

Technical Report TR-10-14
Computational Science Laboratory
Department of Computer Science
Virginia Tech

Space-time adaptive solution of inverse problems
with the discrete adjoint method

Mihai Alexe Adrian Sandu

November 2010



Space-time adaptive solution of inverse problems with the discrete adjoint method

Mihai Alexe and Adrian Sandu

Computational Science Laboratory
Department of Computer Science
Virginia Polytechnic and State University
Blacksburg, VA, 24060, USA

E-mail: mihai@vt.edu, sandu@cs.vt.edu

Abstract. Adaptivity in both space and time has become the norm for solving problems modeled by partial differential equations. The size of the discretized problem makes uniformly refined grids computationally prohibitive. Adaptive refinement of meshes and time steps allows to capture the phenomena of interest while keeping the cost of a simulation tractable on the current hardware. Many fields in science and engineering require the solution of inverse problems where parameters for a given model are estimated based on available measurement information. In contrast to forward (regular) simulations, inverse problems have not extensively benefited from the adaptive solver technology. Previous research in inverse problems has focused mainly on the continuous approach to calculate sensitivities, and has typically employed fixed time and space meshes in the solution process. Inverse problem solvers that make exclusive use of uniform or static meshes avoid complications such as the differentiation of mesh motion equations, or inconsistencies in the sensitivity equations between subdomains with different refinement levels. However, this comes at the cost of low computational efficiency. More efficient computations are possible through judicious use of adaptive mesh refinement, adaptive time steps, and the discrete adjoint method.

This paper develops a framework for the construction and analysis of discrete adjoint sensitivities in the context of time dependent, adaptive grid, adaptive step models. Discrete adjoints are attractive in practice since they can be generated with low effort using automatic differentiation. However, this approach brings several important challenges. The adjoint of the forward numerical scheme may be inconsistent with the continuous adjoint equations. A reduction in accuracy of the discrete adjoint sensitivities may appear due to the intergrid transfer operators. Moreover, the optimization algorithm may need to accommodate state and gradient vectors whose dimensions change between iterations. This work shows that several of these potential issues can be avoided for the discontinuous Galerkin (DG) method. The adjoint model development is considerably simplified by decoupling the adaptive mesh refinement mechanism from the forward model solver, and by selectively applying automatic differentiation on individual algorithms.

In forward models discontinuous Galerkin discretizations can efficiently handle high orders of accuracy, h/p -refinement, and parallel computation. The analysis reveals that this approach, paired with Runge Kutta time stepping, is well suited for the adaptive solutions of inverse problems. The usefulness of discrete discontinuous Galerkin adjoints is illustrated on a two-dimensional adaptive data assimilation problem.

1. Introduction and background

Inverse problems (IPs) consist in using *a priori* available measurements to infer the values of the defining parameters for a given model. IPs arise in various applications of engineering and mathematics, e.g., seismography, meteorology, oceanography, medical imaging, systems biology, and fluid dynamics (see, e.g., [1, 2, 3, 4, 5, 6]). IPs are usually described as constrained optimization problems, where the constraints are ordinary (ODE) or partial differential equations (PDEs). Alternatively, the problem can be stated in terms of probability densities. Using Bayes' theorem we arrive at the optimization formulation [7]. State of the art PDE solvers use adjoint-driven adaptive space-time refinement, and other dynamic computational patterns such as upwinding, slope or flux limiting, interpolations, extrapolations, variable order approximations (*p*-refinement), moving meshes, etc. (see, e.g., [8], and references therein). Space-time adaptivity controls the numerical errors introduced by the spatial and temporal discretizations, preserves the quality of the solution, and helps maximize solver efficiency. In contrast, most attempts at solving inverse problems have favored non-adaptive methods. There has been recently a growing trend of research into the use of adaptive inverse problem solvers (see, e.g., [9, 10, 11, 12, 13, 14, 15]), but there remains a considerable gap between the state of the art forward model solvers and the strategies used in inverse problems. This discrepancy is mainly caused by the difficulties with obtaining and using derivative information in adaptive simulations. This work aims to further close this gap, by demonstrating the feasibility and efficiency of discrete adjoints for time and space-dependent inverse problems.

The discrete adjoint method [16] is very attractive in practice since the discrete adjoint model can be generated with minimal user intervention via automatic differentiation [17]. Given the complexity of current numerical solvers, automatic generation of the adjoint code is a significant advantage over both finite-difference, and hand-coded (analytical) derivatives. Moreover, the discrete approach avoids problems with incompatible cost functionals [18, 19], and bypasses the usually cumbersome derivation of boundary conditions for the continuous adjoint system [11, 20]. We also note that discrete adjoints yield the exact gradient of the discretized cost functional under consideration, while the discretized solution of the continuous adjoint system is not the true gradient of anything, except in the limit of the discretization [16]. However, discretizations of the continuous adjoint problem may sometimes lead to better inversion results, despite poorer gradient consistency than that obtained with the adjoint of the primal discretization [21].

As previously mentioned, most research efforts to date on large scale inverse problems have made use of the continuous sensitivity approach. Fang *et al.* have developed a fully adaptive 3D finite-element ocean model equipped with adjoint sensitivity analysis [12, 14, 15, 22, 23, 24]. The adjoint equations are discretized on spatially-variable grids, with the forward variables interpolated onto the adjoint mesh wherever needed. For faster forward and backward simulations, the POD method [22]

is used to reduce the state space of the model and its adjoint, while capturing their essential dynamics.

The construction and properties of discrete adjoints on adaptive meshes are not yet fully understood. Previous research efforts on adjoints for adaptive mesh refinement problems include the work of Li and Petzold [11]. The authors attempt to combine the advantages of both approaches to adjoint sensitivity, DA (discretization of the adjoint) and AD (adjoint of the discretization) into their ADDA method. However, their approach uses discrete adjoints only for a fixed grid inside an artificial boundary layer, where the DA is assumed to be consistent, while employing the continuous adjoint method in the interior of the domain [11]. This avoids the difficult determination of suitable boundary conditions for the adjoint PDE.

There is a significant body of work in sensitivity analysis for large-scale, nonlinear, steady-state problems. Bangerth [9] introduces a framework for the solution of parameter estimation problems on adaptive grids using finite element discretizations. The continuous optimality system is solved using an *all-at-once* approach in a function space setting. This formulation allows the use of separate (nested) meshes for the model state, the adjoint solution, and the inversion variables. Marta [25] applies automatic differentiation on selected parts of two 3D unstructured-mesh CFD solvers to derive a consistent discrete adjoint model. Mavriplis [26] uses the discrete adjoint method in a Navier–Stokes multigrid optimization problem, discretized on three-dimensional unstructured meshes.

Inverse problems with discontinuous solutions benefit from error-based adaptive mesh refinement; the quality of the discrete adjoint solution improves when more grid points (or elements) are generated near the shock or jump in the forward solution, as observed by Giles and Ulbrich [20, 27, 28]. Adaptive meshing also allows for regularization by discretization [29]: we can control the variation in the inversion variables by error-based mesh coarsening and refinement, and through local variations in the approximation order of the inverse problem solution.

Our paper demonstrates the feasibility and efficiency of the discrete adjoint method for adaptive time-dependent inverse problems. While the efforts of other authors have considered almost exclusively on steady-state problems (with some exceptions, see [22, 30]), we examine discrete adjoints for evolution problems, and highlight the benefits of both time and space adaptivity. Recent advances in adjoint computation strategies have made reversal of time-dependent codes computationally feasible (see [31], and references therein). The main computational advantage of the discrete adjoint approach is that the adjoint model code can be obtained through automatic differentiation, hence saving a significant amount of software development effort. Automatic differentiation tools are available for all the major programming languages used in scientific computing [32, 33, 34, 35, 36]. Specialized finite element software such as `deal.II` allow the mesh adaptation to be performed in a transparent fashion, and independent from the numerical core of the solution algorithm. Thus, automatic differentiation can be used in a targeted fashion, such that we obtain the adjoint of only the time marching procedure,

and that of the (linearized) right hand side of the discrete model equations. This simplifies both the derivation and verification of the discrete adjoint model.

We first present the general form of the discrete adjoint method, and then discuss the issues that arise with adaptive mesh and time refinement. For spectral-type numerical methods such as discontinuous Galerkin (a method particularly amenable to space-time adaptivity), the discrete intergrid operators are implemented through orthogonal projections. Thus, there are no inconsistencies introduced through the use of the adjoint interpolation and restriction operators obtained by automatic differentiation. The analysis is then extended to general meshes, where the same operator properties are verified. However, this orthogonality property is not recovered for all numerical methods. The adjoint analysis of finite volume mesh transfer operators shows the adjoint of high-order interpolation (through solution averaging) to be only first order accurate in the general case.

We then focus on the concept of adjoint consistency for time-dependent discontinuous Galerkin discretizations. The concept of adjoint consistency, defined, e.g., in [37, 38] for elliptic problems, plays an important role in the analysis of the dual (adjoint) problem solution, in the convergence of the primal approximation, as well as in the accuracy of the target functional under consideration. Building on previous duality results for time [39, 40, 41] and space discretizations [37, 42], we develop an unified framework for investigating dual consistency of discretizations for a general type of time-dependent PDEs.

1.1. Organization

This paper is structured as follows. Section 2 discusses the adaptive inverse problem framework and the difficulties associated with adaptivity and the discrete adjoint method. In section 3 we review the mathematical foundation of the discontinuous Galerkin (DG) method. Section 4 concerns intergrid transfer operators for Galerkin-type discretizations. We remark that both h - and p -refinement with structured AMR are performed through orthogonal projections, hence the accuracy of the discrete adjoint solution will not be affected by intergrid solution transfer operators (beyond the intrinsic loss in accuracy associated with mesh coarsening). Section 5 gives a brief overview of the finite-volume method, and discusses the interpolation and restriction operators for h -refinement obtained through polynomial solution reconstructions. Subsequently, section 6 discusses in detail the derivation of formal adjoint systems for general differential problems that obey a given set of compatibility conditions. Section 7 considers the dual consistency of time discretizations in Runge-Kutta DG methods. The accuracy and computational advantages of the discrete adjoint approach are demonstrated on a two-dimensional numerical test problem in section 8. Finally, section 9 summarizes the conclusions, and discusses opportunities for further research.

2. The adaptive inverse problem framework

Consider a dynamical model whose time and space evolution is described by the following initial boundary-value problem (IBVP):

$$\begin{aligned} \mathcal{F}(\mathbf{u}, \mathbf{u}_t, \mathbf{u}_x, \mathbf{u}_{xx}, \dots, \mathbf{p}, t, \mathbf{x}) &= 0, & \mathbf{x} \in \Omega, t^0 \leq t \leq t^N \\ \mathcal{F}_B(\mathbf{u}, \mathbf{u}_n, \dots, \mathbf{p}, t, \mathbf{x}) &= 0, & \mathbf{x} \in \Gamma := \partial\Omega \\ \mathbf{u}(t^0, \mathbf{x}) &= \mathbf{u}^0(\mathbf{p}), & \mathbf{x} \in \Omega. \end{aligned} \quad (1)$$

We discretize the equations (1) in time and space, and arrive at the discrete problem, written in residual form. Find $\mathbf{u}^{h,n} \in \mathcal{U}^{h,n}$ such that:

$$\begin{aligned} \mathcal{F}^{h,n}(\mathbf{u}^{h,n}(\mathbf{x}^h), \mathbf{u}^{h,n-1}(\mathbf{x}^h), \mathbf{p}^h, t^n) &= 0, & \mathbf{x}^h \in \Omega_n^h, 1 \leq n \leq N \\ \mathbf{u}^{h,0}(t^0, \mathbf{x}^h) &= \mathbf{u}^{h,0}, & \mathbf{x}^h \in \Omega_0^h. \end{aligned} \quad (2)$$

The term $\mathcal{F}^{h,n}$ incorporates both volume and boundary residuals.

To fix notation, we assume that (2) is solved on a sequence of time-dependent meshes Ω_n^h with $n = 0 \dots N$. For elliptic problems, the meshes can be associated with different iterations of the nonlinear discrete solver. For time dependent PDE discretizations, we can assume without loss of generality that one mesh is used per time step. All discrete variables will henceforth be marked with superscript h (for spatially discrete variables), and/or n (denoting the time discretization): $\mathbf{u}^{h,n}$ is the solution of (2) at time t^n . The *inversion variables* \mathbf{p}^h are a discrete set of parameters, whose exact values are unknown. For simplicity of exposition, we assume that the parameters \mathbf{p}^h are time-invariant. The vector $\bar{\mathbf{n}}$ is the unit normal to the discrete domain boundary $\partial\Omega_n^h$.

An inverse problem is typically formulated as a constrained minimization problem for a given objective functional \mathcal{J} . In the continuous approach, also called *differentiate-then-discretize*, one defines the minimization problem in terms of the analytical model formulation (1):

$$\text{Find } \mathbf{p}_* = \arg \min_{\mathbf{p} \in P^{\text{adm}}} \mathcal{J}(\mathbf{u}, \mathbf{p}), \text{ subject to (1)}. \quad (3)$$

We instead focus on the discrete approach, also called specifically *discretize-then-optimize*. The optimization problem is formulated in terms of the discretized model and the associated mesh variables:

$$\text{Find } \mathbf{p}_*^h = \arg \min_{\mathbf{p}^h \in P^{\text{adm}}} \mathcal{J}_h(\mathbf{u}^{h,0:N}, \mathbf{p}^h), \text{ subject to (2)}. \quad (4)$$

We note that both (3) and (4) usually contains bound constraints on the parameters: the physically valid values for the inversion variables define the admissible set P^{adm} . A well known example of inverse problem (IP) is four-dimensional data assimilation (henceforth referred to as 4D-Var), ubiquitous in numerical weather prediction [43]. For this type of problems, the cost functional \mathcal{J}_h quantifies the difference between the model state, and a set of *a priori* observations at selected points in the space-time domain. Direct or indirect measurements of the state of the atmosphere at a given time are incorporated in the cost function \mathcal{J}_h , and used to retrieve better approximations to the “true” initial or boundary conditions.

There are two prevalent approaches to solving the discrete optimization problem (4). The one-shot (or *all-in-one*) optimization approach [9] amounts to solving the first order optimality system for the Lagrangian functional [44]

$$\mathcal{L}_h := \mathcal{J}_h - \sum_{n=1}^N (\lambda^{h,n})^T \cdot \mathcal{F}^{h,n}(\mathbf{u}^{h,n}(\mathbf{x}^h), \mathbf{u}^{h,n-1}(\mathbf{x}^h), \mathbf{p}^h, t^n),$$

using a preconditioned linear solver. Here $\lambda^{h,n}$ are the Lagrange multipliers, which will be reinterpreted below as the adjoint or dual variables. The optimal values of the parameters are obtained through a single linear solve, hence the name of the method. The constraints on the inversion variables can be incorporated as additional equations in the optimality system. However, the size of the discretized optimality system can make its solution computationally prohibitive for large-scale problems. Finding a good quality preconditioner matrix for the full (or reduced) optimality system is also far from trivial, and is the subject of current research (see, e.g., [45, 46, 47]). Another approach, and the one we employ in this paper, is reduced-space optimization. We solve the forward and adjoint equations sequentially. The adjoint gradient provides us with a search direction for the optimization algorithm. A line search procedure [44] will then yield the next approximate solution to the inverse problem. Robust optimization algorithms such as L-BFGS-B [48] can transparently handle bound constraints.

2.1. Derivation of the discrete adjoint equations

Consider the steepest descent method [44] applied to minimize the cost functional \mathcal{J}_h . The solution update has the following form:

$$\mathbf{p}_{\text{new}}^h = \mathbf{p}_{\text{old}}^h - \alpha \left(\frac{d\mathcal{J}_h}{d\mathbf{p}^h} \right)^T,$$

where α is a suitably chosen step length, and the gradient (total derivative) reads:

$$\frac{d\mathcal{J}_h}{d\mathbf{p}^h} = \frac{\partial \mathcal{J}_h}{\partial \mathbf{p}^h} + \sum_{n=1}^N \frac{\partial \mathcal{J}_h}{\partial \mathbf{u}^{h,n}} \frac{\partial \mathbf{u}^{h,n}}{\partial \mathbf{p}^h}. \quad (5)$$

Since equation (2) is discretized in residual form, the implicit function theorem gives the following equation, also called the *tangent linear model* (henceforth referred to as the TLM):

$$\frac{\partial \mathcal{F}^{h,n}}{\partial \mathbf{u}^{h,n}} \frac{\partial \mathbf{u}^{h,n}}{\partial \mathbf{p}^h} + \frac{\partial \mathcal{F}^{h,n}}{\partial \mathbf{u}^{h,n-1}} \frac{\partial \mathbf{u}^{h,n-1}}{\partial \mathbf{p}^h} = - \frac{\partial \mathcal{F}^{h,n}}{\partial \mathbf{p}^h}, \quad n = 1 \dots N. \quad (6)$$

Hence, the space-time matrix formulation of the sensitivity equations reads as follows:

$$M \begin{bmatrix} \frac{\partial \mathbf{u}^{h,N}}{\partial \mathbf{p}^h} \\ \frac{\partial \mathbf{u}^{h,N-1}}{\partial \mathbf{p}^h} \\ \vdots \\ \frac{\partial \mathbf{u}^{h,1}}{\partial \mathbf{p}^h} \end{bmatrix} = \begin{bmatrix} - \frac{\partial \mathcal{F}^{h,N}}{\partial \mathbf{p}^h} \\ - \frac{\partial \mathcal{F}^{h,N-1}}{\partial \mathbf{p}^h} \\ \vdots \\ - \frac{\partial \mathcal{F}^{h,0}}{\partial \mathbf{p}^h} \end{bmatrix}, \quad (7)$$

where

$$M := \begin{bmatrix} \frac{\partial \mathcal{F}^{h,N}}{\partial \mathbf{u}^{h,N}} & \frac{\partial \mathcal{F}^{h,N}}{\partial \mathbf{u}^{h,N-1}} & 0 & \dots & 0 \\ 0 & \frac{\partial \mathcal{F}^{h,N-1}}{\partial \mathbf{u}^{h,N-1}} & \frac{\partial \mathcal{F}^{h,N-1}}{\partial \mathbf{u}^{h,N-2}} & 0 & \dots \\ 0 & 0 & \ddots & \ddots & \dots \\ 0 & \dots & \dots & 0 & \frac{\partial \mathcal{F}^{h,0}}{\partial \mathbf{u}^{h,1}} \end{bmatrix}.$$

The sensitivity matrices $\partial \mathbf{u}^{h,n} / \partial \mathbf{p}^h$ are very expensive to compute, since they scale with both the number of states and that of the control variables. When a new control variable is added, (6) needs to be solved anew. The *discrete adjoint method* [16] calculates the gradient (5) at a significantly lower cost than finite differences or forward sensitivities when the number of parameters is large compared to the number of outputs of interest. By defining the N discrete adjoint variables $\lambda^{h,n}$ as the solution components of the *discrete adjoint equation*, which reads:

$$M^T \begin{bmatrix} \lambda^{h,N} \\ \lambda^{h,N-1} \\ \vdots \\ \lambda^{h,1} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}^{h,N}} \right)^T \\ \left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}^{h,N-1}} \right)^T \\ \vdots \\ \left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}^{h,1}} \right)^T \end{bmatrix}, \quad (8)$$

we obtain

$$\frac{d\mathcal{J}_h}{d\mathbf{p}^h} = \frac{\partial \mathcal{J}_h}{\partial \mathbf{p}^h} - \sum_{n=1}^N (\lambda^{h,n})^T \frac{\partial \mathcal{F}^n}{\partial \mathbf{p}^h}. \quad (9)$$

Note that both the adjoint matrix M^T , and the right hand side of (8), are independent of the number of inversion variables. If the size of \mathbf{p}^h does not scale directly with the model state size \mathbf{u}^h , the cost of the discrete adjoint approach does not depend on the number of inversion variables. From the block lower bidiagonal structure of (8), we remark that the adjoint equations are solved backwards in time, i.e. from $n = N$ to $n = 0$. The size of the blocks may vary with n because of the mesh adaptation mechanism. This change in local solution size is easily accommodated by standard single-step Runge-Kutta-type ODE solvers such as the ones employed in this paper. Should a s -stage Runge-Kutta be used, the forward and adjoint systems (7)–(8) will have s non-zero block diagonals, and the computational cost scales accordingly. We do not consider linear multistep methods [49] in this paper, since they are not generally adjoint consistent [50].

2.2. Computational advantages of the discrete adjoint method

One frequently raised objection to the discrete adjoint method (besides consistency issues with the numerical discretization) is that the forward mesh is frequently

sub-optimal for the adjoint problem. While independent mesh refinement for the adjoint problem would enhance both the accuracy and the efficiency of the discrete adjoint solver, the additional complexity may increase the overall cost of the inversion process. The development effort required for the discretization of the continuous adjoint equations becomes significantly larger (there is no possibility of automating the computation). Features in the newly computed adjoint solutions may also be degraded when interpolating the gradient (9) to a reference optimization grid [11].

If we consider the Lagrangian interpretation of the adjoint variables, $\lambda^{h,n}$ tracks how well the forward solution $\mathbf{u}^{h,n}$ satisfies the state equation [29]. Hence, it is a reasonable choice to define both the $\mathbf{u}^{h,n}$ and $\lambda^{h,n}$ on the same mesh. Note from equation (9) that the discrete adjoint method can easily accommodate a different mesh for the inversion variables. While the discrete partial derivatives $\frac{\partial \mathcal{J}^h}{\partial \mathbf{p}^h}$ and $\frac{\partial \mathcal{F}^{h,n}}{\partial \mathbf{p}^h}$ are now defined on a separate “parameter mesh”, the adjoint system (8) and its solutions $\lambda^{h,n}$ do not change. This approach was proven to be very beneficial in practice [29, 51].

2.3. Mesh adaptivity and the discrete adjoint method

The mesh adaptivity raises several complications for the discrete adjoint approach. The issues that arise with adaptive temporal discretizations and automatic differentiation have been discussed in [52, 53]. We focus our analysis on the mesh refinement process. The mesh at a given nonlinear iteration and/or time step is refined or coarsened based on some *a posteriori* (residual-based) error estimation for the current solution approximation. Thus, the forward solver is required to transfer the solution between different meshes. We can write the forward solution process for (2) as:

$$\begin{aligned} \mathbf{u}^{h,0} &= \mathbf{u}^{h,0}(\mathbf{x}^h), \\ \mathbf{u}^{h,n+1} &= \mathcal{I}_{n \rightarrow n+1} \left(\mathcal{S}_{n \rightarrow n+1} \mathbf{u}^{h,n} \right), \quad 0 \leq n \leq N-1. \end{aligned}$$

Here $\mathcal{S}_{n \rightarrow n+1}(\cdot)$ is the nonlinear solution operator that advances $\mathbf{u}^{h,n}$ in time from t^n to t^{n+1} on Ω_n^h . The linear intergrid transfer operator is $\mathcal{I}_{n \rightarrow n+1} : \Omega_n^h \rightarrow \Omega_{n+1}^h$. Hence, the discrete adjoint procedure for solving (8), that may be generated through automatic differentiation, reads:

$$\begin{aligned} \lambda^{h,N} &= \lambda^{h,N}(\mathbf{x}^h) \\ \lambda^{h,n} &= \mathcal{S}'_{n+1 \rightarrow n} \left(\mathcal{I}_{n \rightarrow n+1}^T \lambda^{h,n+1} \right), \quad N-1 \geq n \geq 0, \end{aligned}$$

where the adjoint solution operator

$$\mathcal{S}'_{n+1 \rightarrow n}(\cdot) = - \left(\frac{\partial \mathcal{F}^n}{\partial \mathbf{u}^{h,n}} \right)^{-T} \left[\left(\frac{\partial \mathcal{F}^{n+1}}{\partial \mathbf{u}^{h,n}} \right)^T (\cdot) + \left(\frac{\partial \mathcal{J}_h}{\partial \mathbf{u}^{h,n}} \right)^T \right]$$

is the discrete adjoint (i.e., transpose) of the linearization of $\mathcal{S}_{n \rightarrow n+1}$. The grid transfer operator is $(\mathcal{I}_{n \rightarrow n+1})^T : \Omega_{n+1}^h \rightarrow \Omega_n^h$. Barring consistency issues in the discrete adjoint of the spatial discretization, or in the reversal of the time integration procedure, there

remains the question of the impact of the intergrid operators on the accuracy of the discrete adjoint solution. If

$$\mathcal{I}_{n+1 \rightarrow n} = \mathcal{C} \cdot (\mathcal{I}_{n \rightarrow n+1})^T, \quad (10)$$

with a constant \mathcal{C} independent of mesh and time step size (a valid assumption in most multigrid implementations, see e.g., [54]), the discrete adjoint accuracy is not compromised, and the adjoint code generated by AD can be used as-is (with a simple scaling to take into account the constant factor). In what follows we show that (10) is verified for h/p -adaptive DG (with $\mathcal{C} = 1$ and hierarchical spatial refinement), since both h - and p -refinement operators are orthogonal matrices.

Note that the solution transfer may also be done before $\mathcal{S}_{n \rightarrow n+1}(\cdot)$ is applied:

$$\begin{aligned} \mathbf{u}^{h,0} &= \mathbf{u}^{h,0}(\mathbf{x}^h), \\ \mathbf{u}^{h,n+1} &= \mathcal{S}_{n \rightarrow n+1} \left(\mathcal{I}_{n \rightarrow n+1} \mathbf{u}^{h,n} \right), \quad 0 \leq n \leq N-1. \\ \lambda^{h,N} &= \lambda^{h,N}(\mathbf{x}^h) \\ \lambda^{h,n} &= \mathcal{I}_{n \rightarrow n+1}^T \left(\mathcal{S}'_{n+1 \rightarrow n} \lambda^{h,n+1} \right), \quad N-1 \geq n \geq 0. \end{aligned}$$

With few modifications, the analysis above remains valid.

2.4. Multigrid optimization with the discrete adjoint method

When the inversion variables are spatially distributed, e.g., $\mathbf{p}^h = \mathbf{u}^{h,0}$, they are represented on a given mesh, e.g., Ω_0^h . As the optimization proceeds the shape of the field \mathbf{p}^h changes, and the grid may require adjustments in order to accurately represent the new \mathbf{p}^h . One possible solution is to apply grid refinement operations at the end of each optimization cycle. This approach has the disadvantage that the dimensions of the parameter and gradient vectors change during optimization. We do not know of any optimization algorithm that can handle a variable optimization space. To overcome this problem an alternative strategy is to define a fixed optimization mesh Θ_0^h , to project the parameter and gradient vectors from the computational mesh to the optimization one before the optimizer step, and to project the results back for the next function and gradient evaluation [55]. This approach has the disadvantage that the optimization mesh does not adapt to the changing solution profile; moreover, accuracy can be lost in the repeated interpolation process.

We propose to use a multigrid optimization approach [56]. The optimization grid is the computational grid Ω_0^H and is fixed throughout the inversion process. The optimizer need not accommodate changes in the discrete solution space size, and the code complexity is reduced as no interpolation onto a reference mesh is required. The optimization on the (coarser) Ω_0^H converges to obtain the solution \mathbf{p}^H . Through grid refinement operations both \mathbf{p}^H and $\mathbf{u}^{H,0}$ are projected onto a finer Ω_0^h grid that allows a more accurate representation of the fields of interest. The optimization process is then restarted on Ω_0^h , using the current best solution approximation $\mathbf{p}^h = \mathcal{I}_{\Omega_0^H \rightarrow \Omega_0^h} \mathbf{p}^H$ as the initial iterate.

2.5. Dual consistency for space-time discretizations

In the following we extend the dual consistency analysis in [38, 37] to space-time discretizations. Discretization of the time dimension by a Runge–Kutta quadrature [49] introduces several complications that preclude a simple extension of the spatial dual consistency concept, defined, e.g., in [38]. Consider the following framework for sensitivity analysis:

- The continuous primal problem is defined by (1)–(3).
- The tangent linear problem, i.e. the linearization of the continuous primal formulation:

$$\begin{aligned}\mathcal{F}'[\mathbf{u}, \mathbf{p}](\delta\mathbf{u}, \delta\mathbf{p}) &= 0 \\ \delta\mathcal{J} &:= \mathcal{J}'[\mathbf{u}, \mathbf{p}](\delta\mathbf{u}, \delta\mathbf{p}) .\end{aligned}\tag{11}$$

Here the $'$ symbol denotes the Fréchet derivative of \mathcal{F} , while the bracket notation indicates the state about which the linearization is performed. The direction of differentiation is $(\delta\mathbf{u}, \delta\mathbf{p})$, i.e. the full state vector of the tangent linear model. Note that $\mathcal{J}' = 0$ at the exact solution (\mathbf{u})

- The continuous \mathcal{L}^2 -dual problem:

$$\begin{aligned}\mathcal{F}'^*[\mathbf{u}, \mathbf{p}](\lambda) &= 0 \\ \mathcal{J}^a &:= \mathcal{J}'[\mathbf{u}, \mathbf{p}](\lambda, \mathbf{p}) .\end{aligned}\tag{12}$$

The $*$ superscript denotes an adjoint operator. Also, λ is the continuous dual variable, and $\mathcal{J}^a = \delta\mathcal{J}$ is the expression of Fréchet derivative of \mathcal{J} in terms of the dual variable.

- The discrete primal problem (2)–(4).
- The linearization of the discrete primal, i.e., the discrete tangent linear model

$$\begin{aligned}(\mathcal{F}^{h,n})'[\mathbf{u}^{h,n}, \mathbf{p}^h](\delta\mathbf{u}^{h,n}, \delta\mathbf{p}^h) &= 0 \\ \delta\mathcal{J}_h &:= (\mathcal{J}_h)'[\mathbf{u}^{h,n}, \mathbf{p}^h](\delta\mathbf{u}^{h,n}, \delta\mathbf{p}^h) .\end{aligned}\tag{13}$$

This is obtained directly by Fréchet differentiation of the discrete primal formulation, in the direction $(\delta\mathbf{u}^h, \delta\mathbf{p}^h)$.

- The discrete dual problem, obtained, e.g., through automatic differentiation, directly from the discrete tangent linear model (13):

$$\begin{aligned}(\mathcal{F}^{h,n})'^*[\mathbf{u}^{h,n}, \mathbf{p}^h](\lambda^{h,n}) &= 0 \\ \mathcal{J}_h^a &:= (\mathcal{J}_h)^{'*}[\mathbf{u}^{h,n}, \mathbf{p}^h](\lambda^{h,0:N}, \mathbf{p}^h) .\end{aligned}\tag{14}$$

In the discrete space formulation (14), the adjoint operator is equivalent to a matrix transpose.

2.5.1. *Consistency of the primal and tangent linear discretizations* The primal discretization is space-time consistent of order (α, β) , if, in the limit of the spatial and temporal discretizations, it holds that:

$$\begin{aligned} \left\| \mathcal{F}^{h,n}(\mathbf{u}, \mathbf{p}) \right\| &\sim \mathcal{O}(h^\alpha, \tau^\beta) \\ \left| \mathcal{J}^h(\mathbf{u}, \mathbf{p}) - \mathcal{J}(\mathbf{u}, \mathbf{p}) \right| &\sim \mathcal{O}(h^\alpha, \tau^\beta). \end{aligned} \quad (15)$$

We assume that the discrete primal variables are *a priori* consistent with their continuous counterparts. We will also refer to (15) as the residual consistency condition. Here \mathbf{u} and \mathbf{p} are the exact solutions to the continuous primal problem (1)–(3), while h and τ denote the size of the time and space meshes. Residual consistency is defined in a very similar manner for the discrete TLM (13):

$$\begin{aligned} \left\| \left(\mathcal{F}^{h,n} \right)' [\mathbf{u}, \mathbf{p}] (\delta \mathbf{u}, \delta \mathbf{p}) \right\| &\sim \mathcal{O}(h^\alpha, \tau^\beta) \\ \left| \left(\mathcal{J}_h \right)' [\mathbf{u}, \mathbf{p}] (\delta \mathbf{u}, \delta \mathbf{p}) - \mathcal{J}' [\mathbf{u}, \mathbf{p}] (\delta \mathbf{u}, \delta \mathbf{p}) \right| &\sim \mathcal{O}(h^\alpha, \tau^\beta). \end{aligned} \quad (16)$$

Here $\delta \mathbf{u}$ and $\delta \mathbf{p}$ are the exact solution to (11).

Consider now the convergence of the linearized primal variables. Note that this is a stronger property than (16), for it automatically implies residual consistency. Since the tangent linear problem (13) is linear in $(\delta \mathbf{u}^{h,n}, \delta \mathbf{p}^h)$, stability and residual consistency (if proven) imply convergence of the linearized variables, in the limit of both discretizations:

$$\left\| \delta \mathbf{u}^{h,n} - \delta \mathbf{u}(t^n) \right\| \sim \mathcal{O}(h^\alpha, \tau^\beta), \text{ as } h \rightarrow 0, \tau \rightarrow 0. \quad (17)$$

Note that neither stability nor consistency are automatically inherited by the tangent linear equations from the primal problem.

2.5.2. *Consistency of the dual discretization* Space-time consistency definitions for the dual discretization follow those for the primal problem and its linearization. We say that the adjoint discretization (14) is space-time consistent of order (α, β) if, in the limit of h and τ , the following relations hold:

$$\begin{aligned} \left\| \left(\mathcal{F}^{h,n} \right)'^* [\mathbf{u}, \mathbf{p}] (\lambda) \right\| &\sim \mathcal{O}(h^\alpha, \tau^\beta) \\ \left| \mathcal{J}_h^a(\mathbf{u}, \lambda, \mathbf{p}) - \mathcal{J}^a(\mathbf{u}, \lambda, \mathbf{p}) \right| &\sim \mathcal{O}(h^\alpha, \tau^\beta). \end{aligned} \quad (18)$$

The asymptotic order of consistency in the cost functional may be higher than (α, β) due to super-convergence effects, or dual post-processing of \mathcal{J}_h^a [57].

Equation (18) is equivalent to saying that the linearized primal discretization (13) is *dual consistent* of order (α, β) . Note that the dual discretization (14) automatically inherits the stability properties of the discrete tangent linear formulation (13). A crucial point in the discrete adjoint analysis is the convergence of the discrete adjoint variables. This does not follow automatically from (18). Instead, we need both stability of the dual formulation (14), *and* residual consistency (18). In that case, convergence of the $\lambda^{h,n}$ follows:

$$\left\| \lambda^{h,n} - \lambda(t^n) \right\| \sim \mathcal{O}(h^\alpha, \tau^\beta), \text{ as } h \rightarrow 0, \tau \rightarrow 0. \quad (19)$$

3. The discontinuous Galerkin method

We illustrate the general derivation of the discontinuous Galerkin (DG) method for a general hyperbolic conservation law:

$$\begin{aligned} \mathbf{u}_t + \nabla \cdot \mathbf{F}(\mathbf{u}) &= \mathbf{f}, & \mathbf{x} \in \Omega, t \in [0, T] \\ \mathbf{u}(t, \mathbf{x}) &= \mathbf{g}(t, \mathbf{x}), & \mathbf{x} \in \partial\Omega_{\text{in}} \\ \mathbf{u}(t=0, \mathbf{x}) &= \mathbf{u}^0(\mathbf{x}), \end{aligned} \quad (20)$$

The exact solution to (20) is $\mathbf{u} \in \mathcal{L}^2\{[0, T]; \mathcal{U}\}$, where \mathcal{U} is a function space that guarantees sufficient smoothness for \mathbf{u} . The discrete spatial mesh consists of polyhedral elements, i.e., $\Omega_n^h = \bigcup_{k=1}^{K_n} D_n^k$, and the inflow boundary $\partial\Omega_{\text{in}}$. In the *modal* DG formulation, the solution to (20) is approximated on a given element D_n^k by a truncated expansion of orthonormal polynomials (up to and including degree J):

$$\mathbf{u}^{h,n} \Big|_{D_n^k} = \sum_{j=0}^J \hat{\mathbf{u}}_j^{(k),n} \psi_j(\mathbf{x}^h), \quad (21)$$

with $\int_{D_n^k} \psi_i(\mathbf{x}) \psi_j(\mathbf{x}) d\mathbf{x} = \delta_{ij}$. The unknowns are then the time-dependent expansion coefficients $\hat{\mathbf{u}}_j^{(k),n}$, $j = 0 \dots J$. In the *nodal* DG formulation, the solution is given by:

$$\mathbf{u}^{h,n} \Big|_{D_n^k} = \mathbf{V} \left[\hat{\mathbf{u}}_0^{(k),n} \quad \hat{\mathbf{u}}_1^{(k),n} \quad \dots \quad \hat{\mathbf{u}}_J^{(k),n} \right]^T := \mathbf{V} \hat{\mathbf{u}}^{(k),n}. \quad (22)$$

Here \mathbf{V} is a block-diagonal Vandermonde interpolation matrix [58]. Note that if either the modal or nodal form of DG for (20) is proved to be adjoint consistent, the consistency of the other formulation follows from (22). The global vector of unknown expansion coefficients at t^n is

$$\hat{\mathbf{u}}^n := \begin{bmatrix} \hat{\mathbf{u}}^{(1),n} \\ \hat{\mathbf{u}}^{(2),n} \\ \vdots \\ \hat{\mathbf{u}}^{(K),n} \end{bmatrix}.$$

We follow the DG notation in [37] throughout this derivation. Since we will be concerned mainly with the space discretization, we omit the time dependency for the rest of this section. Let $\mathcal{U}_h \subset \mathcal{U}$ denote the discrete solution space. The test functions \mathbf{v}^h are assumed to be bounded in the \mathcal{H}^1 Sobolev norm on each mesh element. Given two neighboring elements D_-^k and D_+^k (with a common face or edge), we let $\mathbf{u}^\pm := \mathbf{u}|_{\partial D_\pm^k}$ denote the trace of \mathbf{u} taken from the interior of D_\pm^k , respectively. The jump in the solution over an edge (or face) is given by

$$\llbracket \mathbf{u}^h \rrbracket := \mathbf{u}_+^h \vec{\mathbf{n}}_+ + \mathbf{u}_-^h \vec{\mathbf{n}}_-,$$

whereas the average at $\mathbf{x} \in D_-^k \cap D_+^k$ is $\{\mathbf{u}\} := (\mathbf{u}_-^h + \mathbf{u}_+^h)/2$. On a boundary edge, we have that $\{\mathbf{u}^h\} := \mathbf{u}_+^h$, and $\llbracket \mathbf{u} \rrbracket := \mathbf{u}_+^h \vec{\mathbf{n}}_+$.

Define the following two discrete volume and boundary inner products:

$$\begin{aligned} \langle \mathbf{u}^h, \mathbf{v}^h \rangle_{D^h} &:= \int_{D^h} (\mathbf{u}^h)^T \mathbf{v}^h d\mathbf{x} \\ \langle \mathbf{u}^h, \mathbf{v}^h \rangle_{\partial D^h} &:= \int_{\partial D^h} (\mathbf{u}^h)^T \mathbf{v}^h ds, \end{aligned}$$

where ∂D^h is the boundary of the element D^k . A Galerkin projection onto the solution space \mathcal{U}_h , followed by an application of the divergence theorem, lead to the semi-discrete DG formulation: Find $\mathbf{u}^h \in \mathcal{U}_h$, such that for all $\mathbf{v}^h \in \mathcal{U}_h$ we have that

$$\sum_{D^k} \left(\left\langle \frac{d\mathbf{u}^h}{dt}, \mathbf{v}^h \right\rangle_{D^k} - \langle \mathbf{F}(\mathbf{u}^h), \nabla \mathbf{v}^h \rangle_{D^k} + \langle \mathbf{F}^*(\mathbf{u}_-^h, \mathbf{u}_+^h, \vec{\mathbf{n}}), \mathbf{v}^h \rangle_{\partial D^k} \right) - \sum_{D^k} \langle \mathbf{f}, \mathbf{v}^h \rangle_{D^k} = 0 .$$

The numerical flux $\mathbf{F}^{\text{DG}}(\mathbf{u}_-^h, \mathbf{u}_+^h, \vec{\mathbf{n}})$ is obtained through the solution of a Riemann problem at the boundary between two adjacent elements [58]. For Runge–Kutta DG methods, the time derivative is discretized through a Runge–Kutta quadrature [49]. Alternatively, the time dimension may also be discretized by DG, leading to a space-time DG discretization [59, 60]. The consistency of Runge–Kutta time discretizations will be discussed in detail in section 7.

We focus on the discontinuous Galerkin method because of its multiple advantages over the finite volume, finite differences, and continuous Galerkin approaches. DG is particularly amenable to adaptive mesh refinement and parallelization, due to the weak coupling between elements (which are connected only through the boundary fluxes \mathbf{F}^{DG}). The order of the numerical approximation can easily be varied inside each element, since the basis functions have only local support. This avoids the AMR complications introduced by global basis functions (necessary in the continuous Galerkin approach). Also, there is no increase in stencil size when higher order approximations are used, in contrast with the finite difference and finite volume methods, where p -refinement can only be implemented by adding extra points or cells to the computational stencil.

As remarked previously, consistency of discrete adjoints with the continuous problem is by no means guaranteed, even in the fixed mesh case. Hartmann [37] proposed a framework for investigating adjoint consistency of DG schemes for elliptic problems. Lack of this adjoint consistency property leads to non-smooth discrete adjoint solutions, as well as suboptimal rates of convergence for the primal problem [61, 62, 38]. This adjoint consistency concept can be extended to hyperbolic problems by considering a semi-discrete version of the primal problem. In this case, time derivatives can be implemented with a strong-stability preserving Runge–Kutta method [63]. Care must be taken such that source terms are also discretized in a dual consistent manner [42].

We show below that the interpolation and restriction operators for h - and p -refinement DG are exact transposes of each other (note that this transpose relationship holds if we have an embedding between the coarse/fine solution spaces, which may not be the case for curved domains).

4. Adjoint interpolation and restriction operators for h/p -adaptive DG

In this section we investigate the discrete adjoints of the projection and restriction operators for h - and p -adaptive DG. These are the grid transfer operators used in an adaptive adjoint code obtained through automatic differentiation. The analysis does

not consider the differentiation of the mesh refinement logic that decides a new grid size based on truncation error estimates. It is expected that differentiation of the spatial mesh refinement logic leads to inconsistent discrete adjoints, as it does in the case of temporal mesh refinement [53].

4.1. Hierarchical h -refinement

We first analyze the refinement and coarsening operators in the context of hierarchical mesh refinement, i.e., an embedding of nested meshes with finer and finer spacing. Implementation of this refinement strategy is facilitated by data structures such as quad- or octrees [64]. Note that the shape and dimension of the elements is arbitrary; we only assume the existence of smooth bijective mappings from a canonical (reference) element D to the active element. A quick analysis shows that both h - and p -refinement are done using orthogonal projections. Hence, we do not lose solution accuracy (beyond the aliasing introduced by coarsening itself) by using the code generated by AD for the adjoint intergrid solution transfers.

Consider the element D^k that is refined by the AMR mechanism into P smaller elements, i.e. $D^k = \bigcup_{p=1}^P D_p^k$. Let $\mathcal{M} : D \rightarrow D^k$ be a bijective mapping from the reference element D to the active element D^k . Similarly, \mathcal{M}_p is a one-to-one and onto map from D to $D_p^k \subset D^k$. We denote the orthonormal set of basis functions on D by $\{\Psi_j(x)\}_{j=0\dots J}$.

The interpolation operator from D^k to its set of P children elements $\bigcup_p D_p^k$ reads:

$$\mathcal{P}_{H \rightarrow h} := \begin{bmatrix} \mathcal{P}^1 \\ \vdots \\ \mathcal{P}^P \end{bmatrix}, \quad (23)$$

where

$$\mathcal{P}_{ij}^p := \int_{D_p^k} \Psi_j(\mathcal{M}^{-1}(x)) \Psi_i(\mathcal{M}_p^{-1}(x)) dx, \quad i, j = 0 \dots J. \quad (24)$$

Mesh coarsening collapses the P child elements into their parent element. The restriction operator that performs this operation is:

$$\mathcal{R}_{h \rightarrow H} := \begin{bmatrix} \mathcal{R}^1 & \dots & \mathcal{R}^P \end{bmatrix}. \quad (25)$$

with

$$\mathcal{R}_{ij}^p := \int_{D_p^k} \Psi_i(\mathcal{M}^{-1}(x)) \Psi_j(\mathcal{M}_p^{-1}(x)) dx, \quad i, j = 0 \dots J. \quad (26)$$

From (23–24) and (25–26):

$$\mathcal{R}_{h \rightarrow H} = \mathcal{P}_{H \rightarrow h}^T. \quad (27)$$

4.2. p -refinement

Adaptive order refinement (also called p -refinement) is useful for nonlinear problems, and implies the reduction or increase of the local order of the solution on a chosen subset of the grid elements. This is equivalent to adding or removing expansion coefficients (for the modal formulation), or interpolation points (in the nodal approach). Suppose for simplicity that the AMR mechanism flagged a set $\{D^{i_q}\}_{q=1\dots Q}$ of Q out of the K mesh elements for p -refinement. Assume the order of the solution is increased by \widehat{Q} on each of the Q mesh elements. Then, the p -refinement operator on element D^{i_q} has the following form:

$$\mathcal{A}_{J \rightarrow J+\widehat{Q}} := \begin{bmatrix} \mathbf{I}_{J \times J} \\ \mathbf{0}_{\widehat{Q} \times J} \end{bmatrix}. \quad (28)$$

The reverse operation on D^{i_q} can be written in operator form as

$$\begin{bmatrix} \widehat{u}_1^{(i_q)} \\ \vdots \\ \widehat{u}_N^{(i_q)} \end{bmatrix} = \mathcal{A}_{J \rightarrow J+\widehat{Q}}^T \begin{bmatrix} \widehat{u}_1^{(i_q)} \\ \vdots \\ \widehat{u}_N^{(i_q)} \\ \vdots \\ \widehat{u}_{N+\widehat{Q}}^{(i_q)} \end{bmatrix}. \quad (29)$$

Since the solution coefficients on all elements outside the refinement set remain unchanged, the transpose relationship for the global p -refinement operator follows from (28)–(29). This result and equation (27) prove the transpose relationship (10) holds with $\mathcal{C} = 1$:

$$\mathcal{I}_{n+1 \rightarrow n} = \mathcal{I}_{n \rightarrow n+1}^T. \quad (30)$$

This implies that the adjoint grid transfer operators generated via reverse mode automatic differentiation retain the accuracy of their forward model counterparts. Hence, h -adaptivity together with an adjoint consistent DG discretization [37] (see also the next section), lead to a stable and consistent discrete adjoint solution. Moreover, the adjoint DG code can be generated automatically from the forward problem discretization, without requiring any post-processing of the adjoint solution.

4.3. h -refinement with general meshes

We now extend our analysis of interpolation and coarsening to general triangulations. Consider two meshes that cover our domain Ω : $\Omega_A^h = \bigcup_k A^k$ and $\Omega_B^h = \bigcup_m B^m$. Consider also the elements generated by all intersections of elements of Ω_A^h and Ω_B^h : denote them by $C^{k,m} = A^k \cap B^m$. The corresponding mesh is $\Omega_C^h = \bigcup_{k,m} C^{k,m}$.

The solution on A^k is

$$\mathbf{u}_{A^k} = \sum_j a_{\{k,j\}} \phi_{\{k,j\}}(\mathbf{x}), \quad \phi_{\{k,j\}} = \phi_j \left(\mathcal{M}_{A,k}^{-1}(\mathbf{x}) \right),$$

while the solution on element B^m reads:

$$\mathbf{u}_{B^m} = \sum_i b_{\{m,i\}} \psi_{\{m,i\}}(\mathbf{x}) , \quad \psi_{\{m,i\}} = \psi_i \left(\mathcal{M}_{B^m}^{-1}(\mathbf{x}) \right) .$$

We project the solution \mathbf{u}_A defined on Ω_A^h on the basis ψ to obtain the solution \mathbf{u}_B on Ω_B^h . Note that

$$B^m = \bigcup_k C^{k,m} .$$

Consequently,

$$\begin{aligned} b_{\{m,i\}} &= \int_{B^m} \mathbf{u}_{B^m}(\mathbf{x}) \psi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} \\ &= \sum_k \int_{C^{k,m}} \mathbf{u}_{B^m}(\mathbf{x}) \psi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} \\ &= \sum_k \int_{C^{k,m}} \sum_j a_{\{k,j\}} \phi_{\{k,j\}}(\mathbf{x}) \psi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} \\ &= \sum_{\{k,j\}} \left(\int_{C^{k,m}} \phi_{\{k,j\}}(\mathbf{x}) \psi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} \right) a_{\{k,j\}} . \end{aligned}$$

The transfer matrix that maps the solution \mathbf{u}_A (defined by the a coefficients) to the solution \mathbf{u}_B (defined by the b coefficients) is:

$$\{b\} = \mathcal{T}^{A \rightarrow B} \cdot \{a\} , \quad \mathcal{T}_{\{m,i\},\{k,j\}}^{A \rightarrow B} = \int_{C^{k,m}} \phi_{\{k,j\}}(\mathbf{x}) \psi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} .$$

We do similar calculations for the solution transfer from B to A . In the above formulas we interchange the roles of a and b , and of ϕ and ψ to obtain:

$$a_{\{m,i\}} = \sum_{\{k,j\}} \left(\int_{C^{m,k}} \psi_{\{k,j\}}(\mathbf{x}) \phi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} \right) b_{\{k,j\}} .$$

The transfer matrix that maps the solution \mathbf{u}_B (defined by the b coefficients) to the solution \mathbf{u}_A (defined by the a coefficients) is:

$$\{a\} = \mathcal{T}^{B \rightarrow A} \cdot \{b\} , \quad \mathcal{T}_{\{m,i\},\{k,j\}}^{B \rightarrow A} = \int_{C^{m,k}} \psi_{\{k,j\}}(\mathbf{x}) \phi_{\{m,i\}}(\mathbf{x}) \, d\mathbf{x} .$$

Clearly the two intergrid operators are the transpose of one another, which means (30) holds in this more general case:

$$\mathcal{T}_{\{m,i\},\{k,j\}}^{B \rightarrow A} = \mathcal{T}_{\{k,j\},\{m,i\}}^{A \rightarrow B} .$$

5. Adjoint interpolation and restriction operators for the finite volume method

The finite volume method (FVM) [65] is built on the physical properties of the hyperbolic conservation law (20). Like discontinuous Galerkin, it can be used on structured and unstructured meshes, and has excellent geometric flexibility. The formulation of the FVM relies on conservation of “mass” (i.e, some solution average) principles inside a given control volume (or cell). Our discrete spatial mesh at t^n is now defined as the

union of K distinct control volumes: $\Omega_n^h = \bigcup_{k=1}^{K_n} C_n^k$. We define the approximate solution average inside a cell C_n^k at time t^n to be:

$$\mathbf{U}^{k,n} \approx \frac{1}{\text{Vol}(C_n^k)} \int_{C_n^k} \mathbf{u}(\mathbf{x}, t^n) d\mathbf{x}. \quad (31)$$

From (20) we obtain:

$$\frac{1}{\text{Vol}(C_n^k)} \left(\frac{\partial}{\partial t} \int_{C_n^k} \mathbf{u}(\mathbf{x}, t) d\mathbf{x} + \int_{\partial C_n^k} \mathbf{F} \cdot \vec{\mathbf{n}} ds - \int_{C_n^k} \mathbf{f}(\mathbf{x}, t) d\mathbf{x} \right) = 0.$$

Let σ be a boundary edge (or face) for the control cell C_n^k . Using a conservative and consistent approximation $\mathbf{F}_\sigma^{\text{FVM}}(\mathbf{x}^h, t^n)$ for the analytical flux $\mathbf{F}(\mathbf{x}, t^n)$ through σ , we arrive at the semi-discrete finite-volume formulation of (20):

$$\frac{d\mathbf{U}^{k,n}}{dt} + \frac{1}{\text{Vol}(C_n^k)} \left(\sum_{\sigma \in \partial C_n^k} \mathbf{F}_\sigma^{\text{FVM}}(\mathbf{x}^h, t^n) - \int_{C_n^k} \mathbf{f}(\mathbf{x}^h, t^n) d\mathbf{x} \right) = 0.$$

By virtue of the conservation principle, the net flow through the boundaries of the control volume must be zero in the absence of any forcing terms \mathbf{f} . Similar to the DG approach, reconstructing the solution at the control volume interfaces is done locally. Hence, generalizations to unstructured grids and higher dimensions are straightforward. The various choices of numerical fluxes lead to different finite volume methods [65]. However, the FVM in two or three space dimensions is not easily amenable to p -refinement, since higher order solution approximations rely on polynomial reconstructions over multiple cells. In d dimensions, one needs at least

$$\hat{J} = \frac{(J + d - 1)!}{(J - 1)! d!} \quad (32)$$

finite volume cells to build a J -th degree polynomial approximation to $\mathbf{u}(\mathbf{x}, t)$. Structured meshes make p -refinement easier by allowing straightforward stencil size adjustments. However, general unstructured grids are more complex to handle [66, 67, 68]. Another drawback of larger stencils is the order reduction in the primal and dual solutions around the boundaries of the domain [69], where the stencil cannot be extended to allow sufficient accuracy (except in the case of periodic boundary conditions).

We are concerned with the accuracy of intergrid operators for finite volume solvers on structured meshes. For simplicity, we first consider one-dimensional problems; the analysis will be extended to higher dimensions in section 5.2. This one-dimensional discussion fully illustrates the accuracy issues encountered with adjoint (transposed) intergrid operators. Furthermore, we omit the time dependent notation, for ease of notation. The k th finite volume cell is $C^k = [x^{k-1/2}, x^{k+1/2}]$, where $x^{k+1/2} = (x^{k+1} + x^k)/2$. Let $h^k = \text{Vol}(C^k) := x^{k+1} - x^k$. This leads to the finite volume scheme for equation (20):

$$h^k \frac{d\mathbf{U}^k}{dt} + \mathbf{F}_{k+1/2}^{\text{FVM}} - \mathbf{F}_{k-1/2}^{\text{FVM}} = h^k \mathbf{f}^k, \quad \forall k = 1 \dots K. \quad (33)$$

The polynomial reconstruction approach approximates the solution through the interface between two control volumes as a polynomial with unknown coefficients:

$$\mathbf{u}^h(x) := \sum_{j=0}^J a_j (x - x^k)^j . \quad (34)$$

Note that we dropped the explicit time dependency in the numerical solution, since we assume this holds at some particular time point t^n . The a_j coefficients are determined by requiring that

$$\int_{C^j} \mathbf{u}^h(x) = \text{Vol}(C^j) \mathbf{U}^j , \quad \forall j = 0, \dots, J .$$

The reconstruction may be either centered, or biased (upwind). We will show that the transpose relationship between the interpolation and restriction operators does not hold for operators with orders of accuracy higher than one. The adjoints of quadratic or higher order flux interpolations reduce to a simple first order conservative reconstruction. Hence, the adjoint solution cannot be expected to retain the order of accuracy of the forward discretization, due the order reduction in the intergrid solution transfer process. This is an additional drawback of the finite volume method, which negatively impacts the accuracy of black-box generation (via AD) of h -refinement operators in the discrete adjoint solver.

5.1. h -refinement via quadratic centered polynomial solution reconstruction

Assume a smooth exact solution $\mathbf{u}(x, t)$ to (20). Consider three adjacent finite volume cells of size h : C^L , C^C , and C^R , centered at x^{i-1} , x^i , and x^{i+1} , respectively. Their corresponding exact averages are \mathbf{U}_L , \mathbf{U}_C , and \mathbf{U}_R . The cell C^C is split into two cells C_L^C and C_R^C , each with volume $h/2$. The solution inside $C^L \cup C^C \cup C^R$ is approximated by a quadratic polynomial $\mathbf{u}^h(x)$, with the unknown coefficients determined from (34). We then obtain the averages on the two finer cells using equation (31):

$$\begin{bmatrix} \mathbf{U}_L^C \\ \mathbf{U}_R^C \end{bmatrix} = \begin{bmatrix} \frac{1}{8} & 1 & -\frac{1}{8} \\ -\frac{1}{8} & 1 & \frac{1}{8} \end{bmatrix} \begin{bmatrix} \mathbf{U}_L \\ \mathbf{U}_C \\ \mathbf{U}_R \end{bmatrix} . \quad (35)$$

We are interested in the order of accuracy of these approximations, i.e. we want to estimate the errors of the two new cell averages

$$\begin{aligned} E_L &:= \left| \mathbf{U}_L^C - \frac{2}{h} \int_{x^{i-1/2}}^{x^i} \mathbf{u}(x) dx \right| \\ E_R &:= \left| \mathbf{U}_R^C - \frac{2}{h} \int_{x^i}^{x^{i+1/2}} \mathbf{u}(x) dx \right| . \end{aligned} \quad (36)$$

Using (31) (now assumed to hold exactly for the cells of size h), we get that:

$$\max(|E_L|, |E_R|) = \frac{3}{64} h^3 \left| \frac{d^3 \mathbf{u}}{dx^3}(x^i) \right| + \mathcal{O}(h^5) ,$$

hence the approximation (36) is third order accurate for our uniform centered stencil. We now consider the transposed operator that coarsens C_L^C and C_R^C into a single

parent cell C_C . From (35) it is immediately apparent that the transposed coarsening operator is equivalent to first order (conservative) averaging. Moreover, using the adjoint (transpose) of the discrete interpolation operator in equation (35) has the undesired side-effect of perturbing the neighbor averages:

$$\begin{aligned} \mathbf{U}_C &= \mathbf{U}_C^L + \mathbf{U}_C^R \\ \widehat{\mathbf{U}}_L &= \mathbf{U}_L + 1/8 (\mathbf{U}_C^L - \mathbf{U}_C^R) := \mathbf{U}_L + \varepsilon_L \\ \widehat{\mathbf{U}}_R &= \mathbf{U}_R + 1/8 (-\mathbf{U}_C^L + \mathbf{U}_C^R) := \mathbf{U}_R + \varepsilon_R . \end{aligned} \quad (37)$$

Since by our assumptions \mathbf{U}_L and \mathbf{U}_R are exact averages, and $\mathbf{u}(\mathbf{x})$ is smooth over $C^L \cup C^C \cup C^R$, one can bound the perturbations ε_L and ε_R using Taylor approximations:

$$\max(|\varepsilon_L|, |\varepsilon_R|) = \frac{1}{16} h \left| \frac{d\mathbf{u}}{d\mathbf{x}}(x_i) \right| + \mathcal{O}(h^3) .$$

These numerical side-effects should be avoided in practice (preferably through post-processing of the discrete adjoint code). Nevertheless, this grid coarsening operation remains only first order. This can be also shown to hold for higher degree (centered or upwind) polynomial reconstructions, and for higher dimensional problems, as outlined in the next section.

5.2. General intergrid transfer operators in the finite volume method

We now consider a more general formulation of the intergrid transfer operators used in the finite volume method. In what follows \widehat{J} is defined by equation (32). Consider the polynomial reconstruction formula (34) over \widehat{J} cells $C^1, \dots, C^{\widehat{J}}$:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{a}^T \cdot \begin{bmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{\widehat{J}}(\mathbf{x}) \end{bmatrix} ,$$

where $\mathbf{a} = [a_1, \dots, a_{\widehat{J}}]^T$ are the polynomial coefficients, and $\phi_1, \dots, \phi_{\widehat{J}}$ are a set of basis functions for the space of multivariate polynomials of degree J under consideration. Then, the average on cell C_j is

$$\frac{1}{\text{Vol}(C^j)} \int_{C^j} \mathbf{u}^h(\mathbf{x}) \, d\mathbf{x} = \mathbf{w}_j^T \cdot \mathbf{a} = \mathbf{U}_j , \quad (38)$$

with \mathbf{w}_j denoting the integration weights:

$$(\mathbf{w}_j)_i = \frac{1}{\text{Vol}(C^j)} \int_{C^j} \phi_i(\mathbf{x}) \, d\mathbf{x} , \quad \forall i = 1, \dots, \widehat{J} . \quad (39)$$

Equation (38) leads to the matrix formulation over all \widehat{J} cells:

$$\mathbf{W} \mathbf{a} = \mathbf{U} . \quad (40)$$

In equation (40), the $\mathbf{W} \in \mathbb{R}^{\widehat{J} \times \widehat{J}}$ is the weight matrix, and \mathbf{U} denotes the column vector of \widehat{J} cell averages. Suppose now that cell C^i is refined into K non-overlapping sub-cells C_1^i, \dots, C_K^i . The averages inside the smaller K cells are given by:

$$\mathbf{U}_i^k = \mathbf{v}_k^T \mathbf{a} = \mathbf{v}_k^T \mathbf{W}^{-1} \mathbf{U} , \quad \forall i = 1, \dots, K ,$$

where the new integration weights \mathbf{v}_k are defined by

$$(\mathbf{v}_k)_j := \frac{1}{\text{Vol}(C_i^j)} \int_{C_i^j} \phi_j(\mathbf{x}) \, d\mathbf{x}, \quad \forall i = 1, \dots, \hat{J},$$

and satisfy the equation

$$\sum_{k=1}^K \mathbf{v}_k^T = \mathbf{w}_i^T = \mathbf{e}_i^T \mathbf{W}. \quad (41)$$

Here \mathbf{e}_i is the i -th unit basis vector of $\mathbb{R}^{\hat{J}}$. For small values of J , and smooth $\mathbf{u}(\mathbf{x}, t)$, the averages in the finer sub-cells can be shown to be accurate of order h^{J+1} . Larger-sized stencils introduce oscillations in the approximating polynomials, hence some WENO-type stabilization is required [70]. The analysis of the transpose of the stabilization algorithm is beyond the scope of this paper. We can write the refinement operation in matrix form as:

$$\begin{bmatrix} \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{i-1} \\ \mathbf{U}_i^1 \\ \vdots \\ \mathbf{U}_i^K \\ \mathbf{U}_{i+1} \\ \vdots \\ \mathbf{U}_{\hat{J}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \hline \mathcal{P}_K & & \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \mathbf{U}, \quad (42)$$

with the prolongation sub-matrix

$$\mathcal{P}_K = \begin{bmatrix} \mathbf{v}_1^T \mathbf{W}^{-1} \\ \vdots \\ \mathbf{v}_K^T \mathbf{W}^{-1} \end{bmatrix} \in \mathbb{R}^{K \times \hat{J}}. \quad (43)$$

Use of the adjoint of (42) as a coarsening operator yields the following average for cell C_i :

$$\tilde{\mathbf{U}}_i = \sum_{k=1}^K (\mathbf{W}^{-T} \mathbf{v}_k)_i \mathbf{U}_i^k. \quad (44)$$

From (41), we get a first order conservative reconstruction of the solution average inside cell C_i . However, there is one undesired side-effect of this operator. The average solution values inside all of the other $\hat{J} - 1$ cells in the interpolation stencil are polluted by a first order perturbation stemming from the transposed restriction operator (44):

$$\tilde{\mathbf{U}}_j = \mathbf{U}_j + \sum_{k=1}^K (\mathbf{W}^{-T} \mathbf{v}_k)_j \mathbf{U}_i^k, \quad \forall j \neq i.$$

This perturbation is $\mathcal{O}(h)$ on general uniform and non-uniform grids. To establish this estimate, note that, using (41), we have

$$\sum_{k=1}^K (\mathbf{W}^{-T} \mathbf{v}_k)_j = 0, \quad \forall j \neq i.$$

In the general case this first order error term does not vanish.

6. Space-time duality relations in function spaces

As mentioned previously, the concept of adjoint consistency, together with its implications in optimization, have been investigated for steady-state problems by Lu [38], Harriman, Gavaghan and Süli [62], Hartmann and Houston [71, 72], and Oliver and Darmofal [42]. Hartmann [37] proposed a general framework for establishing adjoint consistency for DG discretizations of stationary PDE models. We leverage previous results on dual consistency for temporal [40], and spatial discretizations [71] to give a unified framework for the analysis of adjoint consistency of space-time DG discretizations. This section discusses space-time duality relations for continuous model formulations. A general strategy to construct the adjoint system is given, applicable whenever the cost functional and the associated model differential operators satisfy a set of compatibility conditions. The next section will discuss dual consistency of the time quadratures for Runge–Kutta DG discretizations (assumed to be dual consistent in space).

Consider again equation (1). For simplicity of exposition, we rewrite (1) in the form:

$$\begin{aligned} \mathbf{u}_t &= N[\mathbf{u}] + \mathbf{f}, \quad \mathbf{x} \in \Omega, \quad t \in [0, T] \\ B[\mathbf{u}] &= \mathbf{g}, \quad \mathbf{x} \in \Gamma, \quad t \in [0, T] \\ \mathbf{u}(t=0, \mathbf{x}) &= \mathbf{u}^0, \quad \mathbf{x} \in \Omega. \end{aligned} \tag{45}$$

The PDE system admits solutions $\mathbf{u} : [0, T] \rightarrow \mathcal{U}$, such that $\mathbf{u} \in \mathcal{L}^2([0, T]; \mathcal{U})$, and $\mathbf{u}_t \in \mathcal{L}^2([0, T]; \mathcal{U})$, where \mathcal{U} is an appropriate function space. Here N and B are Fréchet differentiable, nonlinear differential operators, containing spatial and boundary derivative terms. We denote the Fréchet derivatives by:

$$\begin{aligned} L\mathbf{w} &= N'[\mathbf{u}]\mathbf{w} \\ B'\mathbf{w} &= B'[\mathbf{u}]\mathbf{w}. \end{aligned}$$

Consider a nonlinear cost functional of the form

$$\begin{aligned} \mathcal{J}(\mathbf{u}) &= \int_0^T \int_{\Omega} J_{\Omega}[C_{\Omega}\mathbf{u}] \, d\mathbf{x} \, dt + \int_0^T \int_{\Gamma} J_{\Gamma}[C_{\Gamma}\mathbf{u}] \, ds \, dt \\ &\quad + \int_{\Omega} K_{\Omega}[E_{\Omega}\mathbf{u}]_{t=T} \, d\mathbf{x}. \end{aligned} \tag{46}$$

The differential operators C_{Ω} and E_{Ω} act on the domain Ω , while C_{Γ} is a boundary operator (all are assumed to be Fréchet differentiable). Their Fréchet derivatives are denoted by C'_{Ω} , E'_{Ω} , and C'_{Γ} , respectively. Also, let

$$\begin{aligned} j_{\Omega} &= (J'_{\Omega}[C_{\Omega}\mathbf{u}])^T \\ j_{\Gamma} &= (J'_{\Gamma}[C_{\Gamma}\mathbf{u}])^T \\ k_{\Omega} &= (K'_{\Omega}[E_{\Omega}\mathbf{u}])^T. \end{aligned}$$

6.1. The tangent linear PDE

Small variations $\delta \mathbf{u}$ in the solution $\mathbf{u}(\mathbf{x}, t)$ of (45) satisfy (up to first order) the tangent linear model. These equations can be obtained from (45) by linearization in the direction $(\delta \mathbf{u}, \delta \mathbf{f}, \delta \mathbf{g})$:

$$\begin{aligned} \delta \mathbf{u}_t &= N'[\mathbf{u}] \delta \mathbf{u} + \delta \mathbf{f}, \quad \mathbf{x} \in \Omega, \quad t \in [0, T] \\ B'[\mathbf{u}] \delta \mathbf{u} &= \delta \mathbf{g}, \quad \mathbf{x} \in \Gamma, \quad t \in [0, T] \\ \delta \mathbf{u}(t=0, \mathbf{x}) &= \delta \mathbf{u}^0, \quad \mathbf{x} \in \Omega. \end{aligned} \quad (47)$$

We denote

$$\begin{aligned} \langle \mathbf{u}, \mathbf{v} \rangle_{[0, T] \times \Gamma} &:= \int_0^T \int_{\Gamma} \mathbf{u}^T \mathbf{v} \, ds \, dt \\ \langle \mathbf{u}, \mathbf{v} \rangle_{[0, T] \times \Omega} &:= \int_0^T \int_{\Omega} \mathbf{u}^T \mathbf{v} \, d\mathbf{x} \, dt. \end{aligned} \quad (48)$$

The space-time weak form of (47) is written in terms of space-time inner products as:

$$\begin{aligned} \langle \mathbf{w}_t, \mathbf{v} \rangle_{[0, T] \times \Omega} &= \langle L[\mathbf{u}] \mathbf{w}, \mathbf{v} \rangle_{[0, T] \times \Omega} + \langle \delta \mathbf{f}, \mathbf{v} \rangle_{[0, T] \times \Omega} \\ \langle B'[\mathbf{u}] \mathbf{w}, \mathbf{v} \rangle_{[0, T] \times \Gamma} &= \langle \delta \mathbf{g}, \mathbf{v} \rangle_{[0, T] \times \Gamma} \\ \langle \mathbf{w}, \mathbf{v} \rangle_{\Omega}|_{t=0} &= \langle \delta \mathbf{u}^0, \mathbf{v} \rangle_{\Omega}, \quad \forall \mathbf{v} \in \mathcal{L}^2([0, T]; \mathcal{U}). \end{aligned} \quad (49)$$

The space of all possible solutions of (49) is:

$$\mathcal{U}^{\text{tlm}} = \{ \mathbf{w} \in \mathcal{U} \mid \exists \delta \mathbf{f}, \delta \mathbf{g}, \delta \mathbf{u}_0 \text{ s.t. } \mathbf{w} \text{ is a solution of (49)} \}. \quad (50)$$

Clearly \mathcal{U}^{tlm} is a vector subspace of \mathcal{U} . The variation of the cost functional (46) is

$$\delta \mathcal{J} = \mathcal{J}'[\mathbf{u}] \delta \mathbf{u}, \quad (51)$$

where

$$\begin{aligned} \mathcal{J}'[\mathbf{u}] \mathbf{w} &= \int_0^T \int_{\Omega} J'_{\Omega}[C_{\Omega} \mathbf{u}] C'_{\Omega} \mathbf{w} \, d\mathbf{x} \, dt + \int_0^T \int_{\Gamma} J'_{\Gamma}[C_{\Gamma} \mathbf{u}] C'_{\Gamma} \mathbf{w} \, ds \, dt \\ &\quad + \left(\int_{\Omega} K'_{\Omega}[E_{\Omega} \mathbf{u}] E'_{\Omega} \mathbf{w} \, d\mathbf{x} \right) \Big|_{t=T} \\ &:= \langle C'_{\Omega} \mathbf{w}, j_{\Omega} \rangle_{[0, T] \times \Omega} + \langle C'_{\Gamma} \mathbf{w}, j_{\Gamma} \rangle_{[0, T] \times \Gamma} + \langle E'_{\Omega} \mathbf{w}, k_{\Omega} \rangle_{\Omega}|_{t=T}. \end{aligned}$$

To compute the variation (51) due to $\delta \mathbf{u}^0$, $\delta \mathbf{f}$, and $\delta \mathbf{g}$, one runs the TLM (49) to obtain $\delta \mathbf{u}$, and uses it in (51) to compute $\delta \mathcal{J}$. A new TLM solution is needed for each set of perturbations $\delta \mathbf{u}^0$, $\delta \mathbf{f}$, and $\delta \mathbf{g}$.

6.2. The adjoint PDE

We wish to express the variation (51) as

$$\delta \mathcal{J} = \langle C_{\Omega}^{\text{adj}} \lambda, \delta \mathbf{f} \rangle_{[0, T] \times \Omega} + \langle C_{\Gamma}^{\text{adj}} \lambda, \delta \mathbf{g} \rangle_{[0, T] \times \Gamma} + \langle E_{\Omega}^{\text{adj}} \lambda|_{t=0}, \delta \mathbf{u}_0 \rangle_{\Omega}, \quad (52)$$

for any perturbations $\delta \mathbf{u}^0$, $\delta \mathbf{f}$, and $\delta \mathbf{g}$. The adjoint variables λ are obtained by solving the dual problem

$$\begin{aligned} -\lambda_t &= L^* \lambda + \mathbf{f}^{\text{adj}}, \quad \mathbf{x} \in \Omega, \quad t \in [0, T] \\ B^{\text{adj}} \lambda &= \mathbf{g}^{\text{adj}}, \quad \mathbf{x} \in \Gamma, \quad t \in [0, T] \\ \lambda(t = T, \mathbf{x}) &= E_{\Omega}^{\text{adj}} k_{\Omega}, \quad \mathbf{x} \in \Omega. \end{aligned} \quad (53)$$

The operators B^{adj} and C_{Γ}^{adj} need to be chosen such that (52) and (51) are equivalent. Note that duality implies that the following relations hold:

- The domain weight j_{Ω} in the forward linearized cost function (51) determines the adjoint domain forcing \mathbf{f}^{adj} . The domain forcing $\delta \mathbf{f}$ in the forward linearized problem determines the domain weight in the adjoint expression of the cost function (52).
- The boundary weight j_{Γ} in the forward linearized cost function (51) determines the adjoint boundary forcing \mathbf{g}^{adj} . The boundary forcing $\delta \mathbf{g}$ in the forward linearized problem determines the boundary weight in the adjoint expression of the cost function (52).
- The domain forcing at the final time k_{Ω} in the forward linearized cost function (51) determines the final value of the adjoint variable $\lambda(t = T, \mathbf{x})$.

The time-space weak form of (54) is

$$\begin{aligned} -\langle \mathbf{w}, \lambda_t \rangle_{[0, T] \times \Omega} &= \langle \mathbf{w}, L^* \lambda \rangle_{[0, T] \times \Omega} + \langle \mathbf{w}, \mathbf{f}^{\text{adj}} \rangle_{[0, T] \times \Omega} \\ \langle \mathbf{w}, B^{\text{adj}} \lambda \rangle_{[0, T] \times \Gamma} &= \langle \mathbf{w}, \mathbf{g}^{\text{adj}} \rangle_{[0, T] \times \Gamma} \\ \langle \mathbf{w}, \lambda|_{t=T} \rangle_{\Omega} &= \langle \mathbf{w}, \lambda^{\text{F}} \rangle_{\Omega}. \end{aligned} \quad (54)$$

6.3. Compatibility conditions

Consider the integration by parts formulas

$$\langle L \mathbf{w}, \mathbf{v} \rangle_{\Omega} = \langle \mathbf{w}, L^* \mathbf{v} \rangle_{\Omega} + \sum_i \langle F_i^L \mathbf{w}, G_i^L \mathbf{v} \rangle_{\Gamma}, \quad \forall \mathbf{w}, \mathbf{v} \in \mathcal{U} \quad (55a)$$

$$\langle C'_{\Omega} \mathbf{w}, \mathbf{v} \rangle_{\Omega} = \langle \mathbf{w}, C'^*_{\Omega} \mathbf{v} \rangle_{\Omega} + \sum_i \langle F_i^C \mathbf{w}, G_i^C \mathbf{v} \rangle_{\Gamma}, \quad \forall \mathbf{w}, \mathbf{v} \in \mathcal{U} \quad (55b)$$

$$\langle E'_{\Omega} \mathbf{w}, \mathbf{v} \rangle_{\Omega} = \langle \mathbf{w}, E'^*_{\Omega} \mathbf{v} \rangle_{\Omega} + \sum_i \langle F_i^E \mathbf{w}, G_i^E \mathbf{v} \rangle_{\Gamma}, \quad \forall \mathbf{w}, \mathbf{v} \in \mathcal{U} \quad (55c)$$

where $F_i^{L,C,E}$, $G_i^{L,C,E}$ are boundary linear differential operators that come from the integration by parts of the linear operators L , C'_{Ω} , and E'_{Ω} , respectively. We impose a first compatibility condition which ensures that the boundary terms coming from the integration by parts of C'_{Ω} vanish for all w that satisfy the boundary condition (50) of the tangent linear model (49):

$$\begin{aligned} \text{Compatibility condition 1 : } \quad & \sum_i \langle F_i^C \mathbf{w}, G_i^C \mathbf{v} \rangle_{\Gamma} = 0, \\ & \forall \mathbf{v} \in \mathcal{U}, \quad \forall \mathbf{w} \in \mathcal{U}^{\text{tlm}}. \end{aligned} \quad (56)$$

The second compatibility condition ensures that the boundary terms coming from the integration by parts of E'_Ω vanish for all perturbations consistent with (50):

$$\begin{aligned} \text{Compatibility condition 2 : } \quad & \sum_i \langle F_i^E w, G_i^E v \rangle_\Gamma = 0, \\ & \forall \mathbf{v} \in \mathcal{U}, \forall \mathbf{w} \in \mathcal{U}^{\text{tlm}}. \end{aligned} \quad (57)$$

The compatibility conditions (56) and (57) simplify the integration by parts formulas (55) to

$$\langle L \mathbf{w}, \mathbf{v} \rangle_\Omega = \langle \mathbf{w}, L^* \mathbf{v} \rangle_\Omega + \sum_i \langle F_i^L \mathbf{w}, G_i^L \mathbf{v} \rangle_\Gamma, \quad \forall \mathbf{w}, \mathbf{v} \in \mathcal{U} \quad (58a)$$

$$\langle C'_\Omega \mathbf{w}, \mathbf{v} \rangle_\Omega = \langle \mathbf{w}, C'^*_\Omega \mathbf{v} \rangle_\Omega, \quad \forall \mathbf{w} \in \mathcal{U}^{\text{tlm}}, \forall \mathbf{v} \in \mathcal{U} \quad (58b)$$

$$\langle E'_\Omega \mathbf{w}, \mathbf{v} \rangle_\Omega = \langle \mathbf{w}, E'^*_\Omega \mathbf{v} \rangle_\Omega, \quad \forall \mathbf{w} \in \mathcal{U}^{\text{tlm}}, \forall \mathbf{v} \in \mathcal{U}. \quad (58c)$$

After integration by parts the TLM (49) becomes:

$$\begin{aligned} -\langle \mathbf{w}, \lambda_t \rangle_{[0,T] \times \Omega} + \langle \mathbf{w}, \lambda \rangle_\Omega \Big|_0^T &= \langle \mathbf{w}, L^* \lambda \rangle_{[0,T] \times \Omega} \\ &+ \sum_i \langle F_i^L \mathbf{w}, G_i^L \lambda \rangle_{[0,T] \times \Gamma} \\ &+ \langle \delta \mathbf{f}, \lambda \rangle_{[0,T] \times \Omega} \\ \langle B' \mathbf{w}, \lambda \rangle_{[0,T] \times \Gamma} &= \langle \delta \mathbf{g}, \lambda \rangle_{[0,T] \times \Gamma}, \\ \langle \mathbf{w}|_{t=0}, \lambda \rangle_\Omega &= \langle \delta \mathbf{u}^0, \lambda \rangle_\Omega. \end{aligned} \quad (59)$$

Equations (59) and (55) lead to

$$\begin{aligned} \langle \mathbf{w}, \mathbf{f}^{\text{adj}} \rangle_{[0,T] \times \Omega} + \langle \mathbf{w}, \lambda \rangle_\Omega \Big|_{t=T} &= \langle \mathbf{w}, \lambda \rangle_\Omega \Big|_{t=0} + \langle \delta \mathbf{f}, \lambda \rangle_{[0,T] \times \Omega} \\ &+ \sum_i \langle F_i^L \mathbf{w}, G_i^L \lambda \rangle_{[0,T] \times \Gamma}. \end{aligned} \quad (60)$$

The variation of the cost functional (51) can be rewritten as:

$$\begin{aligned} \mathcal{J}'[\mathbf{u}] \mathbf{w} &= \langle C'_\Omega \mathbf{w}, j_\Omega \rangle_{[0,T] \times \Omega} + \langle C'_\Gamma \mathbf{w}, j_\Gamma \rangle_{[0,T] \times \Gamma} + \langle E'_\Omega \mathbf{w}, k_\Omega \rangle_\Omega \Big|_{t=T} \\ &= \langle \mathbf{w}, C'^*_\Omega j_\Omega \rangle_{[0,T] \times \Omega} + \langle C'_\Gamma \mathbf{w}, j_\Gamma \rangle_{[0,T] \times \Gamma} + \langle \mathbf{w}, E'^*_\Omega k_\Omega \rangle_\Omega \Big|_{t=T}. \end{aligned} \quad (61)$$

We make the following identifications:

$$\begin{aligned} \mathbf{f}^{\text{adj}} &= C'^*_\Omega j_\Omega \\ \mathbf{g}^{\text{adj}} &= j_\Gamma \\ \lambda_F &= E'^*_\Omega k_\Omega. \end{aligned}$$

Then, the adjoint problem (54) reads

$$\begin{aligned} -\lambda_t &= L^* \lambda + C'^*_\Omega j_\Omega, \quad \mathbf{x} \in \Omega, \quad t \in [0, T] \\ B^{\text{adj}} \lambda &= j_\Gamma, \quad \mathbf{x} \in \Gamma, \quad t \in [0, T] \\ \lambda(t = T, \mathbf{x}) &= E'^*_\Omega k_\Omega, \quad \mathbf{x} \in \Omega, \end{aligned} \quad (62)$$

and equation (61) becomes:

$$\begin{aligned}
 \mathcal{J}'[\mathbf{u}] \mathbf{w} &= - \langle \mathbf{w}, \lambda \rangle_{\Omega} |_{t=T} + \langle \mathbf{w}, \lambda \rangle_{\Omega} |_{t=0} + \langle \mathbf{w}_t - L \mathbf{w}, \lambda \rangle_{[0,T] \times \Omega} \\
 &\quad - \sum_i \langle F_i^L \mathbf{w}, G_i^L \lambda \rangle_{[0,T] \times \Gamma} + \sum_i \langle F_i^C \mathbf{w}, G_i^C j_{\Omega} \rangle_{[0,T] \times \Gamma} \\
 &\quad + \langle C'_{\Gamma} \mathbf{w}, B^{\text{adj}} \lambda \rangle_{[0,T] \times \Gamma} + \langle \mathbf{w}, \lambda \rangle_{\Omega} |_{t=T} \\
 &= \langle \mathbf{w}, \lambda \rangle_{\Omega} |_{t=0} + \langle \delta \mathbf{f}, \lambda \rangle_{[0,T] \times \Omega} \\
 &\quad + \langle C'_{\Gamma} \mathbf{w}, B^{\text{adj}} \lambda \rangle_{[0,T] \times \Gamma} + \sum_i \langle F_i^L \mathbf{w}, G_i^L \lambda \rangle_{[0,T] \times \Gamma} \\
 &\quad + \sum_i \langle F_i^C \mathbf{w}, G_i^C j_{\Omega} \rangle_{[0,T] \times \Gamma} .
 \end{aligned}$$

If the adjoint boundary condition is defined by the relation

$$\begin{aligned}
 \langle B^{\text{adj}} \lambda, C'_{\Gamma} \mathbf{w} \rangle_{[0,T] \times \Gamma} &= \langle C_{\Gamma}^{\text{adj}} \lambda, B' \mathbf{w} \rangle_{[0,T] \times \Gamma} \\
 &\quad - \sum_i \langle F_i^L \mathbf{w}, G_i^L \lambda \rangle_{[0,T] \times \Gamma} ,
 \end{aligned} \tag{63}$$

then

$$\mathcal{J}'[\mathbf{u}] \mathbf{w} = \langle \delta \mathbf{u}^0, \lambda \rangle_{\Omega} |_{t=0} + \langle \delta \mathbf{f}, \lambda \rangle_{[0,T] \times \Omega} + \langle C_{\Gamma}^{\text{adj}} \lambda, \delta \mathbf{g} \rangle_{[0,T] \times \Gamma} . \tag{64}$$

Equation (63) is ensured by the third compatibility condition. There exist well defined boundary operators B^{adj} and C_{Γ}^{adj} such that the following holds:

Compatibility condition 3 :

$$\begin{aligned}
 (L \mathbf{w}, \mathbf{v})_{\Omega} - (B' \mathbf{w}, C_{\Gamma}^{\text{adj}} \mathbf{v})_{\Gamma} &= (\mathbf{w}, L^* \mathbf{v})_{\Omega} - (C'_{\Gamma} \mathbf{w}, B^{\text{adj}} \mathbf{v})_{\Gamma} \\
 \forall \mathbf{w} \in \mathcal{U}^{\text{tlm}}, \mathbf{v} \in \mathcal{U} .
 \end{aligned} \tag{65}$$

Here we have used (58) to relate (63) and (65).

We say that the cost function and the PDE are compatible if the three compatibility conditions (56), (57), and (65) hold. The third compatibility condition (65) is discussed in [20, 37]. The authors assume $C = I$ (the identity operator), and $E = 0$, therefore (56), and (57) trivially hold. Equation (65) is the only compatibility condition needed in this simpler setting.

6.4. An example: the linear advection-diffusion equation

As an example, we will consider the linear advection-diffusion problem:

$$\begin{aligned}
 \mathbf{u}_t &= - \nabla \cdot (\bar{a} \mathbf{u}) + \Delta \mathbf{u} + \mathbf{f}, \mathbf{x} \in \Omega, t \in [0, T] \\
 \mathbf{u} &= \mathbf{g}_D, \mathbf{x} \in \Gamma_D = \Gamma_- \\
 \bar{\mathbf{n}} \cdot \nabla \mathbf{u} &= \mathbf{g}_N, \mathbf{x} \in \Gamma_N = \Gamma \setminus \Gamma_D \\
 \mathbf{u}(t=0, \mathbf{x}) &= \mathbf{u}^0 .
 \end{aligned} \tag{66}$$

Here $\Gamma_- = \{\mathbf{x} \in \Gamma \mid \bar{a}(\mathbf{x}) \cdot \bar{\mathbf{n}}(\mathbf{x}) < 0\}$ is the advective inflow boundary. The nonlinear cost functional quantifies the mismatch between the model trajectory $\mathbf{u}(t, \mathbf{x})$, and a

given reference state \mathbf{u}^{ref} . We penalize high variations in the boundary derivatives (to avoid boundary layers in our numerical solution). Thus, \mathcal{J} is defined as:

$$\mathcal{J}(\mathbf{u}) = \frac{1}{2} \int_0^T \int_{\Omega} \|\mathbf{u} - \mathbf{u}^{\text{ref}}\|_2^2 \, d\mathbf{x} \, dt + \frac{1}{2} \int_0^T \int_{\Gamma_D} (\nabla \mathbf{u} \cdot \vec{\mathbf{n}})^2 \, ds \, dt .$$

We immediately identify the operators

$$\begin{aligned} J_{\Omega}[C_{\Omega} \mathbf{u}] &= \frac{1}{2} \|\mathbf{u} - \mathbf{u}^{\text{ref}}\|_2^2 ; & j_{\Omega} &= \mathbf{u} - \mathbf{u}^{\text{ref}} ; \\ C_{\Gamma_D}(\mathbf{u}) &= \nabla \mathbf{u} \cdot \vec{\mathbf{n}} ; & C'_{\Gamma_D}(\mathbf{w}) &= \nabla \mathbf{w} \cdot \vec{\mathbf{n}} ; \\ J_{\Gamma_D}[C_{\Gamma_D} \mathbf{u}] &= \frac{1}{2} (\nabla \mathbf{u} \cdot \vec{\mathbf{n}})^2 ; & j_{\Gamma_D} &= \nabla \mathbf{u} \cdot \vec{\mathbf{n}} . \end{aligned}$$

6.4.1. The tangent linear PDE The TLM of (66) reads:

$$\begin{aligned} \mathbf{w}_t &= -\nabla \cdot (\vec{a} \mathbf{w}) + \Delta \mathbf{w} + \delta \mathbf{f}, & \mathbf{x} \in \Omega, & t \in [0, T] \\ B'_D \mathbf{w} := \mathbf{w} &= \delta \mathbf{g}_D, & \mathbf{x} \in \Gamma_D \\ B'_N \mathbf{w} := \vec{\mathbf{n}} \cdot \nabla \mathbf{w} &= \delta \mathbf{g}_N, & \mathbf{x} \in \Gamma_N \\ \mathbf{w}(t=0, \mathbf{x}) &= \delta \mathbf{u}^0, \end{aligned}$$

and the Fréchet derivative of the cost functional \mathcal{J} in the direction \mathbf{w} can be written as:

$$\mathcal{J}'[\mathbf{u}] \mathbf{w} = \left\langle \mathbf{u} - \mathbf{u}^{\text{ref}}, \mathbf{w} \right\rangle_{[0, T] \times \Omega} + \left\langle \frac{\partial \mathbf{u}}{\partial \vec{\mathbf{n}}}, \frac{\partial \mathbf{w}}{\partial \vec{\mathbf{n}}} \right\rangle_{[0, T] \times \Gamma_D} .$$

6.4.2. The adjoint PDE The integration by parts formula becomes

$$\begin{aligned} \langle -\nabla \cdot (\vec{a} \mathbf{w}) + \Delta \mathbf{w}, \mathbf{v} \rangle_{[0, T] \times \Omega} &= \langle \mathbf{w}, \vec{a} \cdot \nabla \mathbf{v} + \Delta \mathbf{v} \rangle_{[0, T] \times \Omega} \\ &+ \left\langle \mathbf{v}, -\mathbf{w} \vec{a} \cdot \vec{\mathbf{n}} + \frac{\partial \mathbf{w}}{\partial \vec{\mathbf{n}}} \right\rangle_{[0, T] \times \Gamma} \\ &+ \left\langle -\frac{\partial \mathbf{v}}{\partial \vec{\mathbf{n}}}, \mathbf{w} \right\rangle_{[0, T] \times \Gamma} . \end{aligned}$$

Then, we can easily identify the volume and boundary operators:

$$\begin{aligned} L \mathbf{w} &= -\nabla \cdot (\vec{a} \mathbf{w}) + \Delta \mathbf{w} \\ L^* \mathbf{v} &= \vec{a} \cdot \nabla \mathbf{v} + \Delta \mathbf{v} \\ F_1 \mathbf{w} &= -\mathbf{w} \vec{a} \cdot \vec{\mathbf{n}} + \frac{\partial \mathbf{w}}{\partial \vec{\mathbf{n}}}, & G_1 \mathbf{v} &= \mathbf{v} \\ F_2 \mathbf{w} &= \mathbf{w}, & G_2 \mathbf{v} &= -\frac{\partial \mathbf{v}}{\partial \vec{\mathbf{n}}} . \end{aligned}$$

The compatibility conditions (56) and (57) are trivially satisfied. The operators B^{adj} and C_{Γ}^{adj} are determined by the third compatibility condition, which for our example reduces to:

$$\begin{aligned} &\langle \mathbf{v}, -\mathbf{w} \vec{a} \cdot \vec{\mathbf{n}} + \vec{\mathbf{n}} \cdot \nabla \mathbf{w} \rangle_{[0, T] \times \Gamma} + \langle -\vec{\mathbf{n}} \cdot \nabla \mathbf{v}, \mathbf{w} \rangle_{[0, T] \times \Gamma} \\ &= \left\langle \mathbf{w}, C_{\Gamma_D}^{\text{adj}} \mathbf{v} \right\rangle_{[0, T] \times \Gamma_D} - \left\langle \vec{\mathbf{n}} \cdot \nabla \mathbf{w}, B_{\Gamma_D}^{\text{adj}} \mathbf{v} \right\rangle_{[0, T] \times \Gamma_D} \\ &+ \left\langle \vec{\mathbf{n}} \cdot \nabla \mathbf{w}, C_{\Gamma_N}^{\text{adj}} \mathbf{v} \right\rangle_{[0, T] \times \Gamma_N} - \left\langle \mathbf{w}, B_{\Gamma_N}^{\text{adj}} \mathbf{v} \right\rangle_{[0, T] \times \Gamma_N} . \end{aligned}$$

Using the linearity of the inner product integrals, we can establish that:

$$\begin{aligned} B_{\Gamma_D}^{\text{adj}} \mathbf{v} &:= -\mathbf{v}, \quad \mathbf{x} \in \Gamma_D \\ B_{\Gamma_N}^{\text{adj}} \mathbf{v} &:= \vec{a} \cdot \vec{\mathbf{n}} \mathbf{v} + \frac{\partial \mathbf{v}}{\partial \vec{\mathbf{n}}}, \quad \mathbf{x} \in \Gamma_N \\ C_{\Gamma_D}^{\text{adj}} \mathbf{v} &:= -\vec{a} \cdot \vec{\mathbf{n}} \mathbf{v} - \frac{\partial \mathbf{v}}{\partial \vec{\mathbf{n}}}, \quad \mathbf{x} \in \Gamma_D \\ C_{\Gamma_N}^{\text{adj}} \mathbf{v} &:= \mathbf{v}, \quad \mathbf{x} \in \Gamma_N, \end{aligned}$$

and write the adjoint final value problem for (66):

$$\begin{aligned} -\lambda_t &= \vec{a} \cdot \nabla \lambda + \Delta \lambda + \mathbf{u} - \mathbf{u}^{\text{ref}}, \quad \mathbf{x} \in \Omega, \quad t \in (T, 0] \\ \lambda &= -\nabla \mathbf{u} \cdot \vec{\mathbf{n}}, \quad \mathbf{x} \in \Gamma_D \\ \vec{\mathbf{n}} \cdot \nabla \lambda + \vec{a} \cdot \vec{\mathbf{n}} \lambda &= 0, \quad \mathbf{x} \in \Gamma_N \\ \lambda(t = T, \mathbf{x}) &= 0. \end{aligned}$$

We will revisit this example below, in the context of fully discrete models.

7. Duality relations and space-time adjoints for discrete models

If the time dimension is discretized by DG (see, e.g., [59, 60]), then we have a space-time DG discretization. The consistency analysis follows closely the one presented in [37]. The only difference is that the integrals are taken in space-time.

We now consider a time discretization by Runge Kutta methods. A semi-discretization in space of the continuous primal problem (45) leads to the following semi-discrete model [37]:

Find $\mathbf{u}^h \in \mathcal{L}^2([0, T]; \mathcal{U}_h)$ such that $(u^h)_t \in \mathcal{L}^2([0, T]; \mathcal{U}_h)$ and

$$\begin{aligned} \left\langle \frac{\partial \mathbf{u}^h}{\partial t}, \mathbf{v}^h(t) \right\rangle_{\Omega} &= \mathcal{N}(t; \mathbf{u}^h, \mathbf{v}^h) + \langle \mathbf{f}, \mathbf{v}^h \rangle_{\Omega} + \mathcal{B}(\mathbf{g}, \mathbf{v}^h) \\ &\forall \mathbf{v}^h \in \mathcal{L}^2([0, T]; \mathcal{U}_h), \quad \text{a.a. } t \in [0, T]. \end{aligned} \quad (67)$$

Here the semi-linear form \mathcal{N} is nonlinear in \mathbf{u}^h , and linear in \mathbf{v}^h . $\mathcal{B}(\cdot, \cdot)$ is a bilinear form defined on the boundary Γ , which depends on the prescribed boundary data \mathbf{g}^h . Let $\mathcal{N}'[\mathbf{u}^h] := \partial \mathcal{N} / \partial \mathbf{u}^h$ be the Fréchet derivative of \mathcal{N} with respect to \mathbf{u}^h . The TLM of (67) reads

$$\begin{aligned} \left\langle \frac{\partial \mathbf{w}^h}{\partial t}, \mathbf{v}^h(t) \right\rangle_{\Omega} &= \mathcal{N}'[\mathbf{u}^h](t; \mathbf{w}^h(t), \mathbf{v}^h) + \langle \delta \mathbf{f}(t), \mathbf{v}^h \rangle + \mathcal{B}(\delta \mathbf{g}, \mathbf{v}^h) \\ &\forall \mathbf{v}^h \in \mathcal{L}^2([0, T]; \mathcal{U}_h^{\text{tlm}}), \quad \text{a.a. } t \in [0, T], \end{aligned} \quad (68)$$

where the TLM solution $\mathbf{w}^h \in \mathcal{L}^2([0, T]; \mathcal{U}_h)$. The semi-discrete cost functional

$$\begin{aligned} \mathcal{J}_h(\mathbf{u}^h) &= \int_0^T \int_{\Omega} j_{\Omega}[C_{\Omega} \mathbf{u}^h] \, d\mathbf{x} \, dt + \int_0^T \int_{\Gamma} j_{\Gamma}[C_{\Gamma} \mathbf{u}^h] \, d\mathbf{s} \, dt \\ &\quad + \int_{\Omega} k_{\Omega}[E_{\Omega} \mathbf{u}^h]_{t=T} \, d\mathbf{x}, \end{aligned} \quad (69)$$

is a discretization of the continuous functional \mathcal{J} in equation (46), and has a variation given by

$$\begin{aligned} \mathcal{J}'_h \mathbf{w}^h &= \int_0^T \left\langle \left(j_\Omega [C_\Omega \mathbf{u}^h(t)] \right)^T, C'_\Omega \mathbf{w}^h(t) \right\rangle_\Omega dt \\ &\quad + \int_0^T \left\langle \left(j_\Gamma [C_\Gamma \mathbf{u}^h(t)] \right)^T, C'_\Gamma \mathbf{w}^h(t) \right\rangle_\Gamma dt \\ &\quad + \left\langle \left(k'_\Omega [E_\Omega, \mathbf{u}^h(T)] \right)^T, E'_\Omega \mathbf{w}^h(T) \right\rangle_\Omega. \end{aligned} \quad (70)$$

A full discretization of the PDE is obtained by discretizing the time derivative in (67) using a Runge–Kutta method [49]. In the following, $\mathbf{u}^n \in \mathcal{U}_h$ is the fully discrete solution at t^n , $\mathbf{U}_i^n \in \mathcal{U}_h$ is the i -th stage vector at time step n , and $T_i^n = t^n + c_i h^{n+1}$ is the stage time moment. The time grid has $N + 1$ points: from $t^0 = 0$, up to $t^N = T$, and $t^{n+1} = t^n + \tau^{n+1}$. For simplicity of notation, we omit the discrete space superscripts in the following discussion. The Runge–Kutta discretization of (67) reads:

$$\begin{aligned} \langle \mathbf{U}_i^n, \mathbf{v} \rangle_\Omega &= \langle \mathbf{u}^n, \mathbf{v} \rangle_\Omega + \tau^{n+1} \sum_{j=1}^s a_{i,j} \left[\mathcal{N}(T_j^n; \mathbf{U}_j^n, \mathbf{v}) \right. \\ &\quad \left. + \langle \mathbf{f}_j^n, \mathbf{v} \rangle + \mathcal{B}(\mathbf{g}_j^n, \mathbf{v}) \right], \quad \forall \mathbf{v} \in \mathcal{U}_h^{\text{tlm}} \\ \langle \mathbf{u}^{n+1}, \mathbf{v} \rangle_\Omega &= \langle \mathbf{u}^n, \mathbf{v} \rangle_\Omega + \tau^{n+1} \sum_{i=1}^s b_i \left[\mathcal{N}(T_i^n; \mathbf{U}_i^n, \mathbf{v}) + \langle \mathbf{f}_i^n, \mathbf{v} \rangle + \mathcal{B}(\mathbf{g}_i^n, \mathbf{v}) \right]. \end{aligned}$$

Due to the linearity of the Runge–Kutta procedure, the TLM of the fully discrete system reads:

$$\begin{aligned} \langle \mathbf{W}_i^n, \mathbf{v} \rangle_\Omega &= \langle \mathbf{w}^n, \mathbf{v} \rangle_\Omega + \tau^{n+1} \sum_{j=1}^s a_{i,j} \left[\mathcal{N}'[\mathbf{U}_j^n](T_j^n; \mathbf{W}_j^n, \mathbf{v}) \right. \\ &\quad \left. + \langle \delta \mathbf{f}_j^n, \mathbf{v} \rangle + \mathcal{B}(\delta \mathbf{g}_j^n, \mathbf{v}) \right], \quad \forall \mathbf{v} \in \mathcal{U}_h \\ \langle \mathbf{w}^{n+1}, \mathbf{v} \rangle_\Omega &= \langle \mathbf{w}^n, \mathbf{v} \rangle_\Omega + \tau^{n+1} \sum_{i=1}^s b_i \left[\mathcal{N}'[\mathbf{U}_i^n](T_i^n; \mathbf{W}_i^n, \mathbf{v}) \right. \\ &\quad \left. + \langle \delta \mathbf{f}_i^n, \mathbf{v} \rangle + \mathcal{B}(\delta \mathbf{g}_i^n, \mathbf{v}) \right]. \end{aligned}$$

The time integration of the cost functional is discretized according to the Runge–Kutta quadrature. The variation of the fully discrete cost functional is:

$$\begin{aligned} \mathcal{J}'_h \mathbf{w} &= \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i=1}^s b_i \left\langle \left(j_\Omega [C_\Omega \mathbf{U}_i^n] \right)^T, C'_\Omega \mathbf{w}(T_i^n) \right\rangle_\Omega \\ &\quad + \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i=1}^s b_i \left\langle \left(j_\Gamma [C_\Gamma \mathbf{U}_i^n] \right)^T, C'_\Gamma \mathbf{w}(T_i^n) \right\rangle_\Gamma \\ &\quad + \left\langle \left(k'_\Omega [E_\Omega \mathbf{u}^N] \right)^T, E'_\Omega \mathbf{w}(t^N) \right\rangle_\Omega. \end{aligned}$$

We rewrite the TLM of the fully discrete system, to outline the use of different discrete test functions $\lambda(t^n, \mathbf{x}) \in \mathcal{U}_h$ (which will later be interpreted as the adjoint variables):

$$\langle \mathbf{w}^0, \lambda^0 \rangle_\Omega = \langle \delta \mathbf{u}^0, \lambda^0 \rangle_\Omega, \quad \forall \lambda^0 \in \mathcal{U}_h$$

$$\begin{aligned}
 \langle \mathbf{W}_i^n, \theta_i^n \rangle_\Omega &= \langle \mathbf{w}^n, \theta_i^n \rangle_\Omega + \tau^{n+1} \sum_{j=1}^s a_{i,j} \left[\mathcal{N}'[\mathbf{U}_j^n] (T_j^n; \mathbf{W}_j^n, \theta_i^n) \right. \\
 &\quad \left. + \langle \delta \mathbf{f}_j^n, \theta_i^n \rangle + \mathcal{B}(\delta \mathbf{g}_j^n, \theta_i^n) \right], \quad \forall \theta_i^n \in \mathcal{U}_h \\
 \langle \mathbf{w}^{n+1}, \lambda^{n+1} \rangle_\Omega &= \langle \mathbf{w}^n, \lambda^{n+1} \rangle_\Omega + \tau^{n+1} \sum_{i=1}^s b_i \left[\mathcal{N}'[\mathbf{U}_i^n] (T_i^n; \mathbf{W}_i^n, \lambda^{n+1}) \right. \\
 &\quad \left. + \langle \delta \mathbf{f}_i^n, \lambda^{n+1} \rangle + \mathcal{B}(\delta \mathbf{g}_i^n, \lambda^{n+1}) \right], \quad \forall \lambda^{n+1} \in \mathcal{U}_h.
 \end{aligned}$$

Consider all of the above relations for $n = 0, \dots, N-1$. We identify the terms involving the same \mathbf{W}_i^n , and \mathbf{w}^n arguments on the left and right hand sides, and obtain the following correspondence:

$$\begin{aligned}
 \langle \mathbf{W}_i^n, \theta_i^n \rangle_\Omega &\leftrightarrow \tau^{n+1} \sum_{\ell=1}^s a_{\ell,i} \mathcal{N}'[\mathbf{U}_i^n] (T_i^n; \mathbf{W}_i^n, \theta_{n,\ell}) \\
 &\quad + \tau^{n+1} b_i \mathcal{N}'[\mathbf{U}_i^n] (T_i^n; \mathbf{W}_i^n, \lambda^{n+1}) \\
 \langle \mathbf{w}^n, \lambda^n \rangle_\Omega &\leftrightarrow \langle \mathbf{w}^n, \lambda_{n+1} \rangle_\Omega + \sum_{i=1}^s \langle \mathbf{w}^n, \theta_i^n \rangle_\Omega.
 \end{aligned}$$

We now define the discrete adjoint system as:

$$\begin{aligned}
 (\mathbf{w}, \theta_i^n)_\Omega &= \tau^{n+1} \mathcal{N}'[\mathbf{U}_i^n] \left(T_i^n; \mathbf{w}, b_i \lambda^{n+1} + \sum_{\ell=1}^s a_{\ell,i} \theta_\ell^n \right) \\
 &\quad - \tau^{n+1} \sum_{i=1}^s b_i \langle (j_\Omega [C_\Omega \mathbf{U}_i^n])^T, C'_\Omega \mathbf{w} \rangle_\Omega \\
 &\quad - \tau^{n+1} \sum_{i=1}^s b_i \langle (j_\Gamma [C_\Gamma \mathbf{U}_i^n])^T, C'_\Gamma \mathbf{w} \rangle_\Gamma, \quad \forall \mathbf{w} \in \mathcal{U}_h^{\text{tlm}} \\
 \langle \mathbf{w}, \lambda^n \rangle_\Omega &= \langle \mathbf{w}, \lambda^{n+1} \rangle_\Omega + \sum_{i=1}^s \langle \mathbf{w}, \theta_i^n \rangle_\Omega, \quad \forall \mathbf{w} \in \mathcal{U}_h^{\text{tlm}}
 \end{aligned} \tag{71}$$

The sum of the TLM relations for $n = 0, \dots, N-1$ gives:

$$\begin{aligned}
 \langle \mathbf{w}^N, \lambda^N \rangle_\Omega &= \langle \delta \mathbf{u}^0, \lambda^0 \rangle_\Omega - \mathcal{J}'_h \mathbf{w}^0 + \left\langle (k'_\Omega [E_\Omega \mathbf{u}^N])^T, E'_\Omega \mathbf{w}^N \right\rangle_\Omega \\
 &\quad + S_{\mathbf{f}} + S_{\mathbf{g}}.
 \end{aligned} \tag{72}$$

where

$$S_{\mathbf{f}} = \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i,j=1}^s a_{i,j} \langle \delta \mathbf{f}_j^n, \theta_i^n \rangle_\Omega + \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i=1}^s b_i \langle \delta \mathbf{f}_i^n, \lambda^{n+1} \rangle_\Omega, \tag{73}$$

and

$$S_{\mathbf{g}} = \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i,j=1}^s a_{i,j} \mathcal{B}(\delta \mathbf{g}^{n,j}, \theta_i^n) + \sum_{n=0}^{N-1} \tau^{n+1} \sum_{i=1}^s b_i \mathcal{B}(\delta \mathbf{g}_i^n, \lambda^{n+1}). \tag{74}$$

We examine in more detail the terms $S_{\mathbf{f}}$ and $S_{\mathbf{g}}$. From the correspondence between the discrete adjoint “stages”

$$\theta_j^n \leftrightarrow \tau^{n+1} b_j \lambda^{n+1} + \tau^{n+1} \sum_{i=1}^s a_{i,j} \theta_i^n, \tag{75}$$

we get that

$$S_{\mathbf{f}} = \sum_{n=0}^{N-1} \sum_{j=1}^s \langle \delta \mathbf{f}_j^n, \theta_j^n \rangle_{\Omega} \quad (76a)$$

$$S_{\mathbf{g}} = \sum_{n=0}^{N-1} \sum_{j=1}^s \mathcal{B}(\delta \mathbf{g}_j^n, \theta_j^n) . \quad (76b)$$

Following Hager [39], we perform a change of variables in (76) using the correspondence (75). First, from (75), assuming all Runge-Kutta coefficients $b_i \neq 0$, we get

$$\frac{1}{\tau^{n+1} b_j} \theta_j^n \leftrightarrow \lambda^{n+1} + \sum_{\ell=1}^s \frac{a_{\ell,j}}{b_j} \theta_{\ell}^n .$$

Let $\tilde{\theta}_j^n$ denote the stages of the formal adjoint Runge–Kutta method (see [73]), where

$$\tilde{\theta}_j^n := \frac{1}{\tau^{n+1} b_j} \theta_j^n .$$

Then, the formal adjoint stage correspondence becomes

$$\tilde{\theta}_j^n \leftrightarrow \lambda^{n+1} + \tau^{n+1} \sum_{\ell=1}^s \frac{a_{\ell,j} b_{\ell}}{b_j} \tilde{\theta}_{\ell}^n .$$

Replacing this expression in equation (76), we arrive at:

$$S_{\mathbf{f}} = \sum_{n=0}^{N-1} \tau^{n+1} \sum_{j=1}^s b_j \langle \tilde{\theta}_j^n, \delta \mathbf{f}_j^n \rangle_{\Omega} \approx \langle \lambda, \delta \mathbf{f} \rangle_{[0,T] \times \Omega} . \quad (77)$$

The last (approximate) equality follows from the consistence theory of Runge–Kutta quadratures for time integrals [49]. We note that for control problems (unlike inverse problems), some additional order conditions are needed for the formal adjoints of Runge–Kutta methods to achieve orders 3 and above [39]. A similar result can be derived for $S_{\mathbf{g}}$, namely:

$$S_{\mathbf{g}} \approx \int_0^T \mathcal{B}(\delta \mathbf{g}, \lambda) dt .$$

We define the final adjoint condition by

$$\langle \lambda^N, \mathbf{w} \rangle_{\Omega} = \left\langle \left(k'_{\Omega} [E_{\Omega} \mathbf{u}^N] \right)^T, E'_{\Omega} \mathbf{w} \right\rangle_{\Omega}, \quad \forall \mathbf{w} \in \mathcal{U}_h^{\text{tlim}} . \quad (78)$$

Then, (72) becomes:

$$\mathcal{J}'_h \mathbf{w} \approx \langle \delta \mathbf{u}^0, \lambda^0 \rangle_{[0,T] \times \Omega} + \langle \delta \mathbf{f}, \lambda \rangle_{[0,T] \times \Omega} + \int_0^T \mathcal{B}(\delta \mathbf{g}, \lambda) dt .$$

The discrete adjoint variables λ^n can yield different sensitivities, depending on the direction in which the Fréchet derivative of \mathcal{J}_h is computed:

- Differentiation of \mathcal{J}_h along $(\delta \mathbf{u}^0, 0, 0)$ yields the gradient of the cost functional with respect to the initial conditions:

$$\left(E_{\Omega}^{\text{adj}} \right)^h \lambda^0 = \frac{d\mathcal{J}_h}{d\mathbf{u}^0} . \quad (79)$$

- If the tangent linear model is obtained by linearization around $(0, \delta \mathbf{f}, 0)$, then we obtain the sensitivities with respect to changes in the primal equation volume forcing:

$$\left(C_{\Omega}^{\text{adj}}\right)^h \lambda^n = \frac{d\mathcal{J}_h}{d\mathbf{f}^n}. \quad (80)$$

- The consistency of the boundary sensitivities does not follow directly from the consistency of the dual discretization. Indeed, for the example given in the next section, we obtain inconsistent boundary sensitivities from a dual consistent DG discretization. Along with dual consistency of the DG discretization, one also needs adjoint consistency for the boundary functional \mathcal{B} (defined by the primal discretization). We say that \mathcal{B} is dual consistent *iff*, for any admissible boundary perturbation $\delta \mathbf{g}$, there exists a consistent discretization $\left(C_{\Gamma}^{\text{adj}}\right)^h$ of the continuous differential operator C_{Γ}^{adj} , such that

$$\mathcal{B}(\delta \mathbf{g}, \lambda) = \left\langle \delta \mathbf{g}, \left(C_{\Gamma}^{\text{adj}}\right)^h \lambda \right\rangle_{\Gamma}. \quad (81)$$

This does not hold true for all discretizations. The next section will look at the symmetric interior penalty DG discretization of the advection-diffusion system (66). While the discretization itself is dual consistent, the boundary functional \mathcal{B}^h is shown to be adjoint inconsistent.

Note that the discrete adjoint model (71)–(78) is obtained by applying the discrete Runge–Kutta adjoint numerical method to the semi-discrete adjoint system

$$\begin{aligned} \left\langle \mathbf{w}, \frac{\partial \lambda}{\partial t} \right\rangle_{\Omega} &= \mathcal{N}'[\mathbf{u}](t; \mathbf{w}, \lambda) - \langle \mathbf{w}, j_{\Omega}[C_{\Omega} \mathbf{u}] \rangle_{\Omega} \\ &\forall \mathbf{w} \in \mathcal{L}^2([0, T]; \mathcal{U}_h^{\text{tIm}}), \text{ a.a. } t \in [0, T]. \end{aligned} \quad (82)$$

According to [40] the discrete adjoint Runge Kutta method provides the same order of consistency as the forward Runge Kutta method.

In conclusion, the fully discrete adjoint model (71)–(78) is equivalent to applying a method of lines discretization to the continuous adjoint PDE. The space discretization is done with the discrete adjoint DG method, and is consistent with the same order as the forward DG discretization. The time discretization is done with the discrete Runge Kutta adjoint method; the time consistency of the adjoint discretization is the same as the one of the forward method.

7.1. Space-time consistency analysis of the upwind SIPG advection-diffusion DG discretization

The upwind penalty DG semi-discretization [37] for the advection-diffusion PDE (66) reads:

$$\left\langle \frac{\partial \mathbf{u}^h}{\partial t}, \mathbf{v}^h \right\rangle_{\Omega} = \mathcal{N}(\mathbf{u}^h, \mathbf{v}^h) + \mathcal{B}(\mathbf{g}^h, \mathbf{v}^h) - \mathcal{L}(\mathbf{f}^h, \mathbf{v}^h), \quad (83)$$

with

$$\mathcal{N}(\mathbf{u}^h, \mathbf{v}^h) := \mathcal{N}_{\text{diff}}(\mathbf{u}^h, \mathbf{v}^h) + \mathcal{N}_{\text{adv}}(\mathbf{u}^h, \mathbf{v}^h),$$

and

$$\begin{aligned} \mathcal{N}_{\text{adv}}(\mathbf{u}^h, \mathbf{v}^h) &= - \int_{\Omega} \mathbf{u}^h \vec{a} \cdot \nabla \mathbf{v}^h \, d\mathbf{x} + \sum_{D^k} \int_{\partial D^k \setminus \Gamma} \vec{a} \cdot \vec{\mathbf{n}} \mathbf{u}^h_- \mathbf{v}^h_+ \, ds \\ &\quad + \sum_{D^k} \int_{\partial D^k_+} \vec{a} \cdot \vec{\mathbf{n}} \mathbf{u}^h_+ \mathbf{v}^h_+ \, ds. \\ \mathcal{N}_{\text{diff}}(\mathbf{u}^h, \mathbf{v}^h) &= \int_{\Omega} \nabla \mathbf{u}^h \cdot \nabla \mathbf{v}^h \, d\mathbf{x} + \sum_{D^k} \int_{\partial D^k \setminus \Gamma} \frac{1}{2} \theta \llbracket \mathbf{u}^h \rrbracket \cdot \nabla \mathbf{v}^h \, ds \\ &\quad - \sum_{D^k} \int_{\partial D^k \setminus \Gamma_N} \{ \nabla \mathbf{u}^h \} \cdot \vec{\mathbf{n}} \mathbf{v}^h \, ds + \sum_{D^k} \int_{\partial D^k} \phi \llbracket \mathbf{u}^h \rrbracket \cdot \vec{\mathbf{n}} \mathbf{v}^h \, ds \\ &\quad + \int_{\Gamma_D} \theta \mathbf{u}^h \vec{\mathbf{n}} \cdot \nabla \mathbf{v}^h \, ds, \end{aligned}$$

Furthermore,

$$\begin{aligned} \mathcal{L}(\mathbf{f}^h, \mathbf{v}^h) &= \int_{\Omega} \mathbf{f}^h \mathbf{v}^h \, d\mathbf{x} \\ \mathcal{B}(\mathbf{g}^h, \mathbf{v}^h) &= \int_{\Gamma_D} \vec{a} \cdot \vec{\mathbf{n}} \mathbf{g}^h_D \mathbf{v}^h \, ds - \int_{\Gamma_D} \theta \mathbf{g}^h_D \nabla \mathbf{v}^h \cdot \vec{\mathbf{n}} \, ds \\ &\quad - \int_{\Gamma_D} \phi \mathbf{g}^h_D \mathbf{v}^h \, ds - \int_{\Gamma_N} \mathbf{g}^h_N \mathbf{v}^h \, ds. \end{aligned}$$

Here we denote the penalization parameter by $\phi \geq \phi_0 > 0$. For the symmetric interior penalty method (SIPG) [74], $\theta = -1$. The residual form discrete adjoint of the bilinear forms $\mathcal{N}_{\text{diff}}$ and \mathcal{N}_{adv} follows from integrating by parts the primal discretizations [37]. Below λ^h is the discrete adjoint variable, and $\mathbf{w}^h \in \mathcal{U}_h$ denote the test functions:

$$\begin{aligned} \mathcal{N}_{\text{adv}}^*(\mathbf{w}^h, \lambda^h) &:= - \int_{\Omega} \mathbf{w}^h \vec{a} \cdot \nabla \lambda^h \, d\mathbf{x} + \sum_{D^k} \int_{\partial D^k_+ \setminus \Gamma} \mathbf{w}^h_+ \vec{a} \cdot \llbracket \lambda^h \rrbracket \, ds \\ &\quad + \int_{\Gamma_N} \mathbf{w}^h \vec{a} \cdot \vec{\mathbf{n}} \lambda^h \, ds. \\ \mathcal{N}_{\text{diff}}^*(\mathbf{w}^h, \lambda^h) &:= - \int_{\Omega} \mathbf{w}^h \Delta \lambda^h \, d\mathbf{x} \\ &\quad + \sum_{D^k} \int_{\partial D^k \setminus \Gamma} \mathbf{w}^h \left[\frac{1}{2} \llbracket \nabla \lambda^h \rrbracket + (1 + \theta) \vec{\mathbf{n}} \cdot \{ \nabla \lambda^h \} + \phi \llbracket \lambda^h \rrbracket \cdot \vec{\mathbf{n}} \right] \, ds \\ &\quad - \sum_{D^k} \int_{\partial D^k \setminus \Gamma} \frac{1}{2} \nabla \mathbf{w} \cdot \llbracket \lambda^h \rrbracket \, ds + \int_{\Gamma_N} \mathbf{w}^h \vec{\mathbf{n}} \cdot \nabla \lambda^h \, ds \\ &\quad + \int_{\Gamma_D} \mathbf{w}^h \left[(1 + \theta) \vec{\mathbf{n}} \cdot \nabla \lambda^h + \phi \lambda^h \right] \, ds - \int_{\Gamma_D} \nabla \mathbf{w}^h \cdot \vec{\mathbf{n}} \lambda^h \, ds. \end{aligned}$$

The semi-discrete formulation of (67) reads:

$$\mathcal{J}_h(\mathbf{u}^h) := \frac{1}{2} \int_0^T \int_{\Omega} \left\| \mathbf{u}^h - \mathbf{u}^{\text{ref}} \right\|^2 \, d\mathbf{x} + \frac{1}{2} \int_0^T \int_{\Gamma_D} (\nabla \mathbf{u}^h \cdot \vec{\mathbf{n}})^2 \, ds,$$

hence its Fréchet derivative is calculated as

$$\mathcal{J}'_h[\mathbf{u}^h](\mathbf{w}^h) = \left\langle \mathbf{u}^h - \mathbf{u}^{\text{ref}}, \mathbf{w}^h \right\rangle_{[0, T] \times \Omega} + \left\langle \nabla \mathbf{u}^h \cdot \vec{\mathbf{n}}, \nabla \mathbf{w}^h \cdot \vec{\mathbf{n}} \right\rangle_{[0, T] \times \Gamma_D}.$$

The semi-discrete adjoint equation has the following form:

$$-\left\langle \mathbf{w}^h, \frac{\partial \lambda^h}{\partial t} \right\rangle_{\Omega} = -\mathcal{N}_{\text{diff}}^*(\mathbf{w}^h, \lambda^h) - \mathcal{N}_{\text{adv}}^*(\mathbf{w}^h, \lambda^h) + \mathcal{J}'_h[\mathbf{u}^h](\mathbf{w}^h). \quad (84)$$

To investigate the dual consistency of the adjoint residuals for our particular discretization, we must first recast (84) in residual-based form [37]. We get:

$$\begin{aligned} \int_{\Omega} \mathbf{w}^h \mathbf{R}_{\Omega}^*(\lambda^h) d\mathbf{x} + \sum_{D^k} \int_{\partial D^k \setminus \Gamma} \mathbf{w}^h \mathbf{r}_{\Omega}^*(\lambda^h) + \nabla \mathbf{w}^h \cdot \rho_{\Omega}^*(\lambda^h) ds \\ + \int_{\Gamma} \mathbf{w}^h \mathbf{r}_{\Gamma}^*(\lambda^h) + \nabla \mathbf{w}^h \cdot \rho_{\Gamma}^*(\lambda^h) ds = 0, \quad \forall \mathbf{w}^h \in \mathcal{U}_h. \end{aligned} \quad (85)$$

From (84)–(85), we identify the following dual residuals:

- Inside Ω :

$$\mathbf{R}_{\Omega}^*(\lambda^h) := -\frac{\partial \lambda^h}{\partial t} + \Delta \lambda^h + \vec{a} \cdot \nabla \lambda^h + (\mathbf{u}^h - \mathbf{u}^{\text{ref}}).$$

From the continuous adjoint equation inside Ω we see that $\mathbf{R}_{\Omega}^*(\lambda) = 0$, so the volume terms of the adjoint semi-discretization (84) are dual consistent.

- On the inter-element boundaries (excluding the domain boundary):

$$\begin{aligned} \mathbf{r}_{\Omega}^*(\lambda^h) &= -\vec{a} \cdot \llbracket \lambda^h \rrbracket - \frac{1}{2} \llbracket \nabla \lambda^h \rrbracket - (1 + \theta) \vec{\mathbf{n}} \cdot \{ \nabla \lambda^h \} + \phi \llbracket \lambda^h \rrbracket \cdot \vec{\mathbf{n}}, \\ \rho_{\Omega}^*(\lambda^h) &= \frac{1}{2} \llbracket \lambda^h \rrbracket. \end{aligned}$$

Using the continuity of the strong form adjoint solution λ , and the fact that $\theta = -1$ for SIPG, we get that both dual residuals are zero when evaluated at λ .

- On the outflow boundary (with respect to the advective flux), Γ_N :

$$\begin{aligned} \mathbf{r}_{\Gamma_N}^*(\lambda^h) &= -\lambda^h \vec{a} \cdot \vec{\mathbf{n}} - \vec{\mathbf{n}} \cdot \nabla \lambda^h + \kappa_N^h, \\ \rho_{\Gamma_N}^*(\lambda^h) &= 0. \end{aligned}$$

Due to the boundary condition of the continuous adjoint system, we have that $\mathbf{r}_{\Gamma_N}^*(\lambda) = 0$. Thus (84) is adjoint consistent on the outflow boundary.

- On the Dirichlet boundary Γ_D :

$$\mathbf{r}_{\Gamma_D}^*(\lambda^h) = -(1 + \theta) \vec{\mathbf{n}} \cdot \nabla \lambda^h - \phi \lambda^h, \quad (86a)$$

$$\rho_{\Gamma_D}^*(\lambda^h) = (\nabla \mathbf{u}^h \cdot \vec{\mathbf{n}} + \lambda^h) \vec{\mathbf{n}}. \quad (86b)$$

While these residuals do not cancel immediately when evaluated at the exact adjoint solution (recall that $\theta = -1$, and $\phi > 0$), they can be made consistent through a change in the target functional \mathcal{J} [37]. Let

$$\widetilde{\mathcal{J}}_h(\mathbf{u}^h) := \mathcal{J}_h(\mathbf{u}^h) - \int_{\Gamma_D} \phi (\mathbf{u}^h - \mathbf{g}_D^h) (\nabla \mathbf{u}^h \cdot \vec{\mathbf{n}}) ds \quad (87)$$

be a consistent modification of \mathcal{J}_h , since $\widetilde{\mathcal{J}}(\mathbf{u}) = \mathcal{J}(\mathbf{u})$. The variation of the modified cost functional (87) is

$$\begin{aligned} \widetilde{\mathcal{J}}'_h[\mathbf{u}^h](\mathbf{w}^h) &= \mathcal{J}'_h[\mathbf{u}^h](\mathbf{w}^h) + \left\langle \phi \nabla \mathbf{u}^h \cdot \vec{\mathbf{n}}, \mathbf{w}^h \right\rangle_{\Gamma_D} \\ &\quad + \left\langle \phi (\mathbf{u}^h - \mathbf{g}_D^h), \nabla \mathbf{w}^h \cdot \vec{\mathbf{n}} \right\rangle_{\Gamma_D}. \end{aligned}$$

All the discrete adjoint residuals remain unchanged, except for (86)–(86), which now become:

$$\mathbf{r}_{\Gamma_D}^*(\lambda^h) = -(1 + \theta) \vec{\mathbf{n}} \cdot \nabla \lambda^h - \phi \left(\lambda^h + \nabla \mathbf{u}^h \cdot \vec{\mathbf{n}} \right), \quad (88a)$$

$$\rho_{\Gamma_D}^*(\lambda^h) = (\nabla \mathbf{u}^h \cdot \vec{\mathbf{n}} + \lambda^h) \vec{\mathbf{n}} + \phi (\mathbf{u}^h - \mathbf{g}_D^h) \vec{\mathbf{n}}. \quad (88b)$$

Both residuals (88)–(88) are now identically zero when evaluated at the exact adjoint solution λ . We have thus proved dual consistency for the DG discretization (83) coupled with the modified functional (87).

7.1.1. Discrete adjoint boundary sensitivities Now let us consider the boundary bilinear form $\mathcal{B}(\mathbf{g}^h, \mathbf{v}^h)$. For any admissible perturbation $\delta \mathbf{g} := (\delta \mathbf{g}_D, \delta \mathbf{g}_N)$ in the boundary conditions, we get that:

$$\begin{aligned} \mathcal{B}(\delta \mathbf{g}^h, \lambda^h) &= \left\langle \delta \mathbf{g}_D, (C_{\Gamma_D}^{\text{adj}})^h \lambda^h \right\rangle_{\Gamma_D} + \left\langle \delta \mathbf{g}_N, (C_{\Gamma_N}^{\text{adj}})^h \lambda^h \right\rangle_{\Gamma_N} \\ &\quad - \phi \left\langle \delta \mathbf{g}_D, \lambda^h \right\rangle_{\Gamma_D}, \end{aligned} \quad (89)$$

where $(C_{\Gamma_D}^{\text{adj}})^h$ and $(C_{\Gamma_N}^{\text{adj}})^h$ are consistent discretizations of the continuous differential operators defined in (67). Note the additional boundary penalty term: the nonzero penalty parameter ϕ ensures stability and convergence of the method. However, it also leads to inconsistencies in the adjoint boundary sensitivities, that must be removed by post-processing of the adjoint Runge–Kutta DG implementation. The adjoint Runge–Kutta time integration, albeit consistent, cannot remove the inconsistent term in the sensitivity formula (89). Proving dual consistency is a crucial step in the analysis of a dual DG discretization, allowing one to establish whether or not the adjoint variable corresponds to the true gradient of the discretized cost functional \mathcal{J}_h . However, if one also seeks derivatives with respect to the boundary values, further investigations pertaining to \mathcal{B} are warranted, that go beyond establishing dual consistency of the primal discretization.

8. A two-dimensional inverse problem

8.1. Problem description

The second test problem is built around the two-dimensional advection equation:

$$\begin{aligned} \mathbf{u}_t + \nabla \cdot (\vec{\beta} \mathbf{u}) &= \mathbf{f} \\ \mathbf{u}(t, \mathbf{x})|_{\Gamma_{\text{in}}} &= \mathbf{g} \\ \mathbf{u}(t^0, \mathbf{x}) &= \mathbf{u}_0(\mathbf{x}), \quad t^0 \leq t \leq t^N, \quad \mathbf{x} \in \Omega. \end{aligned} \quad (90)$$

Here $\Omega = [0, 1]^2$, $\vec{\beta} := \mathbf{x} / \|\mathbf{x}\|$, and $\Gamma_{\text{in}} := \{\mathbf{x} \in \Gamma \mid \vec{\mathbf{n}} \cdot \vec{\beta} < 0\}$.

The discrete cost functional \mathcal{J}_h reads:

$$\mathcal{J}_h(\mathbf{u}^{h,0}) = \mathcal{J}_{h,B} + \mathcal{J}_{h,O}$$

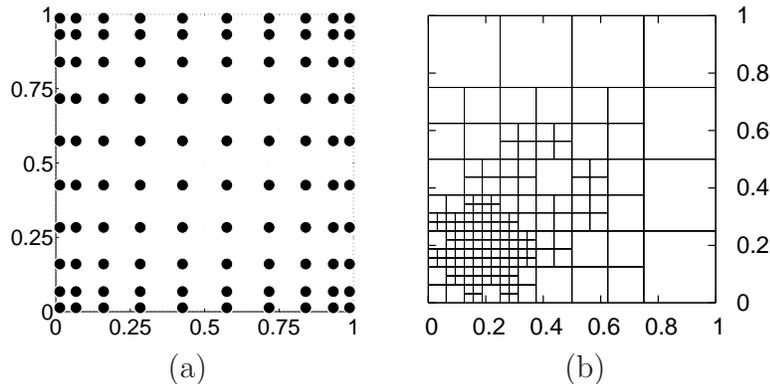


Figure 1. (a) Observation grid for the two-dimensional assimilation problem. (b) Optimization grid Ω_0^h that holds the parameters $\mathbf{p}^h = \mathbf{u}^{h,0}$ throughout the inversion process.

$$\begin{aligned}
&= \frac{1}{2} \left(\mathbf{u}^{h,0} - \mathbf{u}^B \right)^T B^{-1} \left(\mathbf{u}^{h,0} - \mathbf{u}^B \right) \\
&\quad + \frac{1}{2} \sum_{k=1}^K \left(\mathcal{H}_k \mathbf{u}^{h,k} - \mathbf{y}^{h,k} \right)^T R_k^{-1} \left(\mathcal{H}_k \mathbf{u}^{h,k} - \mathbf{y}^{h,k} \right). \quad (91)
\end{aligned}$$

The background term $\mathcal{J}_{h,B}$ quantifies the departure of the inverse solution from a background state $\mathbf{u}_B^{h,0}$. It also acts as a regularization term that guarantees the inverse problem is well-posed. $\mathcal{J}_{h,O}$ quantifies the mismatch between the model predictions and a set of *a priori* available observations $\mathbf{y}^{h,k}$ at selected grid locations and observation times t^k . Our particular choice of observation mesh is shown in figure 1(a).

8.2. The experimental setup

The primal and adjoint RK-DG discretizations are implemented with the `deal.II` library [75]. The optimization routine is a C++ implementation [76] of the well-known L-BFGS-B algorithm [48]. The mesh adaptation is driven by an error estimation mechanism based on a numerical approximation of the gradient $\frac{\partial \mathbf{u}^h}{\partial \mathbf{x}^h}$ [29]. The optimization mesh Ω_0^h holds the inversion variables throughout the optimization process. It is shown in figure 1(b) to be locally refined in regions of high variation in the background state. The final time in the forward simulation is $T = 0.48$, while the observation times are $t^k = 0.03 \times k$, $k = 1 \dots 16$.

8.3. Space-time consistency and accuracy of the discrete adjoint solution

To check the consistency and the empirical order of accuracy of the discrete adjoint solver, we will derive the corresponding continuous adjoint problem. Let $\delta(t)$ denote the Dirac delta distribution. We can rewrite the 4D-Var discrete cost functional as

$$\mathcal{J}_h := \sum_{k=0}^K \widehat{\mathcal{J}}_h(\mathbf{u}^{h,k}) := \int_{t^0}^{t^N} \widehat{\mathcal{J}}_h(\mathbf{u}^h, t) \sum_{k=0}^K \delta(t - t^k) dt. \quad (92)$$

Then, the strong form adjoint of (90)–(92) reads:

$$\begin{aligned}
 -\lambda_t - \vec{\beta} \cdot (\nabla \lambda) &= \frac{\partial \widehat{\mathcal{J}}}{\partial \mathbf{u}} \sum_{k=0}^K \delta(t - t^k), \quad \mathbf{x} \in \Omega, t \in [t^N, 0] \\
 B_{\Gamma}^{\text{adj}} \lambda := \vec{\beta} \cdot \vec{\mathbf{n}} \lambda &= 0, \quad \mathbf{x} \in \Gamma_{\text{out}} = \Gamma \setminus \Gamma_{\text{in}} \\
 \lambda(t^N, \mathbf{x}) &= 0.
 \end{aligned} \tag{93}$$

The exact solution to the inverse problem is chosen to be

$$\mathbf{u}^0(x, y) = A \exp\left(-\frac{(xs - x_c)^2}{\sigma^2}\right) \exp\left(-\frac{(ys - y_c)^2}{\sigma^2}\right), \tag{94}$$

where $A = 10$, $s = 20$, $\sigma = 2$, and $x_c = y_c = 4$.

We use the dual consistent upwind spatial discretization given in [37]. The particular form of the cost functional (91) implies that the discrete adjoint system has a forcing term only at the observation times t^k (93), where it is necessary to add the observation mismatch [77]:

$$\lambda^{h,k} = \lambda^{h,k} + \mathcal{H}_k^T \mathcal{R}^{-1} \left(\mathcal{H}_k \mathbf{u}^{h,k} - \mathbf{y}^{h,k} \right), \quad k = 1 \dots K. \tag{95}$$

The equation (93) is not in conservation form. We rewrite the nonconservative term as

$$\vec{\beta} \cdot \nabla \mathbf{u} = \nabla \cdot (\vec{\beta} \mathbf{u}) - (\nabla \cdot \vec{\beta}) \mathbf{u},$$

Through a Galerkin projection onto the discrete function space \mathcal{U}_h and integration by parts, we arrive at the DG semi-discretization of (93):

$$\begin{aligned}
 \text{For all } n, \text{ find } \bar{\lambda}^{h,n} \text{ such that on } \Omega_n^h, \forall \bar{\mathbf{w}}^h \in \mathcal{U}_h : \\
 - \left\langle \frac{\partial \bar{\lambda}^{h,n}}{\partial t}, \bar{\mathbf{w}}^h \right\rangle_{\Omega} + \sum_{D^k} \left(\left\langle \bar{\lambda}^{h,n+1}, \vec{\beta} \cdot \nabla \bar{\mathbf{w}}^h \right\rangle_{D_n^k} - \left\langle \vec{\beta} \cdot \vec{\mathbf{n}} \bar{\lambda}_-^{h,n+1}, \bar{\mathbf{w}}_+^h \right\rangle_{\partial D_{n+}^k} \right. \\
 \left. - \left\langle \vec{\beta} \cdot \vec{\mathbf{n}} \bar{\lambda}_-^{h,n+1}, \bar{\mathbf{w}}_+^h \right\rangle_{\partial D_{n-}^k \setminus \Gamma_n^h} + \left\langle \nabla \cdot \vec{\beta} \bar{\lambda}^{h,n+1}, \bar{\mathbf{w}}^h \right\rangle_{D_n^k} \right) = 0.
 \end{aligned} \tag{96}$$

Again, at the observation times t^k , we add the mismatch term in (95) to the solution $\bar{\lambda}^{h,n}$.

The dual consistency of the spatial discretization, together with a third order strong stability preserving Runge-Kutta method [78] for time integration, ensure space-time dual consistency of our RK-DG discretization for (90). Moreover, the spatial and temporal order of accuracy of the discrete adjoint solution match that of the corresponding discretization of the continuous model, i.e., in the limit of both the discretizations we have that:

$$\lim_{\Delta t^n, h \rightarrow 0} \frac{\left\| \lambda^{h,n} - \lambda(t^n) \right\|_{L^2(\Omega_n^h)}}{\left\| \bar{\lambda}^{h,n} - \lambda(t^n) \right\|_{L^2(\Omega_n^h)}} = \mathcal{O}(1), \quad \forall n = 1 \dots N. \tag{97}$$

To verify this numerically, both the discrete adjoint, and the discretization of the adjoint problem (96) are compared against a predetermined exact solution

$$\lambda(t, x, y) = \mathbf{u}^0(x - t, y - t). \tag{98}$$

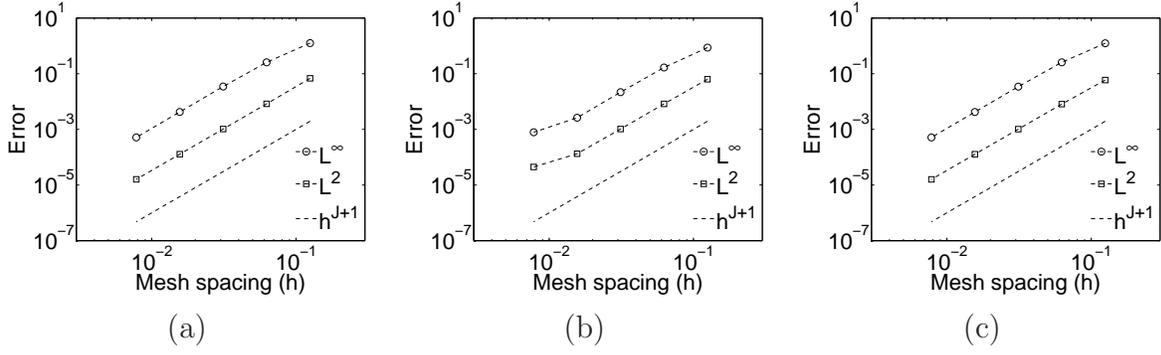


Figure 2. Time-averaged L^2 and L^∞ errors for: (a) the forward state $\mathbf{u}^{h,n}$, (b) the continuous adjoint solution $\bar{\lambda}^{h,n}$, and (c) the discrete adjoint variables $\lambda^{h,n}$. The exact solutions are given by (94), (98), and (99). We use a quadratic Lagrange basis over an uniform mesh. The time integration is performed with a third order fixed-step TVD Runge-Kutta method: $\tau^{n+1} = \tau$, $\forall n = 0 \dots N - 1$. The convergence order is $\mathcal{O}(h^3 + \tau^3)$ for all numerical approximations.

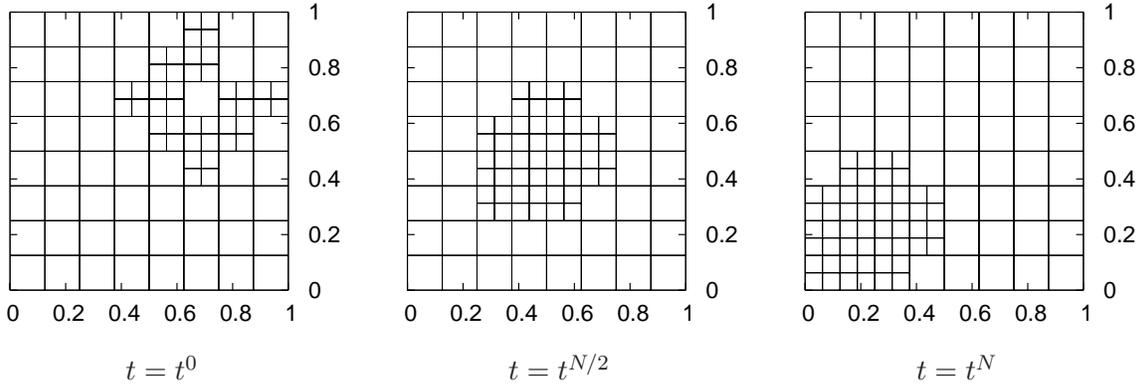


Figure 3. Adaptive spatial meshes used in the determination of the numerical order of accuracy for $\bar{\lambda}^{h,n}$, and $\mathbf{u}^{h,n}$.

Here \mathcal{U}_h are broken spaces of piecewise quadratic Lagrange polynomials. For the primal problem, the volume and boundary forcing terms \mathbf{f} and \mathbf{g} are chosen such that

$$\mathbf{u}(t, x, y) = \mathbf{u}^0(x - t, y - t). \quad (99)$$

Figure 2 shows the order of convergence of the RK-DG discretizations on fixed spatial meshes. To illustrate the behavior of the discretization on adaptive meshes, we run the accuracy experiments on a variable mesh and report the numerical results in Figure 4. All of the numerical results fully confirm our theoretical derivations: both adjoint solutions are third order accurate in space and time. Hence, the adjoint of the primal discretization inherits the order of accuracy of the discrete forward solution.

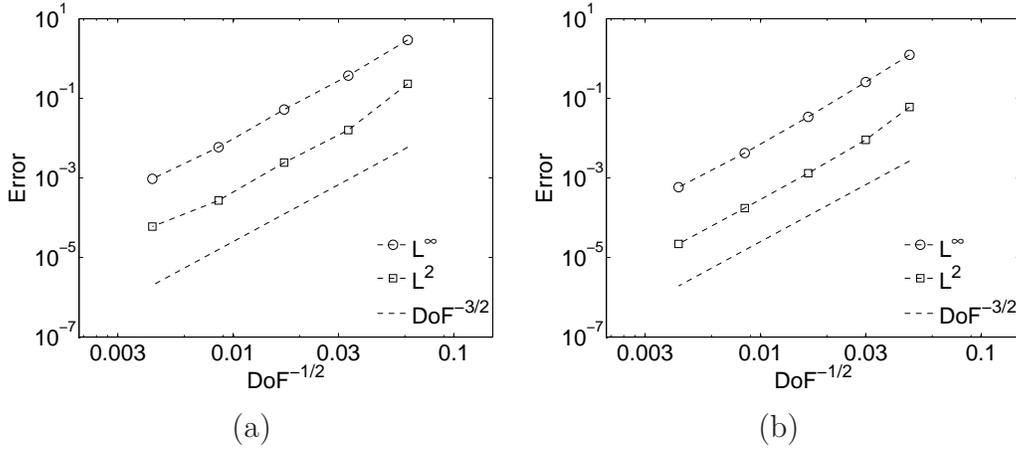


Figure 4. Time-averaged L^2 and L^∞ errors plotted against the mesh degrees of freedom for $\bar{\lambda}^{h,n}$ (left), and $\lambda^{h,n}$ (right). The exact solutions are given by (94), (98), and (99). We use a quadratic Lagrange basis over an adaptive spatial mesh, and a third order Runge–Kutta method for the time integration. The cubic convergence confirms the theoretical estimates for the adaptive DG discretization.

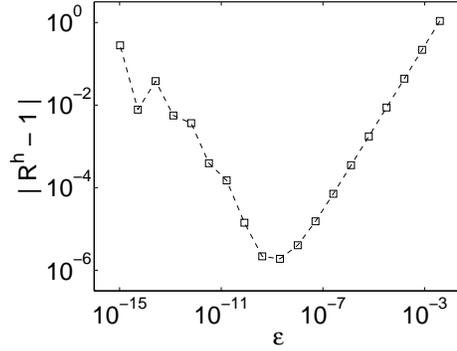


Figure 5. Numerical validation of the discrete adjoint solution using equation (100).

Another approach to adjoint code validation is through a truncated Taylor expansion [79]:

$$\mathcal{J}_h(\mathbf{p}^h + \varepsilon \delta \mathbf{p}^h) = \mathcal{J}_h(\mathbf{p}^h) + \varepsilon \langle \lambda^h, \delta \mathbf{p}^h \rangle_{\Omega^h} + \mathcal{O}(\varepsilon^2 \|\delta \mathbf{p}^h\|^2).$$

Hence, we numerically verify that the following limit holds for small values of ε :

$$\lim_{\varepsilon \rightarrow 0} R^h := \lim_{\varepsilon \rightarrow 0} \frac{\mathcal{J}_h(\mathbf{p}^h + \varepsilon \delta \mathbf{p}^h) - \mathcal{J}_h(\mathbf{p}^h)}{\varepsilon \langle \lambda^h, \delta \mathbf{p}^h \rangle_{\Omega^h}} = 1. \quad (100)$$

As shown in figure 5, the variable mesh discrete adjoint solution is found numerically consistent. For $\varepsilon < 10^{-12}$, truncation errors degrade the quality of the approximation (100).

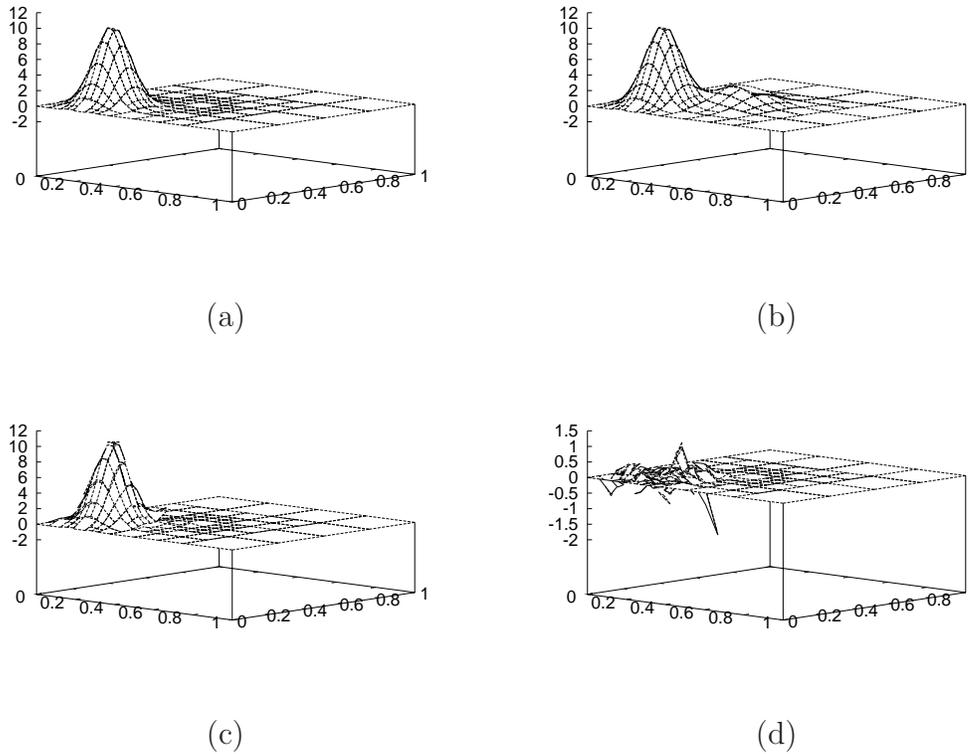


Figure 6. Reference (a), background (b), and analysis (c) states for the two-dimensional data assimilation problem, with a measurement noise level of 5%. The analysis error is shown in (d).

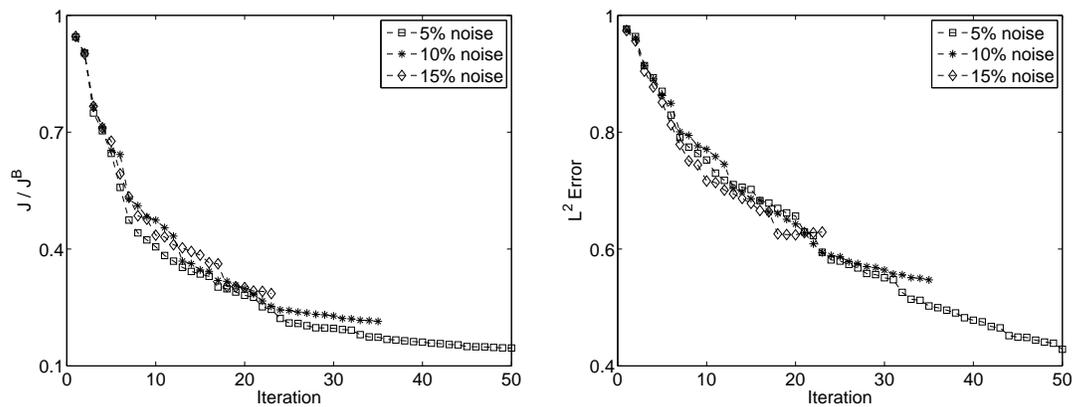


Figure 7. Relative decrease in the cost functional, and in the L^2 -error for the two-dimensional data assimilation experiments, plotted against the number of optimization iterations. Various observation noise levels are shown.

8.4. Numerical results

The numerical results for the two-dimensional data assimilation experiment are shown in Figures 6 and 7. Figure 6 (a)–(b) shows the background (the *a priori* state), and

the reference solution. Parts (c) and (d) of the same figure illustrates the analysis state, and the analysis error, respectively. It is apparent that the quality of the solution approximation is improved significantly over that of the initial guess (the background).

Figure 7 quantifies these improvements. We plot both the decrease in the cost functional throughout the optimization procedure, relative to the initial value at the background state $\mathcal{J}_h(\mathbf{u}_B^{0,h}) := \mathcal{J}^B$. The third order accurate primal and dual solutions lead to a good reconstruction of the optimal solution in our *twin* experiment. The robustness and accuracy of the inversion procedure are tested with various levels of uniformly-distributed noise in the observation values. As expected, the quality of the analysis solution degrades when the noise level is increased. However, as Figure 7 shows, we still get a significant decrease in both the cost functional \mathcal{J}_h , and in the analysis error. This indicates good performance and robustness for the discrete adjoint-based adaptive inversion procedure.

9. Conclusions and future research directions

Numerical solutions of forward problems modeled by partial differential equations rely on adaptive refinements of meshes and time steps to achieve the target accuracy while keeping the simulation costs low. Time and space adaptivity are highly desirable features in the solution of large scale inverse problems as well. This paper develops a framework for the construction and analysis of discrete adjoints for time dependent, adaptive grid, adaptive step models. The research shows that discontinuous Galerkin space discretizations, paired with Runge Kutta time stepping, and discrete adjoint gradients, offer a suitable approach to adaptively solving inverse problems.

Previous research has, with few exceptions, concentrated on steady state inverse problems and their associated discretizations. We develop a general framework for the derivation of a well-posed adjoint system for time dependent partial differential equations and general objective functionals. Building on the work in [20], we derive a more general set of compatibility conditions (between the linearized differential operators and the objective functional); when they are satisfied, the derivation of the adjoint system follows the steps outlined in Section 6.

Since inverse problems are usually formulated as deterministic optimization problems, the discrete adjoint approach is a low cost method to calculate the gradient of the target functional whose minimum is sought. A major advantage of this *discretize-then-differentiate* strategy is that discrete adjoints (i.e., gradients of the numerical solution) can be automatically generated using algorithmic differentiation. However, the discrete adjoints may not provide consistent approximations to the continuous gradients (consistency of the dual discretization is not automatic and requires a careful analysis). This work proposes a unified space and time discrete adjoint consistency analysis in the context of adaptive solvers for time-dependent problems. The dual consistency concepts discussed in [37, 38] can be extended to time dependent systems only when the time discretization is performed by discontinuous Galerkin approach. We show that the use

of Runge–Kutta quadratures for time stepping, together with dual consistent spatial discretizations, result in fully space-time dual consistent adjoint systems. Discrete boundary sensitivities obtained with the adjoint method are also examined. Their values may be incorrect even when the volume discretization is dual consistent, due to the presence of discretization-specific penalty terms. Post-processing of the discrete adjoint code is required in this case to retrieve the correct values for the gradients with respect to boundary conditions.

The spectral intergrid projection operators used in h - and p -refinement are orthogonal \mathcal{L}^2 projections. We show that this property holds for both structured (hierarchical) and unstructured mesh refinement with the discontinuous Galerkin method. Orthogonality of the solution transfer operators is important, because discrete adjoint intergrid operators can be generated from their forward counterparts via automatic differentiation. Adjoint code development is thus simplified, since there is no need to decouple the solution transfer across multiple meshes from the numerical core of the algorithm. The intergrid operators in the finite volume approach do not share these convenient properties. As our general analysis shows, the transpose of high-order refinement operators reduces to a simple first-order averaging, when used for mesh coarsening. Moreover, such adjoints affect the rest of the stencil neighbors, by introducing low order perturbations in their cell average values. Removing such perturbations by code post-processing is non-trivial in large-scale simulations.

The use of discrete adjoint method is illustrated on the solution of a typical inverse problem (4D-Var data assimilation) with a prototypical test system (advection equation). The discrete adjoint solutions provide accurate gradients for the 4D-Var cost functional, and result in a robust and accurate inversion process. Good quality analyses are obtained even in the presence of significant observational noise. The use of the same mesh for both the primal and dual variables eliminates the need for spatial or temporal interpolations during inversion.

On-going work is focused on error-driven adaptation of the optimal solution grid based on primal and dual *a posteriori* error estimates. Future research includes demonstrating the discrete adjoint techniques in inverse problems with fully nonlinear models. The authors plan to investigate the space-time optimality system for the discrete problem, and its relationship to the continuous optimality equations. We will look for general error estimates for the discrete primal and dual solutions that guarantee convergence of the inverse problem solution.

Acknowledgments

The authors thank Wolfgang Bangerth and the `deal.II` user community for their kind assistance with the `deal.II` software. We also acknowledge Amik St-Cyr for suggesting the DoF plotting format for the order estimation with the adaptive simulations. This work was supported in part by the US National Science Foundation, through the following awards: NSF-CCF-0635194, NSF OCI-0904397, NSF CCF-0916493, and NSF

DMS-0915047.

References

- [1] S. R. Arridge and J. C. Schotland. Optical tomography: forward and inverse problems. *Inverse Problems*, 25(12):123010, 2009.
- [2] S. L. Cotter, M. Dashti, J. C. Robinson, and A. M. Stuart. Bayesian inverse problems for functions and applications to fluid mechanics. *Inverse Problems*, 25(11):115008, 2009.
- [3] W. E. Heinz, F. Christoph, K. Philipp, J. Lu, S. Müller, and P. Schuster. Inverse problems in systems biology. *Inverse Problems*, 25(12):123014, 2009.
- [4] S. Huang, J. Xiang, H. Du, and X. Cao. Inverse problems in atmospheric science and their application. *Journal of Physics: Conference Series*, 12(1):45, 2005.
- [5] W. C. Thacker. Oceanographic inverse problems. *Phys. D*, 60(1-4):16–37, 1992.
- [6] W. W. Symes. The seismic reflection inverse problem. *Inverse Problems*, 25(12):123008, 2009.
- [7] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM, Philadelphia, PA, USA, 2nd edition, 2005.
- [8] V. Carey, D. Estep, A. Johansson, M. Larson, and S. Tavener. Blockwise adaptivity for time dependent problems based on coarse scale adjoint solutions. *SIAM Journal on Scientific Computing*, 32(4):2121–2145, 2010.
- [9] W. Bangerth. A framework for the adaptive finite element solution of large-scale inverse problems. *SIAM J. Sci. Comput.*, 30(6):2965–2989, 2008.
- [10] W. Bangerth and A. Joshi. Adaptive finite element methods for the solution of inverse problems in optical tomography. *Inverse Problems*, 24(3):034011 (22pp), 2008.
- [11] S. Li and L. Petzold. Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement. *J. Comput. Phys.*, 198(1):310–325, 2004.
- [12] F. Fang, C.C. Pain, M.D. Piggott, G.J. Gorman, and A.J.H. Goddard. An adaptive mesh adjoint data assimilation method applied to free surface flows. *International Journal for Numerical Methods in Fluids*, 47(8-9):995–1001, 2005.
- [13] F. Fang, M.D. Piggott, C.C. Pain, G.J. Gorman, and A.J.H. Goddard. An adaptive mesh adjoint data assimilation method. *Ocean Modelling*, 15(1-2):39–55, 2006. The Third International Workshop on Unstructured Mesh Numerical Modelling of Coastal, Shelf and Ocean Flows.
- [14] F. Fang, C. C. Pain, M. D. Piggott, G. J. Gorman, and A. J. H. Goddard. Data assimilation of three-dimensional free surface flow using an adaptive adjoint method. part 1: Adjoint model formulation. Technical report, Imperial College London, London, UK, 2007.
- [15] F. Fang, C. C. Pain, M.D. Piggott, G. J. Gorman, and A. J. H. Goddard. Data assimilation of three-dimensional free surface flow using an adaptive adjoint method. part 2: Applications. Technical report, Imperial College London, London, UK, 2007.
- [16] M. D. Gunzburger. *Perspectives in Flow Control and Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002.
- [17] A. Griewank and A. Walther. Algorithm 799: Revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software*, 26(1):19–45, 2000.
- [18] E. Arian and M. D. Salas. Admitting the inadmissible: Adjoint formulation for incomplete cost functionals in aerodynamic optimization. Technical report, 1997.
- [19] E. Arian and M. D. Salas. *Adjoint formulation using auxiliary boundary equations (ABEs) demonstrated on the Stokes equations*, volume 515 of *Lecture Notes in Physics*, pages 355–360. Springer Berlin / Heidelberg, 1998.
- [20] M. B. Giles and N. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behavior. In *Proceedings of the 13th Computational Fluid Dynamics Conference*, number AIAA Paper 97-1850, 1997.
- [21] T. Gou, K. Singh, and A. Sandu. Chemical data assimilation with CMAQ: Continuous vs. discrete

- advection adjoints. In *Computational Science – ICCS 2009*, volume 5545 of *Lecture Notes in Computer Science*, pages 312–321. Springer Berlin / Heidelberg, 2009.
- [22] F. Fang, C. C. Pain, I. M. Navon, G. J. Gorman, M. D. Piggott, P. A. Allison, P. E. Farrell, and A. J. H. Goddard. A POD reduced order unstructured mesh ocean modelling method for moderate reynolds number flows. *Ocean Modelling*, 28(1-3):127–136, 2009.
- [23] F. Fang, C.C. Pain, M.D. Piggott, G.J. Gorman, P.A. Allison, and A.J.H. Goddard. A POD goal-oriented error measure for mesh optimisation. *International Journal for Numerical Methods in Fluids*, (Early View), 2009.
- [24] F. Fang, C.C. Pain, I. M. Navon, G.J. Gorman, M.D. Piggott, and P.A. Allison. The independent set perturbation adjoint method: A new method of differentiating mesh based fluids models. (*Submitted for publication to International Journal for Numerical Methods in Fluids*), 2009.
- [25] A. C. Marta, C. A. Mader, J. R. R. A. Martins, E. Van der Weide, and J. J. Alonso. A methodology for the development of discrete adjoint solvers using automatic differentiation tools. *Int. J. Comput. Fluid Dyn.*, 21(9-10):307–327, 2007.
- [26] D. J. Mavriplis. Discrete adjoint-based approach for optimization problem on three-dimensional unstructured meshes. *AAIA Journal*, 45(4):740–750, April 2007.
- [27] M. Giles and S. Ulbrich. Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 1: Linearized approximations and linearized output functionals. *SIAM Journal on Numerical Analysis*, 48(3):882–904, 2010.
- [28] M. Giles and S. Ulbrich. Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. part 2: Adjoint approximations and extensions. *SIAM Journal on Numerical Analysis*, 48(3):905–921, 2010.
- [29] W. Bangerth. *Adaptive Finite Element Methods for the Identification of Distributed Parameters in Partial Differential Equations*. PhD thesis, University of Heidelberg, 2002.
- [30] L. Wang, D. J. Mavriplis, and W. K. Anderson. Unsteady discrete adjoint formulation for high-order discontinuous galerkin discretizations in time dependent flows. In *AIAA Aerospace Sciences Meeting*, number AIAA Paper 2010-0367, 2009.
- [31] N. K. Yamaleev, B. Diskin, and E. J. Nielsen. Local-in-time adjoint-based method for design optimization of unsteady flows. *Journal of Computational Physics*, In Press, Corrected Proof:–, 2010.
- [32] R. Giering. *Tangent linear and Adjoint Model Compiler, Users manual 1.4*, 1999.
- [33] L. Hascöet and V. Pascual. TAPENADE 2.1 User’s guide. Technical Report 0300, INRIA, Sophia Antipolis, France, 2004.
- [34] R. Giering and T. Kaminski. Applying TAF to generate efficient derivative code of Fortran 77-95 programs. *Proceedings in Applied Mathematics and Mechanics*, 2(1):54–57, 2003.
- [35] C. H. Bischof, L. Roh, and A. J. Mauer-Oats. ADIC: an extensible automatic differentiation tool for ANSI-C. *Software: Practice & Experience*, 27(12):1427–1456, 1997.
- [36] S. A. Forth. An efficient overloaded implementation of forward mode automatic differentiation in matlab. *ACM Trans. Math. Softw.*, 32(2):195–222, 2006.
- [37] R. Hartmann. Adjoint consistency analysis of discontinuous Galerkin discretizations. *SIAM J. Numer. Anal.*, 45(6):2671–2696, 2007.
- [38] J. Lu. *An a Posteriori Error Control Framework for Adaptive Precision Optimization Using the Discontinuous Galerkin Finite Element Method*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge, MA, 2005.
- [39] W. W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87(2):247–282, 2000.
- [40] A. Sandu. On the properties of Runge - Kutta discrete adjoints. In *International Conference on Computational Science (4)*, pages 550–557, 2006.
- [41] A. Walther. Automatic differentiation of explicit Runge-Kutta methods for optimal control. *Computational Optimization and Applications*, 36(1):83–108, 2007.
- [42] T. Oliver and D. L. Darmofal. Analysis of dual consistency for discontinuous Galerkin

- discretizations of source terms. *SIAM Journal on Numerical Analysis*, 47(5):3507–3525, 2009.
- [43] E. Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, 2002.
- [44] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research. Springer-Verlag, 2nd edition, 2006.
- [45] E. Haber and U. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17(6):1847, 2001.
- [46] R. Herzog and E. Sachs. Preconditioned conjugate gradient method for optimal control problems with control and state constraints. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2291–2317, 2010.
- [47] A. Zaslavski and A. Griewank. Projected hessians for preconditioning in one-step one-shot design optimization. In Panos Pardalos, G. Pillo, and M. Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 151–171. Springer US, 2006.
- [48] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.
- [49] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations: Stiff and Differential-Algebraic Problems*, volume II of *Computational Mathematics*. Springer-Verlag, 1994.
- [50] A. Sandu. On consistency properties of discrete adjoint linear multistep methods. Technical Report TR-07-40, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2007.
- [51] W. Bangerth, A. Joshi, and E. M. Sevick-Muraca. Inverse biomedical imaging using separately adapted meshes for parameters and forward model variables. In *Proceedings of the IEEE International Symposium on Biomedical Imaging, Arlington, VA, 2007*, pages 1368–1371. IEEE, 2007.
- [52] P. Eberhard and C. Bischof. Automatic differentiation of numerical integration algorithms. *Mathematics of Computation*, 68(226):717–731, 1999.
- [53] M. Alexe and A. Sandu. On the discrete adjoints of adaptive time stepping algorithms. *J. Comput. Appl. Math.*, 233(4):1005–1020, 2009.
- [54] L. Wang and D. J. Mavriplis. Adjoint based h - p adaptive discontinuous Galerkin methods for aerospace applications. In *AIAA Aerospace Sciences Meeting*, number AIAA Paper 2009-0952, 2009.
- [55] R. Serban, S. Li, and L. R. Petzold. Adaptive algorithms for optimal control of time-dependent partial differential-algebraic equation systems. *Int. J. Numer. Meth. Eng.*, (57):1457–1469, 2003.
- [56] R. M. Lewis and S. G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.*, 26:1811–1837, June 2005.
- [57] N. A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Rev.*, 42(2):247–264, 2000.
- [58] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Verlag, 2007.
- [59] J. Palaniappan, R. B. Haber, and R. L. Jerrard. A spacetime discontinuous galerkin method for scalar conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 193(33-35):3607–3631, 2004.
- [60] J. J. Sudirham, J. J. W. van der Vegt, and R. M. J. van Damme. Space-time discontinuous galerkin method for advection-diffusion problems on time-dependent domains. *Appl. Numer. Math.*, 56(12):1491–1518, 2006.
- [61] M. B. Giles, M. G. Larson, J. M. Levenstam, and E. Suli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. Technical Report 97/06, Oxford University Computing Lab, Numerical Analysis Group, Oxford, UK, 1997.
- [62] K. Harriman, D. Gavaghan, and E. Süli. The importance of adjoint consistency in the

- approximation of linear functionals using the discontinuous galerkin finite element method. Technical Report 04/18, Oxford University Computing Lab, Numerical Analysis Group, Oxford, UK, 2004.
- [63] S. Gottlieb and L-A. J. Gottlieb. Strong stability preserving properties of Runge–Kutta time discretization methods for linear constant coefficient operators. *J. Sci. Comput.*, 18(1):83–109, 2003.
- [64] E. Haber, S Heldmann, and U. Ascher. Adaptive finite volume method for distributed non-smooth parameter identification. *Inverse Problems*, 23(4):1659, 2007.
- [65] R. J. Leveque. *Finite volume methods for hyperbolic methods*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, UK, 2002.
- [66] R. E. Harris and Z.J. Wang. High-order adaptive quadrature-free spectral volume method on unstructured grids. *Computers and Fluids*, 38(10):2006–2025, 2009.
- [67] Z. J. Wang. Spectral (finite) volume method for conservation laws on unstructured grids. basic formulation: Basic formulation. *Journal of Computational Physics*, 178(1):210–251, 2002.
- [68] Z. J. Wang and Y. Liu. Spectral (finite) volume method for conservation laws on unstructured grids: II. extension to two-dimensional scalar equation. *Journal of Computational Physics*, 179(2):665–697, 2002.
- [69] Z. Liu and A. Sandu. On the properties of discrete adjoints of numerical methods for the advection equation. *International Journal for Numerical Methods in Fluids*, 56(7):769–803, 2007.
- [70] Y. Liu, C. W. Shu, E. Tadmor, and M. Zhang. Non-oscillatory hierarchical reconstruction for central and finite volume schemes. *Communications in Computational Physics*, 2(5):933–963, 2007.
- [71] R. Hartmann. Error estimation and adjoint based refinement for an adjoint consistent dg discretisation of the compressible euler equations. *Int. J. Comput. Sci. Math.*, 1(2-4):207–220, 2007.
- [72] R. Hartmann and P. Houston. An optimal order interior penalty discontinuous galerkin discretization of the compressible navier-stokes equations. *Journal of Computational Physics*, 227(22):9670–9685, 2008.
- [73] A. Sandu and P. Miehe. Forward, tangent linear, and adjoint Runge-Kutta methods in KPP–2.2 for efficient chemical kinetic simulations. Technical Report TR-06-17, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2006.
- [74] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, May 2001.
- [75] W. Bangerth, R. Hartmann, and G. Kanschat. *deal.II Differential Equations Analysis Library, Technical Reference*. <http://www.dealii.org>.
- [76] libLBFGS: a library of limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), 2009.
- [77] A. Sandu and L. Zhang. Discrete second order adjoints in atmospheric chemical transport modeling. *J. Comput. Phys.*, 227(12):5949–5983, 2008.
- [78] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge–Kutta schemes. *Math. Comput.*, 67(221):73–85, 1998.
- [79] Z. Wang, I. M. Navon, F. X. Le Dimet, and X. Zou. The second order adjoint analysis: Theory and applications. *Meteorology and Atmospheric Physics*, 50:3–20, 1992. 10.1007/BF01025501.