
13:  fim se
14: fim se
15: devolva  $in(c) < \min\{in(a), in(b)\}$  { $a$  e  $c$  ou  $b$  e  $c$  são especiais}

```

---

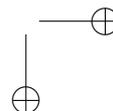
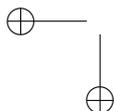




A linha 2 do Algoritmo 4.5 nos diz que o predicado modificado se comporta de mesma forma que o predicado original caso nenhum dos vértices de entrada seja especial. Isto está de acordo com a hipótese que diz que nenhum dos pontos especiais,  $p_0$ ,  $p_{-1}$  e  $p_{-2}$ , pode residir em um círculo cuja a circunferência é definida por três pontos de  $P$ . Logo, a aresta  $[a, b]$  é trocada pela aresta  $[c, p]$  se, e somente se, ela não é localmente Delaunay em  $\mathcal{TD}(S_j)$ . As linhas 4 a 14 tratam do caso 2. Em particular, a linha 6 lida com o caso 2(i). As observações que fizemos antes sobre o caso 2(i) nos dizem que a aresta  $[a, b]$  *jamais deve ser trocada* pela aresta  $[c, p]$  quando  $c$  for o único ponto especial de  $\{a, c, b\}$ . De fato, por hipótese, o ponto  $c$ , que é o único ponto especial em  $\{a, c, b, p\}$ , não reside no círculo cuja circunferência é definida por  $a$ ,  $b$  e  $p$ . Logo, a aresta  $[a, b]$  é localmente Delaunay em  $\mathcal{TD}(S_j)$  e, portanto, o resultado do Algoritmo 4.5 está correto. As linhas 8 a 12 lidam com o caso 2(ii). As linhas 9 e 10 determinam se o quadrilátero  $\square(a, c, b, p)$  é convexo e a aresta  $[c, p]$  substitui a aresta  $[a, b]$  em caso afirmativo. De fato, por hipótese, o ponto  $a$  (resp.  $b$ ), que é o único ponto especial em  $\{a, c, b, p\}$ , não reside no círculo cuja circunferência é definida por  $b$  (resp.  $a$ ),  $c$  e  $p$ . Logo, a aresta  $[c, p]$  é localmente Delaunay em  $\mathcal{TD}(S_j)$  e, portanto, o Algoritmo 4.5 também produz o resultado que desejamos no caso 2(ii).

No caso (3), temos apenas duas possibilidades para os pontos especiais (ver Figura 4.38): os dois pontos especiais são obrigatoriamente  $a$  e  $c$  ou  $b$  e  $c$ , pois a combinação  $a$  e  $b$  implicaria em  $c$  ou  $p$  ser um ponto no exterior do triângulo  $\tau_0 = [p_0, p_{-1}, p_{-2}]$ , o que não é possível. Em ambos os casos, devemos trocar a aresta  $[a, b]$  pela aresta  $[c, p]$  se, e somente se, o índice,  $in(a)$ , de  $a$  (resp. o índice,  $in(b)$ , de  $b$ ) é menor do que o índice,  $in(c)$ , de  $c$ , em que o índice,  $in(p_i)$ , de um ponto  $p_i$  é o inteiro  $i$ , com  $i \in \{-2, -1, 0\}$ , se  $p_i$  é ponto especial e o valor 1, caso contrário. Por hipótese, se  $p_j$  e  $p_k$  são quaisquer dois dos três pontos especiais tais que  $in(p_j) < in(p_k)$ , então  $p_j$  reside no exterior de todo círculo cuja circunferência é definida por quaisquer três pontos de  $P \cup \{p_0, p_k\}$ . Logo, a aresta  $[a, b]$  é localmente Delaunay em  $\mathcal{TD}(S_j)$  se, e somente se,  $in(c) < \min\{in(a), in(b)\}$ . Isto significa que a linha 15 do Algoritmo 4.5 devolve o resultado correto no caso (3).

Observe que a única diferença entre as triangulações  $\mathcal{TD}_{|P|}$  e



$\mathcal{TD}(S_{|P|})$  é puramente “geométrica”. A triangulação  $\mathcal{TD}_{|P|}$  é calculada pelo segundo passo do Algoritmo 4.2 usando as coordenadas de  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  da Subseção 4.6.1, enquanto  $\mathcal{TD}(S_{|P|})$  é a triangulação de Delaunay de  $S_{|P|}$  calculada com as coordenadas de  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  satisfazendo as condições que descrevemos acima. Pelo que acabamos de provar, as triangulações  $\mathcal{TD}_{|P|}$  e  $\mathcal{TD}(S_{|P|})$  exatamente os mesmos triângulos e arestas. No entanto, podemos imaginar que  $\mathcal{TD}_{|P|}$  foi obtida arrastando-se os vértices correspondentes aos pontos  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  em  $\mathcal{TD}(S_{|P|})$  para a posição dos vértices correspondentes aos pontos  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  em  $\mathcal{TD}_{|P|}$ . Em outras palavras, as duas triangulações contêm a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ , mas  $\mathcal{TD}_{|P|}$  pode não ser uma triangulação de Delaunay e isso se deve inteiramente às posições dos vértices correspondentes aos pontos  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ .

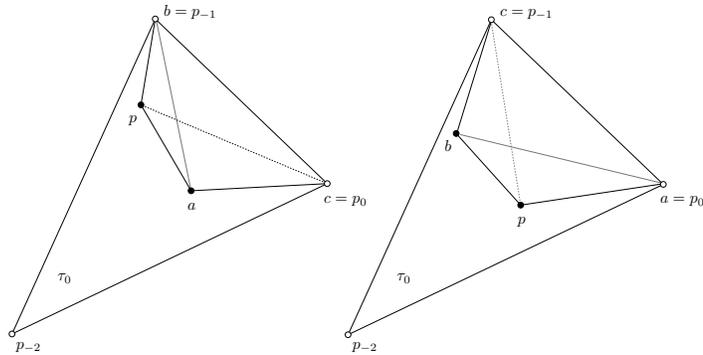
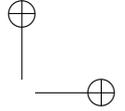
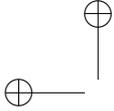


Figura 4.38: Os pontos  $b$  e  $c$  são os pontos especiais de  $\{a, b, c\}$  (esquerda). Os pontos  $a$  e  $c$  são os pontos especiais de  $\{a, b, c\}$  (direita).

O fato de  $\mathcal{TD}_{|P|}$  não ser necessariamente uma triangulação de Delaunay não é importante, pois as arestas e triângulos de  $\mathcal{TD}_{|P|}$  que não são Delaunay estão do “lado de fora” de  $\mathcal{TD}(P)$ , que é a triangulação que queremos calcular. O terceiro e último passo do Algoritmo 4.2 remove essas arestas e triângulos, juntamente com os vértices de  $\mathcal{TD}_{|P|}$  correspondentes a  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ . Isto pode ser feito pela simples remoção de todas as arestas e triângulos incidentes



em qualquer um dos pontos especiais. O resultado é a triangulação  $\mathcal{TD}(P)$ .

#### 4.6.4 Complexidade

Nesta seção, realizamos uma análise da complexidade de tempo *esperado* do Algoritmo 4.2. A nossa discussão abrange dois aspectos importantes do algoritmo, os quais não foram detalhados anteriormente. O primeiro é a geração de uma sequência *aleatória*,  $p_1, \dots, p_n$ , consistindo de todos os  $n$  pontos do conjunto,  $P$ , dado como entrada para o algoritmo (ver linha 1 do Algoritmo 4.2). O segundo é o procedimento que encontra um triângulo da triangulação corrente,  $\mathcal{TD}_{j-1}$ , que contém o ponto  $p_j$  a ser inserido em  $\mathcal{TD}_{j-1}$  para gerar  $\mathcal{TD}_j$ .

Para motivar nossa discussão sobre geração de uma sequência aleatória com os pontos de entrada, vamos considerar um conjunto,  $P$ , de  $n = 2m$  pontos de  $\mathbb{E}^2$ , com  $n, m \in \mathbb{N}$ , da seguinte forma: há  $m$  pontos,  $e_1, \dots, e_m$ , em  $P$  cujas coordenadas são da forma dada a seguir:

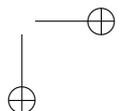
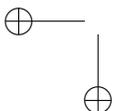
$$e_i = \left( -\frac{2 \cdot (i-1)}{m-1}, 0 \right),$$

para todo  $i = 1, \dots, m$ . Os  $m$  pontos restantes,  $d_1, \dots, d_m$ , são tais que

$$d_i = \left( \frac{2 \cdot (m-i)}{m-1}, 1 \right),$$

para todo  $i = 1, \dots, m$ . Se o Algoritmo 4.2 construísse a triangulação  $\mathcal{TD}(S_{|P|})$  inserindo os pontos  $p_{-2}, p_{-1}, p_0, e_1, \dots, e_m, d_1, \dots, d_m$ , nesta ordem, o número de trocas de arestas efetuado durante a inserção de *cada*  $d_i$  é  $m-1$ , pois o ponto  $d_i$  está conectado a cada  $e_j$  em  $\mathcal{TD}(S_{m+i})$ , para  $j = 1, \dots, m$  (ver Figura 4.39). Logo, o algoritmo incremental não parece melhor do que o Algoritmo 4.1 visto na Seção 4.5.

De fato, o exemplo que acabamos de ver representa o pior caso do Algoritmo 4.2 e, em relação ao desempenho no pior caso, o Algoritmo 4.2 não é diferente do Algoritmo 4.1. No entanto, se gerarmos a subsequência  $p_1, \dots, p_n$  de  $p_{-2}, p_{-1}, p_0, p_1, \dots, p_n$  através de uma permutação aleatória dos elementos de  $P$ , de modo que *cada permutação ocorra com a mesma probabilidade*, mostraremos que o tempo de



execução *esperado* do Algoritmo 4.2, no pior caso, é  $\mathcal{O}(n \lg n)$ . O cálculo de uma permutação aleatória dos pontos de  $P$  tem o efeito de fazer com que o Algoritmo 4.2 se torne *probabilístico* [89]. Isto significa que o tempo de execução do algoritmo passa a ser uma *variável aleatória*. É por isso que usamos o termo *tempo de execução esperado*, pois não há como classificar *a priori* uma entrada,  $P$ , do algoritmo como representativa do pior caso, já que a ordem de inserção dos pontos é aleatória.

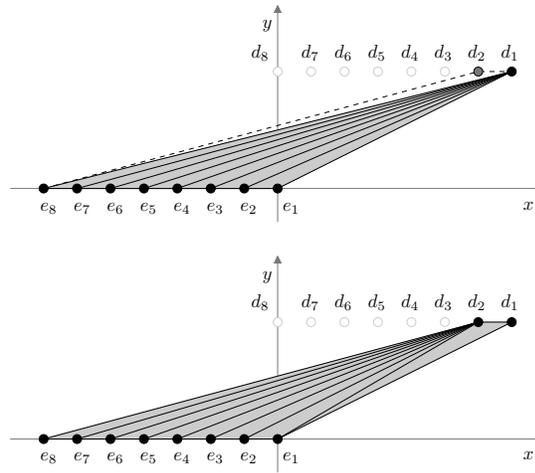
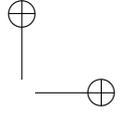
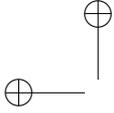


Figura 4.39: Um conjunto de entrada representativo do pior caso do Algoritmo 4.2. Quando o ponto  $d_2$  é inserido na triangulação  $\mathcal{TD}_9$ , cada aresta conectando  $d_1$  a um  $e_j$ , para todo  $j = 2, \dots, 8$ , é substituída por uma aresta conectando  $d_2$  a  $e_j$ , totalizando 7 trocas de arestas.

Por outro lado, se removêssemos a aleatorização da sequência  $p_{-2}, p_{-1}, p_0, p_1, \dots, p_n$ , a ordem de inserção dos pontos seria determinada pela ordem em que os pontos de  $P$  são fornecidos como entrada para o Algoritmo 4.2. Neste caso, poderíamos classificar uma entrada,  $P$ , como sendo representativa do pior caso do algoritmo, tal qual o exemplo dado na Figura 4.39, e o algoritmo seria *determinístico*. É



importante ressaltar que o emprego de aleatorização não faz com que o pior caso desapareça e nem reduz a complexidade do pior caso da versão determinística do algoritmo. O benefício da aleatorização, no caso particular do Algoritmo 4.2, é o de permitir que analisemos a complexidade de pior caso do tempo de execução *esperado*. Se as entradas que caracterizam o pior caso da versão determinística do algoritmo ocorrem com uma probabilidade muito baixa, o tempo de execução *esperado* passa a ser mais relevante<sup>5</sup> do que o tempo de pior caso. O emprego de aleatorização para transformar um algoritmo determinístico em um algoritmo probabilístico, tal como fizemos com o Algoritmo 4.2 ao gerar uma permutação aleatória de  $P$ , foi empregada com bastante sucesso em vários problemas de geometria computacional [14, 93].

O Algoritmo 4.6 faz uso de um *gerador de números pseudo-aleatórios* para gerar uma permutação de uma sequência qualquer de números armazenada em um arranjo. Em geral, as principais linguagens de programação possuem uma função que exerce o papel de gerador de números pseudo-aleatórios. No Algoritmo 4.6, este gerador é representado pela função  $\text{RANDOM}()$ , que recebe dois inteiros,  $a$  e  $b$ , tais que  $a < b$ , como entrada e devolve um número inteiro,  $x$ , com  $x \in [a, b] \subset \mathbb{Z}$ , tal que, para qualquer inteiro  $d$  no intervalo  $[a, b]$ , temos

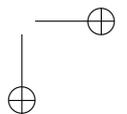
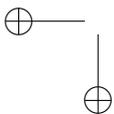
$$\Pr[x = d] = \frac{1}{b - a + 1}.$$

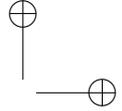
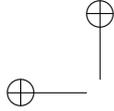
Isto é, o número  $x$  pode ser qualquer inteiro de  $[a, b]$  com a mesma probabilidade. Sendo assim, a linha 1 do Algoritmo 4.2 é a chamada

PERMUTA( $P$ ).

O tempo de execução do Algoritmo 4.2 pode ser escrito como a soma entre o (1) total de tempo gasto para localizar um triângulo contendo o ponto  $p_j$  na triangulação  $\mathcal{TD}_{j-1}$ , para  $j = 1, \dots, n$  e o (2) total de tempo gasto para inserir o ponto na triangulação atual e para convertê-la na triangulação de Delaunay. Note que (2) é diretamente proporcional ao total de trocas de arestas realizado pelo algoritmo para criar  $\mathcal{TD}_0, \dots, \mathcal{TD}_{|P|}$ . Logo, se obtivermos uma cota superior para o total de trocas de arestas, teremos uma cota para o tempo em

<sup>5</sup>Pelo menos, em termos práticos.






---

**Algoritmo 4.6** PERMUTA( $P$ )

---

```

1:  $n \leftarrow |P|$ 
2: para  $i = 1, \dots, n$  faça
3:    $j \leftarrow \text{RANDOM}(i, n)$ 
4:    $t \leftarrow P[i]$ 
5:    $P[i] \leftarrow P[j]$ 
6:    $P[j] \leftarrow t$ 
7: fim para

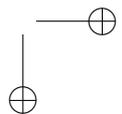
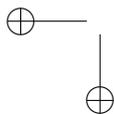
```

---

(2). Na obtenção de uma cota superior para (2), levamos em conta o total de *triângulos de Delaunay* criados pelo algoritmo, pois este total é mais fácil de ser obtido e é sempre maior ou igual ao total de trocas de arestas. O termo “triângulo de Delaunay” (ver Observação 4.8) designa um triângulo que pertence a alguma triangulação em

$$\mathcal{TD}_0, \dots, \mathcal{TD}_{|P|}.$$

De forma geral, na geração de  $\mathcal{TD}_j$  a partir de  $\mathcal{TD}_{j-1}$ , toda troca de aresta elimina um triângulo que pertence a  $\mathcal{TD}_{j-1}$  — e, portanto, é um triângulo de Delaunay — e um triângulo “intermediário”, incidente em  $p_j$ , que não pertence a  $\mathcal{TD}_{j-1}$  e nem a  $\mathcal{TD}_j$ . Logo, como uma mesma aresta não é considerada para troca mais de uma vez (ver Lema 4.21), os triângulos removidos pela troca de aresta também não surgem novamente durante a mesma execução do Algoritmo 4.2. Logo, podemos associar cada troca de aresta a um único triângulo de Delaunay removido de alguma triangulação. Daí, o número total de trocas de arestas é limitado superiormente pelo número total,  $\beta$ , de triângulos de Delaunay criados pelo algoritmo. Provaremos a seguir que o valor esperado,  $E[\beta]$ , de  $\beta$  está em  $\mathcal{O}(n)$ . Uma noção crucial usada pela demonstração é a de *escopo* de um triângulo  $\sigma$ , que é definido como o número de pontos em  $S = \{p_{-2}, p_{-1}, p_0\} \cup P$  que residem no interior do círculo circunscrito a  $\sigma$  (ver Figura 4.40). O escopo,  $\Psi(\sigma)$ , de um triângulo  $\sigma$  só pode assumir valores no intervalo que vai de 0 a  $n$ , pois  $S$  contém  $n + 3$  pontos e três deles são os vértices de  $\sigma$ . Observe que  $\sigma$  se refere a qualquer triângulo cujos vértices são pontos de  $S$ . Observe também que  $\Psi(\tau_0) = n$  e que se  $\Psi(\sigma)$  é igual a zero, então  $\sigma$  é um triângulo de  $\mathcal{TD}(S)$ .



Um triângulo  $\sigma = [p_h, p_k, p_l]$  surge como um dos triângulos de alguma das triangulações em  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$  se, e somente se, os pontos  $p_h, p_k$  e  $p_l$  precedem, na sequência de inserção, *todos* os pontos de  $S$  que pertencem ao interior do círculo cuja circunferência passa por  $p_h, p_k$  e  $p_l$ . Logo, quanto menor for o escopo,  $\Psi(\sigma)$ , de  $\sigma$  maior a probabilidade de  $\sigma$  surgir como um dos triângulos de alguma das triangulações em  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$ . Esta probabilidade é fornecida pelo Lema 4.25 dado abaixo. Para facilitar o cálculo desta probabilidade, *a prova do lema assume que a permutação aleatória é uma permutação de*

$$\{p_{-2}, p_{-1}, p_0\} \cup P.$$

Sabemos que este não é o caso, pois os pontos especiais são sempre os três primeiros pontos inseridos. Porém, podemos mostrar que as cotas superiores que derivadas do Lema 4.25 se mantêm (Problema 4.46).

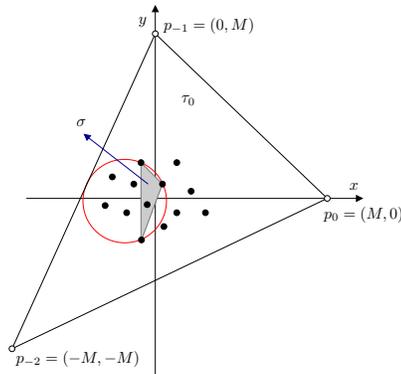
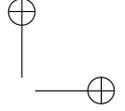
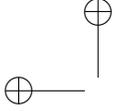


Figura 4.40: O escopo,  $\Psi(\sigma)$ , do triângulo  $\sigma$  é 5.

**Lema 4.25.** *Seja  $\sigma$  um triângulo definido por três pontos de  $S = \{p_{-2}, p_{-1}, p_0\} \cup P$  e tal que  $\Psi(\sigma) = k$ . Então, a probabilidade que  $\sigma$  é um triângulo de Delaunay (isto é, de alguma triangulação em  $\mathcal{TD}_0, \dots, \mathcal{TD}_{|P|}$ ), durante a execução do Algoritmo 4.2 na entrada  $P$ , é*

$$\frac{6}{(k+1)(k+2)(k+3)}.$$



*Demonstração.* Seja  $n$  o número de pontos de  $P$ . Então, o número de elementos de  $S$  é  $m = n + 3$ . O conjunto  $S$  pode ser particionado em três subconjuntos: (a) o conjunto que consiste dos três vértices,  $p_h, p_k$  e  $p_l$ , de  $\sigma$ , (b) o conjunto que consiste dos  $k$  pontos,  $q_1, \dots, q_k$ , de  $S$  que pertencem ao interior do círculo circunscrito a  $\sigma$  e (c) os  $m - (k + 3)$  elementos de  $S$  que não estão em (a) nem em (b). O número de maneiras de distribuir os elementos do subconjunto em (c) entre os elementos da sequência  $p_h, p_k, p_l, q_1, \dots, q_k$  é dado pela fórmula

$$\frac{m!}{(m - (k + 3))! \cdot (k + 3)!}.$$

Mas, se permitirmos que os três pontos,  $p_h, p_k$  e  $p_l$ , do subconjunto em (a) permutem entre si, que os  $k$  pontos,  $q_1, \dots, q_k$ , do subconjunto em (b) permutem entre si e que os  $m - (k + 3)$  elementos restantes de  $S$  no subconjunto (c) permutem entre si, então obtemos um total de

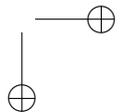
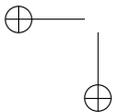
$$\frac{m!}{(m - (k + 3))! \cdot (k + 3)!} \cdot 3! \cdot k! \cdot (m - (k + 3))! = \frac{m! \cdot 3! \cdot k!}{(k + 3)!},$$

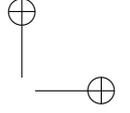
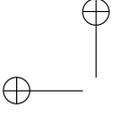
que é exatamente o número de sequências de pontos de  $S$  em que nenhum de  $p_h, p_k$  e  $p_l$  é precedido por qualquer ponto de  $S$  que pertença ao interior do círculo circunscrito a  $\sigma$ . Logo, a probabilidade que  $\sigma$  surgirá como um triângulo de alguma triangulação da sequência  $\mathcal{TD}_0, \dots, \mathcal{TD}_{|P|}$ , durante a execução do Algoritmo 4.2 na entrada  $P$ , é

$$\frac{\frac{m! \cdot 3! \cdot k!}{(k + 3)!}}{m!} = \frac{3! \cdot k!}{(k + 3)!} = \frac{6}{(k + 1)(k + 2)(k + 3)}.$$

□

Denote por  $T_k$  o conjunto de todos os triângulos  $\sigma$  com vértices em  $S$  tais que  $\Psi(\sigma) = k$ , para  $0 \leq k \leq n$ . Seja  $T_{\leq k} = \bigcup_{j=0}^k T_j$ ; isto é,  $T_{\leq k}$  é o conjunto de triângulos  $\sigma$  com vértices em  $S$  tais que  $\Psi(\sigma) \leq k$ , para  $0 \leq k \leq n$ . Nós podemos mostrar que o número,  $\|T_{\leq k}\|$ , de triângulos em  $T_{\leq k}$  está em  $\mathcal{O}(n(k + 1)^2)$  [59]. Usando esta cota superior e o Lema 4.25, estabelecemos o resultado que queremos; a saber:





**Teorema 4.26.** *O número esperado de triângulos de Delaunay (isto é, triângulos de alguma das triangulações em  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$ ) que são criados, durante a execução do Algoritmo 4.2 na entrada  $P$ , está em  $O(n)$ .*

*Demonstração.* Seja  $\Pr(\sigma)$  a probabilidade que o triângulo  $\sigma$ , com vértices em  $S = \{p_{-2}, p_{-1}, p_0\} \cup P$ , surja como um triângulo em algumas das triangulações da sequência  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$ , durante a execução do Algoritmo 4.2 para  $P$ . Então, o valor esperado,  $E[\beta]$ , do número,  $\beta$ , total de triângulos de Delaunay criados durante a execução do Algoritmo 4.2 na entrada  $P$  é, por definição de valor esperado, dado por

$$E[\beta] = \sum_{\sigma \in \Lambda} 1 \cdot \Pr(\sigma),$$

em que  $\Lambda$  é o conjunto de todos os triângulos com vértices em  $S$ . Mas,

$$\sum_{\sigma \in \Lambda} 1 \cdot \Pr(\sigma) = \sum_{j=0}^n \sum_{\sigma \in T_j} \Pr(\sigma) = \sum_{j=0}^n \sum_{\sigma \in T_j} \frac{6}{(j+1)(j+2)(j+3)}, \quad (4.2)$$

em que

$$\sum_{\sigma \in T_j} \frac{6}{(j+1)(j+2)(j+3)} = \frac{6 \cdot \|T_j\|}{(j+1)(j+2)(j+3)}.$$

Daí,

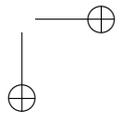
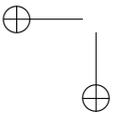
$$E[\beta] = 6 \cdot \left( \sum_{j=0}^n \frac{\|T_j\|}{(j+1)(j+2)(j+3)} \right). \quad (4.3)$$

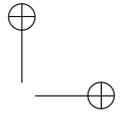
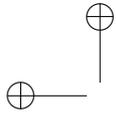
Podemos reescrever o lado direito da Equação (4.3) como a soma

$$\|T_0\| + 6 \cdot \left( \sum_{j \geq 1}^n \frac{\|T_{\leq j}\|}{(j+1)(j+2)(j+3)} - \frac{\|T_{\leq j-1}\|}{(j+1)(j+2)(j+3)} \right), \quad (4.4)$$

em que

$$\frac{\|T_{\leq (j-1)}\|}{(j)(j+1)(j+2)} - \frac{\|T_{\leq (j-1)}\|}{(j+1)(j+2)(j+3)}$$





é

$$\frac{\|T_{\leq(j-1)}\|}{(j)(j+1)(j+2)(j+3)}.$$

Logo, a soma telescópica na Equação (4.4) pode ser escrita como

$$18 \cdot \left( \sum_{j=0}^n \frac{\|T_{\leq(j)}\|}{(j+1)(j+2)(j+3)(j+4)} \right). \quad (4.5)$$

Como  $\|T_{\leq(j)}\| \in \mathcal{O}(n(j+1)^2)$ , a soma na Equação (4.5) está em

$$\mathcal{O} \left( n \cdot \sum_{j=0}^n \frac{1}{(j+3)^2} \right),$$

ou seja,

$$E[\beta] \in \mathcal{O}(n),$$

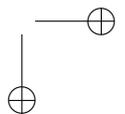
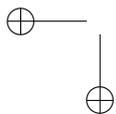
pois

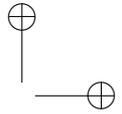
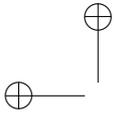
$$\sum_{j=0}^n \frac{1}{(j+3)^2} < \sum_{j=0}^n \frac{1}{(j+1)^2} < \sum_{j=1}^{\infty} \frac{1}{j^2} < 2.$$

□

O Teorema 4.26 nos permite concluir que o total de trocas de arestas efetuado pelo Algoritmo 4.2 está em  $\mathcal{O}(n)$ . Como dissemos antes, esta cota superior também limita superiormente o tempo total *esperado* gasto pelo Algoritmo 4.2 para inserir pontos e converter triangulações na triangulação de Delaunay. O que nos resta é determinar uma cota superior para o tempo total *esperado* gasto pelo Algoritmo 4.2 para localizar os triângulos contendo os pontos a serem inseridos.

O problema de encontrar um triângulo de uma dada triangulação que contém o ponto dado é denominado de *localização de ponto*. Esta tarefa é o “gargalo” do Algoritmo 4.2 com relação a tempo de execução. Guibas, Knuth e Sharir propuseram uma estrutura de dados,  $\mathcal{D}$ , que nos permite localizar um ponto de forma simples e eficiente [59]. A ideia é manter uma hierarquia formada pelas triangulações resultantes de cada criação de triângulos. Mais especificamente, sempre que um triângulo  $\sigma$  é eliminado de uma triangulação, devido a uma inserção de um ponto ou a uma troca de





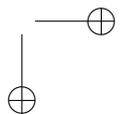
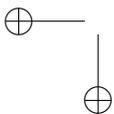
aresta, o triângulo  $\sigma$ , removido da triangulação corrente, é marcado como “substituído” em  $\mathcal{D}$  e um apontador de  $\sigma$  para cada um dos dois (ou três) novos triângulos é inserido em  $\mathcal{D}$  (ver Figura 4.41). Note que cada nível da hierarquia corresponde a exatamente uma mudança estrutural realizada pelo Algoritmo 4.2 em alguma das triangulações em

$$\mathcal{T}\mathcal{D}_0, \dots, \mathcal{T}\mathcal{D}_n.$$

A mudança estrutural pode ser (i) a criação de um único triângulo (isto é,  $\tau_0$ ), (ii) a criação de três (resp. quatro) triângulos, devido à inserção de um ponto em uma triangulação  $\mathcal{T}\mathcal{D}_j$ , seguida da remoção de um (resp. dois) triângulos e (iii) a criação de dois novos triângulos e a remoção de dois outros devido a uma troca de aresta. A mudança (i) inicializa a estrutura de dados,  $\mathcal{D}$ , com a criação do primeiro nível da hierarquia: o nível que contém  $\tau_0 = [p_{-2}, p_{-1}, p_0]$ . A mudança (ii) acarreta a criação de um novo nível na hierarquia e três (resp. quatro) apontadores. O novo nível contém os três (resp. quatro) triângulos novos. Cada apontador conecta um triângulo removido a um triângulo criado. A mudança (iii) também acarreta a criação de um novo nível na hierarquia e dois apontadores, conectando, cada um deles, um triângulo removido a triângulo criado. Como todo apontador aponta para um triângulo criado pelo Algoritmo 4.2, e como um mesmo triângulo só é criado uma vez (isso é uma consequência imediata do Lema 4.21), o Teorema 4.26 implica que o número esperado de apontadores e o próprio tamanho esperado de  $\mathcal{D}$  também estão em  $\mathcal{O}(n)$ .

**Observação 4.9.** *A estrutura de dados hierárquica,  $\mathcal{D}$ , mantém todos os triângulos criados pelo Algoritmo 4.2: os triângulos de Delaunay (isto é, de  $\mathcal{T}\mathcal{D}_j$ , para algum  $j \in \{0, \dots, n\}$ ) e os intermediários.*

Quando um ponto,  $p_j$ , de  $P$  é inserido na triangulação  $\mathcal{T}\mathcal{D}_{j-1}$ , o algoritmo localiza o triângulo de  $\mathcal{T}\mathcal{D}_{j-1}$  contendo  $p_j$  usando  $\mathcal{D}$ . Para tal, o algoritmo percorre  $\mathcal{D}$  a partir de seu primeiro nível (aquele contendo  $\tau_0$ ) e visita todos os triângulos — tanto triângulos de Delaunay quanto triângulos intermediários — que contêm  $p_j$  em ordem cronológica do surgimento desses triângulos. Para mudar de nível, o algoritmo precisa determinar, entre os triângulos que foram criados a partir de um triângulo,  $\sigma$ , do nível atual, aquele que contém o ponto



$p_j$ . Há, no máximo, três triângulos acessíveis, via apontadores, a partir de  $\sigma$  e, portanto, a determinação de qual triângulo do próximo nível contém  $p_j$  pode ser feita com, no máximo, três testes. Cada teste requer tempo constante e pode ser realizado, cada um, com no máximo três chamadas ao predicado LEFTON() (ver Apêndice A).

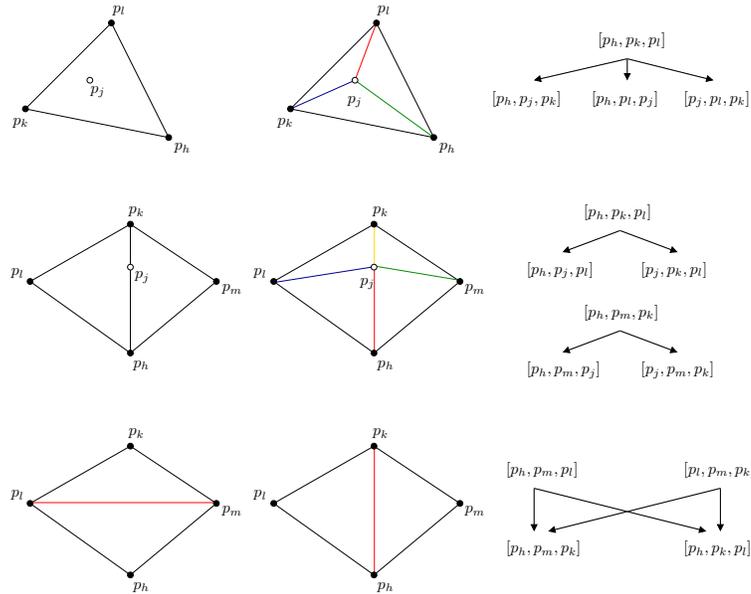
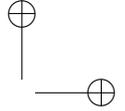
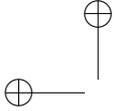


Figura 4.41: Todas as possíveis mudanças estruturais (além da criação de  $\tau_0$ ) realizadas pelo Algoritmo 4.2 nas triangulações  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$  e representadas pela estrutura de dados hierárquica,  $\mathcal{D}$ .

Pelo que observamos acima, o tempo para encontrar o triângulo contendo  $p_j$  na triangulação atual é proporcional ao número de visitas a triângulos de  $\mathcal{D}$  desde o triângulo  $\tau_0$ , no primeiro nível. Desejamos determinar o número *esperado* de triângulos visitados. Para tal, considere um triângulo,  $[a, b, c]$ , visitado durante a busca por um triângulo de  $\mathcal{TD}_{j-1}$  contendo  $p_j$ . Suponha que  $[a, b, c]$  não seja um



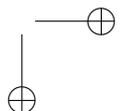
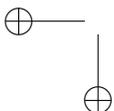
triângulo de  $\mathcal{TD}_{j-1}$ . Por construção, o triângulo  $[a, b, c]$  foi removido de alguma triangulação anterior à inserção de  $p_j$  em  $\mathcal{TD}_{j-1}$ , devido

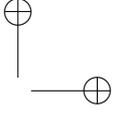
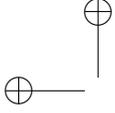
- (1) à inserção de um ponto  $p_k$  localizado em  $[a, b, c]$ , com  $k < j$ , o que fez com que surgissem de dois a três novos triângulos; ou
- (2) a uma troca de arestas que substituiu  $[a, b, c]$  e um triângulo adjacente, digamos  $[a, b, d]$ , por dois novos triângulos:  $[a, c, d]$  e  $[b, d, c]$ .

No caso (1), sabemos que o círculo inscrito a  $[a, b, c]$  não contém nenhum ponto de  $S_k = \{p_{-2}, p_{-1}, p_0, p_1, \dots, p_{k-1}\}$  em seu interior. Isto é, o triângulo  $[a, b, c]$  pertence a triangulação de Delaunay,  $\mathcal{TD}(S_k)$ , de  $S_k$  e, portanto, é um triângulo de Delaunay. No caso (2), a aresta,  $[a, b]$ , comum a  $[a, b, c]$  e  $[a, b, d]$ , não é localmente Delaunay. Logo, o ponto  $d$  pertence ao interior do círculo circunscrito ao triângulo  $[a, b, c]$  e o ponto  $a$  pertence ao interior do círculo circunscrito ao triângulo  $[a, b, d]$ . Além disso, um dos dois vértices,  $a$  ou  $d$ , foi o vértice inserido na triangulação da qual  $[a, b, c]$  e  $[a, b, d]$  foram removidos e exatamente um desses dois triângulos é de Delaunay. Em ambos os casos, o ponto  $p_j$  pertence ao interior dos círculos circunscritos aos triângulos  $[a, b, c]$  e  $[a, b, d]$ . Daí, tanto no caso (1) quanto no caso (2), podemos associar um único triângulo *de Delaunay*, removido em um estágio anterior à inserção de  $p_j$ , à visita ao triângulo  $[a, b, c]$  que ocorre durante a localização de  $p_j$  em  $\mathcal{TD}_{j-1}$ . No caso (1), este triângulo é o próprio  $[a, b, c]$ . No caso (2), este triângulo é o próprio  $[a, b, c]$  se  $[a, b, c]$  é um triângulo de Delaunay; se ele não for, então  $[a, b, d]$  é. Daí, a visita a  $[a, b, c]$  é associada ao triângulo  $[a, b, d]$ .

Seja  $\mu$  um triângulo de Delaunay tal que  $\Psi(\mu) = k$ . Então, por definição de  $\Psi$ , o triângulo  $\mu$  contém exatamente  $k$  pontos de  $S$  no interior do círculo circunscrito a ele. Pelo que acabamos de observar, o triângulo  $\mu$  pode ser associado a, no máximo,  $k$  visitas, pois um mesmo ponto de  $S$  não é localizado mais de uma vez em uma mesma execução de Algoritmo 4.2. Logo, o *total*  $t$  de visitas a triângulos de  $\mathcal{D}$  durante toda a execução do Algoritmo 4.2 é limitado superiormente por

$$n + \sum_{\mu \in \Lambda} \Psi(\mu),$$





em que  $\Lambda$  denota o conjunto de todos os triângulos de Delaunay visitados durante uma execução do Algoritmo 4.2. O termo  $n$  leva em conta a visita ao triângulo contendo o ponto a ser inserido na triangulação em que o ponto será inserido. Como  $\Lambda \subseteq \Lambda'$ , em que  $\Lambda'$  é o conjunto de todos os triângulos de Delaunay criados pelo algoritmo, temos

$$t \leq n + \sum_{\mu \in \Lambda'} \Psi(\mu).$$

Mas,

$$E[t] \leq E \left[ n + \sum_{\mu \in \Lambda'} \Psi(\mu) \right] = n + \sum_{j=0}^n \sum_{\mu \in T_j} j \cdot \Pr(\mu),$$

em que  $\Pr(\mu)$  é a probabilidade que  $\mu$  surja como um triângulo de Delaunay, ou seja, um triângulo de alguma triangulação em  $\mathcal{TD}_0, \dots, \mathcal{TD}_n$ . Pelo mesmo argumento da prova do Teorema 4.26, temos:

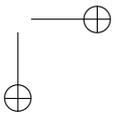
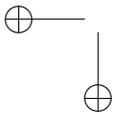
$$\begin{aligned} \sum_{j=0}^n \sum_{\mu \in T_j} j \cdot \Pr(\mu) &= 6 \cdot \left( \sum_{j=0}^n \frac{j \cdot \|T_j\|}{(j+1)(j+2)(j+3)} \right) \\ &= \frac{\|T_0\|}{4} + 6 \cdot \left( \sum_{j=1}^n \frac{j \cdot \|T_{\leq j}\| - j \cdot \|T_{\leq (j-1)}\|}{(j+1)(j+2)(j+3)} \right) \\ &= \frac{\|T_0\|}{4} + 12 \cdot \left( \sum_{j=1}^n \frac{(j-1) \cdot \|T_{\leq j}\|}{(j+1)(j+2)(j+3)(j+4)} \right) \\ &\leq 12 \cdot \left( \sum_{j=0}^n \frac{\|T_{\leq j}\|}{(j+2)(j+3)(j+4)} \right). \end{aligned}$$

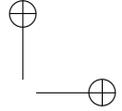
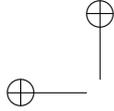
Como

$$\|T_{\leq j}\| \in \mathcal{O}(n(j+1)^2),$$

temos que

$$\frac{\|T_{\leq j}\|}{(j+2)(j+3)(j+4)} \in \mathcal{O}\left(\frac{n}{j+1}\right)$$





e, portanto,

$$12 \cdot \left( \sum_{j=0}^n \frac{\|T_{\leq j}\|}{(j+2)(j+3)(j+4)} \right) \in \mathcal{O}(n \lg n).$$

Logo,

$$E[t] \in \mathcal{O}(n + n \lg n) = \mathcal{O}(n \lg n).$$

Isto é, o total de tempo esperado gasto para localizar todos os pontos inseridos pelo Algoritmo 4.2 está em  $\mathcal{O}(n \lg n)$ . Então, a soma dos totais de tempo esperado para localizar todos os pontos e inserir os pontos localizados é  $\mathcal{O}(n \lg n) + \mathcal{O}(n) = \mathcal{O}(n \lg n)$ . Logo, temos o seguinte:

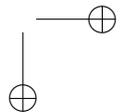
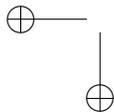
**Teorema 4.27.** *O tempo de execução esperado do Algoritmo 4.2, quando a entrada é o conjunto  $P$ , está em  $\mathcal{O}(n \lg n)$ , em que  $n = |P|$ .*

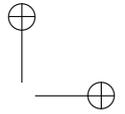
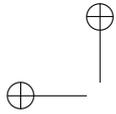
## 4.7 Cota inferior

A exemplo do que fizemos para o problema do fecho convexo na Seção 3.5, também derivaremos uma cota inferior para o problema de determinar a triangulação de Delaunay de um dado subconjunto,  $P$ , de  $n$  pontos de  $\mathbb{E}^2$ . Para tal, também reduziremos o problema da ordenação de  $n$  números para o problema de construir uma triangulação de  $P$ , que não é necessariamente a triangulação de Delaunay de  $P$ . Mais especificamente, seja  $U = (u_1, \dots, u_n)$  uma sequência de  $n$  números reais positivos. Tal sequência é uma entrada para o problema da ordenação. Dada  $S$ , vamos construir um conjunto  $P$  tal que

$$P = \{(0, 0)\} \cup \left( \bigcup_{i=1}^n \{(u_i, 1)\} \right).$$

A construção de  $P$  a partir de  $S$  pode ser feita com  $\mathcal{O}(n)$  operações primitivas. O conjunto  $P$  admite uma única triangulação,  $\mathcal{T}(P)$  (ver Figura 4.42), que também é a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ . Suponha que utilizemos qualquer algoritmo  $\mathcal{A}$  para calcular  $\mathcal{T}(P)$ . Dada  $\mathcal{T}(P)$  em uma estrutura de dados que nos permita realizar consultas sobre as relações de adjacência entre os elementos de





$\mathcal{T}(P)$  (por exemplo, a estrutura *quadedge* [58]), percorremos a fronteira,  $\partial(|\mathcal{T}(P)|)$ , do espaço subjacente,  $|\mathcal{T}(P)|$ , de  $\mathcal{T}(P)$  e obtemos, com  $\mathcal{O}(n)$  operações, as coordenadas dos pontos de  $P$  ordenadas em relação à abscissa. Como o problema da ordenação por comparações de chaves requer  $\Omega(n \lg n)$  operações primitivas para produzir uma sequência ordenada a partir de  $S$ , o Teorema 3.9 nos diz que o problema da triangulação requer, pelo menos,  $n \lg n - \mathcal{O}(n)$  operações primitivas para ser resolvido. Como  $(n \lg n - \mathcal{O}(n)) \in \Omega(n \lg n)$ , podemos concluir que se a complexidade de pior caso de  $\mathcal{A}$  é dada por

$$f : \mathbb{N} \rightarrow \mathbb{R},$$

então

$$f(n) \in \Omega(n \lg n).$$

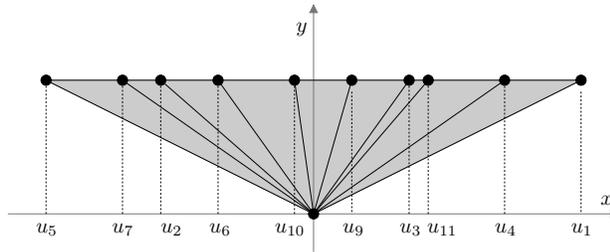
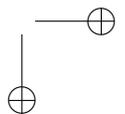
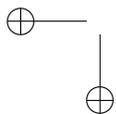
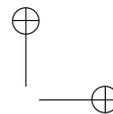
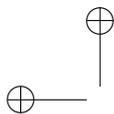


Figura 4.42: Transformação de uma entrada do problema da ordenação na entrada do problema da triangulação para reduzir aquele a este.

O que acabamos de provar nos diz que, no pior caso, qualquer algoritmo para construir a triangulação de Delaunay,  $\mathcal{TD}(P)$ , de um subconjunto,  $P$ , de  $n$  pontos de  $\mathbb{E}^2$  requer  $\Omega(n \lg n)$  operações primitivas. Felizmente, existem algoritmos que utilizam  $\mathcal{O}(n \lg n)$  operações primitivas para gerar  $\mathcal{TD}(P)$  e, portanto, são ótimos para o problema da triangulação. Um deles é o algoritmo baseado no paradigma de divisão-e-conquista [105, 58]. Observe que o Algoritmo 4.2 *não é ótimo com relação ao tempo de execução de pior caso*, mas podemos afirmar que ele é ótimo com relação ao tempo de execução *esperado*





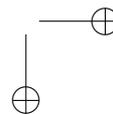
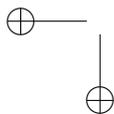
(ver Teorema 4.27). Além disso, o Algoritmo 4.2 é a base para o desenvolvimento do algoritmo de geração de malhas que estudaremos no Capítulo 5. Esta foi a razão pela qual optamos por apresentá-lo neste capítulo.

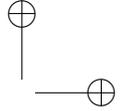
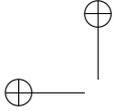
## 4.8 Aspectos de implementação

A nossa descrição do algoritmo incremental (aleatorizado) para gerar a triangulação de Delaunay de um subconjunto finito de pontos de  $\mathbb{E}^2$  (Algoritmo 4.2) priorizou a corretude e a complexidade do algoritmo. Por mais que tenhamos feito um esforço para descrever, com detalhes, o pseudocódigo de cada procedimento utilizado pelo algoritmo, alguns aspectos importantes e relacionados à implementação desses procedimentos não foram abordados. Em se tratando de um algoritmo para construir triangulações de Delaunay, existe um longo caminho a ser percorrido entre um pseudocódigo — por mais detalhado que ele seja — e uma implementação eficiente e robusta do algoritmo.

Antes de mais nada, faz-se necessária uma estrutura de dados para representar triangulações em  $\mathbb{E}^2$  e que permita uma implementação eficiente do algoritmo. Uma boa opção é a estrutura de dados *quedge*, que está descrita com riqueza de detalhes em [58]. A descrição de uma implementação do Algoritmo 4.2 baseada na estrutura de dados *quedge* e na linguagem C foi fornecida por Lischinski em [76]. Esta implementação, no entanto, faz uso de aritmética de ponto flutuante de precisão dupla e, portanto, não oferece nenhuma garantia de robustez no cálculo dos predicados geométricos utilizados pelo algoritmo.

Quem se interessar por implementar o Algoritmo 4.2 para uso em uma aplicação real deve adotar alguma biblioteca de aritmética de precisão múltipla ou que implemente predicados geométricos, tais como o *incircle*, de forma robusta [65, 117, 13, 74]. Isto porque a implementação de algoritmos geométricos, com aritmética de ponto flutuante (de precisão simples ou dupla), é bastante susceptível a erros de arredondamento [49, 116, 68]. A principal razão para definirmos todos os predicados geométricos que utilizamos por meio do cálculo de um determinante é o fato de existirem algoritmos que nos permitem



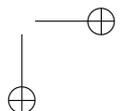
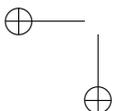


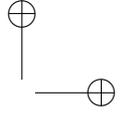
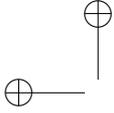
obter o *sinal* correto de um determinante usando aritmética de ponto flutuante [117, 99]. Pois, o que de fato importa para obtermos o valor lógico correto do predicado é o sinal do determinante e não seu valor absoluto.

Um outro aspecto prático diz respeito à inclusão de pontos especiais no conjunto  $P$  de entrada do algoritmo. Esta inclusão se justifica pelo uso da estrutura de dados hierárquica,  $\mathcal{D}$ , que permite que pontos sejam localizados eficientemente na triangulação atual (ver Subseção 4.6.4). No entanto, esta estrutura de dados, por mais simples que seja, requer uma quantidade extra de memória para armazenar os apontadores e os triângulos intermediários (uma quantidade em  $\mathcal{O}(n)$ ), além de tornar o algoritmo um pouco mais complicado de se implementar. Como alternativa, alguns pesquisadores propuseram boas heurísticas que, embora sejam menos eficientes do que o uso de  $\mathcal{D}$ , dispensam o uso de qualquer estrutura de dados auxiliar e de pontos especiais [57, 58, 91, 35, 137, 127, 31]. Essas heurísticas foram testadas e comparadas e, como mostram as observações experimentais, o desempenho dos algoritmos baseados em algumas delas não é muito diferente daquele obtido com o uso da estrutura hierárquica,  $\mathcal{D}$ .

## 4.9 Exercícios

- 4.1 Mostre que o fecho convexo de qualquer subconjunto (próprio) não vazio do conjunto de vértices de um  $k$ -simplexo  $\sigma$  em  $\mathbb{E}^d$ , com  $k \in \{0, \dots, d\}$ , gera um simplexo em  $\mathbb{E}^d$  (chamado de *face* de  $\sigma$ ).
- 4.2 Mostre que a condição (3) do Teorema 4.6 não é uma condição necessária, independentemente da condição (1) ser ou não válida.
- 4.3 Dê um algoritmo para subdividir um polígono convexo em triângulos, de tal forma que o conjunto de triângulos (e seus vértices e arestas) definam uma triangulação dos vértices do polígono. A entrada do algoritmo é uma lista com as coordenadas dos vértices do polígono na ordem em que eles ocorrem em um percurso anti-horário pela fronteira do polígono. A saída é o





conjunto de triângulos da triangulação representada na forma de uma lista de triplas de vértices, em que cada tripla contém as coordenadas dos vértices de um triângulo distinto da triangulação.

- 4.4 Seja  $P_n$  um polígono *convexo* em  $\mathbb{E}^2$  com exatamente  $n$  vértices, em que  $n \in \mathbb{N}$  e  $n \geq 3$ . Mostre que o número,  $t_n$ , de triangulações (cheias) de  $P_n$  satisfaz a seguinte relação de recorrência

$$\begin{cases} t_n &= t_2 \cdot t_{n-1} + t_3 \cdot t_{n-2} + \cdots + t_{n-1} \cdot t_2, & \text{para } n \geq 3 \\ t_2 &= 1, \end{cases}$$

em que  $t_2$  pode ser visto como o número de triangulações de um segmento de reta (ou seja, uma triangulação degenerada), isto é, 1.

- 4.5 Mostre que o número  $t_n$  do Problema 4.4, com  $n \in \mathbb{N}$  e  $n \geq 3$ , é

$$\frac{1}{n-1} \binom{2n-4}{n-2}.$$

- 4.6 O  $n$ -ésimo número Catalão, em que  $n \in \{0\} \cup \mathbb{N}$ , é o número  $C_n$  definido pela seguinte fórmula recursiva:  $C_0 = 1$  e, para todo  $n > 0$ ,

$$C_n = \sum_{k=0}^{n-1} C_k C_{n-k-1}.$$

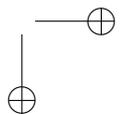
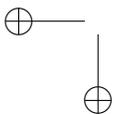
Mostre que  $C_n = t_{n+2}$ , em que  $t_{n+2}$  é o número de triangulações de um polígono convexo com exatamente  $(n+2)$  vértices, com  $t_2 = 1$ .

- 4.7 Mostre que o  $n$ -ésimo número Catalão,  $C_n$ , é tal que  $C_0 = 1$  e

$$C_n = \frac{1}{n+1} \binom{2n}{n}, \quad \forall n \in \mathbb{N}.$$

- 4.8 Encontre uma fórmula fechada para o  $n$ -ésimo número Catalão,  $C_n$ , definido recursivamente no Problema 4.6. Para tal, denote por  $F : \mathbb{R} \rightarrow \mathbb{R}$  a *função geradora* da sequência  $(C_i)_{i=1}^{\infty}$ , em que

$$F(x) = \sum_{i=0}^{\infty} C_i x^i.$$



Então,

(a) Prove que

$$F^2(x) = (F(x))^2 = \sum_{i=0}^{\infty} C_{i+1}x^i = \frac{F(x) - 1}{x}.$$

(b) Deduza que

$$F(x) = \frac{1 + \sqrt{1 - 4x}}{2x}.$$

Para tal, resolva a equação  $xF^2(x) = F(x) - 1$  assumindo que a função  $F(x)$  é a variável. Em seguida, ignore a solução que diverge quando  $x \mapsto 0^+$ , pois  $F$  deve ser tal que

$$\lim_{x \rightarrow 0^+} F(x) = 1.$$

(c) O teorema binomial de Newton nos diz que

$$(1 + z)^{\frac{1}{2}} = \sum_{k=0}^{\infty} \binom{\frac{1}{2}}{k} z^k,$$

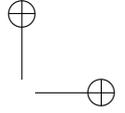
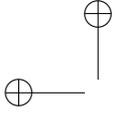
em que o binômio envolvendo a fração  $\frac{1}{2}$  é definido como

$$\binom{\frac{1}{2}}{k} = \frac{\frac{1}{2} \cdot (\frac{1}{2} - 1) \cdot (\frac{1}{2} - 2) \cdots (\frac{1}{2} - k + 1)}{k!}.$$

Faça  $z = -4x$  e expresse o binômio fracionário em termos de  $\binom{2k}{k}$ .

4.9 Mostre que o número  $t_{n+2}$  de triangulações de um polígono convexo com exatamente  $(n+2)$  vértices, em que  $n \in \mathbb{N}$  e  $t_2 = 1$  (por convenção), é igual ao número de seqüências,  $(s_i)_{i=1}^{2n}$ , de comprimento  $2n$  tais que (1)  $s_i = 1$  ou  $s_i = 0$  e (2)  $s_0, s_1, \dots, s_j$  contém um número de elementos iguais a 0 igual ou maior do que o número de elementos iguais a 1, para todo  $j \in \{1, \dots, 2n\}$ .

4.10 Seja  $P$  um subconjunto de  $\mathbb{E}^2$  com exatamente quatro pontos. Mostre que se  $P$  possui um subconjunto afimemente independente com três pontos em  $\mathbb{E}^2$ , então há exatamente duas triangulações de  $P$  em  $\mathbb{E}^2$ .



4.11 Um *complexo politopal*  $\mathcal{K}$  é a coleção finita de politopos em  $\mathbb{E}^d$  tal que (1) se  $\mathcal{P} \in \mathcal{K}$  então todas as faces de  $\mathcal{P}$  estão em  $\mathcal{K}$  também e (2) a interseção  $\mathcal{P} \cap \mathcal{Q}$  de dois politopos,  $\mathcal{P}$  e  $\mathcal{Q}$ , de  $\mathcal{K}$  é uma face de  $\mathcal{P}$  e de  $\mathcal{Q}$ . O *espaço subjacente*,  $|\mathcal{K}|$ , de  $\mathcal{K}$  é o conjunto (de pontos)  $|\mathcal{K}| = \bigcup_{\mathcal{P} \in \mathcal{K}} \mathcal{P} \subset \mathbb{E}^d$ . Mostre que a subdivisão de Delaunay de um conjunto de pontos  $P$ , que satisfaz as condições (1) e (3) da Definição 4.3, é um complexo politopal cujo conjunto de vértices é  $P$  e tal que  $|\mathcal{K}| = FC(P)$ .

4.12 Seja  $P$  um subconjunto de  $\mathbb{E}^2$  com exatamente  $n$  pontos, para algum  $n \in \mathbb{N}$ . Então, mostre que não podem existir mais do que

$$2^{\binom{n}{2}}$$

triangulações de  $P$ . Mostre que existe  $P$  tal que há pelo menos

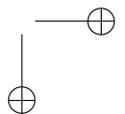
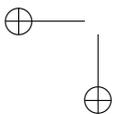
$$2^{n-2\sqrt{n}}$$

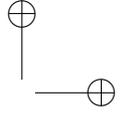
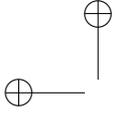
triangulações de  $P$ .

4.13 O *grau* ou *valência* de um vértice de uma triangulação de um subconjunto,  $P$ , de  $n$  pontos de  $\mathbb{E}^2$  é o número de arestas da triangulação que incidem sobre o vértice, isto é, que contém o vértice. Forneça um exemplo de um tal conjunto  $P$  em que sempre haverá um vértice de grau  $n - 1$  em *toda* triangulação de  $P$ .

4.14 Seja  $\mathcal{T}(P)$  uma triangulação de um subconjunto finito,  $P$ , de pontos de  $\mathbb{E}^2$ . Por definição, sabemos que  $|\mathcal{T}(P)| = FC(P)$ . É verdade que todo vértice de  $\mathcal{T}(P)$  pertencente à fronteira de  $|\mathcal{T}(P)|$  é um ponto extremo de  $FC(P)$ ? Justifique a sua resposta.

4.15 Seja  $\mathcal{T}(P)$  uma triangulação de um subconjunto,  $P$ , de pontos de  $\mathbb{E}^2$ . Sejam  $n$  o número de pontos em  $P$  e  $n_b$  o número de pontos de  $P$  que pertencem à fronteira de  $|\mathcal{T}(P)|$ , ou seja, à fronteira de  $FC(P)$  (e não são necessariamente pontos extremos). Então, mostre que os números de arestas e de triângulos de  $\mathcal{T}(P)$  são iguais a  $3n - n_b - 3$  e  $2n - n_b - 2$ , respectivamente.





4.16 Sejam  $\mathcal{T}_1$  e  $\mathcal{T}_2$  duas triangulações de um mesmo subconjunto finito,  $P$ , de pontos de  $\mathbb{E}^2$ . Sejam  $A(\mathcal{T}_1)$  e  $A(\mathcal{T}_2)$  os conjuntos de arestas de  $\mathcal{T}_1$  e  $\mathcal{T}_2$ , respectivamente. Mostre que existe uma bijeção,  $\Phi : A(\mathcal{T}_1) \rightarrow A(\mathcal{T}_2)$ , tal que toda aresta  $\sigma$  em  $A(\mathcal{T}_1)$  coincide ou cruza a aresta  $\Phi(\sigma)$  em  $A(\mathcal{T}_2)$ . A bijeção  $\Phi$  é denominada de *emparelhamento perfeito*, pois o problema proposto pode ser resolvido pela aplicação de um teorema clássico da Teoria dos Grafos sobre emparelhamentos perfeitos em grafos bipartidos.

4.17 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ . Então, mostre que o número de triangulações  $P$  é maximizado quando  $P$  goza da seguinte propriedade: *não há três pontos colineares em  $P$* . Em outras palavras, mostre que se houver três pontos colineares em qualquer  $P$  dado, então existe um subconjunto  $Q$  de  $\mathbb{E}^2$ , com o mesmo número  $n$  de pontos de  $P$ , sendo que quaisquer três deles não são colineares, tal que o número de triangulações de  $Q$  é maior ou igual ao número de triangulações de  $P$ .

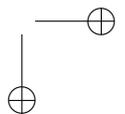
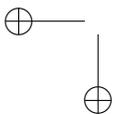
4.18 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ , tal que não há três pontos colineares em  $P$ . Seja  $\mathcal{T}(P)$  qualquer triangulação de  $P$ . Para cada natural  $i$ , com  $i \geq 3$ , denote por  $v_i$  o número de vértices interiores de  $\mathcal{T}(P)$  com valência igual a  $i$ . Para cada natural  $j \geq 2$ , denote por  $b_j$  o número de vértices de fronteira de  $\mathcal{T}(P)$  com valência igual a  $j$ . Então, mostre que

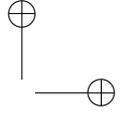
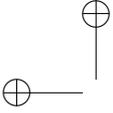
$$\sum_{i=3}^{n-1} (6-i) \cdot v_i + \sum_{j=2}^{n-1} (4-j) \cdot b_j = 6$$

e

$$3v_3 + 2v_4 + v_5 + 2b_2 + b_3 \geq 6 + \sum_{i=7}^{n-1} v_i + \sum_{j=5}^{n-1} b_j.$$

4.19 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ , tal que não há três pontos colineares em  $P$ . Seja  $\mathcal{T}(P)$  uma triangulação qualquer de  $P$  e seja  $x$  um ponto de  $\mathbb{E}^2$  que não está em  $P$ . Dizemos que uma triangulação,  $\mathcal{T}(P \cup \{x\})$ , é *obtida pela*





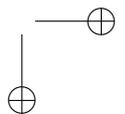
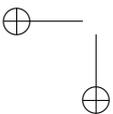
*inserção de  $x$  em  $\mathcal{T}(P)$*  se todos os triângulos de  $\mathcal{T}(P \cup \{x\})$  que não incidem sobre  $x$  (ou seja, dos quais  $x$  não é um vértice) são triângulos de  $\mathcal{T}(P)$  também. Para cada  $i \in \mathbb{N}$ , denote por  $h_i$  o número de triangulações,  $\mathcal{T}(P \cup \{x\})$ , de  $P \cup \{x\}$ , em que  $x \notin P$ , tais que  $x$  possui valência igual a  $i$  e que podem ser obtidas pela inserção de  $x$  em  $\mathcal{T}(P)$ . Assuma que  $P \cup \{x\}$  não contém três pontos colineares e mostre que  $h_3 = 1$ ,  $h_4 \leq 4$ ,  $h_5 \leq 9$  e  $h_6 \leq 28$  se  $x$  é um ponto interior de  $FC(P \cup \{x\})$ . Caso contrário, mostre que  $h_2 \leq 1$ ,  $h_3 \leq 1$  e  $h_4 \leq 2$ .

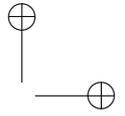
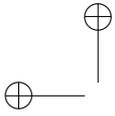
- 4.20 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ . Então, mostre que o número de triangulações de  $P$  é limitado acima por

$$\frac{59^v \cdot 7^b}{\binom{v+b+6}{6}},$$

em que  $v$  e  $b$  denotam os números de pontos de  $P$  no interior e na fronteira de  $FC(P)$ , respectivamente. *Faça uso dos resultados do enunciado do Problema 4.18 e do enunciado do Problema 4.19.*

- 4.21 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ , tal que não há três pontos colineares em  $P$ . Então, mostre que  $P$  admite pelo menos  $C_{b-2} \cdot 2^{v-b+2} \in \Omega(2^n \cdot n^{-\frac{3}{2}})$  triangulações, em que  $b$  e  $v$  são os números de pontos de  $P$  interiores e na fronteira de  $FC(P)$ , respectivamente,  $n = v + b$  e  $C_k$  é o  $k$ -ésimo número Catalão.
- 4.22 Descreva (construa) um subconjunto  $P$  de  $n + 2$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N} \cup \{0\}$ , tal que o número de triangulações de  $P$  seja maior do que  $C_n$ , em que  $C_n$  denota o  $n$ -ésimo número Catalão.
- 4.23 Considere o conjunto de pontos,  $P$ , ilustrado na Figura 4.43. Este conjunto é formado por duas “cadeias” de pontos (exibidas por arestas tracejadas) tais que todo par de pontos com um ponto de cada cadeia é visível pelo outro; ou seja, o segmento de reta que os une não intersecta nenhuma aresta (tracejada) das cadeias. Mostre que as arestas (tracejadas) das cadeias devem





pertencer a *toda* triangulação de  $P$ . Além disso, se há  $n$  pontos em cada cadeia, determine o número exato de triangulações de  $P$ .

- 4.24 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ , tal que não há três pontos colineares em  $P$ . Então, mostre que todas as arestas de  $FC(P)$  são arestas de qualquer triangulação,  $\mathcal{T}(P)$ , de  $P$ .
- 4.25 Forneça um algoritmo para *contar* o número de triangulações de um subconjunto  $P$  qualquer de  $n$  pontos de  $\mathbb{E}^2$ . Qual é a complexidade do seu algoritmo? Se exigirmos que a complexidade do algoritmo seja polinomial, isto é,  $\mathcal{O}(n^k)$ , para alguma constante  $k \in \mathbb{R}$ , com  $k \geq 1$ , o problema ainda está *em aberto*. O que podemos afirmar com segurança é que tal algoritmo não pode fazer a contagem do número de triangulações de  $P$  por “enumeração”, já que há, em geral, um número exponencial de triangulações de  $P$ . Este fato pode ser deduzido do Problema 4.12.

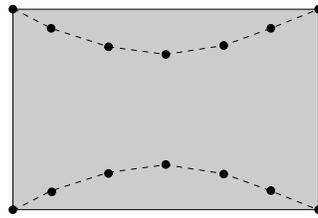
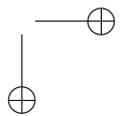
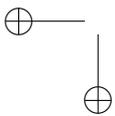
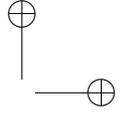
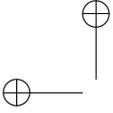


Figura 4.43: Definição do conjunto  $P$  do Problema 4.23.

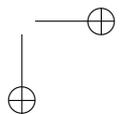
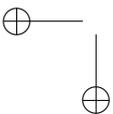
- 4.26 A grande maioria dos livros que tratam de Geometria Computacional definem triangulação de pontos de  $\mathbb{E}^2$  por meio da noção de *subdivisão*. Dado um subconjunto  $P$  de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ , uma subdivisão pode ser definida por meio de um mergulho (desenho) de um grafo,  $G = (V, E)$ , em  $\mathbb{E}^2$  tal que os vértices de  $G$  são os pontos de  $P$  e as arestas de  $G$  são segmentos de reta que não se cruzam e incidem sobre os pontos de  $P$ . Em outras palavras, o grafo  $G$  é *planar*. Este mergulho

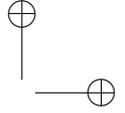
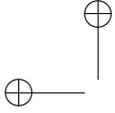




particiona o plano em “regiões” poligonais limitadas e ilimitadas de cujos vértices e arestas são aqueles do mergulho de  $G$  em  $\mathbb{E}^2$ . Uma *subdivisão* de  $P$  é então definida como o conjunto de vértices, arestas e regiões *limitadas* obtidas de um mergulho de um grafo  $G = (V, E)$  em  $\mathbb{E}^2$  tal que os vértices de  $G$  são os pontos de  $P$  e as arestas de  $G$  são segmentos de reta que não se cruzam e incidem sobre os pontos de  $P$ . Dizemos que a subdivisão é *maximal* quando não podemos adicionar uma aresta  $e$  à subdivisão, conectando dois pontos de  $P$ , sem cruzar com alguma aresta existente. Uma triangulação de  $P$  é então definida como uma subdivisão maximal de  $P$ . Prove que esta definição é equivalente à definição de triangulação (Definição 4.2) da Seção 4.2.

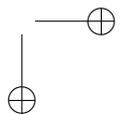
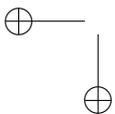
- 4.27 Seja  $P$  qualquer subconjunto de  $n$  pontos de  $\mathbb{E}^2$ , com  $n \in \mathbb{N}$ . Mostre que toda triangulação,  $\mathcal{T}(P)$ , possui um vértice de grau  $\leq 5$ .
- 4.28 Sejam  $P$  e  $Q$  dois subconjuntos de pontos de  $\mathbb{E}^2$  com exatamente  $n$  pontos cada, em que  $n \in \mathbb{N}$ . Sejam  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$  duas triangulações quaisquer de  $P$  e de  $Q$ , respectivamente. Dizemos que  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$  são *compatíveis* se existir uma bijeção,  $\Phi : P \rightarrow Q$ , entre os conjuntos  $P$  e  $Q$  tal que existe um triângulo em  $\mathcal{T}(P)$  com vértices nos pontos  $x, y$  e  $z$  de  $P$  se, e somente se, existe um triângulo em  $\mathcal{T}(Q)$  com vértices nos pontos  $\Phi(x), \Phi(y)$  e  $\Phi(z)$  de  $Q$ . Dados dois subconjuntos,  $P$  e  $Q$ , com  $n$  pontos de  $\mathbb{E}^2$  cada e tais que  $FC(P)$  e  $FC(Q)$  possuem o mesmo número de vértices, é verdade que sempre existem triangulações,  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$ , de  $P$  e  $Q$ , respectivamente, que são compatíveis? Esta pergunta foi respondida apenas para o caso em que há no máximo três pontos de  $P$  (resp.  $Q$ ) no interior de  $FC(P)$  (resp.  $FC(Q)$ ), e o caso geral permanece em aberto (refira-se a [1], por exemplo).
- 4.29 No Problema 4.28, devemos mostrar que existe uma bijeção  $\Phi$  entre  $P$  e  $Q$  tal que duas triangulações,  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$ , de  $P$  e  $Q$ , respectivamente, são compatíveis. No entanto, se  $\Phi$  for fornecida e fixada *a priori*, então o problema passa a ser o de determinar se existem duas triangulações,  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$ , de  $P$  e  $Q$ , respectivamente, que são compatíveis *com relação à bijeção*

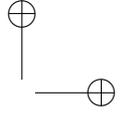
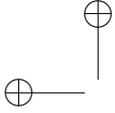




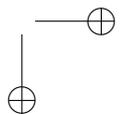
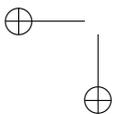
$\Phi$ . Neste caso, nem sempre há triangulações compatíveis [114]. Construa dois subconjuntos,  $P$  e  $Q$ , de  $n$  pontos de  $\mathbb{E}^2$ , com o mesmo número  $h$  de pontos na fronteira de  $FC(P)$  e  $FC(Q)$ , e especifique uma bijeção,  $\Phi : P \rightarrow Q$ , para a qual não existem triangulações compatíveis,  $\mathcal{T}(P)$  e  $\mathcal{T}(Q)$ , de  $P$  e  $Q$ , respectivamente.

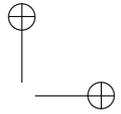
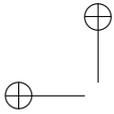
- 4.30 Sejam  $P$  e  $Q$  dois subconjuntos de pontos de  $\mathbb{E}^2$  com exatamente  $n$  pontos cada, em que  $n \in \mathbb{N}$ . Mostre que se adicionarmos dois pontos em cada um desses dois conjuntos, então sempre existirão triangulações compatíveis,  $\mathcal{T}(P')$  e  $\mathcal{T}(Q')$ , de  $P'$  e  $Q'$ , respectivamente, em que  $P'$  e  $Q'$  são os conjuntos resultantes da adição de dois pontos a  $P$  e  $Q$ . Observe que o número de pontos de  $P$  na fronteira de  $FC(P)$  não precisa ser igual ao número de pontos de  $Q$  na fronteira de  $FC(Q)$ . *Escolha dois pontos para inserir em  $P$  (resp.  $Q$ ) no exterior de  $FC(P)$  (resp.  $FC(Q)$ ).*
- 4.31 Modifique o algoritmo Graham *scan* (Algoritmo 3.1) para gerar uma triangulação,  $\mathcal{T}(P)$ , do conjunto,  $P$ , de pontos de  $\mathbb{E}^2$  dado como entrada. Assuma que  $P$  goza das propriedades (1)-(3) do Teorema 4.6, mas não descarte a hipótese de  $P$  possuir três pontos colineares.
- 4.32 Mostre que o grafo de troca de arestas de um subconjunto finito e não vazio de pontos de  $\mathbb{E}^2$  é conexo; isto é, mostre que todo par de nós do grafo está conectado por um caminho no grafo. Lembre-se de que um caminho em um grafo é uma sequência de nós do grafo tal que se  $v$  e  $w$  são nós consecutivos da sequência, então tem de existir uma aresta do grafo conectando os nós  $v$  e  $w$ .
- 4.33 Mostre que o grafo de troca de arestas de um subconjunto finito e não vazio de pontos de  $\mathbb{E}^2$  não pode conter um ciclo de comprimento 3. O comprimento de um caminho de  $n$  nós é igual a  $n - 1$  e que um ciclo é um caminho “fechado”, ou seja, o primeiro e o último elemento da sequência representando o caminho são iguais (o mesmo nó do grafo). Um ciclo de comprimento 3 é uma sequência composta por quatro nós do grafo, no qual o primeiro e o último nó são exatamente os mesmos nós.





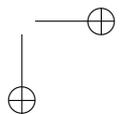
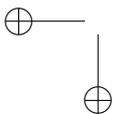
- 4.34 Mostre que o diâmetro do grafo de troca de arestas de um subconjunto de  $n$  pontos de  $\mathbb{E}^2$  não é maior do que  $(n-2)(n-3)$ . O *diâmetro* de um grafo é igual ao comprimento do caminho *simples* mais comprido entre dois nós quaisquer do grafo, em que um caminho simples é aquele que não possui nós repetidos. Para resolver este problema, você talvez queira consultar a referência [33] (páginas 178 a 180) para estudar a triangulação *incremental* (tradução livre do inglês *pushing triangulation*) de um subconjunto,  $P$ , finito e não vazio de pontos de  $\mathbb{E}^2$  e fazer uso do resultado que diz que qualquer triangulação de  $P$  pode ser transformada na triangulação incremental de  $P$  realizando, no máximo,  $\binom{n-2}{2}$  trocas de arestas. A partir deste resultado, você pode provar o que se pede aqui usando indução finita em  $n$ .
- 4.35 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6 e a condição de que não existem três pontos colineares em  $P$ . Sejam  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$  duas triangulações quaisquer de  $P$ . Denote por  $T_{12}$  o diagrama que se obtém pela sobreposição dos desenhos dos vértices e arestas de  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$ . Então, mostre que há um caminho no grafo de troca de arestas de  $P$ , conectando os nós correspondentes a  $\mathcal{T}_1(P)$  e  $\mathcal{T}_2(P)$ , cujo comprimento não excede o número de cruzamentos de arestas em  $T_{12}$ . Este resultado foi provado em [62].
- 4.36 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6 e a condição de que não existem três pontos colineares em  $P$ . Suponha que  $P$  possua  $n_b$  pontos no fecho de  $P$  e  $n_i$  pontos no interior de  $FC(P)$ . Então, prove que o número de cruzamento de arestas na sobreposição dos desenhos dos vértices e arestas de quaisquer duas triangulações de  $P$  é, no máximo, igual a  $(3n_i + n_b - 3)^2$  (consulte [62]).
- 4.37 Seja  $P$  um subconjunto de  $n$  pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6 e a condição de que não existem três pontos colineares em  $P$ . Forneça um algoritmo que, dados um natural  $k$  e quaisquer duas triangulações,  $\mathcal{T}_1(P)$  e

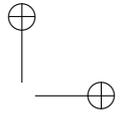
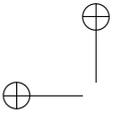




$\mathcal{T}_2(P)$ , de  $P$ , determina se existe uma sequência de troca de arestas com, no máximo,  $k$  trocas que converte  $\mathcal{T}_1(P)$  em  $\mathcal{T}_2(P)$ . Qual é a complexidade de seu algoritmo? O problema de encontrar um algoritmo cuja complexidade é polinomial em  $n$  (isto é,  $\mathcal{O}(n^c)$ , para alguma constante natural  $c$ ) *está em aberto há mais de 25 anos*. Ver [79, 2] para resultados recentes sobre o problema.

- 4.38 Seja  $P$  um subconjunto de pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6. Seja  $\mathcal{T}(P)$  qualquer triangulação de  $P$ . Dizemos que uma aresta  $e$  de  $\mathcal{T}(P)$  é *Delaunay* se, e somente se, a aresta  $e$  também pertence à triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ . Mostre que se  $e$  é uma aresta de  $\mathcal{TD}(P)$ , então a aresta  $e$  é uma corda de um círculo,  $C$ , tal que  $P \cap \text{int}(C) = \emptyset$ .
- 4.39 Seja  $P$  um subconjunto de pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6. Seja  $\mathcal{T}(P)$  qualquer triangulação de  $P$ . Então, prove a equivalência das três seguintes afirmações:
- $\mathcal{T}(P)$  é a triangulação de Delaunay de  $P$ .
  - Toda aresta de  $\mathcal{T}(P)$  é Delaunay.
  - Toda aresta de  $\mathcal{T}(P)$  é localmente Delaunay.
- 4.40 Forneça um conjunto  $P$  de entrada para o Algoritmo 4.1 que seja representante das entradas de pior caso do algoritmo e que não seja igual ao da Figura 4.21. Dê uma descrição em termos do número  $n$  de elementos de  $P$ , ou seja, não fixe um valor para  $n$ .
- 4.41 Usando a definição de aresta localmente Delaunay dada na Definição 4.5, mostre que o Algoritmo 4.1 sempre termina e que produz uma triangulação do conjunto,  $P$ , de pontos de entrada na qual toda aresta é localmente Delaunay (de acordo com a Definição 4.5). Assuma que o conjunto  $P$  satisfaz as condições (1) e (3) do Teorema 4.6, mas não necessariamente a condição (2).
- 4.42 Usando a definição de aresta localmente Delaunay dada na Definição 4.5, mostre que o Algoritmo 4.4 produz uma triangulação



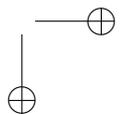
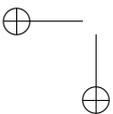


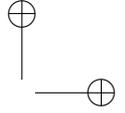
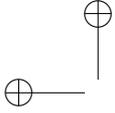
do conjunto,  $P$ , de pontos de entrada na qual toda aresta é localmente Delaunay (de acordo com a Definição 4.5) e todo triângulo é de Delaunay. Assuma que  $P$  satisfaz as condições (1) e (3) do Teorema 4.6, mas não necessariamente a condição (2).

- 4.43 Seja  $P$  o subconjunto de pontos de  $\mathbb{E}^2$  dado como entrada para o algoritmo Algoritmo 4.2. Então, descreva um procedimento para calcular as coordenadas dos pontos especiais,  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ , de tal forma que (a) nenhum círculo cuja circunferência seja definida por três pontos de  $P$  contém  $p_0$ , (b) nenhum círculo cuja circunferência seja definida por três pontos de  $P \cup \{p_0\}$  contém  $p_{-1}$  e (c) nenhum círculo cuja circunferência seja definida por três pontos de  $P \cup \{p_0, p_{-1}\}$  contém  $p_{-2}$ . Qual é a complexidade do seu procedimento? O seu procedimento é ótimo? É verdade que *qualquer* procedimento para calcular  $p_{-2}$ ,  $p_{-1}$  e  $p_0$  está fadado ao mesmo esforço computacional de seu procedimento em alguma entrada  $P$  representativa de pior caso?
- 4.44 Seja  $\mathcal{TD}_j$  a triangulação de Delaunay de  $S_j = S_{j-1} \cup \{p_j\}$  obtida a partir da inserção do ponto  $p_j$  na triangulação  $\mathcal{TD}_{j-1}$  de  $S_{j-1}$  durante a execução do segundo passo do Algoritmo 4.2, em que

$$S_{j-1} = \{p_{-2}, p_{-1}, p_0\} \cup P_{j-1} \quad \text{e} \quad P_{j-1} = \{p_1, \dots, p_{j-1}\}.$$

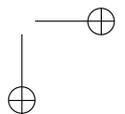
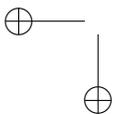
Mostre os triângulos que estão em  $\mathcal{TD}_{j-1}$ , mas não fazem parte da triangulação  $\mathcal{TD}_j$ , são exatamente aqueles triângulos de  $\mathcal{TD}_{j-1}$  cujos círculos circunscritos contém o ponto  $p_j$  no interior. Em seguida, mostre também que a união desses triângulos resulta em uma região poligonal,  $\mathcal{R}$ , que contém  $p_j$  em seu interior e cujos vértices de sua fronteira são todos “visíveis” a partir de  $p_j$ ; isto é, todo segmento que conecta  $p_j$  a um vértice qualquer de  $\mathcal{R}$  está inteiramente contido em  $\mathcal{R}$ . Finalmente, mostre que a triangulação de  $\mathcal{R}$  obtida através da adição de arestas conectando  $p_j$  aos vértices de  $\mathcal{R}$  consiste exatamente dos triângulos que estão em  $\mathcal{TD}_j$ , mas que não estão em  $\mathcal{TD}_{j-1}$ . Usando os resultados obtidos aqui, descreva um procedimento

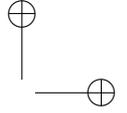
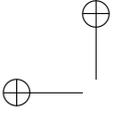




para substituir `TROCAARESTAS()` no Algoritmo 4.2 que encontra todos os triângulos de  $\mathcal{TD}_{j-1}$  que não estão em  $\mathcal{TD}_j$ , remove esses triângulos de  $\mathcal{TD}_{j-1}$  e, por último, acrescenta os triângulos que estão em  $\mathcal{TD}_j$ , mas não estão em  $\mathcal{TD}_{j-1}$ . Qual é a complexidade de tempo de *pior caso* do seu procedimento?

- 4.45 Complete a prova do Corolário 4.24 desenvolvendo a indução.
- 4.46 A prova do Lema 4.25 assume que todas as permutações de  $S = \{p_{-2}, p_{-1}, p_0\} \cup P$  são possíveis e igualmente prováveis. No entanto, sabemos que o Algoritmo 4.2 sempre insere os pontos especiais,  $p_{-2}$ ,  $p_{-1}$  e  $p_0$ , primeiro, criando um triângulo inicial,  $\tau_0$ , cujos vértices estão localizados nos pontos especiais. Logo, nem toda permutação considerada pela prova do Lema 4.25 é possível de ocorrer. Mostre, no entanto, que as cotas superiores obtidas nas provas do Teorema 4.26 e Teorema 4.27 continuam válidas quando consideramos apenas as permutações possíveis.
- 4.47 O grafo,  $G = (V, E)$ , de Gabriel (GG) de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  é definido como segue: (a) o conjunto,  $V$ , de vértices de  $G$  é o próprio  $P$  e (b)  $\mu = \{r, s\}$  é uma aresta em  $E$  se, e somente se, o círculo de diâmetro  $\overline{rs}$  não contém nenhum outro ponto de  $P$  além de  $r$  e  $s$ . Se  $P$  satisfaz as condições (1)-(3) do Teorema 4.6, então mostre que  $E$  está contido no conjunto de arestas da triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ .
- 4.48 A árvore geradora mínima euclidiana (AGME) de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  é uma árvore que conecta todos os pontos de  $P$  e cuja soma dos comprimentos das arestas é mínimo com respeito a todas as árvores geradoras definidas em  $P$ . Suponha que  $P$  satisfaz as condições (1)-(3) do Teorema 4.6 e mostre que todas as arestas de uma AGME de  $P$  pertencem ao conjunto de arestas da triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ .
- 4.49 O grafo,  $G = (V, E)$ , de vizinhança relativa (GVR) de um subconjunto,  $P$ , não vazio e finito de pontos de  $\mathbb{E}^2$  é definido como segue: (a) o conjunto,  $V$ , de vértices de  $G$  é o próprio  $P$  e (b)





$\mu = \{r, s\}$  é uma aresta em  $E$  se, e somente se,  $r$  e  $s$  são tais que

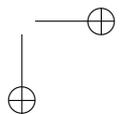
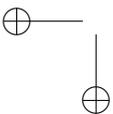
$$d(r, s) \leq \min_{t \in P, t \neq r, s} \max\{d(r, t), d(s, t)\},$$

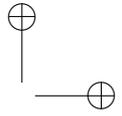
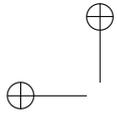
em que  $d(\cdot, \cdot)$  denota a distância euclidiana entre dois pontos de  $\mathbb{E}^2$ . Dados dois pontos  $r$  e  $s$  em  $P$ , denotamos por  $lua(r, s)$  a região em forma de lua definida como a interseção dos círculos centrados em  $r$  e  $s$  e cujos raios são iguais a  $d(r, s)$ . Prove que  $r$  e  $s$  são incidentes em uma aresta de  $G$  se, e somente se, o interior de  $lua(r, s)$  não contém nenhum ponto de  $P$ . Suponha que  $P$  satisfaz as condições (1)-(3) do Teorema 4.6 e mostre que todas as arestas do grafo de vizinhança relativa de  $P$  pertencem ao conjunto de arestas da triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ .

4.50 Seja  $P$  um subconjunto não vazio e finito de pontos de  $\mathbb{E}^2$  satisfazendo as condições (1)-(3) do Teorema 4.6. Para cada ponto  $r \in P$ , definimos a *região de Voronoi*,  $V_r$ , de  $r$  como sendo o conjunto

$$V_r = \{x \in \mathbb{E}^2 \mid d(x, r) \leq d(x, s), \forall s \in P, s \neq r\},$$

isto é, o conjunto  $V_r$  consiste exatamente dos pontos de  $\mathbb{E}^2$  que estão mais próximos ou tão próximos de  $r$  quanto de qualquer outro ponto de  $P$ . Para cada par,  $(r, s)$ , de pontos de  $P$  tal que  $V_r \cap V_s \neq \emptyset$ , definimos a *aresta de Voronoi*,  $V_{rs}$ , de  $r$  e  $s$  como sendo o conjunto  $V_{rs} = V_r \cap V_s$ . Para cada tripla  $(r, s, t)$ , de pontos de  $P$  tal que  $V_r \cap V_s \cap V_t \neq \emptyset$ , definimos o *vértice de Voronoi*,  $V_{rst}$ , de  $r$ ,  $s$  e  $t$  como sendo o conjunto  $V_{rst} = V_r \cap V_s \cap V_t$ . O conjunto,  $\mathcal{V}(P)$ , de todas as regiões, arestas e vértices de Voronoi de  $P$  definem uma *subdivisão* de  $\mathbb{E}^2$  denominada *diagrama de Voronoi* (ver Figura 4.44). Mais especificamente, a união de todas as regiões, arestas e vértices de Voronoi de  $P$  é igual a  $\mathbb{E}^2$  e a interseção do *interior* de quaisquer dois conjuntos da subdivisão é vazia. Podemos mostrar que  $V_r$ ,  $V_{rs}$  e  $V_{rst}$  são conjuntos convexos (e conexos). Além disso, podemos mostrar que o fecho afim,  $FA(V_r)$ , de  $V_r$  possui dimensão 2, que o fecho afim,  $FA(V_{rs})$ , de  $V_{rs}$  possui dimensão 1 e que  $V_{rst}$  é um único ponto em  $\mathbb{E}^2$ . No entanto, aqui, pedimos para que o leitor





mostre que  $\mathcal{V}(P)$  é *dual* da triangulação de Delaunay,  $\mathcal{TD}(P)$ , de  $P$ . Isto é, mostre que a aresta  $[r, s]$  pertence a  $\mathcal{TD}(P)$  se, e somente se,  $V_r \cap V_s \neq \emptyset$  e, portanto,  $V_{rs}$  é uma aresta de  $\mathcal{V}(P)$ . Mostre também que o triângulo  $[r, s, t]$  pertence a  $\mathcal{TD}(P)$  se, e somente se,  $V_r \cap V_s \cap V_t \neq \emptyset$  e, portanto,  $V_{rst}$  é um vértice de  $\mathcal{V}(P)$ .

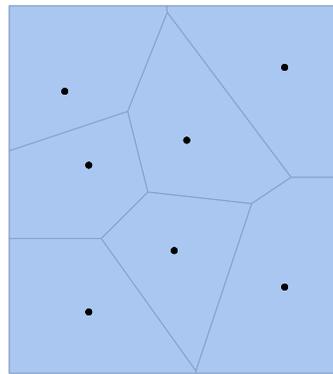
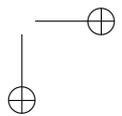
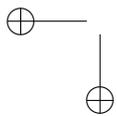
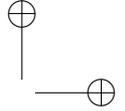
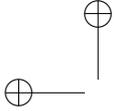


Figura 4.44: O diagrama de Voronoi do mesmo conjunto de pontos sobre o qual a triangulação de Delaunay (dual) da Figura 4.12 foi definida.

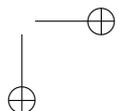
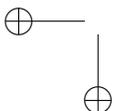
## 4.10 Notas bibliográficas

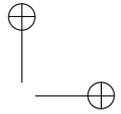
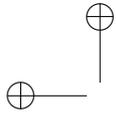
A Seção 4.2 e a Seção 4.5 foram inspiradas na recente obra sobre triangulações escrita por De Loera, Rambau e Santos [33] e no livro de Devadoss e O’Rourke [34]. Os fatos sobre politopos foram retirados do livro de Ziegler [138]. Em particular, recomendamos ao leitor que procure conhecer o livro [33], pois ele se constitui em uma fonte enciclopédica sobre os aspectos combinatórios das triangulações, é extremamente bem ilustrado e contém informações atuais sobre resultados recentes de pesquisa acerca de triangulações em  $\mathbb{E}^n$ . É importante enfatizar, também, que nossa apresentação de triangulação de Delaunay





se diferencia daquela dada pela maioria dos livros, pois não vinculamos a definição de triangulação de Delaunay ao conhecimento prévio da definição de Diagrama de Voronoi. Acreditamos que esta separação ajuda a estreitar o foco do nosso livro e a ressaltar aspectos da triangulação de Delaunay que não são apresentados em outros textos. A Seção 4.6 é bem próxima do Capítulo 9 do livro [30]. A análise do algoritmo incremental para construção da triangulação de Delaunay, na Subseção 4.6.4, foi baseada no artigo de Guibas, Knuth e Sharir [59]. Os problemas 4.4-4.9, 4.14-4.21 foram retirados ou elaborados a partir do vasto material do livro de De Loera, Rambau e Santos [33]. Um outro livro que serviu de fonte e inspiração para os problemas deste capítulo é o escrito por Devadoss e O’Rourke [34], do qual retiramos e/ou elaboramos o Problema 4.10 e os problemas 4.22 -4.30. Já os problemas 4.12-4.13 e 4.47-4.49 foram retirados de [30]. O procedimento `GERADELAUNAY()` é atribuído a Charles Lawson [73]. A observação que a triangulação de Delaunay maximiza o menor ângulo, entre todas as possíveis triangulações do mesmo conjunto de pontos, é creditada a Sibson [126]. Algoritmos para calcular a triangulação de Delaunay de um subconjunto de pontos de  $\mathbb{E}^2$  surgiram nos anos 70. A origem do algoritmo incremental é creditada a Lawson [73] e a Green e Sibson [57]. A primeira extensão do algoritmo incremental para construir triangulações de subconjuntos de pontos em  $\mathbb{E}^n$  é atribuída a Watson [135] e Bowyer [18]. Guibas e Stolfi escreveram um artigo, [58], que se tornou um “marco” na área de Geometria Computacional. Neste artigo, eles propõem uma estrutura de dados eficiente para manter, ao mesmo tempo, a triangulação de Delaunay e o seu dual, o diagrama de Voronoi (ver Problema 4.50), e descrevem pseudocódigos com todos os detalhes necessários para implementar o algoritmo de divisão-e-conquista [105, 40] e o algoritmo incremental para construir triangulações de subconjuntos de pontos em  $\mathbb{E}^2$ . O algoritmo incremental *aleatorizado*, como apresentado na Seção 4.6, foi dado por Guibas, Knuth e Sharir em [59]. Uma comparação experimental de vários algoritmos para calcular a triangulação de Delaunay em  $\mathbb{E}^2$  foi descrita por Su e Drysdale em [128].



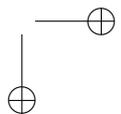
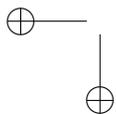


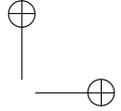
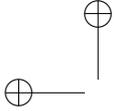
## Capítulo 5

# Geração de malhas

### 5.1 Considerações iniciais

Este capítulo é dedicado ao estudo de um algoritmo extremamente elegante, o *algoritmo de Ruppert*, para resolver um problema de extrema relevância prática: o *problema da geração de malhas*. Aqui, estamos interessados em uma das instâncias mais simples do problema da geração de malhas, que consiste na geração de uma malha de triângulos de um domínio planar. Uma malha de triângulos é uma triangulação. Então, em princípio, o problema poderia ser resolvido com o algoritmo que estudamos no Capítulo 4. Infelizmente, uma malha não é uma mera triangulação. Em geral, uma malha é uma triangulação que deve satisfazer certos *critérios de otimalidade* os quais são dependentes das aplicações em que as malhas são empregadas. Além disso, uma entrada para o problema de geração de malhas não composta apenas por um conjunto de pontos, mas sim por um conjunto de pontos e segmentos de reta. Estes segmentos devem ser representados pela malha, o que significa que cada segmento deve ser uma aresta ou uma união de arestas da malha. Este requisito é denominado *conformidade*. Finalmente, o espaço subjacente da malha nem sempre é convexo. A exigência de conformidade e os critérios de otimalidade tornam o problema de geração de malhas bem mais complexo do que o problema de geração de uma triangulação de um conjunto de pontos. A Seção 5.2 descreve os principais critérios de



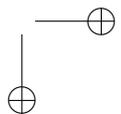
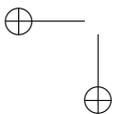


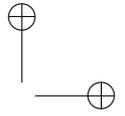
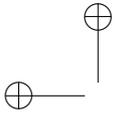
otimalidade. A Seção 5.3 apresenta a técnica de refinamento de Delaunay para geração de malhas e discute, em detalhes, um dos algoritmos baseados nesta técnica: o algoritmo de Ruppert. A Seção 5.4 prova que o algoritmo de Ruppert sempre termina, caso algumas restrições na entrada do algoritmo sejam satisfeitas. A Seção 5.5 mostra que o algoritmo de Ruppert satisfaz alguns dos principais critérios de otimalidade. A Seção 5.6 mostra que uma das restrições impostas sobre a entrada do algoritmo de Ruppert, a qual limita a aplicação prática do algoritmo, pode ser relaxada sem que as malhas geradas pelo algoritmo deixem de satisfazer os principais critérios de otimalidade.

## 5.2 Critérios de otimalidade

Técnicas de geração de malha são amplamente aplicadas em várias áreas da computação, matemática aplicada e engenharia, principalmente aquelas que envolvem simulações numéricas de modelos físicos descritos por uma equação diferencial parcial (EDP). Para essa finalidade, um dos métodos numéricos mais utilizados é o *método de elementos finitos* (MEF), introduzido no Capítulo 6. Quando aplicado a um problema cujo domínio é uma região do plano euclidiano — domínio onde um determinado fenômeno físico ocorre e que geralmente possui uma geometria complexa — uma das etapas do MEF consiste em decompor o domínio em uma malha de elementos, tais como quadriláteros e triângulos.

O tempo computacional para resolver o sistema de equações proveniente da solução aproximada de uma EDP usando o MEF é diretamente proporcional ao número de elementos da malha. Por outro lado, a qualidade da solução numérica via MEF (isto é, quão bem ela aproxima a solução exata) depende da resolução da malha: uma *malha fina* com vários triângulos pequenos geralmente fornece uma solução mais precisa que uma *malha grossa* com poucos triângulos. De acordo com essas observações temos um conflito de requisitos baseado na resolução da malha: mirando na eficiência computacional, a malha deve ter um número mínimo de elementos e, ao mesmo tempo, deve ser densa o suficiente para garantir a acurácia e precisão da solução numérica. Para que tenhamos o melhor dos dois mundos,





faz-se necessário o uso de uma malha adaptativa, ou seja, uma malha que tenha uma resolução que varie ao longo do domínio concentrando esforço computacional em regiões que afetam diretamente a acurácia da solução. Dessa forma, a malha deve ser fina em regiões onde a solução de um problema através do MEF varia rapidamente e grossa em regiões de baixa variação. Em adição, a transição espacial (do inglês *grading*) entre os triângulos pequenos e grandes deve ser bem suave.

Um outro aspecto a ser considerado é o formato dos triângulos da malha, pois ele afeta a estabilidade, convergência e acurácia da solução numérica fornecida pelo MEF. Triângulos *magros*, isto é, triângulos alongados com ângulos internos pequenos ou grandes podem acarretar cálculos espúrios e instáveis de derivadas. A Figura 5.1 mostra dois triângulos magros, um com ângulo próximo de  $180^\circ$  e outro com um ângulo pequeno e uma aresta muito mais curta do que as demais.

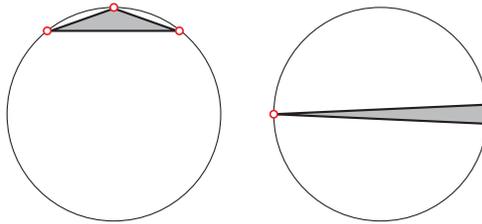
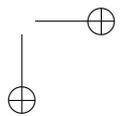
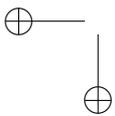


Figura 5.1: Triângulos magros possuem grandes circuncírculos.

Uma maneira de evitar triângulos magros é gerar triângulos que sejam os mais próximos possíveis de triângulos equiláteros. Na Seção 4.5, vimos que, dentre todas triangulações possíveis de um subconjunto de pontos de  $\mathbb{E}^2$ , a triangulação de Delaunay é ótima com relação à maximização do menor ângulo interno da triangulação. Entretanto, não há nenhum limite com relação a quão pequeno este menor ângulo possa ser. Os algoritmos de geração de malhas adotam métricas de qualidade de formato de triângulos para medir quão ruim (ou bom) o formato de um triângulo gerado por eles é. Uma delas é a *razão circunraio-aresta*. Seja  $\tau$  um triângulo, então a razão



circunraio-aresta,  $\rho(\tau)$ , de  $\tau$  é dada por

$$\rho(\tau) = \frac{R}{l}, \tag{5.1}$$

em que  $R$  é o circunraio de  $\tau$  e  $l$  é o comprimento da aresta mais curta de  $\tau$ . Se os lados de  $\tau$  medem  $a$ ,  $b$  e  $c$ , então o valor de  $R$  é dado por

$$R = \frac{a \cdot b \cdot c}{4 \cdot \text{área}(\tau)}.$$

A aresta mais curta é sempre oposta ao menor ângulo  $\theta_{\min}$  de  $\tau$  (Figura 5.2) e a relação entre a razão circunraio-aresta e  $\theta_{\min}$  é dada por

$$\rho(\tau) = \frac{1}{2 \text{sen}(\theta_{\min})}. \tag{5.2}$$

Isso significa que a razão circunraio-aresta é uma função decrescente que varia de  $\infty$  para um triângulo degenerado até  $1/[2 \text{sen}(60^\circ)] = 1/\sqrt{3} \approx 0.577$  para um triângulo com o “melhor” formato, isto é, um triângulo equilátero. Logo, maximizar  $\theta_{\min}$  corresponde a minimizar  $\rho(\tau)$ .

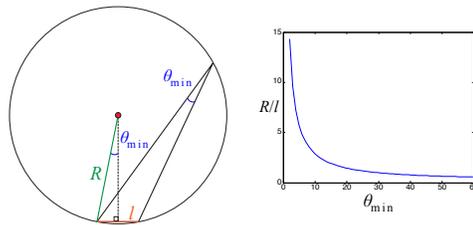


Figura 5.2: Relação entre a razão circunraio-aresta  $R/l$  e o ângulo mínimo  $\theta_{\min}$  de um triângulo. Maximizar  $\theta_{\min}$  corresponde a minimizar  $R/l$ .

Uma outra medida de qualidade de forma de triângulos utilizada é a *razão de aspecto*, dada por  $A(\tau) = r/R$ , em que  $r$  é o inraio e  $R$ , o circunraio de um triângulo  $\tau$ . A razão de aspecto de  $\tau$  e o seu menor

ângulo  $\theta_{\min}$  estão relacionados da seguinte forma (ver Problema 5.2):

$$2 \operatorname{sen}^2 \left( \frac{\theta_{\min}}{2} \right) \leq A(\tau) \leq 2 \tan \left( \frac{\theta_{\min}}{2} \right),$$

o que significa que a razão de aspecto se aproxima de zero na medida em  $\theta_{\min}$  também se aproxima de zero (triângulo degenerado), e vice-versa.

Por fim, é possível provar que a estimativa do erro, devido à interpolação linear de uma função sobre um triângulo, depende das derivadas parciais de segunda ordem da função sendo interpolada e da forma do triângulo. Essa questão é importante porque, conforme veremos no Capítulo 6, ao introduzirmos a formulação do método dos elementos finitos, o valor de uma função  $f$  em um ponto qualquer sobre um triângulo  $\tau$  pode ser aproximado a partir da interpolação linear

$$\tilde{f} = \sum_{i=1}^3 N_i f_i,$$

em que  $f_i$  é o valor de  $f$  avaliado na posição do vértice  $i$ , e  $N_i$  é a função de forma associada ao vértice  $i$ , para  $i = 1, 2, 3$ . O seguinte teorema estabelece a relação entre o erro  $f - \tilde{f}$ , o ângulo mínimo,  $\theta_{\min}$ , e o comprimento da maior aresta,  $h$ , de  $\tau$ .

**Teorema 5.1** ([45]). *Se  $f$  tem derivadas segundas contínuas, então*

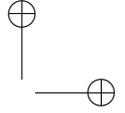
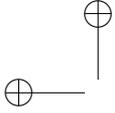
$$\begin{aligned} \|f - \tilde{f}\|_{\infty} &\leq \frac{1}{2} h^2 \|D^2 f\|_{\infty}, \\ \|\nabla(f - \tilde{f})\|_{\infty} &\leq \frac{3}{\operatorname{sen}(\theta_{\min})} h \|D^2 f\|_{\infty}, \end{aligned}$$

em que

$$D^2 f = \left[ \sum_{i,j=1}^2 \left( \frac{\partial^2 f}{\partial x_i \partial x_j} \right)^2 \right]^{1/2}. \quad (5.3)$$

Se  $\nabla f$  é contínuo, então

$$\|f - \tilde{f}\|_{\infty} \leq h \|Df\|_{\infty}.$$



Note que, uma vez que o gradiente da estimativa do erro depende do recíproco do seno do menor ângulo de  $\tau$ , o limite do erro da interpolação aumenta à medida que o triângulo se torna mais fino.

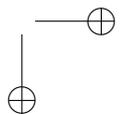
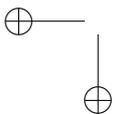
### 5.3 Refinamento de Delaunay

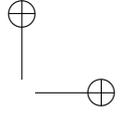
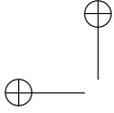
Métodos de geração de malhas baseados em *refinamento de Delaunay* produzem malhas de boa qualidade através da inserção de vértices em um triangulação de Delaunay inicial. Novos vértices são criados e adicionados na triangulação com a finalidade de eliminar triângulos magros. Mais ainda, a própria construção da triangulação de Delaunay fornece uma estratégia de adicionar novos vértices distantes dos demais, evitando a criação de arestas que sejam arbitrariamente mais curtas do que as existentes na triangulação de Delaunay inicial.

Esta seção apresenta um algoritmo, baseado em refinamento de Delaunay, para resolver o *problema de geração de uma malha* de uma região poligonal de  $\mathbb{E}^2$ . Este problema é ligeiramente diferente do problema da triangulação de um subconjunto de pontos em  $\mathbb{E}^2$ , que foi objeto de estudo do Capítulo 4, pois embora uma malha (de triângulos) de uma região poligonal de  $\mathbb{E}^2$  também seja uma triangulação de um certo subconjunto de pontos de  $\mathbb{E}^2$ , uma malha tem de satisfazer, em geral, restrições topológicas e geométricas adicionais e *de otimalidade* que serão detalhadas mais adiante. Logo, podemos imaginar o problema de geração de malhas, que estudaremos aqui, como uma versão de “otimização” do problema de geração de uma triangulação.

#### 5.3.1 Entrada e saída

Para declarar formalmente o problema de geração de malhas de regiões poligonais em  $\mathbb{E}^2$ , precisamos definir em que consistem a entrada e a saída do problema. Em vez de um subconjunto de pontos de  $\mathbb{E}^2$ , a entrada do problema de geração de malhas tem como entrada um complexo simplicial,  $\mathcal{G}$ , em  $\mathbb{E}^2$  cuja dimensão,  $\dim(\mathcal{G})$ , é igual a 1 (ver Definição 4.1). Isto é, o complexo  $\mathcal{G}$  é um conjunto finito de pontos e segmentos de reta em  $\mathbb{E}^2$  que goza das propriedades a seguir:





- (E1) os dois extremos de todo segmento de  $\mathcal{G}$  são vértices de  $\mathcal{G}$  e
- (E2) a interseção entre um vértice e um segmento de  $\mathcal{G}$  é vazia ou é um ponto extremo do segmento; analogamente, a interseção entre dois segmentos de  $\mathcal{G}$  é vazia ou um ponto extremo de ambos.

Para simplificar a descrição do algoritmo de refinamento de Delaunay que estudaremos mais adiante, assumimos que  $\mathcal{G}$  também goza da propriedade (E3): a fronteira,  $\partial(FC(\mathcal{V}))$ , do fecho convexo,  $FC(\mathcal{V})$ , do conjunto,  $\mathcal{V}$ , de vértices de  $\mathcal{G}$  é a união de um subconjunto do conjunto,  $\mathcal{S}$ , de segmentos de  $\mathcal{G}$  (Figura 5.3). Na literatura de geração de malhas, o complexo simplicial  $\mathcal{G}$  é comumente denominado de *grafo planar de segmentos de retas* (GPSR): um grafo  $G = (V, E)$  em que o conjunto,  $V$ , de vértices é  $\mathcal{V}$  e o conjunto,  $E$ , de arestas é  $\mathcal{S}$ . De agora em diante, reservamos o termo *aresta* para designar uma aresta de uma triangulação e o termo *segmento* para designar uma aresta de  $\mathcal{G}$ .

A entrada do problema de geração de malhas também conta com um *parâmetro de otimalidade*,  $\bar{\rho}$ , cujo valor é uma cota inferior para o menor ângulo interno que qualquer triângulo da malha de saída pode conter. Uma entrada para o problema de geração de malhas é, portanto, um par da forma  $(\mathcal{G}, \bar{\rho})$ . O valor de  $\bar{\rho}$  tem de ser escolhido tal que

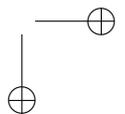
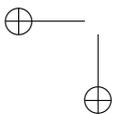
$$\bar{\rho} > \sqrt{2}.$$

Caso contrário, o algoritmo de refinamento de Delaunay pode não terminar, como veremos mais adiante. A cota inferior de  $\bar{\rho}$  implica que o menor ângulo de todos os triângulos é maior ou igual a  $\theta_{\min}$ , com

$$\theta_{\min} = \arcsen\left(\frac{1}{2\bar{\rho}}\right) < \arcsen\left(\frac{1}{2\sqrt{2}}\right) \approx 20,7^\circ$$

e o ângulo máximo,  $\theta_{\max}$ , é, por consequência,  $\theta_{\max} = 180^\circ - 2\theta_{\min} > 139^\circ$ . Assumimos também que quaisquer dois segmentos de  $\mathcal{G}$  não podem definir um ângulo menor do que  $\pi/2$ , seja ele interno ou externo a  $FC(\mathcal{V})$ . Na Seção 5.6, discutimos as mudanças que precisam ser realizadas no algoritmo para que a restrição acima possa ser removida.

**Observação 5.1.** De agora em diante, assumimos que qualquer grafo planar de segmentos de retas (GPSR),  $\mathcal{G}$ , ao qual nos referirmos



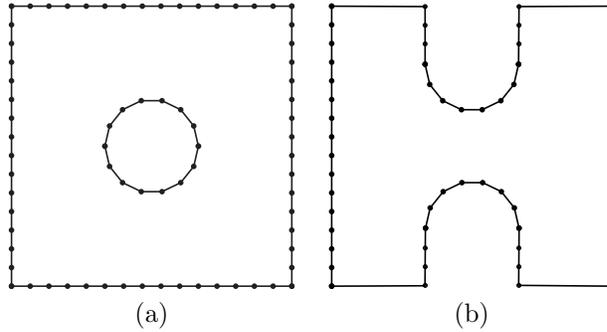


Figura 5.3: (a) Um complexo simplicial que goza das propriedades (E1)-(E3) que toda entrada para o problema de geração de malhas deve possuir. (b) Um complexo simplicial que não goza da propriedade (E3).

*goza das propriedades (E1)-(E3) dadas anteriormente e, portanto, constitui-se em parte de uma entrada para o problema de geração de malhas. Além disso, assumimos também que quaisquer dois segmentos de  $\mathcal{G}$  não definem um ângulo menor do que  $\pi/2$ , seja ele interno ou externo.*

Para definir a saída do problema de geração de malhas, utilizamos a noção de *refinamento* de um GPSR dada a seguir (refira-se à Figura 5.4):

**Definição 5.1.** *Um GPSR  $\mathcal{G}' = (\mathcal{V}', \mathcal{S}')$  é um refinamento do GPSR,  $\mathcal{G} = (\mathcal{V}, \mathcal{S})$ , se, e somente se, temos que  $\mathcal{V} \subset \mathcal{V}'$  e que cada segmento em  $\mathcal{S}$  é um segmento em  $\mathcal{S}'$  ou a união de dois ou mais segmentos em  $\mathcal{S}'$ .*

**Definição 5.2.** *Dado um GPSR  $\mathcal{G} = (\mathcal{V}, \mathcal{S})$ , uma triangulação,  $\mathcal{T}(\mathcal{G})$ , de  $\mathcal{G}$  é uma triangulação do subconjunto,  $\mathcal{V}$ , de pontos de  $\mathbb{E}^2$  tal que todo segmento em  $\mathcal{S}$  é uma aresta de  $\mathcal{T}(\mathcal{G})$ . A triangulação  $\mathcal{T}(\mathcal{G})$  é mais comumente conhecida pelo termo triangulação restrita de  $\mathcal{G}$ .*

A Figura 5.5 exhibe um GPSR  $\mathcal{G}$  e uma triangulação restrita de  $\mathcal{G}$ .

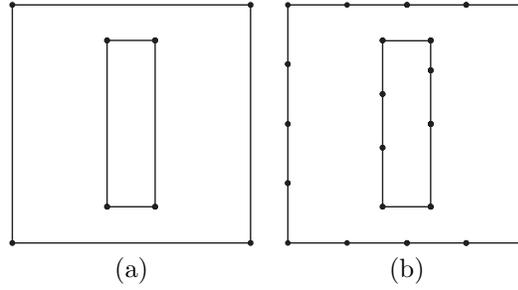


Figura 5.4: (a) Um GPSR  $\mathcal{G}$  e (b) um refinamento,  $\mathcal{G}'$ , de  $\mathcal{G}$ .

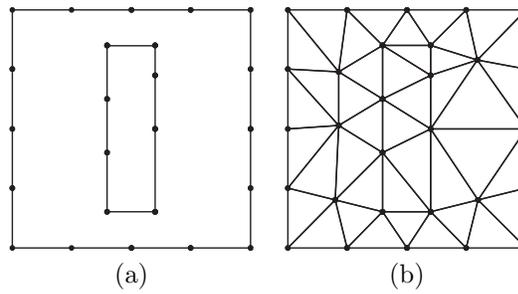
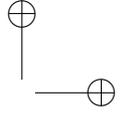
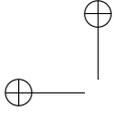


Figura 5.5: (a) Um GPSR  $\mathcal{G}$  e (b) uma triangulação de  $\mathcal{G}$ .

Dada uma entrada,  $(\mathcal{G}, \bar{\rho})$ , para o problema de geração de malhas, em que  $\mathcal{G}$  é um GPSR e  $\bar{\rho}$  não é menor ou igual a  $\sqrt{2}$ , uma solução para o problema é uma triangulação restrita,  $\mathcal{T}(\mathcal{G}')$ , de um GPSR  $\mathcal{G}'$ , em que  $\mathcal{G}'$  é o próprio  $\mathcal{G}$  ou um refinamento de  $\mathcal{G}$ , tal que o menor ângulo de qualquer triângulo de  $\mathcal{T}(\mathcal{G}')$  não é menor do que  $\theta_{\min}$ , em que

$$\theta_{\min} = \arcsen\left(\frac{1}{2\bar{\rho}}\right).$$



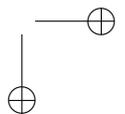
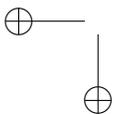
### 5.3.2 O algoritmo de Ruppert

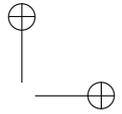
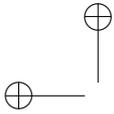
Os algoritmos baseados na técnica conhecida por *refinamento de Delaunay* produzem uma triangulação de Delaunay,  $\mathcal{TD}(P)$ , de um subconjunto,  $P$ , de pontos de  $\mathbb{E}^2$  como solução do problema de geração de malhas em  $\mathbb{E}^2$ . A triangulação  $\mathcal{TD}(P)$  é uma triangulação restrita,  $\mathcal{T}(\mathcal{G}')$ , em que  $\mathcal{G}' = \mathcal{G}$  ou  $\mathcal{G}'$  é um refinamento do GPSR,  $\mathcal{G}$ , de entrada. Em particular, temos que  $P$  é o conjunto,  $\mathcal{V}'$ , de vértices de  $\mathcal{G}'$ .

Em se tratando de algoritmos de refinamento de Delaunay é mais natural definir a saída,  $\mathcal{TD}(P)$ , produzida pelos algoritmos em termos do GPSR,  $\mathcal{G}$ , que compõe a entrada do problema de geração de malhas. Sendo assim, dizemos que os algoritmos de refinamento de Delaunay produzem uma triangulação de Delaunay,  $\mathcal{TD}(P)$ , tal que todo vértice de  $\mathcal{G}$  é um vértice de  $\mathcal{TD}(P)$  (isto é,  $\mathcal{V} \subseteq P$ ) e todo segmento de  $\mathcal{G}$  é uma aresta de  $\mathcal{TD}(P)$  ou a união de arestas de  $\mathcal{TD}(P)$ . Observe que  $\mathcal{TD}(P)$  não é necessariamente uma triangulação restrita de  $\mathcal{G}$ . Este é o caso se, e somente se,  $P = \mathcal{V}$ . Por outro lado, a triangulação  $\mathcal{TD}(P)$  é dita ser uma *triangulação de Delaunay conforme (TDC) de  $\mathcal{G}$* , pois  $\mathcal{TD}(P)$  contém o GPSR  $\mathcal{G}$  ou um refinamento de  $\mathcal{G}$ .

A ideia por trás de qualquer algoritmo de refinamento de Delaunay é calcular  $\mathcal{TD}(P)$  incrementalmente. Para tal, o algoritmo calcula a triangulação de Delaunay,  $\mathcal{TD}(Q)$ , do subconjunto,  $Q$ , de pontos de  $\mathbb{E}^2$  tal que  $Q = \mathcal{V}$ ; isto é, o conjunto  $Q$  é exatamente igual ao conjunto de vértices do GPSR dado como entrada. Em seguida, o algoritmo verifica se (i)  $\mathcal{TD}(Q)$  é uma TDC de  $\mathcal{G}$  e se (ii) o menor ângulo interno de  $\mathcal{TD}(Q)$  é maior ou igual a  $\theta_{\min}$ . Se qualquer uma dessas duas condições é falsa, então o algoritmo acrescenta um ponto  $p$  de  $\mathbb{E}^2$  a  $Q$ , com  $p \in FC(\mathcal{V}) - Q$ , e recalcula  $\mathcal{TD}(Q)$ . Este passo é repetido até que as condições (i) e (ii) acima sejam satisfeitas pela triangulação  $\mathcal{TD}(Q)$  corrente. Quando este for o caso, a execução do algoritmo termina e produz  $\mathcal{TD}(Q)$  como saída, como mostra o Algoritmo 5.1.

A linha 2 do Algoritmo 5.1 corresponde a uma chamada ao Algoritmo 4.2, que estudamos na Seção 4.6, com entrada  $P = Q$ . Por sua vez, o recálculo de  $\mathcal{TD}(Q)$ , na linha 3, após inserirmos o ponto  $p$  em  $Q$  pode ser feito com uma única chamada ao Algoritmo 4.3 com





entrada dada pelo par  $(p, \mathcal{TD}(Q))$ . O que nos resta, portanto, é lidar com os detalhes que cercam a verificação das condições (i) e (ii) na linha 3 e com a garantia de término do algoritmo. Observe que a condição (i) é de *conformidade*, enquanto a condição (ii) é de *otimalidade*. A primeira garante que a saída do algoritmo é uma TDC de  $\mathcal{G}$ , enquanto a segunda garante que a saída do algoritmo é uma triangulação de boa qualidade. Intuitivamente, devemos mostrar que o Algoritmo 5.1 consegue satisfazer as duas condições após um número finito de inserções de pontos no conjunto  $Q = \mathcal{V}$ . Se este for o caso, o algoritmo termina. No entanto, a prova de corretude e término do Algoritmo 5.1 procede às avessas: mostramos que o algoritmo termina e, portanto, a triangulação devolvida tem de satisfazer as condições (i) e (ii).

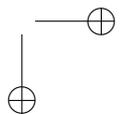
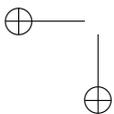
---

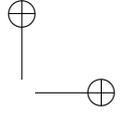
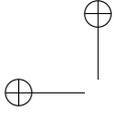
**Algoritmo 5.1** REFINAMENTO DELAUNAY( $\mathcal{G}, \bar{\rho}$ )

---

- 1:  $Q \leftarrow \mathcal{V}$  { $\mathcal{V}$  é o conjunto de vértices de  $\mathcal{G}$ }
  - 2: calcule  $\mathcal{TD}(Q)$
  - 3: se **não** ( (i)  $\mathcal{TD}(Q)$  é uma TDC de  $\mathcal{G}$  e se (ii) o menor ângulo interno de  $\mathcal{TD}(Q)$  é maior ou igual a  $\arcsen(1/2\bar{\rho})$  ), então escolha  $p \in FC(\mathcal{V})$ , faça  $Q \leftarrow Q \cup \{p\}$ , recalcule a triangulação  $\mathcal{TD}(Q)$  e repita o passo.
  - 4: **devolva**  $\mathcal{TD}(Q)$
- 

Para garantir o término do algoritmo, valemo-nos da seguinte observação: se escolhermos o ponto  $p$ , a ser inserido em  $Q$  na linha 3 de Algoritmo 5.1, de tal forma que a distância entre  $p$  e o ponto mais próximo dele em  $Q$  seja sempre maior ou igual ao comprimento da aresta mais curta da triangulação calculada na linha 1, haverá um momento em que não existirá nenhum ponto  $p$  em  $FC(\mathcal{V})$  que satisfaça o critério de escolha. Devemos mostrar que este é o caso após um número finito de inserções de pontos em  $Q$ , é claro. Além disso, devemos mostrar que quando não tivermos uma escolha para  $p$  — *de acordo com o critério de escolha acima* — as condições (i) e (ii) estarão satisfeitas. O lema a seguir nos diz que, se o critério de escolha acima for adotado, então o algoritmo não pode inserir infinitos pontos em  $Q$  e, portanto, terminará, no pior caso, por falta de opção.





**Lema 5.2** (Lema do Empacotamento [23]). *Seja  $D$  um conjunto limitado de  $\mathbb{E}^2$ . Sejam  $P \subset D$  um conjunto de pontos de  $D$  e  $\lambda$  um número real positivo tais que, para todo par de pontos distintos,  $u$  e  $v$ , em  $S$ , a distância,  $d(u, v)$ , entre  $u$  e  $v$  não é menor do que  $\lambda$ . Então, há uma constante  $\xi$ , que depende apenas de  $D$  e  $\lambda$ , tal que  $|P| \leq \xi$ .*

*Demonstração.* Seja

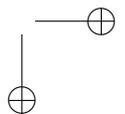
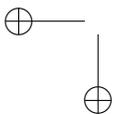
$$D_{\lambda/2} = \{x \in \mathbb{E}^2 \mid d(x, D) \leq \lambda/2\},$$

a soma de Minkowski de  $D$  com uma bola euclidiana,  $B(c, \lambda/2)$ , de centro  $c \in D$  e raio  $\lambda/2$ , em que a distância de um ponto a um conjunto é dada por  $d(x, D) = \inf\{d(x, y) \mid y \in D\}$ . Toda bola  $B(p, \lambda/2)$  com um centro em  $p \in P$  está em contida em  $D_{\lambda/2}$ . Além disso,  $D_{\lambda/2}$  é limitado, pelo fato de  $D$  ser limitado. Por hipótese temos que  $d(p, q) \geq \lambda$ , para todo  $p, q \in P$ . Logo,  $B(p, \lambda/2) \cap B(q, \lambda/2) = \emptyset$ , para todo  $p, q \in P$ . Esse conjunto de bolas é chamado de *empacotamento*. Portanto, pelo fato de  $\text{área}(B(p, \lambda/2)) = \pi\lambda^2/4$ , para todo ponto  $p \in P$ , podemos concluir que a cardinalidade,  $|P|$ , de  $P$  é limitada acima por:

$$|P| \leq \frac{4}{\pi\lambda^2} \cdot \text{área}(D_{\lambda/2}) = \xi.$$

□

Por ora, vamos ignorar a condição (i) da linha 3 do Algoritmo 5.1 e vamos nos concentrar apenas na condição (ii). Para determinar se o menor ângulo,  $\theta_\tau$ , de um triângulo  $\tau$  da triangulação  $\mathcal{TD}(Q)$  é menor do que  $\theta_{\min}$ , faremos uso da razão circunraio-aresta,  $\rho(\tau)$ , de  $\tau$  (ver Equação (5.1)), pois  $\theta_\tau < \theta_{\min}$  se, e somente se,  $\rho(\tau) > \bar{\rho}$ , com  $\bar{\rho} > \sqrt{2}$ . Daí, a condição (ii) pode ser escrita como  $\rho(\tau) \leq \bar{\rho}$ , em que  $\tau$  é qualquer triângulo de  $\mathcal{TD}(Q)$ . Vamos assumir que se  $\tau$  não satisfaz a condição (ii), então o ponto  $p$  é escolhido como o centro do círculo,  $C_\tau$ , circunscrito a  $\tau$ . Seja  $\lambda$  a menor distância entre quaisquer dois pontos do conjunto de vértices,  $Q$ , antes de  $p$  ser inserido em  $Q$ . Sejam  $r$  o raio de  $C_\tau$  e  $l$  o comprimento da aresta mais curta de  $\tau$ . Como  $\mathcal{TD}(Q)$  é uma triangulação de Delaunay antes da inserção de  $p$  em  $Q$ , o círculo  $C_\tau$  não pode conter nenhum ponto de  $Q$  dentro dele (Observação 4.8). Logo, temos que  $d(p, Q) \geq r = \rho(\tau) \cdot l > \bar{\rho} \cdot \lambda$ .



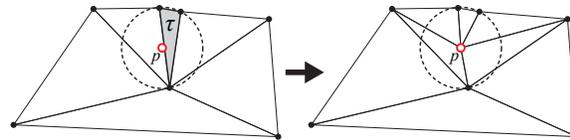
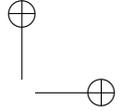
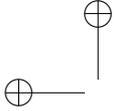


Figura 5.6: Inserção de um vértice no circuncentro  $p$  do triângulo  $\tau$ .

Há, entretanto, um pequeno “furo” na argumentação acima. De acordo com a linha 3 do Algoritmo 5.1, o ponto  $p$  a ser inserido em  $Q$  é um ponto do fecho convexo,  $FC(\mathcal{V})$ , do conjunto de vértices,  $\mathcal{V}$ , de  $\mathcal{G}$ . Infelizmente, o circuncentro de  $\tau$  (isto é, o centro de  $C_\tau$ ) pode não pertencer a  $FC(\mathcal{V})$ . Surpreendentemente, há uma solução para o problema da escolha de  $p$ , quando  $p \notin FC(\mathcal{V})$ , que faz com que a triangulação  $\mathcal{TD}(Q)$  devolvida pelo Algoritmo 5.1 satisfaça tanto a condição (i) quanto a condição (ii). Esta solução requer que mudemos o foco da nossa estratégia: primeiro, garantimos que a condição (i) é satisfeita e, só depois, garantimos a condição (ii). Para descrever a solução, necessitamos da definição de ponto invasor e segmento invadido.

**Definição 5.3** (invasão). *Um vértice  $v$  de  $\mathcal{TD}(Q)$  que reside no círculo,  $C_s$ , que possui o segmento  $s$  como diâmetro, mas não é um ponto extremo de  $s$ , é denominado de invasor. Dizemos que  $v$  invadiu (os limites do) segmento  $s$  e que  $s$  é um segmento invadido por  $v$ .*

Para satisfazer a condição (i), o Algoritmo 5.1 deve forçar cada segmento de  $\mathcal{G}$  a ser representado por uma aresta de  $\mathcal{TD}(Q)$  ou uma união de arestas de  $\mathcal{TD}(Q)$ . Para tal, durante o refinamento, o Algoritmo 5.1 mantém um conjunto,  $E$ , de *subsegmentos*. Inicialmente, o



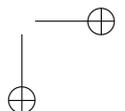
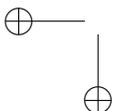
conjunto  $E$  é o próprio conjunto,  $\mathcal{S}$ , de segmentos de  $\mathcal{G}$ , mas quando o algoritmo insere novos pontos em  $Q$  (na linha 3), esses segmentos serão subdivididos em subsegmentos e esses subsegmentos poderão ser subdivididos em subsegmentos mais curtos e assim por diante. Vamos assumir que o Algoritmo 5.1 tenha uma forma de identificar, unicamente, cada segmento. De agora em diante, o termo *subsegmento* é usado denotar para qualquer elemento de  $E$ , seja ele um segmento de  $\mathcal{G}$  ou um subsegmento gerado a partir da divisão de algum segmento de  $\mathcal{G}$ .

**Observação 5.2.** *A solução para satisfazer a condição (i) se vale da seguinte: se  $s \in E$  não é um subsegmento invadido por um ponto em  $Q$ , então  $s$  é uma aresta de Delaunay com relação a  $Q$  e, portanto, está em  $\mathcal{TD}(Q)$  (Proposição 4.15). Por outro lado, se  $s \in E$  é um subsegmento invadido por um ponto em  $Q$ , então não podemos afirmar que  $s$  pertence a  $\mathcal{TD}(Q)$ , pois pode haver um círculo,  $C$ , (não necessariamente o círculo diametral), do qual  $s$  é uma corda e tal que*

$$\text{int}(C) \cap Q \neq \emptyset.$$

Suponha que optemos por definir a condição (i) da seguinte forma: *nenhum subsegmento  $s$  em  $E$  é invadido por um vértice de  $\mathcal{TD}(Q)$ . Se tal condição for falsa, então existe um subsegmento  $s$  em  $E$  que foi invadido por um vértice de  $\mathcal{TD}(Q)$ . Pela Observação 5.2,  $s$  pode, inclusive, ser uma aresta de  $\mathcal{TD}(Q)$ . No entanto, vamos supor que, neste caso, a linha 3 do Algoritmo 5.1 sempre execute o Algoritmo 5.2. Este procedimento insere o ponto médio,  $p$ , de  $s$  em  $Q$ , remove  $s$  de  $E$  e insere os dois subsegmentos que resultam da divisão de  $s$  em dois subsegmentos de mesmo tamanho (ver Figura 5.7). É imediato ver que se a condição (i) for verdadeira, então todo subsegmento de  $\mathcal{G}$  é uma aresta da  $\mathcal{TD}(Q)$  atual ou uma união de arestas dela. A transformação de  $\mathcal{TD}(Q)$  em uma triangulação conforme do GPSR  $\mathcal{G}$ , pela subdivisão sucessiva de subsegmentos de  $\mathcal{G}$ , é denominada *costura*.*

Precisamos mostrar que o processo de costura não implica na inserção de uma sequência infinita de pontos em  $Q$ . Isto será feito mais adiante. Por ora, vamos voltar nossa atenção novamente para a condição (ii). Como provaremos mais adiante, quando a condição (i) é satisfeita, o circuncentro de qualquer triângulo de  $\mathcal{TD}(Q)$  pertence



---

**Algoritmo 5.2** DIVIDESUBSEGMENTO( $s, Q, E$ )

---

- 1: calcule o ponto médio  $p$  de  $s$
  - 2: troque  $s$  por seus dois subsegmentos (metades) em  $E$
  - 3: **devolva**  $p$
- 

a  $FC(\mathcal{V})$ . Em princípio, isto nos permite tratar a condição (ii) como vimos anteriormente. No entanto, há um outro obstáculo: embora o circuncentro de um triângulo pertença a  $FC(\mathcal{V})$ , ele pode se tornar tão próximo de uma aresta de  $\mathcal{TD}(Q)$  em  $\partial(FC(\mathcal{V}))$  quanto possível. Conseqüentemente, o Lema 5.2 não mais poderia ser aplicado para garantir o término do algoritmo. Porém, uma ligeira modificação na solução que propomos anteriormente para garantir a condição (ii) remove o último obstáculo que temos pela frente (ver Algoritmo 5.3).

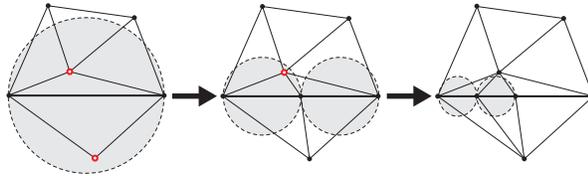


Figura 5.7: Segmentos são divididos recursivamente até que nenhum subsegmento seja mais invadido por um vértice da triangulação atual.

Seja  $\tau$  um triângulo de  $\mathcal{TD}(Q)$  tal que  $\rho(\tau) > \bar{\rho}$ . Então, nós consideramos a inserção do circuncentro,  $p$ , de  $\tau$  em  $Q$ , como fizemos antes. Porém, se  $p$  invadir (os limites de) um subsegmento  $s$ , rejeitamos  $p$  e inserimos o ponto médio,  $m$ , de  $s$  em  $Q$  (Figura 5.8). Se  $m$  pertencer ao círculo circunscrito a  $\tau$ , o triângulo  $\tau$  não pertencerá à nova triangulação  $\mathcal{TD}(Q)$ . Se não, o triângulo  $\tau$  sobreviverá, mas será removido de  $\mathcal{TD}(Q)$  em uma execução futura da linha 3 do Algoritmo 5.1. O algoritmo resultante da definição das condições (i) e (ii), como acabamos de descrever, é denominado de *algoritmo de Ruppert* [113]. O Algoritmo 5.4 é apenas uma das possíveis especializações do algoritmo de refinamento (genérico) dado em Algoritmo 5.1. Mostramos a seguir que se a condição (i) é satisfeita, então o circuncentro de todo triângulo da triangulação  $\mathcal{TD}(Q)$  atual está

em  $FC(\mathcal{V})$ . A prova de término do Algoritmo 5.4 será tratada em uma seção exclusiva, a Seção 5.4, pois ela não é uma mera aplicação do Lema 5.2.

---

**Algoritmo 5.3** DIVIDETRIÂNGULO( $\tau, Q, E$ )

---

- 1: calcule o circuncentro  $p$  de  $\tau$
  - 2: **se** existir um subsegmento  $s \in E$  invadido por  $p$  **então**
  - 3:    $p \leftarrow \text{DIVIDESUBSEGMENTO}(s, Q, E)$
  - 4: **fim se**
  - 5: **devolva**  $p$
- 

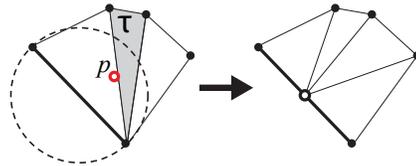
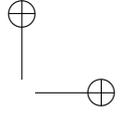
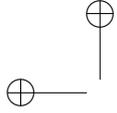


Figura 5.8: O circuncentro de  $\tau$  não é inserido pelo algoritmo, pois ele invade um subsegmento. Em vez dele, o ponto médio do subsegmento é inserido.

**Proposição 5.3.** *Seja  $\mathcal{T}(\mathcal{G})$  uma triangulação de Delaunay conforme de  $\mathcal{G}$ . Se nenhuma aresta de  $\mathcal{T}(\mathcal{G})$ , que é um segmento ou está contida em um segmento de  $\mathcal{G}$ , é um subsegmento invadido por um vértice de  $\mathcal{T}(\mathcal{G})$ , então o centro do círculo circunscrito a qualquer triângulo de  $\mathcal{T}(\mathcal{G})$  pertence a  $FC(\mathcal{V})$ , em que  $\mathcal{V}$  é o conjunto de vértices de  $\mathcal{G}$ .*

*Demonstração.* Suponha, por contradição, que o centro,  $c$ , do círculo,  $C_\tau$ , circunscrito a algum triângulo,  $\tau$ , de  $\mathcal{T}(\mathcal{G})$  está no exterior de  $FC(\mathcal{V})$ . Seja  $q$  qualquer ponto no interior do triângulo  $\tau$  (refira-se à Figura 5.9). Como  $c$  está no exterior de  $FC(\mathcal{V})$  e  $\tau$  pertence a  $FC(\mathcal{V})$ , o segmento  $\overline{qc}$  cruza a fronteira,  $\partial(FC(\mathcal{V}))$ , de  $FC(\mathcal{V})$  e, portanto, deve cruzar um subsegmento  $s$  que pertence a  $\partial(FC(\mathcal{V}))$ , pois  $\partial(FC(\mathcal{V}))$  é uma união de subsegmentos de  $\mathcal{G}$ , por hipótese. Nós mostraremos que  $s$  é um subsegmento invadido por um vértice de  $\mathcal{T}(\mathcal{G})$ . Seja  $C_s$  o círculo diametral de  $s$ . Seja  $\ell$  uma linha orientada que contém  $s$  e



suponha que  $\ell = Nuc(g_s)$ , em que  $g_s : \mathbb{E}^2 \rightarrow \mathbb{R}$  é uma forma afim. Sem perda de generalidade, suponha que  $q \in H_+(g_s)$  e  $c \in H_-(g_s)$ . Por construção, temos que  $C_s \cap H_+(g_s)$  é um semicírculo. Observe que o interior,  $int(C_\tau)$ , de  $C_\tau$  intersecta  $s$ , pois  $\overline{qc} \subset int(C_\tau)$  e os vértices de  $s$  não pertencem a  $int(C_\tau)$ , já que  $\tau$  é um triângulo de Delaunay. Mas, como  $c \in H_-(g_s)$ , temos que  $\tau \subset (H_+(g_s) \cap C_\tau)$  e  $(H_+(g_s) \cap C_\tau) \subset C_s$ . Logo,  $C_s$  contém todos os três vértices de  $\tau$ . Dois desses vértices podem ser os vértices de  $s$ , mas o terceiro é um vértice invasor, o que implica que  $s$  é um subsegmento invadido: um absurdo.  $\square$

---

**Algoritmo 5.4** GERAMALHA( $\mathcal{G}, \bar{\rho}$ )

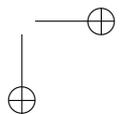
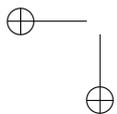
---

```

1:  $Q \leftarrow \mathcal{V}$ 
2: calcule  $\mathcal{TD}(Q)$ 
3:  $E \leftarrow$  todos os subsegmentos invadidos de  $\mathcal{G}$ 
4:  $R \leftarrow$  todos os triângulos  $\tau$  de  $\mathcal{TD}(Q)$  tais que  $\rho(\tau) > \bar{\rho}$ 
5: enquanto  $E \neq \emptyset$  ou  $R \neq \emptyset$  faça
6:   enquanto  $E \neq \emptyset$  faça
7:     retire um subsegmento  $s$  de  $E$ 
8:      $p \leftarrow$  DIVIDESUBSEGMENTO( $s, Q, E$ )
9:     insira  $p$  em  $\mathcal{TD}(Q)$ 
10:     $Q \leftarrow Q \cup \{p\}$ 
11:   fim enquanto
12:   se  $R \neq \emptyset$  então
13:     retire um triângulo  $\tau$  de  $R$ 
14:      $p \leftarrow$  DIVIDETRIÂNGULO( $\tau, Q, E$ )
15:     insira  $p$  em  $\mathcal{TD}(Q)$ 
16:      $Q \leftarrow Q \cup \{p\}$ 
17:   fim se
18: remova de  $R$  triângulos que não mais existem em  $\mathcal{TD}(Q)$ 
19: insira em  $E$  novos subsegmentos invadidos por um vértice de
    $Q$ 
20: insira em  $R$  novos triângulos  $\tau$  de  $\mathcal{TD}(Q)$  tais que  $\rho(\tau) > \bar{\rho}$ 
21: fim enquanto
22: devolva  $\mathcal{TD}(Q)$ 

```

---



### 5.3.3 Implementação

Assim como ocorre com a implementação do algoritmo de triangulação incremental (aleatorizado) para gerar a triangulação de Delaunay (ver Seção 4.6), há um longo caminho a ser percorrido entre o pseudocódigo e a implementação do algoritmo de Ruppert. O primeiro passo é a *extensão* de uma estrutura de dados eficiente para manter triangulações. A extensão se deve ao fato de necessitarmos armazenar o conjunto,  $E$ , de subsegmentos invadidos e o conjunto,  $R$ , de triângulos magros, ou seja, os triângulos para os quais a razão circunraio-aresta é maior do que  $\bar{\rho}$ . Idealmente, precisamos escolher uma estrutura de dados para  $E$  e  $R$  que nos permita inserir, remover e consultar elementos da forma mais eficiente possível. Em geral, utilizamos uma fila (comum ou de prioridades) para representar  $E$  e  $R$  e criamos apontadores de elementos da fila para seus correspondentes na triangulação e vice-versa para realizar cada operação em tempo constante.

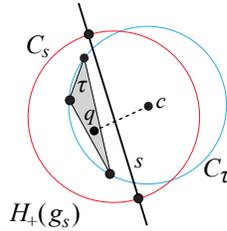
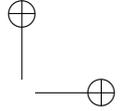
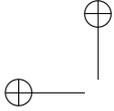


Figura 5.9: Ilustração do argumento da prova da Proposição 5.3.

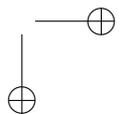
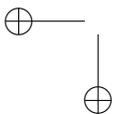
A inicialização da fila  $R$ , na linha 4 do Algoritmo 5.4, pode ser feita com  $\mathcal{O}(n_t)$  operações primitivas, em que  $n_t$  é o número de triângulos da triangulação de Delaunay inicial,  $\mathcal{TD}(\mathcal{V})$ . Simplesmente, percorremos cada triângulo  $\tau$  de  $\mathcal{TD}(\mathcal{V})$ , calculamos  $\rho(\tau)$  e inserimos  $\tau$  em  $R$  se, e somente se,  $\rho(\tau) > \bar{\rho}$ . Para inicializar  $E$ , percorremos a lista de segmentos de  $\mathcal{G}$  e, para cada segmento  $s$  dessa lista, verificamos se existe uma aresta em  $\mathcal{TD}(\mathcal{V})$  conectando os dois vértices de  $s$ . Se não houver, podemos inserir  $s$  em  $E$ . Caso contrário, precisamos determinar se  $s$  é ou não é um segmento invadido. Para

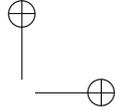
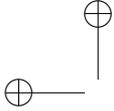


tal, valemo-nos da seguinte observação: toda aresta de  $\mathcal{TD}(Q)$  que corresponde a um subsegmento invadido é adjacente a um triângulo de cujo ângulo oposto à aresta é maior ou igual a  $90^\circ$  graus. Logo, o segmento  $s$  é inserido em  $E$  se, e somente se, ele é uma aresta compartilhada por dois triângulos e o ângulo oposto a  $s$ , em pelo menos um deles, é maior ou igual a  $90^\circ$ . Logo, a inicialização da fila  $E$  requer  $\mathcal{O}(h+n_t)$  operações primitivas, em que  $h$  é o número de segmentos de  $\mathcal{G}$  que não presentes em  $\mathcal{TD}(V)$ . Como veremos na Subseção 5.3.5, o valor de  $h$  não excede  $n_e$ , o número de arestas de  $\mathcal{TD}(V)$ . Do enunciado do Problema 4.15, temos que  $\mathcal{O}(h+n_t) = \mathcal{O}(n_t)$  e, portanto, as linhas 3 e 4 do Algoritmo 5.4 requerem  $\mathcal{O}(n_t)$  operações primitivas (cada).

Para acelerar a execução do algoritmo, podemos optar por não remover de  $R$  — na linha 18 do Algoritmo 5.4 — os triângulos que não mais existem em  $\mathcal{TD}(Q)$ . Mais especificamente, marcamos os triângulos que deveriam ser removidos como *inexistentes* e deixamos que eles sejam retirados de  $R$  na execução da linha 13. No entanto, só devemos executar a linha 14 após testarmos e descobrirmos que o triângulo não está marcado como inexistente. Observe que triângulos são removidos da  $\mathcal{TD}(Q)$  atual pelo Algoritmo 4.3, que é o responsável por atualizar  $\mathcal{TD}(Q)$  com a inserção de um ponto em  $Q$ . Então, este algoritmo pode ser modificado para devolver uma lista com os triângulos removidos. Por outro lado, quando um ponto é inserido em  $\mathcal{TD}(Q)$ , apenas os triângulos criados necessitam passar pelo teste de qualidade (isto é,  $\rho(\tau) > \bar{\rho}$ ) e apenas as arestas desses novos triângulos necessitam ser verificadas para saber se elas correspondem a subsegmentos invadidos. O Algoritmo 4.3 também pode devolver uma lista com os novos triângulos (e suas arestas) de modo que as operações das linhas 19 e 20 do Algoritmo 5.4 sejam feitas rapidamente.

A maneira mais fácil de determinarmos se o centro,  $p$ , do círculo circunscrito a um triângulo magro invade um subsegmento (ver linha 2 de Algoritmo 5.3) é inserir  $p$  na triangulação atual  $\mathcal{TD}(Q)$  e, em seguida, verificar as arestas opostas a  $p$  em todos os triângulos de  $\mathcal{TD}(Q \cup \{p\})$  incidentes em  $p$ . Por observações que fizemos anteriormente, apenas essas arestas podem representar novos subsegmentos invadidos. Além disso, a verificação é simples: o subsegmento correspondente à aresta é invadido se, e somente se, o ângulo definido em  $p$





é maior ou igual a  $90^\circ$ . É vantajoso armazenar os triângulos e arestas removidos de  $\mathcal{TD}(Q)$  com a inserção de  $p$ , pois se  $p$  for rejeitado, então  $\mathcal{TD}(Q)$  pode ser obtida a partir de  $\mathcal{TD}(Q \cup \{p\})$  mais rapidamente. É importante perceber que um triângulo,  $\tau$ , removido de  $\mathcal{TD}(Q)$  durante a geração de  $\mathcal{TD}(Q \cup \{p\})$  só deve ser inserido em  $R$ , após  $\mathcal{TD}(Q)$  ser restaurada, se ele estava em  $R$  antes de  $\mathcal{TD}(Q \cup \{p\})$  ser gerada. Isto vale, também, para o triângulo cujo circuncentro foi rejeitado, pois este mesmo circuncentro pode ser inserido em  $\mathcal{TD}(Q)$  no futuro.

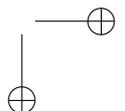
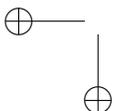
Avaliações experimentais do algoritmo de Ruppert indicam que se (1) a cota superior,  $\bar{\rho}$ , para a razão circuncentro-aresta é (bem) menor <sup>1</sup> do que  $\sqrt{2}$  e se (2)  $R$  for uma fila de prioridades, tal que quanto maior o circunraio do triângulo maior a prioridade, então o número de triângulos da malha final pode ser reduzido em até 35% [23]. No entanto, se  $\bar{\rho}$  é maior do que  $\sqrt{2}$ , então a ordem de processamento dos triângulos magros parece ter pouca influência no número de triângulos da malha final. Além disso, a inserção de circuncentros de triângulos muito magros tende a eliminar mais triângulos magros do que a inserção de circuncentros de triângulos moderadamente magros.

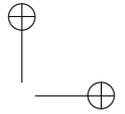
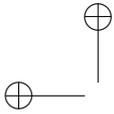
### 5.3.4 Complexidade

A análise da complexidade do algoritmo de Ruppert ainda é tópico sujeito à pesquisa, embora muitos resultados quase “definitivos” tenham sido publicados recentemente. Barbič e Miller [7] mostraram que, para algum natural  $n$  e qualquer valor de  $\bar{\rho}$ , com  $\bar{\rho} > 0$ , existe um GPSR  $\mathcal{G}$  com exatamente  $n + 4$  vértices, distribuídos em uma caixa de dimensões  $1 \times \alpha \cdot n$ , em que  $\alpha$  é um real positivo, tal que a malha produzida pelo algoritmo de Ruppert na entrada  $(\mathcal{G}, \bar{\rho})$  possui  $\Omega(n^2)$  vértices, o que nos diz que o pior caso do algoritmo de Ruppert requer  $\Omega(n^2)$  operações primitivas para produzir uma malha da entrada.

Para provar a cota inferior,  $\Omega(n^2)$ , Barbič e Miller mostraram que há uma sequência de escolha de triângulos magros a serem eliminados que faz com que o algoritmo insira  $N$  pontos na malha inicial,

<sup>1</sup>Um valor menor do que  $\sqrt{2}$  pode fazer com que o algoritmo de Ruppert não termine.





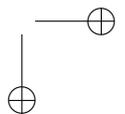
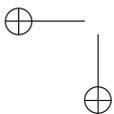
com  $N \in \Omega(n^2)$ . Por outro lado, o próprio Miller [83] propôs uma modificação do algoritmo de refinamento de Delaunay que requer um número subquadrático,  $\mathcal{O}((n \lg \Gamma + N) \lg N)$ , de operações primitivas para gerar a malha final, em que  $\Gamma$  é um parâmetro limitado superiormente pela razão entre a maior distância,  $L$ , entre dois vértices de  $\mathcal{G}$  e a menor distância,  $l$ , entre dois elementos disjuntos (isto é, dois vértices, um vértice e um segmento ou dois segmentos) de  $\mathcal{G}$ . Logo em seguida, Har-Peled e Üngör [63] propuseram um outro algoritmo de refinamento de Delaunay cujo tempo de pior caso está em  $\mathcal{O}(n \lg n + N)$ .

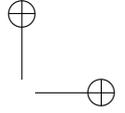
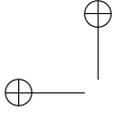
O fator  $n \lg n$  se deve ao tempo para construir a triangulação inicial,  $\mathcal{TD}(\mathcal{V})$ . Logo, o algoritmo é ótimo e é considerado o *primeiro algoritmo ótimo para refinamento de Delaunay*. Entre as principais inovações estão o uso de uma estratégia que insere pontos na “vizinhança” do circuncentro dos triângulos magros [133] e a adoção de uma estrutura de dados auxiliar que possibilita que os pontos inseridos possam ser calculados, localizados e inseridos muito rapidamente. Hudson, Miller e Phillips [67] propuseram outro algoritmo ótimo para refinamento de Delaunay, mas que se baseia no diagrama de Voronoi. Notamos, porém, que avaliações experimentais do algoritmo de Ruppert (Algoritmo 5.4) indicam que o seu tempo de execução, *em entradas tipicamente encontradas na prática*, está em  $\mathcal{O}(n \lg n + N)$  [23].

### 5.3.5 A triangulação de Delaunay restrita

Dedicamos esta seção à discussão sobre como remover uma restrição que impusemos (propositalmente) à entrada do problema de geração de malhas para facilitar a nossa exposição do algoritmo de Ruppert na Subseção 5.3.2. Considere o complexo simplicial na Figura 5.3(b). Este complexo não goza da propriedade (E3) da definição de uma entrada do problema da geração de malhas. No entanto, o espaço subjacente desse complexo é semelhante à fronteira de inúmeros domínios planares de aplicações que demandam o uso de malhas em  $\mathbb{E}^2$ . Se não levarmos em conta este tipo de entrada, limitaremos em demasiado a aplicação do algoritmo de Ruppert a problemas práticos.

Vamos, portanto, definir a entrada do problema de uma forma

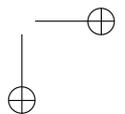
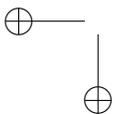


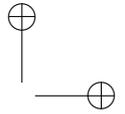
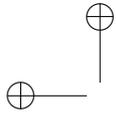


mais flexível. Para tal, vamos fazer uso da noção de região poligonal [14]. Uma *região poligonal*,  $\mathcal{R}$ , é um subconjunto conexo, limitado e fechado de  $\mathbb{E}^2$  cuja fronteira,  $\partial(\mathcal{R})$ , é formada por uma ou mais curvas poligonais simples, fechadas e disjuntas. Intuitivamente, uma região poligonal é um polígono em  $\mathbb{E}^2$  que pode possuir “buracos”. A grande maioria dos domínios planares dos problemas cujas soluções requerem o uso de uma malha pode ser representada por regiões poligonais (ver Capítulo 6). Agora, vamos assumir que a entrada do algoritmo de Ruppert é uma região poligonal,  $\mathcal{R}$ , dada na forma de um GPSR,  $\mathcal{G}$ . Em outras palavras, *todo vértice e todo segmento de  $\mathcal{G}$  está contido em  $\mathcal{R}$  e  $\partial(\mathcal{R})$  é uma união de arestas e segmentos de  $\mathcal{G}$ .*

De acordo com a nossa nova definição de entrada para o problema de geração de malhas, o GPSR da Figura 5.3(b) é uma entrada perfeitamente válida. Por outro lado, o GPSR da Figura 5.3(b) pode representar duas regiões poligonais distintas: uma que possui um buraco circular ou outra que corresponde a um retângulo. Nesta última, a curva poligonal mais interna não é uma curva da fronteira da região, mas sim um conjunto de vértices e segmentos que devem fazer parte da triangulação do retângulo. Para evitar ambiguidades como esta, vamos assumir que os segmentos do GPSR  $\mathcal{G}$  dado como entrada para o algoritmo são *rotulados*. Os segmentos e vértices de  $\mathcal{G}$  que pertencem à fronteira de  $\mathcal{R}$  são denominados *segmentos e vértices de fronteira*, respectivamente. Os demais, cujo interior pertence ao interior de  $\mathcal{R}$ , são denominados de *segmentos e vértices internos*. Assumimos que os rótulos dos vértices e segmentos de  $\mathcal{G}$  são informados ao algoritmo, juntamente com as informações geométricas (isto é, coordenadas) e topológicas (isto é, as relações de adjacência). A Figura 5.10(a) mostra uma triangulação do GPSR da Figura 5.4(a) quando os vértices e segmentos correspondentes ao retângulo menor são rotulados como internos.

Há, no entanto, um problema com a nossa definição de entrada: o algoritmo de Ruppert sempre gera uma triangulação de Delaunay como saída e o espaço subjacente deste tipo de triangulação é, por definição, um conjunto *convexo*; isto é, o fecho convexo dos vértices da triangulação. Felizmente, sabemos que todos os vértices e segmentos do GPSR  $\mathcal{G}$  são vértices e (união de) arestas da triangulação de saída. Logo, tudo que temos de fazer é descartar os triângulos, arestas e





vértices que não fazem parte da região poligonal representada pelo GPSR.

Embora a solução descrita acima funcione, ela não é adotada na prática, pois o algoritmo pode fazer mais “esforço” do que o necessário; por exemplo, melhorando a qualidade de triângulos que serão descartados mais adiante. A solução adotada consiste em mudar o tipo de triangulação inicial. Mais especificamente, em vez de construir uma triangulação de Delaunay,  $\mathcal{TD}(\mathcal{V})$ , do conjunto de vértices,  $\mathcal{V}$ , de  $\mathcal{G}$ , construímos uma *triangulação de Delaunay restrita (TDR) de  $\mathcal{G}$*  [24].

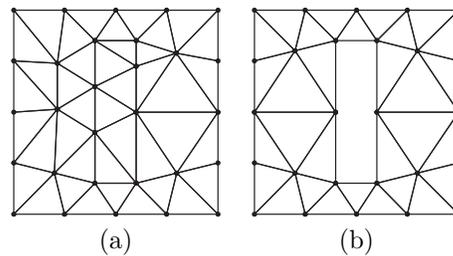
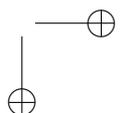
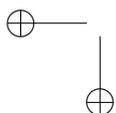


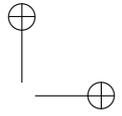
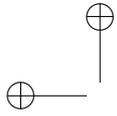
Figura 5.10: Duas possíveis triangulações de um mesmo GPSR.

Uma TDR de  $\mathcal{G}$  é a triangulação restrita de  $\mathcal{G}$  “mais próxima” de  $\mathcal{TD}(\mathcal{V})$ . Formalmente, uma triangulação  $\mathcal{T}$  é uma TDR de  $\mathcal{G}$ , denotada por  $\mathcal{TD}(\mathcal{G})$ , se, e somente se, ela é uma triangulação restrita de  $\mathcal{G}$  tal que para cada aresta  $e$  de  $\mathcal{T}$  que não é um segmento de  $\mathcal{G}$ , existe um círculo  $C_e$  que goza das seguintes propriedades: (1) os vértices de  $e$  pertencem à circunferência de  $C_e$  e (2) se  $v$  é qualquer vértice de  $\mathcal{G}$  (isto é, de  $\mathcal{T}$ ) tal que  $v$  pertence ao interior de  $C_e$ , então  $v$  não pode ser “visto” a partir de pelo menos um dos vértices de  $e$ ; isto é, pelo menos um dos dois segmentos de reta que conectam  $v$  a um dos vértices de  $e$  devem intersectar o interior de um segmento de  $\mathcal{G}$ .

A Figura 5.11 ilustra a definição de triangulação de Delaunay restrita.

Podemos construir  $\mathcal{TD}(\mathcal{G})$  a partir de  $\mathcal{TD}(\mathcal{V})$  usando trocas de arestas apenas. Para tal, inserimos em  $\mathcal{TD}(\mathcal{V})$  cada segmento  $s$  de  $\mathcal{G}$  que não está presente nela e determinamos os triângulos de  $\mathcal{TD}(\mathcal{V})$





intersectados por  $s$ . A união desses triângulos define um polígono denominado de *região de influência*. Os triângulos contidos na região de influência, juntamente com suas arestas e vértices, definem uma triangulação da região. Dyn, Goren e Rippa provaram que esta triangulação pode ser convertida em qualquer outra triangulação do mesmo polígono [41]. A conversão entre triangulações pode ser realizada apenas com trocas de arestas. O nosso objetivo é encontrar uma das triangulações que tenha  $s$  como aresta. Para tal, as arestas trocadas são exatamente aquelas que intersectam  $s$  e a triangulação obtida é tal que todas as arestas, com exceção de uma,  $\nu$ , estão de um lado ou de outro de uma linha orientada passando pelos extremos de  $s$  (ver Figura 5.12). O último passo consiste em trocar  $\nu$  por  $s$ . Uma explicação detalhada deste algoritmo pode ser encontrada em [64]. Um algoritmo mais eficiente foi dado recentemente por Shewchuk e Brown [123].

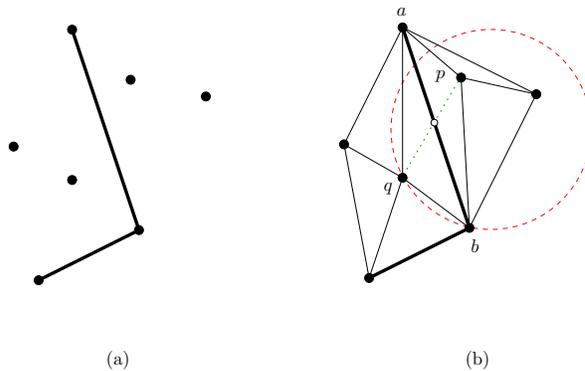
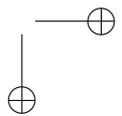
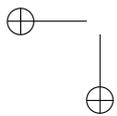
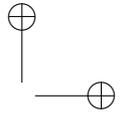
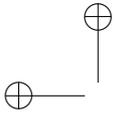


Figura 5.11: (a) Um GPSR  $\mathcal{G}$ . (b) A triangulação de Delaunay restrita,  $\mathcal{TD}(\mathcal{G})$ , de  $\mathcal{G}$ . Observe que a aresta  $[q, b]$  pertence a um círculo tal que os vértices de  $\mathcal{TD}(\mathcal{G})$  no interior deste círculo não podem ver  $p$ .

Uma vez que  $\mathcal{TD}(\mathcal{G})$  tenha sido construída, aplicamos o refinamento de Delaunay a  $\mathcal{TD}(\mathcal{G})$ , ignorando os triângulos e arestas que estão do lado de fora da região poligonal,  $\mathcal{R}$ , representada por  $\mathcal{TD}(\mathcal{G})$ .





Por ignorar, queremos dizer que não nos preocupamos em melhorar a qualidade dos triângulos que estão do lado de fora de  $\mathcal{R}$ . Além disso, modificamos o algoritmo de inserção de pontos para que ele nunca aplique uma troca de aresta a uma aresta de  $\mathcal{G}$  ou que pertença a um segmento de  $\mathcal{G}$  (ou seja, a aresta é um subsegmento). É importante ressaltar apenas que um segmento  $s$  é invadido por um dado ponto  $p$  se, e somente se,  $p$  pertence ao círculo diametral de  $s$  e houver um ponto  $q$  em  $s$  tal que o segmento  $\overline{qp}$  não interseque o interior de nenhum segmento de  $\mathcal{G}$ . No mais, o algoritmo funciona exatamente como antes. A triangulação resultante do refinamento não é necessariamente a triangulação de Delaunay do conjunto de vértices, mas ela é restrita a  $\mathcal{G}$  ou a um refinamento de  $\mathcal{G}$  e todos os triângulos dentro de  $\mathcal{R}$  serão de boa qualidade. Antes de devolver a triangulação, removemos todos os triângulos, arestas e vértices que não pertencem a  $\mathcal{R}$ .

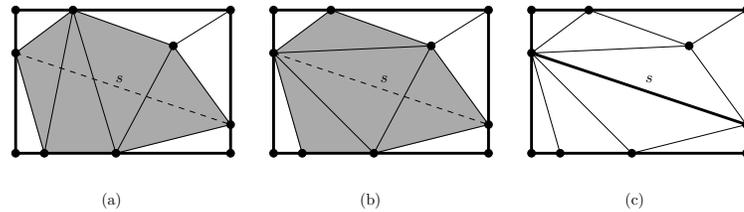
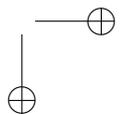
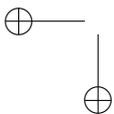
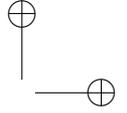
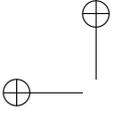


Figura 5.12: (a) O segmento  $s$  a ser inserido e a região de influência de  $s$  (cinza). (b) Todas as arestas que intersectam  $s$ , com exceção de uma, são trocadas. (c) A troca da última aresta insere  $s$  na triangulação.

## 5.4 Término

A condição de parada do Algoritmo 5.4 é  $E = \emptyset$  e  $R = \emptyset$ . Quando esta condição é satisfeita não haverá nenhum subsegmento invadido e nenhum triângulo magro na triangulação  $\mathcal{TD}(Q)$ . No entanto, se esta condição nunca for satisfeita, o laço **enquanto** das linhas 5 a 21 executará para sempre, eliminando e criando triângulos magros e





subsegmentos invadidos. Se conseguirmos provar que o algoritmo não pode criar triângulos magros e subsegmentos invadidos para sempre, então concluímos que ele termina. De fato, se  $\bar{\rho} > \sqrt{2}$  e nenhum par de subsegmentos de  $\mathcal{G}$  definir um ângulo menor do que  $90^\circ$ , podemos mostrar que o Algoritmo 5.4 termina. O argumento de nossa prova consiste em mostrar que o algoritmo mantém uma cota inferior para a distância entre quaisquer dois vértices de  $\mathcal{TD}(Q)$  e, portanto, poderemos invocar o Lema 5.2 para obter o resultado desejado. Uma noção chave para obtermos a cota inferior à qual nos referimos é uma função, denominada *lfs* (abreviação do inglês *local feature size*), dada a seguir:

**Definição 5.4.** A local feature size (*LFS*) de um GPSR  $\mathcal{G}$  é uma função,  $lfs : \mathbb{E}^2 \rightarrow \mathbb{R}$ , tal que para todo  $p \in \mathbb{E}^2$ , o valor de  $lfs(p)$  é igual ao raio do menor círculo centrado em  $p$  que intersecta dois elementos disjuntos (isto é, dois vértices, um vértice  $v$  e um segmento  $s$  tal que  $v \notin s$  ou dois segmentos,  $s_1$  e  $s_2$ , tais que  $s_1 \cap s_2 = \emptyset$ ) do GPSR  $\mathcal{G}$ .

A Figura 5.13 ilustra a definição da função *lfs*.

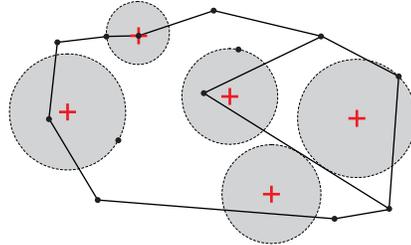
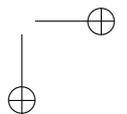
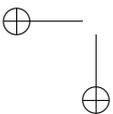
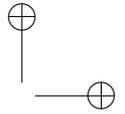
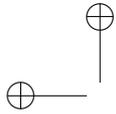


Figura 5.13: Em cada ponto,  $p$ , representado por uma cruz vermelha, o valor,  $lfs(p)$ , da função *lfs* em  $p$  é igual ao raio do menor círculo centrado no ponto que intersecta dois elementos disjuntos do GPSR associado.

**Definição 5.5.** Dizemos que uma função  $f : \mathbb{E}^d \rightarrow \mathbb{R}$  é Lipschitz contínua com constante de Lipschitz  $k_f$  em  $\mathbb{E}^d$  (ou simplesmente,  $k_f$ -Lipschitz) se existir uma constante,  $k_f$ , com  $k_f \in \mathbb{R}$  e  $k_f \geq 0$ , tal





que

$$|f(p) - f(q)| \leq k_f \cdot d(p, q),$$

para todo par de pontos,  $p$  e  $q$ , em  $\mathbb{E}^2$ , sendo  $d(p, q)$  a distância de  $p$  a  $q$ .

Como indicado pela notação, a constante  $k_f$  na Definição 5.5 depende da função,  $f$ , e pode variar de ser muito pequena para uma função a ser muito grande para outra função. Intuitivamente, se  $k_f$  é pequena, então  $f(x)$  deve variar muito pouco em uma vizinhança de  $x$  em  $\mathbb{E}^d$ . Por outro lado, se  $k_f$  é grande, então  $f(x)$  deve variar bastante em uma vizinhança de  $x$  em  $\mathbb{E}^d$ . De forma geral, se uma função  $g : \mathbb{E}^d \rightarrow \mathbb{R}$  é  $k_g$ -Lipschitz, então  $g$  é uma função contínua. E, se  $g$  também é diferenciável, então  $\|\nabla(g)\| \leq k_g$  (ver Problema 5.4). A função  $lfs$  goza da importante propriedade dada a seguir (ver Figura 5.14):

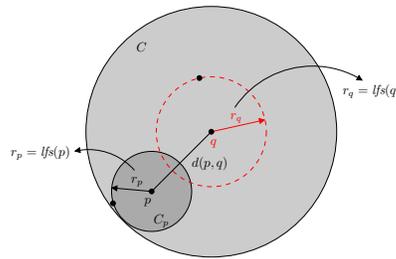
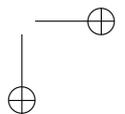
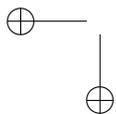
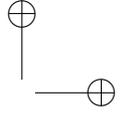
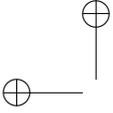


Figura 5.14: Ilustração do argumento da prova da Proposição 5.4.

**Proposição 5.4.** *A função  $lfs$  é 1-Lipschitz.*

*Demonstração.* Seja  $\mathcal{G}$  o GPSR associado a  $lfs$ . Sejam  $p$  e  $q$  dois pontos arbitrários de  $\mathbb{E}^2$ . Denote por  $C_p$  o círculo centrado em  $p$  de raio  $lfs(p)$ . Da Definição 5.4, temos que  $C_p$  intersecta dois elementos disjuntos de  $\mathcal{G}$ . Observe que  $C_p$  está contido no círculo  $C$  centrado em  $q$  e cujo raio é igual a  $lfs(p) + d(q, p)$ , já que  $p$  e  $q$  são dois vértices distintos de  $\mathcal{G}$  e, portanto, o raio de  $C_p$  é, no máximo, igual a  $d(p, q)$ . Daí, concluímos que  $lfs(q) \leq lfs(p) + d(q, p)$ . Usando o mesmo argumento com  $q$  substituído por  $p$ , podemos concluir que





$lfs(p) \leq lfs(q) + d(p, q)$ . Como a função distância é simétrica, temos que

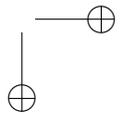
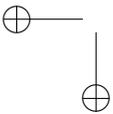
$$|lfs(p) - lfs(q)| \leq d(p, q).$$

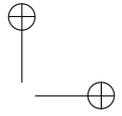
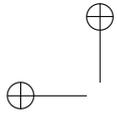
□

A cada ponto  $p \in \mathbb{E}^2$  inserido pelo Algoritmo 5.4 na triangulação  $\mathcal{TD}(Q)$ , definimos o *raio de inserção* de  $p$ , denotado por  $r_p$ , como sendo a distância,  $d(p, Q)$ , de  $p$  a um elemento de  $Q$  mais próximo de  $p$ . Observe que  $Q$  representa o conjunto de vértices de  $\mathcal{TD}(Q)$  imediatamente antes de  $p$  ser inserido em  $\mathcal{TD}(Q)$ . Observe também que, imediatamente após  $p$  ser inserido em  $\mathcal{TD}(Q)$ , o valor de  $r_p$  é o comprimento da aresta *mais curta* de  $\mathcal{TD}(Q \cup \{p\})$  incidente em  $p$ . Todo ponto  $p$  que o Algoritmo 5.3 rejeita (isto é, um circuncentro) por ele invadir um subsegmento também possui um raio de inserção, que também é definido como a distância de  $p$  para um elemento mais próximo de  $p$  em  $Q$  no exato momento em que  $p$  é considerado para inserção e rejeitado. Para mostrar que o Algoritmo 5.4 termina, provaremos que raios de inserção dos pontos inseridos e rejeitados não podem ser arbitrariamente menores do que os valores de  $lfs$  nos pontos.

**Definição 5.6.** *Seja  $p$  um ponto inserido ou rejeitado pelo Algoritmo 5.4 durante uma execução na entrada  $(\mathcal{G}, \tilde{p})$ . O ponto  $p$  é do tipo 0 se ele é um vértice do conjunto,  $\mathcal{V}$ , de vértices do GPSR  $\mathcal{G}$ . O ponto  $p$  é do tipo 1 se ele foi inserido em  $\mathcal{TD}(Q)$  no ponto médio de um subsegmento invadido pelo Algoritmo 5.2. O ponto  $p$  é do tipo 2 se  $p$  é o circuncentro de um triângulo e foi inserido em  $\mathcal{TD}(Q)$  pelo Algoritmo 5.3 ou rejeitado pelo Algoritmo 5.3 por invadir um subsegmento.*

Todo ponto inserido ou rejeitado pelo Algoritmo 5.4 possui um *pai*, exceto se o ponto for do tipo 0. Informalmente, o pai de um ponto  $p$ , do tipo 1 ou 2, é o ponto  $q$  responsável pela tentativa de inserção de  $p$  em  $\mathcal{TD}(Q)$ , independentemente da tentativa resultar em uma inserção ou rejeição. Se  $p$  é um vértice do tipo 1, então ele é sempre inserido em  $\mathcal{TD}(Q)$  pelo Algoritmo 5.2 como um ponto médio de algum subsegmento invadido,  $s$ . Há duas possibilidades para o pai de  $p$ : (1) ele é um vértice em  $\mathcal{V}$  (ver Figura 5.15(a)) ou





(2) ele é um circuncentro rejeitado (ver Figura 5.15(b)). No caso (1), definimos o pai,  $q$ , de  $p$  como sendo o ponto de  $\mathcal{V}$  mais próximo de  $p$  e, portanto, o raio de inserção,  $r_p$ , de  $p$  é igual a  $d(q, p)$ . No caso (2), o pai de  $p$  é o ponto invasor,  $q$ , correspondente ao circuncentro rejeitado e, portanto, o raio de inserção,  $r_p$ , de  $p$  é igual ao raio do círculo diametral de  $s$ , já que os pontos de  $Q$  mais próximos de  $p$  são os dois extremos de  $s$ . Como o ponto  $q$  pertence ao círculo diametral de  $s$ , temos

$$d(q, p) \leq r_p.$$

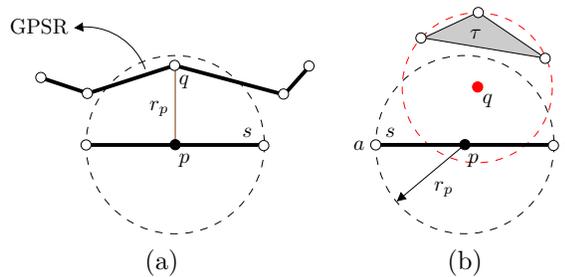
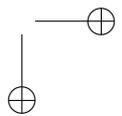
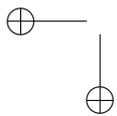


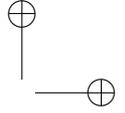
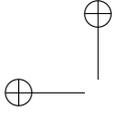
Figura 5.15: (a) O ponto  $q$  é um vértice em  $\mathcal{V}$ . (b) O ponto  $q$ , que é o circuncentro de um triângulo,  $\tau$ , foi rejeitado. Em ambos os casos, o ponto  $p$  é do tipo 1, o ponto  $q$  é o pai de  $p$  e  $q$  invade o subsegmento  $s$ .

Se  $p$  é um vértice do tipo 2, então  $p$  é o circuncentro de algum triângulo, digamos  $\tau$ . O pai,  $q$ , de  $p$  é definido como o vértice da aresta mais curta de  $\tau$  inserido em  $\mathcal{TD}(Q)$  por último (em ordem cronológica). Se os dois vértices da aresta mais curta de  $\tau$  pertencem a  $\mathcal{V}$ , então consideramos que eles foram inseridos em  $\mathcal{TD}(Q)$  ao mesmo tempo e escolhemos qualquer um deles para ser o pai,  $q$ , de  $p$  (ver Figura 5.16). Como  $\tau$  é um triângulo de Delaunay com relação a  $Q$ , temos que o círculo,  $C_\tau$ , circunscrito a  $\tau$  não pode possuir nenhum ponto de  $Q$  em seu interior. Logo, o raio de inserção,  $r_p$ , de  $p$  é igual a

$$r_p = d(q, p).$$

A proposição dada a seguir estabelece uma cota inferior para o





raio de inserção de cada ponto inserido ou rejeitado em função do raio do pai do ponto ou do valor da função  $lfs$  no ponto. Mais tarde, este fato nos permitirá estabelecer uma cota inferior para os comprimentos das arestas da malha final, que será eventualmente devolvida pelo Algoritmo 5.4.

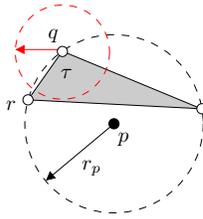
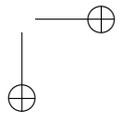
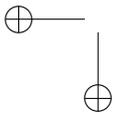


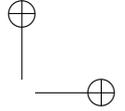
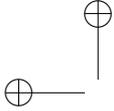
Figura 5.16: O ponto  $p$  é o circuncentro do triângulo  $\tau$  e pode ser inserido ou rejeitado. O pai de  $p$  é um dos dois vértices da aresta mais curta de  $\tau$ . Se os dois vértices estão em  $\mathcal{V}$ , o ponto  $q$  é qualquer um dos dois. Se não,  $q$  é o mais recente dos dois inserido em  $\mathcal{TD}(Q)$ .

**Proposição 5.5.** *Seja  $(\mathcal{G}, \bar{\rho})$  a entrada fornecida para uma execução do Algoritmo 5.4. Seja  $p$  um ponto inserido em  $Q$  ou um ponto rejeitado pelo Algoritmo 5.3. Seja  $q$  o pai de  $p$  (se existir). Então, temos que*

- (a) *se  $p$  é do tipo 0, então  $r_p \geq lfs(p)$ ;*
- (b) *se  $p$  é do tipo 1 e seu pai,  $q$ , é do tipo 0 ou 1, então  $r_p \geq lfs(p)$ ;*
- (c) *se  $p$  é do tipo 1 e seu pai,  $q$ , é do tipo 2, então  $r_p \geq r_q/\sqrt{2}$  e*
- (d) *se  $p$  é do tipo 2, então  $r_p > \bar{\rho} \cdot r_q$ .*

*Demonstração.* Considere o caso (a). Se  $p$  é do tipo 0, então, pela definição de  $lfs$ , o círculo,  $C_p$ , de centro em  $p$  e raio  $lfs(p)$  contém  $p$  em seu interior e intersecta outro elemento de  $\mathcal{G}$  (um vértice ou um segmento) em sua fronteira. Logo, temos que  $r_p \geq lfs(p)$ , pois  $r_p$  é a distância de  $p$  a um vértice mais próximo dele em  $\mathcal{V}$ . Considere o caso (d) e refira-se à Figura 5.16. Se  $p$  é do tipo 2, então  $p$  é o circuncentro





de um triângulo de Delaunay,  $\tau$ , no momento em que  $p$  é considerado para inserção. Mas, para  $p$  ser considerado para inserção, devemos ter que  $\rho(\tau) > \bar{\rho}$ . Por definição, o pai,  $q$ , de  $p$  é um dos vértices da aresta mais curta de  $\tau$ . Mais especificamente, qualquer um dos dois vértices, se ambos pertencem a  $\mathcal{V}$ , ou o vértice inserido mais recentemente em  $Q$  se um ou ambos não pertencem a  $\mathcal{V}$ . Seja  $r$  o outro vértice da aresta mais curta de  $\tau$ . Em ambos os casos, temos que  $r_q \leq d(r, q)$ . Sendo assim, o raio de inserção,  $r_p$ , de  $p$ , que é igual ao raio do círculo circunscrito a  $\tau$ , é  $r_p = d(p, q) = \rho(\tau) \cdot l > \bar{\rho} r_q$ , em que  $l = d(q, r)$  é o comprimento da aresta,  $[q, r]$ , mais curta de  $\tau$ . Nos casos (b) e (c), sabemos que  $p$  é o ponto médio de um subsegmento,  $s'$ , de um segmento  $s$  de  $\mathcal{G}$ . Além disso, o pai,  $q$ , de  $p$  é um invasor de  $s'$ . No caso (b), o ponto  $q$  é do tipo 0 ou 1. No caso (c), ele é do tipo 2. Suponha que  $q$  seja do tipo 0 e refira-se à Figura 5.17. Então, o círculo,  $C_p$ , de centro em  $p$  e raio  $lfs(p)$  intersecta, pela definição da função  $lfs$ , dois elementos disjuntos de  $\mathcal{G}$ ; a saber,  $q$  e  $s$ . Além disso, temos

$$lfs(p) \leq d(p, q) = r_p.$$

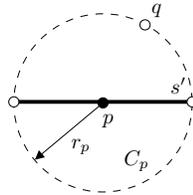
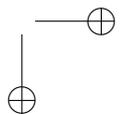
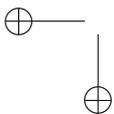
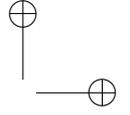
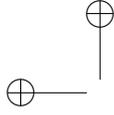


Figura 5.17: O ponto  $p$  é o ponto médio de um subsegmento,  $s'$ , de um segmento,  $s$ , de  $\mathcal{G}$  que foi invadido pelo pai,  $q$ , de  $p$  que é do tipo 0.

Suponha agora que  $q$  seja do tipo 1 e refira-se à Figura 5.18. Isto nos diz que  $q$  foi previamente inserido em outro subsegmento,  $t'$ , de um segmento,  $t$ , de  $\mathcal{G}$ . Os segmentos  $s$  e  $t$  de  $\mathcal{G}$  não podem compartilhar um vértice porque a presença de  $q$  no círculo diametral de  $s'$  implicaria que  $s$  e  $t$  definem um ângulo menor do que  $90^\circ$ , o que contradiz uma das duas restrições imposta à entrada  $(\mathcal{G}, \bar{\rho})$  do Algoritmo 5.4 (a outra é que  $\bar{\rho} > \sqrt{2}$ ). Então, o círculo,  $C_p$ , centrado





em  $p$  e de raio  $d(p, q)$  intersecta dois segmentos disjuntos,  $s$  e  $t$ , de  $\mathcal{G}$  e, portanto,

$$lfs(p) \leq d(p, q) = r_p.$$

Finalmente, considere o caso em que  $q$  é do tipo 2: o caso (c). Neste caso, o ponto  $q$  é um circuncentro rejeitado por invadir  $s'$ , o que nos permite concluir que ele reside no círculo diametral de  $s'$  cujo centro é  $p$ . Seja  $b$  o vértice de  $s'$  mais próximo de  $q$ . Sabemos que  $\angle qpb \leq \pi/2$  e que  $d(q, b) \leq \sqrt{2} \cdot d(p, b)$ , pois  $d(p, b)$  é o raio do círculo diametral de  $s'$ . Sabemos também que o Algoritmo 5.4 executa uma chamada ao Algoritmo 5.3 somente quando nenhum subsegmento foi invadido. Daí, podemos concluir que  $r_p = d(p, b)$  (isto é, os únicos pontos de  $Q$  no círculo diametral de  $s'$ , antes da inserção de  $p$ , são  $a$  e  $b$ ). Por sua vez, o pai,  $q$ , de  $p$  é o centro de um círculo circunscrito a um triângulo de Delaunay em relação a  $Q$  (isto é, um triângulo de  $\mathcal{TD}(Q)$ ). Como não pode haver nenhum ponto de  $Q$  no interior deste círculo, o ponto  $b$  não pode pertencer a ele (ver Figura 5.15(b)). Logo, concluímos que

$$r_q \leq d(q, b) \leq \sqrt{2} \cdot r_p \Rightarrow r_p \geq \frac{r_q}{\sqrt{2}}.$$

□

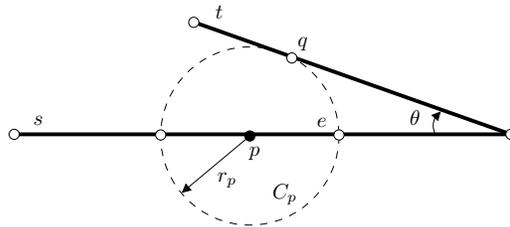
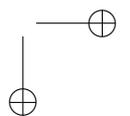
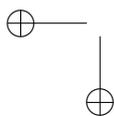
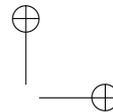
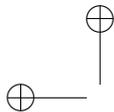


Figura 5.18: Os pontos  $p$  e  $q$  são do tipo 1 e  $q$  é o pai de  $p$ . Se os segmentos,  $s$  e  $t$ , de  $\mathcal{G}$  contendo  $p$  e  $q$ , respectivamente compartilhassem um vértice em  $\mathcal{V}$ , eles formariam um ângulo,  $\theta$ , tal que  $\theta$  é menor do que  $\pi/2$ .

O Lema 5.6 a seguir estabelece que os vértices inseridos em  $\mathcal{TD}(Q)$  pelo Algoritmo 5.4 são espaçados de maneira proporcional ao valor da





função  $lfs$  neles. Para provar o lema, fazemos uso da Proposição 5.5 acima.

**Lema 5.6.** *Seja  $(\mathcal{G}, \bar{\rho})$  a entrada fornecida para uma execução do Algoritmo 5.4. Seja  $p$  um ponto inserido em  $Q$  ou um ponto rejeitado pelo Algoritmo 5.3. Então, temos que*

$$(a) \text{ se } p \text{ é do tipo 1, então } r_p > \frac{lfs(p)}{C_S} \text{ e}$$

$$(b) \text{ se } p \text{ é do tipo 2, então } r_p > \frac{lfs(p)}{C_T},$$

em que

$$C_S = \frac{(\sqrt{2} + 1)\bar{\rho}}{\bar{\rho} - \sqrt{2}} \quad \text{e} \quad C_T = \frac{\bar{\rho} + 1}{\bar{\rho} - \sqrt{2}},$$

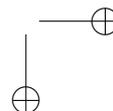
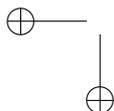
são constantes reais positivas.

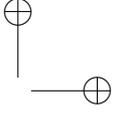
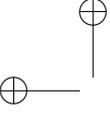
*Demonstração.* Como veremos durante a demonstração, as constantes  $C_S$  e  $C_T$  são soluções das equações  $C_S = \sqrt{2}C_T + 1$  e  $C_T = 1 + C_S/\bar{\rho}$ . Observe que  $C_S > C_T > 1$ . A prova procede por indução no comprimento da sequência de pontos inseridos em  $Q$  ou rejeitados pelo Algoritmo 5.4. O caso base corresponde ao primeiro ponto,  $p$ , considerado para inserção após a triangulação  $\mathcal{TD}(Q)$  ser construída, com  $Q = \mathcal{V}$ , na linha 2 de Algoritmo 5.4. Se  $p$  é do tipo 1, então o pai,  $q$  de  $p$  é um vértice de  $\mathcal{V}$  que invade o segmento,  $s$ , do qual  $p$  é o ponto médio. O segmento  $s$  não está em  $\mathcal{TD}(Q)$  e o Algoritmo 5.2 é chamado na linha 8 para subdividir o segmento  $s$  em dois segmentos de mesmo tamanho e que compartilham  $p$ . Pela Proposição 5.5, temos que

$$r_p \geq lfs(p) > \frac{lfs(p)}{\lambda},$$

para toda constante  $\lambda \in \mathbb{R}$ , com  $\lambda > 1$ . Se  $p$  é do tipo 2, então  $p$  é o circuncentro de um triângulo magro. Além disso, nenhum segmento de  $\mathcal{TD}(Q)$  é invadido por um vértice de  $Q$  antes de  $p$  ser considerado para inserção. Caso contrário, o ponto  $p$  não seria o primeiro ponto a ser considerado para inserção, mas sim o ponto médio de um segmento invadido. Logo, o pai,  $q$ , de  $p$  é um vértice de  $\mathcal{V}$  mais próximo de  $p$ . Daí,

$$r_p > \frac{lfs(p)}{\xi},$$





para toda constante  $\xi \in \mathbb{R}$ , com  $\xi > 1$ , pois  $r_p = d(p, q) = lfs(p)$ . Agora, suponha que a afirmação é válida para os  $h$  primeiros vértices inseridos ou rejeitados pelo Algoritmo 5.4, após a triangulação inicial,  $\mathcal{TD}(\mathcal{V})$ , ter sido construída. Seja  $p$  o  $(h+1)$ -ésimo ponto considerado para inserção. Suponha que  $p$  seja do tipo 1. Seja  $q$  o pai de  $p$ . Se o tipo de  $q$  for 0 ou 1, então temos que  $r_p \geq lfs(p)$ , pela Proposição 5.5. Daí,

$$r_p > \frac{lfs(p)}{\lambda},$$

para toda constante  $\lambda \in \mathbb{R}$ , com  $\lambda > 1$ . Mas, se  $q$  é do tipo 2, então a Proposição 5.5 nos diz que  $r_p \geq r_q/\sqrt{2}$ , com  $r_p \geq d(p, q)$ . Usando a hipótese de indução, temos que  $r_q > lfs(q)/C_T$ . Pela Proposição 5.4, temos

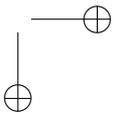
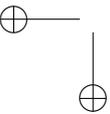
$$lfs(p) \leq lfs(q) + d(p, q) < r_q \cdot C_T + r_p \leq (\sqrt{2} \cdot C_T + 1) \cdot r_p. \quad (5.4)$$

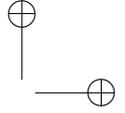
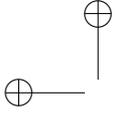
Se  $p$  é do tipo 2, então, pela hipótese de indução, o pai,  $q$ , de  $p$  é tal que  $r_q > lfs(p)/C_S$ . Pela Proposição 5.5, temos que  $r_p > \bar{\rho} \cdot r_q$ . Por sua vez, pela Proposição 5.4,  $lfs(p) \leq lfs(q) + d(p, q) = lfs(q) + r_p$ . Daí, temos

$$r_p > \bar{\rho} \cdot r_q > \frac{\bar{\rho}}{C_S} \cdot lfs(p) \geq \frac{\bar{\rho}}{C_S} \cdot (lfs(p) - r_p) \Rightarrow r_p > \frac{lfs(p)}{1 + \frac{C_S}{\bar{\rho}}}. \quad (5.5)$$

Finalmente, observe que se tomarmos  $C_S = \sqrt{2} \cdot C_T + 1$  e  $C_T = 1 + C_S/\bar{\rho}$ , a Equação (5.4) satisfará a afirmação (a) e a Equação (5.5) satisfará a afirmação (b). É fácil verificar que a solução das equações para  $C_S$  e  $C_T$  são aquelas dada no enunciado do lema e que  $C_S > C_T > 1$ .  $\square$

O Lema 5.6 nos oferece uma cota inferior para a distância entre um ponto  $p$  recém inserido na triangulação  $\mathcal{TD}(Q)$  corrente e todos os pontos inseridos antes dele. Para mostrar que o Algoritmo 5.4 termina, precisamos de uma cota inferior que relacione a distância entre  $p$  e todos os pontos inseridos pelo algoritmo, incluindo aqueles que são inseridos após a inserção de  $p$ . O fato da função  $lfs$  ser 1-Lipschitz nos permite obter este resultado, como mostra o lema a seguir:





**Lema 5.7.** *Seja  $(\mathcal{G}, \bar{\rho})$  a entrada fornecida para uma execução do Algoritmo 5.4. Sejam  $p$  e  $q$  quaisquer dois pontos inseridos em  $Q$  pelo Algoritmo 5.3. Então, temos que*

$$d(p, q) \geq \frac{lfs(p)}{C_S + 1}.$$

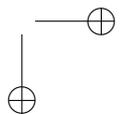
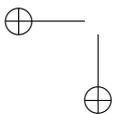
*Demonstração.* Se  $p$  é inserido em  $Q$  depois de  $q$ , então, o Lema 5.6 nos diz que  $d(p, q) \geq lfs(p)/C_S$ . Caso contrário, temos que  $d(q, p) \geq lfs(q)/C_S$ . Como a função  $lfs$  é 1-Lipschitz, temos que  $lfs(p) \leq lfs(q) + d(p, q) \leq d(q, p) \cdot C_S + d(p, q) = (1 + C_S) \cdot d(p, q)$ , pois  $d(q, p) = d(p, q)$ .  $\square$

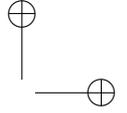
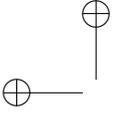
Enfim, podemos estabelecer o resultado principal:

**Teorema 5.8.** *Seja  $(\mathcal{G}, \bar{\rho})$  a entrada fornecida para uma execução do Algoritmo 5.4. Então, a execução do Algoritmo 5.4 na entrada  $(\mathcal{G}, \bar{\rho})$  termina e devolve uma triangulação de Delaunay conforme de  $\mathcal{G}$  na qual todos os triângulos possuem razão circunraio-aresta menor ou igual a  $\bar{\rho}$ .*

*Demonstração.* Seja  $lfs_{\min} = \min_{p \in FC(\mathcal{V})} lfs(p)$ . Como  $\mathcal{G}$  possui um número finito de elementos e quaisquer dois elementos disjuntos de  $\mathcal{G}$  estão separados por uma distância positiva, temos que  $lfs_{\min} > 0$ . Pelo Lema 5.7, quaisquer dois pontos inseridos por Algoritmo 5.4 estão separados por uma distância de, no mínimo,  $lfs_{\min}/C_S + 1 > 0$ . Pelo Lema 5.2, o Algoritmo 5.4 não pode inserir um número infinito de pontos no conjunto de pontos  $Q$ . Mas, se o processo de inserção de pontos é finito, então o algoritmo termina. Por outro lado, a inserção de pontos só é cessada quando todos os segmentos de  $\mathcal{G}$  estão representados por arestas ou união de arestas da triangulação final e nenhum triângulo da triangulação final é magro. Caso contrário, o algoritmo continuaria a inserir pontos. Logo, a malha devolvida por Algoritmo 5.4 é uma triangulação de Delaunay conforme de  $\mathcal{G}$  na qual todos os triângulos possuem razão circunraio-aresta menor ou igual a  $\bar{\rho}$ .  $\square$

É importante ressaltar que a conclusão do Teorema 5.8 pressupõe que a entrada,  $(\mathcal{G}, \bar{\rho})$ , satisfaz as restrições que impusemos anteriormente:  $\bar{\rho} > \sqrt{2}$  e nenhum par de segmentos adjacentes de  $\mathcal{G}$  pode





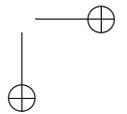
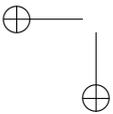
definir um agudo (isto é, menor do que  $90^\circ$ ) no vértice comum aos dois. No entanto, a cota inferior de  $\sqrt{2}$  parece ser bastante pessimista na prática, pois, em geral, é quase sempre possível executar o Algoritmo 5.4 e obter uma malha com um ângulo mínimo de  $33^\circ$  [118]. Além disso, certas modificações na escolha da posição do ponto a ser inserido<sup>2</sup> melhoram ainda mais o algoritmo e sugerem que uma cota inferior para o ângulo mínimo de, pelo menos,  $40^\circ$  possa ser obtida [60].

Por outro lado, Steven Pav forneceu um exemplo bastante simples que mostra que o algoritmo de Ruppert (original), isto é, o Algoritmo 5.4, não termina para qualquer escolha de um ângulo mínimo  $\kappa$ , com  $\kappa > 30^\circ$  [101]. Suponha que o GPSR  $\mathcal{G}$  dado como entrada para o Algoritmo 5.4 possua dois segmentos adjacentes,  $s_1$  e  $s_2$ , tais como mostrado na Figura 5.19, os quais definem um ângulo de  $7\pi/12 - \epsilon$ , em que  $\epsilon$  é uma pequena constante positiva. Suponha também que a razão entre os comprimentos de  $s_1$  e  $s_2$  seja  $\sqrt{2}$ . Denote por  $\tau$  o triângulo formado pelos três pontos,  $a$ ,  $b$  e  $c$ , de  $Q$  que são vértices de  $s_1$  e  $s_2$ . O ângulo oposto a  $s_2$  é igual a  $\pi/6 + \mathcal{O}(\epsilon)$  e o terceiro,  $\pi/4 + \mathcal{O}(\epsilon)$ .

Se especificarmos um ângulo mínimo maior do que o ângulo oposto a  $s_2$  em  $\tau$ , então o algoritmo tentará remover  $\tau$  através da inserção de seu circuncentro,  $c_1$ . Mas, o ponto  $c_1$  pertence ao círculo diametral de  $s_1$ . Consequentemente,  $c_1$  é rejeitado e o segmento  $s_1$  é subdividido e, na triangulação resultante, o triângulo  $[m_1, b, c]$  aparece, em que  $m_1$  é o ponto médio de  $s_1$ . Este triângulo é semelhante ao triângulo  $\tau$ . Logo, o algoritmo tenta eliminá-lo com a inserção de seu circuncentro, digamos  $c_2$ . Mas, desta vez,  $c_2$  pertence ao círculo diametral de  $s_2$  e, portanto,  $c_2$  é rejeitado e o segmento  $s_2$  é subdividido. Na triangulação resultante, o triângulo  $[m_1, b, m_2]$  aparece, em que  $m_2$  é o ponto médio de  $s_2$ . Este triângulo também é semelhante a  $\tau$ ; em particular, cada lado corresponde a metade de um lado distinto de  $\tau$ . Logo, este ciclo de refinamento de subsegmentos continua para sempre.

Todos os resultados que obtivemos até então dependem fortemente da restrição sobre o ângulo definido por dois segmentos ad-

<sup>2</sup>Por exemplo, inserir um ponto na vizinhança do circuncentro em vez do próprio.



jaçentes de  $\mathcal{G}$ . Infelizmente, restringir tais segmentos a formarem um ângulo maior ou igual a  $\pi/2$  limita a aplicação do algoritmo de Ruppert na prática. Na Seção 5.6, veremos que esta restrição pode ser relaxada com pequenos ajustes nos principais resultados obtidos até então.

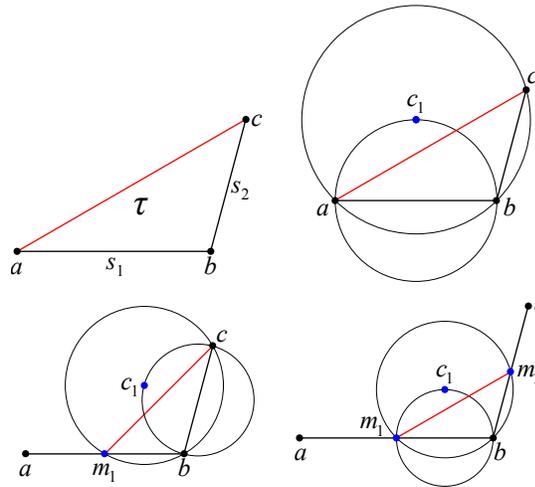
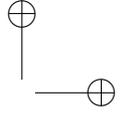
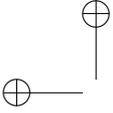


Figura 5.19: Um exemplo em que o algoritmo de Ruppert não termina se  $\bar{\rho} \approx 0,999913$ , pois ele insere uma sequência infinita de pontos do tipo 1 nos segmentos  $s_1$  e  $s_2$  [101]. Cada ponto inserido tem como pai o circuncentro de um triângulo cujo menor ângulo é  $\pi/6 + 0,0005$ . Todos os triângulos mostrados são semelhantes. O menor ângulo definido pelos segmentos é igual a  $7\pi/12 - 0,001$  e, portanto, é maior do que  $\pi/2$ .

## 5.5 Otimalidade

O algoritmo de Ruppert gera triangulação de Delaunay que, além de satisfazerem as restrições de conformidade e de ângulo mínimo, também satisfazem duas outras propriedades importantes, a saber:



*tamanho ótimo e adaptatividade.* Antes de provar tal afirmação, precisamos definir formalmente o que queremos dizer por cada um desses termos.

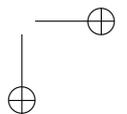
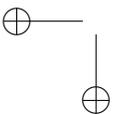
Seja  $\mathcal{C}$  a classe de todos os GPSR que satisfazem algum critério. Para todo  $\mathcal{G} \in \mathcal{C}$ , seja  $\mathcal{T}(\mathcal{G}, \bar{\rho})$  a triangulação que possui o menor número de triângulos entre todas as triangulações restritas de  $\mathcal{G}$  ou de algum refinamento de  $\mathcal{G}$  e tais que a razão circuncentro-raio de qualquer triângulo não excede  $\bar{\rho}$ . Seja  $\mathcal{M}(\mathcal{G}, \bar{\rho})$  a triangulação restrita de  $\mathcal{G}$  ou de algum refinamento de  $\mathcal{G}$  gerada por algum algoritmo que garante que nenhum triângulo de  $\mathcal{M}(\mathcal{G}, \bar{\rho})$  possui razão circuncentro-raio maior do que  $\bar{\rho}$ . As triangulações geradas pelo algoritmo possuem *tamanho ótimo* se, e somente se, para cada  $\mathcal{G} \in \mathcal{C}$ , o número de triângulos de  $\mathcal{M}(\mathcal{G}, \bar{\rho})$  é, no máximo, igual a  $\alpha$  vezes o número de triângulos de  $\mathcal{T}(\mathcal{G}, \bar{\rho})$ , em que  $\alpha$  é uma constante que depende apenas de  $\mathcal{G}$ .

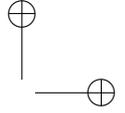
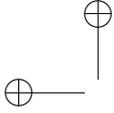
O algoritmo de Ruppert gera malhas de tamanho ótimo quando aplicado a classe  $\mathcal{C}$  de todos os GPSR que representam uma região poligonal convexa e que não possuem nenhum par de segmentos que se intersectam em um ângulo menor do que  $90^\circ$ . Além disso, a garantia é dada para entradas com  $\bar{\rho} > \sqrt{2}$ , assim como antes. Para provar a otimalidade de tamanho, Ruppert se valeu da seguinte observação [113]: o valor da função *lfs* em um ponto  $x$  qualquer do espaço subjacente da malha mede, aproximadamente, o maior tamanho possível de um triângulo contendo  $x$  em uma triangulação em que todos os triângulos são quase equiláteros. Sendo assim, ele concluiu que uma região em que o valor da função *lfs* é pequeno requer triângulos de tamanho pequeno. Logo, a distribuição de vértices deveria ser proporcional ao valor da função *lfs*, ou seja, a densidade de triângulos da malha é inversamente proporcional ao quadrado do valor de *lfs*.

**Lema 5.9.** *Seja  $\mathcal{T}$  a triangulação de Delaunay gerada pelo Algoritmo 5.4 em uma entrada  $(\mathcal{G}, \bar{\rho})$ . Então, o número de triângulos em  $\mathcal{T}$  é menor do que*

$$\frac{8(3 + 2C_S)^2}{\pi} \cdot \int_{|\mathcal{T}|} \frac{dx}{lfs(x)^2},$$

em que  $C_S$  é uma constante que depende apenas de  $\bar{\rho}$  (ver a definição





no Lema 5.6) e  $dx$  representa uma medida infinitesimal de área em  $\mathbb{E}^2$ .

*Demonstração.* Seja  $Q$  o conjunto de vértices de  $\mathcal{T}$  e denote por  $|Q|$  a cardinalidade de  $Q$ . Para cada vértice  $v \in Q$ , considere círculo,  $C_v$ , de centro  $v$  e raio  $r_v$  igual a  $lfs(v)/(2 + 2C_S)$ . Pelo Lema 5.7, os interiores dos círculos associados aos vértices de  $Q$  são disjuntos. Como dois segmentos em  $\mathcal{G}$  não podem definir um ângulo agudo em um vértice comum a ambos, pelo menos um quarto de cada círculo está incluído em  $|\mathcal{T}|$ . Daí,

$$\begin{aligned} \int_{|\mathcal{T}|} \frac{dx}{lfs(x)^2} &> \sum_{v \in Q} \int_{C_v \cap |\mathcal{T}|} \frac{dx}{lfs(x)^2} \\ &> \sum_{v \in Q} \int_{C_v \cap |\mathcal{T}|} \frac{dx}{(lfs(v) + r_v)^2} \\ &\geq \frac{1}{4} \sum_{v \in Q} \int_{C_v} \frac{dx}{(lfs(v) + r_v)^2} \\ &= \frac{1}{4} \sum_{v \in Q} \frac{\pi r_v^2}{(lfs(v) + r_v)^2}. \end{aligned}$$

Observe que

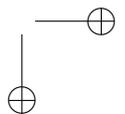
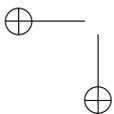
$$\begin{aligned} (lfs(v) + r_v)^2 &= lfs(v)^2 + 2lfs(v)r_v + r_v^2, \\ \frac{\pi r_v^2}{(lfs(v) + r_v)^2} &= \frac{\pi}{\frac{(lfs(v) + r_v)^2}{r_v^2}} \end{aligned}$$

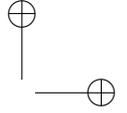
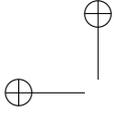
e

$$\frac{lfs(v)}{r_v} = 2C_S + 2.$$

Logo,

$$\begin{aligned} \frac{(lfs(v) + r_v)^2}{r_v^2} &= (2C_S + 2)^2 + 2 \cdot (2C_S + 2) + 1 \\ &= 4C_S^2 + 8C_S + 4 + 4C_S + 4 + 1 \\ &= 4C_S^2 + 12C_S + 9 \\ &\leq (2C_S + 3)^2. \end{aligned}$$





Daí,

$$\frac{1}{4} \sum_{v \in Q} \frac{\pi r_v^2}{(lfs(v) + r_v)^2} \geq \frac{1}{4} \sum_{v \in Q} \frac{\pi}{(2C_S + 3)^2} = \frac{\pi}{4(2C_S + 3)^2} \cdot |Q|.$$

Como uma triangulação  $\mathcal{T}'$  com  $|Q|$  possui  $2|Q| - h - 2$  triângulos, em que  $h$  é o número de vértices na fronteira,  $\partial(\mathcal{T}')$ , do espaço subjacente,  $|\mathcal{T}'|$  (ver Problema 4.15), de  $\mathcal{T}'$ , podemos concluir que  $\mathcal{T}$  tem, no máximo,  $2|Q| - 5$  triângulos e, portanto, a nossa afirmação é válida.  $\square$

Seja  $\mathcal{T}$  qualquer triangulação restrita de um GPSR  $\mathcal{G}$  ou de um refinamento de  $\mathcal{G}$  tal que  $|\mathcal{T}|$  é convexo. Ruppert mostrou que, para qualquer triângulo  $\tau$  de  $\mathcal{T}$  e para qualquer ponto  $p$  no interior de  $\tau$ , o valor,  $\ell_{\max}(p)$  da função  $\ell_{\max}$  em  $p$  é menor ou igual a  $\eta \cdot lfs(p)$ , em que  $\ell_{\max}$  é a função que associa a  $p$  o comprimento da aresta mais longa do triângulo,  $\tau$ , que o contém e  $\eta$  é uma constante positiva que depende apenas do valor de  $\phi$ , com  $\phi = \max_{\sigma \in \mathcal{T}} \rho(\sigma)$  [113]. Com base neste fato, ele mostrou que o seu algoritmo gera malhas de tamanho ótimo.

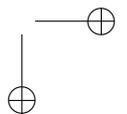
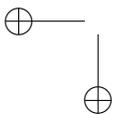
**Lema 5.10.** *Seja  $\mathcal{G}$  um GPSR tal que a fronteira do fecho convexo dos vértices de  $\mathcal{G}$  está contida na união dos segmentos de  $\mathcal{G}$ . Seja  $\mathcal{T}$  qualquer triangulação restrita de  $\mathcal{G}$  ou de um refinamento de  $\mathcal{G}$ . Seja  $\phi = \max_{\sigma \in \mathcal{T}} \rho(\sigma)$ . Então, o número de triângulos de  $\mathcal{T}$  é, pelo menos,*

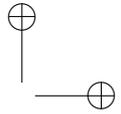
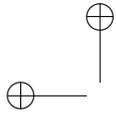
$$\frac{4}{\sqrt{3}\eta^2} \cdot \int_{|\mathcal{T}|} \frac{dx}{lfs(x)^2},$$

em que  $\eta$  é uma constante real positiva que depende apenas de  $\phi$ .

*Demonstração.* Seja  $\ell_{\max}$  a função que mapeia cada ponto  $p$  no interior de um triângulo de  $\mathcal{T}$  para o comprimento da aresta mais longa do triângulo de  $\mathcal{T}$  que contém  $p$ . Como dito anteriormente, Ruppert mostrou que  $\ell_{\max}(p) \leq \eta \cdot lfs(p)$ , em que  $\eta$  é uma constante real positiva que depende apenas de  $\phi$ . Logo, podemos concluir que

$$\int_{|\mathcal{T}|} \frac{dx}{lfs(x)^2} \leq \eta^2 \cdot \int_{|\mathcal{T}|} \frac{dx}{\ell_{\max}(x)^2} = \eta^2 \sum_{\tau \in \mathcal{T}} \left( \int_{\tau} \frac{dx}{\ell_{\max}(x)^2} \right).$$





Observe que  $\ell_{\max}$  não é definida nas arestas de um triângulo, mas como as arestas possuem medida nula, elas não contribuem para a integral. Além disso, a área de um triângulo  $\tau$  cuja aresta mais longa é, no máximo,  $\sqrt{3}/4 \cdot \ell_\tau^2$ , em que  $\ell_\tau$  é o comprimento da aresta mais longa de  $\tau$ . Este limite superior é atingido quando  $\tau$  é equilátero. Daí,

$$\sum_{\tau \in \mathcal{T}} \left( \int_{\tau} \frac{dx}{\ell_{\max}(x)^2} \right) = \eta^2 \sum_{\tau \in \mathcal{T}} \frac{\text{área}(\tau)}{\ell_\tau^2} \leq \frac{\sqrt{3}}{4} \eta^2 \sum_{\tau \in \mathcal{T}} \frac{\ell_\tau^2}{\ell_\tau^2} \leq \frac{\sqrt{3}}{4} \eta^2 \sum_{\tau \in \mathcal{T}} 1,$$

o que mostra que o número de triângulos de  $\mathcal{T}$  é, pelo menos, igual a

$$\frac{4}{\sqrt{3}\eta^2} \cdot \int_{|\mathcal{T}|} \frac{dx}{\ell_{\max}(x)^2}.$$

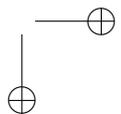
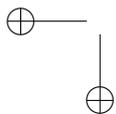
□

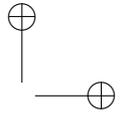
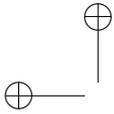
Em [23], os autores mostram que a constante  $\eta$  no Lema 5.10 é igual a

$$4\phi(4 - 1/\phi^2)^{0,5+90^\circ/\arcsen(1/2\phi)}.$$

O valor acima é menor do que aquele apresentado originalmente por Ruppert [113]. Em [23] também se discute como remover a restrição de  $|\mathcal{T}|$  ter de ser um conjunto convexo. Esta restrição se deve ao fato de que se o domínio representado por  $\mathcal{G}$  não é convexo, então o algoritmo de Ruppert pode não gerar malhas de tamanho ótimo. No entanto, se usarmos a triangulação de Delaunay restrita como triangulação inicial (ver discussão na Subseção 5.3.5), podemos mostrar que as malhas geradas pelo algoritmo modificado têm tamanho ótimo [23].

Para finalizar nossa discussão sobre os aspectos de otimalidade do algoritmo de Ruppert, vamos considerar o Lema 5.7 novamente. Este lema nos diz, de maneira indireta, que o comprimento da menor aresta de um triângulo qualquer,  $\tau$ , de uma malha gerada pelo algoritmo de Ruppert é igual a uma constante vezes o maior comprimento possível de uma aresta na região coberta pelo triângulo. Por outro lado, pelo que vimos acima, o comprimento da aresta mais longa de  $\tau$  triângulo também é igual a uma constante vezes o maior comprimento possível de uma aresta na região coberta pelo triângulo. Isto nos diz





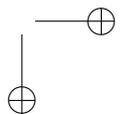
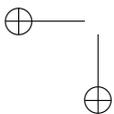
que a malha gerada pelo algoritmo de Ruppert é “bem adaptada” às características geométricas do domínio correspondente ao espaço subjacente da malha. O termo *transição espacial ótima* (tradução livre do inglês *optimal grading*) é algumas vezes usado para qualificar algoritmos que geram malhas bem adaptadas no sentido explicado acima.

## 5.6 Ângulos pequenos

Dependendo do domínio de entrada, o algoritmo de Ruppert (Algoritmo 5.4) pode não funcionar de forma correta ou até mesmo gerar uma malha de má qualidade. Dado um GPSR  $\mathcal{P}$  em  $\mathbb{E}^2$ , a fim de garantir o bom funcionamento do algoritmo, uma restrição severa a ser imposta é a que os segmentos adjacentes de  $\mathcal{P}$  não formem ângulos agudos. Na prática, o algoritmo funciona mesmo na presença de ângulos agudos, porém para ângulos menores que  $45^\circ$ , maior será o risco de ocorrer uma falha no término do algoritmo. Nessa seção discutiremos duas modificações no algoritmo de refinamento para que ele possa tratar domínios com ângulos pequenos.

Um caso patológico ocorre quando dois segmentos adjacentes de tamanhos distintos formando um ângulo pequeno produzem um ciclo vicioso no qual um segmento é invadido por um ponto inserido no ponto médio do outro segmento gerando subsegmentos cada vez menores, conforme ilustrado na Figura 5.20. Essa recursão infinita é conhecida como *invasão ping-pong*.

Em outros casos, gerar um triângulo de boa qualidade é uma tarefa impossível. Por exemplo, se dois segmentos adjacentes de  $\mathcal{P}$  formarem um ângulo de  $3^\circ$ , então a malha resultante vai possuir pelo menos um triângulo com um ângulo menor ou igual a  $3^\circ$ . Além disso, pode ocorrer uma situação em que um ângulo pequeno de  $\mathcal{P}$  faça surgir novos ângulos pequenos não herdados de  $\mathcal{P}$ . Logo, dado  $\mathcal{P}$  com ângulos pequenos, um método de geração de malhas deve saber quando é necessário aceitar um triângulo de má qualidade. Uma modificação consiste em alterar a forma que o algoritmo divide os segmentos invadidos e a outra se baseia na rejeição de triângulos magros durante o refinamento. Combinadas, essas modificações produzem uma variante do algoritmo original de Ruppert que além de garantia



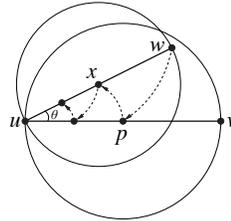
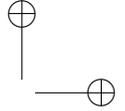
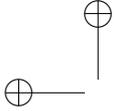


Figura 5.20: Invasão ping-pong causada por um ângulo pequeno  $\theta$ : o segmento  $\overline{uw}$  é invadido por  $w$ , o qual é dividido em  $p$ . O segmento  $\overline{uw}$  é invadido por  $p$ , o qual é dividido em  $x$ , assim por diante.

de término possui algumas propriedades interessantes. Vale a pena ressaltar que o algoritmo fornece a garantia que nenhum triângulo com ângulo maior que  $138.6^\circ$  será gerado e que triângulos magros surgirão somente entre segmentos adjacentes que possuam um ângulo pequeno. Em aplicações usando MEF, majorar o ângulo máximo e mais importante que o ângulo mínimo, pois ele é utilizado no cálculo da estimativa de erro de uma aproximação via MEF.

A primeira modificação é dividir segmentos invadidos em um ponto que não seja o ponto médio. Imagine que cada vértice do GPSR de entrada é o centro de vários círculos concêntricos cujos raios são potências  $2^i$ , para todo  $i \in \mathbb{Z}$  (Figura 5.21). Se um vértice de um segmento invadir um segmento adjacente e eles formarem um ângulo agudo, o segmento invadido ao invés de ser dividido no seu ponto médio, é dividido no ponto de intersecção com um dos círculos concêntricos centrados no vértice comum aos dois segmentos, logo um dos novos segmentos criados tem comprimento de uma potência de dois. O novo vértice criado dessa divisão é chamado de *vértice fora do centro*. A escolha do círculo é baseada naquele que fornece melhor divisão no sentido que os comprimentos dos novos subsegmentos estejam balanceados, isto é, os comprimentos dos subsegmentos estejam entre  $1/3$  e  $2/3$  do comprimento do segmento original. Quando ambos os vértices de um segmento invadirem outros segmentos adjacentes, basta escolher o vértice que fornece o melhor balanceamento na divisão. Vamos chamar de *divisão não centrada* quando um segmento



não é dividido ao meio.

A estratégia de usar círculos concêntricos evita que uma invasão ping-pong seja estabelecida entre dois segmentos adjacentes que se encontram em um ângulo agudo. Entretanto, uma outra situação deve ser considerada. Antes vamos relembrar que uma das motivações de se usar o refinamento de Delaunay era que nenhuma aresta de comprimento menor que a menor aresta existente na malha seria criada. Imagine o caso em que dois segmentos adjacentes formando um ângulo pequeno em que um dos segmentos seja dividido ao meio devido a uma invasão ocasionada pelo ponto médio do outro segmento, a nova aresta que conecta os dois pontos médios pode violar essa regra. Essa pequena aresta criada pode causar um refinamento recursivo sem fim, devido ao fato do algoritmo sempre tentar eliminar triângulos magros, fazendo que os subsegmentos gerados sejam divididos novamente criando arestas cada vez menores. Uma forma de quebrar esse ciclo sem fim é evitar que arestas pequenas sejam geradas.

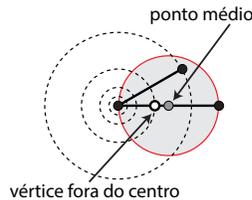
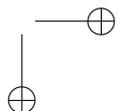
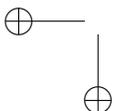


Figura 5.21: Se um segmento é invadido por um segmento adjacente formando um ângulo agudo, ele é dividido em um ponto fora do centro, ponto onde ocorre a sua interseção com um círculo concêntrico de raio  $2^i$  para algum  $i \in \mathbb{Z}$ .

**Definição 5.7.** *Uma aresta é dita insurgente quando cada um de seus vértices estiverem em segmentos distintos que se unem formando um ângulo menor que  $60^\circ$ .*

A Figura 5.22 (a) exhibe um exemplo de uma aresta insurgente. Os vértices de uma aresta insurgente estão localizados no mesmo



círculo concêntrico, pois surgem da divisão no ponto médio de segmentos adjacentes.

A segunda modificação é simplesmente desistir de tentar dividir um triângulo magro cuja menor aresta é insurgente. Essa precaução previne que o comprimento pequeno de aresta insurgentes se propague ao longo da malha. Triângulos com ângulos pequenos podem sobreviver a etapa de refinamento, mas somente em segmentos adjacentes que possuam um ângulo pequeno. Portanto, o algoritmo requer ambas as modificações para evitar que o refinamento seja feito eternamente. Figura 5.22 (b) mostra uma malha gerada com o algoritmo de refinamento utilizando as duas modificações discutidas nessa seção.

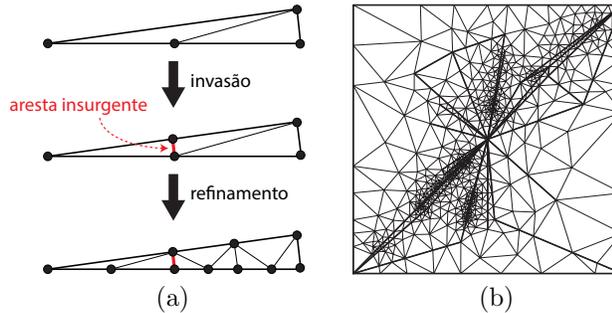
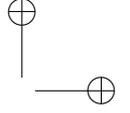
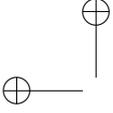


Figura 5.22: Arestas insurgentes surgem em segmentos adjacentes separados por um ângulo pequeno (esquerda). O refinamento de triângulos magros podem gerar novos segmentos fazendo com que eles sejam divididos novamente. O algoritmo de refinamento modificado em ação (direita).

O término do algoritmo modificado se deve ao fato de que *arestas indevidamente pequenas*, isto é, arestas menores que as admitidas pela Proposição 5.11, podem ser criadas em apenas duas circunstâncias: através de subsegmentos gerados por divisões não centradas ou através de uma recursão de bisseções entre segmentos adjacentes, como mostrado na Figura 5.22 (a). Porém, essas arestas são todas insurgentes, logo serão rejeitadas pelo algoritmo.

**Proposição 5.11.** *Seja  $\bar{\rho} > \sqrt{2}$  a razão circunraio-aresta máxima*



permitida de um triângulo cuja menor aresta não é insurgente. Dado um GPSR  $\mathcal{G}$  em  $\mathbb{E}^2$  sem restrição alguma sobre os ângulos formados por seus segmentos. O algoritmo de Ruppert modificado, usando círculos concêntricos na divisão de segmentos e que evita dividir triângulos em que a menor aresta é insurgente, possui garantia de término. Além disso, os ângulos dos triângulos da malha final estarão limitados por

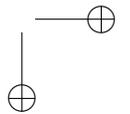
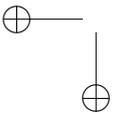
$$\frac{\text{sen}(\phi_{\min})}{\sqrt{5 - 4 \cos(\phi_{\min})}} \leq \phi \leq 180^\circ - 2 \arcsen\left(\frac{1}{2\rho}\right),$$

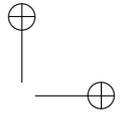
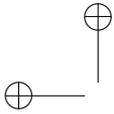
em que  $\phi_{\min}$  é o menor ângulo formado entre dois segmentos de  $\mathcal{G}$ .

*Demonstração.* Seja  $\mathcal{G}'$  uma cópia modificada de  $\mathcal{G}$  por incluir todos vértices fora do centro inseridos na fase de divisão de segmento do algoritmo. Pelo menos duas divisões não centradas ocorrem em cada segmento de  $\mathcal{P}$ , logo  $\mathcal{P}'$  possui uma quantidade finita de vértices extras. Sejam  $f = lfs|_{\mathcal{P}'}$  e  $f_{\min} = \min_{x \in |\mathcal{P}'|} \{f(x)\}$ .

Considere um subconjunto  $S_z$  de segmentos incidentes em um vértice  $z \in \mathcal{G}'$ , onde cada par de segmentos adjacentes de  $S_z$  formam ângulos menores que  $60^\circ$ . O comprimento de cada segmento de  $S_z$  é uma potência de dois, pois cada um foi obtido através de sucessivas subdivisões pelo ponto médio durante o refinamento. Em um certo passo do refinamento, se o comprimento do menor subsegmento de  $S_z$  for  $2^i$ , então os vértices  $\text{vert}(S_z) - \{z\}$  estão em círculos concêntricos centrados em  $z$  e de raio  $j \cdot 2^i$  com  $j \in \mathbb{Z}^+$ ; no caso em que a distância entre dois vértices de  $\text{vert}(S_z) - \{z\}$  for menor que  $2^i$  implica que eles estão no mesmo círculo. Se um vértice  $v \in \text{vert}(S_z) - \{z\}$  invadir um subsegmento  $s \in \text{vert}(S_z)$ , então  $s$  cruza o círculo em que  $v$  está. Logo, os dois subsegmentos criados quando  $s$  é dividido não podem ser menores que os dois subsegmentos originados da divisão de  $\overline{zv}$ . Segue que o efeito cascata gerado por invasões mutuas em  $S_z$  não pode criar um segmento menor do que o já existente em  $S_z$ .

Dizemos que um par de vértices  $(x, v)$  é *insurgente* se eles estão em segmentos distintos de  $\mathcal{G}'$  formando um ângulo menor que  $60^\circ$  em um vértice em comum  $z$ , e  $d(x, z) = d(v, z)$  (estão no mesmo círculo concêntrico). Assim,  $\overline{xv}$  é uma aresta insurgente se a malha a contém. Afirmamos que o algoritmo modificado nunca irá gerar dois vértices cuja distância entre eles é menor que  $f_{\min}$ , exceto quando



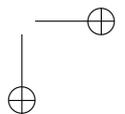
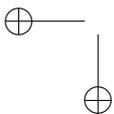


eles formarem um par insurgente. Suponhamos por contradição que  $v$  é o primeiro vértice adicionado que contradiz a afirmação. Logo, existe um vértice  $x$  tal que  $(v, x)$  não é insurgente mas  $d(v, x) < f_{\min}$ . Seja  $w$  o vértice mais próximo de  $v$  no momento em  $v$  foi inserido. Logo,  $d(v, w) \leq d(v, x) < f_{\min}$ .

Afirmamos que  $v$  não é um circuncentro nem um vértice tipo 1 cujo pai é um circuncentro rejeitado. Se  $v$  é o circuncentro de um triângulo magro  $\tau$ , a menor aresta de  $\tau$  tem comprimento maior ou igual a  $f_{\min}$ , pois o algoritmo não divide um triângulo cuja menor aresta seja insurgente, e por hipótese de indução, segue que todas as arestas não insurgentes têm comprimento maior ou igual a  $f_{\min}$  antes de  $v$  ser inserido. Por outro lado,  $\rho(\tau) > \bar{\rho}$ , então seu circunraio é maior que  $\bar{\rho}f_{\min} > \sqrt{2}f_{\min}$ . Logo,  $d(v, w) > \sqrt{2}f_{\min}$ , uma contradição. Se  $v$  é um vértice tipo 1 cujo pai  $p$  é um circuncentro rejeitado, analogamente  $r_p > \sqrt{2}f_{\min}$ , então pela Proposição 5.5 (c) temos que  $r_v \geq r_p/\sqrt{2} > f_{\min}$ . Logo,  $d(v, w) \geq r_v > f_{\min}$ , uma contradição.

Portanto,  $v$  é um vértice tipo 1 inserido em um subsegmento invadido  $s$  de um segmento  $e$ , e o círculo diametral de  $s$  contém algum vértice invasor, logo contém  $w$ . Assim,  $w$  não é um circuncentro (que teria sido rejeitado). O fato que  $d(v, w) < f_{\min}$  implica que  $w \notin \text{vert}(\mathcal{G}')$  e não está em um subsegmento de  $e$ . O mesmo é verdade para  $x$ . Portanto,  $w$  está em um segmento de  $\mathcal{G}'$  adjacente a  $e$  em um vértice comum  $z$ . Pelo nosso raciocínio acima, os dois subsegmentos criados pela divisão de  $s$  não são menores que os dois subsegmentos que se conectam em  $w$ , os quais possuem comprimentos maiores ou iguais a  $f_{\min}$  por hipótese de indução. Mas  $d(v, x) < f_{\min}$ , então  $x$  está no círculo diametral de  $s$ . Assim  $x$ , como  $w$ , está em um segmento  $e'$  que é adjacente a  $e$  em  $z$ , além de unir dois subsegmentos cujos comprimentos são maiores ou iguais a  $f_{\min}$ . O fato que  $d(v, x) < f_{\min}$  implica que  $v$  e  $x$  estão no mesmo círculo concêntrico. O raio do círculo é pelo menos  $f_{\min}$ , então  $e$  e  $e'$  formam um ângulo menor que  $60^\circ$ . Isso contradiz nossa suposição que  $(v, x)$  não é insurgente. Seque que somente pares insurgentes podem estar separados por uma distância menor que  $f_{\min}$ .

Pelo fato que todo subsegmento possui comprimento de pelo menos  $f_{\min}$ , toda aresta insurgente possui comprimento de pelo menos



$2f_{\min} \operatorname{sen}\left(\frac{\phi_{\min}}{2}\right)$ . Segue do Lema do Empacotamento (Lema 5.2) que o algoritmo modificado termina.

No término do algoritmo, cada triângulo cuja menor aresta não é insurgente não terá nenhum ângulo menor que  $\operatorname{arcsen}\left(\frac{1}{2\rho}\right)$ , consequentemente nenhum ângulo maior que  $180^\circ - 2\operatorname{arcsen}\left(\frac{1}{2\rho}\right)$ . Para limitar os ângulos dos outros triângulos, considere a aresta insurgente  $\overline{wx}$  na Figura 5.23. Seus vértices estão em dois segmentos distintos incidentes que se encontram em um vértice  $v$  formando um ângulo  $\phi < 60^\circ$ , e o vértice  $x$  é o ponto médio de  $\overline{wy}$ . Se  $\overline{wx}$  é a menor aresta de triângulo, logo seu maior ângulo não pode exceder  $\angle wxy = 90^\circ + \phi/2 < 120^\circ$ , o que estabelece uma cota superior para  $\phi$ .

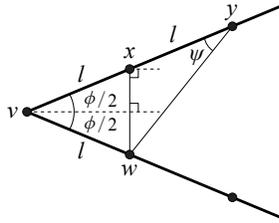


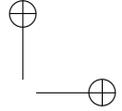
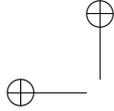
Figura 5.23: Se a menor aresta  $\overline{wx}$  de um triângulo, a qual subtende um ângulo de entrada  $\phi$ , é insurgente. Logo, o triângulo não possui nenhum ângulo menor que  $\psi$  e nem maior que  $90^\circ + \phi/2$ .

Seja  $\psi = \angle xyw$ , e observe que  $\psi < \phi$ . Nenhum triângulo de Delaunay com a menor aresta  $\overline{wx}$  pode ter um ângulo menor que  $\psi$ , devido a Proposição 4.7, tal triângulo teria  $y$  ou  $v$  no interior de seu circuncírculo. Pela Lei dos Senos,

$$\frac{\operatorname{sen}(\psi)}{d(v, w)} = \frac{\operatorname{sen}(\angle vwy)}{d(v, y)}.$$

Segue que,

$$\begin{aligned} 2 \operatorname{sen}(\psi) &= \operatorname{sen}(\angle vwy) = \operatorname{sen}(180^\circ - \phi - \psi) = \operatorname{sen}(\phi + \psi) \\ &= \operatorname{sen}(\phi) \cos(\psi) + \cos(\phi) \operatorname{sen}(\psi). \end{aligned}$$



Portanto,

$$[2 - \cos(\phi)]^2 \operatorname{sen}^2(\psi) = \operatorname{sen}^2(\phi) \cos^2(\psi) = \operatorname{sen}^2(\phi)(1 - \operatorname{sen}^2(\psi)).$$

Rearranjando os termos da equação acima, obtemos

$$\operatorname{sen}(\psi) = \frac{\operatorname{sen}(\phi)}{\sqrt{5 - 4 \cos(\phi)}},$$

o que estabelece uma cota inferior para  $\phi$ . □

## 5.7 Exercícios

5.1 Dado um triângulo  $\tau$ . Mostre que a relação entre a razão circunraio-aresta  $\rho(\tau)$  e o ângulo mínimo  $\theta_{\min}$  de  $\tau$  é dada por

$$\rho(\tau) = \frac{1}{2 \operatorname{sen}(\theta_{\min})}.$$

5.2 Dado um triângulo  $\tau$ . Mostre que razão de aspecto  $A(\tau)$  e  $\theta_{\min}$  o ângulo mínimo de  $\tau$ , satisfazem as seguintes desigualdades:

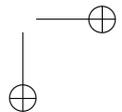
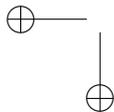
$$2 \operatorname{sen}^2\left(\frac{\theta_{\min}}{2}\right) \leq A \leq 2 \tan\left(\frac{\theta_{\min}}{2}\right).$$

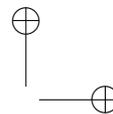
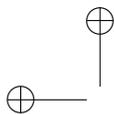
5.3 Dado um triângulo  $\tau$  cujas medidas de seu lados são  $a$ ,  $b$  e  $c$ . Mostre que a razão de aspecto de  $\tau$  pode ser escrita da forma

$$A(\tau) = \frac{abc}{8(s-a)(s-b)(s-c)},$$

onde  $s = (a + b + c)/2$  é o semiperímetro de  $\tau$ .

5.4 Seja  $f : \mathbb{E}^d \rightarrow \mathbb{R}$ . Prove que se  $f$  é  $k_f$ -Lipschitz, então  $f$  é contínua. A recíproca é verdadeira? Prove que se  $f$  também é diferenciável, então  $\|\nabla(f)\| \leq k_f$ , em que  $\nabla(f)$  é o gradiente de  $f$ .

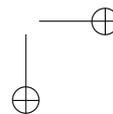
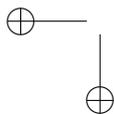


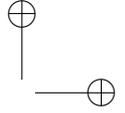
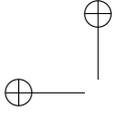


- 5.5 Forneça uma fórmula para calcular as coordenadas do centro do círculo circunscrito a um dado triângulo a partir das coordenadas de seus três vértices.
- 5.6 Descreva um exemplo de GPSR para o qual o algoritmo de Ruppert gera uma sequência de inserção com quatro pontos, na qual o primeiro ponto é do tipo 0 e os demais do tipo 1. Além disso, o primeiro é pai do segundo, o segundo é pai do terceiro e o terceiro é pai do quarto.
- 5.7 Suponha que o Algoritmo 5.4 receba como entrada um GPSR  $\mathcal{G}$  no qual alguns segmentos definem ângulos menores do que  $90^\circ$ , mas nunca menores de que  $60^\circ$ . Então, a Proposição 5.5 não é mais suficiente, pois um vértice inserido em um segmento pode invadir um subsegmento em um segmento adjacente. Mostre, no entanto, que o Algoritmo 5.4 deve terminar para qualquer escolha de  $\bar{\rho} > \sqrt{2}$ .
- 5.8 Seja  $\mathcal{TD}(\mathcal{G})$  uma triangulação de Delaunay restrita do GPSR  $\mathcal{G}$ . Mostre que se  $\mu$  é uma aresta (resp. um triângulo) de Delaunay com relação ao conjunto,  $\mathcal{V}$ , de vértices de  $\mathcal{G}$ , então  $\mu$  deve ser uma aresta (resp. triângulo) de  $\mathcal{TD}(\mathcal{G})$ . A recíproca é verdadeira?
- 5.9 Escreva o pseudocódigo de um algoritmo para gerar uma triangulação de Delaunay restrita de um GPSR  $\mathcal{G}$  a partir de uma dada triangulação restrita de  $\mathcal{G}$ . Prove que o seu algoritmo está correto.

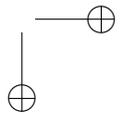
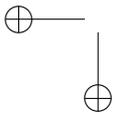
## 5.8 Notas bibliográficas

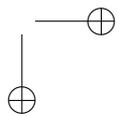
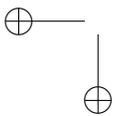
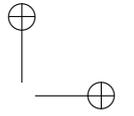
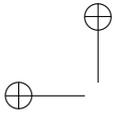
A primeira solução, com alguma garantia teórica, para o problema de geração de malhas de uma região poligonal de  $\mathbb{E}^2$  é creditada a Baker, Grosse e Rafferty [6]. Esta solução é restrita a polígonos simples, mas garante que nenhum triângulo possui um ângulo maior do que  $90^\circ$  e menor do que  $13^\circ$  (desde que o polígono não tenha nenhum ângulo menor do que  $13^\circ$ , é claro). Bern e Eppstein [9] propuseram o primeiro algoritmo de geração de malhas com garantia de forma e de cardinalidade. O algoritmo também está restrito a polígonos,

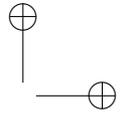
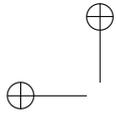




mas a análise de tamanho ótimo originou os argumentos usados até hoje. Melissaratos e Souvaine [82] propuseram extensões para o algoritmo de Bern e Eppstein. Todos algoritmos mencionados acima são baseados em uma técnica de decomposição do espaço conhecida como *quadtrees*. Embora esta técnica tenha proporcionado resultados com garantias teóricas, essas garantias se mostraram insuficientes na prática. Além disso, as malhas geradas tendem a ter arestas alinhadas com os eixos cartesianos, o que faz com que as malhas sejam menos “isotrópicas”. Paul Chew introduziu o primeiro algoritmo baseado em refinamento de Delaunay com garantias teóricas [25]. Este algoritmo fornece uma cota inferior de  $30^\circ$  para o ângulo mínimo e também produz malhas de tamanho ótimo, mas elas não são possuem transição espacial e, portanto, podem possuir um número excessivo de triângulos para serem úteis na prática. Ruppert forneceu o primeiro algoritmo de refinamento de Delaunay com garantias teóricas de valor “prático” [113]. No entanto, o algoritmo de Ruppert tinha uma séria limitação: ele pode não terminar caso o GPSR dado como entrada possua dois segmentos que definem um ângulo menor do que  $90^\circ$  em um vértice comum. Shewchuk desenvolveu uma solução para este problema [120], que nos permite aplicar o algoritmo de Ruppert em qualquer GPSR. Ele também implementou, de maneira robusta, o algoritmo de Ruppert com a modificação proposta por ele [118]. Steven Pav cita algumas desvantagens e limitações da solução proposta por Shewchuk e propõe soluções melhores para lidar com ângulos pequenos [84, 101]. Este capítulo foi quase que inteiramente baseado no recente livro de Cheng, Dey e Shewchuk sobre refinamento de Delaunay [23]. A tese de doutorado de Steven Pav [101], a tese de doutorado de Alexander Rand [107] e o artigo [113] no qual Ruppert descreve os detalhes de seu algoritmo também foram consultados. Os trabalhos recentes de Alexander Rand sobre exemplos em que o algoritmo de Ruppert não termina, quando a cota inferior no ângulo mínimo é menor do que  $\sqrt{2}$ , também contribuíram para a escrita do capítulo [109, 108], assim como sua análise do segundo algoritmo baseado em refinamento de Delaunay proposto por Paul Chew [110].







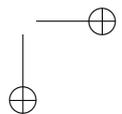
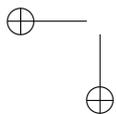
# Capítulo 6

## Aplicação

### 6.1 Considerações iniciais

Neste capítulo, apresentamos um emprego prático do processo de geração de malhas descrito no Capítulo 5 com uma aplicação do método dos elementos finitos (MEF) em um problema de análise elástica 2D. O processo de análise considerado aqui, esquematizado na Figura 6.1, começa com uma idealização do problema físico, a partir de hipóteses, quase sempre simplificadoras, baseadas em observações que devem ser comprovadas pela experimentação. Com isso, elaboramos uma teoria cujas equações definem o *modelo matemático* do problema. Neste capítulo, usamos a teoria da *mecânica do contínuo* para descrever certo tipo de comportamento de corpos elásticos 2D. Para o caso geral de geometria e condições de contorno, podemos obter uma solução do modelo matemático a partir de formulações derivadas de métodos de *discretização* do sistema contínuo, como o MEF, o qual passa a ser representado, dessa forma, por um conjunto finito de componentes sobre os quais são definidas aproximações para as variáveis do problema físico.

Dado um sólido elástico cuja espessura é pequena quando comparada com suas outras dimensões, o problema tratado aqui consiste em determinar quais os deslocamentos, deformações e tensões de cada partícula do sólido quando este é submetido à ação de forças estáticas externas aplicadas paralelamente ao seu plano médio. Nes-



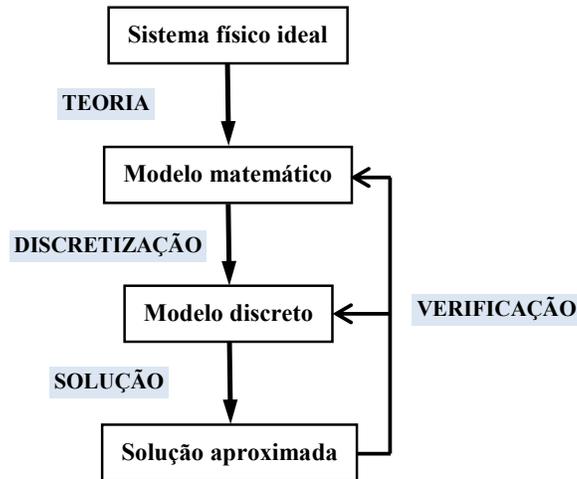
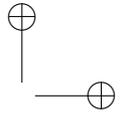
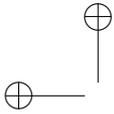


Figura 6.1: Processo de análise de um problema via MEF.

nas condições, o sólido pode ser considerado como uma *membrana* sujeita a um *estado plano de tensões*. Na Seção 6.2, introduzimos os conceitos de mecânica do contínuo associados ao problema, tais como forças, tensões, deformações, equilíbrio estático e elasticidade. O objetivo é fornecer ao leitor os elementos necessários à compreensão dos termos envolvidos na formulação do modelo matemático do estado plano de tensões, mas uma abordagem mais detalhada está além do escopo deste capítulo. Assim, vários enunciados e equações são apresentados sem as devidas demonstrações, as quais podem ser verificadas em qualquer texto introdutório de mecânica do contínuo, por exemplo [80].

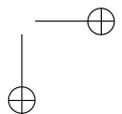
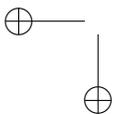
Embora soluções via MEF para problemas de mecânica do contínuo em geral possam ser matematicamente derivadas de princípios variacionais, empregamos neste capítulo o *método dos resíduos ponderados*. Primeiro, as equações diferenciais governantes do problema físico são transformadas na chamada *forma fraca* de resíduos ponderados, isto é, em equações integrais de domínio obtidas a partir da adoção de funções aproximadoras das variáveis do problema que satisfazem as *condições de contorno essenciais* do modelo matemático. Além

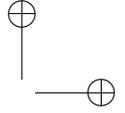
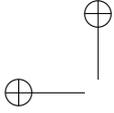


disso, no MEF, aplicamos o método de Galerkin, ou seja, as funções ponderadoras de resíduos usadas na forma fraca satisfazem a versão homogênea das condições de contorno essenciais e são tomadas do mesmo conjunto das funções aproximadoras. Em seguida, o domínio é subdividido em um número finito de regiões ou células, os *elementos finitos*, os quais são interconectados através de um número finito de *nós*. Com a discretização do domínio, a forma fraca pode ser escrita como uma soma de integrais sobre o domínio e contorno de cada elemento finito, resultando em um sistema de equações algébricas que, uma vez resolvido, fornece a solução das variáveis do problema físico nos pontos nodais da discretização. Os fundamentos concernentes à formulação do método dos elementos finitos são introduzidos na Seção 6.3, o esquema computacional na Seção 6.4, e a derivação do elemento finito considerado na análise do estado plano de tensões na Seção 6.5. Novamente, o objetivo não é abordar todos os detalhes, mas os elementos necessários ao entendimento dos passos do método e, particularmente, sua aplicação na solução do problema proposto. Uma discussão completa e aprofundada do MEF pode ser vista em [141, 139, 140].

Em problemas 2D, como o estado plano de tensões, a geometria do domínio discretizado é essencialmente representada por uma malha de elementos triangulares e/ou quadrangulares. De fato, um gerador de malhas é um componente fundamental em um sistema de análise numérica via MEF, constituindo parte intrínseca do *pré-processador* do sistema. Na Seção 6.6, usamos um programa de geração de malhas planares que, empregando as técnicas descritas no Capítulo 5, gera uma malha de triângulos a partir de segmentos de linhas que representam os contornos exterior e interiores de uma membrana, juntamente com os carregamentos e vínculos que definem suas condições de contorno. Discutimos, também, como estender a geometria e construir um domínio a partir da malha, associando aos vértices e triângulos da última nós e elementos do primeiro, aos quais são adicionados atributos específicos da aplicação, como deslocamentos e forças nodais e propriedades elásticas dos elementos. Apresentamos, por fim, um programa de elementos finitos em Matlab para análise do estado plano de tensões.<sup>1</sup>

<sup>1</sup>Disponível em <http://projetos.facom.ufms.br/orthrus/cbm>.





## 6.2 Modelo matemático do problema

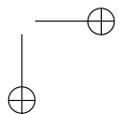
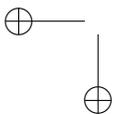
Seja um corpo *contínuo*<sup>2</sup> que, em determinado instante  $t_0 = 0$ , ocupa um volume  $V_0$  no espaço  $\mathbb{E}^3$ , delimitado por uma superfície  $S_0$ , e sobre o qual atuam forças externas. Vamos supor que o corpo tenha *vínculos* suficientes para restringir os movimentos de corpo rígido de parte de suas partículas, isto é, translações e rotações que não causam deformações. Como consequência da aplicação das forças, as partículas do corpo que não estão sujeitas a algum vínculo sofrem um *deslocamento* de suas posições atuais e passam a ocupar, em um instante  $t > t_0$ , novas posições que definem a *configuração deformada* do corpo. Nessa configuração, o corpo ocupa um volume  $V$ , delimitado por uma superfície  $S$ . Vamos admitir que a aplicação das forças externas seja feita estaticamente, isto é, que o tempo  $t$  seja longo o suficiente tal que impactos e outros efeitos dinâmicos decorrentes da aplicação possam ser negligenciados.

### Tensões e deformações

As forças externas podem ser de dois tipos: *forças de volume* e *forças de superfície*. As primeiras atuam sobre elementos de volume ou de massa do interior do corpo, como, por exemplo, a força da gravidade. Usaremos o vetor  $\mathbf{b}$  para denotar a força externa por unidade de volume que atua sobre um elemento infinitesimal  $dV$  do volume do corpo, no instante  $t$ . Neste capítulo, consideraremos somente sistemas de coordenadas cartesianas. Para simplificar a notação, usaremos o mesmo símbolo,  $\mathbf{b}$ , para denotar tanto o vetor — que em  $\mathbb{R}^3$  representa uma grandeza cuja magnitude e direção independem do sistema de coordenadas adotado — quanto a matriz coluna  $[b_1 \ b_2 \ b_3]^T$  ou  $[b_x \ b_y \ b_z]^T$  contendo os componentes cartesianos do vetor. Além disso, denotaremos por  $b_i$ , para  $i = 1, 2, 3$ , qualquer componente cartesiano de  $\mathbf{b}$ . Um componente específico poderá ser denotado, por exemplo, por  $b_1$  ou  $b_x$ , quando conveniente.

Forças de superfície são forças de contato que atuam sobre elementos de superfície do contorno do corpo. Usaremos o vetor  $\mathbf{p}^{(n)}$

<sup>2</sup>Um material é contínuo se preenche completamente o espaço que ocupa, sem vazios, e suas propriedades puderem ser descritas por funções contínuas. Essa é a hipótese primeira da teoria da mecânica do contínuo.



para denotar a força externa por unidade de área, ou *tensão*, que atua sobre um elemento infinitesimal  $dS$  em um plano com normal  $\mathbf{n}$  e que passa por um ponto  $q$  da superfície do corpo, no instante  $t$ . De fato, podemos determinar a tensão em qualquer ponto  $q$ , na direção de um versor  $\mathbf{n}$ , se seccionarmos o corpo por um plano imaginário que passa por  $q$  e tem normal  $\mathbf{n}$ , como ilustrado na Figura 6.2. Nesse caso, a tensão  $\mathbf{p}^{(\mathbf{n})}$  representa a *ação interna* que uma das porções do corpo exerce sobre a outra.

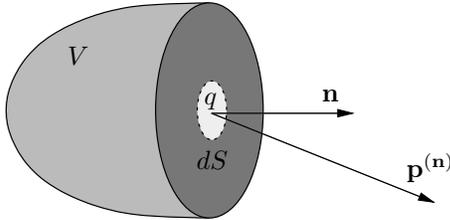


Figura 6.2: Tensão em um ponto  $q$  na direção  $\mathbf{n}$ .

A expressão para o vetor de tensão pode ser obtida a partir dos vetores de tensão, no ponto  $q$ , em planos perpendiculares a cada um dos eixos coordenados  $x_i$ . Os componentes cartesianos desses vetores de tensão definem os componentes cartesianos do *tensor de tensões de Cauchy*,  $\boldsymbol{\sigma}$ , no ponto  $q$ , os quais podem ser organizados matricialmente como

$$\boldsymbol{\sigma} = [\mathbf{p}^{(i_1)} \quad \mathbf{p}^{(i_2)} \quad \mathbf{p}^{(i_3)}]^\text{T} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix},$$

em que  $\sigma_{ij} = p_j^{(i_i)}$ , para  $1 \leq i, j \leq 3$ , denota um componente qualquer do tensor — o primeiro índice indicando o plano de atuação e o segundo a direção de atuação, considerada positiva no sentido positivo do eixo cartesiano correspondente —, e  $\mathbf{p}^{(i_i)}$  é o vetor de tensão do plano perpendicular ao eixo  $x_i$ , conforme ilustrado na Figura 6.3. O tensor de tensões de Cauchy no ponto  $q$  representa uma transformação linear, chamada *transformação de tensão de Cauchy*, que leva o vetor normal  $\mathbf{n}$  no vetor de tensão  $\mathbf{p}^{(\mathbf{n})}$ . Em coordenadas cartesianas, tal transformação pode ser expressa pelo produto matricial

$\mathbf{p}^{(\mathbf{n})} = \mathbf{n}^T \cdot \boldsymbol{\sigma}$ , ou, em *notação indicial*, por  $p_i^{(\mathbf{n})} = \sigma_{ji}n_j$ , em que o índice  $j$ , repetido uma vez no mesmo termo, é chamado *índice mudo* e indica somatório, ou seja, para  $i = 1, 2, 3$ ,

$$p_i^{(\mathbf{n})} = \sum_{j=1}^3 \sigma_{ji}n_j = \sigma_{1i}n_1 + \sigma_{2i}n_2 + \sigma_{3i}n_3.$$

Na ausência de momentos de volume, como admitido aqui, o tensor de tensões de Cauchy é simétrico, isto é,  $\sigma_{ij} = \sigma_{ji}$ .

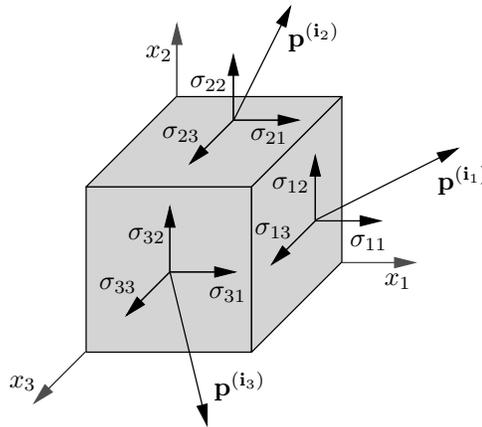


Figura 6.3: Componentes do tensor de tensões de Cauchy.

Se um corpo está em *equilíbrio estático* na configuração deformada, então

$$\int_V \mathbf{b} dV + \int_S \mathbf{p} dS = 0. \quad (6.1)$$

Podemos transformar a integral de superfície da Equação (6.1) em uma integral de volume, primeiro aplicando a transformação de tensão de Cauchy e, depois, utilizando o teorema da divergência,

$$\int_S p_i dS = \int_S \sigma_{ji}n_j dS = \int_V \frac{\partial \sigma_{ji}}{\partial x_j} dV. \quad (6.2)$$

Combinando as Equações (6.1) e (6.2), obtemos a *equação de equilíbrio*

$$\int_V \frac{\partial \sigma_{ji}}{\partial x_j} dV + \int_V b_i dV = 0,$$

ou

$$\frac{\partial \sigma_{ji}}{\partial x_j} + b_i = 0. \tag{6.3}$$

Em notação indicial, a equação de equilíbrio pode ser escrita como  $\sigma_{ji,j} + b_i = 0$ , em que a vírgula indica derivada parcial de  $\sigma_{ji}$  em relação a  $x_j$ . Note que  $j$  é índice mudo; logo, para cada direção  $i = 1, 2, 3$ , temos  $\sigma_{1i,1} + \sigma_{2i,2} + \sigma_{3i,3} + b_i = 0$ .

Além da continuidade, vamos assumir que, sob ação de forças externas, um corpo apresentará *pequenos deslocamentos e pequenas deformações*, quando comparados com a unidade. Em mecânica do contínuo, as descrições de deformação são baseadas em medidas quantitativas de certos tipos de deslocamentos relativos entre partículas vizinhas do corpo. (Um sólido, em geral, resiste a tais deslocamentos relativos de suas partes, o que resulta nas tensões internas discutidas anteriormente; contudo, nem todos os tipos de deslocamentos relativos causam deformações e tensões em um sólido, como é o caso da parte rotacional de um movimento de corpo rígido.) Sejam, então, duas partículas vizinhas  $p$  e  $q$  de um corpo que, na configuração não deformada, ocupam as posições  $\mathbf{X}$  e  $\mathbf{X} + d\mathbf{X}$ , respectivamente, em que o vetor  $d\mathbf{X}$  representa o elemento de linha infinitesimal, de comprimento  $dL$ , formado por  $p$  e  $q$ , como mostrado na Figura 6.4.

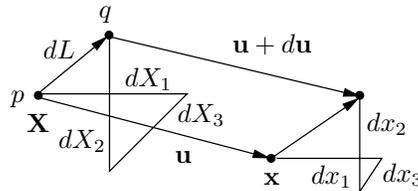


Figura 6.4: Deformação de um elemento de linha infinitesimal.

Observe que estamos representando a posição do *ponto*, no espaço  $\mathbb{E}^3$ , onde a partícula  $p$  está, por um *vetor*  $\mathbf{X}$ , uma vez que, fixado um

sistema de coordenadas cartesianas, podemos definir  $p = O + \mathbf{X}$ , em que  $O$  é a origem do sistema, conforme visto no Capítulo 2.

Após a aplicação das forças externas, as partículas passam a ocupar, no instante  $t$ , as posições  $\mathbf{x} = \mathbf{X} + \mathbf{u}$  e  $\mathbf{x} + d\mathbf{u}$ , respectivamente, em que  $d\mathbf{u}$  é o deslocamento relativo de  $q$  em relação a  $p$ . O vetor de *deslocamento relativo unitário* é definido como

$$\frac{du_i}{dL} = \frac{\partial u_i}{\partial X_j} \frac{dX_j}{dL},$$

(somatório em  $j$ ), ou,

$$\begin{bmatrix} \frac{du_x}{dL} \\ \frac{du_y}{dL} \\ \frac{du_z}{dL} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_x}{\partial X} & \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial Z} \end{bmatrix} \begin{bmatrix} \frac{dX}{dL} \\ \frac{dY}{dL} \\ \frac{dZ}{dL} \end{bmatrix},$$

a qual pode ser escrita abreviadamente como

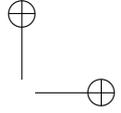
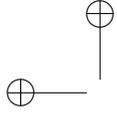
$$\frac{d\mathbf{u}}{dL} = \mathbf{J}_u \mathbf{n},$$

em que  $\mathbf{n}$  é o versor na direção de  $d\mathbf{X}$ . A matriz quadrada  $\mathbf{J}_u$ , chamada *matriz de deslocamentos relativos unitários*, pode ser composta a partir da soma de duas matrizes, uma antissimétrica,

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \frac{1}{2} \left( \frac{\partial u_x}{\partial Y} - \frac{\partial u_y}{\partial X} \right) & \frac{1}{2} \left( \frac{\partial u_x}{\partial Z} - \frac{\partial u_z}{\partial X} \right) \\ \frac{1}{2} \left( \frac{\partial u_y}{\partial X} - \frac{\partial u_x}{\partial Y} \right) & 0 & \frac{1}{2} \left( \frac{\partial u_y}{\partial Z} - \frac{\partial u_z}{\partial Y} \right) \\ \frac{1}{2} \left( \frac{\partial u_z}{\partial X} - \frac{\partial u_x}{\partial Z} \right) & \frac{1}{2} \left( \frac{\partial u_z}{\partial Y} - \frac{\partial u_y}{\partial Z} \right) & 0 \end{bmatrix},$$

e outra simétrica,

$$\boldsymbol{\epsilon} = \begin{bmatrix} \frac{\partial u_x}{\partial X} & \frac{1}{2} \left( \frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \right) & \frac{1}{2} \left( \frac{\partial u_x}{\partial Z} + \frac{\partial u_z}{\partial X} \right) \\ \frac{1}{2} \left( \frac{\partial u_y}{\partial X} + \frac{\partial u_x}{\partial Y} \right) & \frac{\partial u_y}{\partial Y} & \frac{1}{2} \left( \frac{\partial u_y}{\partial Z} + \frac{\partial u_z}{\partial Y} \right) \\ \frac{1}{2} \left( \frac{\partial u_z}{\partial X} + \frac{\partial u_x}{\partial Z} \right) & \frac{1}{2} \left( \frac{\partial u_z}{\partial Y} + \frac{\partial u_y}{\partial Z} \right) & \frac{\partial u_z}{\partial Z} \end{bmatrix},$$



em que  $\boldsymbol{\Omega}$  e  $\boldsymbol{\epsilon}$  são os tensores de pequenas rotações e de pequenas deformações, respectivamente, associados ao campo de deslocamentos infinitesimais  $\mathbf{u}$ . É possível mostrar que, se  $\omega_x = -\Omega_{yz}$ ,  $\omega_y = -\Omega_{zx}$  e  $\omega_z = -\Omega_{xy}$  forem todos pequenos comparados a um radiano, então  $\boldsymbol{\Omega} \cdot \mathbf{n}$  representa o deslocamento relativo unitário produzido por uma pequena rotação do elemento de linha infinitesimal  $d\mathbf{X}$  em torno de um eixo com direção  $\boldsymbol{\omega}$ , de um ângulo  $|\boldsymbol{\omega}|$ .  $\boldsymbol{\epsilon}$ , por sua vez, leva  $\mathbf{n}$  à parcela do deslocamento relativo unitário devido às deformações do material na vizinhança de  $p$ . Como os deslocamentos são pequenos, podemos escrever, em notação indicial,

$$\epsilon_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} \right) = \frac{1}{2} (u_{i,j} + u_{j,i}). \quad (6.4)$$

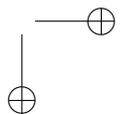
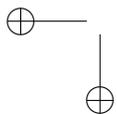
O tensor  $\boldsymbol{\epsilon}$ , simétrico, tem seis componentes distintos definidos em função de somente três componentes de deslocamentos. Daí, resulta que as deformações não são todas independentes, mas devem satisfazer certas *condições de compatibilidade*, a fim de que o campo de deslocamentos associado seja um campo contínuo e possível de ser obtido a partir das ações externas aplicadas ao corpo. As condições de compatibilidade de deformações são

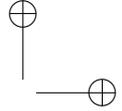
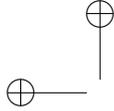
$$\epsilon_{ij,kl} + \epsilon_{kl,ij} = \epsilon_{ik,jl} + \epsilon_{jl,ik}. \quad (6.5)$$

### Relações constitutivas elásticas

Tensões e deformações podem ser associadas através das chamadas *equações constitutivas*. Vamos considerar somente materiais contínuos que são *homogêneos*, *isótropos* e idealmente *elásticos*. Um material é homogêneo se tem propriedades idênticas em todos os pontos; isótropo em relação a determinadas propriedades se estas forem as mesmas em todas as direções; e elástico se um corpo formado pelo material recupera completamente sua forma original, cessadas as forças externas causadoras da deformação, e quando há uma relação unívoca entre o estado de tensão e deformação, para uma dada temperatura.

As relações constitutivas elásticas, chamadas *lei de Hooke generalizada*, são nove equações que expressam os componentes de tensão





como funções lineares homogêneas dos nove componentes de deformação:

$$\sigma_{ij} = C_{ijrs} \epsilon_{rs}. \quad (6.6)$$

O tensor de quarta ordem  $\mathbf{C}$ , chamado *tensor de módulos elásticos*, possui 81 componentes, mas como  $\boldsymbol{\sigma}$  e  $\boldsymbol{\epsilon}$  são simétricos, podemos supor, sem perda de generalidade, que  $C_{ijrs} = C_{jirs}$  e  $C_{ijrs} = C_{ijsr}$ . Nesse caso, somente 36 componentes do tensor são independentes. Para materiais isótropos,  $\mathbf{C}$  é um tensor *isótropo*<sup>3</sup> de quarta ordem definido como

$$C_{ijrs} = \lambda \delta_{ij} \delta_{rs} + \mu (\delta_{ir} \delta_{js} + \delta_{is} \delta_{jr}), \quad (6.7)$$

em que  $\lambda$  e  $\mu$  são as constantes de Lamé, e

$$\delta_{ij} = \begin{cases} 1 & \text{se } i = j, \\ 0 & \text{se } i \neq j \end{cases}$$

é o *delta de Kronecker*. A versão isótropa da lei de Hooke generalizada, Equação (6.6), fica

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij}. \quad (6.8)$$

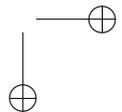
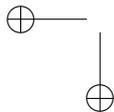
As constantes de Lamé são relacionadas com o módulo de cisalhamento  $G$ , o módulo de elasticidade  $E$  (ou módulo de Young) e o coeficiente de Poisson  $\nu$  como segue:

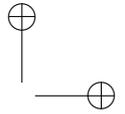
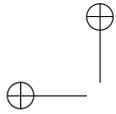
$$G = \mu = \frac{E}{2(1 + \nu)}, \quad E = \frac{\mu(3\lambda + 2\mu)}{\lambda + \mu} \quad \text{e} \quad \nu = \frac{\lambda}{2(\lambda + \mu)}.$$

### Modelo matemático geral

Em resumo, o modelo matemático que governa o comportamento de um corpo de material contínuo, homogêneo, isótropo e elástico, submetido a forças estáticas causadoras de pequenos deslocamentos e pequenas deformações, é composto de: equações de equilíbrio

<sup>3</sup>Um tensor isótropo é aquele cujos componentes são invariantes ao sistema de coordenadas cartesianas adotado.





(6.3), equações de deformação-deslocamento (6.4) e equações constitutivas (6.8), em um total de quinze equações distintas para três deslocamentos e (devido à simetria dos tensores de deformações e tensões) seis deformações e seis tensões.

Podemos ter as seguintes *condições de contorno* para as equações de campo do modelo:

- 6.1 *Deslocamentos prescritos*, com os três deslocamentos  $u_i$  conhecidos no contorno do corpo.
- 6.2 *Forças de superfície prescritas*, com os três componentes de tensão  $p_i = \sigma_{ji}n_j$  conhecidos no ponto onde o contorno do corpo possui normal  $\mathbf{n}$  (vamos omitir o superescrito ( $\mathbf{n}$ ) para simplificar).
- 6.3 *Condições de contorno mistas*, com (a) deslocamentos prescritos sobre uma parte do contorno e forças de superfície prescritas no restante do contorno, ou (b)  $u_i$  ou  $p_i$  prescrito em cada ponto do contorno, mas não ambos.

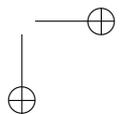
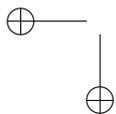
A solução de um problema envolvendo quinze equações para quinze incógnitas não é uma tarefa fácil. Há várias maneiras de reformular o modelo matemático com o propósito de obtermos um número menor de incógnitas e um número menor de equações. O método mais direto consiste em substituir a Equação (6.4) em (6.8) para obtermos as tensões em termos dos gradientes de deslocamentos, e, então, substituir o resultado na Equação (6.3) para obtermos três equações diferenciais parciais de segunda ordem para os três componentes de deslocamento:

$$G u_{j,kk} + \frac{G}{1-2\nu} u_{k,jk} + b_j = 0. \quad (6.9)$$

A Equação (6.9) é conhecida como *equação de Navier da elasticidade*. (Uma vez que temos deslocamentos como incógnitas e que estes são relacionados às forças causadoras da deformação, as condições de compatibilidade (6.5) não necessitam ser adicionadas ao modelo.)

### Estado plano de tensões

Uma *membrana* é um objeto geometricamente definido por duas superfícies planas, como ilustrado na Figura 6.5. A distância entre



as superfícies,  $h$ , ou espessura da membrana, é admitida pequena quando comparada com as demais dimensões do objeto. O plano equidistante às duas superfícies que definem a membrana é denominado plano médio, ou simplesmente o “plano” da membrana, para simplificar. Admitiremos, ainda, que os carregamentos sejam sempre paralelos ao plano, aplicados por unidade de comprimento no contorno e uniformemente distribuídos ao longo da espessura.

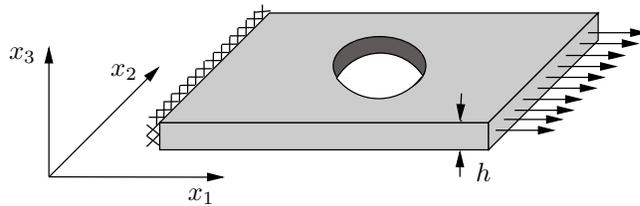


Figura 6.5: Geometria e carregamento de uma membrana.

O modelo matemático de uma membrana é uma simplificação do modelo geral apresentado anteriormente, sendo caracterizado por uma distribuição bidimensional de tensões na qual supomos

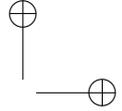
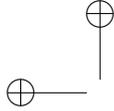
$$\sigma_{i3} = \sigma_{3i} = 0, \quad (6.10)$$

sendo a direção 3 perpendicular ao plano da membrana. Podemos supor, também, que as tensões sejam constantes ao longo da espessura da membrana e que todas as hipóteses da elasticidade linear, definidas anteriormente, sejam válidas.<sup>4</sup> O estado de tensões em um ponto qualquer do plano da membrana pode ser então especificado somente pelos componentes  $\sigma_{11}$ ,  $\sigma_{22}$  e  $\sigma_{12} = \sigma_{21}$  do tensor de tensões, sendo denominado *estado plano de tensões*.

Note que o objeto é livre para contrair ou expandir na direção 3. A partir da condição (6.10), podemos determinar o componente de deformação fora do plano,  $\epsilon_{33}$ . Usando a lei de Hooke, obtemos

$$\epsilon_{33} = -\frac{\lambda}{\lambda + 2\mu} \epsilon_{ll}, \quad l = 1, 2. \quad (6.11)$$

<sup>4</sup>A hipótese (6.10) viola algumas das condições de compatibilidade (6.5) e a variação de tensões ao longo da espessura ocorre, mas podem ser ignoradas para membranas “suficientemente delgadas”.



(O componente de deformação fora do plano é obtido em função dos componentes de deformação no plano.) A lei de Hooke isotrópica para o estado plano de tensões é definida substituindo a Equação (6.11) na expressão (6.8), resultando

$$\sigma_{ij} = \delta_{ij} \frac{2\lambda\mu}{\lambda + 2\mu} \epsilon_{kk} + 2\mu \epsilon_{ij}, \quad i, j, k = 1, 2. \quad (6.12)$$

Matricialmente, podemos escrever  $\boldsymbol{\sigma} = \mathbf{C} \cdot \boldsymbol{\epsilon}$ , em que

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ 2\epsilon_{xy} \end{bmatrix} \quad \text{e} \quad \mathbf{C} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}. \quad (6.13)$$

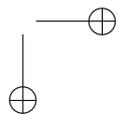
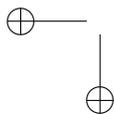
### 6.3 Formulação do MEF

Seja um corpo elástico de domínio  $\Omega$  e contorno  $\Gamma$  para o qual desejamos resolver a equação de Navier (6.9). Tal solução fornece os deslocamentos das partículas do corpo na configuração deformada, a partir dos quais podemos determinar as deformações e tensões, de acordo com as relações descritas na Seção 6.2. Para o estado plano de tensões, admitindo que a espessura  $h$  é constante ao longo da membrana,  $\Omega$  é uma superfície planar, possivelmente com cavidades, e  $\Gamma$  as curvas externa e internas que a delimitam. Nesse problema, cada partícula do corpo tem dois componentes de deslocamento, ou *graus de liberdade*,  $u_1$  e  $u_2$  (ou  $u_x$  e  $u_y$ ), aos quais são associadas forças  $f_1$  e  $f_2$  (ou  $f_x$  e  $f_y$ ), como mostrado adiante. Para unicidade da solução, devemos impor, em partes  $\Gamma_u$  e  $\Gamma_p$  (não necessariamente contíguas) do contorno, condições de contorno essenciais  $u_k = \bar{u}_k$  e naturais  $p_k = \bar{p}_k$ , respectivamente.

#### Resíduos ponderados

Para obtermos uma solução para os deslocamentos, vamos utilizar, ao invés da equação de Navier, a equação de equilíbrio estático em sua forma indicial,

$$\sigma_{jk,j} + b_k = 0 \quad \text{em } \Omega, \quad (6.14)$$



Seja o campo de deslocamentos  $\mathbf{u}_0$ , desconhecido, a solução exata do problema, a qual será aproximada no domínio  $\Omega$  por

$$\mathbf{u}_0 \approx \mathbf{u} = \sum_{i=1}^n \alpha_i \phi_i, \quad (6.15)$$

em que  $\alpha_i$  são parâmetros, quase todos incógnitos, e  $\phi_i$  são funções linearmente independentes tomadas de um conjunto completo de funções. Se admitirmos que a função aproximadora do campo de deslocamentos  $\mathbf{u}$  (ou  $u_k$ , em notação indicial) seja  $C^2$  e satisfaça exatamente as condições de contorno essenciais e naturais do problema, mas não a equação de equilíbrio (6.14), então temos um *resíduo*  $\sigma_{jk,j} + b_k \neq 0$  em  $\Omega$ . O propósito é tornar tal resíduo tão “pequeno” quanto possível, o que pode ser feito através de uma distribuição ponderada sobre o domínio,

$$\int_{\Omega} (\sigma_{jk,j} + b_k) u_k^* d\Omega = 0, \quad (6.16)$$

em que  $\mathbf{u}^*$  é uma *função ponderadora* que representa um campo de deslocamentos qualquer sobre um domínio  $\Omega^*$  qualquer.<sup>5</sup> Assumiremos que os gradientes do campo de deslocamentos ponderador  $\mathbf{u}^*$  sejam pequenos em relação à unidade (ou seja, as relações deslocamento-deformação (6.4) são válidas para  $\mathbf{u}^*$ ) e que as equações constitutivas elásticas (6.6) sejam satisfeitas em  $\Omega^*$ .

A Equação (6.16) pode ser generalizada para o caso da função aproximadora  $u_k$  não satisfazer as condições de contorno do problema. Nesse caso,

$$\begin{aligned} u_k - \bar{u}_k &\neq 0 && \text{em } \Gamma_u, \text{ e} \\ p_k - \bar{p}_k &\neq 0 && \text{em } \Gamma_p \end{aligned} \quad (6.17)$$

são os erros, ou resíduos, cometidos ao aproximarmos as condições de contorno essenciais e naturais, respectivamente. Para obtermos uma expressão que relacione os erros de domínio e de contorno, vamos

<sup>5</sup>Como tensões dependem das deformações e estas são derivadas de deslocamentos, o integrando da Equação (6.16) envolve derivadas segundas dos deslocamentos, por isso a ordem de continuidade  $C^2$  requerida para a função aproximadora  $\mathbf{u}$ .

integrar por partes<sup>6</sup> a Equação (6.16),

$$-\int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega + \int_{\Omega} b_k u_k^* d\Omega = -\int_{\Gamma} p_k u_k^* d\Gamma, \quad (6.18)$$

em que  $\epsilon_{jk}^*$  é o campo de deformações associado à função ponderadora  $u_k^*$ . Devido à simetria do tensor de módulos elásticos  $C_{jkli}$ , o primeiro termo da Equação (6.18) pode ser escrito como

$$\int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega = \int_{\Omega} \sigma_{jk}^* \epsilon_{jk} d\Omega. \quad (6.19)$$

Integrando por partes o primeiro termo de (6.18) e levando em consideração a reciprocidade (6.19), obtemos a *forma transposta* da Equação (6.16),

$$\int_{\Omega} \sigma_{jk,j}^* u_k d\Omega + \int_{\Omega} b_k u_k^* d\Omega = -\int_{\Gamma} p_k u_k^* d\Gamma + \int_{\Gamma} u_k p_k^* d\Gamma. \quad (6.20)$$

Integrando por partes duas vezes o primeiro termo e depois substituindo as condições de contorno na Equação (6.20), obtemos a *sentença original* de resíduos ponderados para o problema elastostático,

$$\int_{\Omega} (\sigma_{jk,j} + b_k) u_k^* d\Omega = \int_{\Gamma_p} (p_k - \bar{p}_k) u_k^* d\Gamma - \int_{\Gamma_u} (u_k - \bar{u}_k) p_k^* d\Gamma, \quad (6.21)$$

supondo uma função  $\mathbf{u}$  que aproxima a equação diferencial em  $\Omega$  e as condições de contorno em  $\Gamma$ .

Admitimos até agora que a função aproximadora  $\mathbf{u}$  possui ordem de continuidade tal que  $\sigma_{jk,j} \neq 0$  em  $\Omega$  (desde que  $b_k \neq 0$ ). Em muitos casos é possível, e vantajoso, reduzir a ordem de continuidade requerida para a função  $\mathbf{u}$  através de uma integração por partes da sentença original (6.21). Além disso, vamos supor que a função aproximadora satisfaça as condições de contorno essenciais, isto é,  $u_k \equiv \bar{u}_k$  em  $\Gamma_u$ . Assim, a Equação (6.21) fica

$$\int_{\Omega} \sigma_{jk} \epsilon_{jk}^* d\Omega - \int_{\Omega} b_k u_k^* d\Omega = \int_{\Gamma_p} \bar{p}_k u_k^* d\Gamma + \int_{\Gamma_u} p_k u_k^* d\Gamma. \quad (6.22)$$

<sup>6</sup>Podemos utilizar, por exemplo, o teorema de divergência.

Na Equação (6.22), uma ordem menor de continuidade para  $\mathbf{u}$  é requerida, ao custo de uma ordem maior para a função ponderadora  $\mathbf{u}^*$ . Esta equação é uma *forma fraca* de resíduos ponderados para o problema elastostático, supondo que a função  $\mathbf{u}$  aproxima a equação diferencial em  $\Omega$  e as condições de contorno naturais em  $\Gamma_p$ .

### Princípio dos trabalhos virtuais

Derivaremos a formulação do método dos elementos finitos a partir de uma variação da forma fraca de resíduos ponderados chamada *método de Galerkin*. De acordo com o método de Galerkin, escolhemos uma função ponderadora  $u_k^* = \delta u_k$ , em que  $\delta u_k$  é um campo de deslocamentos infinitesimais definido a partir do mesmo conjunto de funções linearmente independentes utilizado na definição de  $u_k$ , Equação (6.15). Vamos supor que esse campo de deslocamentos satisfaça a versão homogênea das condições de contorno essenciais, ou seja,  $u_k^* = \delta u_k \equiv 0$  em  $\Gamma_u$ . Então, a Equação (6.22) se reduz a

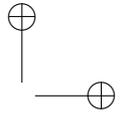
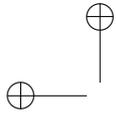
$$\int_{\Omega} \sigma_{jk} \delta \epsilon_{jk} d\Omega = \int_{\Omega} b_k \delta u_k d\Omega + \int_{\Gamma_p} \bar{p}_k \delta u_k d\Gamma,$$

ou, em notação matricial,

$$\int_{\Omega} \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} d\Omega = \int_{\Omega} \delta \mathbf{u}^T \mathbf{b} d\Omega + \int_{\Gamma_p} \delta \mathbf{u}^T \mathbf{p} d\Gamma, \quad (6.23)$$

sendo  $\boldsymbol{\sigma}$  e  $\boldsymbol{\epsilon}$  vetores de componentes de tensão e de deformação, respectivamente, os quais, para o estado plano de tensões, são dados pelas expressões (6.13).

A Equação (6.23) é a expressão do *princípio dos trabalhos virtuais* (PTV). Se considerarmos que os deslocamentos  $\delta \mathbf{u}$  são *deslocamentos virtuais* aplicados a uma configuração de equilíbrio de um corpo, então podemos interpretar o lado direito da Equação (6.23) como o trabalho externo  $\delta W_{\text{ext}}$  realizado pelas forças de volume e de superfície atuando no domínio e no contorno do corpo durante os deslocamentos imaginários  $\delta \mathbf{u}$ . O lado esquerdo da equação pode ser interpretado como a energia de deformação  $\delta U$  armazenada no corpo durante os deslocamentos imaginários  $\delta \mathbf{u}$ . Na configuração de



equilíbrio,  $\delta U = \delta W_{\text{ext}}$ . A função ponderadora  $\delta \mathbf{u}$  deve satisfazer a versão homogênea das condições de contorno essenciais porque os deslocamentos virtuais devem ser *cinematicamente admissíveis*, ou seja, devem satisfazer quaisquer deslocamentos prescritos. (Desde que os deslocamentos virtuais são deslocamentos *adicionais* impostos a uma configuração de equilíbrio de um corpo, um componente de deslocamento virtual deve ser nulo sempre que o deslocamento atual seja prescrito por uma condição de contorno.) Embora a expressão (6.23) possa ser derivada da aplicação de métodos variacionais em mecânica dos sólidos, foi escrita aqui simplesmente como uma forma fraca da equação de equilíbrio (6.14), supondo uma função aproximadora satisfazendo as condições de contorno essenciais e uma função ponderadora de Galerkin satisfazendo a versão homogênea das condições essenciais.

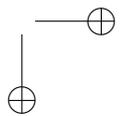
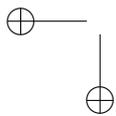
### Modelo discreto do MEF

O MEF é baseado na subdivisão do domínio  $\Omega$  e do contorno  $\Gamma$  de um corpo em um conjunto de *elementos finitos* interconectados em pontos discretos, ou *nós*, como ilustrado com a membrana da Figura 6.6.<sup>7</sup> Com a subdivisão do corpo em NE elementos finitos, a Equação (6.23) pode ser escrita como somas de integrais sobre o domínio e o contorno de cada elemento finito:

$$\begin{aligned} \sum_{e=1}^{\text{NE}} \int_{\Omega^{(e)}} [\delta \boldsymbol{\epsilon}^{(e)}]^T \boldsymbol{\sigma}^{(e)} d\Omega = \\ \sum_{e=1}^{\text{NE}} \int_{\Omega^{(e)}} [\delta \mathbf{u}^{(e)}]^T \mathbf{b}^{(e)} d\Omega - \sum_{e=1}^{\text{NE}} \int_{\Gamma_p^{(e)}} \delta \mathbf{u}^{(e)T} \mathbf{p}^{(e)} d\Gamma, \end{aligned} \quad (6.24)$$

em que, para cada elemento finito  $e = 1, 2, \dots, \text{NE}$ ,  $\boldsymbol{\sigma}^{(e)}$  são os componentes cartesianos das tensões em um ponto qualquer de  $e$ ,  $\mathbf{b}^{(e)}$  e  $\mathbf{p}^{(e)}$  são, respectivamente, as forças de volume e de superfície atuantes

<sup>7</sup>A peça da Figura 6.6 é uma adaptação para o plano de um modelo fornecido como exemplo em um programa de geração de malhas chamado `Gmsh`, disponível em [geuz.org/gmsh](http://geuz.org/gmsh). Tanto o modelo geométrico quanto à malha foram criados com o programa que desenvolvemos para este capítulo.



em  $e$ , e  $\delta \mathbf{u}^{(e)}$  e  $\delta \boldsymbol{\epsilon}^{(e)}$  são, respectivamente, os componentes cartesianos dos deslocamentos e deformações “virtuais” em um ponto qualquer de  $e$ . Assim, ao invés de construir uma função aproximadora global  $\mathbf{u}$ , Equação (6.15), com um número de coeficientes a determinar igual ao número de graus de liberdade não restringidos do sistema discreto, podemos encontrar uma aproximação (e sua variação  $\delta \mathbf{u}$ ) contínua por partes, a partir das contribuições individuais de cada elemento finito.

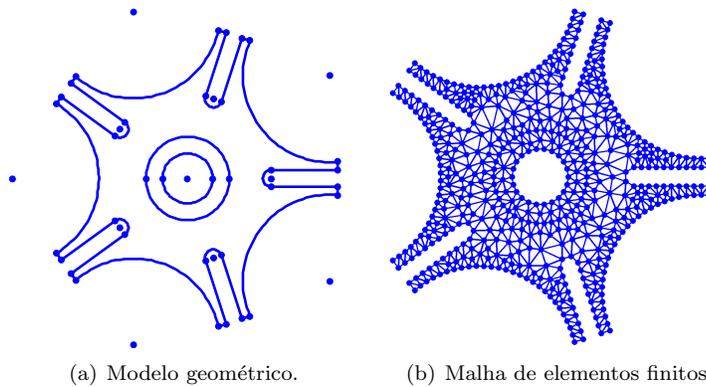


Figura 6.6: Discretização de um corpo em elementos finitos.

Consideremos, então, uma malha com  $NN$  nós, com coordenadas globais tomadas em relação a um sistema de coordenadas cartesianas, cada nó rotulado com um *índice global*  $v = 1, 2, \dots, NN$ . Seja um elemento finito  $e$  definido por  $n$  nós, cada nó rotulado com um *índice local*  $i = 1, 2, \dots, n$ , e  $q$  um ponto qualquer de  $e$ , com coordenadas  $\boldsymbol{\xi}$  tomadas em relação a um *sistema de coordenadas normalizadas* (também chamadas de *coordenadas intrínsecas* ou *coordenadas homogêneas*) do elemento, em que  $0 \leq \xi_i \leq 1$ . No caso de elementos planares, um sistema de coordenadas intrínsecas é definido por dois eixos não colineares  $\xi_1$  e  $\xi_2$  (ou  $\xi$  e  $\eta$ ), como ilustrado na Figura 6.7 para o elemento triangular linear.

A cada nó  $i$  de um elemento associamos uma função escalar da posição  $\boldsymbol{\xi}$ ,  $N_i^{(e)}$ , tal que  $N_i^{(e)}(\boldsymbol{\xi}_j^{(e)}) = \delta_{ij}$ , em que  $\boldsymbol{\xi}_i^{(e)}$  são as coordenadas normalizadas do nó  $i$ . A partir de tais funções, podemos mapear

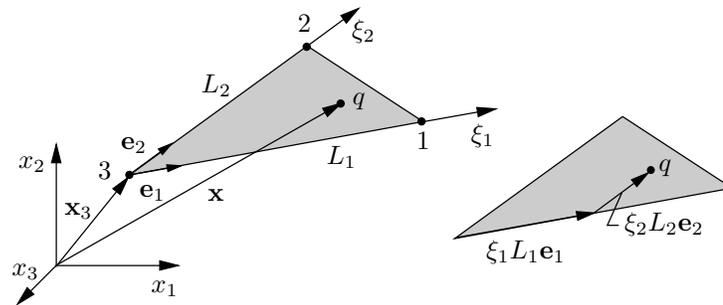


Figura 6.7: Coordenadas normalizadas do triângulo linear.

Da Figura 6.7, temos que o vetor posição do ponto  $q$  é

$$\mathbf{x} = \mathbf{x}_3 + \xi_1 L_1 \mathbf{e}_1 + \xi_2 L_2 \mathbf{e}_2, \quad (6.26)$$

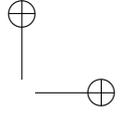
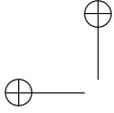
em que

$$\mathbf{e}_1 = \frac{x_{1k} - x_{3k}}{L_1} \mathbf{i}_k, \quad \mathbf{e}_2 = \frac{x_{2k} - x_{3k}}{L_2} \mathbf{i}_k \quad (6.27)$$

(somatório em  $k$ ), sendo  $x_{i_k}$  é o  $k$ -ésimo componente de  $\mathbf{x}_i$ . Levando a Equação (6.27) na Equação (6.26), podemos escrever

$$\mathbf{x} = \xi_1 \mathbf{x}_1 + \xi_2 \mathbf{x}_2 + (1 - \xi_1 - \xi_2) \mathbf{x}_3 = \sum_{i=1}^3 \xi_i \mathbf{x}_i, \quad (6.28)$$

em que, por conveniência, definimos  $\xi_3 = 1 - \xi_1 - \xi_2$  (de fato, a coordenada baricêntrica  $\xi_3$  não é uma nova coordenada intrínseca,



uma vez que não é independente de  $\xi_1$  e  $\xi_2$ ). Portanto, as funções de forma do triangular linear são

$$\begin{aligned} N_1 &= \xi_1 = \xi, \\ N_2 &= \xi_2 = \eta, \\ N_3 &= \xi_3 = 1 - \xi - \eta. \end{aligned} \tag{6.29}$$

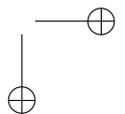
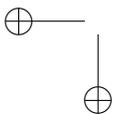
Similarmente, associamos aos nós do elemento funções escalares da posição  $\xi$  com as quais podemos interpolar, em  $q$ , o valor de uma grandeza física qualquer do problema — por exemplo, o deslocamento —, a partir dos valores de tal grandeza nos nós do elemento. Em geral, essas funções interpoladoras podem ser distintas das funções interpoladoras de geometria, mas, neste capítulo, adotaremos as *mesmas* funções para interpolar tanto a geometria quanto as grandezas físicas. Elementos finitos com tal característica são denominados *isoparamétricos*, e as funções interpoladores são chamadas genericamente de *funções de forma*. Para o problema do estado plano de tensões, consideraremos somente elementos finitos 2D do tipo mais simples, o isoparamétrico triangular linear, cujas funções de forma são dadas pelas expressões (6.29). A Figura 6.8 mostra, a título de ilustração, o elemento triangular *quadrático*, com três nós adicionais no meio de cada aresta, cujas funções de forma são

$$\begin{aligned} N_1 &= \xi_1 (2 \xi_1 - 1), \\ N_2 &= \xi_2 (2 \xi_2 - 1), \\ N_3 &= \xi_3 (2 \xi_3 - 1), \\ N_4 &= 4 \xi_1 \xi_2, \\ N_5 &= 4 \xi_2 \xi_3, \\ N_6 &= 4 \xi_3 \xi_1. \end{aligned}$$

Com as funções de forma, o vetor de deslocamentos em um ponto qualquer  $q$  do elemento  $e$  pode ser aproximado por

$$\mathbf{u}^{(e)} = \sum_{i=1}^n N_i^{(e)} \mathbf{u}_i^{(e)}, \tag{6.30}$$

sendo  $\mathbf{u}_i^{(e)}$  o vetor de deslocamentos do nó  $i$ . Matricialmente, pode-



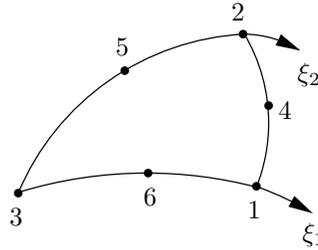


Figura 6.8: Triângulo quadrático.

mos escrever a Equação (6.30) como

$$\mathbf{u}^{(e)} = \begin{bmatrix} \mathbf{N}_1^{(e)} & \mathbf{N}_2^{(e)} & \dots & \mathbf{N}_n^{(e)} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^{(e)} \\ \mathbf{u}_2^{(e)} \\ \dots \\ \mathbf{u}_n^{(e)} \end{bmatrix} = \mathbf{N}^{(e)} \mathbf{U}^{(e)}, \quad (6.31)$$

sendo  $\mathbf{N}^{(e)}$  a *matriz de funções de forma* e  $\mathbf{U}^{(e)}$  é o vetor dos deslocamentos de todos os nós do elemento  $e$ . A matriz de funções de forma é constituída pelas submatrizes

$$\mathbf{N}_i^{(e)} = N_i^{(e)} \mathbf{I},$$

em que  $\mathbf{I}$  é a matriz identidade de ordem igual ao número de graus de liberdade de um nó (dois, no estado plano de tensões).

A partir da Equação (6.31), temos que o vetor de componentes de pequenas deformações em um ponto qualquer  $q$  de um elemento finito  $e$  é dado por

$$\boldsymbol{\epsilon}^{(e)} = \mathbf{B}^{(e)} \mathbf{U}^{(e)} = \sum_{i=1}^n \mathbf{B}_i^{(e)} \mathbf{u}_i^{(e)}, \quad (6.32)$$

em que  $\mathbf{B}^{(e)}$  é a *matriz de deformação-deslocamento* de  $e$  e  $\mathbf{B}_i^{(e)}$  é a submatriz deformação-deslocamento associada ao nó  $i$ , definida, para

o estado plano de tensões, como

$$\mathbf{B}_i^{(e)} = \begin{bmatrix} \frac{\partial N_i^{(e)}}{\partial x} & 0 \\ 0 & \frac{\partial N_i^{(e)}}{\partial y} \\ \frac{\partial N_i^{(e)}}{\partial y} & \frac{\partial N_i^{(e)}}{\partial x} \end{bmatrix}. \quad (6.33)$$

O vetor de componentes de tensão em um ponto qualquer de um elemento finito  $e$ , considerando as relações constitutivas elásticas e a Equação (6.32), é

$$\boldsymbol{\sigma}^{(e)} = \mathbf{C}^{(e)} \mathbf{B}^{(e)} \mathbf{U}^{(e)} = \mathbf{C}^{(e)} \left( \sum_{i=1}^n \mathbf{B}_i^{(e)} \mathbf{u}_i^{(e)} \right), \quad (6.34)$$

em que  $\mathbf{C}^{(e)}$ , a *matriz constitutiva* do elemento  $e$ , para o estado plano de tensões, é dada na Equação (6.13).

As versões “virtuais” dos deslocamentos e das deformações em um ponto  $q$  do elemento  $e$  podem agora ser definidas como

$$\begin{aligned} \delta \mathbf{u}^{(e)} &= \mathbf{N}^{(e)} \delta \mathbf{U}^{(e)} = \sum_{i=1}^n \mathbf{N}_i^{(e)} \delta \mathbf{u}_i^{(e)}, \quad e \\ \delta \boldsymbol{\epsilon}^{(e)} &= \mathbf{B}^{(e)} \delta \mathbf{U}^{(e)} = \sum_{i=1}^n \mathbf{B}_i^{(e)} \delta \mathbf{u}_i^{(e)}, \end{aligned} \quad (6.35)$$

em que  $\delta \mathbf{U}^{(e)}$  é o vetor de deslocamentos virtuais de todos os nós de  $e$ . Substituindo as Equações (6.34) e (6.35) na expressão do PTV (6.23), e levando em conta que os deslocamentos virtuais  $\delta \mathbf{U}^{(e)}$  são arbitrários, obtemos, para o elemento  $e$ ,

$$\int_{\Omega^{(e)}} [\mathbf{B}^{(e)}]^T \mathbf{C}^{(e)} \mathbf{B}^{(e)} \mathbf{U}^{(e)} d\Omega = \int_{\Omega^{(e)}} [\mathbf{N}^{(e)}]^T \mathbf{b}^{(e)} d\Omega + \int_{\Gamma_p^{(e)}} [\mathbf{N}^{(e)}]^T \mathbf{p}^{(e)} d\Gamma,$$

a qual pode ser colocada na forma

$$\mathbf{K}^{(e)} \mathbf{U}^{(e)} = \mathbf{F}^{(e)}. \quad (6.36)$$

Na Equação (6.36),  $\mathbf{K}^{(e)}$  é a *matriz de rigidez do elemento* e  $\mathbf{F}^{(e)}$  é o *vetor de carregamentos nodais equivalentes do elemento*, definidos, respectivamente, como

$$\mathbf{K}^{(e)} = \int_{\Omega^{(e)}} [\mathbf{B}^{(e)}]^T \mathbf{C}^{(e)} \mathbf{B}^{(e)} d\Omega, \quad (6.37)$$

$$\mathbf{F}^{(e)} = \int_{\Omega^{(e)}} [\mathbf{N}^{(e)}]^T \mathbf{b}^{(e)} d\Omega + \int_{\Gamma_p^{(e)}} [\mathbf{N}^{(e)}]^T \mathbf{p}^{(e)} d\Gamma = \mathbf{F}_b^{(e)} + \mathbf{F}_p^{(e)}. \quad (6.38)$$

A partir das relações (6.37) e (6.38), a Equação (6.24) pode ser escrita como

$$\sum_{e=1}^{NE} \mathbf{K}^{(e)} \mathbf{U}^{(e)} = \sum_{e=1}^{NE} \mathbf{F}^{(e)}, \quad (6.39)$$

ou

$$\mathbf{K} \mathbf{U} = \mathbf{F}, \quad (6.40)$$

em que  $\mathbf{K}$  é a *matriz de rigidez global*,  $\mathbf{F}$  é o *vetor de carregamentos nodais equivalentes* e  $\mathbf{U}$  é o vetor de incógnitas nodais de todos os nós do modelo discreto.

Os somatórios da Equação (6.39) são efetuados através da adição das contribuições de cada elemento de tal forma que  $\mathbf{K}$ ,  $\mathbf{U}$  e  $\mathbf{F}$  sejam organizados em partições nodais, conforme descrito na Seção 6.4. A solução do sistema linear (6.40), após a introdução das condições de contorno, fornece os deslocamentos incógnitos do problema, a partir dos quais podemos determinar as deformações e tensões em todos os nós do modelo discreto e, por interpolação, em qualquer ponto do domínio. Devido ao fato dos deslocamentos nodais serem incógnitos, a formulação do método dos elementos finitos resumida aqui é chamada de *método dos deslocamentos*.

## 6.4 Esquema computacional

De acordo com a formulação introduzida na Seção 6.3, a análise elastostática de um corpo pelo método dos elementos finitos pode ser efetuada pelos seguintes passos, discutidos a seguir: (1) discretização

do corpo, (2) computação das contribuições dos elementos, (3) montagem do sistema linear, (4) introdução das condições de contorno, (5) solução do sistema linear, e (6) computação dos valores de domínio.

### Discretização do corpo

Neste passo inicial, o domínio  $\Omega$  e o contorno  $\Gamma$  do corpo são subdivididos em NE elementos finitos sobre os quais são aproximados, em termos de funções de interpolação e de parâmetros nodais, as variáveis do problema físico. Essa etapa é chamada pré-processamento. Na Seção 6.6 descrevermos o pré-processador utilizado para geração de malhas triangulares no programa de análise do estado plano de tensões.

### Computação das contribuições dos elementos

Neste passo, a matriz de rigidez  $\mathbf{K}^{(e)}$  e os carregamentos nodais  $\mathbf{F}^{(e)}$  são determinados para cada elemento finito  $e$ .

Para o caso geral, as integrais envolvidas nas Equações (6.37) e (6.38) são normalmente avaliadas com o emprego de técnicas de integração numérica, tais como a *quadratura gaussiana*, ou, para triângulos, a *quadratura de Hammer*. Para um elemento triangular, por exemplo, a integração de uma função  $f(\boldsymbol{\xi})$  qualquer sobre o domínio  $\Omega^{(e)}$  do elemento pode ser escrita como

$$I = \int_{\Omega^{(e)}} f(\boldsymbol{\xi}) d\Omega = \int_0^1 \int_0^{1-\xi_2} f(\boldsymbol{\xi}) J d\xi_1 d\xi_2. \quad (6.41)$$

Na Equação (6.41), efetuamos uma mudança dos limites de integração, de coordenadas cartesianas para coordenadas normalizadas, uma vez que  $f$  é uma função da posição  $\boldsymbol{\xi}$ . Como consequência, transformamos o elemento de domínio  $d\Omega$  para o sistema de coordenadas normalizadas através da relação

$$d\Omega = J d\xi_1 d\xi_2,$$

em que o *jacobiano* da transformação, para o elemento triangular, é definido como<sup>8</sup>

$$J = \frac{1}{2} \left| \frac{\partial \mathbf{x}}{\partial \xi_1} \times \frac{\partial \mathbf{x}}{\partial \xi_2} \right|$$

<sup>8</sup>Dados dois vetores  $\mathbf{u}$  e  $\mathbf{v}$  em  $\mathbb{R}^3$ , o *produto vetorial*  $\mathbf{w} = \mathbf{u} \times \mathbf{v}$  é um vetor

A integral (6.41) pode ser computada numericamente como

$$I \approx \sum_{i=1}^n J f(\boldsymbol{\xi}^{(i)}) w^{(i)}, \quad (6.42)$$

em que  $\boldsymbol{\xi}^{(i)}$  são as coordenadas normalizadas do  $i$ -ésimo *ponto de integração* e  $w^{(i)}$  é o *peso* associado, sendo o número de pontos de integração,  $n$ , escolhido em função da precisão desejada (para mais detalhes, bem como tabelas de coordenadas e pesos para diversos valores de  $n$ , veja [61]).

Para o triângulo linear,  $J = A$  em todo ponto de  $\Omega^{(e)}$ , em que  $A$  é a área do triângulo, e por isso o jacobiano pode ser colocado fora do somatório na Equação (6.42). Para o estado plano de tensões, especificamente, não necessitaremos de integração numérica, pois, como veremos na Seção 6.5, a matriz de rigidez e o vetor de esforços nodais equivalentes serão computados através de formas fechadas.

### Montagem do sistema linear

Neste passo, o lado esquerdo  $\mathbf{K}$  e o lado direito  $\mathbf{F}$  do sistema (6.40) são montados a partir das contribuições de todos os NE elementos finitos do modelo discreto.

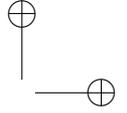
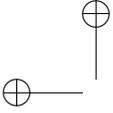
Consideremos a matriz de rigidez  $\mathbf{K}^{(e)}$  de um elemento finito  $e$  com  $n$  nós. O número de linhas e colunas de matriz é igual a  $n \times d$ , em que  $d$  é o número de graus de liberdade por nó (para o estado plano de tensões,  $d = 2$ ). Cada elemento da matriz de rigidez relaciona um elemento do vetor de deslocamentos  $\mathbf{U}^{(e)}$ , definido na Equação (6.35), com um elemento do vetor de esforços nodais equivalentes

$$\mathbf{F}^{(e)} = \left[ \mathbf{f}_1^{(e)} \quad \mathbf{f}_2^{(e)} \quad \dots \quad \mathbf{f}_n^{(e)} \right]^T, \quad (6.43)$$

sendo  $\mathbf{f}_i^{(e)}$  os esforços equivalentes do nó  $i$ . A relação entre o esforço nodal equivalente associado ao  $a$ -ésimo grau de liberdade do nó  $i$ ,  $F_l^{(e)} = \mathbf{f}_{i_a}^{(e)}$ , e o  $b$ -ésimo grau de liberdade do nó  $j$ ,  $U_c^{(e)} = \mathbf{u}_{j_b}^{(e)}$ , é

$$F_l^{(e)} = K_{lc}^{(e)} U_c^{(e)}, \quad (6.44)$$

com direção perpendicular ao plano do paralelogramo formado por  $\mathbf{u}$  e  $\mathbf{v}$  (se não nulos e não colineares), magnitude igual à área do paralelogramo, e componentes  $w_x = u_y v_z - u_z v_y$ ,  $w_y = u_z v_x - u_x v_z$ , e  $w_z = u_x v_y - v_y u_x$ .



em que os índices locais dos graus de liberdade são

$$l = (i - 1) \times d + a, \quad c = (j - 1) \times d + b, \quad 1 \leq a, b \leq d. \quad (6.45)$$

A mesma expressão (6.44) vale globalmente, ou seja, cada um dos  $NN \times d$  elementos da matriz de rigidez global  $\mathbf{K}$ , definida na Equação (6.40), relaciona um elemento do vetor de esforços nodais equivalentes,  $\mathbf{F}$ , com um elemento do vetor de deslocamentos,  $\mathbf{U}$ , do sistema discreto:

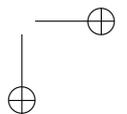
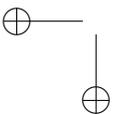
$$F_r = K_{rs} U_s. \quad (6.46)$$

Os índices globais  $r$  e  $s$  usados na Equação (6.46), correspondentes aos índices locais  $l$  e  $c$  da Equação (6.45), são dados por

$$r = (v^{(i)} - 1) \times d + a, \quad s = (v^{(j)} - 1) \times d + b, \quad (6.47)$$

em que  $v^{(i)}$  e  $v^{(j)}$  são os índices globais dos nós  $i$  e  $j$  do elemento  $e$ , respectivamente. Portanto, todo elemento finito  $e$  que incidir em nós  $i$  e  $j$  com índices globais  $v^{(i)}$  e  $v^{(j)}$ , terá o elemento  $K_{lc}^{(e)}$  da sua matriz de rigidez local, correspondente aos graus de liberdade locais  $l$  e  $c$ , adicionado ao elemento  $K_{rs}$  da matriz de rigidez global, correspondente aos graus de liberdade globais  $r$  e  $s$ , de acordo com as relações (6.46). Da forma similar, o elemento  $F_l^{(e)}$  do vetor de esforços nodais equivalentes de  $e$  será adicionado ao elemento  $F_r$  do vetor de esforços nodais equivalentes global. Assim, a montagem do sistema linear (6.40), definido pelos somatórios da Equação (6.39), pode ser feita como descrito no Algoritmo 6.1, o qual supõe que  $\mathbf{K}$  e  $\mathbf{F}$ , inicialmente, têm todos os elementos nulos.

A matriz de rigidez global é *esparsa*, uma vez que, em uma linha  $r$ , são não nulos somente aqueles elementos  $K_{rs}$  em que  $r$  e  $s$  são graus de liberdade de nós conectados através de pelo menos um elemento finito. Além disso, a matriz para o problema elastostático é simétrica e tem característica de *banda*, isto é, os elementos não nulos se concentram em uma faixa acima e abaixo da diagonal principal, sendo a *largura de banda* dependente da maior diferença dos índices globais de quaisquer nós adjacentes. Tais propriedades motivam o emprego de esquemas próprios de armazenamento — por exemplo, arranjos retangulares contendo apenas a semibanda superior (incluída a diagonal) —, bem como esquemas de numeração nodal que tem por



---

**Algoritmo 6.1** Montagem do sistema linear do MEF.

---

```

1: para  $e = 1, 2, \dots, NE$  faça
2:    $n \leftarrow$  número de nós de  $e$ 
3:   para  $i = 1, 2, \dots, n$  faça
4:     para  $a = 1, 2, \dots, d$  faça
5:        $l \leftarrow (i - 1) \times d + a$ 
6:        $r \leftarrow (v^{(i)} - 1) \times d + a$ 
7:        $F_r \leftarrow F_r + F_l^{(e)}$ 
8:       para  $j = 1, 2, \dots, n$  faça
9:         para  $b = 1, 2, \dots, d$  faça
10:           $c \leftarrow (j - 1) \times d + b$ 
11:           $s \leftarrow (v^{(j)} - 1) \times d + b$ 
12:           $K_{rs} \leftarrow K_{rs} + K_{lc}^{(e)}$ 
13:        fim para
14:      fim para
15:    fim para
16:  fim para
17: fim para

```

---

objetivo minimizar a largura de banda. Uma discussão de tais esquemas está além do escopo deste capítulo. No programa desenvolvido para o estado plano de tensões, confiamos ao Matlab o armazenamento esparsa da matriz de rigidez global.

### Introdução das condições de contorno

A matriz de rigidez ainda tem a seguinte propriedade: para toda linha  $r$ ,

$$K_{rr} - \sum_{r \neq s} K_{rs} = 0, \quad K_{rr} > 0,$$

isto é, a matriz é singular e positiva semidefinida. Neste passo,  $\mathbf{K}$ , originalmente singular, é transformada em uma matriz regular com a consideração de um número apropriado de condições de contorno essenciais, as quais são definidas pela vinculação do corpo.

Há várias formas de aplicar as condições de contorno. Vamos empregar, aqui, um esquema que consiste em *reduzir* a matriz de rigidez,

eliminando as linhas e colunas correspondentes a graus de liberdade cujos valores são prescritos. Na prática, tal redução não se dá *após* a montagem do sistema de linear, descrita anteriormente, mas *conjuntamente* com esta. Assim, vamos escrever uma nova versão do Algoritmo 6.1 na qual a matriz de rigidez é montada em função das condições de contorno essenciais. Antes, porém, vamos determinar, conforme o Algoritmo 6.2, o número de equações do sistema reduzido, e atribuir, a cada grau de liberdade não prescrito, o número de sua equação no sistema linear. Para isso, vamos considerar que as informações a respeito das condições de contorno são armazenadas em um vetor  $\mathbf{V}$  com  $NV$  elementos, em que um elemento  $i$  é uma estrutura com três campos:  $V_i.v$ , o índice global  $1, 2, \dots, NN$  do nó no qual o grau de liberdade é prescrito,  $V_i.a$ , o número  $1, 2, \dots, d$  do grau de liberdade, e  $V_i.u$ , o valor do deslocamento prescrito. Os números das equações dos deslocamentos incógnitos, isto é, não prescritos, serão armazenados em um vetor  $\mathbf{R}$  com  $NN \times d$  elementos, inicialmente iguais a zero. Ao final do algoritmo, se  $R_i = -i$ , então  $U_i$  é prescrito, caso contrário, é incógnita correspondente à linha  $R_i$  do sistema linear.

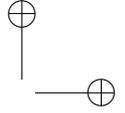
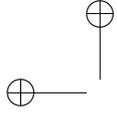
---

**Algoritmo 6.2** Numeração dos graus de liberdade incógnitos.

---

- 1: **para**  $i = 1, 2, \dots, NV$  **faça**
  - 2:    $r \leftarrow (V_i.v - 1) \times d + V_i.a$
  - 3:    $U_r \leftarrow V_i.u$
  - 4:    $R_r \leftarrow -r$
  - 5: **fim para**
  - 6:  $r \leftarrow 0$
  - 7: **para**  $l = 1, 2, \dots, NN \times d$  **faça**
  - 8:   **se**  $R_l = 0$  **então**
  - 9:      $r \leftarrow r + 1$
  - 10:     $R_l \leftarrow r$
  - 11:   **fim se**
  - 12: **fim para**
  - 13: **devolva**  $r$
- 

Vamos denotar por  $\mathbf{R}^{(e)}$  o vetor com  $n \times d$  elementos, obtidos de  $\mathbf{R}$ , contendo os números de equações dos graus de liberdade do elemento finito  $e$ . O esquema de montagem do sistema linear com



redução é descrito no Algoritmo 6.3. Na linha 5, a condição  $r > 0$  é verdadeira se o grau de liberdade  $r$  não é prescrito, ou seja, se  $U_r$  é incógnito, caso em que o elemento finito  $e$  contribui para o elemento  $F_r$  do vetor de esforços nodais equivalentes e para a linha  $r$  da matriz de rigidez global. Caso contrário, a linha  $r$  é eliminada do sistema. Da mesma forma, na linha 12, a condição  $s > 0$  é verdadeira se o grau de liberdade  $s$  não é prescrito, caso em que o elemento finito  $e$  contribui para o elemento  $K_{rs}$  da matriz de rigidez global. Caso contrário, a coluna  $s$  é eliminada do sistema, mas, se o deslocamento prescrito é não nulo, então o valor correspondente à força  $K_{lc}^{(e)} \times U_{-s}$  é transportado para o lado direito do sistema, como feito na linha 12.

---

**Algoritmo 6.3** Montagem do sistema com condições de contorno.

---

```

1: para  $e = 1, 2, \dots, NE$  faça
2:    $n \leftarrow$  número de nós de  $e$ 
3:   para  $l = 1, 2, \dots, n \times d$  faça
4:      $r \leftarrow R_l^{(e)}$ 
5:     se  $r > 0$  então
6:        $F_r \leftarrow F_r + F_l^{(e)}$ 
7:       para  $c = 1, 2, \dots, n \times d$  faça
8:          $s \leftarrow R_c^{(e)}$ 
9:         se  $s > 0$  então
10:           $K_{rs} \leftarrow K_{rs} + K_{lc}^{(e)}$ 
11:        senão
12:           $F_r \leftarrow F_r - K_{lc}^{(e)} \times U_{-s}$ 
13:        fim se
14:      fim para
15:    fim se
16:  fim para
17: fim para

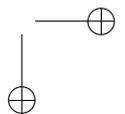
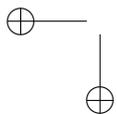
```

---

**Solução do sistema linear**

Neste passo, obtemos numericamente a solução do sistema linear (6.40), o qual, após a imposição das condições de contorno, fica

$$\bar{\mathbf{K}} \bar{\mathbf{U}} = \bar{\mathbf{F}}, \quad (6.48)$$



em que  $\bar{\mathbf{K}}$  e  $\bar{\mathbf{F}}$  são a matriz e o vetor de esforços nodais equivalentes reduzidos, respectivamente. A solução de (6.48) fornece os deslocamentos incógnitos,  $\bar{\mathbf{U}}$ , nos pontos e direções em que esses valores não são prescritos. Usando o vetor  $\mathbf{R}$ , que contém o número da equação de cada incógnita do sistema conforme determinado no Algoritmo 6.2, podemos completar, com o Algoritmo 6.4, o vetor de deslocamentos globais  $\mathbf{U}$ .

---

**Algoritmo 6.4** Preenchimento do vetor de deslocamentos.

---

- 1: **para**  $i = 1, 2, \dots, NN \times d$  **faça**
  - 2:   **se**  $R_i \neq 0$  **então**
  - 3:      $U_i \leftarrow \bar{U}_{R_i}$
  - 4:   **fim se**
  - 5: **fim para**
- 

Uma discussão a respeito de métodos de solução de sistemas lineares (com matrizes positivas semidefinidas), que nesse caso dependem do esquema de armazenamento adotado para a matriz de rigidez, está além do escopo desse livro. O programa de análise do estado plano de tensões desenvolvido para este capítulo, novamente, confia nos métodos de solução empregados pelo Matlab.

### Computação de valores no domínio

Conhecidos os deslocamentos nodais  $\mathbf{U}$  do modelo discreto, podemos determinar os deslocamentos, deformações e tensões em qualquer ponto do domínio e do contorno. Em adição, é desejável o emprego de técnicas de *visualização* que nos permitem compreender, mais fácil e expressamente, a distribuição das variáveis físicas computadas na análise. Essa etapa é chamada de pós-processamento e não será abordada nesse texto. Como exemplo, a Figura 6.9(a) mostra um *mapa de cores* correspondente ao componente de deslocamento  $u_2$  para o modelo da Figura 6.6, quando submetido a forças de volume constantes na direção  $-y$  e com os deslocamentos restritos ao longo do contorno interno. A Figura 6.9(b) mostra a malha de elementos finitos na configuração deformada em uma escala aumentada em 800 vezes. Outros resultados de análise podem ser vistos na Seção 6.6.

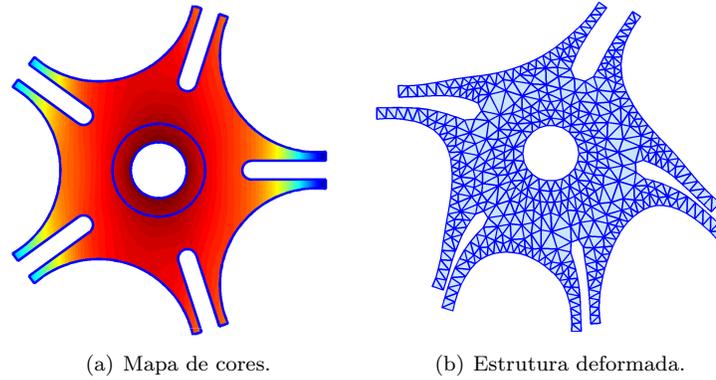


Figura 6.9: Exemplo de visualização de resultados de análise.

## 6.5 Elemento finito CST

Vamos admitir que o triângulo da Figura 6.7 esteja contido no plano  $xy$  e, para maior clareza, vamos denotar por  $x$  e  $y$  as coordenadas cartesianas do ponto  $q$  e por  $x_i$  e  $y_i$  as coordenadas cartesianas do vértice  $i$ . A Equação (6.28), em que  $\xi_1 + \xi_2 + \xi_3 = 1$ , pode ser matricialmente escrita como

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}. \quad (6.49)$$

A área do triângulo pode ser dada por

$$2A = \begin{vmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{vmatrix} = A_{23} + A_{31} + A_{12}, \quad (6.50)$$

em que

$$\begin{aligned} A_{23} &= x_2 y_3 - x_3 y_2, \\ A_{31} &= x_3 y_1 - x_1 y_3, \\ A_{12} &= x_1 y_2 - x_2 y_1. \end{aligned}$$

A área definida pela Equação (6.50) é um número com sinal, sendo positivo se os vértices do triângulo são numerados no sentido anti-

horário, com o eixo  $z$  saindo do plano. Invertendo a Equação (6.49), obtemos

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} A_{23} & y_{2-3} & x_{3-2} \\ A_{31} & y_{3-1} & x_{1-3} \\ A_{12} & y_{1-2} & x_{2-1} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}, \quad (6.51)$$

em que  $y_{i-j} = y_i - y_j$  e  $x_{i-j} = x_i - x_j$ . A partir da Equação (6.51), podemos escrever as derivadas das funções de forma do triângulo como

$$\frac{\partial N_i}{\partial x} = \frac{y_{j-k}}{2A}, \quad \frac{\partial N_i}{\partial y} = \frac{x_{k-j}}{2A}, \quad (6.52)$$

sendo os índices  $i, j, k$  uma permutação cíclica de 1, 2, 3.

Podemos, agora, formular todas as relações matriciais para elemento triangular linear. A matriz de funções de forma do elemento, definida na Equação (6.30), fica

$$\mathbf{N}^{(e)} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}. \quad (6.53)$$

Usando as relações (6.52), a matriz de deformação-deslocamento do elemento, definida na Equação (6.32), fica

$$\mathbf{B}^{(e)} = \frac{1}{2A} \begin{bmatrix} y_{2-3} & 0 & y_{3-1} & 0 & y_{1-2} & 0 \\ 0 & x_{3-2} & 0 & x_{1-3} & 0 & x_{2-1} \\ x_{3-2} & y_{2-3} & x_{1-3} & y_{3-1} & x_{2-1} & y_{1-2} \end{bmatrix}. \quad (6.54)$$

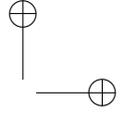
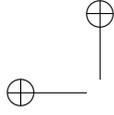
Note que as deformações são *constantes* sobre o elemento, motivo pelo qual o triângulo linear, para o estado plano de tensões, é denominado CST (*constant strain triangle*).

A matriz de rigidez do elemento, definida pela Equação (6.37), é

$$\mathbf{K}^{(e)} = [\mathbf{B}^{(e)}]^T \mathbf{C}^{(e)} \mathbf{B}^{(e)} \int_{\Omega^{(e)}} d\Omega,$$

uma vez que as matrizes  $\mathbf{B}^{(e)}$  e  $\mathbf{C}^{(e)}$ , definidas na Equação (6.13) e pela Equação (6.54), respectivamente, são constantes e, por isso, escritas fora da integral, a qual se reduz a  $\int_{\Omega^{(e)}} d\Omega$ , que é a própria área do triângulo. Portanto, temos que a matriz de rigidez do elemento CST é dada pela forma fechada

$$\mathbf{K}^{(e)} = A [\mathbf{B}^{(e)}]^T \mathbf{C}^{(e)} \mathbf{B}^{(e)}. \quad (6.55)$$



O vetor de esforços nodais equivalentes do elemento, definido pela Equação (6.38), envolve um termo referente a forças de volume e outro a forças de superfície. Levando a Equação (6.53) na Equação (6.38), o termo correspondente às forças de volume fica

$$\mathbf{F}_b^{(e)} = \int_{\Omega^{(e)}} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix} d\Omega, \quad (6.56)$$

o que implica que temos que avaliar as integrais  $\int_{\Omega^{(e)}} N_i d\Omega$ . Ao invés de usarmos quadratura de Hammer, como discutido na Seção 6.3, vamos aplicar a seguinte regra particular de integração de termos polinomiais para o triângulo linear:

$$\frac{1}{2A} \int_{\Omega^{(e)}} \xi_1^i \xi_2^j \xi_3^k d\Omega = \frac{i! j! k!}{(1 + j + k + 2)!}, \quad i, j, k \geq 0, \quad (6.57)$$

com a qual obtemos

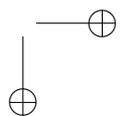
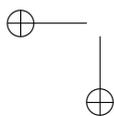
$$\int_{\Omega^{(e)}} N_i d\Omega = \frac{A}{3}. \quad (6.58)$$

Usando a expressão (6.58) na Equação (6.56), temos que o termo do vetor de esforços nodais equivalentes devido a forças de volume constantes sobre o elemento é

$$\mathbf{F}_b^{(e)} = \frac{A}{3} [b_x \quad b_y \quad b_x \quad b_y \quad b_x \quad b_y]^T. \quad (6.59)$$

Para o termo devido às forças de superfície, vamos considerar, sem perda de generalidade, uma carga uniformemente distribuída por unidade de comprimento,  $\mathbf{p}$ , aplicada ao longo da aresta 3-1 do triângulo. Vamos denotar por  $p_{3_x}$  e  $p_{3_y}$  os valores do carregamento no nó 3, e por  $p_{1_x}$  e  $p_{1_y}$  os valores do carregamento no nó 1. Ao longo da aresta, temos  $N_2 = \xi_2 = 0$  e, portanto,  $N_3 = 1 - N_1$ . Para qualquer ponto sobre a aresta temos, por interpolação,

$$\begin{aligned} p_x &= N_1 p_{1_x} + (1 - N_1) p_{3_x}, \\ p_y &= N_1 p_{1_y} + (1 - N_1) p_{3_y}. \end{aligned} \quad (6.60)$$



Levando a Equação (6.53) na Equação (6.38), o termo correspondente às forças de superfície fica

$$\mathbf{F}_p^{(e)} = \int_{\Gamma_p^{(e)}} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ 0 & 0 \\ 0 & 0 \\ 1 - N_1 & 0 \\ 0 & 1 - N_1 \end{bmatrix} \begin{bmatrix} N_1 p_{1_x} + (1 - N_1) p_{3_x} \\ N_1 p_{1_y} + (1 - N_1) p_{3_y} \end{bmatrix} d\Gamma. \quad (6.61)$$

Fazendo uma mudança dos limites de integração na integral de linha (6.61), de coordenadas cartesianas para coordenadas normalizadas, temos que

$$d\Gamma = L_1 d\xi,$$

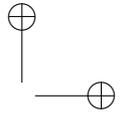
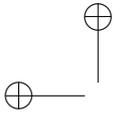
em que o comprimento da aresta 3-1,  $L_1$ , é o jacobiano da transformação do elemento de contorno  $d\Gamma$  para o sistema de coordenadas normalizadas. Assim, a integração analítica da Equação (6.61) fornece

$$\mathbf{F}_p^{(e)} = L_1 \begin{bmatrix} (2p_{1_x} + p_{3_x})/6 \\ (2p_{1_y} + p_{3_y})/6 \\ 0 \\ 0 \\ (p_{1_x} + 2p_{3_x})/6 \\ (p_{1_y} + 2p_{3_y})/6 \end{bmatrix}. \quad (6.62)$$

A Equação (6.62) pode ser diretamente generalizada para forças de superfície uniformemente distribuídas ao longo de qualquer aresta  $i-j$  do triângulo.

## 6.6 Aspectos de implementação

A implementação consiste de pacotes contendo funções e classes de objetos, escritos em Matlab, com as quais é possível definir o modelo geométrico de uma região plana, gerar uma malha a partir da geometria, construir um domínio a partir da malha, analisar tal domínio com o MEF, e visualizar os resultados de análise. As principais funções e classes são comentadas a seguir.



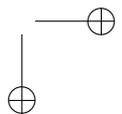
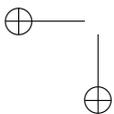
### Definição do modelo geométrico

A Figura 6.10 ilustra o modelo geométrico de uma superfície simples. A construção do modelo começa com a definição de *pontos*, mostrados como pequenos círculos na Figura 6.10(a). Em seguida, os pontos são usados para definição de *segmentos* de curvas. Para simplificar, consideramos dois tipos de segmentos: *linha*, definida pelo ponto inicial e ponto final, e *arco* de círculo, definido pelo centro e pontos inicial e final (os quais devem ser equidistantes do centro), como ilustrado na Figura 6.10(b). (Podemos adotar quaisquer outros tipos de segmentos, por exemplo, *splines*.) Uma sequência  $s_1, s_2, \dots, s_n$  de segmentos, tal que o ponto final do segmento  $s_i$ , para  $i = 1, 2, \dots, n$ , é igual ao ponto inicial do segmento  $s_j$ , em que  $j = (i \bmod n) + 1$ , forma um *laço* (fechado) que representa um contorno da superfície, como mostrado na Figura 6.10(c). A superfície é definida por um ou mais laços, sendo o primeiro laço o contorno externo e os demais, se houver, contornos internos (cujos pontos são supostos no interior do laço externo), como ilustrado na Figura 6.10(d). Se a *orientação* de um laço interno é oposta à orientação do laço externo, então o contorno interno correspondente é uma *cavidade* da superfície; caso contrário, representa uma curva cujos pontos devem ser preservados na malha a ser gerada (por exemplo, para aplicação de forças de volume).

O pacote `geo` contém as funções e classes para definição do modelo geométrico.<sup>9</sup> Para cada entidade do modelo há uma classe de objetos que a representa. Assim, a classe `geo.Point` representa pontos, a classe abstrata `geo.Segment` representa um segmento genérico (da qual derivam as classes concretas `geo.Line` e `geo.Arc`), e as classes `geo.Loop` e `geo.Surface` representam laços e superfícies, respectivamente. Instâncias de tais classes são mantidas em um contêiner da classe `geo.Geo`, a qual declara métodos que permitem a criação de pontos, segmentos, laços e superfícies. Comandos escritos a partir de tais métodos podem ser agrupados em um *script* Matlab, como exemplificado a seguir.

Para definir a superfície da Figura 6.10, o primeiro passo é criar um objeto do tipo `geo.Geo`:

<sup>9</sup>Admitiremos que o leitor tenha conhecimentos básicos de orientação a objetos e Matlab.



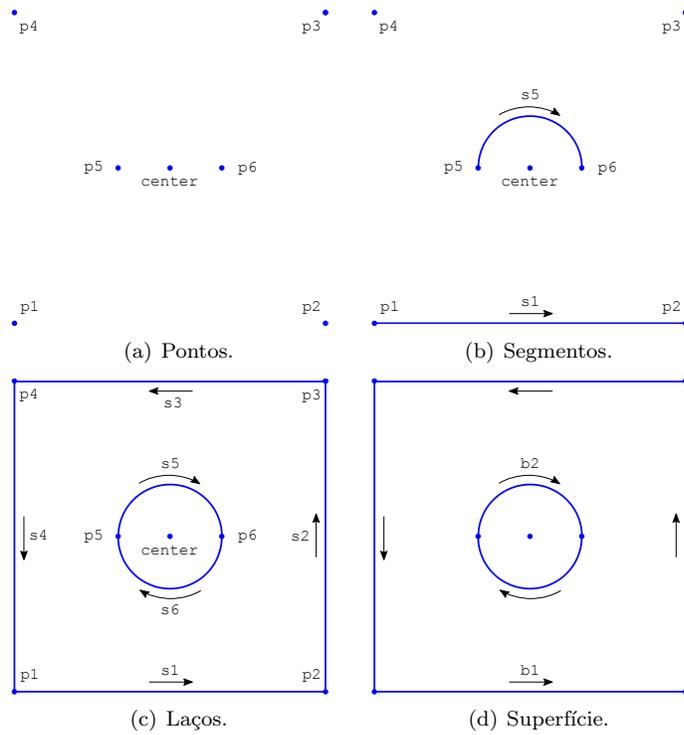
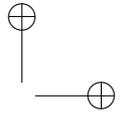
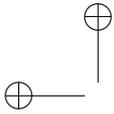


Figura 6.10: Modelo geométrico de uma superfície.

```
1 > g = geo.Geo(graphics.openFigure);
```

O comando acima instancia uma nova figura, devolvida pela função `graphics.openFigure`, para exibição dos elementos da superfície a ser criada. Em seguida, podemos definir os pontos, mostrados na Figura 6.10(a), através dos seguintes comandos:

```
2 > p1 = g.point(-3, -3);
3 > p2 = g.point(+3, -3);
4 > p3 = g.point(+3, +3);
5 > p4 = g.point(-3, +3);
6 > center = g.point(0, 0);
```



```
7 > p5 = g.point(-1, 0);
8 > p6 = g.point(+1, 0);
```

O método `geo.Geo.point(x, y)` toma como argumentos as coordenadas e devolve como saída o identificador do ponto criado. Com os identificadores dos pontos, podemos criar segmentos de linha e de arco. Os segmentos da Figura 6.10(c) são criados com

```
9 > s1 = g.line(p1, p2);
10 > s2 = g.line(p2, p3);
11 > s3 = g.line(p3, p4);
12 > s4 = g.line(p4, p5);
13 > s5 = g.arc(center, p5, p6, false);
14 > s6 = g.arc(center, p6, p5, false);
```

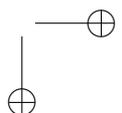
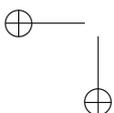
O método `geo.Geo.line(p1, p2)` toma como argumentos os identificadores dos pontos inicial e final e devolve como saída o identificador da linha criada. O método `geo.Geo.arc(c, p1, p2, ccw)` toma como argumentos os identificadores do centro e dos pontos inicial e final e, opcionalmente, um valor lógico que indica se a orientação do arco é no sentido anti-horário, e devolve o identificador do arco criado. Se omitido, o valor assumido para o argumento `ccw` é verdadeiro (nesse exemplo, portanto, os arcos têm sentido horário). A partir dos identificadores dos segmentos, podemos criar os laços externo e interno:

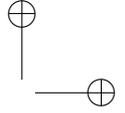
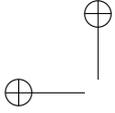
```
15 > b1 = g.loop([s1 s2 s3 s4]);
16 > b2 = g.loop([s5 s6]);
```

O método `geo.Geo.loop(s)` toma como argumento um vetor com os identificadores dos segmentos e devolve como saída o identificador do laço criado. Note que os laços `b1` e `b2` têm orientações distintas, o primeiro com os pontos no sentido anti-horário e o segundo com os pontos no sentido horário. Com os identificadores dos laços, podemos finalmente criar a superfície da Figura 6.10(d):

```
17 > surface = g.surface([b1 b2]);
```

O método `geo.Geo.surface(b)` toma como argumento um vetor com os identificadores dos laços (o primeiro elemento do vetor identificando o contorno externo) e devolve como saída o objeto da classe `geo.Surface` criado.





Definida a superfície, podemos especificar seu material:

```
18 > surface.material = fem.Material(100000, 0.3, 1);
```

A classe `fem.Material` representa um material elástico com três propriedades: módulo de elasticidade,  $E$ , coeficiente de Poisson,  $\nu$ , e densidade,  $\rho$  (usada para cálculo do peso). Nesse exemplo, o material tem  $E = 100000$  (unidades de pressão),  $\nu = 0.3$  e  $\rho = 1$  (unidades de massa por unidade de volume).

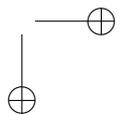
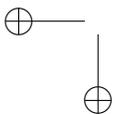
Também podemos definir, no modelo geométrico, vínculos e carregamentos uniformemente distribuídos ao longo de segmentos dos contornos da superfície, os quais são representados, respectivamente, pelas classes `geo.Constraint` e `geo.Load`, ambas derivadas da classe abstrata `geo.BC`. Por exemplo,

```
19 > c = g.constraint(s1, [1 2]);
20 > f = g.load(s3, 0, -10);
```

O método `geo.Geo.constraint(s, d, u)` toma como argumentos o identificador do segmento, um vetor com os números dos graus de liberdade restringido, e o valor do deslocamento prescrito para todos os pontos do segmento, e devolve como saída o identificador da restrição criada. O método `geo.Geo.load(s, t1, t2)` toma como argumentos o identificador do segmento e as forças aplicadas nos pontos inicial e final do segmento, e devolve como saída o identificador do carregamento criado. Nesse exemplo, os componentes de deslocamento nas direções  $x$  e  $y$  de todos os pontos do segmento `s1` têm valor prescrito igual a zero (valor assumido se o argumento `u` não é especificado), e o segmento `s3` é submetido a um carregamento uniforme constante de 10 (unidades de força por unidade de comprimento) na direção  $-y$ . Segmentos restringidos são marcados com um traço pontilhado em vermelho, enquanto segmentos com carregamentos são marcados com um traço pontilhado em verde, como ilustrado na Figura 6.11.

### Geração da malha

O pacote `mesh` contém as funções e classes de objetos para geração e visualização da malha, entre outras. Uma malha é um objeto da classe `mesh.Mesh`, caracterizado por um vetor de vértices e um vetor de faces triangulares. Um vértice é definido pelas suas coordenadas



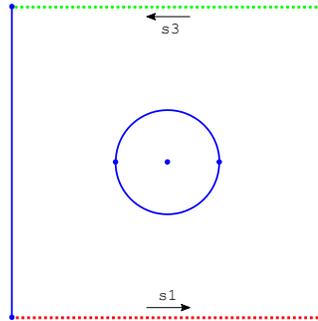


Figura 6.11: Modelo geométrico com condições de contorno.

cartesianas  $x$  e  $y$ , e uma face pela sua *lista de incidência*, isto é, o vetor com os índices dos (três) vértices da face. Essa estrutura, embora a mais simples possível, é suficiente para nossos propósitos porque não necessitaremos efetuar, sobre a malha gerada, qualquer operação cuja eficiência depende da representação explícita de relações de adjacência mais complexas.

Para a geração da malha, usamos um programa escrito em C chamado `triangle`, desenvolvido por Jonathan Richard Shewchuk e disponível em [www.cs.cmu.edu/~quake/triangle.html](http://www.cs.cmu.edu/~quake/triangle.html). Esse programa implementa métodos próprios de triangulação de uma superfície planar definida por pontos e segmentos de linha (que é exatamente como definimos nossas superfícies, conforme descrito anteriormente), mas também o algoritmo de Ruppert apresentado no Capítulo 5. Para gerar o arquivo de dados de entrada de `triangle` para a superfície da Figura 6.10, usamos a função

```
21 > geo.writeSurface(surface, 'example1', 0.2);
```

Os argumentos são um objeto do tipo `geo.Surface`, o nome do arquivo a ser gerado (ao qual é acrescido o sufixo `.poly`) e, opcionalmente, um número real `h` (igual a 0.2 nesse exemplo). Uma vez que a superfície de entrada do algoritmo de Ruppert implementado em `triangle` é dada por segmentos de linha, todos os segmentos de arco de `surface` devem ser discretizados em uma polilinha. Embora não por esse motivo, subdividimos também todos os segmentos de linha,

como ilustrado na Figura 6.12. O argumento `h` define o comprimento mínimo entre dois pontos quaisquer resultantes da subdivisão dos segmentos da superfície, e também o *comprimento mínimo das arestas* resultantes da triangulação. Se o argumento for omitido, o valor assumido é `surface.minLength`, igual ao comprimento do menor segmento da superfície.

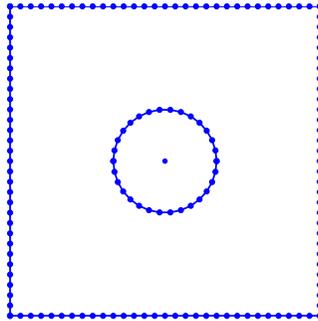


Figura 6.12: Subdivisão dos segmentos do modelo geométrico.

Uma vez gerado o arquivo de entrada, executamos `triangle` a partir do Matlab usando a função homônima

```
22 > triangle('example1');
```

A função tem como argumento o nome do arquivo de dados e, opcionalmente, o valor do ângulo mínimo da triangulação (se omitido, como nesse exemplo, o valor assumido é igual a  $20.7^\circ$ , o maior ângulo mínimo admitido para o algoritmo de Ruppert, conforme definido no Capítulo 5).

A saída de `triangle` são dois arquivos, um com a descrição dos vértices e outro com a descrição das faces da malha gerada. Os dados desses arquivos são lidos e usados para criação de um objeto da classe `mesh.Mesh`, o que é feito com a função

```
23 > m = mesh.readMesh('example1');
```

Para visualizar a malha gerada em uma nova figura, usamos

```
24 > mr = m.render(graphics.openFigure);
```

O resultado é mostrado na Figura 6.13(a). O método toma com argumento uma figura e devolve um objeto da classe `mesh.Renderer`, com o qual podemos controlar propriedades do desenho da malha, tais como cor e forma dos vértices, cor e traço das arestas e cor das faces. Para obtermos o desenho da Figura 6.13(b), por exemplo, usamos

```
25 > mr.showVertices(false);
26 > mr.setEdgeColor('black');
27 > mr.setFaceColor([0.8 0.9 1]);
```

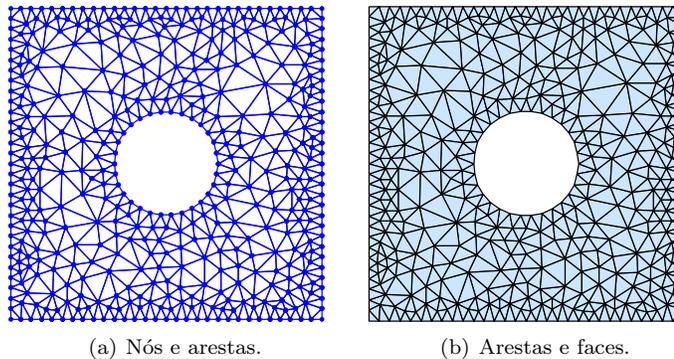
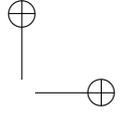
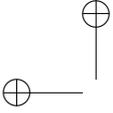


Figura 6.13: Malha gerada.

Note que, até o momento, não consideramos quaisquer atributos específicos da aplicação, tais como deslocamentos e forças nodais equivalentes, ou tensões e deformações. Apesar de termos definido o material da superfície, bem como vínculos e carregamentos em seu contorno, a geração da malha, baseada exclusivamente no algoritmo de Ruppert, não carrega tais dados em sua estrutura. Em outras palavras, a malha ainda não representa uma discretização do domínio e do contorno da superfície em elementos finitos, isto é, seus vértices ainda não são nós e suas faces ainda não são elementos finitos.

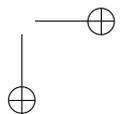
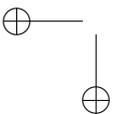
Uma alternativa seria escrevermos uma classe genérica para malhas em que poderíamos ter um número qualquer de atributos quaisquer associados a vértices e/ou faces. Opcionalmente, poderíamos ter uma classe base simples, como a adotada aqui, da qual derivaríamos classes com atributos e métodos particulares de determi-

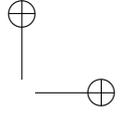
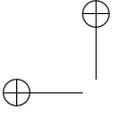


nada aplicação, o que implicaria em classes de malhas distintas para atributos distintos. Além disso, não queremos alterar o programa de geração de malhas para que este produza malhas com atributos próprios de uma aplicação particular; ao contrário, queremos que o programa possa ser usado em o maior número possível de aplicações em que malhas de triângulos são requeridas. Qualquer aplicação em que se deseja visualizar uma superfície qualquer, por exemplo, pode se valer de uma estrutura simples de malha de triângulos da superfície, uma vez que bibliotecas gráficas, tais como a OpenGL, podem desenhar malhas de triângulos muito eficientemente. De qualquer forma, **triangle** permite que associemos a pontos e segmentos de linha, dados como entrada do programa, números inteiros que Shewchuk denominou de *marcas de contorno*. Se um segmento de linha reta definido por dois pontos  $p$  e  $q$  tem marca  $m$ , então todos as arestas da triangulação que estão sobre o segmento  $\overline{pq}$  também terão a mesma marca  $m$ . Se relacionarmos a marca  $m$  a uma determinada condição de contorno, por exemplo, um carregamento sobre  $\overline{pq}$ , então podemos associar a condição de contorno a todas as arestas com a marca  $m$ . Contudo, uma vez que podemos substituir **triangle** por outro gerador de malhas, adotamos uma estratégia distinta de construção de domínio, descrita a seguir.

### Construção do domínio

O pacote **fem** contém as funções e classes de objetos para construção, análise por elementos finitos e visualização de domínios. Um domínio é um objeto da classe **fem.Domain**, um contêiner de material, nós e elementos, cuja geometria é definida por uma malha a ele associada. Um material (elástico) é um objeto da classe **fem.Material**, cujas propriedades já descrevemos anteriormente quando especificamos, na linha 18, o material da superfície da Figura 6.10. Um nó é um objeto da classe **fem.Node**, cujas propriedades são o identificador do vértice da malha associado ao nó e vetores com os deslocamentos e as forças equivalentes na direção de cada grau de liberdade. (Para o estado plano de tensões, um nó carrega os deslocamentos  $u_1$  e  $u_2$  e forças  $f_1$  e  $f_2$ , conforme descrito na Seção 6.3.) A classe abstrata **fem.FiniteElement** representa um elemento finito genérico ao qual é associada, no caso 2D, uma face da malha. A classe declara métodos



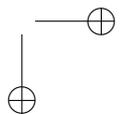
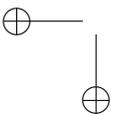


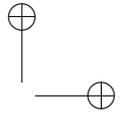
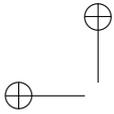
abstratos para computação da matriz de rigidez,  $\mathbf{K}^{(e)}$ , vetor de carregamentos nodais equivalentes,  $\mathbf{F}^{(e)}$ , e vetor de números das equações dos graus de liberdade,  $\mathbf{R}^{(e)}$  (usado na montagem do sistema linear de acordo com o Algoritmo 6.3), de um elemento finito. A classe `fem.CST`, derivada de `fem.FiniteElement`, representa um elemento finito do tipo CST, cuja formulação foi apresentada na Seção 6.5.

Com esse esquema, ao invés de adicionarmos à malha os atributos próprios da aplicação, construímos, a partir dela, um objeto distinto (o domínio) contendo objetos (material, nós, elementos) que contêm tais atributos, os quais fornecem todos os métodos necessários para manipulação dos atributos. Como consequência, a estrutura de dados da malha, embora relacionada, é independente da estrutura do modelo discreto de elementos finitos. Tudo que precisamos de uma malha, a fim de construirmos um domínio, são métodos para obtermos as coordenadas de seus vértices e a conectividade de suas faces.

Um domínio contém também um vetor gravidade, cujo valor inicial é  $(0, -9.8)$  unidades de aceleração, e coleções de restrições e de carregamentos nodais. Uma restrição a um grau de liberdade nodal, isto é, uma condição de contorno essencial, é um objeto da classe `fem.Constraint`, caracterizado pelo número do nó no qual a restrição é aplicada, o número do grau de liberdade, e o valor do deslocamento prescrito (ou seja, o objeto é um elemento do vetor  $\mathbf{V}$ , usado no Algoritmo 6.2). Embora possamos definir restrições diretamente sobre os nós de contorno do domínio, estas serão geradas mais adequadamente a partir das restrições aplicadas aos segmentos do modelo geométrico, como feito na linha 19. Nesse exemplo, todos os nós do contorno do domínio, correspondentes aos vértices da malha sobre o segmento `s1` do modelo geométrico, terão os deslocamentos nas direções 1 e 2 (ou  $x$  e  $y$ ) restringidos e iguais a zero.

Um carregamento nodal é um objeto da classe `fem.PointLoad`, caracterizado pelo número do nó em que é aplicado e pelos componentes do carregamento. Novamente, embora possamos aplicá-los diretamente em nós do contorno do domínio, os carregamentos nodais serão derivados daqueles especificados sobre segmentos do modelo geométrico, como feito na linha 20. Nesse exemplo, todos os nós do contorno do domínio, correspondentes aos vértices da malha sobre o segmento `s3` do modelo geométrico, estarão sujeitos a uma carga uniformemente distribuída constante de  $(0, -10)$  unidades de força





por unidade de comprimento. O elemento finito CST usa esses valores para cálculo do termo do vetor de esforços nodais equivalentes correspondentes às forças de superfície, dado pela Equação (6.62). Para o termo correspondente a forças de volume constantes, dado pela Equação (6.59), consideramos o peso do elemento, calculado em função da densidade do material e do vetor gravidade do domínio.

Para construirmos o domínio, portanto, precisamos da malha e das informações a respeito das condições de contorno aplicadas ao modelo geométrico. A primeira é fornecida por `triangle`, e as últimas pela função `geo.writeSurface()`, usada na linha 21. Esta gera, além do arquivo de entrada de `triangle`, um arquivo com os dados de material e das restrições e carregamentos associados a cada vértice da malha gerado sobre um segmento da superfície sobre o qual foi especificado um vínculo ou um carregamento. Assim, se especificarmos que um segmento do modelo geométrico tem um determinado vínculo, por exemplo, então todos os pontos resultantes da subdivisão do segmento em polilinha (pontos esses que serão vértices na triangulação) também terão seus graus de liberdade restringidos conforme o vínculo. Para gerarmos o domínio da malha da Figura 6.13, usamos a função

```
28 > d = fem.readDomain(m, 'example1');
```

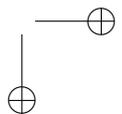
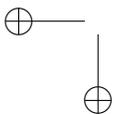
Os argumentos são um objeto da classe `mesh.Mesh` e o nome do arquivo com os dados de material, restrições e carregamentos. A função cria um domínio com um elemento finito da classe `fem.CST` para cada triângulo da malha `m`.

### Análise e visualização dos resultados

Construído o domínio, podemos efetuar a análise estática usando

```
29 > fem.staticAnalysis(d);
```

A função monta o sistema linear (6.40) exatamente como descrito nos Algoritmos 6.2 e 6.3, tomando as contribuições dos elementos CST exatamente de acordo com as equações da Seção 6.5. A solução do sistema fornece os deslocamentos incógnitos, os quais são adicionados ao vetor de deslocamentos globais  $\mathbf{U}$  do domínio `d` exatamente conforme o Algoritmo 6.4.



Para visualizar o campo de deslocamentos sobre o domínio em uma nova figura, usamos

```
30 > dr = d.render(graphics.openFigure);
```

O resultado é mostrado na Figura 6.14. O método toma como argumento uma figura e devolve um objeto da classe `fem.Renderer`, com o qual podemos controlar propriedades do desenho do domínio, tais como cor e forma dos nós, cor e traço das arestas e cor dos elementos, bem como a visualização de mapas de cores e da estrutura deformada. Inicialmente, o desenho apresenta os nós e arestas dos elementos do domínio, bem como o mapa de cores de  $u_1$  (produzido com as funções gráficas de desenho de malhas do Matlab).

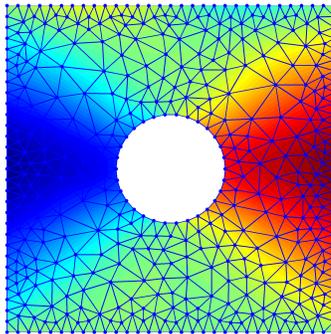


Figura 6.14: Domínio e mapa de cores de  $u_1$ .

Podemos mostrar os resultados da análise sobre o modelo geométrico da superfície. Para isso, ao invés de uma nova figura, vamos desenhar o domínio sobre a figura do modelo geométrico. A Figura 6.15(a) é obtida com os seguintes comandos:

```
31 > g.showPoints(false);
32 > g.showBCs(false);
33 > gdr = d.render(g.axes);
34 > gdr.showNodes(false);
35 > gdr.showEdges(false);
36 > gdr.showColorbar;
```

Primeiro, eliminamos os pontos e as condições de contorno do desenho da geometria. Em seguida, usamos o comando da linha 38 para obter um objeto `fem.Renderer` para `d`, `gdr`, que usa a mesma figura de `g`, dada pela propriedade `g.axes`. Por fim, usamos `gdr` para eliminar os nós e arestas e adicionar uma barra de cores ao desenho do domínio, com a qual podemos verificar que os deslocamentos na direção  $x$  estão entre  $-0.002$  e  $0.002$ . A Figura 6.15(b) é obtida com

```
37 > gdr.showColorMap('u', 2);
```

O método exibe o mapa de cores de  $u_2$ , cujos valores estão entre  $-0.01$  e  $0$ . Note que, na parte inferior do contorno, correspondente ao segmento `s1`, os deslocamentos são nulos.

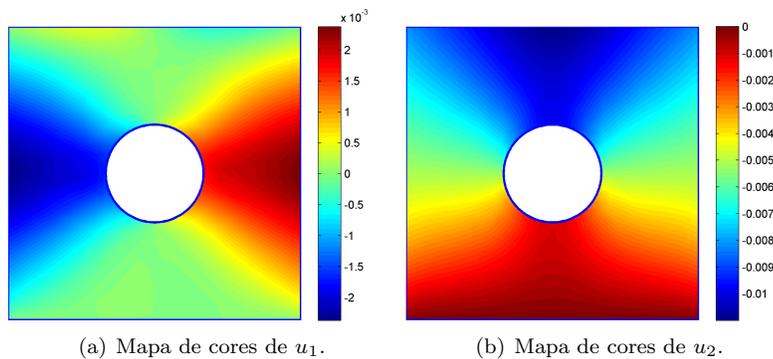


Figura 6.15: Malha de cores sobre o modelo geométrico.

Para visualizar a estrutura deformada, usamos

```
38 > dr.setFaceColor([0.8 0.8 1]);
39 > dr.setWarpScale(500);
```

O método `fem.Renderer.setWarpScale(s)` toma como argumento um fator de escala, igual a 500, nesse exemplo, e desenha o domínio na configuração deformada, como ilustrado na Figura 6.16. Nessa configuração, as coordenadas de cada nó são somadas ao seu deslocamento multiplicado por `s`.

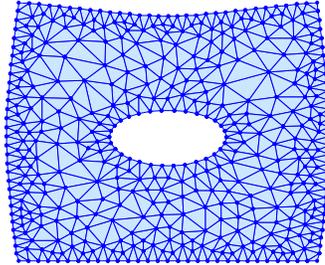
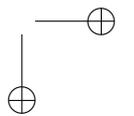
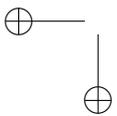
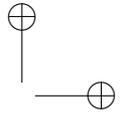
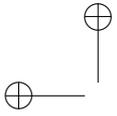
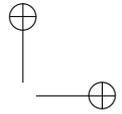
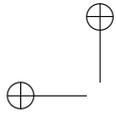


Figura 6.16: Estrutura deformada.

## 6.7 Notas bibliográficas

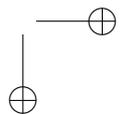
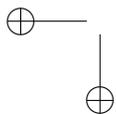
O conteúdo abordado neste capítulo envolve disciplinas que, isoladamente, poderiam ser cobertas em um semestre de um curso de graduação e/ou pós-graduação em matemática aplicada ou engenharia. Dado o enfoque do livro, nosso objetivo foi introduzir os principais conceitos necessários à descrição do modelo matemático de um problema simples de mecânica do contínuo 2D, bem como à formulação do MEF em uma aplicação prática ilustrativa do método de geração de malhas apresentado no Capítulo 5. Parte do resumo de mecânica do contínuo apresentado na Seção 6.2 foi baseada no texto de Malvern [80]. Na Seção 6.3, usamos principalmente a última edição dos “clássicos” sobre elementos finitos de Zienkiewicz e outros [141, 139]. Sobre resíduos ponderados, consultados também o livro de elementos de contorno de Brebbia e outros [19].





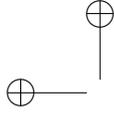
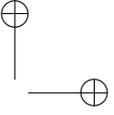
## Bibliografia

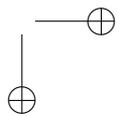
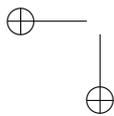
- [1] O. Aichholzer, F. Aurenhammer, H. Krasser, and F. Hurtado. Towards compatible triangulations. In J. Wang, editor, *Computing and Combinatorics*, volume 2108 of *Lecture Notes in Computer Science*, pages 101–110. Springer Berlin Heidelberg, 2001.
- [2] O. Aichholzer, W. Mulzer, and A. Pilz. Flip distance between triangulations of a simple polygon is NP-complete. In H. L. Bodlaender and G. F. Italiano, editors, *Algorithms – ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pages 13–24. Springer Berlin Heidelberg, 2013.
- [3] F. B. Atalay, S. Ramaswami, and D. Xu. Quadrilateral meshes with provable angle bounds. *Engineering with Computers*, 28(1):31–56, 2012.
- [4] F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi diagrams and Delaunay triangulations*. World Scientific, 2013.
- [5] I. Babuška and A. K. Aziz. On the angle condition in the finite element method. *SIAM Journal on Numerical Analysis*, 13(2):214–226, 1976.
- [6] B. S. Baker, E. Grosse, and C. S. Rafferty. Nonobtuse triangulation of polygons. *Discrete & Computational Geometry*, 3(1):147–168, 1988.
- [7] J. Barbic and G. Miller. A quadratic running time example for Ruppert’s refinement algorithm. Technical Report 12-925,



- Computer Science Department, University of Southern California, Los Angeles, CA, USA, 2012.
- [8] J. L. M. Barbosa. *Geometria Euclidiana Plana*. Coleção do Professor de Matemática. SBM, nona edition, 2006.
  - [9] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series in Computing*, pages 23–90. World Scientific, 1992.
  - [10] M. Bern and D. Eppstein. Quadrilateral meshing by circle packing. In *Proceedings of the 6th International Meshing Roundtable*, pages 7–19, Park City, Utah, USA, October 13-15 1997.
  - [11] M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *Journal of Computer and System Sciences*, 48(3):384–409, 1994.
  - [12] M. Bern and P. Plassmann. Mesh generation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*. Elsevier Science, 2000.
  - [13] J.-D. Boissonnat, O. Devillers, S. Pion, M. Teillaud, and M. Yvinec. Triangulations in {CGAL}. *Computational Geometry*, 22(1-3):5–19, 2002. 16th {ACM} Symposium on Computational Geometry.
  - [14] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998.
  - [15] C. Boivin and C. Ollivier-Gooch. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *International Journal for Numerical Methods in Engineering*, 55(10):1185–1213, 2002.
  - [16] P. Bose and F. Hurtado. Flips in planar graphs. *Computational Geometry: Theory and Applications*, 42(1):60–80, 2009.
  - [17] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. A. K. Peters, Ltd., 2010.

- [18] A. Bowyer. Computing dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.
- [19] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary Element Techniques: Theory and Applications in Engineering*. Springer-Verlag, 1984.
- [20] S.-W. Cheng, T. K. Dey, E. Ramos, and T. Ray. Quality meshing for polyhedra with small angles. In *Proceedings of the 20th Annual ACM Symposium on Computational Geometry*, SoCG '04, pages 290–299, New York, NY, USA, June 9-11 2004. ACM.
- [21] S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. *Discrete & Computational Geometry*, 43(1):121–166, 2010.
- [22] S.-W. Cheng, T. K. Dey, and T. Ray. Weighted Delaunay refinement for polyhedra with small angles. In *Proceedings of the 14th International Meshing Roundtable*, IMR '14, pages 325–342, San Diego, California, USA, September 11-14 2005.
- [23] S.-W. Cheng, T. K. Dey, and J. R. Shewchuk. *Delaunay Mesh Generation*. CRC Press, 2013.
- [24] L. P. Chew. Constrained Delaunay triangulations. *Algorithmica*, 4(1-4):97–108, 1989.
- [25] L. P. Chew. Guaranteed-quality triangular meshes. Technical Report 89-983, Computer Science Department, Cornell University, Ithaca, NY, USA, 1989.
- [26] F. U. Coelho and M. L. Lourenço. *Um Curso de Álgebra Linear*. Edusp, 2010.
- [27] D. Cohen-Steiner, É. C. de Verdière, and M. Yvinec. Conforming Delaunay triangulations in 3d. *Computational Geometry: Theory and Applications*, 28(2-3):217–233, 2004. Special Issue on the 18th Annual ACM Symposium on Computational Geometry.

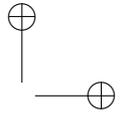
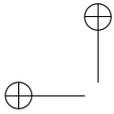
- 
- 
- [28] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Algoritmos: Teoria e Prática*. Elsevier Acadêmico, terceira edição, 2012.
- [29] E. F. D’Azevedo and R. B. Simpson. On optimal interpolation triangle incidences. *SIAM Journal on Scientific and Statistical Computing*, 10(6):1063–1075, 1989.
- [30] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, terceira edição, 2008.
- [31] P. M. M. de Castro and O. Devillers. Practical distribution-sensitive point location in triangulations. *Computer Aided Geometric Design*, 30(5):431–450, 2013.
- [32] L. H. de Figueiredo e Paulo César Pinto Carvalho. *Introdução à Geometria Computacional*. IMPA, Rio de Janeiro, RJ, Brazil, 1991.
- [33] J. A. De Loera, J. Rambau, and F. Santos. *Triangulations: Structures for Algorithms and Applications*. Algorithms and Computation in Mathematics. Springer, 2010.
- [34] S. L. Devadoss and J. O’Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.
- [35] O. Devillers, S. Pion, and M. Teillaud. Walking in a triangulation. *International Journal of Foundations of Computer Science*, 13(2):181–199, 2002.
- [36] T. K. Dey, C. L. Bajaj, and K. Sugihara. On good triangulations in three dimensions. In *Proceedings of the First ACM Symposium on Solid Modeling Foundations and CAD/CAM Applications*, SMA ’91, pages 431–441, New York, NY, USA, 1991. ACM.
- [37] T. K. Dey and J. A. Levine. Delaunay meshing of piecewise smooth complexes without expensive predicates. *Algorithms*, 2(4):1327–1349, 2009.



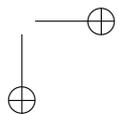
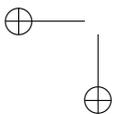
- [38] Q. Du and D. Wang. Boundary recovery for three dimensional conforming Delaunay triangulation. *Computer Methods in Applied Mechanics and Engineering*, 193(23-26):2547–2563, 2004.
- [39] Q. Du, D. Wang, and L. Zhu. On mesh geometry and stiffness matrix conditioning for general finite element spaces. *SIAM Journal on Numerical Analysis*, 47(2):1421–1444, 2009.
- [40] R. A. Dwyer. A faster divide-and-conquer algorithm for constructing Delaunay triangulations. *Algorithmica*, 2(1-4):137–151, 1987.
- [41] N. Dyn, I. Goren, and S. Rippa. Transforming triangulations in polygonal domains. *Computer Aided Geometric Design*, 10(6):531–536, 1993.
- [42] H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge University Press, 2001.
- [43] D. Eppstein. Diamond-kite meshes: Adaptive quadrilateral meshing and orthogonal circle packing. In X. Jiao and J.-C. Weill, editors, *Proceedings of the 21st International Meshing Roundtable*, pages 261–277. Springer Berlin Heidelberg, 2013.
- [44] J. Erickson. Efficiently hex-meshing things with topology. *Discrete & Computational Geometry*, 52(3):427–449, 2014.
- [45] K. Eriksson, D. Estep, and C. Jonhson. *Applied Mathematics: Body and Soul*, volume III. Springer, 2004.
- [46] H. Everett, W. J. Lenhart, M. Overmars, T. C. Shermer, and J. Urrutia. Strictly convex quadrilateralizations of polygons. In *Proceedings of the 4th Canadian Conference on Computational Geometry*, pages 77–82, St. John’s, Newfoundland, Canada, 1992.
- [47] C. E. Ferreira and Y. Wakabayashi. *Combinatória Poliédrica e Planos-de-Corte Faciais*. 10<sup>a</sup> Escola de Computação. Unicamp, 8 a 13 de junho de 1996, Campinas, SP, Brasil, 1996.
- [48] S. Fortune. A note on Delaunay diagonal flips. *Pattern Recognition Letters*, 14(9):723–726, 1993. Computational Geometry.

- [49] S. Fortune. Numerical stability of algorithms for 2d Delaunay triangulations. *International Journal of Computational Geometry & Applications*, 5(1-2):193–213, 1995.
- [50] P. J. Frey and P.-L. George. *Mesh Generation: Application to Finite Elements*. ISTE, second edition, 2010.
- [51] J. Gallier. *Curves and Surfaces in Geometric Modeling: Theory and Algorithms*. Morgan-Kaufmann, 2000.
- [52] J. Gallier. *Geometric Methods and Applications for Computer Science and Engineering*. Springer, segunda edition, 2011.
- [53] G. G. Garbi. *C.Q.D.* Livraria da Física, 2010.
- [54] M. R. Garey and D. S. Johnson. *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [55] P.-L. George and H. Borouchaki. *Delaunay Triangulation and Meshing*. Hermes, 1998.
- [56] R. L. Graham. An efficient algorithm for determining the convex hull of a planar set. *Information Processing Letters*, 1:132–133, 1972.
- [57] P. J. Green and R. Sibson. Computing Dirichlet tessellations in the plane. *The Computer Journal*, 21(2):168–173, 1978.
- [58] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.
- [59] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7(1-6):381–413, 1992.
- [60] E. Hale and A. Üngör. Quality triangulations with locally optimal steiner points. *SIAM Journal on Scientific Computing*, 31(3):2103–2130, 2009.

- [61] P. C. Hammer, O. J. Marlowe, and A. H. Stroud. Numerical integration over simplexes and cones. *Math. Tables Aids Comput.*, 10:130–137, 1956.
- [62] S. Hanke, T. Ottmann, and S. Schuierer. The edge-flipping distance of triangulations. *Journal of Universal Computer Science*, 2(8):570–579, August 1996.
- [63] S. Har-Peled and A. Üngör. A time-optimal delaunay refinement algorithm in two dimensions. In *Proceedings of the Twenty-first Annual Symposium on Computational Geometry*, SoCG '05, pages 228–236, New York, NY, USA, 2005. ACM.
- [64] Ø. Hjelle and M. Dæhlen. *Triangulations and Applications*. Springer, 2006.
- [65] C. M. Hoffmann, J. E. Hopcroft, and M. S. Karasick. Towards implementing robust geometric computations. In *Proceedings of the Fourth Annual Symposium on Computational Geometry*, SoCG '88, pages 106–117, New York, NY, USA, 1988. ACM.
- [66] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introdução à Teoria dos Automômatos, Linguagens e Computação*. Elsevier Acadêmico, 2002.
- [67] B. Hudson, G. Miller, and T. Phillips. Sparse voronoi refinement. In P. P. Pébay, editor, *Proceedings of the 15th International Meshing Roundtable*, pages 339–356. Springer Berlin Heidelberg, 2006.
- [68] L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap. Classroom examples of robustness problems in geometric computations. *Computational Geometry: Theory and Applications*, 40(1):61–78, 2008.
- [69] M. Kremer, D. Bommers, I. Lim, and L. Kobbelt. Advanced automatic hexahedral mesh generation from surface quad meshes. In J. Sarrate and M. Staten, editors, *Proceedings of the 22nd International Meshing Roundtable*, IMR '13, pages 147–164. Springer International Publishing, October 13-16 2013.



- [70] F. Labelle and J. R. Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Transactions on Graphics*, 26(3), July 2007.
- [71] M.-J. Lai and L. Schumaker. *Spline Functions on Triangulations*. Cambridge University Press, 2007.
- [72] E. Langetepe and G. Zachmann. *Geometric Data Structure for Computer Graphics*. A. K. Peters, 2006.
- [73] C. L. Lawson. Software for  $c^1$  interpolation. In J. Rice, editor, *Mathematical Software III*, pages 161–194. Academic Press, 1977.
- [74] C. Li, S. Pion, and C. K. Yap. Recent progress in exact geometric computation. *The Journal of Logic and Algebraic Programming*, 64(1):85–111, 2005. Practical development of exact real number computation.
- [75] X. Liang, M. S. Ebeida, and Y. Zhang. Guaranteed-quality all-quadrilateral mesh generation with feature preservation. *Computer Methods in Applied Mechanics and Engineering*, 199(29-32):2072–2083, 2010.
- [76] D. Lischinski. Graphics gems iv. chapter Incremental Delaunay triangulation, pages 47–59. Academic Press Professional, Inc., San Diego, CA, USA, 1994.
- [77] V. D. Liseikin. *Grid Generation Methods*. Springer, second edition, 2010.
- [78] D. S. H. Lo. *Finite Element Mesh Generation*. CRC Press, 2015.
- [79] A. Lubiw and V. Pathak. Flip distance between two triangulations of a point-set is NP-complete. *CoRR*, abs/1205.2425, 2012.
- [80] L. Malvern. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall, 1969.



- [81] S. Mao and Z. Shi. Error estimates of triangular finite elements under a weak angle condition. *Journal of Computational and Applied Mathematics*, 230(1):329–331, 2009.
- [82] E. A. Melissaratos and D. L. Souvaine. Coping with inconsistencies: A new approach to produce quality triangulations of polygonal domains with holes. In *Proceedings of the 8th Annual ACM Symposium on Computational Geometry*, SoCG '92, pages 202–211, New York, NY, USA, 1992. ACM.
- [83] G. Miller. A time efficient Delaunay refinement algorithm. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '04, pages 400–409, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.
- [84] G. Miller, S. Pav, and N. J. Walkington. When and why Ruppert’s algorithm works. In *Proceedings of the 12th International Meshing Roundtable*, pages 91–102, Santa Fe, New Mexico, USA, September 14-17 2003.
- [85] G. Miller, D. Talmor, S.-H. Teng, and N. J. Walkington. A Delaunay based numerical method for three dimensions: Generation, formulation, and partition. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computation*, STOC '95, pages 683–692, New York, NY, USA, May 29 - June 01 1995. ACM.
- [86] G. L. Miller, T. Phillips, and D. Sheehy. Size competitive meshing without large angles. In L. Arge, C. Cachin, T. Jurdziński, and A. Tarlecki, editors, *Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 655–666. Springer Berlin Heidelberg, 2007.
- [87] G. L. Miller, T. Phillips, and D. R. Sheehy. Linear-size meshes. In *Proceedings of the 20th Canadian Conference in Computational Geometry*, CCCG '08, pages 175–178, August 13-15 2008.
- [88] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the 8th Annual ACM*

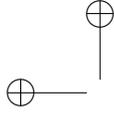
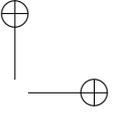
- Symposium on Computational Geometry*, pages 212–221, New York, NY, USA, June 10–12 1992. ACM.
- [89] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.
- [90] W. Morris and V. Soltan. The Erdős-Szekeres problem on points in convex position – a survey. *Bulletin of the American Mathematical Society*, 37(4):437–458, 2000.
- [91] E. P. Mücke, I. Saias, and B. Zhu. Fast randomized point location without preprocessing in two- and three-dimensional Delaunay triangulations. *Computational Geometry: Theory and Applications*, 12(1-2):63–83, 1999.
- [92] M. Müller-Hannemann and K. Weihe. Minimum strictly convex quadrangulations of convex polygons. In *Proceedings of the 13th ACM Symposium on Computational Geometry*, pages 193–202, Centre Universitaire Méditerranéen, Nice, France, June 4–6 1997.
- [93] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Pearson, 1993.
- [94] M. Murphy, D. M. Mount, and C. W. Gable. A point-placement strategy for conforming Delaunay tetrahedralization. *International Journal of Computational Geometry & Applications*, 11(6):669–682, 2001.
- [95] O. R. Musin. Properties of the Delaunay triangulation. In *Proceedings of the Thirteenth Annual Symposium on Computational Geometry*, SoCG '97, pages 424–426, New York, NY, USA, 1997. ACM.
- [96] A. C. M. Neto. *Geometria*. Coleção PROFMAT. SBM, 2013.
- [97] J. O’Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

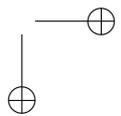
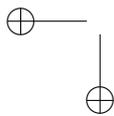
- [98] S. Oudot, L. Rineau, and M. Yvinec. Meshing volumes bounded by smooth surfaces. In B. W. Hanks, editor, *Proceedings of the 14th International Meshing Roundtable*, pages 203–220, September, 11-14 2005.
- [99] K. Ozaki, T. Ogita, and S. Oishi. A robust algorithm for geometric predicate by error-free determinant transformation. *Information and Computation*, 216(0):3–13, 2012. Special Issue: 8th Conference on Real Numbers and Computers.
- [100] S. Pav and N. J. Walkington. Robust three dimensional Delaunay refinement. In *Proceedings of the 13th International Meshing Roundtable*, IMR '13, pages 145–156, Williamsburg, VA, USA, September 19-22 2004.
- [101] S. E. Pav. *Delaunay refinement algorithms*. PhD thesis, Carnegie Mellon University, Pittsburg, PA, USA, 2003.
- [102] S. E. Pav and N. J. Walkington. Delaunay refinement by corner lopping. In *Proceedings of the 14th International Meshing Roundtable*, pages 165–181, San Diego, California, USA, September 11-14 2005.
- [103] B. Pellenard, P. Alliez, and J.-M. Morvan. Isotropic 2d quadrangle meshing with size and orientation control. In W. R. Quadros, editor, *Proceedings of the 20th International Meshing Roundtable*, pages 81–98. Springer Berlin Heidelberg, 2012.
- [104] H. Pirzadeh. Computational geometry with the rotating calipers. Master’s thesis, School of Computer Science, McGill University, Montréal, Québec, Canada, 1999.
- [105] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer, 1985.
- [106] S. Ramaswami, M. Siqueira, T. Sundaram, J. Gallier, and J. Gee. Constrained quadrilateral meshes of bounded size. *International Journal of Computational Geometry & Applications*, 15(1):55–98, 2005.

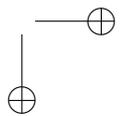
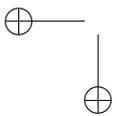
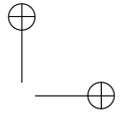
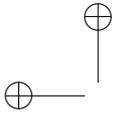
- [107] A. Rand. *Delaunay refinement algorithms for numerical methods*. PhD thesis, Carnegie Mellon University, Pittsburg, PA, USA, 2009.
- [108] A. Rand. Improved examples of non-termination for Ruppert’s algorithm. *CoRR*, abs/1103.3903, 2011.
- [109] A. Rand. On the non-termination of Ruppert’s algorithm. *CoRR*, abs/1101.1071, 2011.
- [110] A. Rand. Where and how Chew’s second Delaunay refinement algorithm works. In *Proceedings of the 23rd Canadian Conference on Computational Geometry*, Toronto, ON, Canada, August 10-12 2011.
- [111] A. Rand and N. Walkington. Collars and intestines: Practical conforming Delaunay refinement. In B. W. Clark, editor, *Proceedings of the 18th International Meshing Roundtable*, pages 481–497. Springer Berlin Heidelberg, 2009.
- [112] A. Rand and N. J. Walkington. Delaunay refinement for estimating local feature size in 2d and 3d. *International Journal of Computational Geometry & Applications*, 21(5):507–543, 2011.
- [113] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, May 1995.
- [114] A. Saalfeld. Joint triangulations and triangulation maps. In *Proceedings of the Third Annual Symposium on Computational Geometry*, SoCG ’87, pages 195–204, New York, NY, USA, 1987. ACM.
- [115] T. Schiffer, F. Aurenhammer, and M. Demuth. Computing convex quadrangulations. *Discrete Applied Mathematics*, 160(4-5):648–656, 2012.
- [116] S. Schirra. Robustness and precision issues in geometric computation. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 597–632. North-Holland, Amsterdam, 2000.

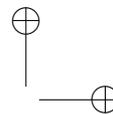
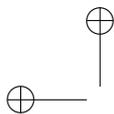
- [117] J. R. Shewchuk. Robust adaptive floating-point geometric predicates. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry*, SoCG'96, pages 141–150, New York, NY, USA, 1996. ACM.
- [118] J. R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and Delaunay triangulator. In *Selected Papers from the Workshop on Applied Computational Geometry, Towards Geometric Engineering*, FCRC '96/WACG '96, pages 203–222, London, UK, 1996. Springer-Verlag.
- [119] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proceedings of the 14th Annual ACM Symposium on Computational Geometry*, SCG '98, pages 86–95, New York, NY, USA, 1998. ACM.
- [120] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3):21–74, 2002.
- [121] J. R. Shewchuk. What is a good linear element? Interpolation, conditioning, and quality measures. In *Proceeding of the 11th International Meshing Roundtable*, pages 115–126, Ithaca, NY, USA, September 15-18 2002.
- [122] J. R. Shewchuk. Unstructured mesh generation. In U. Naumann and O. Schenk, editors, *Combinatorial Scientific Computing*, chapter 10, pages 257–297. CRC Press, 2012.
- [123] J. R. Shewchuk and B. C. Brown. Fast segment insertion and incremental construction of constrained Delaunay triangulations. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 299–308, New York, NY, USA, 2013. ACM.
- [124] J. R. Shewchuk and H. Si. Higher-quality tetrahedral mesh generation for domains with small angles by constrained Delaunay refinement. In *Proceedings of the 13th Annual Symposium on Computational Geometry*, SoCG '14, pages 290–299, New York, NY, USA, 2014. ACM.

- [125] H. Si. Constrained Delaunay tetrahedral mesh generation and refinement. *Finite Elements in Analysis and Design*, 46(1-2):33–46, 2010. Mesh Generation - Applications and Adaptation.
- [126] R. Sibson. Locally equiangular triangulations. *The Computer Journal*, 21(3):243–245, 1978.
- [127] R. Soukal, M. Málková, and I. Kolingerová. Walking algorithms for point location in TIN models. *Computational Geosciences*, 16(4):853–869, 2012.
- [128] P. Su and R. L. S. Drysdale. A comparison of sequential Delaunay triangulation algorithms. *Computational Geometry*, 7(5-6):361–385, 1997. 11th {ACM} Symposium on Computational Geometry.
- [129] S.-H. Teng and C. W. Wong. Unstructured mesh generation: Theory, practice, and perspectives. *International Journal of Computational Geometry & Applications*, 10(3):227–266, 2000.
- [130] J. Thurston. Looking back and again: the triangulated irregular network (TIN). *GEOinformatics*, 6(7):32–35, 2003.
- [131] B. Topping, J. Muylle, P. Iványi, R. Putanowicz, and B. Cheng. *Finite Element Mesh Generation*. Saxe-Coburg Publications, 2004.
- [132] V. J. D. Tsai. Delaunay triangulations in TIN creation: an overview and a linear-time algorithm. *International journal of geographical information systems*, 7(6):501–524, 1993.
- [133] A. Üngör. Off-centers: A new type of steiner points for computing size-optimal quality-guaranteed Delaunay triangulations. *Computational Geometry: Theory and Applications*, 42(2):109–118, 2009.
- [134] C. S. Verma and T. Tautges. Jaal: Engineering a high quality all-quadrilateral mesh generator. In W. R. Quadros, editor, *Proceedings of the 20th International Meshing Roundtable*, pages 511–530. Springer, 2012.

- 
- 
- [135] D. F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–172, 1981.
- [136] Y. Yokosuka and K. Imai. Guaranteed-quality anisotropic mesh generation for domains with curved boundaries. *Computer-Aided Design*, 41(5):385–393, 2009.
- [137] B. Zhu. On Lawson’s oriented walk in random Delaunay triangulations. In A. Lingas and B. J. Nilsson, editors, *Fundamentals of Computation Theory*, volume 2751 of *Lecture Notes in Computer Science*, pages 222–233. Springer Berlin Heidelberg, 2003.
- [138] G. M. Ziegler. *Lectures on Polytopes*. Springer, 1994.
- [139] O. Zienkiewicz, R. Talor, and D. Fox. *The Finite Element Method for Solid and Structural Mechanics*. Elsevier, seventh edition, 2014.
- [140] O. Zienkiewicz, R. Talor, and P. Nithiarasu. *The Finite Element Method for Fluid Dynamics*. Elsevier, seventh edition, 2014.
- [141] O. Zienkiewicz, R. Talor, and J. Zhu. *The Finite Element Method: Its Basis and Fundamentals*. Elsevier, seventh edition, 2013.
- [142] O. C. Zienkiewicz and J. Z. Zhu. Adaptivity and mesh generation. *International Journal for Numerical Methods in Engineering*, 32(4):783–810, 1991.





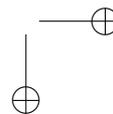
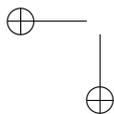


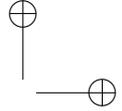
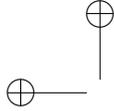
# Apêndice A

## Predicados geométricos

### A.1 Considerações iniciais

Os algoritmos que estudamos nos capítulos 3, 4 e 5 se valem de *testes geométricos*. Esses testes nos permitem verificar se um dado objeto geométrico goza de uma certa propriedade (geométrica). Por exemplo, o algoritmo que estudamos na Seção 3.4 para calcular o fecho convexo de um subconjunto finito de pontos em  $\mathbb{E}^2$  faz uso de um teste geométrico que determina se um dado ponto  $P \in \mathbb{E}^2$  está do lado “esquerdo” de uma dada reta orientada em  $\mathbb{E}^2$ . Testes geométricos como esse são denominados *predicados geométricos*, pois o resultado é sempre o valor lógico verdadeiro ou falso. Neste capítulo, introduzimos os principais predicados geométricos que utilizamos nos algoritmos dos capítulos 3, 4 e 5. Em particular, iniciamos nossa exposição com uma formalização na noção de *orientação* (Seção A.2), que é central para o entendimento de todos os predicados. Na Seção A.3, introduzimos o *teste de orientação*, que nos permite determinar se uma sequência de três (resp. quatro) pontos de  $\mathbb{E}^2$  (resp.  $\mathbb{E}^3$ ) estão orientados positiva ou negativamente. O teste de orientação é a base para os algoritmos que implementam os predicados *left*, *lefton* e *collinear* (Seção A.4), *classify* (Seção A.5) e *incircle* (Seção A.6). Os predicados *left* foi utilizado pelo algoritmo *Graham scan* (Capítulo 3) e, juntamente com os predicados *lefton*, *collinear* e *classify*, *left* também é utilizado para localizar um ponto em um





triângulo em uma dada triangulação de Delaunay de um subconjunto de pontos de  $\mathbb{E}^2$  (Capítulo 4). Finalmente, o predicado *incircle* nos permite determinar se uma dada aresta de uma dada triangulação de um subconjunto de pontos de  $\mathbb{E}^2$  é localmente Delaunay (Capítulo 4).

## A.2 Orientação

Seja  $\mathbb{E}^n$  o espaço euclidiano afim de dimensão  $n$  sobre  $\mathbb{R}$ . Por uma *orientação* de  $\mathbb{E}^n$ , entende-se uma orientação de seu espaço vetorial associado,  $\mathbb{R}^n$ , que, por sua vez, corresponde à escolha de uma base para  $\mathbb{R}^n$ .

**Definição A.1.** *Uma orientação de  $\mathbb{R}^n$  é simplesmente a escolha de uma das duas classes de equivalência da relação de equivalência no conjunto de bases definida da seguinte forma: sejam  $B = (\mathbf{u}_1, \dots, \mathbf{u}_n)$  e  $B' = (\mathbf{v}_1, \dots, \mathbf{v}_n)$  duas bases para  $\mathbb{R}^n$ . Então, dizemos que elas possuem a mesma orientação se, e somente se, o determinante da matriz  $\mathbf{M}_{B' \leftarrow B} \in M_n(\mathbb{R})$  é positivo, em que  $\mathbf{M}_{B' \leftarrow B}$  é a matriz de mudança de base da base  $B$  para a base  $B'$ . Toda base na classe escolhida é dita possuir orientação positiva ou ser positiva. Uma orientação de  $\mathbb{E}^n$  é simplesmente uma orientação de seu espaço vetorial associado,  $\mathbb{R}^n$ .*

Por exemplo, considere as bases

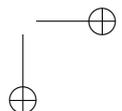
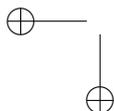
$$B_1 = ((1, 0), (0, 1)) \quad \text{e} \quad B_2 = \left( \left( \frac{1}{2}, \frac{\sqrt{3}}{2} \right), \left( -\frac{\sqrt{3}}{2}, \frac{1}{2} \right) \right)$$

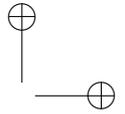
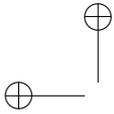
de  $\mathbb{R}^2$ . Então,

$$\mathbf{M}_{B_2 \leftarrow B_1} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \in M_2(\mathbb{R}).$$

Como,

$$\det(\mathbf{M}_{B_2 \leftarrow B_1}) = \begin{vmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{vmatrix} = \frac{1}{4} + \frac{3}{4} = 1 > 0,$$





as bases  $B_1$  e  $B_2$  possuem a mesma orientação. Note que a base  $B_2$  é simplesmente uma rotação de  $60^\circ$  da base  $B_1$  no sentido anti-horário. O significado intuitivo de  $\det(\mathbf{M}_{B_2 \leftarrow B_1}) > 0$  pode ser compreendido com o auxílio de sistemas de coordenadas definidos em termos de  $B_1$  e  $B_2$ .

Considere dois sistemas de coordenadas com origem no mesmo ponto  $O = (0, 0) \in \mathbb{E}^2$ . O primeiro deles é tal que os eixos  $x$  e  $y$  possuem a mesma direção e sentido dos vetores  $(1, 0)$  e  $(0, 1)$  da base  $B_1$ , respectivamente. O segundo deles é tal que os eixos  $x$  e  $y$  possuem a mesma direção e sentido dos vetores  $(1/2, \sqrt{3}/2)$  e  $(-\sqrt{3}/2, 1/2)$  da base  $B_2$ , respectivamente (veja a Figura A.1). Considere os pontos

$$p_1 = (2, 0), \quad p_2 = (1, 2) \quad \text{e} \quad p_3 = (0, 2)$$

da figura. Quando os visitamos na ordem  $p_1, p_2$  e  $p_3$ , fazemos um percurso anti-horário com relação ao primeiro sistema de coordenadas. A transformação linear associada a  $\mathbf{M}_{B_2 \leftarrow B_1}$  leva  $\overrightarrow{Op_1}$ ,  $\overrightarrow{Op_2}$  e  $\overrightarrow{Op_3}$  em

$$\overrightarrow{Op'_1} = (1, \sqrt{3}), \quad \overrightarrow{Op'_2} = \left( \frac{1}{2} - \sqrt{3}, \frac{\sqrt{3}}{2} + 1 \right) \quad \text{e} \quad \overrightarrow{Op'_3} = (-\sqrt{3}, 1),$$

respectivamente. Ao percorrermos  $p'_1, p'_2$  e  $p'_3$ , fazemos um percurso anti-horário com relação ao segundo sistema de coordenadas. Em outras palavras, o sentido do percurso não muda de um sistema para outro.

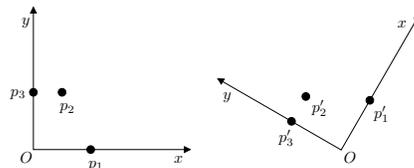
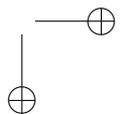
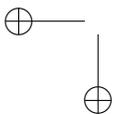
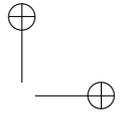
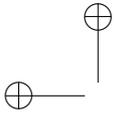


Figura A.1: Dois sistemas de coordenadas com a mesma orientação.

Considera a base  $B_3 = ((-1, 0), (0, 1))$  de  $\mathbb{R}^2$ . A matriz  $\mathbf{M}_{B_3 \leftarrow B_1}$





é

$$\mathbf{M}_{B_3 \leftarrow B_1} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Note que

$$\det(\mathbf{M}_{B_3 \leftarrow B_1}) = -1 < 0.$$

Isto significa que  $B_1$  e  $B_3$  não possuem a mesma orientação. O que isso significa? Se definirmos um terceiro sistema de coordenadas com origem em  $O = (0, 0) \in \mathbb{E}^2$  e eixos  $x$  e  $y$  com mesma direção e sentido dos vetores  $(-1, 0)$  e  $(0, 1)$ , respectivamente, então os vetores  $\overrightarrow{Op_1} = (2, 0)$ ,  $\overrightarrow{Op_2} = (1, 2)$  e  $\overrightarrow{Op_3} = (0, 2)$  são transformados nos vetores

$$\overrightarrow{Op'_1} = (-2, 0) \quad \overrightarrow{Op'_2} = (-1, 2) \quad \text{e} \quad \overrightarrow{Op'_3} = (0, 2)$$

pela transformação linear associada à matriz  $\mathbf{M}_{B_3 \leftarrow B_1}$ . Ao percorrermos  $p'_1$ ,  $p'_2$  e  $p'_3$ , nesta ordem, fazemos um percurso horário, como mostrado na Figura A.2. Portanto, o sentido do percurso muda do sistema de coordenadas associado a  $B_1$  para o sistema associado a  $B_3$ .

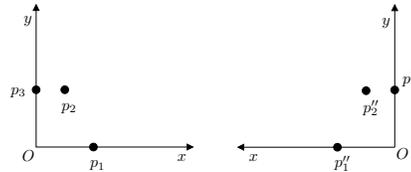
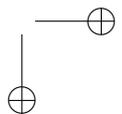
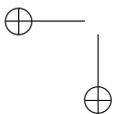
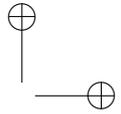
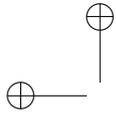


Figura A.2: Dois sistemas de coordenadas com orientações opostas.

Observe que se  $B_1$  tem orientação positiva, então  $B_2$  também possui orientação positiva, enquanto  $B_3$  possui orientação negativa. De forma análoga, se  $B_1$  tem orientação negativa, então  $B_2$  também possui orientação negativa, enquanto  $B_3$  possui orientação positiva. Observe também que a escolha de uma base como sendo positiva é completamente arbitrária. No entanto, ao fazermos esta escolha para *uma* base qualquer, *toda* base estará classificada como positiva ou negativa.





### A.3 Teste de orientação

Suponha que uma orientação tenha sido escolhida para  $\mathbb{E}^n$ . Então, dados  $n + 1$  pontos,  $p_1, \dots, p_n, p_0$ , em  $\mathbb{E}^n$ , o teste de orientação consiste em determinar se os  $n$  vetores  $\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}$  formam uma base positiva de  $\mathbb{E}^n$ . Lembre-se de que uma base  $B$  é considerada positiva se  $\det(M_{B \leftarrow B'}) > 0$ , em que  $M_{B \leftarrow B'}$  é matriz de mudança de base da base,  $B'$ , escolhida como orientação de  $\mathbb{E}^n$  para  $B$ . O teste de orientação pode ser efetuado através do cálculo do produto misto dos vetores

$$\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}.$$

O produto misto nos fornece o volume sinalizado de uma dada “forma de volume”. Se  $n = 2$ , esta forma é um paralelogramo; se  $n = 3$ , um paralelepípedo. Para definir produto misto, necessitamos do seguinte:

**Lema A.1.** *Suponha que uma orientação de  $\mathbb{R}^n$  tenha sido escolhida. Então, para qualquer lista  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$  de  $n$  vetores de  $\mathbb{R}^n$  e duas bases ortonormais quaisquer,  $B_1 = (\mathbf{u}_1, \dots, \mathbf{u}_n)$  e  $B_2 = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ , positivas,*

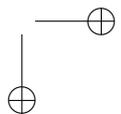
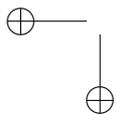
$$\det_{B_1}(\mathbf{w}_1, \dots, \mathbf{w}_n) = \det_{B_2}(\mathbf{w}_1, \dots, \mathbf{w}_n).$$

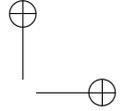
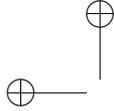
*Demonstração.* Seja  $\mathbf{M}_{B_2 \leftarrow B_1}$  a matriz de mudança de base da base  $B_1 = (\mathbf{u}_1, \dots, \mathbf{u}_n)$  para a base  $B_2 = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ . Como  $B_1$  e  $B_2$  são ortonormais, a matriz  $P$  é ortogonal e, portanto,  $\det(\mathbf{M}_{B_2 \leftarrow B_1}) = \pm 1$ . Mas, por hipótese  $B_1$  e  $B_2$  possuem orientação positiva. Então,  $\det(\mathbf{M}_{B_2 \leftarrow B_1}) = +1$ . Seja  $\mathbf{M}_1$  a matriz cujas colunas são as coordenadas dos vetores  $\mathbf{w}_j$  com relação à base  $B_1$ , e seja  $\mathbf{M}_2$  a matriz cujas colunas são as coordenadas dos vetores  $\mathbf{w}_j$  com relação à base  $B_2$ . Então, pela definição de matriz de mudança de base, temos que

$$\mathbf{M}_2 = \mathbf{M}_{B_2 \leftarrow B_1} \cdot \mathbf{M}_1.$$

Como

$$\det_{B_1}(\mathbf{w}_1, \dots, \mathbf{w}_n) = \det(\mathbf{M}_1) \quad \text{e} \quad \det_{B_2}(\mathbf{w}_1, \dots, \mathbf{w}_n) = \det(\mathbf{M}_2),$$





temos

$$\begin{aligned}
 \det_{B_1}(\mathbf{w}_1, \dots, \mathbf{w}_n) &= \det(\mathbf{M}_1) \\
 &= \det(\mathbf{M}_{B_2 \leftarrow B_1} \cdot \mathbf{M}_2) \\
 &= \det(\mathbf{M}_{B_2 \leftarrow B_1}) \cdot \det(\mathbf{M}_2) \\
 &= \det(\mathbf{M}_{B_2 \leftarrow B_1}) \cdot \det_{B_2}(\mathbf{w}_1, \dots, \mathbf{w}_n) \\
 &= \det_{B_2}(\mathbf{w}_1, \dots, \mathbf{w}_n).
 \end{aligned}$$

□

O Lema A.1 nos permite definir produto misto de forma consistente:

**Definição A.2.** Dada uma orientação de  $\mathbb{R}^n$ , para qualquer lista,  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$ , de  $n$  vetores em  $\mathbb{R}^n$ , o valor comum,  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n)$ , de

$$\det_B(\mathbf{w}_1, \dots, \mathbf{w}_n)$$

com relação a todas as bases ortonormais positivas,  $B$ , de  $\mathbb{R}^n$  é chamado produto misto (ou forma de volume) dos vetores  $\mathbf{w}_1, \dots, \mathbf{w}_n$ .

A terminologia *forma de volume* é justificada, pois o número  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n)$  é o volume de algum objeto geométrico. Em particular, este objeto é o *paralelepípedo* definido por  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$ , isto é, o conjunto

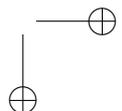
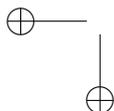
$$\{O + \lambda_1 \cdot \mathbf{w}_1 + \dots + \lambda_n \cdot \mathbf{w}_n \mid 0 \leq \lambda_i \leq 1, 1 \leq i \leq n\} \subset \mathbb{E}^n.$$

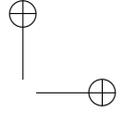
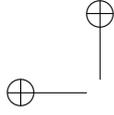
Podemos mostrar que o volume do paralelepípedo acima é de fato igual a  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n)$ . Em particular, quando  $n = 2$ , o paralelepípedo é o paralelogramo definido por dois vetores e  $\lambda_{\mathbb{R}^2}(\mathbf{w}_1, \mathbf{w}_2)$  é a área sinalizada do paralelogramo. Além disso, temos as seguintes propriedades:

- (1)  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n) = 0$  se, e somente se,  $\{\mathbf{w}_1, \dots, \mathbf{w}_n\}$  é l.i.
- (2) Uma base  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$  é positiva se, e somente se, o produto misto

$$\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n)$$

é positivo.





As propriedades (1) e (2) acima são cruciais para determinar o resultado do teste de orientação. De fato, se tomarmos  $\mathbf{w}_i = \overrightarrow{p_0 p_i}$ , para todo  $i = 1, \dots, n$ , o teste de orientação se reduz a determinar se  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$  é uma base de  $\mathbb{R}^n$  e, se for, se ela é positiva. De acordo com a propriedade (1), se  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n) = 0$  então  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$  não é uma base de  $\mathbb{R}^n$ . Por sua vez, se (1) é verdadeira, (2) nos diz que se  $\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n) > 0$  então  $(\mathbf{w}_1, \dots, \mathbf{w}_n)$  é uma base positiva. Caso contrário, ela é negativa. Então, o teste se resume a calcular o valor de

$$\lambda_{\mathbb{R}^n}(\mathbf{w}_1, \dots, \mathbf{w}_n).$$

De agora em diante, assumimos que a base canônica,  $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ , para  $\mathbb{R}^n$  é uma base *positiva*. Dados  $n + 1$  pontos  $p_1, \dots, p_n, p_0$ , de  $\mathbb{E}^n$ , denotamos por  $p_{ij}$  a  $j$ -ésima coordenada de  $p_i$ . Então, temos que

$$\lambda_{\mathbb{R}^n}(\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n}) = \begin{vmatrix} p_{11} - p_{01} & p_{12} - p_{02} & \cdots & p_{1n} - p_{0n} \\ p_{21} - p_{01} & p_{22} - p_{02} & \cdots & p_{2n} - p_{0n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} - p_{01} & p_{n2} - p_{02} & \cdots & p_{nn} - p_{0n} \end{vmatrix}.$$

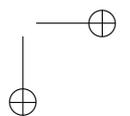
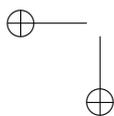
Note que

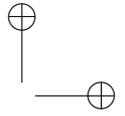
$$\begin{vmatrix} p_{11} - p_{01} & p_{12} - p_{02} & \cdots & p_{1n} - p_{0n} \\ p_{21} - p_{01} & p_{22} - p_{02} & \cdots & p_{2n} - p_{0n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} - p_{01} & p_{n2} - p_{02} & \cdots & p_{nn} - p_{0n} \end{vmatrix}$$

é igual a

$$\begin{vmatrix} p_{11} & p_{12} & \cdots & p_{1n} & 1 \\ p_{21} & p_{22} & \cdots & p_{2n} & 1 \\ \vdots & \vdots & \ddots & \vdots & 1 \\ p_{n1} & p_{n2} & \cdots & p_{nn} & 1 \\ p_{01} & p_{02} & \cdots & p_{0n} & 1 \end{vmatrix}$$

e ambos são iguais a  $\det_B(\overrightarrow{p_0 p_1}, \dots, \overrightarrow{p_0 p_n})$ , em que  $B$  é *qualquer* base ortonormal positiva para  $\mathbb{R}^n$ . Por “positiva”, entende-se que  $B$  possui a mesma orientação da base canônica,  $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ , de  $\mathbb{E}^n$ , a qual escolhemos *arbitrariamente* como sendo positiva. Quando  $n = 2$  ou





$n = 3$ , o teste de orientação tem uma simples interpretação geométrica.

Suponha que  $n$  é igual a 2. Então, se  $\lambda_{\mathbb{R}^2}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2})$  é positivo, dizemos que a *sequência*  $p_0, p_1$  e  $p_2$  de pontos de  $\mathbb{E}^2$  tem sentido anti-horário, ou seja, faz uma “curva” para a esquerda. Se  $\lambda_{\mathbb{R}^2}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2})$  é negativo, dizemos que a *sequência*  $p_0, p_1$  e  $p_2$  de pontos de  $\mathbb{E}^2$  tem sentido horário, ou seja, faz uma “curva” para a direita. Se  $\lambda_{\mathbb{R}^2}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2})$  é igual a 0, os pontos  $p_0, p_1$  e  $p_2$  são colineares.

Suponha que  $n$  é igual a 3. Então, se  $\lambda_{\mathbb{R}^3}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}, \overrightarrow{p_0p_3})$  é positivo, dizemos que a *sequência*  $p_1, p_2$  e  $p_3$  de pontos de  $\mathbb{E}^3$  tem sentido horário com relação ao ponto  $p_0 \in \mathbb{E}^3$ , ou seja, o plano que passa por  $p_1, p_2$  e  $p_3$  e é perpendicular ao vetor  $\overrightarrow{p_1p_2} \times \overrightarrow{p_1p_3}$  é visto de “baixo” por  $p_0$ . Se  $\lambda_{\mathbb{R}^3}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}, \overrightarrow{p_0p_3})$  é negativo, dizemos que a *sequência*  $p_1, p_2$  e  $p_3$  de pontos de  $\mathbb{E}^3$  tem sentido anti-horário com relação ao ponto  $p_0 \in \mathbb{E}^3$ , ou seja, o plano que passa por  $p_1, p_2$  e  $p_3$  e é perpendicular ao vetor  $\overrightarrow{p_1p_2} \times \overrightarrow{p_1p_3}$  é visto de “cima” por  $p_0$ . Se o valor de  $\lambda_{\mathbb{R}^3}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}, \overrightarrow{p_0p_3})$  é igual a 0, então os pontos  $p_0, p_1, p_2$  e  $p_3$  são coplanares. A Figura A.3 ilustra todos os dois primeiros casos.

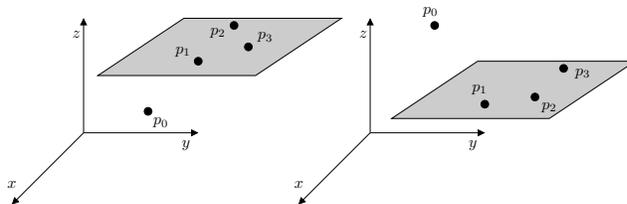
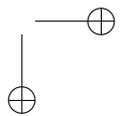
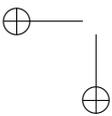


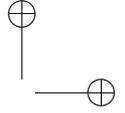
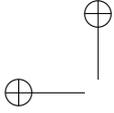
Figura A.3: Interpretação geométrica do teste de orientação em  $\mathbb{E}^3$ .

Daqui em diante, denotaremos os produtos mistos

$$\lambda_{\mathbb{R}^2}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}) \quad \text{e} \quad \lambda_{\mathbb{R}^3}(\overrightarrow{p_0p_1}, \overrightarrow{p_0p_2}, \overrightarrow{p_0p_3})$$

por  $O2D(p_0, p_1, p_2)$  e  $O3D(p_0, p_1, p_2, p_3)$ , respectivamente.





## A.4 Os predicados *left*, *lefton* e *collinear*

Nesta seção, apresentamos algoritmos para três predicados: *left*, *lefton* e *collinear*. Mais especificamente, dada uma sequência,  $p$ ,  $q$  e  $r$ , de três pontos distintos em  $\mathbb{E}^2$ , o predicado *left* determina se o ponto  $r$  está estritamente à esquerda da reta orientada de  $p$  para  $q$ ; o predicado *lefton* determina se o ponto  $r$  está à esquerda ou sobre a reta orientada de  $p$  para  $q$ ; e o predicado *collinear* determina se os pontos  $p$ ,  $q$  e  $r$  são colineares. Como foi visto na Seção A.3, todos esses predicados podem ser implementados com base no resultado do teste de orientação  $O2D(p, q, r)$ . Portanto, usamos o algoritmo  $O2D()$  (veja o Algoritmo A.1) a seguir para implementar os algoritmos correspondentes aos predicados. Todos os algoritmos neste capítulo assumem que a base canônica,  $(\mathbf{e}_1, \mathbf{e}_2)$ , de  $\mathbb{E}^2$  é uma base positiva.

---

### Algoritmo A.1 $O2D(p, q, r)$

---

- 1: sejam  $(px, py)$  as coordenadas de  $p$
  - 2: sejam  $(qx, qy)$  as coordenadas de  $q$
  - 3: sejam  $(rx, ry)$  as coordenadas de  $r$
  - 4: **devolva**  $px * qy - py * qx + py * rx - px * ry + qx * ry - rx * qy$
- 

$O2D()$  recebe como entrada os pontos  $p$ ,  $q$  e  $r$  cujas coordenadas são dadas com respeito a algum referencial afim,  $(O, (\mathbf{i}, \mathbf{j}))$ , tal que a base  $(\mathbf{i}, \mathbf{j})$  tenha orientação positiva. Como mostrado na Figura A.4, o ponto  $r$  está estritamente à esquerda da reta,  $\ell$ , orientada de  $p$  para  $q$  se, e somente se, os três pontos não são colineares e a sequência  $p, q$  e  $r$  define um percurso anti-horário. De forma equivalente, o ponto  $r$  está estritamente à esquerda de  $\ell$  se, e somente se,  $O2D(p, q, r) > 0$ . Logo, podemos implementar o predicado *left* (veja o Algoritmo A.2) comparando o resultado devolvido por uma chamada a  $O2D()$  com zero.

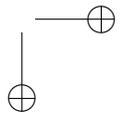
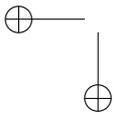
---

### Algoritmo A.2 $LEFT(p, q, r)$

---

- 1: **devolva**  $O2D(p, q, r) > 0$
- 

Um outro predicado de interesse é *lefton*. Este predicado determina se  $r$  está à esquerda da reta  $\ell$  (ou seja, estritamente à esquerda



ou sobre  $\ell$ ). Logo, o predicado *lefton* assume o valor verdadeiro se, e somente se, o resultado do teste  $O2D(p, q, r)$  é maior ou igual a zero. Por sua vez, o predicado *collinear* assume o valor verdadeiro se, e somente se, o resultado do teste  $O2D(p, q, r)$  é igual a zero. Os algoritmos `LEFTON()` e `COLLINEAR()` em Algoritmo A.3 e Algoritmo A.4 dados a seguir, respectivamente, implementam os predicados homônimos.

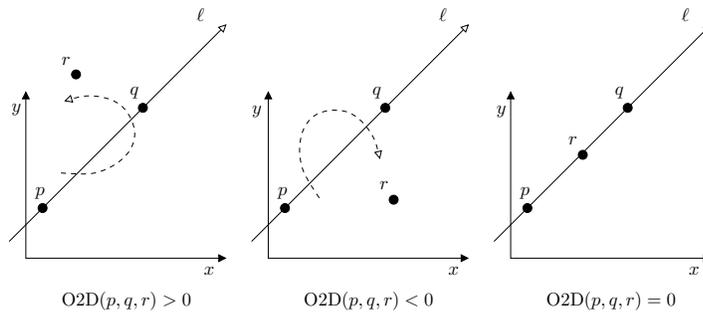


Figura A.4: Os possíveis resultados do teste  $O2D(p, q, r)$ .

---

**Algoritmo A.3** `LEFTON`( $p, q, r$ )

---

1: **devolva**  $O2D(p, q, r) \geq 0$

---

## A.5 O predicado *classify*

O procedimento de determina o triângulo ou aresta de uma triangulação de Delaunay contendo um dado ponto de  $\mathbb{E}^2$  (veja a Seção 4.6) requer uma *classificação* mais detalhada da posição de um ponto  $r$  de  $\mathbb{E}^2$  com respeito à reta orientada,  $\ell$ , de um ponto  $p$  para um ponto  $q$  de  $\mathbb{E}^2$ , com  $p \neq q$ . Esta classificação determina qual dos seguintes casos mutuamente exclusivos ocorrem: (a)  $r$  está estritamente à esquerda de  $\ell$ , (b)  $r$  está estritamente à direita de  $\ell$ , (c)  $r$  é igual a  $p$ , (d)  $r$  é igual a  $q$ , (e)  $r$  pertence ao interior do segmento  $\overline{pq}$ , (f)  $r$

é um ponto da reta  $\ell$  “aquém” de  $p$  ou (g)  $r$  é um ponto da reta  $\ell$  “além” de  $q$ . Na Seção A.4, aprendemos a determinar se os casos (a) e (b) ocorrem. A ocorrência do caso (c) ou (d) pode ser trivialmente determinada. Então, o que nos resta é estudar os casos (e), (f) e (g).

---

**Algoritmo A.4** COLLINEAR( $p, q, r$ )

---

1: **devolva** O2D( $p, q, r$ ) = 0

---

Suponha que os casos (a)-(d) tenham sido descartados. Então, já sabemos que  $r$  pertence à reta  $\ell$ . O que queremos é determinar “onde”  $r$  está com respeito ao segmento  $\overline{pq}$ . Se  $(px, py)$ ,  $(qx, qy)$  e  $(rx, ry)$  são as coordenadas de  $p$ ,  $q$  e  $r$  com respeito a um referencial  $(O, \mathbf{i}, \mathbf{j})$  de  $\mathbb{E}^2$ , em que  $(\mathbf{i}, \mathbf{j})$  é uma base ortonormal positiva de  $\mathbb{R}^2$ , então distinguimos duas situações: (1)  $\ell$  não é vertical (isto é,  $px \neq qx$ ) e (2)  $\ell$  é vertical (isto é,  $px = qx$ ). No caso (1), o ponto  $r$  estará no interior de  $\overline{pq}$  se, e somente se,  $(rx - px) * (qx - rx) > 0$ . Se  $(rx - px) * (qx - rx) < 0$ , então o ponto  $r$  estará “aquém” de  $p$  se  $|px - rx| < |qx - rx|$ . Caso contrário, o ponto  $r$  estará “além” de  $q$  (veja a Figura A.5). O caso (2) pode ser tratado de forma semelhante: o ponto  $r$  estará no interior de  $\overline{pq}$  se  $(ry - py) * (qy - ry) > 0$ ; se  $(ry - py) * (qy - ry) < 0$  então o ponto  $r$  estará “aquém” de  $p$  se  $|py - ry| < |qy - ry|$ ; caso contrário, o ponto  $r$  estará “além” do ponto  $q$ .

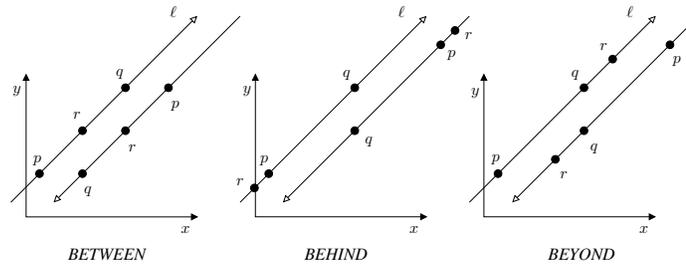


Figura A.5: Classificação de um ponto  $r$  com relação à reta  $\ell$ .

Por convenção, dizemos que a classificação de  $r$  com relação à reta  $\ell$  é *LEFT*, *RIGHT*, *ORIGIN*, *DESTINATION*, *BETWEEN*,

*BEHIND* e *BEYOND* se  $r$  satisfaz os casos (a), (b), (c), (d), (e), (f) e (g), respectivamente. O Algoritmo A.5 dado a seguir recebe como entrada os pontos  $p$ ,  $q$  e  $r$  e devolve a classificação de  $r$  com respeito à reta  $\ell$ .

---

**Algoritmo A.5** CLASSIFY( $p, q, r$ )

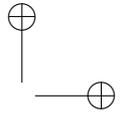
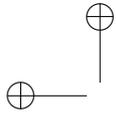
---

```

1: sejam  $(px, py)$ ,  $(qx, qy)$  e  $(rx, ry)$  as coordenadas de  $p$ ,  $q$  e  $r$ 
2:  $o2d \leftarrow O2D(p, q, r)$ 
3: se  $o2d > 0$  então
4:   devolva LEFT
5: senão se  $o2d < 0$  então
6:   devolva RIGHT
7: senão se  $px = rx$  e  $py = ry$  então
8:   devolva ORIGIN
9: senão se  $qx = rx$  e  $qy = ry$  então
10:  devolva DESTINATION
11: senão se  $px \neq qx$  então
12:   $d1 = rx - px$ 
13:   $d2 = qx - rx$ 
14:  se  $(d1 * d2) > 0$  então
15:    devolva BETWEEN
16:  senão se  $|px - rx| < |qx - rx|$  então
17:    devolva BEHIND
18:  fim se
19:  devolva BEYOND
20: fim se
21:  $d1 = ry - py$ 
22:  $d2 = qy - ry$ 
23: se  $(d1 * d2) > 0$  então
24:  devolva BETWEEN
25: senão se  $|py - ry| < |qy - ry|$  então
26:  devolva BEHIND
27: fim se
28: devolva BEYOND

```

---



## A.6 O predicado *incircle*

Todos os predicados que vimos até então são aplicações diretas do teste de orientação  $O2D()$ . Agora, estudaremos um predicado, denominado *incircle*, que pode ser implementado através dos testes de orientação  $O2D()$  e  $O3D()$ . Este predicado é extremamente importante para nosso estudo de triangulações de Delaunay (Capítulo 4). Em particular, o predicado *incircle* nos diz se um dado ponto  $q$  de  $\mathbb{E}^2$  pertence ao interior do círculo definido por três pontos,  $p_0$ ,  $p_1$  e  $p_2$ , de  $\mathbb{E}^2$ .

Obviamente, *incircle* pode ser implementado da seguinte forma: calculamos o centro  $o$  e o raio  $\rho$  do círculo, obtemos a distância,  $d(q, o)$ , entre  $q$  e  $o$  e, finalmente, comparamos  $d(q, o)$  com  $\rho$ . O ponto  $q$  reside no interior do círculo se, e somente se,  $d(q, o) < \rho$ . Entretanto, a solução que estudaremos aqui faz uso de  $O2D()$  e  $O3D()$  e, portanto, permite que todos os predicados estudados nesta apêndice dependam fundamentalmente do cálculo de um ou dois determinantes. A forma de cálculo do predicado *incircle* que estudaremos agora se baseia no mapa de elevação,  $\omega : \mathbb{E}^2 \rightarrow \mathbb{E}^3$ , tal que  $\omega(x, y) = x^2 + y^2$  (Seção 4.3). Mais especificamente, para todo ponto  $p = (p_x, p_y) \in \mathbb{E}^2$ , temos o ponto

$$p^\omega = (p_x, p_y, \omega(p_x, p_y)) = (p_x, p_y, p_x^2 + p_y^2) \in \mathbb{E}^3.$$

A função  $\omega$  leva  $\mathbb{E}^2$  no parabolóide de revolução,  $\Gamma$ , em  $\mathbb{E}^3$  definido como

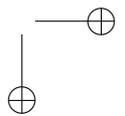
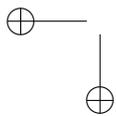
$$\Gamma = \{(x, y, z) \in \mathbb{E}^3 \mid z = x^2 + y^2\}.$$

Seja

$$C = \{(x, y) \in \mathbb{E}^2 \mid (x - o_x)^2 + (y - o_y)^2 - \rho^2 = 0\}$$

a circunferência de centro  $o = (o_x, o_y) \in \mathbb{E}^2$  e raio  $\rho \in \mathbb{R}$ , com  $\rho > 0$ , que passa por pelos pontos  $p_0$ ,  $p_1$  e  $p_2$  de  $\mathbb{E}^2$ . Então, o conjunto  $C^\omega$  é tal que

$$\begin{aligned} C^\omega &= \{(x, y, z) \in \mathbb{E}^3 \mid (x, y) \in C \text{ e } z = x^2 + y^2\} \\ &= \{(x, y, z) \in \mathbb{E}^3 \mid x^2 + y^2 - 2o_x x - 2o_y y + o_x^2 + o_y^2 - \rho^2 = 0\} \\ &\cap \{(x, y, z) \in \mathbb{E}^3 \mid z = x^2 + y^2\} \\ &= \{(x, y, z) \in \mathbb{E}^3 \mid z - 2o_x x - 2o_y y + o_x^2 + o_y^2 - \rho^2 = 0\}. \end{aligned}$$



Observe que

$$z - 2o_x x - 2o_y y + o_x^2 + o_y^2 - \rho^2 = 0$$

é a equação de um plano  $H$  (se tratarmos  $z$  de forma independente de  $x$  e  $y$ ). O plano  $H$  passa por  $p_0^\omega, p_1^\omega$  e  $p_2^\omega$ . A interseção de  $H$  e  $\Gamma$  é exatamente  $C^\omega$  ou, de forma equivalente, a elevação dos pontos de  $C$  a  $\Gamma$  por  $\omega$ . Além disso,  $\Gamma - C^\omega$  consiste de dois conjuntos conexos: (1) o conjunto,  $A$ , de pontos de  $\mathbb{E}^3$  que consiste da elevação dos pontos do exterior do círculo cuja circunferência é  $C$  a  $\Gamma$  por  $\omega$  e (2) o conjunto,  $B$ , de pontos de  $\mathbb{E}^3$  que consiste da elevação dos pontos do interior do círculo cuja circunferência é  $C$  a  $\Gamma$  por  $\omega$ . Logo, o ponto  $q$  reside no interior de  $C$  se, e somente se, o ponto  $q^\omega$  pertence ao conjunto  $B$ . A Figura A.6 ilustra a afirmação que acabamos de fazer acerca de  $q$  e do interior do círculo cuja circunferência é  $C$  (refira-se à Seção 4.3).

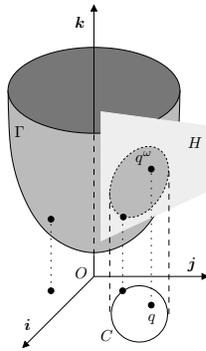
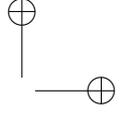
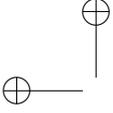


Figura A.6: O ponto  $q \in \mathbb{E}^2$  é levado por  $\omega$  para  $\Gamma$ .

Suponha que os pontos  $p_0, p_1$  e  $p_2$ , nesta ordem, definam um percurso no sentido anti-horário, isto é,  $O2D(p_0, p_1, p_2) > 0$ . Então, se  $q^\omega$  pertence a  $B$  (o interior do círculo de circunferência  $C$ ), então  $q^\omega$  “verá”  $p_0^\omega, p_1^\omega$  e  $p_2^\omega$ , nesta ordem, no sentido horário, ou seja, de *baixo* do plano  $H$ , desde que a normal de  $H$  seja definida por  $\overrightarrow{p_0^\omega p_1^\omega} \times \overrightarrow{p_0^\omega p_2^\omega}$ . Caso contrário, se  $q^\omega$  pertence a  $A$  (o exterior do círculo de circunferência  $C$ ), então  $q^\omega$  “verá”  $p_0^\omega, p_1^\omega$  e  $p_2^\omega$ , nesta ordem, no sentido anti-horário, ou seja, de *cima* do plano  $H$ , desde que a normal



de  $H$  seja definida por  $\overrightarrow{p_0^{\omega} p_1^{\omega}} \times \overrightarrow{p_0^{\omega} p_2^{\omega}}$ . Finalmente, observe que se  $q^{\omega} \in B$  então  $\text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) > 0$ . No entanto, se  $q^{\omega} \in A$  então  $\text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) < 0$ . Logo, como  $\text{O2D}(p_0, p_1, p_2)$  é positivo, o ponto  $q$  está no interior do círculo de circunferência  $C$  se, e somente se,

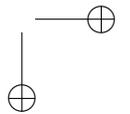
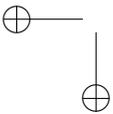
$$\text{O2D}(p_0, p_1, p_2) \cdot \text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) > 0. \quad (\text{A.1})$$

Suponha agora que os pontos  $p_0, p_1$  e  $p_2$ , nesta ordem, definam um percurso no sentido horário, isto é,  $\text{O2D}(p_0, p_1, p_2) < 0$ . Então, se  $q^{\omega}$  pertence a  $B$  (o interior da círculo de circunferência  $C$ ), então  $q^{\omega}$  “verá”  $p_0^{\omega}, p_1^{\omega}$  e  $p_2^{\omega}$ , nesta ordem, no sentido anti-horário, ou seja, de *cima* do plano  $H$ , desde que a normal de  $H$  seja definida por  $\overrightarrow{p_0^{\omega} p_1^{\omega}} \times \overrightarrow{p_0^{\omega} p_2^{\omega}}$ . Caso contrário, se  $q^{\omega}$  pertence a  $A$  (o exterior do círculo de circunferência  $C$ ), então  $q^{\omega}$  “verá”  $p_0^{\omega}, p_1^{\omega}$  e  $p_2^{\omega}$ , nesta ordem, no sentido horário, ou seja, de *baixo* do plano  $H$ , desde que a normal de  $H$  seja definida por  $\overrightarrow{p_0^{\omega} p_1^{\omega}} \times \overrightarrow{p_0^{\omega} p_2^{\omega}}$ . Finalmente, observe que se  $q^{\omega} \in B$  então  $\text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) < 0$ . Por outro lado, se  $q^{\omega} \in A$  então  $\text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) > 0$ . Portanto, como  $\text{O2D}(p_0, p_1, p_2) < 0$ , o ponto  $q$  reside no interior do círculo de circunferência  $C$  se, e somente se,

$$\text{O2D}(p_0, p_1, p_2) \cdot \text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) > 0. \quad (\text{A.2})$$

As observações nos dois parágrafos anteriores nos dizem que, independentemente do valor de  $\text{O2D}(p_0, p_1, p_2)$ , podemos dizer que  $q$  pertence ao interior do círculo de circunferência  $C$  sempre que o produto em Eq. (A.1) (resp. Eq. (A.2)) for positivo. Além disso, quando ele for zero, então  $\text{O2D}(p_0, p_1, p_2) = 0$  ou  $\text{O3D}(q^{\omega}, p_0^{\omega}, p_1^{\omega}, p_2^{\omega}) = 0$ . No primeiro caso, os pontos  $p_0, p_1$  e  $p_2$  são colineares. No segundo caso, os pontos  $p_0^{\omega}, p_1^{\omega}, p_2^{\omega}$  e  $q^{\omega}$  são coplanares. Mas, pela definição de  $\omega$ , este é o caso se, e somente se, o ponto  $q$  reside na circunferência  $C$ .

O algoritmo `INCIRCLE()` (veja o Algoritmo A.6) implementa o predicado *incircle* com base em `O2D()` e `O3D()`, como descrito acima. A entrada do algoritmo consiste dos pontos  $p_0, p_1, p_2$  e  $q$ , nesta ordem. A saída é o valor lógico verdadeiro se  $q$  está no interior do círculo definido por  $p_0, p_1$  e  $p_2$ . Caso contrário, a saída é o valor lógico falso. O algoritmo assume que  $p_0, p_1$  e  $p_2$  são distintos e não colineares. O Algoritmo A.7 implementa o teste de orientação `O3D()`.



---

**Algoritmo A.6** INCIRCLE( $p_0, p_1, p_2, q$ )

---

- 1: sejam  $(p_0x, p_0y)$  as coordenadas de  $p_0$
  - 2: sejam  $(p_1x, p_1y)$  as coordenadas de  $p_1$
  - 3: sejam  $(p_2x, p_2y)$  as coordenadas de  $p_2$
  - 4: sejam  $(qx, qy)$  as coordenadas de  $q$
  - 5:  $u_0 \leftarrow (p_0x, p_0y, p_0x * p_0x + p_0y * p_0y)$
  - 6:  $u_1 \leftarrow (p_1x, p_1y, p_1x * p_1x + p_1y * p_1y)$
  - 7:  $u_2 \leftarrow (p_2x, p_2y, p_2x * p_2x + p_2y * p_2y)$
  - 8:  $q_2 \leftarrow (qx, qy, qx * qx + qy * qy)$
  - 9:  $o_2d \leftarrow \text{O2D}(p_0, p_1, p_2)$
  - 10:  $o_3d \leftarrow \text{O3D}(q_2, u_0, u_1, u_2)$
  - 11: **devolva**  $o_2d * o_3d > 0$
- 

---

**Algoritmo A.7** O3D( $p, q, r, s$ )

---

- 1: sejam  $(px, py, pz)$  as coordenadas de  $p$
  - 2: sejam  $(qx, qy, qz)$  as coordenadas de  $q$
  - 3: sejam  $(rx, ry, rz)$  as coordenadas de  $r$
  - 4: sejam  $(sx, sy, sz)$  as coordenadas de  $s$
  - 5: sejam  $(qx, qy)$  as coordenadas de  $q$
  - 6: sejam  $(rx, ry)$  as coordenadas de  $r$
  - 7: **devolva**

$$\begin{aligned}
 & -(qx*ry*sz) + (px*ry*sz) + (qy*rx*sz) - (py*rx*sz) - (px*qy*sz) \\
 & + (py*qx*sz) + (qx*rz*sy) - (px*rz*sy) - (qz*rx*sy) + (pz*rx*sy) \\
 & + (px*qz*sy) - (pz*qx*sy) - (qy*rz*sx) + (px*rz*sz) + (qz*ry*sx) \\
 & - (pz*ry*sx) - (px*qy*sx) + (pz*qy*sz) + (px*qy*rz) - (py*qx*rz) \\
 & - (px * qz * ry) + (pz * qx * sy) + (py * qz * rx) - (pz * qy * rx)
 \end{aligned}$$
- 

## A.7 Notas bibliográficas

Os resultados apresentados na Seção A.2 foram retirados do Capítulo 8 do livro de Jean Gallier [52]. A descrição do teste de orientação da Seção A.3 foi fortemente baseado no livro de Langetepe e Zachmann [72]. Uma explicação alternativa do teste de orientação pode ser encontrada em [97]. Os predicados e algoritmos discutidos nas seções A.4 foram retirados do livro de Joseph O’Rourke [97]. A discussão sobre o predicado *incircle* também foi baseada em [72].