BIT manuscript No.
(will be inserted by the editor)

A posteriori error estimates for explicit time integration methods

J. B. Collins, Don Estep, Simon Tavener

the date of receipt and acceptance should be inserted later

Abstract In this work we consider a posteriori error analysis of approximations of ordinary differential equations obtained via an explicit finite difference method. Two classes of finite difference methods are reformulated as finite element methods to allow for this analysis. The error is separated into various contributions, each corresponding to a different type of approximation used in the method. Examination of these contributions are used to determine the optimal method of adaptation to reduce the error. Also, a modified adjoint is formulated corresponding to the explicit method and is used to determine when the method is numerically stable for a given quantity of interest.

Keywords a

posteriori error estimate, explicit time integration methods

Mathematics Subject Classification (2000) MSC 34 · MSC 65

1 Introduction

We develop and test a posteriori error estimates for a wide class of explicit time integration schemes for an ordinary differential equation: Compute $y \in$

Department of Statistics, Colorado State University, Fort Collins, CO 80523 (estep@stat.colostate.edu). D. Estep's work is supported in part by the Defense Threat Reduction Agency (HDTRA1-09-1-0036), Department of Energy (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909, DE-SC0001724, DE-SC0005304, INL00120133), Idaho National Laboratory (00069249, 00115474), Lawrence Livermore National Laboratory (B584647, B590495), National Science Foundation (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559, DMS-1065046, DMS-1016268, DMS-FRG-1065046), National Institutes of Health (#R01GM096192).

Address(es) of author(s) should be given

$C^1([0,T];\mathbb{R}^d)$ satisfying,

$$\begin{cases} \dot{y}(t) = f(y(t),t), & 0 < t \le T, \\ y(0) = y_0, \end{cases} \tag{1.1}$$

where $\dot{y}(t) = \frac{d}{dt}y(t)$ and $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is Lipschitz continuous. A posteriori error analysis is a computational approach to error estimation that has been widely employed for finite elements [1,2,4,9,13]. The goal is to produce *accurate* estimates. The approach to a posteriori error analysis adopted here uses variational analysis, adjoint problems, and computable residuals to produce an error estimate that quantifies the relative contributions of all sources of discretization error as well as the accumulation, propagation, and cancellation of their effects.

Prior work [4,5,11,13,15] on a posteriori error estimates for evolution problems has treated implicit methods, which eases the definition of an adjoint operator among other things. We are particularly interested in computing estimates that quantify the effects on accuracy and stability that result from using explicit discretization.

The variational analysis is facilitated by adopting a finite element formulation of finite difference methods. Indeed, there are many ways to do this [3, 12,14]. One of the purposes of this manuscript is to construct a finite element description of popular explicit time integration schemes that allows for a posteriori error estimation. The estimates we derive can be applied directly in a finite difference code without referring to the finite element description.

The finite element description makes it easy to distinguish two components of discretization: (1) approximation of the solution space by a finite dimensional piecewise polynomial space and (2) approximations of the differential operator acting on the spaces. It is important to distinguish these two components because, in general, they introduce numerical error in different ways and their effects accumulate and cancel in different ways. To treat explicit methods, we introduce special operators in the formulation of the numerical method, and then quantify the effects of these operators on the numerical error.

One important issue, however, is the fact that there is not a unique way to define an adjoint for a nonlinear problem. In particular, the differences between the stability properties of implicit and explicit scheme suggests that an a posteriori error analysis of an explicit scheme might require the definition of a different adjoint problem. Differences in adjoint operators has proved crucial to the analysis of such discretization "crimes" as operator splitting [7]. Below, we carry out an analysis using a standard approach to defining an adjoint based on linearization of the perturbation equation. This adjoint problem is defined as: Find $\varphi \in H^1([0,T];\mathbb{R}^d)$ such that,

$$\begin{cases} \langle v, -\dot{\varphi} - \bar{A}^*\varphi\rangle_{[0,T]} = \langle v, \psi\rangle_{[0,T]}, & \forall\, v \in L^2([0,T];\mathbb{R}^d), \\ \varphi(T) = \psi_T. \end{cases} \tag{1.2}$$

where

$$\bar{A} = \int_0^1 f'(sy + (1-s)Y, t)ds.$$

We also present an analysis that uses adjoints to both the original differential equation and the numerical discretization.

1.1 Two stage construction of numerical solutions

We divide the construction of the approximation into two stages. In the first stage, we approximate the solution space by a space of piecewise polynomial functions using a finite element discretization of the variational formulation of (1.1). This reads: Find $y \in H^1([0,T])$ such that,

$$\begin{cases} \mathcal{N}(y,v) := \langle \dot{y} - f(y,t), v \rangle_{[0,T]} = 0, & \forall\, v \in L^2([0,T]), \\ y(0) = y_0, \end{cases} \tag{1.3}$$

where

$$\langle g, h \rangle_{[a,b]} = \int_a^b (g(t), h(t))\, dt,$$

and $(\cdot,\cdot)$ denotes the $\mathbb{R}^d$ inner product. The finite element approximation assumes analytic evaluation of all integrals. In the second stage, we introduce various approximations of the integrands and integrals in (1.3). This amounts to approximating the differential operator. The approximations are chosen to produce approximate solutions equivalent to specific finite difference schemes.

2 Approximation of the solution space

We begin by constructing and analyzing the finite element approximation assuming all integrals in the variational formulation are evaluated exactly.

2.1 Finite element discretization

The finite element discretization of (1.1) involves computing an approximate solution in a piecewise polynomial space. To cover a wide spectrum of finite difference schemes, we consider the so-called continuous and discontinuous Galerkin methods [5,12]. The finite element methods produce piecewise polynomial approximations on the domain $[0,T]$ corresponding to a grid,

$$0 = t_0 < t_1 < \ldots < t_{N-1} < t_N = T,$$

with time steps $k_n = t_n - t_{n-1}$ and subintervals $I_n = [t_{n-1}, t_n]$. The space of continuous piecewise polynomials is,

$$\mathcal{C}^q([0,T]) = \{w \in C([0,T]) : w \in \mathcal{P}^q(I_n), 1 \le n \le N\},$$

where $\mathcal{P}^q(I_n)$ is the space of polynomials of degree $\leq q$ valued in $\mathbb{R}^d$ on I_n. The space of discontinuous piecewise polynomials is,

$$\mathcal{D}^q([0,T]) = \{w \in L^2([0,T]) : w \in \mathcal{P}^q(I_n), 1 \leq n \leq N\}.$$

The continuous Galerkin method of order $q+1$, cG(q), is defined interval-by-interval as: Compute $Y \in V = \mathcal{C}^q([0,T])$ with $Y(0) = y_0$ and for $n = 1, \ldots, N$,

$$\begin{cases} \langle \dot{Y} - f(Y,t), v_k \rangle_{I_n} = 0, & \forall\, v_k \in \mathcal{P}^{q-1}(I_n), \\ Y(t_{n-1}^+) = Y(t_{n-1}^-). \end{cases} \tag{2.1}$$

We can combine these N equations in (2.1) to obtain a global formulation: Compute $Y \in \mathcal{C}^q([0,T])$ such that,

$$\begin{cases} \displaystyle\sum_{n=1}^{N} \langle \dot{Y} - f(Y,t), v_k \rangle_{I_n} = 0, & \forall\, v_k \in V' = \mathcal{D}^{q-1}([0,T]), \\ Y(0) = y_0. \end{cases}$$

The interval-by-interval formulation of the discontinuous Galerkin method of order $q+1$, dG(q), is: Compute $Y \in V = \mathcal{D}^q([0,T])$ such that $Y(0^-) = y_0$ and,

$$\langle \dot{Y} - f(Y,t), v_k \rangle_{I_n} + ([Y]_{n-1}, v_k(t_{n-1}^+)) = 0, \; \forall\, v_k \in \mathcal{P}^q(I_n),\; n = 1, \ldots, N, \tag{2.2}$$

with $[Y]_n = Y(t_n^+) - Y(t_n^-)$. We can interpret the jump condition as imposing continuity weakly. The global formulation is: Compute $Y \in V = \mathcal{D}^q([0,T])$ such that,

$$\begin{cases} \displaystyle\sum_{n=1}^{N} \Big(\langle \dot{Y} - f(Y,t), v_k \rangle_{I_n} + \big([Y]_{n-1}, v_k(t_{n-1}^+)\big) \Big) = 0, \; \forall\, v_k \in V' = \mathcal{D}^q([0,T]), \\ Y(0^-) = y_0. \end{cases} \tag{2.3}$$

In both of these methods, we are assuming that all integrals are computed exactly. The discretizations (2.1) and (2.2) yield a (nonlinear) system of equations for the coefficients of the approximation with respect to the chosen basis of $\mathcal{P}^q(I_n)$ in each interval. For linear problems, these approximations agree with some standard finite difference schemes at node values. We say two approximations are **nodally equivalent** if they yield the same approximation values on any given set of nodes $\{t_n\}$ that partition the domain. For linear constant coefficient problems, the dG(0) scheme is nodally equivalent with backward Euler, the dG(1) scheme is nodally equivalent with a subdiagonal Pade scheme, and the cG(1) scheme is nodally equivalent with Crank-Nicolson. The dG and cG approximations are *not* nodally equivalent with any commonly encountered finite difference scheme for nonlinear problems in general.

2.2 A priori convergence results

A priori analysis [5] of the methods shows that the dG(q) scheme is order $q+1$ at every point in time while it enjoys a so-called $2q+1$ "superconvergence" in the approximation value at time nodes t_n under certain conditions. The nominal order of $q+1$ is optimal in the finite element sense, which means that is the accuracy expected in a piecewise polynomial approximation of the solution. The extra accuracy obtained at time nodes agrees with the expected accuracy of the nodally equivalent finite difference scheme in the cases there is such a scheme. Likewise, the cG(q) scheme is order $q+1$ globally with superconvergence $2q$ in the approximation values at time nodes.

2.3 A posteriori error analysis

Theorem 1 (General Error Representation Formula) *If $Y(t)$ is an approximation of* (1.3) *obtained via the cG(q) method, then the error in the quantity of interest defined by ψ and ψ_T is given by,*

$$\langle e, \psi \rangle_{[0,T]} + (e(T), \psi_T) = \sum_{n=1}^{N} \langle \mathcal{R}(Y), \varphi - \pi_k \varphi \rangle_{I_n}. \tag{2.4}$$

where φ solves the adjoint problem (1.2)*, π_k is a projection onto V', and $\mathcal{R}$ is evaluated in the interior of each interval.*

If instead, $Y(t)$ is a dG(q) approximation, then

$$\begin{aligned}\langle e, \psi \rangle_{[0,T]} &+ (e(T), \psi_T) \\ &= \sum_{n=1}^{N} \Big(\langle \mathcal{R}(Y), \varphi - \pi_k \varphi \rangle_{I_n} - \big([Y]_{n-1}, \varphi(t_{n-1}) - \pi_k \varphi(t_{n-1}^{+}) \big) \Big). \end{aligned} \tag{2.5}$$

We provide the proof of this result for reference below.

Proof We prove for the first case, where Y is a discontinuous approximation. Since $e \in L^2([0,T]; \mathbb{R}^d)$, and can impose e as a legitimate test function in (1.2) and use integration by parts and the facts $e(0) = 0$ and $\varphi(T) = \psi_T$,

$$\begin{aligned}
\langle e,\psi\rangle_{[0,T]} &= \sum_{n=1}^{N} \langle e, -\dot\varphi - \bar A^*\varphi\rangle_{I_n} \\
&= \langle \dot y - \bar f(y,t), \varphi\rangle_{[0,T]} + (y_0, \varphi(0)) - (y(T), \psi_T) \\
&\quad - \sum_{n=1}^{N} \Big(\langle \dot Y - f(Y,t), \varphi\rangle_{I_n} + ([Y]_{n-1}, \varphi(t_{n-1}))\Big) - (y_0, \varphi(0)) + (Y(T^-), \psi_T) \\
&= \sum_{n=1}^{N} \Big(\langle \dot e - \bar A e, \varphi\rangle_{I_n} - ([Y]_{n-1}, \varphi(t_{n-1}))\Big) - (e(T), \psi_T) \\
&= \sum_{n=1}^{N} \Big(\langle \mathcal{R}(Y), \varphi\rangle_{I_n} - ([Y]_{n-1}, \varphi(t_{n-1}))\Big) - (e(T), \psi_T), \qquad (2.6)
\end{aligned}$$

Finally, using the Galerkin orthogonality condition in (2.3), we obtain the error representation formula.

The proof for the continuous Galerkin approximation is almost identical modulo the fact that continuity removes the jump terms.

2.4 Illustrative example

To illustrate the effects of subsequent stages of discretization, we present results for a very simple linear problem with a periodic solution. We present more interesting examples in Section 4.3.

The problem is:

$$\dot y(t) = \begin{bmatrix} 0 & e^{\cdot 2t} \\ -e^{\cdot 2t} & 0 \end{bmatrix} y(t), \qquad t \in [0,T], \qquad y(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \qquad (2.7)$$

The quantity of interest is defined by $\psi = \mathbf{0}$, $\psi_T = \mathbf{1}$. In Fig. 2.1, we plot the error estimate versus final time for the cG(1) method. The adjoint problem is solved numerically using the higher-order cG(2) method. The resulting error estimates are very accurate. We see that there is an overall increasing exponential trend in the errors with increasing time, yet there is also substantial variation in the error as accumulation and cancellation of errors occurs.

3 Approximation of the differential operator using quadrature

Next, we consider finite element methods that involve various approximations of the differential operator. These approximations are chosen to yield finite element approximations that are nodally equivalent to specific finite difference schemes. Showing two methods are nodally equivalent is useful for analysis of finite difference methods, as this provides a path to derive a posteriori error estimates using variational analysis and adjoint problems. We also note that

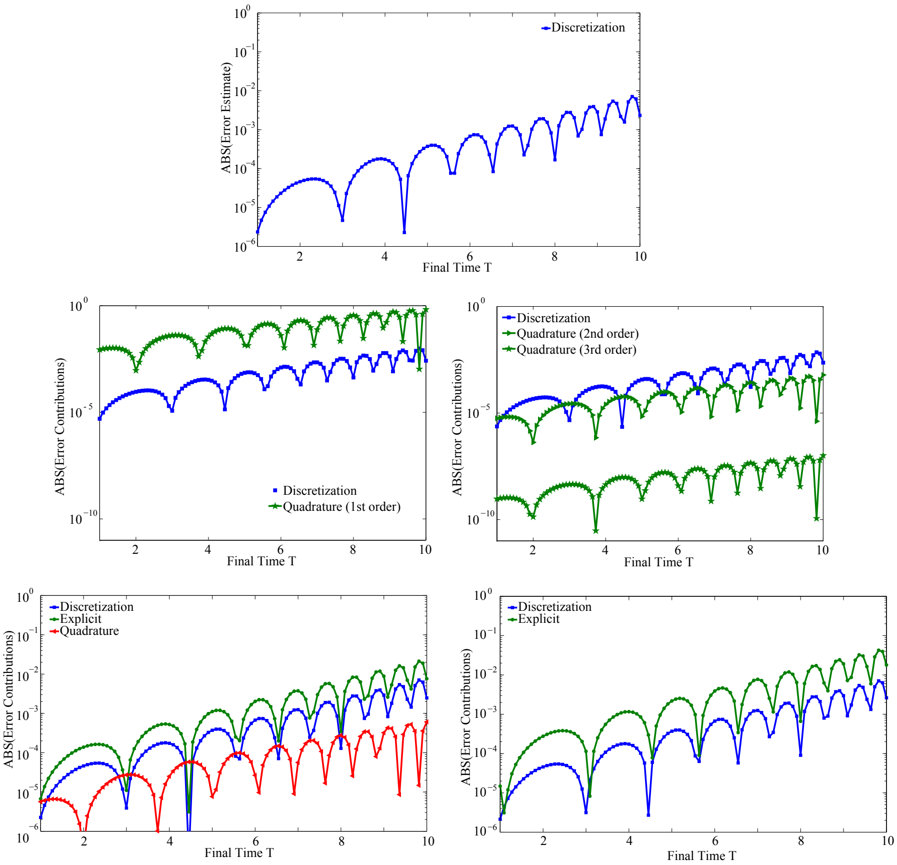

Fig. 2.1 Upper row: The a posteriori error estimate (2.4) for the cG(1) approximation with exact quadrature for (2.7) versus final time T. Middle row: Left: Absolute error contributions versus final time using the cG(1) scheme with first order quadrature; Right: Absolute error contributions versus final time for the cG(1) method using second and third order quadratures. Lower row: Left: Absolute error contributions using the Taylor series error representation formula for the explicit trapezoid method; Right: Absolute error contributions using the extrapolation error representation formula for the second-order Adams-Bashforth method.

the finite element approximation is a function defined for all times, so the finite element values between nodes might be considered a “natural” way to define values of a finite difference approximation between nodes.

We begin with quadrature applied to the integrals in the weak formulation, mainly the integral involving f. There are obvious practical advantages to writing codes that avoid exact integration of f. Also, specific choices of quadrature yield formulas that are nodally equivalent to specific finite difference schemes.

The cG(q) method with quadrature is written as: Find $Y \in \mathcal{C}^q(I_n)$ such that $Y(0) = y_0$ and for $n = 1, \cdots, N$,

$$\langle \dot{Y}, v_k \rangle_{I_n} - \langle f(Y, t), v_k \rangle_{I_n, k_n} = 0, \quad \forall\, v_k \in \mathcal{P}^{q-1}(I_n), \tag{3.1}$$

where the nonlinear term uses the approximate inner product

$$\langle g, h \rangle_{I_n, k_n} = \sum_{i=1}^{L_n} g(s_{i,n}) h(s_{i,n}) w_{i,n},$$

defined by nodes $s_{i,n}$ and weights $w_{i,n}$ associated with I_n. A similar change is made to the discontinuous Galerkin method to implement quadrature.

As a simple example, we consider the cG(1) method with the trapezoid rule quadrature, defined by using

$$\langle g, h \rangle_{I_n, k_n} = \frac{k_n}{2}(g(t_{n-1})h(t_{n-1}) + g(t_n)h(t_n)).$$

in (3.1) to obtain,

$$Y(t_n) = Y(t_{n-1}) + \frac{k_n}{2}(f(Y(t_{n-1}), t_{n-1}) + f(Y(t_n), t_n)).$$

So, the cG(1) method with trapezoid rule quadrature is nodally equivalent to the implicit trapezoid method. Similarly, the cG(1) method with the midpoint rule quadrature is nodally equivalent to the implicit midpoint method.

3.1 A priori convergence results

Following standard finite element convergence analysis, using a quadrature formula of sufficient accuracy preserves the nominal optimal order of convergence of the method. Superconvergence results are more difficult to show. Perhaps the easiest approach is to find a nodally equivalent finite difference scheme and use the corresponding convergence results.

3.2 A posteriori error analysis

We note that the quadrature approximation (3.1) has the property that the approximation error can be made as small as desired, e.g. by increasing the order or number of internal nodes in a composite rule. This suggests the standard approaching of using the previous adjoint problem (1.2) for a posteriori error estimation. This yields,

Theorem 2 (Quadrature Error Representation Formula) *If $Y(t)$ is an approximation of* (1.3) *obtained via the cG(q) method with a quadrature defined by the inner product $\langle\cdot,\cdot\rangle_{I_n,k_n}$, then,*

$$\langle e,\psi\rangle_{[0,T]} + (e(T),\psi_T) = \sum_{n=1}^{N}\Bigg(\underbrace{\langle \mathcal{R}(Y), \varphi - \pi_k\varphi\rangle_{I_n}}_{Discretization\ Contribution} + \underbrace{\langle f(Y,t),\pi_k\varphi\rangle_{I_n} - \langle f(Y,t),\pi_k\varphi\rangle_{I_n,k_n}}_{Quadrature\ Contribution}\Bigg). \tag{3.2}$$

where φ and π_k are defined as above.

If instead, $Y(t)$ is a dG(q) approximation, then,

$$\langle e,\psi\rangle_{[0,T]} + (e(T),\psi_T) = \sum_{n=1}^{N}\Bigg(\underbrace{\langle \mathcal{R}(Y), \varphi - \pi_k\varphi\rangle_{I_n} - ([Y]_{n-1}, \varphi(t_{n-1}) - \pi_k\varphi(t_{n-1}^{+}))}_{Discretization\ Contribution} + \underbrace{\langle f(Y,t),\pi_k\varphi\rangle_{I_n} - \langle f(Y,t),\pi_k\varphi\rangle_{I_n,k_n}}_{Quadrature\ Contribution}\Bigg),$$

Proof As before we prove the first case where $Z(t)$ is a discontinuous Galerkin approximation. The proof is identical up to the point that Galerkin orthogonality is used, therefore using (2.6) we have,

$$\langle e,\psi\rangle_{[0,T]} + (e(T),\psi_T) = \sum_{n=1}^{N}\Big(\langle \mathcal{R},\varphi\rangle_{I_n} - \big([Y]_{n-1},\varphi(t_{n-1})\big)\Big).$$

We next add zero in two ways, firstly by using Galerkin orthogonality, and secondly by adding and subtracting a term,

$$\begin{aligned}\langle e,\psi\rangle_{[0,T]} + (e(T),\psi_T) = \sum_{n=1}^{N}\Big(&\langle f(Y,t) - \dot{Y},\varphi\rangle_{I_n} - ([Y]_{n-1},\varphi(t_{n-1}))\\ &+ \langle \dot{Y},\pi_k\varphi\rangle_{I_n} - \langle f(Y,t),\pi_k\varphi\rangle_{I_n,k_n}\\ &+ ([Y]_{n-1},\pi_k\varphi(t_{n-1}^{+})) \pm \langle f(Y,t),\pi_k\varphi\rangle_{I_n}\Big).\end{aligned}$$

By rearranging the terms we obtain the error representation formula for an approximation with quadrature.

As before, it is a simple matter to prove the error representation formula for the continuous Galerkin approximation, as it is a simplification.

In these estimates, we have distinguished contributions from discretization of the solution space from the effects of quadrature, which approximates the differential operator. This is important for two reasons. First, we see that the discretization contribution involves Galerkin orthogonality, leading to an adjoint weight of $\varphi - \pi_k\varphi$, while the quadrature term only involves $\pi_k\varphi$. The consequence is that these two sources of discretization error accumulate, propagate, and cancel in different ways in general. The second important issue is that we can identify situations in which the discretization of the solution space is a more or less significant contribution to the total error as compared to the approximation of the differential operator. In the former case, the solution is to choose smaller time steps or use a higher order piecewise polynomial space. In the latter case, we can decrease the error by decreasing the time step. But we can also simply use a higher order quadrature, which is generally a more efficient approach. We illustrate this dichotomy in the examples below. We also note that in the case that a reasonable quadrature formula is used, the discretization contributions of the finite element approximations with and without quadrature are close.

3.3 Illustrative example

Returning to (2.7), we solve with quadratures of three different orders applied to cG(1) to show the effect of the quadrature on the error. To explore relative differences, Fig. 2.1 shows the contributions to the error defined from (3.2) as,

$$\underbrace{\left|\sum_{n=1}^{N}\langle \mathcal{R}, \varphi - \pi_k\varphi\rangle_{I_n}\right|}_{\text{Discretization Contribution}} + \underbrace{\left|\sum_{n=1}^{N}\Big(\langle f(Y,t), \pi_k\varphi\rangle_{I_n} - \langle f(Y,t), \pi_k\varphi\rangle_{I_n,k_n}\Big)\right|}_{\text{Quadrature Contribution}}$$

We note that there may be significant cancellation between the “discretization” and “quadrature” contributions in the estimate, so the absolute values of these quantities are examined only to consider their relative size. In this case, we find that the discretization contributions are roughly the same regardless of the quadrature. However, the first order quadrature contribution dominates the discretization, while for third order quadrature, the discretization contribution dominates. The error contributions are roughly the same when second order quadrature is used. This suggests the standard rule of thumb for finite element quadratures, which is to use a quadrature order that is the same as the finite element method.

However when the quadrature error contribution is large, then using a higher order quadrature to evaluate the integrals is generally less computationally expensive than increasing the order of approximation or using a smaller time step for the finite element discretization. Hence in some circumstances, using a higher order quadrature formula is an efficient way to increase accuracy.

4 Explicit Approximations

We next consider explicit discretizations, i.e., methods that do not require solution of a (non)linear system of equations for the approximation on each time step. We are particularly interested in the most common explicit finite difference methods, e.g., forward Euler, Explicit Trapezoid, Runge-Kutta 4, and the family of Adams-Bashforth methods.

In the finite element framework, explicit discretizations result from using *extrapolation* of some approximation of f in the variational integrals. There are many possible ways to produce an extrapolating approximation. However, we discuss two approaches that provide a natural avenue for a posteriori error analysis.

The first approach uses a local series expansion, and this yields a family of one-step explicit methods such as Runge-Kutta. Often these expansions are approximations of a truncated Taylor series, though they can be more general. Equivalence with explicit methods can be obtained by applying the finite element discretizations including quadrature, e.g. (3.1), to the nominal modified problem of the form: Find $\tilde{y} \in H^1([0,T];\mathbb{R}^d)$ such that,

$$\begin{cases} \displaystyle\sum_{n=1}^{N} \langle \dot{\tilde{y}} - \sum_{i=1}^{L} \alpha_i f(P_n^i \tilde{y}, t), v\rangle_{I_n}, \qquad \forall\, v \in L^2([0,T];\mathbb{R}^d), \\ \tilde{y}(0) = y_0, \end{cases} \tag{4.1}$$

where $\sum_{i=1}^{L} \alpha_i = 1$ and the operators P_n^i are described below.

The second approach replaces f by an extrapolation of a polynomial interpolant computed from previous time nodes. This approach yields a family of methods that include the multi-step explicit methods such as Adams-Bashforth. The nominal modified equation for this type of approximation is: Find $\tilde{y} \in H^1([0,T];\mathbb{R}^d)$ such that,

$$\begin{cases} \displaystyle\sum_{n=1}^{\ell-1} \langle \dot{\tilde{y}} - f(\tilde{y}, t), v\rangle_{I_n} + \sum_{n=\ell}^{N} \langle \dot{\tilde{y}} - Q_n^\ell f(\tilde{y}, t), v\rangle_{I_n} = 0, \qquad \forall\, v \in L^2([0,T];\mathbb{R}^d), \\ \tilde{y}(0) = y_0, \end{cases} \tag{4.2}$$

where the operators Q_n^ℓ are described below.

4.1 Taylor Series approximation

As discussed, this approach uses an approximate truncated Taylor Series expansion around each time node. We first discuss the construction of the approximation, then use the approximation to define a solution method. We express the approximation as the result of an operator P applied to piecewise polynomials. The operator P is the composition of two operators $P = TS$.

Around a given node t_{n-1}, the operator P maps a BV function to a polynomial defined on $[t_{n-2}, t_n]$. The first operator S projects a function with limited regularity into a space of functions with sufficient regularity for a truncated Taylor series to be defined. This is needed since we apply P to finite element functions that have discontinuities in value and/or derivative at time nodes. Given n and $v \in BV$ on an interval containing time nodes $\{t_i,\ i \in \mathcal{I}_n\}$, we define S as the polynomial that interpolates v with values $\{v(t_i^-), i \in \mathcal{I}_n\}$. Typically, $\mathcal{I}_n$ includes $n, n-1, n-2, \ldots$ for the number of nodes equal to the order of the Taylor series expansion. Therefore S performs a spline-type interpolation on the left-hand side of the nodes.

The second operator T maps a sufficiently smooth function to a polynomial of degree k valued on $[t_{n-2}, t_n]$ that is, roughly speaking, an approximation of a Taylor polynomial of order k at t_{n-1}. Given a function $v \in C^k([t_{n-2}, t_n]; \mathbb{R}^d)$, we define

$$Tv = v(t_{n-1}) + \sum_{i=1}^{k} \frac{d^{(i-1)}}{dt^{(i-1)}} f(v(t_{n-1}), t_{n-1}) \frac{(t - t_{n-1})^i}{i!},$$

where the time derivatives of f are computed using the chain rule. This expression is motivated by considering v to be an approximation of the solution y of (1.1), for which

$$\frac{dy}{dt} = f(y), \quad \frac{d^2y}{dt^2} = \frac{d}{dt} f(y) = \frac{df}{dy}\frac{dy}{dt}, \quad \frac{d^3y}{dt^3} = \frac{d^2}{dt^2} f(y), \cdots$$

Since $P = TS$ is defined in a neighborhood of each t_{n-1} and used in $[t_{n-1}, t_n]$, it is convenient to denote its restriction to $[t_{n-1}, t_n]$ by P_n. We then write the first explicit cG(q) method as: Find $Y \in \mathcal{C}^q(I_n)$ such that,

$$\begin{cases} \langle \dot{Y}, v_k \rangle_{I_n} = \langle f(P_n Y, t), v_k \rangle_{I_n, k_n}, & \forall\, v_k \in \mathcal{P}^{q-1}(I_n),\ n = 1, \ldots, N, \\ Y(0) = y_0. \end{cases} \tag{4.3}$$

Note that we have once again used quadrature to evaluate the nonlinear term. The definition of the dG(q) method is analogous.

We note that these cG and dG approximations can be obtained by applying the finite element discretizations including quadrature, e.g. (3.1), to the nominal modified problem (4.1). Introduction of (4.1) also turns out to be convenient for error analysis.

4.1.1 Examples

As a simple example, we consider the Taylor series of order zero, so

$$P_n Y(t) = Y(t_{n-1}^-),$$

The weak form of the modified problem (4.1) becomes

$$\sum_{n=1}^{N} \langle \dot{Y} - f(Y(t_{n-1}^-), t_{n-1}^-), v\rangle_{I_n} = 0, \qquad \forall\, v \in L^2(\Omega).$$

Using dG(0) with the left hand quadrature rule, we obtain

$$Y(t_n^-) = Y(t_{n-1}^-) + k_n f(Y(t_{n-1}^-), t_{n-1}^-),$$

which is the update formula for the forward Euler method.

In the next examples, we consider the Taylor series of order one,

$$P_n Y = Y(t_{n-1}^-) + f(Y(t_{n-1}^-), t_{n-1}^-)(t - t_{n-1}^-).$$

The cG(1) approximation is determined by,

$$Y(t_n) = Y(t_{n-1}) + \langle f(Y(t_{n-1}) + f(Y(t_{n-1}), t_{n-1}^-)(t - t_{n-1}^-), 1\rangle_{I_n,k_n}.$$

By varying the quadrature, we can obtain different nodally equivalent finite difference schemes. For example, the midpoint quadrature rule yields the explicit midpoint method,

$$\begin{aligned} \hat{Y}_n &= Y(t_{n-1}) + \frac{k_n}{2} f(Y(t_{n-1}), t_{n-1}^-) \\ Y(t_n) &= Y(t_{n-1}) + k_n f(\hat{Y}_n, t_{n-1/2}), \end{aligned}$$

with $t_{n-1/2} = t_n - \frac{k_n}{2}$. If we use the trapezoid rule for quadrature, we obtain the explicit trapezoid or RK2 method,

$$\begin{aligned} \hat{Y}_n &= Y(t_{n-1}) + k_n f(Y(t_{n-1}), t_{n-1}^-) \\ Y(t_n) &= Y(t_{n-1}) + \frac{k_n}{2}(f(Y(t_{n-1}), t_{n-1}^-) + f(\hat{Y}_n, t_n^-)). \end{aligned}$$

We can derive new methods for a given P_n by changing the quadrature. This can increase accuracy on certain problems considerably at a mild increase in computational cost. For instance, if we replace the trapezoid rule in the RK2 method with Simpson's rule,

$$\langle g, h\rangle_{I_n,k_n} = \frac{k_n}{6}(g(t_{n-1})h(t_{n-1}) + 4g(t_{n-1/2})h(t_{n-1/2}) + g(t_n)h(t_n)),$$

this gives a method we call RK2/4,

$$\begin{aligned} Y_1 = Y(t_{n-1}), \quad Y_2 = Y_1 + \frac{k_n}{2} f(Y_1, t_{n-1}), \quad Y_3 = Y_1 + k_n f(Y_1, t_{n-1}) \\ Y(t_n) = Y(t_{n-1}) + \frac{k_n}{6}(f(Y_1, t_{n-1}) + 4f(Y_2, t_{n-1/2}) + f(Y_3, t_n)). \end{aligned}$$

Note that while we use a fourth order quadrature in RK2/4 with the cost of a single additional function evaluation, the method is still second order overall. However, this can still lead to significant improvement in accuracy when the quadrature contribution in the RK2 approximation is dominant, as we illustrate below.

4.1.2 Runge-Kutta 4

The fourth order Runge-Kutta method (RK4) for solving ODEs is well used as a catch-all method. It has fairly high order with relatively few function evaluations, making it a method worth trying before moving on to more complex methods. Therefore, we show how such a method can be written as a nodally equivalent finite element method, so that the above error analysis can be implemented.

As with RK2, RK4 is a one-step method. The update formula is given as,

$$\begin{aligned}
Y^1 &= Y(t_{n-1}), & Y^2 &= Y^1 + \frac{k_n}{2} f(Y^1, t_{n-1}) \\
Y^3 &= Y^1 + \frac{k_n}{2} f(Y^2, t_{n-1/2}), & Y^4 &= Y^1 + k_n f(Y^3, t_{n-1/2})
\end{aligned}$$

$$\begin{aligned}
Y(t_n) = Y(t_{n-1}) + \frac{k_n}{6}\big(&f(Y^1, t_{n-1}) + 2f(Y^2, t_{n-1/2}) \\
&+ 2f(Y^3, t_{n-1/2}) + f(Y^4, t_n)\big)
\end{aligned}$$

Four separate mappings are needed to show equivalence for RK4 with coefficient,

$$\alpha_1 = \frac{1}{6} \qquad \alpha_2 = \frac{2}{6} \qquad \alpha_3 = \frac{2}{6} \qquad \alpha_4 = \frac{1}{6}.$$

Each mapping corresponds to an intermediate value Y^i, and they have the property,

$$\begin{aligned}
(P_n^1 Y)(t_{n-1}) &= Y(t_{n-1}) =: Y^1 \\
(P_n^2 Y)(t_{n-1/2}) &= Y(t_{n-1}) + \frac{k_n}{2} f(Y(t_{n-1}), t_{n-1}) =: Y^2 \\
(P_n^3 Y)(t_{n-1/2}) &= Y(t_{n-1}) + \frac{k_n}{2} f(Y^2, t_{n-1/2}) =: Y^3 \\
(P_n^4 Y)(t_n) &= Y(t_{n-1}) + k_n f(Y^3, t_{n-1/2}) =: Y^4
\end{aligned}$$

We state the definitions of the mappings P_n^i and show that when applied to the exact solution $y(t)$, they can be written as approximations of the full Taylor series,

$$y = \tilde{P}_n y = \sum_{i=0}^{\infty} \frac{1}{i!} y^{(i)}(t_{n-1})(t - t_{n-1})^i.$$

We look at each mapping individually.

The first mapping is defined as,

$$P_n^1 y = y(t_{n-1})$$

and is a simple truncation of the full Taylor series $\tilde{P}_n y$. Defining the second mapping as,

$$P_n^2 y = y(t_{n-1}) + f(y(t_{n-1}), t_{n-1})(t - t_{n-1}),$$

we see that is too is a truncated Taylor series, since $y'(t_{n-1}) = f(y(t_{n-1}), t_{n-1})$.

The third and fourth mapping cannot be written simply as a truncated Taylor series. However, they can be written as a series of approximations to the Taylor series. We use the fact that $y(t) = \tilde{P}_n y$ and $P_n^2 y \approx \tilde{P}_n y$. We also introduce the projection $\hat{P}_n y = y(t_{n-1/2})$. Starting with $\tilde{P}_n y$, we use certain approximations to obtain the definition of P_n^3 and P_n^4.

$$\begin{aligned}\tilde{P}_n y &= y(t_{n-1}) + \int_{t_{n-1}}^{t} f(y(s), s)\, ds \\ &\approx y(t_{n-1}) + \hat{P}_n f(y,t)(t - t_{n-1}) \\ &= y(t_{n-1}) + \hat{P}_n f(\tilde{P}_n y, t)(t - t_{n-1}) \\ &\approx y(t_{n-1}) + \hat{P}_n f(P_n^2 y, t)(t - t_{n-1}).\end{aligned}$$

Therefore we define the third mapping as,

$$P_n^3 y = y(t_{n-1}) + \hat{P}_n f(P_n^2 y, t)(t - t_{n-1}).$$

Through a very similar derivation we obtain the fourth mapping,

$$P_n^4 y = y(t_{n-1}) + \hat{P}_n f(P_n^3 y, t)(t - t_{n-1}).$$

The difference in the derivation being that we use the approximation $P_n^3 y \approx \tilde{P}_n y$.

Finally we must apply certain quadratures to obtain equivalence with RK4. We define the following discrete inner products to represent various quadrature rules.

$$\begin{aligned}\langle g, h\rangle_{I_n,L} &= \text{Left Hand Rule} \\ \langle g, h\rangle_{I_n,M} &= \text{Midpoint Rule} \\ \langle g, h\rangle_{I_n,R} &= \text{Right Hand Rule}\end{aligned}$$

Using these mapping and quadratures, the following finite element method is equivalent to RK4: Find $Y \in \mathcal{C}^q(I_n)$ such that,

$$\begin{cases} \langle \dot{Y}, v_k\rangle_{I_n} = \frac{1}{6}\big[\langle f(P_n^1 Y, t), v_k\rangle_{I_n,L} + 2\langle f(P_n^2 Y, t) + f(P_n^3 Y, t), v_k\rangle_{I_n,M} \\ \qquad +\langle f(P_n^4 Y, t), v_k\rangle_{I_n,R}\big] \qquad \forall\, v_k \in \mathcal{P}^{q-1}(I_n), \quad n = 1, \ldots, N \\ Y(0) = y_0. \end{cases} \tag{4.4}$$

Finally, we solve (4.4) with cG(3) finite elements as this will yield the expected order of superconvergence which matches the order of accuracy of the finite difference scheme. Nodal equivalence is easily proved, as the piecewise constant function is a test function for cG(3). The additional degrees of freedom give intermediate values of the finite element solution within I_n that are needed for the finite difference scheme. These intermediate values can be determined explicitly from the values of the approximation at t_{n-1} and t_n.

Finally we note that while the finite difference scheme is fourth order, the approximations to the Taylor series and the quadrature used in the finite element formulation are less than fourth order. This suggests that the RK4 scheme achieves its order through a special cancelation of error contributions. This has an impact on the a posteriori error estimate, because each individual contribution to the overall error are not in themselves fourth order, yet their sum yields fourth order accuracy. In particular, this presents a difficulty for adaptivity, as the effect of cancelation cannot be easily accounted for. It is possible to create a method that is fourth order in each term, by using a mapping that is the fourth order Taylor series,

$$P_n Y = Y(t_{n-1}) + \sum_{i=1}^{3} \frac{1}{i!} f^{(i)}(Y(t_{n-1}), t_{n-1})(t - t_{n-1})^i,$$

and a fourth order quadrature such as Simpson's rule. This would make each term fourth order.

4.1.3 A posteriori error analysis

We note that the truncated Taylor series approximation has the property that the approximations become exact in the limit of increasing order. Hence, the sequence of "explicit" modified problems (4.1) nominally approach the true problem in the limit of increasing order. With intuition analogous to the use of quadrature, this suggests using the same adjoint problem for error analysis. The effect of introducing the operator P_n is a modification of the residuals.

Theorem 3 (Taylor Series Error Representation Formula) *If $Y(t)$ is an approximation of* (4.1) *obtained via the cG(q) method with quadrature, then,*

$$\begin{aligned}\langle e, \psi\rangle_{[0,T]} + (e(T), \psi_T) = \sum_{n=1}^{N} \Bigg(&\underbrace{\langle \mathcal{R}_P(Y), \varphi - \pi_k \varphi\rangle_{I_n}}_{\textit{Discretization Contribution}} \\ &+ \underbrace{\langle f(Y,t) - \sum_{i=1}^{L} \alpha_i f(P_n^i Y, t), \varphi\rangle_{I_n}}_{\textit{Explicit Contribution}} \\ &+ \underbrace{\sum_{i=1}^{L} \alpha_i \langle f(P_n^i Y, t), \pi_k \varphi\rangle_{I_n} - \sum_{i=1}^{L} \alpha_i \langle f(P_n^i Y, t), \pi_k \varphi\rangle_{I_n, k_n^i}}_{\textit{Quadrature Contribution}} \Bigg),\end{aligned}$$

where φ and π_k are defined as above, and $\mathcal{R}_P(Z) = \sum_{i=1}^{L} \alpha_i f(P^i Z, t) - \dot{Z}$ is the modified residual.

If instead, $Y(t)$ is a dG(q) approximation, then,

$$
\begin{aligned}
&\langle e, \psi\rangle_{[0,T]} + (e(T), \psi_T) \\
&\quad = \sum_{n=1}^{N} \Bigg(\underbrace{\langle \mathcal{R}_P(Y), \varphi - \pi_k\varphi\rangle_{I_n} - ([Y]_{n-1}, \varphi(t_{n-1}) - \pi_k\varphi(t_{n-1}^{+}))}_{\textit{Discretization Contribution}} \\
&\qquad + \underbrace{\langle f(Y,t) - \sum_{i=1}^{L} \alpha_i f(P_n^i Y, t), \varphi\rangle_{I_n}}_{\textit{Explicit Contribution}} \\
&\qquad + \underbrace{\sum_{i=1}^{L} \alpha_i \langle f(P_n^i Y, t), \pi_k\varphi\rangle_{I_n} - \sum_{i=1}^{L} \alpha_i \langle f(P_n^i Y, t), \pi_k\varphi\rangle_{I_n, k_n^i}}_{\textit{Quadrature Contribution}} \Bigg),
\end{aligned}
$$

Proof As before, we begin by defining the following nonlinear form,

$$
\mathcal{N}_P(Z, v) = \sum_{n=1}^{N} \langle \dot{Z} - \sum_{i=1}^{L} \alpha_i f(P_n^i Z, t), v\rangle_{I_n}, \tag{4.5}
$$

for $Z \in H^1([0,T];\mathbb{R}^d)$ and $v \in L^2([0,T];\mathbb{R}^d)$. We evaluate (1.3) and (4.5) at y and Y respectively and subtract to obtain,

$$
\begin{aligned}
\sum_{n=1}^{N} \langle \mathcal{R}_P(Y), v\rangle_{I_n} &= \mathcal{N}(y, v) - \mathcal{N}_P(Y, v) \\
&= \int_0^1 \mathcal{N}'(sy + (1-s)Y, v; e)\, ds + \mathcal{N}(Y, v) - \mathcal{N}_P(Y, v).
\end{aligned}
$$

This yields a relation between the error and the residual,

$$
\sum_{n=1}^{N} \langle \mathcal{R}_P(Y), v\rangle_{I_n} = \sum_{n=1}^{N} \Bigg(\langle \dot{e} - \bar{A}e, v\rangle_{I_n} - \langle f(Y,t) - \sum_{i=1}^{L} \alpha_i f(P_n Y, t), v\rangle_{I_n} \Bigg).
$$

Following the previous proofs, we use (2.6) and obtain,

$$
\begin{aligned}
\langle e, \psi\rangle_{[0,T]} + (e(T), \psi_T) &= \sum_{n=1}^{N} \Big(\langle \dot{e} - \bar{A}e, \varphi\rangle_{I_n} - \big([Y]_{n-1}, \varphi(t_{n-1})\big) \Big) \\
&= \sum_{n=1}^{N} \Bigg(\langle \mathcal{R}_P(Y), \varphi\rangle_{I_n} - ([Y]_{n-1}, \varphi(t_{n-1})) \\
&\qquad + \langle f(Y,t) - \sum_{i=1}^{L} \alpha_i f(P_n Y, t), \varphi\rangle_{I_n} \Bigg).
\end{aligned}
$$

Next we use Galerkin orthogonality defined in (4.3) to obtain the error representation formula. As before, proving the representation formula for a continuous Galerkin approximation follows a similar argument.

4.1.4 Illustrative example

Returning to (2.7), we have three error contributions; discretization, quadrature, and a new “explicit” contribution. In Fig. 2.1, we plot these contributions for the explicit trapezoid method. We see that the explicit contribution is larger than the discretization error while the quadrature contribution is significantly smaller overall. However, the quadrature contribution is “out of phase” with the other two contributions.

4.2 Polynomial extrapolation

Once again, this approximation involves insertion of an operator. The operator $Q_n^\ell : BV([0,T]) \to \mathcal{P}^{\ell-1}(I_n)$ produces a polynomial $Q_n^\ell f$ that interpolates a given function f at the nodes $t_{n-\ell}, \ldots, t_{n-1}$. Subsequently, $Q_n^\ell f$ is extrapolated to replace f on I_n. For notational purposes we define the operator Q^ℓ to be such that $Q^\ell|_{I_n} = Q_n^\ell$. Properly speaking, Q_n^ℓ is defined only for $[t_{\ell-1}, T]$. In the initial stage $[0, t_{\ell-1}]$, we have to use another process to define the approximation. It is important to use a method that preserves the order of the general method. There are various options, including use of Taylor series approximation or a nested sequence of nodal approximations that build up the appropriate order. The a posteriori error analysis has to be modified to account for the two different discretization stages. To simplify the presentation, we solve the initial stage with the implicit cG($\ell - 1$) method, which has the same order as the corresponding explicit method.

The cG(q) method with polynomial extrapolation is given by: Find $Y \in \mathcal{C}^q(\Omega)$ such that,

$$\begin{cases} \langle \dot{Y}, v_k \rangle_{I_n} - \langle f(Y,t), v_k \rangle_{I_n} = 0 & \forall\, v_k \in \mathcal{P}^{q-1}(I_n),\ n = 1, \ldots, \ell - 1, \\ \langle \dot{Y}, v_k \rangle_{I_n} - \langle Q_n^\ell f(Y,t), v_k \rangle_{I_n} = 0 & \forall\, v_k \in \mathcal{P}^{q-1}(I_n),\ n = \ell, \ldots, N, \\ Y(0) = y_0. \end{cases} \tag{4.6}$$

No quadrature is used since Q_n^ℓ is a polynomial and the integrands can be integrated analytically. As above, we can view (4.6) as applying the cG(q) method to the modified problem (4.2).

4.2.1 Examples

The finite element approximations are nodally equivalent to the Adams-Bashforth multi-step finite difference methods. To see this, we apply the cG(1) finite element method to the modified problem (4.2) with $\ell = 2$. This yields,

$$\begin{aligned} Y(t_n) &= Y(t_{n-1}) + \left\langle f(Y(t_{n-1}), t_{n-1}) \frac{t - t_{n-2}}{k_n} + f(Y(t_{n-2}), t_{n-2}) \frac{t_{n-1} - t}{k_n}, 1 \right\rangle_{I_n} \\ &= Y(t_{n-1}) + \frac{3}{2} k_n f(Y(t_{n-1}), t_{n-1}) - \frac{1}{2} k_n f(Y(t_{n-2}), t_{n-2}), \end{aligned}$$

which is the update formula for the second order Adams-Bashforth method. For a general projection Q_n^ℓ the finite element approximation of (4.2) is nodally equivalent to the ℓth order Adams-Bashforth method.

4.2.2 A posteriori error analysis

We again use the adjoint problem (1.2). However, this choice is potentially more problematic than for the previous methods. First, there is a finite limit to the accuracy of a family of polynomial interpolants of increasing order computed on a fixed sequence of time nodes because of stability limitations of polynomial interpolation. Thus, unlike the Taylor series approach, the polynomial extrapolation approximation does not produce a family of modified problems (4.2) that converge to the true problem. Moreover, we are extrapolating rather than interpolating the polynomial. Hence, there is reasonable concern that the discretized solution operator is associated with a different adjoint operator than the true solution operator. We explore this below in Section 5.

Theorem 4 (Extrapolation Error Representation Formula) *If $Y(t)$ is an approximation of* (4.2) *obtained via the cG(q) method, then,*

$$\begin{aligned}\langle e,\psi\rangle_{[0,T]}+&(e(T),\psi_T)\\ &=\sum_{n=1}^{\ell-1}\Bigg(\underbrace{\langle\mathcal{R}(Y),\varphi-\pi_k\varphi\rangle_{I_n}}_{\textit{Initial Contribution}}+\sum_{n=\ell}^{N}\underbrace{\langle\mathcal{R}_Q^\ell(Y),\varphi-\pi_k\varphi\rangle_{I_n}}_{\textit{Discretization Contribution}}\\ &\qquad\qquad+\underbrace{\langle f(Y,t)-Q_n^\ell f(Y,t),\varphi\rangle_{I_n}}_{\textit{Explicit Contribution}}\Bigg)\end{aligned}$$

where φ and π_k are defined as above, and $\mathcal{R}_Q^\ell(Z)=Q^\ell f(Z,t)-\dot{Z}$ is the modified residual.

If instead, $Y(t)$ is a dG(q) approximation, then,

$$\begin{aligned}\langle e,\psi\rangle_{[0,T]}+(e(T),\psi_T)=&\sum_{n=1}^{\ell-1}\underbrace{\langle\mathcal{R}(Y),\varphi-\pi_k\varphi\rangle_{I_n}}_{\textit{Initial Contribution}}\\ &+\sum_{n=\ell}^{N}\Bigg(\underbrace{\langle\mathcal{R}_Q^\ell(Y),\varphi-\pi_k\varphi\rangle_{I_n}-([Y]_{n-1},\varphi(t_{n-1})-\pi_k\varphi(t_{n-1}^+))}_{\textit{Discretization Contribution}}\\ &\qquad+\underbrace{\langle f(Y,t)-Q_n^\ell f(Y,t),\varphi\rangle_{I_n}}_{\textit{Explicit Contribution}}\Bigg),\end{aligned}$$

Note that the initial contribution is the same as provided by Theorem 1.

Proof The proof follows the argument used for Theorem 3 except for the treatment of the initial stage. There are at least two ways to incorporate the error in the initial stage into the error representation formula. In the results above, we divide the time step contributions into those from the initial stage and those from the remaining stage. We then apply the appropriate form of Galerkin orthogonality to each piece independently.

Another approach involves applying Theorem 4 on the second stage $[t_{\ell-1}, T]$ using $t_{\ell-1}$ as an "initial" time. The error representation formula for a cG(q) solution is given by,

$$\begin{aligned}\langle e, \psi\rangle_{[t_{\ell-1},T]} + (e(T), \psi_T) = \sum_{n=\ell}^{N} \Big(&\langle \mathcal{R}_Q^{\ell}(Y), \varphi - \pi_k \varphi\rangle_{I_n} \\ &+ \langle f(Y,t) - Q_n^{\ell} f(Y,t)\rangle_{I_n} \Big) \\ &+ (e(t_{\ell-1}), \varphi(t_{\ell-1})),\end{aligned}$$

where the last term $(e(t_{\ell-1}), \varphi(t_{\ell-1}))$ represents "inherited" error from the initial stage. We then define the appropriate adjoint problem for the method on the initial stage using data $\varphi(t_{\ell-1})$ to define the correct quantity of interest.

4.2.3 Illustrative example

Returning to (2.7), we show two error contributions, i.e., the discretization and "explicit" contributions in Fig. 2.1. As before, the discretization contribution is almost identical to the discretization contributions for previous methods. The explicit error contribution clearly dominates.

4.3 Numerical experiments

We explore various aspects of the a posteriori error estimates using several examples chosen to stress particular characteristics. We are particularly focussed on the relative sizes of error contributions and the overall accuracy of the estimates. To measure the latter, we consider the effectivity ratio defined by,

$$\mathcal{E} = \frac{\text{Estimated Error}}{\text{Exact Error}}$$

and the "error" or divergence from 1, $|1 - \mathcal{E}|$. In some cases, we construct the problem to have a known solution so the error is computable. In other cases, we solve the problem using a higher order method with very fine time steps to get a much more accurate approximation we use to approximate the true error.

In the following examples, we consider the explicit trapezoid (RK2), RK4/2, and second order Adams-Bashforth (AB2) methods. All of these methods are second order and the approximate solutions are obtained by solving the corresponding modified equation with cG(1). We solve the adjoint problems using the third order cG(2) method. When we wish comparison with the modified equation, we solve that problem using the fully implicit cG(1) method.

4.3.1 The Lorenz equation

The first example is the Lorenz system,

$$\begin{cases} \dot{y}_1 = -\sigma y_1 + \sigma y_2 \\ \dot{y}_2 = r y_1 - y_2 - y_1 y_3 \\ \dot{y}_3 = -b y_3 + y_1 y_2 \\ y(0) = [-9.408, -9.096, 28.581]^T, \end{cases} \tag{4.7}$$

where we choose common values $\sigma = 10.1, r = 28.0$ and $b = 8/3$. This is a well-known chaotic system [10,18]. In [18], the properties of the a posteriori error estimate for the implicit dG and cG methods is explored. The conclusion is that the estimates are accurate up to a critical time, at which point the error in the solution gets large and linearization error leads to inaccurate estimates.

In the following examples, we use a uniform time step of $k_n = 1e - 2$ and set $\psi = \mathbf{0}, \psi_T = \mathbf{1}$ for a sequence of final times T between 1 and 20. To approximate the true error, we solve the problem with a high order finite element method and very fine time step. Fig. 4.1 shows the error in the effectivity ratio for the cG(1), RK2, and AB2 methods. We see that there is little qualitative difference between the three methods. The estimates for the explicit methods are less accurate than for the implicit method. We note that the error in the explicit methods grows more rapidly, which means the linearization error in the definition of the adjoint increases more quickly as well. On the other hand, the difference in the effectivity ratios is not significant, and estimate for the explicit methods is reasonably accurate.

In Fig. 4.1, we also plot the absolute error contributions for the RK2 and AB2 methods. In both cases, the explicit contribution dominates, and it dominates more in the extrapolation method. This is common to most examples we tested. We also see that the quadrature error is negligible. This is because the Lorenz system is almost linear, containing only two bilinear terms.

4.3.2 A highly nonlinear example

The next example is

$$\begin{cases} \dot{y} = y(1 + \tanh(\alpha(y - .3))) - te^{-t}\left(-2 + 2\,t + t\tanh\left(\alpha\left(t^2e^{-t} - .3\right)\right)\right), \\ y(0) = 0, \end{cases} \tag{4.8}$$

where the forcing term is constructed to give the exact solution $y(t) = t^2e^{-t}$. The hyperbolic tangent, with $\alpha = 100$, in the nonlinearity leads to a sudden change in value as the solution changes, providing a challenge for extrapolation methods. We again compare the cG(1), RK2, and AB2 methods. We set $\psi =$

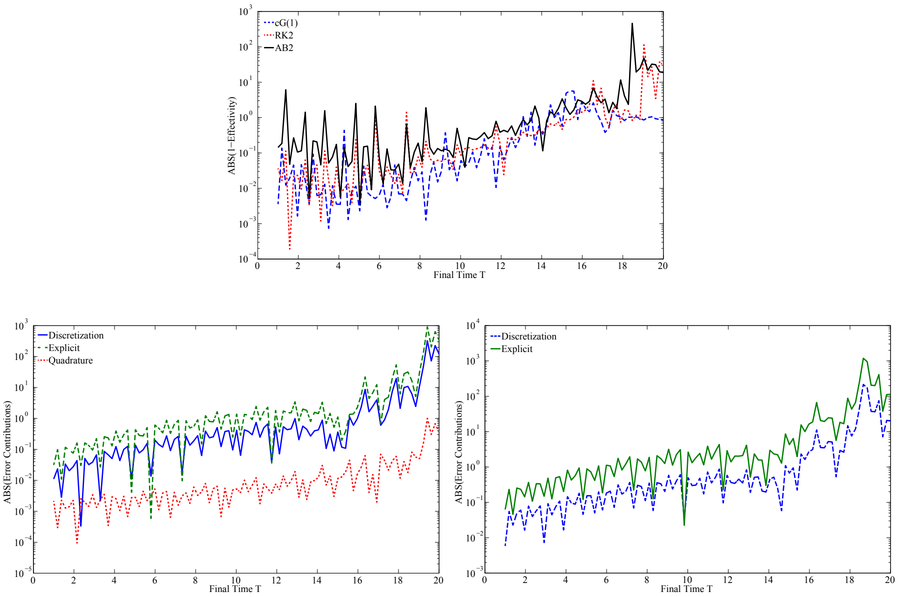

Fig. 4.1 Results for the Lorenz problem (4.7). Left: Error in the effectivity ratio for the cG(1), RK2 and AB2 methods. Middle: Error contributions for RK2. Right: Error contributions for AB2.

$0, \psi_T = \mathbf{1}$ for a sequence of final times T. The modified equations in the explicit methods are solved with a uniform time step of $k_n = 1e-3$.

Fig. 4.2 shows solutions and the error in the effectivity ratio for all three methods. Around $t = 4$, there is a large jump in the error of the effectivity for the explicit methods corresponding to a sudden loss of accuracy. We also plot the error contributions for the RK2 and AB2 methods. For the RK2 method, the quadrature error dominates all contributions while the explicit contribution dominates the discretization contribution. For AB2, the explicit contribution is dominant.

4.3.3 The two body problem

We next consider the well known two body problem

$$\begin{cases} \dot{y}_1 = y_3, \quad \dot{y}_3 = \dfrac{-y_1}{(y_1^2+y_2^2)^{3/2}}, \\ \dot{y}_2 = y_4, \quad \dot{y}_4 = \dfrac{-y_2}{(y_1^2+y_2^2)^{3/2}} \\ y(0) = [0.4, 0, 0, 2.0]^\top. \end{cases}$$

The two body problem is a Hamiltonian system with a complicated dynamic structure that can be exploited in numerical methods. However, we use it here to investigate the accuracy of standard integration methods. For the specified

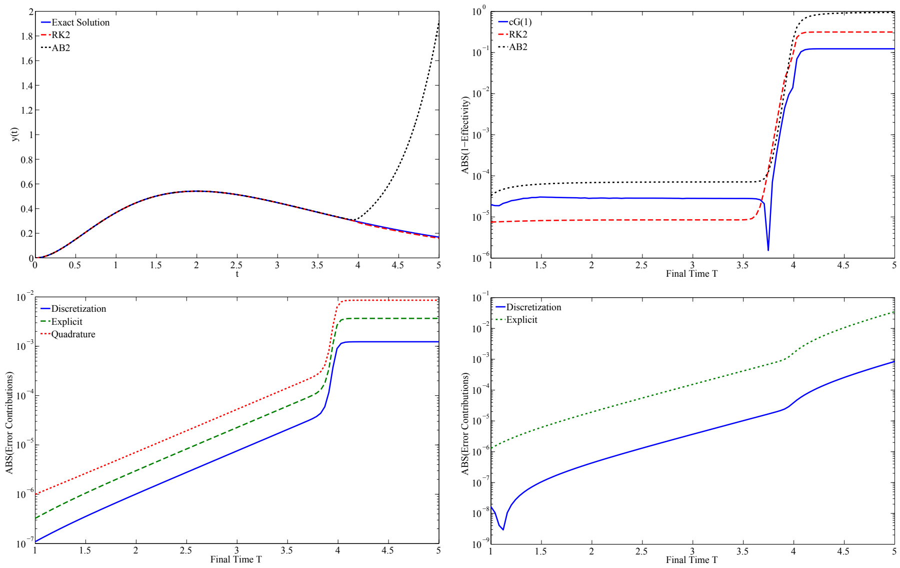

Fig. 4.2 Results for the nonlinear problem (4.8). Upper left: Exact and approximate solutions, note that explicit methods becomes highly inaccurate around $T = 4$. Upper right: Error in the effectivity ratios. Lower left: Error contributions for RK2. Lower right: Error contributions for AB2.

choice of initial value, there is an exact analytic periodic solution determined by the equation,

$$y = \left[\cos(\tau) - .6, .8\sin(\tau), \frac{-\sin(\tau)}{1 - .6\cos(\tau)}, \frac{.8\cos(\tau)}{1 - .6\cos(\tau)} \right]^\top,$$

where τ solves $.6\cos(\tau) = 1$.

Fig. 4.3 show the error in the effectivity ratios for the cG(1), RK2, and AB2 methods. We see that the error estimate becomes quite inaccurate around specific times during the first part of the solution and the inaccuracy gradually increases as time passes.

Next, we examine the absolute error contributions from each time interval during one computation. For this we use the quantity of interest defined by $\psi \equiv \mathbf{1}$ and $\psi_T = \mathbf{0}$, which gives the error in a weighted average over $[0, T]$. We solve up to time $T = 12.55$ using a uniform time step of $k_n = .01$. In Fig. 4.3, we plot the error contributions for RK2. We note that the quadrature contribution dominates during periods of the solution, so we also present results for the RK2/4 method. As expected, the discretization and explicit contributions significantly dominate the quadrature contribution in RK2/4 because we are computing integrals in the variational formulation more accurately.

Since the quadrature error only dominates over part of the domain, while the RK2/4 method costs more per time step than RK2, this suggest use of an

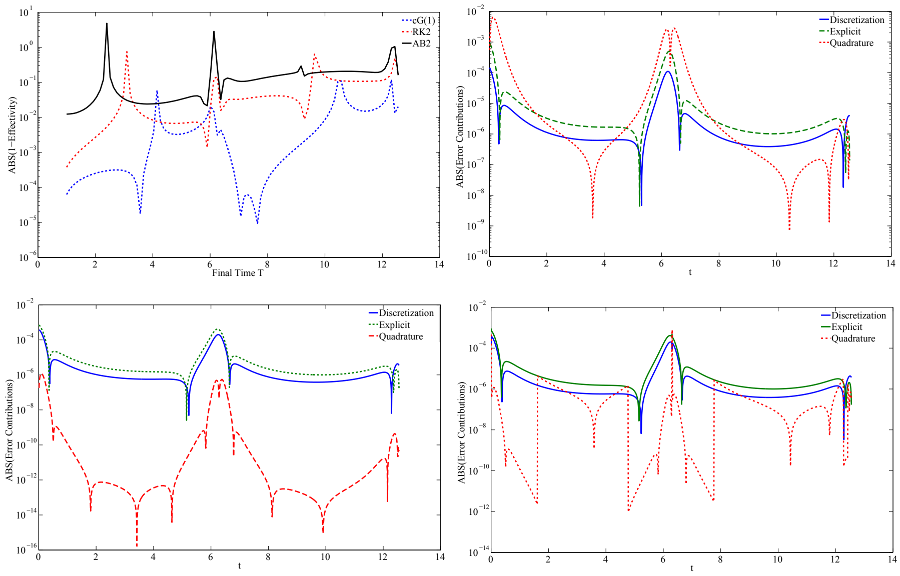

Fig. 4.3 Results for the two body problem. Upper left: Error in the effectivity ratio for cG(1), RK2, and AB2. Upper right: Absolute error contributions from each time interval for RK2. Lower left: Absolute error contributions from each time interval for RK2/4. Lower right: Absolute error contributions from each time interval for the adaptive trapezoid method.

adaptive quadrature approach in which the higher order quadrature is only used when the quadrature contribution is dominant. The algorithm is:

```
IF(ABS(Quadrature) < MAX(ABS(Discretization),ABS(Explicit))) THEN
Solve with RK2
ELSE
Solve with RK2/4
END
```

We present the results for this adaptive algorithm and see the quadrature error has been reduced so that it no longer dominates.

In Table 4.1, we give errors for various quantities of interest for RK2, RK2/4, the adaptive quadrature method, and RK2 with the time step cut in half. We see that the adaptive scheme obtains comparable accuracy to the RK2/4 scheme. Also, for the average error over the whole domain and the error at the final time, the error for the adaptive scheme is better than the error obtained by halving the time step which has more computational cost. However, if we consider the error in the second component as the quantity of interest, the adaptive scheme and RK2/4 do not improve the accuracy, while of course halving the time step does. This suggests that for certain quantities of interest, an adaptive quadrature scheme can give improved accuracy for less cost than halving the time step.

	$\psi = \mathbf{1}, \psi_T = \mathbf{0}$	$\psi = \mathbf{0}, \psi_T = \mathbf{1}$	$\psi = (0100)^\top, \psi_T = \mathbf{0}$
RK2	1.34e-1	-1.06e-1	-3.16e-2
RK2/4	-1.19e-2	-1.49e-2	-2.95e-2
Adaptive Quad.	-1.08e-2 (37.5%)	-1.58e-2 (45.3%)	-3.60e-2 (28.9%)
RK2 with $\frac{k_n}{2}$	3.11e-2	-3.04e-2	-7.68e-3

Table 4.1 Errors in various quantities of interest for the explicit trapezoid method and it's variations.

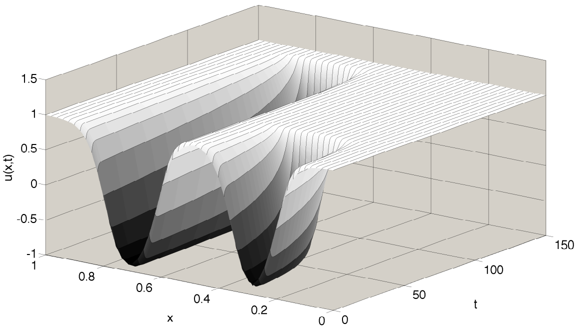

Fig. 4.4 Exact solution of the Bistable problem with given initial data. One well collapses at $t \approx 41$ while the other collapses at $t \approx 141$.

4.3.4 The Bistable problem

We next consider solution of a large dimension system obtained by a method of lines discretization in space of the well known bistable, or Allen-Cahn, parabolic problem,

$$\begin{cases} u_t = u - u^3 + \epsilon\, u_{xx}, & 0 < x < 1,\ 0 < t, \\ \frac{\partial u}{\partial x}(0,t) = \frac{\partial u}{\partial x}(1,t) = 0, & 0 < t, \\ u(x,0) = u_0(x), & 0 < x < 1. \end{cases}$$

For small ϵ, the solution of this problem exhibits "metastability", that is long periods of quasi-steady state behavior punctuated by rapid transients [11]. We consider initial data that gives two metastable periods over $[0, 150]$,

$$u_0(x) = \begin{cases} \tanh((.2 - x)/(2\sqrt{\epsilon})), & 0 \le x < .28, \\ \tanh((x - .36)/(2\sqrt{\epsilon})) & .28 \le x < .4865, \\ \tanh((.613 - x)/(2\sqrt{\epsilon})) & .4865 \le x < .7065, \\ \tanh((x - .8)/(2\sqrt{\epsilon})) & .7065 \le x < 1, \end{cases}$$

We show the numerical solution with $\epsilon = .0009$ in Fig. 4.4. The solution begins with two "wells" and at $t \approx 41$ and $t \approx 141$, the wells sharply collapse.

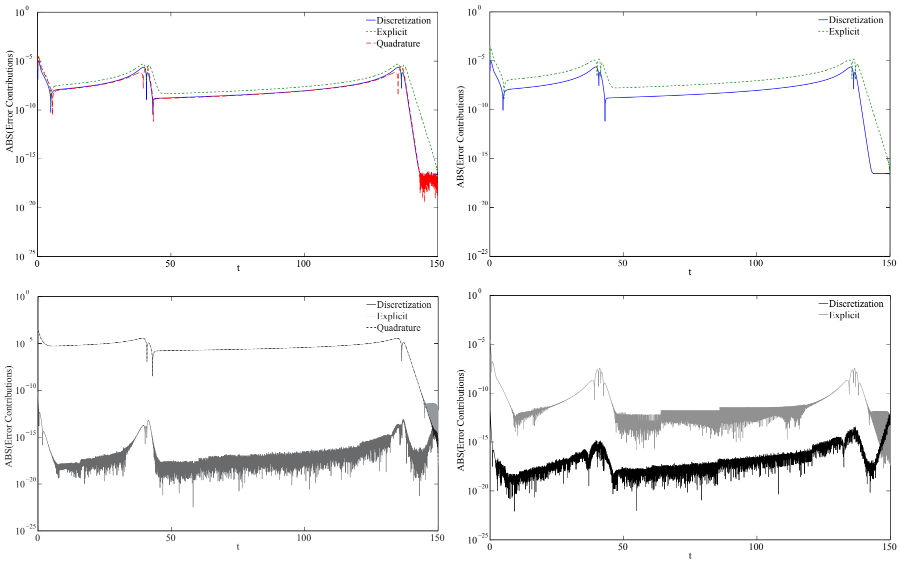

Fig. 4.5 Results for the discretized Bistable problem. Upper left: Absolute error contributions from each time interval for RK2. Upper right: Absolute error contributions from each time interval for AB2. Lower left: Absolute error contributions from each time interval for RK4. Lower right: Absolute error contributions from each time interval for AB4.

We discretize the spatial variable with a standard cG(1) finite element method using a uniform mesh size of $h = 0.02$ to obtain,

$$\begin{cases} \dot{\mathbf{u}} = \mathbf{u} - \mathbf{u}^3 + \dfrac{\epsilon}{h^2} A\mathbf{u}, & 0 < t, \\ \mathbf{u}(0) = u_0(\mathbf{x}), & \end{cases}, \qquad A = \begin{bmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{bmatrix},$$

where $\mathbf{x}$ is the discretized spatial variable. We present results for the RK2 and AB2 methods as well as the well known fourth order Runge-Kutta method (RK4) and fourth order Adams-Bashforth method (AB4), all using the uniform time step of $k_n = .01$. For the latter two, we solve the respective modified equations with the cG(3) finite element method and the adjoint problems with the cG(4) method. For a quantity of interest, we consider the average of the error over both the temporal and spatial domain. We solve up to time $T = 150$ in order to include both collapses of the wells.

We plot the absolute error contributions from each time interval for one solution for all four methods in Fig. 4.5. We see two sharp jumps in the error contributions arising from rapid transitions of the solution. It is also interesting to observe the difference between the explicit contribution and the discretization contribution. For RK2 they are fairly close, while for AB2 the contributions are about twice as far apart. This makes sense as the extrapolation approximation is less accurate than the Taylor series approximation. In the higher order methods, we see that the explicit contribution is significantly larger than the discretization contribution. This suggest that explicit contribution is more significant for higher order methods.

5 Another a posteriori error estimate

In the a posteriori results presented above, we use the same adjoint problem that is typically employed for analysis of the original fully implicit dG and cG finite element methods. This choice has proved reasonable for dealing with quadrature approximations, but whether or not this is reasonable for the Taylor series and polynomial extrapolation variations is less clear. In this section, we present a different approach to a posteriori analysis that uses both an adjoint associated with the true solution operator and an adjoint associated with the discretization.

5.1 Using modified and continuous adjoints

We note above that both the Taylor series and extrapolation approximation methods can be obtained by applying the standard dG and cG methods plus quadrature to a modified equation. We use the respective modified equations to define adjoints associated with the discrete scheme. We linearize each problem independently in a neighborhood of a given function. We then define adjoint problems for those linear problems and use those to derive an a posteriori error estimate. Any specified smooth function can be used for this approach. A typical choice is a common steady-state solution, see [7].

We present the result for the Taylor series approximation. We begin by linearizing $\mathcal{N}(y,v)$ about the specified function $w(t)$,

$$\begin{aligned}
0 = \mathcal{N}(y,v) &= \mathcal{N}(w,v) + \int_0^1 \mathcal{N}'(sy+(1-s)w, v; y-w)\,ds \\
&= \mathcal{N}(w,v) + \langle \dot{y} - \dot{w} - A_w(y-w), v\rangle_{[0,T]} \\
&= \langle A_w w - f(w,t), v\rangle_{[0,T]} + \langle \dot{y} - A_w y, v\rangle_{[0,T]},
\end{aligned}$$

and linearize $\mathcal{N}_P(Y,v)$ about the same function,

$$\begin{aligned}
-\sum_{n=1}^{N}\langle \mathcal{R}_P(Y),v\rangle_{I_n} &= \mathcal{N}_P(Y,v)=\mathcal{N}_P(w,v)\\
&\qquad + \int_0^1 \mathcal{N}_P'(sY+(1-s)w,v;Y-w)\,ds\\
&= \mathcal{N}_P(w,v)+\sum_{n=1}^{N}\langle \dot{Y}-\dot{w}-A_{P_n,w}B_{n,w}(Y-w)\rangle_{I_n}\\
&= \sum_{n=1}^{N}\Big(\langle A_{P_n,w}B_{n,w}w-f(P_nw,t)\rangle_{I_n}\\
&\qquad + \langle \dot{Y}-A_{P_n,w}B_{n,w}Y\rangle_{I_n}\Big),
\end{aligned}$$

where,

$$\begin{aligned}
A_w &:= \int_0^1 f'(sy+(1-s)w,t)\,ds\\
A_{P_n,w} &:= \int_0^1 f'(sP_nY+(1-s)P_nw,t)\,ds\\
B_{n,w} &:= \int_0^1 P_n'(sY+(1-s)w)\,ds
\end{aligned}$$

and P_n' denotes the Frechet derivative of the Taylor series map P_n. We then have the following linearized differential problems,

$$\langle f(w,t)-A_ww,v\rangle_{[0,T]}=\langle \dot{y}-A_wy,v\rangle_{[0,T]} \tag{5.1}$$

$$\sum_{n=1}^{N}\Big(\langle f(P_nw,t)-A_{P_n,w}B_{n,w}w-\mathcal{R}_P(Y),v\rangle_{I_n}=\sum_{n=1}^{N}\langle \dot{Y}-A_{P_n,w}B_{n,w}Y\rangle_{I_n}\Big). \tag{5.2}$$

We now define adjoint problems for these linear problems. For the original problem (5.1) we have the following adjoint problem: Find $\varphi_w\in H^1([0,T];\mathbb{R}^d)$ such that,

$$\begin{cases}\langle v,-\dot{\varphi}_w-A_w^*\varphi_w\rangle_{[0,T]}=\langle v,\psi\rangle_{[0,T]} \quad \forall\ v\in L^2([0,T];\mathbb{R}^d)\\ \varphi_w(T)=\psi_T\end{cases} \tag{5.3}$$

For the modified problem (5.2), we have the following adjoint problem: Find $\varphi_{P_w}\in H^1([0,T];\mathbb{R}^d)$ such that,

$$\begin{cases}\sum_{n=1}^{N}\{\langle v,-\dot{\varphi}_{P_w}\rangle_{I_n}-\langle B_{n,w}v,A^*_{P_n,w}\varphi_{P_w}\rangle_{I_n}\}=\langle v,\psi\rangle_{\Omega},\ \forall\ v\in L^2([0,T];\mathbb{R}^d)\\ \varphi_{P_w}(T)=\psi_T.\end{cases} \tag{5.4}$$

Using these adjoint problems, we prove:

Theorem 5 (Two Adjoint Error Representation Formula) *If $Y(t)$ is an approximation of* (4.1) *obtained via the cG(q) method with quadrature, and $w \in L^2([0,T];\mathbb{R}^d)$, then,*

$$\begin{aligned}
&\langle e, \psi\rangle_{[0,T]} + (e(T), \psi_T) \\
&= \sum_{n=1}^{N} \big(\underbrace{\langle \mathcal{R}_P(Y), \varphi_{P_w} - \pi_k \varphi_{P_w}\rangle_{I_n}}_{\textit{Discretization Contribution}} \big) \\
&\quad + \sum_{n=1}^{N} \big(\underbrace{\langle f(P_n Y, t), \pi_k \varphi_{P_w}\rangle_{I_n} - \langle f(P_n Y, t), \pi_k \varphi_{P_w}\rangle_{I_n, k_n}}_{\textit{Quadrature Contribution}} \big) \\
&\quad + \underbrace{\big(y_0, \varphi_w(0) - \varphi_{P_w}(0)\big)}_{\textit{Explicit Contribution}} \qquad (5.5) \\
&\quad + \sum_{n=1}^{N} \big(\underbrace{\langle f(w,t) - A_w w, \varphi_w\rangle_{I_n} - \langle f(P_n w, t) - A_{P_n,w} B_{n,w} w, \varphi_{P_w}\rangle_{I_n}}_{\textit{Difference in Linearization Error}} \big)
\end{aligned}$$

where φ_w and φ_{P_w} are solutions of (5.3) *and* (5.4) *respectives, and π_k and $\mathcal{R}_P(Y)$ are as above.*

If instead, $Y(t)$ is a dG(q) approximation, then,

$$\begin{aligned}
&\langle e, \psi\rangle_{[0,T]} + (e(T), \psi_T) \\
&= \sum_{n=1}^{N} \big(\underbrace{\langle \mathcal{R}_P(Y), \varphi_{P_w} - \pi_k \varphi_{P_w}\rangle_{I_n} - \big([Y]_{n-1}, \varphi_{P_w}(t_{n-1}) - \pi_k \varphi_{P_w}(t_{n-1}^+)\big)}_{\textit{Discretization Contribution}} \big) \\
&\quad + \sum_{n=1}^{N} \big(\underbrace{\langle f(P_n Y, t), \pi_k \varphi_{P_w}\rangle_{I_n} - \langle f(P_n Y, t), \pi_k \varphi_{P_w}\rangle_{I_n, k_n}}_{\textit{Quadrature Contribution}} \big) \\
&\quad + \underbrace{\big(y_0, \varphi_w(0) - \varphi_{P_w}(0)\big)}_{\textit{Explicit Contribution}} \qquad (5.6) \\
&\quad + \sum_{n=1}^{N} \big(\underbrace{\langle f(w,t) - A_w w, \varphi_w\rangle_{I_n} - \langle f(P_n w, t) - A_{P_n,w} B_{n,w} w, \varphi_{P_w}\rangle_{I_n}}_{\textit{Difference in Linearization Error}} \big)
\end{aligned}$$

Proof We begin by splitting the error,

$$\langle e, \psi\rangle_{[0,T]} = \langle y, \psi\rangle_{[0,T]} - \langle Y, \psi\rangle_{[0,T]}. \qquad (5.7)$$

We deal with each of these terms separately. For the first term, we use y as the test function in (5.3) to obtain,

$$\begin{aligned}
\langle y, \psi\rangle_{[0,T]} &= \langle y, -\dot{\varphi}_w - A_w^* \varphi_w\rangle_{[0,T]} \\
&= \langle \dot{y} - A_w y, \varphi_w\rangle_{[0,T]} + (y_0, \varphi_w(0)) - (y(T), \psi_T) \\
&= \langle f(w,t) - A_w w, \varphi_w\rangle_{[0,T]} + (y_0, \varphi_w(0)) - (y(T), \psi_T).
\end{aligned}$$

Looking at the second term, we use Y as the test function in (5.4) to obtain,

$$\begin{aligned}\langle Y, \psi\rangle_{[0,T]} &= \sum_{n=1}^{N} \Big(\langle Y, -\dot{\varphi}_{P_w}\rangle_{I_n} - \langle B_{n,w} Y, A^*_{P_n,w}\varphi_{P_w}\rangle_{I_n} \Big) \\ &= \sum_{n=1}^{N} \Big(\langle \dot{Y} - A_{P_n,w} B_{n,w} Y, \varphi_{P_w}\rangle_{I_n} + \big([Y]_{n-1}, \varphi_{P_w}(t_{n-1})\big) \Big) \\ &\qquad\qquad + (y_0, \varphi_{P_w}(0)) - (Y(T), \psi_T) \\ &= \sum_{n=1}^{N} \Big(\langle f(P_n w, t) - A_{P_n,w} B_{n,w} w - R_P(Y), \varphi_{P_w}\rangle_{I_n} \\ &\qquad + \big([Y]_{n-1}, \varphi_{P_w}(t_{n-1})\big) \Big) + (y_0, \varphi_{P_w}(0)) - (Y(T), \psi_T).\end{aligned}$$

Then equation (5.7) becomes,

$$\begin{aligned}\langle e, \psi\rangle_{[0,T]} = \sum_{n=1}^{N} \{ &\langle \mathcal{R}_P(Y), \varphi_{P_w}\rangle_{I_n} - \big([Y]_{n-1}, \varphi_{P_w}(t_{n-1})\big) \\ &+ \langle f(w,t) - A_w w, \varphi_w\rangle_{I_n} - \langle f(P_n w, t) - A_{P_n,w} B_{n,w} w, \varphi_{P_w}\rangle_{I_n} \} \\ &+ (y_0, \varphi_w(0) - \varphi_{P_w}(0)) - (e(T), \psi_T).\end{aligned}$$

We can then use Galerkin orthogonality to complete the proof. The proof for the cG methods is analogous.

A error estimate can also be derived for the extrapolation methods that use the modified equation (4.2). The estimate is very similar, the two main differences being in the definition of the residual and of the adjoint of the modified equation.

The a posteriori results (5.6) and (5.5) are considerably more complicated to use in practice than the previous a posteriori error estimates. Direct implementation requires the solution of two adjoint problems, while in addition approximating the adjoint solution φ is generally difficult in circumstances that require the use of explicit methods. Moreover, the contribution from "Difference in Linearization Error" is problematic to evaluate because it involves the true solution. In practice, we substitute the numerical solution, but this requires further manipulation to quantity the effect of this substitution. In practice, these estimates are often manipulated further to obtain expressions more amenable to computation plus additional terms that cannot be estimated but are provably higher order, see [7,8].

5.2 A numerical example

In this section, we explore the connection between stability issues arising with explicit integration and the differences between the adjoints to the continuous and modified problems as defined in Theorem 5.

We consider the solution of the system arising from a method of lines discretization of the linear heat equation,

$$\begin{cases} u_t = u_{xx}, & 0 < x < 3\pi,\ t > 0, \\ u(0,t) = u(1,t) = 0, & t > 0, \\ u(x,0) = u_0(x), & 0 < x < 3\pi, \end{cases}$$

where the initial data is given by $u_0(x) = \sin(x)$. Using a standard cG(1) method in space on a uniform mesh with spacing $h \approx .304$ gives,

$$\begin{cases} \dot{\mathbf{u}} = \frac{1}{h^2} A\mathbf{u}, & t > 0, \\ \mathbf{u}(0) = u_0(\mathbf{x}), \end{cases}, \quad A = \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}. \tag{5.8}$$

We solve (5.8) using the forward Euler method. The dG(0) finite element method is used to solve the corresponding modified equation. From the point of view of stability for numerical solution of the full partial differential equation, the forward Euler scheme with a uniform time step k_n must satisfy,

$$\gamma := \frac{k_n}{h^2} \le \frac{1}{2}.$$

Since instability does not occur in the true solution, the solution of the adjoint problem (1.2) does not indicate instability. However, the adjoint to the discretization (5.4) should reflect instability in the numerical solution. We plot relative error between the modified adjoint φ_{P_w} and the continuous adjoint φ_w, i.e.

$$RE(t) := \frac{\|\varphi_w(t) - \varphi_{P_w}(t)\|_\infty}{\|\varphi_w(t)\|_\infty}$$

where the norms are in $\mathbb{R}^N$, for two different values of γ. We consider two quantities of interest: (1) the average value over the entire spatio-temporal domain with final time $T = 5$ and (2) the average over the time interval $[0, 3]$ × the spatial domain.

We show the results in Fig. 5.1. When $\gamma = .4$, the stability condition is satisfied and there is relatively little difference between the adjoint solutions for either quantity of interest. However, the stability condition is violated when $\gamma = .6$ and we can see the large difference in the two adjoint solutions. Examining the results for the second quantity of interest shows that different quantities may have different stability properties.

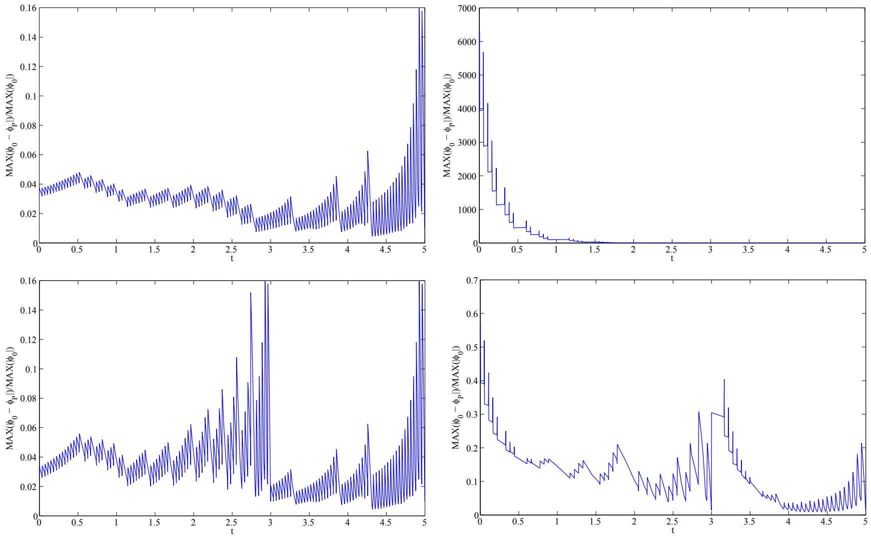

Fig. 5.1 Results for the discretized heat equation. Upper row: Relative difference in the adjoint solutions when the quantity of interest is the average value over the entire domain for $\gamma = .4$ (left) and $\gamma = .6$ (right). Lower row: Relative difference in the adjoint solutions when the quantity of interest is the average value over the first half of the domain for $\gamma = .4$ (left) and $\gamma = .6$ (right).

6 Conclusions

We present an a posteriori error analysis for approximate solutions of nonlinear ordinary differential equations solved with explicit finite difference methods. To obtain this analysis, we represent two classes of explicit finite difference methods as finite element methods, whose solution is defined over the entire domain. In particular, we distinguish between error contributions from the substitution of an approximation space, the approximation of the differential operator, and the use of quadrature to evaluate the integrals. This distinction allows us to determine the best method of adaptation to reduce the error in our approximation. Finally, we give an adjoint problem for the explicit method, and show how it can be used to determine numerical stability when compared with the adjoint of the ODE.

References

1. Ainsworth, M., Oden, J.: A posteriori error estimation in finite element analysis. Computer Methods in Applied Mechanics and Engineering **142**(1), 1–88 (1997)
2. Becker, R., Rannacher, R.: An optimal control approach to a posteriori error estimation in finite element methods. Acta Numerica 2001 **10**(1), 1–102 (2001)
3. Delfour, M., Trochu, F., Hager, W.: Discontinuous Galerkin methods for ordinary differential equations. mathematics of computation **36**(154), 455–473 (1981)

4. Eriksson, K., Estep, D., Hansbo, P., Johnson, C.: Introduction to adaptive methods for differential equations. Acta numerica **4**, 105–158 (1995)
5. Estep, D.: A posteriori error bounds and global error control for approximation of ordinary differential equations. SIAM Journal on Numerical Analysis pp. 1–48 (1995)
6. Estep, D.: Error estimates for multiscale operator decomposition for multiphysics models. In: J. Fish (ed.) Multiscale methods: bridging the scales in science and engineering. Oxford University Press, USA (2009)
7. Estep, D., Ginting, V., Ropp, D., Shadid, J., Tavener, S.: An a posteriori-a priori analysis of multiscale operator splitting. SIAM Journal on Numerical Analysis **46**(3), 1116–1146 (2008)
8. Estep, D., Ginting, V., Tavener, S.: A posteriori analysis of a multirate numerical method for ordinary differential equations (2009)
9. Estep, D., Holst, M., Mikulencak, D.: Accounting for stability: a posteriori error estimates based on residuals and variational analysis. Communications in numerical methods in engineering **18**(1), 15–30 (2002)
10. Estep, D., Johnson, C.: The pointwise computability of the Lorenz system. Mathematical Models and Methods in Applied Sciences **8**(8), 1277–1306 (1998)
11. Estep, D., Larson, M., Williams, R., Society, A.M.: Estimating the error of numerical solutions of systems of reaction-diffusion equations. American Mathematical Society (2000)
12. Estep, D., Stuart, A.: The dynamical behavior of the discontinuous Galerkin method and related difference schemes. Mathematics of computation **71**(239), 1075–1104 (2002)
13. Giles, M., Süli, E.: Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. Acta Numerica **11**(1), 145–236 (2002)
14. Gottlieb, S., Wei, G., Zhao, S.: A unified discontinuous Galerkin framework for time integration (2010)
15. Houston, P., Rannacher, R., Süli, E.: A posteriori error analysis for stabilised finite element approximations of transport problems. Computer methods in applied mechanics and engineering **190**(11), 1483–1508 (2000)
16. Marchuk, G.: Adjoint equations and analysis of complex systems, vol. 295. Springer (1995)
17. Marchuk, G., Agoshkov, V., Shutiaev, V.: Adjoint equations and perturbation algorithms in nonlinear problems. CRC (1996)
18. Sandelin, J.: Global estimate and control of model, numerical, and parameter error. Ph.D. thesis, Colorado State University (2007)