ADAPTATION OF THRESH AND GRIND FOR HPD

J.M. HOWIE

CERN, Geneva

There are three major changes which need to be made to the THRESH-GRIND system of programs in order that the data from the HPD might be processed satisfactorily. These modifications will be described under the following general headings.

1. Input to THRESH
2. Reconstruction of the Vertices
3. The Labelling System

THRESH is the program which makes the geometrical reconstruction of the tracks in space, and GRIND is the kinematics program. In order to distinguish between the two versions of THRESH, ie. the versions before and after modification, I will refer to the unmodified version as "THRESH", and to the modified version for the HPD as "HPD THRESH".

1. Input

The input to THRESH can be split up into two parts. Firstly there is the basic geometry data necessary for any reconstruction in space eg. camera coordinates, fiducial coordinates, refractive indices etc. This data is contained in TITLE 1 and is read in at the beginning of THRESH. (TITLES 2,3, 4 and 5 are other sets of basic data which are used in GRIND). Since this data depends on the particular experiment and chamber used, it will not normally have to be changed during a THRESH run.

The main part of the input to THRESH is of course the measurement data for each event. This is in the form of a series of BCD records on magnetic tape, the first record of an event being an idenfication record containing the word 444444 (ie. 4's record). The last event on the input tape is terminated by a 7's record (ie. the BCD word 777777). The data for one event is read into THRESH by a Fortran Subroutine (EVENT 2), and then processed completely by THRESH, before the data for the next event is read into store.

For HPD THRESH, the TITLE 1 data is read in as before, but the input data, which is the output tape from HAZE is in an entirely different format from the normal input to THRESH. Therefore a new input subroutine (EVENT 2) has been written for HPD THRESH. This has been written in FAP because there is a large amount of unpacking of words to be done. The input tape for HPD THRESH has the following form :

```
LABEL RECORD (BCD)
E.O.F.
10,000 RECORD (BINARY)
11,000 RECORD view 3 (BINARY) }
11,000 RECORD view 2 (BINARY) }  for each event
11,000 RECORD view 1 (BINARY) }
          etc.
```

The 10,000 and 11,000 records are so called because the first words of these records are $(10000)_{octal}$ and $(11000)_{octal}$ respectively. The 10,000 record contains a few identification words, but these are not used at all by THRESH. A 10,000 record is written by HAZE every time the roll of film is changed. The reason why the views are given in the order 3,2, 1 is because the views for each event have to be merged together within HAZE, and this is the final order when the merging process is completed. Because of the new input format for HPD THRESH, one or two additional small changes have had to be made.

a) <u>Coordinates</u>

In THRESH, all coordinates (ie. of points on tracks, or fiducials or vertices) are read into a two-dimensional array BSTR (2,1500) ie. BSTR = $\begin{pmatrix} X_1 & X_2 & X_3 & X_4 & \text{-----} \\ Y_1 & Y_2 & Y_3 & Y_4 & \text{-----} \end{pmatrix}$

In the HPD THRESH input, the X and Y coordinates of a point are packed into one word.

```
          X           Y
ie  |-----------+-----------|
    0           17          35
```

These are stored in the array BSTR as if it were a one dimensional array

ie BSTR = $\begin{pmatrix} X_1 & Y_1 & X_3 & Y_3 & X_5 & Y_5 & \text{-----} \\ X_2 & Y_2 & X_4 & Y_4 & X_6 & Y_6 & \text{-----} \end{pmatrix}$

To make the HPD THRESH as compatible as possible with THRESH, a function FNBSTR has been introduced, where FNBSTR (I, J) is defined as follows :
If I = 1, the function unpacks the first half of the J^{th} word of BSTR, and if I = 2 it unpacks the second half of the word.

Hence one merely has to replace BSTR (I,J) by FNBSTR (I,J) everywhere it appears in the program.

b) Fiducials

The labelling notation used in the THRESH-GRIND system is such that track vertices are denoted by letters and fiducial marks by numbers. Allowance is made for up to 10 fiducial marks and these are numbered 0,1, ---9. In the normal THRESH input, fiducials are recognised by their numbers, and the next four numbers following the fiducial number give the measurements for this fiducial in views 1,2,3,4.

In the output from HAZE there are facilities for the measurements of 6 fiducials on each view, but no indication is given of what the number of each fiducial is. With HPD THRESH this information is now provided to the program through TITLE 1. In TITLE 1, four words of 6 BCD characters each are read into store, where the words denote the views 1,2,3,4 respectively and the 6 characters (which are all numbers) denote the order in which the fiducials are measured in each view.

e.g. suppose the third word of this block is 247389, then this means that, in view 3

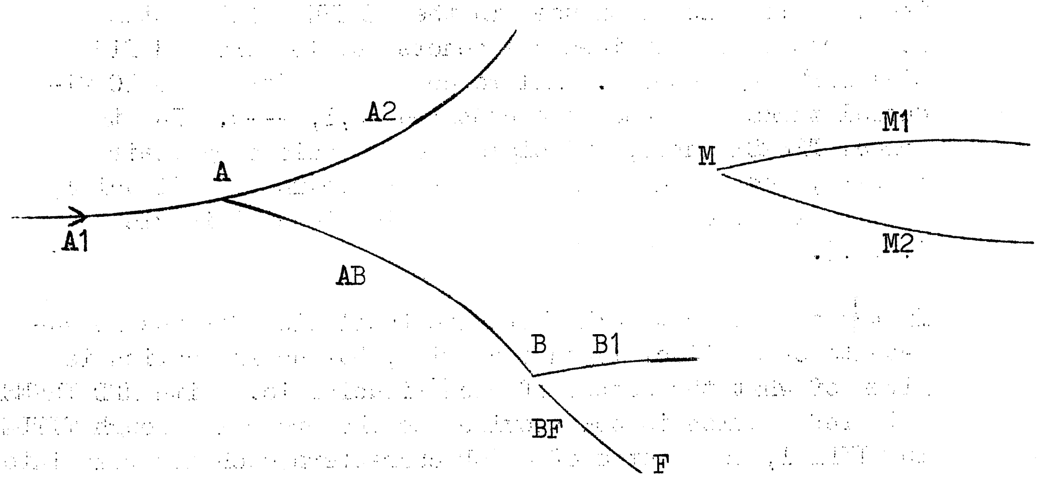
the first	fiducial	measured	is	fiducial	2
the second	"	"	"	"	4
the third	"	"	"	"	7

etc.

c) Track labels

In THRESH, tracks are labelled by two characters, either two different letters or else a letter and a number. Two different letters, labels of the type AB, specify a track which connects the two vertices A and B. Vertex A will be the beginning point of the track and vertex B will either be a point of secondary scattering, a decay point or else the stopping point in the chamber of this track. Labels of the type A2 specify a track starting at vertex A, which either passes out of the chamber, or else the end of the track is of no importance.

e.g. the network of tracks shown below could be labelled as indicated.



With the HPD the vertices and tracks are given numbers at the scan table, a vertex being given a number between 1 and 9, and a track being given a number between 0 and 99. On the HAZE output the label of each track is given by a BCD word $(V_2 - - V_1\ T\ T)$.

where V_1 is the vertex of the track

V_2 is the connecting vertex if it exists

and T T are the two digits of the track number

In HPD THRESH the vertex numbers are converted to letters by adding $(20)_{octal}$ to the number in store

e.g. $1 = (01)_o \longrightarrow (21)_o = A$

$9 = (11)_o \longrightarrow (31)_o = I$

The track is then labelled as in normal THRESH, i.e. by the two letters if there is a connecting vertex, the track number in this case being ignored, or else by a letter and a number, the number being the track number already given (T T).

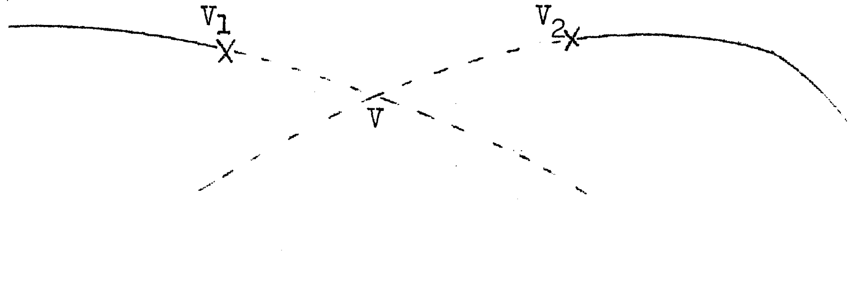
d) Incoming tracks

The output of HAZE is such that measurements on a track are given in the order of the direction of motion of the particle. Therefore a track coming into a vertex, (A say), will have the vertex point A at the end of the track, whilst all the outgoing tracks will have the vertex A at the beginning of the track. For the geometrical reconstruction, THRESH expects to have all tracks starting at the vertex point, therefore for HPD THRESH all incoming tracks have to be reversed. There is no difficulty in identifying these tracks because on the scan table all incoming tracks are given a track number of zero.

2. Reconstruction of the vertex

In the input to THRESH, the track vertices are usually given as measured points on each view and hence are easily reconstructed. THRESH then fits helices to all tracks and finally transfers the geometrical information so determined into GRIND; i.e. the coordinates of the vertices and the parameters of the track helices. For the case when a vertex is not given as a measured point, THRESH, for each of the tracks leading from this vertex, reconstructs the first measured point on one of the views as the vertex of the track. Hence a separate determination of the vertex is made for each track leaving the vertex. THRESH then fits helices to all the tracks as before, but does not attempt to make a more accurate determination of any unmeasured vertices. This is done in GRIND using linear extrapolation. As an example we consider the case of just two tracks.

Let V_1 be the determination of the vertex on the first track and V_2 be the determination of the vertex on the second track.



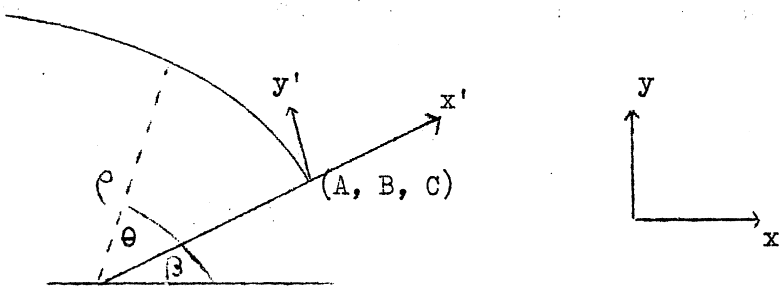
The actual vertex V of the two tracks is found by linearly extrapolating backwards from the approximate vertices V_1 and V_2, and solving by least squares to determine the best "point of intersection" of the two lines. The fact that only linear extrapolation is used to construct the vertex point in this case is rather unsatisfactory, but on the whole not too serious, since THRESH was primarily expected to deal with measured vertices, and it is usual practice to measure all relevent vertices at the IEP.

With the HPD though, it is not possible to measure the track vertices satisfactorily and hence no vertices are given as measured points. In HPD THRESH therefore, a better method for constructing the vertex points has been incorporated, which will provide more accurate determinations of the vertices.

The difference between the two methods lies in the final reconstruction of the vertex. We assume we have reached the stage where an approximate vertex has been constructed for each track, and also a helix has been fitted to each track. To reconstruct a particular vertex, we now attempt to find the best "point of intersection" of all the track helices which emit from this vertex, i.e. we extrapolate back along the helices instead of doing a simple linear extrapolation.

In the explanation of the method I shall use the following notation :

1) The basic system of coordinates is x, y, z.
2) The axes of the helices are all parallel to the z-axis.
3) The coordinates of the approximate vertex on a track are A,B,C.
4) The equation of a helix is given with respect to an axis system x' y' z', where the origin of these coordinates is transferred to the vertex point (A, B, C), and the axes are rotated through an angle β in the (x, y)-plane, so that the x'-axis is pointing outwards along the normal to the helix, i.e.



the equation of the helix is

$$x' = \rho\,(\cos\theta - 1)$$
$$y' = \rho\ \sin\theta \qquad (1)$$
$$z' = \rho\theta\tan\alpha$$

where ρ is the radius and α is the dip angle. θ is the parameter which defines the points on the helix; $\theta = 0$ being the origin, i.e. the point (A, B, C).

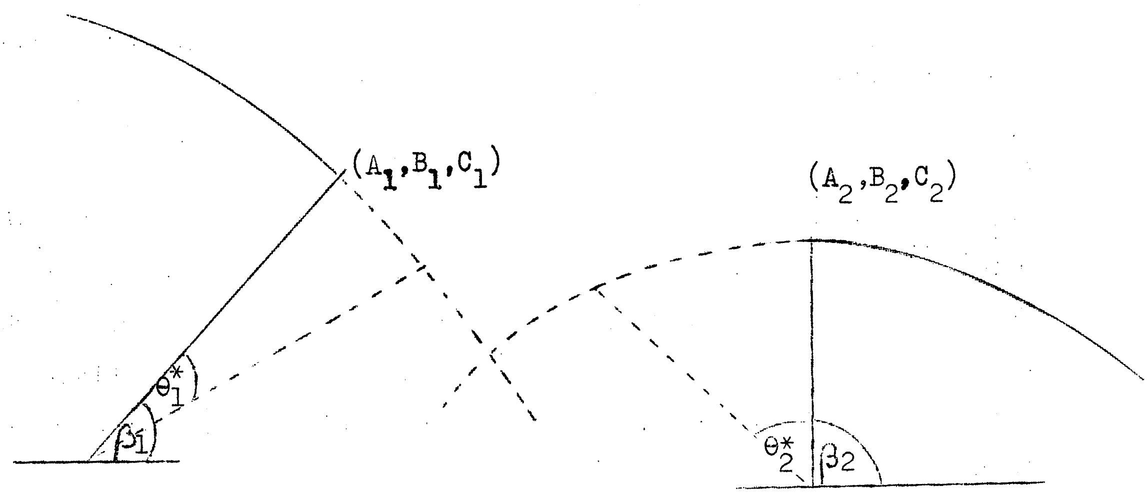
Referred to the basic coordinate system, a point on the helix is given by

$$x = A + x'\cos\beta - y'\sin\beta$$
$$y = B + x'\sin\beta + y'\cos\beta \qquad (2)$$
$$z = C + z'$$

Since θ is the only parameter of the helix which varies, we may express a point on the helix as a function of θ, i.e.

$$x = x\,(\theta)$$
$$y = y\,(\theta) \qquad (3)$$
$$z = z\,(\theta)$$

For the sake of simplicity we consider now the case of one vertex and just two tracks emitting from the vertex. Let quantities with suffix 1 refer to one track and those with suffix 2 refer to the second track.



We suppose that $\theta_1 = \theta_1^*$ and $\theta_2 = \theta_2^*$ are approximations for the true vertex point on tracks 1 and 2 respectively, and that the true values of θ_1 and θ_2 are

$$\theta_1 = \theta_1^* + \Delta\theta_1 \qquad\qquad \theta_2 = \theta_2^* + \Delta\theta_2$$

By equating the x,y and z coordinates of the two points on the respective helices we have from Eq (3) that

$$\left.\begin{aligned} x_1(\theta_1^* + \Delta\theta_1) &= x_2(\theta_2^* + \Delta\theta_2) \\ y_1(\theta_1^* + \Delta\theta_1) &= y_2(\theta_2^* + \Delta\theta_2) \\ z_1(\theta_1^* + \Delta\theta_1) &= z_2(\theta_2^* + \Delta\theta_2) \end{aligned}\right\} \qquad (4)$$

Now by expanding the left and right hand sides of the above equations in Taylor series about θ_1^* and θ_2^* respectively, and ignoring powers of $\Delta\theta_1$ and $\Delta\theta_2$ greater than the first we arrive at a linear set of equations for the unknown $\Delta\theta_1$ and $\Delta\theta_2$. i.e.

$$\left.\begin{aligned} x_1(\theta_1^*) + \frac{dx_1}{d\theta}(\theta_1^*)\Delta\theta_1 &= x_2(\theta_2^*) + \frac{dx_2}{d\theta}(\theta_2^*)\Delta\theta_2 \\ y_1(\theta_1^*) + \frac{dy_1}{d\theta}(\theta_1^*)\Delta\theta_1 &= y_2(\theta_2^*) + \frac{dy_2}{d\theta}(\theta_2^*)\Delta\theta_2 \\ z_1(\theta_1^*) + \frac{dz_1}{d\theta}(\theta_1^*)\Delta\theta_1 &= z_2(\theta_2^*) + \frac{dz_2}{d\theta}(\theta_2^*)\Delta\theta_2 \end{aligned}\right\} \qquad (5)$$

The coefficients in the above equations are determined from equations (1) and (2) and the system solved by least squares for the unknowns $\Delta\theta_1$ and $\Delta\theta_2$. We now replace θ_1^* by $\theta_1^* + \Delta\theta_1$ and θ_2^* by $\theta_2^* + \Delta\theta_2$ and repeat the process to determine further increments $\Delta\theta_1$ and $\Delta\theta_2$. The iteration converges when the values of $\Delta\theta_1$ and $\Delta\theta_2$ become small enough. (For the first approximation we put $\theta_1^* = \theta_2^* = 0$). When the iteration converges, we then have two approximations for the vertex given by the final value of θ_1^* for track 1 and the final value of θ_2^* for track 2. The true vertex point is therefore taken to be the mean of these two approximations.

The above analysis is only for the simplified model of two tracks, but the method can be extended in an obvious way to deal with more than two tracks. In this case we would have to consider all possible intersections of tracks, which would unfortunately mean a large increase in the number of equations. More precisely if there were n tracks, then we would have a system of

$$\frac{3 \times n \times (n-1)}{2} \text{ equations.}$$

Because of this large increase in the number of equations, (eg, for 6 tracks we would have 45 equations), not all combinations are considered in the HPD THRESH routine. Instead, combinations are taken such that, if there are more than two tracks, then each track is only used twice. The true vertex point is, of course taken to be the average of all the separate determinations. At the moment no weighting is used in the calculation of this average, but it is intended to prescribe weights depending on the angles between the tracks at the vertex. The implication is that an intersection point, determined by two tracks which make an angle of about 90°, will be more reliable than one determined by two tracks with only a small angle between them.

A further point concerning the above analysis is that we have made the implicit assumption that all the parameters occuring in the equations (i.e. ρ, α, β, A,B,C) have been determined exactly and are without error. This is of course not true, since for each track, the values of ρ, α, β, A,B and C are themselves determined from least squares equations and all have errors. Therefore in our equations (5), we also have 6 measured variables for each helix as well as the two unknowns $\Delta\theta_1$ and $\Delta\theta_2$. The solution of these equations can be determined by using a more generalized method of least squares. Details of the method can be found in various publications on statistics or least squares methods, but for a compact account see Böck (1960) [1]. By introducing these errors on the measured variables, one in fact does weight the solution in favour of the good measurements, but the analysis becomes much more complicated. So far at CERN we have not made any elaborate analysis of this sort. To safeguard against using tracks which have been badly measured though, we only use tracks which have converged in the least squares helix fitting procedure.

3. The labelling system

The third major modification which has to be made to THRESH is caused by the labelling system. In normal THRESH, the labels which are given to the vertices and tracks at the scan table are not allocated arbitrarily, but are used to carry important information from the measuring machines into THRESH and GRIND. Depending on the different kinds of interactions between which one wishes to distinguish, one sets up a series of interaction classes. To the classes are allocated letters and numbers which may be used as labels for vertices and tracks of a particular interaction. In this way one can, for instance, distinguish between a decay point of a neutral particle or a point of secondary scattering. There are also other classes which, for example, define a vertex as an end point of a stopping track, or a track as a straight track. GRIND uses these label classes to give each vertex and each track a number which is called the "Nature" of the point or track. The numerical value of the "nature" of the point or track is equivalent to defining its class or type, and hence one sees that the labelling system is just a means of indicating this "nature".

With the HPD, the labels are prescribed at the Milady scan table, and depend entirely on the order in which the points and tracks are measured. Since the labelling system is now to a large extent a function of the measuring machine rather than the operator, the labels cannot be used in the same manner as before. The "nature" of the points and tracks must therefore be determined in some other way. This in fact is not too difficult, as some of the information which used to be transmitted to the programs through the labelling system is now contained directly in the input data to HPD THRESH. This additional information arises from decisions which are made at the scan table and is as follows :

1) whether a track is straight or not,
2) whether a track stops within the chamber,
3) whether a track should be considered to have zero range,
4) a mass code and a charge code are also given, by which the scanner may indicate either the mass or charge of a particle (or both), if he really wishes.

We now give a list of all the properties which are used to define the "nature" of a point or a track and we also indicate how each property is determined in HPD THRESH.

a) <u>Tracks</u>

1) stopping track : specified in HPD THRESH input
2) straight track : specified in HPD THRESH input
3) zero-range track : specified in HPD THRESH input,
4) beam track : the beam track will always be the first track measured and will also have a

track number of zero. If there was no beam track then the first track would not have a label zero.

5) connecting track : i.e. a track which has a vertex at both ends; specified in HPD THRESH input.

b) Points

The "nature" of the points can not be determined quite so directly as that of the tracks. From the input data to HPD THRESH we know whether a vertex is charged or uncharged (i.e. if there is an incoming track or not), and we can also determine the total number of measured tracks leaving each vertex, from the labelling system.

By using this information we can distinguish between the various interaction classes used for points. The classes generally used in GRIND are as follows :

1) a) Points of beam interactions (with an odd number of measured tracks).
   b) Points of beam interactions (with an even number of measured tracks). This point will always be the first measured point anyway.

2) Decay points of charged particles (with two measured tracks).
3) Decay points of neutral particles (with two measured tracks).
4) Secondary scattering of particles (with three measured tracks).

Reference

1. BOCK R. (1960) Application of a generalized method of least squares for kinematical analysis of tracks in bubble chambers. CERN 60-30.

DISCUSSION

BUDDE: I would like to ask whether the extrapolation to the apex isn't a dangerous procedure, especially for high energy V^os. If you find the apex by extrapolation, as long as it is on the line of flight, the kinematics is still alright, but if now you want to compute decay corrections, you need to know the actual path length which the V^o travelled. Now I think the error introduced by extrapolation will be a systematic one, and this means the path lengths you find will have a tendency to be too short.

HOWIE: Yes, it was this kind of phenomena which I indicated could be accounted for by weighting but obviously if one has only the two secondaries of an energetic V^o, the method would be unsatisfactory. Therefore in particular cases this procedure would have to be modified. Speaking generally though, I feel that this would be a good way of determining the vertex, because one doesn't have to know the beginning point. One just takes a good point on the helix and if one has a good fit, then one should obtain a good fit for the vertex.

BURD: I take it that since you didn't mention anything about it, you don't have time sharing problems ?

HOWIE: At the moment we are not considering this in detail. I think we can cut down the space taken by THRESH a fair amount. There are certain routines we can leave out, for example, there is a routine which constructs vertex points from actual measured points, and another routine which orders the measurements on a track (with the HPD we can assume that the measurements will be given in the correct order).

BURD: And you think you will share time on the 709 * between HAZE and THRESH ?

HOWIE: We hope so when we have the system running on the 7090.

TYCKO: You say you have a routine in THRESH which reconstructs vertex points from measured points. Why don't you continue to use that technique. Have you considered reconstructing unmeasured vertex points from tracks using the measurements themselves on each view and then going into space ?

HOWIE: This is an alternative method, but I feel a spatial reconstruction will be generally more accurate.

---

* CERN will have a 7090 as from October 1963.

BURD: It is a much simpler alternative. But you say it is better to do it in space.

HOWIE: If you can get good fits to the helices, then, except for these cases where you only have 2 tracks at a very small angle, I feel you must get a good vertex fit.

CALKIN: I don't see that there can be a real difference. You are probably looking at the same thing in a different coordinate system.

HUMPHREY: There might be a danger in trying to reconstruct the point on the film in that for strongly dipping tracks you can get very peculiar shapes on the film which may be very difficult to extrapolate to the vertex. Dick Hartung has developed a method of fitting all tracks simultaneously in which the vertex point automatically falls out but it does involve solving for more variables. I was wondering if you had looked into this method?

HOWIE: I have discussed this approach with a statistician at CERN, but we haven't gone into any details. He thought it was feasible. I think you have to solve the problem by the method of maximum likelihood.

HUMPHREY: I think Dick Hartung has actually written a program which does this.

HALL: Could you say what your least squares criterion is with the present scheme? In other words what are you minimising?

HOWIE: We have an overdetermined set of linear equations and we simply minimize the residuals. This is, physically equivalent to finding the values of θ_1 and θ_2 which give you the shortest distance between the two helices.