Published in final edited form as:

SIAM J Numer Anal. 2016 ; 54(5): 2974–3002. doi:10.1137/16M1079014.

A POSTERIORI ERROR ANALYSIS OF TWO STAGE COMPUTATION METHODS WITH APPLICATION TO EFFICIENT DISCRETIZATION AND THE PARAREAL ALGORITHM

JEHANZEB HAMEED CHAUDHRY*, DON ESTEP†, SIMON TAVENER‡, VARIS CAREY§, and JEFF SANDELIN¶

*Department of Mathematics & Statistics, The University of New Mexico, Albuquerque, NM 87131

†Department of Statistics, Colorado State University, Fort Collins, CO 80523

‡Department of Mathematics, Colorado State University, Fort Collins, CO 80523

§Department of Mathematical and Statistical Sciences, University of Colorado Denver, Denver, CO 80204

¶PD&L, DigitalGlobe, Inc., Longmont, Colorado 80503

Abstract

We consider numerical methods for initial value problems that employ a two stage approach consisting of solution on a relatively coarse discretization followed by solution on a relatively fine discretization. Examples include adaptive error control, parallel-in-time solution schemes, and efficient solution of adjoint problems for computing a posteriori error estimates. We describe a general formulation of two stage computations then perform a general a posteriori error analysis based on computable residuals and solution of an adjoint problem. The analysis accommodates various variations in the two stage computation and in formulation of the adjoint problems. We apply the analysis to compute “dual-weighted” a posteriori error estimates, to develop novel algorithms for efficient solution that take into account cancellation of error, and to the Parareal Algorithm. We test the various results using several numerical examples.

Keywords

A posteriori error estimation; adjoint problem; cancellation of error; efficient time step selection; initial value problem; parallel in time; Parareal algorithm; two stage computation

1. Introduction

We consider the numerical solution of a system of ordinary differential equations,

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

where the goal is to approximate the value of a quantity of interest (QoI),

$$Q(u) = \int_0^T (\psi(t), y(t)) \, dt, \quad (1.2)$$

for a specified $\psi(t) \in L_2(0, T]$, where $(\cdot, \cdot)$ is the Euclidean inner-product. Examples include $\psi \equiv 1/T$ for the average and an approximate delta function at a time of interest.

We consider numerical solutions that employ two stages of computation;

1. A first stage solution computed on a relatively coarse discretization,
2. A second stage solution computed on a relatively fine discretization.

Two situations that use such an approach are:

1. **Adaptive error control.** The first stage solution is used to compute information about the error that is then used to generate a second stage discretization which achieves accuracy with some semblance of efficiency. Classical adaptive error control algorithms perform stepsize selection on a step by step basis while “dual-weighted” approaches adapt based on a coarse global solve.
2. **Parallel-in-time methods.** The two stages are used to devise algorithms that provide a way to exploit parallelization for time integration [35, 15, 28], e.g. the *Parareal Algorithm* [39, 37, 38, 41, 29, 30]. This generally involves iteratively exchanging information computed from coarse scale and fine scale numerical solutions at time nodes that span a number of numerical time steps.

The primary focus of this paper is the derivation and testing of accurate a posteriori estimates for the error in the QoI computed from a two stage computation. Such estimates are a critical component of uncertainty quantification in which all sources of error and uncertainty have to be quantified, e.g. see [26, 27]. This is especially important in situations in which adaptive error control or parallel-in-time solutions are employed, since those correspond to situations in which numerical error is expected to be significant.

A posteriori error estimation provides another motivation for the use of a two stages. We use a posteriori error analysis that involves computational residuals of the numerical solution, the solution of an adjoint problem for (1.1), and variational analysis to produce an *accurate* error estimate for the QoI [23, 19, 25, 3, 1]. However, the numerical solution of the adjoint problem can involve significant computational cost, motivating the study:

- 3 **Computation of a posteriori error estimates.** Use a two stage computation to reduce the computational cost of computing numerical solutions of an adjoint equation.

It turns out that it is possible to carry out a unified a posteriori error analysis that applies to all three situations.

1.1. A comment on the finite element formulation of the discretizations

The a posteriori error analysis employs a variational argument using the ingredients of computable residuals and the solution of an adjoint problem related to the QoI. It is naturally suited to finite element discretizations based on a variational formulation of (1.1).

However, we stress that the analysis can be extended to a wide variety of finite difference schemes at the cost of additional terms in the estimate. Namely, if we employ a quadrature formula to evaluate the integral involving the right-hand side f in a finite element discretization of (1.1), the resulting equations can be interpreted as finite difference equations for the coefficients determining the finite element solution. For example, the Crank-Nicolson or trapezoidal rule is obtained when we apply the trapezoidal rule quadrature formula to evaluate the integral in the continuous Galerkin piecewise linear finite element method and the implicit/backward Euler scheme results when we apply the right-hand quadrature rule to the integral for the discontinuous Galerkin piecewise constant method. We do not know if *all* finite difference schemes can be represented as a finite element method combined with an appropriate quadrature formula, but a great number of common schemes can. The early papers [12, 13] contain equivalencies for a number of finite difference schemes. Finite difference schemes for which a posteriori error estimates have been derived include explicit schemes, IMEX methods, BDF, Runge-Kutta, Lax-Wendroff, and multirate methods [17, 36, 22, 6, 8, 9, 7].

The a posteriori error analysis for the finite element method plus quadrature is similar to the analysis carried out below, but requires the addition of new terms quantifying the effects of using a quadrature formula. The new terms typically have the form of a adjoint-weighted quadrature residual. In some cases, it is necessary to modify the definition of the adjoint operator. In any case, combining the changes made to deal with quadrature and the analysis of the two stage approach below is completely straightforward.

1.2. Outline

In §2, we present an abstract formulation of two stage computations. We derive various a posteriori error estimates for the general two stage computation in §3, allowing for variations in the forward solve and in the solution of an adjoint problem. We describe applications of the estimates to standard dual weighted adaptive error control, new strategies for constructing efficient discretizations that take into account cancellation of errors, and to the Parareal algorithm in §4. Conclusions and future directions are presented in §5.

2. Generic formulation of two stage computation

We begin with a generic formulation of a two stage computation. Stage 1 is a relatively coarse discretization for which we assume it is possible to compute a classic a posteriori error estimate including solution of an adjoint problem. Stage 2 is a relatively accurate discretization computed with various modifications of a standard discretization, see Figure

2.1. We also consider various modifications in the standard computation of an a posteriori estimate for the Stage 2 solution.

For the sake of simplicity, we present the analysis for the continuous Galerkin method for each stage [21]. We describe the modifications needed for the discontinuous Galerkin method in §3.3.

We set $0 = T_0 < \dots < T_{p-1} < T_p < \dots < T_P = T$ to be a partition of the time domain. The Stage 1 and Stage 2 solves may exchange information at these time nodes, which can be viewed as “synchronization” times. We call the time interval $[T_{p-1}, T_p]$ subdomain p .

2.1. Stage 1

We compute the approximation $\hat{Y}^p(t)$ of (1.1) in $(T_{p-1}, T_p]$ with initial condition a at $t = T_{p-1}$ using a numerical scheme $\hat{G}^p[\alpha](t)$, where the superscript p indicates that $t \in (T_{p-1}, T_p]$. For simplicity, we assume the numerical solver $\hat{G}$ is the continuous Galerkin method of order q , $cG(q)$. We partition $[T_{p-1}, T_p]$ as $T_{p-1} = t_0^p < \dots < t_{N^p-1}^p < t_{N^p}^p = T_p$, as shown in Figure 2.1, with $\hat{I}_{\hat{n}}^p = [t_{\hat{n}-1}^p, t_{\hat{n}}^p]$. Let $\hat{\mathcal{T}}^p = \{\hat{I}_1^p, \dots, \hat{I}_{\hat{n}}^p, \dots, \hat{I}_{N^p}^p\}$. The approximation is in the space $\hat{\mathcal{V}}_{\hat{n}}^{p,q}$, which is the set of polynomials of degree q on the interval $\hat{I}_{\hat{n}}^p$. The $cG(q)$ method is to find $\hat{Y}^p \in \hat{\mathcal{V}}_{\hat{n}}^{p,q}$ such that,

$$\int_{t_{\hat{n}-1}^p}^{t_{\hat{n}}^p} (\dot{\hat{Y}}^p, v) dt = \int_{t_{\hat{n}-1}^p}^{t_{\hat{n}}^p} (f(\hat{Y}^p, t), v) dt, \quad \forall v \in \hat{\mathcal{V}}_{\hat{n}}^{p,q-1}, \quad (2.1)$$

for $1 \leq \hat{n} \leq N^p$ and $\hat{Y}^p(T_{p-1}) = \hat{z}_{p-1}$. For $p = 1$, we set $\hat{Y}^p(0) = y(0)$. If we set $\hat{z}_{p-1} = \hat{Y}^{p-1}(T_{p-1})$, we obtain the global $cG(q)$ method on $[0, T]$, which we call the “serial” solution. Other options for $\hat{z}_{p-1}$ are explored in §4.3.

2.2. Stage 2

For Stage 2, we use a second numerical scheme $H^p[\alpha](t)$ which is more accurate than $\hat{G}$, e.g. employing a finer discretization or a higher order method. For simplicity, we use the $cG(q)$ method with a finer partition as $T_{p-1} = t_0^p < \dots < t_{N^p-1}^p < t_{N^p}^p = T_p$ as shown in Figure 2.1. We let $I_n^p = [t_{n-1}^p, t_n^p]$ and $\mathcal{T}^p = \{I_1^p, \dots, I_n^p, \dots, I_{N^p}^p\}$. The $cG(q)$ method is to find $Y^p \in \mathcal{V}_n^{q,p}$ such that,

$$\int_{t_{n-1}^p}^{t_n^p} (\dot{Y}^p, v) dt = \int_{t_{n-1}^p}^{t_n^p} (f(Y^p, t), v) dt, \quad \forall v \in \mathcal{V}_n^{p,q-1}, \quad (2.2)$$

with initial conditions $Y^p(T_{p-1}) = z_{p-1}$. For subdomain 1, we always have $z_0 = y(0)$. For the other subdomains, we consider two options for conditions at the left-hand time node:

1. Set $z_{p-1} = Y^{p-1}(T_{p-1})$ yielding the standard $cG(q)$ method on $[0, T]$.

2. Initialize the solution in each subdomain with the values from the solution from Stage 1 by setting $z_{p-1} = \hat{z}_{p-1} = \hat{Y}^p(T_{p-1})$. Note the solution may be discontinuous at time nodes T_p for $1 < p < P - 1$. In this approach, the solutions in each subdomain may be computed in parallel.

Remark 1—We use $\hat{\cdot}$ to indicate a Stage 1 quantity and a super-script p to reference the subdomain $[T_{p-1}, T_p]$.

3. A posteriori analysis for two stage computation

The aim is to estimate the error in a QoI for the two stage computation in an accurate yet computationally efficient manner. There are multiple choices for an “adjoint” problem that yield estimates involving a trade-off between accuracy and computational cost.

3.1. A posteriori analysis for Stage 1

We use the traditional adjoint-based a posteriori error analysis for Stage 1. The error in the QoI is defined as,

$$\int_0^T (\hat{e}^p, \psi) dt = \int_0^T (y - \hat{Y}^p, \psi) dt. \quad (3.1)$$

We choose an appropriate adjoint problem to (1.1) relative to the goal of estimating the error in the QoI (3.1). However, while a linear operator is associated to a unique adjoint operator, in general there are a number of possible “adjoint-like” operators associated to a given *nonlinear* operator [40, 20]. The issue is to choose an adjoint that is useful for the purpose in hand.

There is a standard choice of adjoint for error and perturbation analysis which uses the Integral Mean Value Theorem [40, 20]. For (1.1), if we define

$$\overline{f'(y, \hat{Y}^p)} = \int_0^1 \frac{\partial}{\partial s} f(sy + (1-s)\hat{Y}^p) ds, \quad (3.2)$$

then

$$f(y) - f(\hat{Y}^p) = \overline{f'(y, \hat{Y}^p)} \hat{e}^p. \quad (3.3)$$

The Stage 1 adjoint is then,

$$\begin{cases} -\dot{\hat{\phi}}(t) = \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi} + \psi(T), & t \in [0, T), \\ \hat{\phi}(T) = 0. \end{cases} \quad (3.4)$$

We note that definition (3.2) requires both the approximation and the true solution, which is impractical in practice. We address this in Remark 2 below.

The a posteriori analysis begins with a local representation.

Lemma 3.1—On $[T_{p-1}, T_p]$, we have,

$$(\hat{\phi}, \hat{e}^p)(T_p) + \int_{T_{p-1}}^{T_p} (\psi, \hat{e}^p) dt = (\hat{\phi}, \hat{e}^p)(T_{p-1}) + \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}), \quad (3.5)$$

where $\hat{\mathcal{R}}^p = \dot{\hat{Y}}^p + f(\hat{Y}^p)$ is the Stage 1 residual and

$$\hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}) = \sum_{\hat{n}=1}^{\hat{N}^p} \int_{\hat{I}_{\hat{n}}^p} (\hat{\mathcal{R}}^p, \hat{\phi}) dt. \quad (3.6)$$

Note that $\hat{Y}^p$ implicitly depends on $\hat{\mathcal{J}}^p$.

Proof—Using (3.4) on $\hat{I}_{\hat{n}}^p$,

$$\int_{\hat{I}_{\hat{n}}^p} (\psi, \hat{e}^p) dt = \int_{\hat{I}_{\hat{n}}^p} (-\dot{\hat{\phi}} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{e}^p) dt = -(\hat{\phi}, \hat{e}^p)_{\hat{n},p} + (\hat{\phi}, \hat{e}^p)_{\hat{n}-1,p} + \int_{\hat{I}_{\hat{n}}^p} (\hat{\phi}, \dot{y} - \dot{\hat{Y}}^p) - (\hat{\phi}, f(y) - f(\hat{Y}^p)) dt.$$

Now (1.1) implies,

$$\int_{\hat{I}_{\hat{n}}^p} (\psi, \hat{e}^p) dt = -(\hat{\phi}, \hat{e}^p)_{\hat{n},p} + (\hat{\phi}, \hat{e}^p)_{\hat{n}-1,p} + \int_{\hat{I}_{\hat{n}}^p} (\hat{\mathcal{R}}^p, \hat{\phi}) dt. \quad (3.7)$$

Summing (3.7) from $\hat{n}=1$ to $\hat{n}=\hat{N}^p$ yields (3.5). $\square$

Theorem 3.1—The error in the QoI for the Stage 1 solution satisfies,

$$\int_0^T (\psi, \hat{e}^p) dt = \sum_{p=2}^P (\hat{\phi}(T_{p-1}), \hat{Y}^{p-1}(T_{p-1}) - \hat{z}_{p-1}) + \sum_{p=1}^P \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}).$$

Proof—Summing (3.5) for $p = 1, \dots, P$,

$$\int_0^T (\psi, \hat{e}^p) dt = \sum_{p=1}^P \left[-(\hat{\phi}, y - \hat{Y}^p)(T_p) + (\hat{\phi}, y - \hat{Y}^p)(T_{p-1}) + \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}) \right].$$

Since $\hat{\phi}$ and y are continuous in $[0, T]$, $y(0) = \hat{Y}^p(0)$ and $\hat{\phi}(T) = 0$ we have,

$$\int_0^T (\psi, \hat{e}^p) dt = \sum_{p=1}^{P-1} (\hat{\phi}, \hat{Y}^p)(T_p) - \sum_{p=2}^P (\hat{\phi}, \hat{Y}^p)(T_{p-1}) + \sum_{p=1}^P \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}), = \sum_{p=2}^P (\hat{\phi}, \hat{Y}^{p-1} - \hat{Y}^p)(T_{p-1}) + \sum_{p=1}^P \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}).$$

Finally observing that $\hat{Y}^p(T_{p-1}) = \hat{z}_{p-1}$ proves the result.

An immediate consequence:

Corollary 3.1—Let $\hat{z}_{p-1} = \hat{Y}^{p-1}(T_{p-1})$, then,

$$\int_0^T (\psi, \hat{e}^p) dt = \sum_{p=1}^P \hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}). \quad (3.8)$$

Remark 2—The linearization (3.2) requires the unknown true solution. In practice, we create an approximate adjoint problem by linearizing around a computed numerical solution and then compute the adjoint solution numerically. Hence, we obtain an approximate error estimate from the exact error representation. This is well established in the literature [18, 25, 20] and produces robust accuracy. This approach is employed for all of the adjoint problems discussed in this paper. For the sake of simplicity, we do not write approximate versions of the exact error representations.

3.2. A posteriori analysis for Stage 2

The error representation for Stage 2 depends on how the solutions Y^p on each subdomain p are initialized and the choice of adjoint problem.

3.2.1. Analysis for Stage 2 using the Stage 1 adjoint problem—In this approach, we use the adjoint solution computed for Stage 1 to form an estimate. The different linearization points for the adjoint problems for Stage 1 and 2 means that the analog of (3.3) does not hold.

Lemma 3.2: We have,

$$\begin{aligned} (\hat{\phi}, y - Y^p)(T_p) \\ + \int_{T_{p-1}}^{T_p} (\psi, e^p) dt = (\hat{\phi}, y - Y^p)(T_{p-1}) + \sum_{n=1}^{N^p} \left[\int_{I_n^p} (\mathcal{R}^p, \hat{\phi}) dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{Y}^p - Y^p) dt + \int_{I_n^p} (\overline{f'(y, Y^p)}) \right] \end{aligned} \quad (3.9)$$

where $\mathcal{R}^p = -\dot{\hat{Y}}^p + f(Y^p)$ is the Stage 2 residual.

Proof: On each interval $I_n^p \in (T_{p-1}, T_p)$ we have,

$$\int_{I_n^p} (\psi, e^p) dt = \int_{I_n^p} (-\dot{\hat{\phi}} - \overline{f'(y, \hat{Y}^p)})^\top \hat{\phi}, e^p dt = \int_{I_n^p} (-\dot{\hat{\phi}} - \overline{f'(y, Y^p)})^\top \hat{\phi}, e^p dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, e^p dt = -(\hat{\phi}, e^p)_{n,p}.$$

Summing from $n = 1$ to $n = N^p$,

$$\int_{T_{p-1}}^{T_p} (\psi, e^p) dt = (\hat{\phi}, y - Y^p)(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) + \sum_{n=1}^{N^p} \int_{I_n^p} (\mathcal{R}^p, \hat{\phi}) dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, e^p dt.$$

Adding and subtracting $\sum_{n=1}^{N^p} \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{Y}^p dt$ leads to,

$$\int_{T_{p-1}}^{T_p} (\psi, e^p) dt = (\hat{\phi}, y - Y^p)(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) + \sum_{n=1}^{N^p} \left[\int_{I_n^p} (\mathcal{R}^p, \hat{\phi}) dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, Y^p)}^\top \hat{\phi}, \hat{Y}^p - Y^p dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{Y}^p - Y^p dt \right]$$

Theorem 3.2: The error in QoI for the Stage 2 solution using the adjoint for Stage 1 (3.4) is,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\hat{\phi}, Y^{p-1} - Y^p)(T_{p-1}) + \sum_{n=1}^{N^p} \left(\int_{I_n^p} (\mathcal{R}^p, \hat{\phi}) dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{Y}^p - Y^p dt + \int_{I_n^p} (\overline{f'(y, Y^p)})^\top \hat{\phi} - \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}, \hat{Y}^p - Y^p dt \right) \right]$$

(3.10)

Proof: The proof follows from Lemma 3.2.

The last term in (3.10) involves the true solution y , so is non-computable. Analogous non-computable terms in an a posteriori error representation arise for some kinds of discretizations, e.g. explicit methods, and in numerical solution of some complex problems, e.g., coupled multiphysics models. In a number of situations, such terms can be proved to be higher order than the computable terms, and hence can be safely neglected when computing the error estimate, at least in the limit of refined discretization [22, 5]. An example is discussed in Remark 2. However other discretization choices lead to non-computable expressions in the error analysis that cannot be shown to be higher order. We show an example in §4.1 where the non-computable terms are not higher order, and hence neglecting them affects the accuracy of the computed error estimate. However, even in these cases, the a posteriori estimate obtained by considering only the computable terms may still be useful, e.g., as an error indicator useful for adaptive error control.

3.2.2. Analysis for Stage 2 using various Stage 2 adjoints—We use an adjoint problem based on linearization involving the Stage 2 solution, but with various modifications designed to yield a computationally inexpensive adjoint problem. The Stage 2 adjoint on the p^{th} subdomain is defined as,

$$\begin{cases} -\dot{\phi}^p(t) = \overline{f'(y, Y^p)}^\top \phi^p + \psi(T), & t \in [T_{p-1}, T_p), \\ \phi^p(T_p) = w_p, \end{cases} \quad (3.11)$$

with $w_p = 0$. We describe and analyze three strategies based on different choices of the initial condition w_p , $p = 1, \dots, P-1$:

1. A globally continuous adjoint solution for $w_p = \phi^{p+1}(T_p)$.
2. Choosing $w_p = 0$, $p = 1, \dots, P-1$, which yields an adjoint solution that is continuous in a subdomain p but may be discontinuous across T_p , $p = 1, \dots, P-1$.
3. Use the Stage 1 adjoint by setting $w_p = \hat{\phi}(T_p)$.

Choices 2 and 3 open the possibility of solving the subdomain adjoint problems in parallel.

First, we derive a generic result. On each interval $I_n^p = (T_{p-1}, T_p)$ we have,

$$\int_{I_n^p} (\psi, e^p) dt = \int_{I_n^p} (-\dot{\phi}^p - \overline{f'(y, Y^p)}^\top \phi^p, e^p) dt = -(\phi^p, e^p)_{n,p} + (\phi^p, e^p)_{n-1,p} + \int_{I_n^p} (\mathcal{R}^p, \phi^p) dt.$$

Summing from $n = 1$ to $n = N^p$, using the continuity of y and $Y^p(T_{p-1}) = z_{p-1}$ and $\phi^p(T_p) = w_p$ we have,

$$\int_{T_{p-1}}^{T_p} (\psi, e^p) dt = (\phi^p, y - z_{p-1})(T_{p-1}) - (w_p, y - Y^p)(T_p) + \sum_{n=1}^{N^p} \int_{I_n^p} (\mathcal{R}^p, \phi^p) dt. \quad (3.12)$$

Summing (3.12) over all P subdomains,

$$\int_0^T (\psi, e^p) dt = \sum_{p=1}^P [(\phi^p, y - z_{p-1})(T_{p-1}) - (w_p, y - Y^p)(T_p) + \mathcal{D}(Y^p, \phi^p) dt], \quad (3.13)$$

where,

$$\mathcal{D}(Y^p, \phi^p) = \sum_{n=1}^{N^p} \int_{I_n^p} (\mathcal{R}^p, \phi^p). \quad (3.14)$$

As above, Y^p depends implicitly on $\mathcal{T}^p$.

Now we investigate different choices for z_{p-1} and w_p .

Analysis using the global Stage 2 adjoint problem—We assume the initial conditions, $w_p = \phi^{p+1}(T_p)$, which leads to a globally defined adjoint as in the classic analysis. From (3.13) we have,

$$\int_{T_{p-1}}^{T_p} (\psi, e^p) dt = \sum_{p=1}^P (\phi^p, y - z_{p-1})(T_{p-1}) - (\phi^{p+1}, y - Y^p)(T_p) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.15)$$

We consider the two choices, (i) $z_{p-1} = Y^{p-1}(T_{p-1})$ and (ii) $z_{p-1} = \hat{Y}^p(T_{p-1})$.

Theorem 3.3: Let $z_{p-1} = Y^{p-1}(T_{p-1})$ for $p = 2, \dots, P$ and $w_p = \phi^{p+1}(T_p)$ for $p = 1, \dots, P-1$ then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.16)$$

Proof: The proof follows directly from (3.15).

Theorem 3.4: Let $z_{p-1} = \hat{Y}^p(T_{p-1})$ for $p = 2, \dots, P$ and $w_p = \phi^{p+1}(T_p)$ for $p = 1, \dots, P-1$. Then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P (\phi^p, Y^{p-1} - \hat{Y}^p)(T_{p-1}) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.17)$$

A drawback to this error representation is that it requires a fine-scale serial solution of the adjoint problem.

Analysis using the Stage 2 adjoint problem with homogeneous initial conditions—In this choice, we set $w_p = 0$ in (3.13),

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P (\phi^p, y - z_{p-1})(T_{p-1}) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.18)$$

We again consider the two choices, (i) $z_{p-1} = Y^{p-1}(T_{p-1})$ and (ii) $z_{p-1} = \hat{Y}^p(T_{p-1})$.

Theorem 3.5: Let $w_p = 0$ for $p = 1, \dots, P-1$ and $z_{p-1} = Y^{p-1}(T_{p-1})$ for $p = 2, \dots, P$. Then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\phi^p, \hat{Y}^{p-1} - Y^{p-1})(T_{p-1}) + \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a}) \right] + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p),$$

where each $\hat{\phi}^{p,a}$ solves the auxiliary “adjoint” problem,

$$-\dot{\hat{\phi}}^{p,a} = \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}^{p,a}, \quad t \in [0, T_{p-1}), \quad (3.19)$$

with initial conditions $\hat{\phi}^{p,a}(T_{p-1}) = \phi^p(T_{p-1})$.

Proof: From (3.18) we have,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P (\phi^p, y - Y^{p-1})(T_{p-1}) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p) dt. \quad (3.20)$$

Adding and subtracting appropriate terms leads to,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\phi^p, y - \hat{Y}^{p-1})(T_{p-1}) + (\phi^p, \hat{Y}^{p-1} - Y^{p-1})(T_{p-1}) \right] + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.21)$$

The term $\sum_{p=2}^P (\phi^p, y - \hat{Y}^{p-1})(T_{p-1})$ is not computable. However, since this term involves the Stage 1 solution $\hat{Y}^{p-1}$, we can estimate this term by defining auxiliary adjoint problems. Consider the auxiliary QoI,

$$\Psi(y) = (\phi^p, y)(T_{p-1}). \quad (3.22)$$

Using the auxiliary adjoint in (3.19), definition (3.6) and Theorem 3.1, we obtain,

$$(\phi^p, y - \hat{Y}^{p-1})(T_{p-1}) = \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a}). \quad (3.23)$$

To compute all the terms in $\sum_{p=2}^P (\phi^p(T_{p-1}), y(T_{p-1}) - \hat{Y}^{p-1}(T_{p-1}))$, (3.23) is solved for $2 < p < P$ for $P-1$ corresponding auxiliary QoI's (3.22).

Combining (3.21) and (3.23) proves the theorem. $\square$

Theorem 3.6: Let $w_p = 0$ for $p = 1, \dots, P-1$ and $z_{p-1} = \hat{Y}^p(T_{p-1})$ for $p = 2, \dots, P$. Then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\phi^p, \hat{Y}^{p-1} - \hat{Y}^p)(T_{p-1}) + \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a}) \right] + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p).$$

where $\hat{\phi}^{k,a}$ is the solution of the auxiliary adjoint problem (3.19).

Proof: From (3.18) we have,

$$\begin{aligned} \int_0^T (\psi, e^p) dt &= \sum_{p=2}^P (\phi^p, y \\ &\quad - \hat{Y}^p)(T_{p-1}) \\ &\quad + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p) \\ &= \sum_{p=2}^P \left[(\phi^p, y - \hat{Y}^{p-1})(T_{p-1}) + (\phi^p, \hat{Y}^{p-1} - \hat{Y}^p)(T_{p-1}) \right] \\ &\quad + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \end{aligned} \quad (3.24)$$

Treating the term $\sum_{p=2}^P (\phi^p, y - \hat{Y}^{p-1})(T_{p-1})$ as above proves the result. $\square$

Analysis using the Stage 2 adjoint problem initialized using the Stage 1 adjoint solution—Next, we set $w_p = \hat{\phi}(T_p)$ in (3.13),

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P (\phi^p, y - z_{p-1})(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.25)$$

We again consider the two choices: (i) $z_{p-1} = Y^{p-1}(T_{p-1})$ and (ii) $z_{p-1} = \hat{Y}^p(T_{p-1})$.

Theorem 3.7: Let $w_p = \hat{\phi}(T_p)$ for $p = 1, \dots, P-1$ and $z_{p-1} = Y^{p-1}(T_{p-1})$ for $p = 2, \dots, P$. Then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\phi^p - \hat{\phi}, \hat{Y}^{p-1} - Y^{p-1})(T_{p-1}) + \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a}) \right] + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p), \quad (3.26)$$

where each $\hat{\phi}^{k,a}$ solves an auxiliary adjoint problem,

$$-\dot{\hat{\phi}}^{p,a} = \overline{f'(y, \hat{Y}^p)}^\top \hat{\phi}^{p,a}, \quad t \in [0, T_{p-1}), \quad (3.27)$$

with initial conditions $\hat{\phi}^{p,a}(T_{p-1}) = \phi^p(T_{p-1}) - \hat{\phi}(T_{p-1})$.

Proof: We set $z_{p-1} = Y^{p-1}(T_{p-1})$ in (3.25) and use initial conditions to arrive at,

$$\int_0^T (\psi, e^p) dt = \sum_{p=1}^P \left[(\phi^p, y - Y^{p-1})(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) + \mathcal{D}(Y^p, \phi^p) \right], = \sum_{p=2}^P (\phi^p - \hat{\phi}, y - Y^{p-1})(T_{p-1}) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.28)$$

Adding and subtracting appropriate terms leads to,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P (\phi^p - \hat{\phi}, y - \hat{Y}^{p-1})(T_{p-1}) + \sum_{p=2}^P (\phi^p - \hat{\phi}, \hat{Y}^{p-1} - Y^{p-1})(T_{p-1}) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.29)$$

The term $\sum_{p=2}^P (\phi^p - \hat{\phi}, y - \hat{Y}^{p-1})(T_{p-1})$ is not computable. We define $P-1$ auxiliary QoIs as,

$$\Psi(y) = (\phi^p - \hat{\phi}, y)(T_{p-1}). \quad (3.30)$$

Employing the adjoint (3.27) and Theorem 3.1 gives the following error representation,

$$(\phi^p - \hat{\phi}, y - \hat{Y}^{p-1})(T_{p-1}) = \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a}). \quad (3.31)$$

Combining (3.29) and (3.31) proves the result $\square$

Theorem 3.8: Let $w_p = \hat{\phi}(T_p)$ for $p = 1, \dots, P-1$ and $z_{p-1} = \hat{Y}^p(T_{p-1})$ for $p = 2, \dots, P$. Then,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[(\hat{\phi}, Y^{p-1} - \hat{Y}^p)(T_{p-1}) + \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a})(\phi^p - \hat{\phi}, \hat{Y}^{p-1} - \hat{Y}^p)(T_{p-1}) \right] + \sum_{p=1}^P \hat{\mathcal{D}}(Y^p, \phi^p) \quad (3.32)$$

where $\hat{\phi}^{k,a}$ is the solution to the auxiliary adjoint problem in (3.27).

Proof: We set $z_{p-1} = \hat{Y}^p(T_{p-1})$ in (3.25) and get,

$$\int_0^T (\psi, e^p) dt = \sum_{p=1}^P (\phi^p, y - \hat{Y}^p)(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) + \sum_{p=1}^P \mathcal{D}(Y^p, \phi^p). \quad (3.33)$$

Considering the first sum in (3.33), applying boundary conditions and adding and subtracting appropriate terms, we arrive at,

$$\begin{aligned} & \sum_{p=1}^P \left[(\phi^p, y - \hat{Y}^p)(T_{p-1}) - (\hat{\phi}, y - Y^p)(T_p) \right] \\ &= \sum_{p=2}^P \left[(\phi^p - \hat{\phi}, y - \hat{Y}^p) + (\hat{\phi}, Y^{p-1} - \hat{Y}^p) \right] (T_{p-1}) \\ &= \sum_{p=2}^P \left[(\phi^p - \hat{\phi}, y - \hat{Y}^{p-1}) + (\phi^p - \hat{\phi}, \hat{Y}^{p-1} - \hat{Y}^p) + (\hat{\phi}, Y^{p-1} - \hat{Y}^p) \right] (T_{p-1}). \end{aligned} \quad (3.34)$$

The term $\sum_{p=2}^P (\phi^p - \hat{\phi}, y - \hat{Y}^{p-1})(T_{p-1})$ is treated as in Theorem 3.7 to arrive at the representation (3.31). Combining (3.33), (3.34) and (3.31) completes the proof. $\square$

Theorems 3.5–3.8 involving solving a number of auxiliary adjoint problems. However, these auxiliary adjoint problems are solved on the Stage 1 discretization which is much coarser than the Stage 2 discretization, and are solved independently in parallel. Hence, this approach is computationally feasible. Moreover, analogously to Corollary 3.1, if

$\hat{z}_{p-1} = \hat{Y}^p(T_{p-1}) = \hat{Y}^{p-1}(T_{p-1})$, then the error representation for Theorems 3.5–3.8 simplify as the term $\sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1})$ drops out.

3.3. Extension to the discontinuous Galerkin method

The two most common finite element time discretizations are the cG(q) family described above and the discontinuous Galerkin (dG(q)) family [12, 13, 18]. The dG(q) method for Stage 1 is to find $\hat{Y}^p \in \hat{\mathcal{V}}_{\hat{n}}^{p,q}$ such that,

$$([\hat{Y}^p]_{n-1,p}, v^+) + \int_{t_{n-1}^p}^{t_n^p} (\dot{\hat{Y}}^p, v) dt = \int_{t_{n-1}^p}^{t_n^p} (f(\hat{Y}^p, t), v) dt, \quad \forall v \in \hat{\mathcal{V}}_{\hat{n}}^{p,q}, \quad (3.35)$$

for $1 \leq \hat{n} \leq \hat{N}^p$, where $[U]_{n,p}$ denotes the jump across $t_{n,p}$ by $[U]_{n,p} = U_{n,p}^+ - U_{n,p}^-$, where $U_{n,p}^\pm = \lim_{t \rightarrow t_{n,p}^\pm} U(t)$. The extension of the analysis to dG(q) involves modifications in the definitions of the terms $\hat{\mathcal{D}}$ and $\mathcal{D}$ in (3.6) and (3.14) respectively. The resulting modifications are,

$$\hat{\mathcal{D}}(\hat{Y}^p, \hat{\phi}) = \sum_{\hat{n}=1}^{\hat{N}^p} -([\hat{Y}^p]_{\hat{n}-1,p}, \hat{\phi}_{\hat{n}-1,p}) + \int_{t_{\hat{n}}^p} (-\dot{\hat{Y}}^p + f(\hat{Y}^p), \hat{\phi}) dt,$$

and

$$\mathcal{D}(Y^p, \phi^p) = \sum_{n=1}^{N^p} -([Y^p]_{n-1,p}, \phi_{n-1,p}^p) + \int_{t_n^p} (-\dot{Y}^p + f(Y^p), \phi^p) dt.$$

4. Applications of the estimates

We present three applications of the a posteriori analysis for two stage computations.

4.1. Standard “dual-weighted” a posteriori error estimation

We estimate the error in the Stage 2 cG(1) approximation, Y^p , for the nonlinear test problem,

$$\begin{cases} \dot{y} = -0.25 - \sin(\pi t)y^2, & 0 < t < T, \\ y(0) = 1.0, \end{cases} \quad (4.1)$$

where solution is computed serially, that is, $z_{p-1} = Y^{p-1}(T_{p-1})$. This problem has varying stability regimes yet has a computable exact solution so that the accuracy of estimates is easily evaluated.

We present results for the error in the stage two solution using four different estimates based on Theorems 3.2, 3.3 and 3.7. The first estimate, “Stage 1 adjoint”, uses the (continuous) Stage 1 adjoint and we estimate the error using (3.10) of Theorem 3.2, dropping the non-computable term. The second estimate, “Stage 2 classical”, uses the continuous stage 2 adjoint and we estimate the error using equation (3.16) of Theorem 3.3. The third and fourth

estimates “Stage 2 Parallel” and “Stage 2 Parallel no Auxiliary Terms” are both based on equation (3.26) of Theorem 3.7 which uses the Stage 2 adjoint computed in parallel on each of the p subdomains initialized with Stage 1 adjoint solution values. The fourth estimate

drops the auxiliary terms $\sum_{k=2}^{p-1}(\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \hat{\mathcal{D}}(\hat{Y}^k, \hat{\phi}^{k,a})$, hence a comparison of the third and fourth estimates highlights the importance of these auxiliary terms.

For the numerical solutions, we choose $T = 3$, the Stage 1 time step $\Delta t = 0.1$, the Stage 2 time step $\tau = 0.00625$ and $P = 10$. The adjoint solutions are numerically estimated by the cG(1) method with time steps that are four times smaller than the time steps for the Stage 1 and Stage 2 solutions throughout the paper. The QoI is specified by choosing $\psi(t) = 1$. The results are shown in Table 4.1, in which we also present,

$$\text{Effectivity ratio} = \frac{\text{Estimated error}}{\text{True error}}.$$

The results indicate that the “Stage 1 Adjoint” estimate is quite inaccurate. The schemes “Stage 2 Classical” and “Stage 2 Parallel” are quite accurate, as expected. The scheme “Stage 2 Parallel No Auxiliary Terms” is not as accurate as the last two schemes, but is still reasonably accurate with an effectivity ratio of 1.09.

We further investigate the effect of the auxiliary terms by using the implicit Euler method for the primal solutions $\hat{Y}^p$ and Y^p , and keeping the other parameters the same. The results are shown in Table 4.2, where we now see that “Stage 2 Parallel No Auxiliary Term” is quite inaccurate when compared to the “Stage 2 Parallel” estimator which still yields a good error estimate.

4.2. Construction of efficient discretizations

We discuss strategies for constructing efficient discretizations for the Stage 2 solution, based on the accurate a posteriori error estimate computed for the Stage 1 solution. These strategies are “efficient” in the sense of exploiting the cancellation of local contributions to the error.

In our experience, cancellation of local contributions is generic outside of linear unstable and “blow-up” problems. Even chaotic problems do not exhibit uniform growth of numerical error as this would eliminate the other properties, e.g. “mixing”, needed for chaotic behavior [24]. Thus, computing *accurate* error estimates generally requires accounting for the effects of cancellation. But, exploiting cancellation of error for adaptive error control is a difficult problem.

Standard error analyses generally yield an expression of the form,

$$\text{error} \approx \sum_{n=1}^N \text{local error contributions}_n, \quad (4.2)$$

where N is the number of time steps and the form of the local contributions depends on the analysis and method [31, 32, 16, 14, 11, 10, 4, 33, 34, 42]. For standard adaptive error control strategies, this is replaced by the bound,

$$|\text{error}| \leq \sum_{n=1}^N |\text{local error contributions}_n|, \quad (4.3)$$

or some variation obtained with additional estimation of the local contributions. There is no cancellation of contributions among the absolute local contributions in (4.3). Then a model of computational “work” per step is introduced and an optimization problem for the optimal sequence of time steps is derived. The solution to the optimization problem leads to the conclusion that the absolute local error contributions should be equal. In some literature, this is called the *Principle of Equidistribution*.

Such strategies are optimally efficient as far as controlling the error bound (4.3), but are not optimal in terms of controlling the actual error (4.2). In the common situation in which the signs of the local contributions vary, so that there is significant cancellation among the local contributions, the error bound (4.3) is typically much larger than the error estimate (4.2). In this case, stepsizes chosen by standard adaptive time step controls are far too large. This is reflected in the general difficulty in choosing a suitable local error tolerance for a standard adaptive error control scheme in order to achieve a desired accuracy in the result.

To illustrate, we solve (4.1) using the implicit Euler method with a time step of 0.01 on $[0, 3]$. The QoI is taken to be the weighted average value of the solution by setting $\psi = 1$ in (1.2) and the error estimate for the QoI is -0.0023 . The local error contributions obtained from the Stage 1 a posteriori estimate are plotted in Figure 4.1. The figure shows a period of positive contributions followed by a period of negative contributions and finally another period of positive contributions. This leads to significant cancellation of error contributions over the entire time period of solution. If we refine intervals 11 to 20, corresponding to the time period $[0.1, 0.2]$, by a factor of 10, then the total error increases in magnitude to -0.0034 ! The reason is that the mesh refinement leads to a significant decrease in size of the positive error contributions, leading to a reduced cancellation of contributions overall.

To describe the cancellation of error contributions, we define the accumulated contributions to the error at end of interval k as,

$$\text{Accumulated contributions}_k = \left| \sum_{n=1}^k \text{local error contributions}_n \right|, \quad (4.4)$$

for $k = 1, \dots, N$. The accumulated contributions corresponding to Figure 4.1 are plotted in Figure 4.1. This illustrates the effects of cancellation of error contributions.

We indicated that *localized* time step refinement in a region over which significant cancellation of local error contributions occurs can lead to an increase in error. This appears

to be a generic phenomena. Experimentation also suggests that employing *uniform* refinement over a region with significant cancellation of local error contributions preserves cancellation. For this reason, we propose an adaptive strategy that employs uniform refinement over time periods. The problem becomes identifying suitable time periods over which significant cancellation of local error contributions occur. We call such time periods “mesoscale regions” since they have size significantly larger than individual time steps but potentially much smaller than the entire time period of solution.

We emphasize that the proposed approach is an extension of classic adaptive error control strategies applied to regions of time steps rather than used on a stepwise basis. The proposed approach can be easily implemented in any code that computes “dual-weighted” a posteriori error estimates.

We assume that the Stage 1 solution is solved serially, that is, the Stage 1 solution in subdomain p gets its initial condition from the Stage 1 solution at the end of subdomain $p - 1$. For simplicity, we relabel $\hat{N} = \sum_{p=1}^P \hat{N}^p$, $\hat{I}_{\hat{k}} = [\hat{t}_{k-1}, \hat{t}_k] = \hat{I}_{\hat{n}}^p$ if $\hat{t}_{k-1} = \hat{t}_{\hat{n}-1}^p$ and $\hat{t}_k = \hat{t}_{\hat{n}}^p$ for some $\hat{n}$ and p , $\hat{Y}(t) = \hat{Y}^p(t)$ if $t \in [T_{p-1}, T_p]$ as illustrated in Figure 4.2.

We rewrite the Stage 1 error representation (3.8) as,

$$\int_0^T (\psi, \hat{e}) dt = \sum_{\hat{k}=1}^{\hat{N}} \int_{\hat{I}_{\hat{k}}} (-\dot{\hat{Y}} + f(\hat{Y}), \hat{\phi}) dt, \quad (4.5)$$

where $\hat{e} = y - \hat{Y}$, as illustrated in Figure 4.2. That is, we have now expressed (4.5) as the standard error representation for a global cG(q) solution. We define the contribution of the interval $\hat{I}_{\hat{k}}$ to the error as $\hat{E}_{\hat{k}} = \int_{\hat{I}_{\hat{k}}} (-\dot{\hat{Y}} + f(\hat{Y}), \hat{\phi}) dt$. Then (4.5) reads,

$$\int_0^T (\psi, \hat{e}) dt = \sum_{\hat{k}=1}^{\hat{N}} \hat{E}_{\hat{k}}. \quad (4.6)$$

4.2.1. A “mesoscale region” approach to grouping time steps—We define mesoscale regions as time intervals of “maximal” cancellation of local error contributions. We perform uniform refinement within each mesoscale region in order to preserve cancellation. We then seek to balance the contributions of the different mesoscale regions in order to obtain the desired accuracy in the error in the QoI.

Using a computed a posteriori error estimate, we compute accumulated contributions to the error as,

$$A\hat{E}_{\hat{k}} = \left| \sum_{\hat{i}=1}^{\hat{k}} \hat{E}_{\hat{i}} \right|. \quad (4.7)$$

This is easily obtained after the estimate has been computed.

We illustrate the accumulated contributions for the harmonic oscillator,

$$\frac{d^2 w}{dt^2} = -\frac{k}{m}w - \frac{c}{m}\frac{dw}{dt} + \frac{F_0}{m}\cos(\gamma t + \theta_d), \quad (4.8)$$

with $k = 50$, $m = 0.25$, $c = 1$, $F_0 = 50$, $\theta_d = 0$, $\gamma = 10$. The solution is shown in Figure 4.3.

The second-order ODE is converted to a first-order ODE by setting $y_1 = w$, $y_2 = \dot{y}_1$ and initial conditions $y(0) = [y_1(0) \ y_2(0)]^T = [5 \ 0]^T$. The QoI is specified by choosing $\psi(t) = [1 \ 0]$, that is, the average value of the first component. The accumulated contributions for the Stage 1 solution using the implicit Euler method with a time step of 0.1 is shown in Figure 4.3.

The mesoscale regions are computed by first determining the global minimum of the accumulated contributions after any initial accumulation of error. The process is then repeated from the end of this mesoscale region, resulting in Algorithm 1. The mesoscale regions for the harmonic oscillator example are shown by the vertical dotted lines in Figure 4.3.

Algorithm 1

Mesoscale Region Definition Algorithm

$\hat{N}$ = total number of intervals

Compute the accumulated contributions, $A\hat{E}_{\hat{k}} = \left| \sum_{\hat{i}=1}^{\hat{k}} \hat{E}_{\hat{i}} \right|$ for $\hat{k} = 1, \dots, \hat{N}$

Set $\hat{i} = 1$, $start = 1$ and $end = \hat{N}$

while $start$ is not equal to end

 Find the end of the initial increase in error $P_{\hat{i}}$ from $start$

 Find the global minimum interval $M_{\hat{i}}$ between $P_{\hat{i}}$ and end

 Set $start = M_{\hat{i}}$

 Increment $\hat{i}$

4.2.2. Algorithms for efficient discretizations based on mesoscale regions—

We describe two algorithms that attempt to construct Stage 2 discretization efficiently using piecewise uniform refinement over mesoscale regions. Both approaches are based on balancing the sum of the absolute contributions of the mesoscale regions to the error using an argument similar to the classic Principle of Equidistribution. We emphasize the fact that both algorithms involve applying standard error control concepts to groups of timesteps instead of a single timestep. They are easily implemented in existing codes that compute “dual-weighted” estimates and there is little additional overhead associated with either

algorithm as compared to standard adaptive error control. Moreover, if there is no significant cancellation of local contributions then the algorithms reduce to standard residual/local error control while if the local contributions oscillate frequently between adjacent time steps, then they default to uniform refinement.

The algorithms attempt to solve different adaptive problems:

Allocate Fixed Resources. Determine the optimal distribution of fixed computational resources across mesoscale regions in order to minimize the error estimate.

Achieve a Given Tolerance. Determine the minimal number of timesteps for each mesoscale region that results in an error estimate that is smaller than a specified tolerance.

Allocation of fixed resources—We describe an algorithm based on deriving the optimal distribution of time steps across mesoscale regions to minimize the absolute contributions to the error from the mesoscale regions.

Let R denote the number of mesoscale regions, $\hat{N}_i$ (resp. N_i) the number of time steps for Stage 1 (resp. Stage 2) within the i^{th} mesoscale region, and $\hat{\mathcal{E}}_i$ (resp. $\mathcal{E}_i$) the error contribution for Stage 1 (resp. Stage 2) of the i^{th} mesoscale region. That is, assume that the mesoscale region i starts at interval $\hat{q}$ and ends at interval $\hat{r}$, then $\hat{\mathcal{E}}_i = A\hat{E}_{\hat{r}} - A\hat{E}_{\hat{q}}$. Note that we have $N_i = l_i h_i^{-1}$, where h_i is the length of the time steps in region i and l_i is the length of the region. We assume that the absolute error contributions for the Stage 1 and Stage 2 solutions within each mesoscale region, $\hat{\mathcal{E}}_i$ and $\mathcal{E}_i$ satisfy,

$$\hat{\mathcal{E}}_i = \frac{c_i}{\hat{N}_i^{q_i}}, \quad \mathcal{E}_i = \frac{c_i}{N_i^{q_i}}, \quad (4.9)$$

where c_i and q_i are some positive constants. The exponents q_i are determined by the order of the method. Then we apply the Principle of Equidistribution to obtain:

Theorem 4.1 (Fixed resources allocation): If the number of total intervals N is fixed, $q_i = q$ for all i and the error in each mesoscale regions satisfies (4.9), then the total error is minimized if,

$$\frac{c_i}{N_i^{q+1}} = \text{constant}, \quad \text{for all } i. \quad (4.10)$$

Proof: Since the number of total intervals is fixed, we have the constraint,

$$\sum_{i=1}^R N_i = N. \quad (4.11)$$

We minimize the total error, given by (4.9) subject to the constraint (4.11). To accomplish this we define the Lagrangian as

$$L(N_1, N_2, \dots, N_R, \lambda) = \sum_{i=1}^R \frac{c_i}{N_i^{q_i}} + \lambda \left(\sum_{i=1}^R N_i - N \right). \quad (4.12)$$

Setting the first variation of L with respect to N_i , $1 \leq i \leq R$ to 0 yields,

$$\frac{q_i c_i}{N_i^{q_i+1}} = \text{constant}. \quad (4.13)$$

Thus, the error is minimized, when (4.13) is (approximately) satisfied. Now, since $q_i = q$ for all i , we have the optimality condition (4.10). $\square$

During implementation, the constants c_i are approximated by using the Stage 1 error estimate (4.9) and the relation $c_i = \hat{N}_i^{q_i} \hat{\mathcal{E}}_i$, where $\hat{N}_i$ are the number of Stage 1 time steps for mesoscale region i .

Numerical example: We show the effect of applying the fixed resources allocation strategy in Theorem 4.1 for the harmonic oscillator in (4.8). Both Stage 1 and Stage 2 solutions are computed serially by employing the implicit Euler method on $[0, 3]$. The Stage 1 solution is computed with a time step of 0.01 and the error is estimated to be $-6.5E-3$. The Stage 1 mesh has $\hat{N} = 300$ intervals. We choose $N = 3000$ for Stage 2 number of intervals. Theorem 4.1 is then applied to calculate the new time step for each mesoscale region, with the additional constraint that we do not coarsen the mesh to ensure reliability. The results using allocation strategy in Theorem 4.1 and uniform refinement (with 3000 intervals) are shown in Table 4.3. The allocation strategy in Theorem 4.1 leads to a much lower error than uniform refinement.

Achieve a given tolerance—In this case, we assume that the goal is to insure that the

Stage 2 error estimate is smaller than a given tolerance $TOL > 0$, i.e. so $\sum_{i=1}^R \mathcal{E}_i < TOL$.

Note that by definition $\mathcal{E}_i \geq 0$. For Stage 1 errors, we assume that the total error exceeds the tolerance (otherwise, we are done, and there is no need for Stage 2). That is, we have

$\sum_{i=1}^R \hat{\mathcal{E}}_i > TOL$. Let w_i be reduction factors associated with mesoscale region i , $0 < w_i \leq 1$.

Then we want $\sum_{i=1}^R w_i \hat{\mathcal{E}}_i \leq TOL$. Based on this, the discretization for Stage 2 mesh is computed. For example, if the order of the method employed in both stages is q_i , then the relationship between Stage 1 and Stage 2 number of time steps for a mesoscale region i is

$$w_i = \frac{\mathcal{E}_i}{\hat{\mathcal{E}}_i} = \left(\frac{\hat{N}_i}{N_i} \right)^{q_i}, \quad (4.14)$$

from which the value of N_i and hence the time step size of mesoscale region i , is computed.

There are many strategies to compute the factors w_i . Here we outline three strategies.

1. Choose $w_i = \frac{TOL}{\sum_{i=1}^R \hat{\mathcal{E}}_i}$, corresponding to uniform refinement.
2. Choose $w_i = \frac{TOL}{R \hat{\mathcal{E}}_i}$. This strategy, called “Mesoscale Simple”, is often (but not always) better than uniform refinement.
3. Choose w_i by solving a constrained minimization problem. This strategy, called “Mesoscale Optimized”, minimizes the objective function, $\sum_{i=1}^R N_i$, subject to the constraints that $0 < w_i \leq 1$ and $\sum_{i=1}^R w_i \hat{\mathcal{E}}_i \leq TOL$. The objective function, which is the number of intervals in the Stage 2 mesh, is formed in terms of the reduction factors w_i by employing (4.14).

Numerical example: We show the effect of mesoscale refinement for the harmonic oscillator (4.8). Both Stage 1 and Stage 2 solutions are computed serially by employing the implicit Euler method on $[0, 3]$. The Stage 1 solution is computed with a time step of 0.01 and the error is estimated to be $-6.5E-3$. The tolerance, TOL , for the Stage 2 solution is set to be $6.5E-4$. The Stage 1 mesh has 300 intervals. If we use uniform refinement to meet the tolerance, we have to set the time step for Stage 2 to be 10 times smaller. The results using the two mesoscale refinement strategies and uniform refinement are shown in Table 4.4. The Matlab function `fmincon` was used for solving the constrained minimization problem. The Mesoscale Simple strategy meets the refinement goal with about a third of resources compared to uniform refinement, highlighting the potential of taking the cancellation of error into account for choosing a refinement strategy. The Mesoscale Optimized strategy performs even better, achieving a lower error with the fewest number of intervals.

The common step-wise strategies effectively consider each time step as a mesoscale region and apply the simple and optimized strategies step by step. Such approaches ignore the cancellation of error and this results in a much larger optimization problem. For the harmonic oscillator problem, the step-wise simple strategy results in 45722 steps and a Stage 2 error of $-8.51E-06$. The step-wise optimized strategy applied step by step results in 40986 steps and a Stage 2 error of $-2.15E-05$. Moreover, the optimization strategy required 536572 objective function evaluations, as compared to 135 objective function evaluations for the Mesoscale Optimized strategy.

4.3. A posteriori analysis of the Parareal Algorithm

The last application is to the Parareal Algorithm, which is a variation of the two stage computation. In particular, it computes corrections for the Stage 1 solution using the more accurate Stage 2 solution, and then iterates for a finite number of iterations. This is analogous to a domain decomposition approach for parallel computation. We extend the notation by the addition of a superscript (m) indicating the iteration number. Note that the

Stage 1 solution is often called the coarse scale solution, and the Stage 2 solution the fine scale solution [37, 2].

4.3.1. Alternative formulation of the Parareal Algorithm—We start by deriving a variational formulation of the Parareal algorithm given in Algorithm 2 [39, 29, 30]. The subscript p indicates values at time T_p . Once again, the intervals $[T_{p-1}, T_p]$ are significantly larger than the individual time steps used in the numerical integrators, $\hat{G}^p$ and F^p , for each scale.

Algorithm 2

Parareal Algorithm

Input: y_0 : initial condition
return: $\hat{Y}_p^{(M)}$: Coarse solution at T_p , $p = 1, \dots, P$
 $Y_p^{(M)}$: Fine solution at T_p , $p = 1, \dots, P$

$C_p^{(0)} = 0$, $p = 0, \dots, P$ {initialize corrections}

for $m = 1, 2, \dots, M$

$\hat{Y}_p^{(m)} = \hat{G}^p[\hat{Y}_{p-1}^{(m)}](T_p) + C_p^{(m-1)}$, $p = 1, \dots, P$	{serial coarse solve}
$Y_p^{(m)} = F^p[\hat{Y}_{p-1}^{(m)}](T_p)$, $p = 1, \dots, P$	{parallel fine solve}
$C_p^{(m)} = Y_p^{(m)} - \hat{G}^p[\hat{Y}_{p-1}^{(m)}](T_p)$, $p = 1, \dots, P$	{update corrections}

Note that we have re-ordered the operations slightly as compared to the presentation in [37, 2], since we are primarily interested in the fine-scale solution. The evaluation of the QoI requires the discrete solutions, $Y^{(m)}$ and $\hat{Y}^{(m)}$, to be interpreted as functions of time rather than just their values at the time nodes T_p , $p = 0, \dots, P$. We obtain the fine scale solution as a function of time by considering the solutions $F^p[\hat{Y}_{p-1}^{(m)}](t)$. In this form of the Parareal algorithm, the coarse scale solutions are problematic due to the addition of the corrections $C_p^{(m)}$. In Alg. 3 we construct a Parareal algorithm which considers corrections $\bar{C}_{p-1}^{(m-1)}$ to the initial conditions for the subdomain $[T_{p-1}, T_p]$. In Lemma 4.1 we show the equivalence of our new algorithm with the original Parareal algorithm given by Alg. 2. Solutions from the new algorithm are distinguished by a superscript p , indicating they are defined over the subdomain $[T_{p-1}, T_p]$, as opposed to a subscript p which indicates solution values defined only at the times T_p .

Lemma 4.1 (Equivalence of the Parareal algorithms): The algorithms 2 and 3 are equivalent in the sense that,

$$\hat{Y}_p^{(M)} = \hat{Y}^{p,(M)}(T_p) + \bar{C}_p^{(M-1)},$$

$$Y_p^{(M)} = Y^{p,(M)}(T_p),$$

$$C_p^{(M)} = \overline{C}_p^{(M)},$$

for $p = 1, \dots, P$.

Proof

Algorithm 3

Parareal Algorithm: Variational version

Input: y_0 : initial condition
return: $\hat{Y}^{p,(M)}(t)$: Coarse solution defined in $[T_{p-1}, T_p]$, $p = 1, \dots, P$
 $Y^{p,(M)}(t)$: Fine solution defined in $[T_{p-1}, T_p]$, $p = 1, \dots, P$

$\overline{C}_p^{(0)} = 0$, $p = 0, \dots, P$ {initialize corrections}

for $m = 1, 2, \dots, M$

$\hat{Y}^{p,(m)}(t) = \hat{G}^p[\hat{Y}^{p-1,(m)}(T_{p-1}) + \overline{C}_{p-1}^{(m-1)}](t)$, $p = 1, \dots, P$
 $Y^{p,(m)}(t) = F^p[\hat{Y}^{p,(m)}(T_{p-1})](t)$, $p = 1, \dots, P$
 $\overline{C}_p^{(m)} = Y^{p,(m)}(T_p) - \hat{G}^p[\hat{Y}^{p,(m)}(T_{p-1})](T_p)$, $p = 1, \dots, P$

} {serial coarse solve}
} {parallel fine solve}
} {update corrections}

The proof is by induction on p and m . First, the base case for the iteration. Since

$C_p^{(0)} = \overline{C}_p^{(0)} = 0$, we have the equivalence for $m = 1$. Now assume that the equivalence holds for iteration $m - 1$. To show equivalence for iteration m , we perform induction on p . For $p = 1$ we have

$$\hat{Y}_1^{(m)} = \hat{G}^p[y_0] + C_1^{(m-1)},$$

whereas,

$$\hat{Y}^{1,(m)}(T_1) = \hat{G}^p[y_0 + \overline{C}_0^{(m-1)}] = \hat{G}^p[y_0],$$

and hence $\hat{Y}_1^{(m)} = \hat{Y}^{1,(m)}(T_1) + C_1^{(m-1)}$. Further, since $Y_0^{(m)} = Y^{1,(m)}(T_0) = y_0$, we have

$Y_1^{(m)} = Y^{1,(m)}(T_1) = F^p[y_0]$. This also implies that $C_1^{(m)} = \overline{C}_1^{(m)}$. Now assume the hypothesis for p is true for $p - 1$ such that $1 < p - 1 < P$. Then,

$$\hat{Y}_p^{(m)} = \hat{G}^p[\hat{Y}_{p-1}^{(m)}] + C_p^{(m)},$$

whereas by induction hypothesis,

$$\hat{Y}^{p,(m)}(T_p) = \hat{G}^p[\hat{Y}^{p-1,(m)}(T_{p-1}) + \bar{C}_{p-1}^{(m)}] = \hat{G}^p[\hat{Y}^{p-1,(m)}(T_{p-1}) + C_{p-1}^{(m)}] = \hat{G}^p[\hat{Y}_{p-1}^{(m)}],$$

and hence $\hat{Y}_p^{(m)} = \hat{Y}^{p,(m)}(T_p) + C_p^{(m)}$. For the fine scale solutions we have,

$$Y_p^{(m)} = F^p[\hat{Y}_{p-1}^{(m)}],$$

while,

$$Y^{p,(m)}(T_p) = F^p[\hat{Y}^{p,(m)}(T_{p-1})] = F^p[\hat{Y}^{p-1,(m)}(T_{p-1}) + C_{p-1}^{(m)}] = F^p[\hat{Y}_{p-1}^{(m)}],$$

and hence $Y_p^{(m)} = Y^{p,(m)}(T_p)$. Also, since $\hat{Y}^{p,(m)}(T_{p-1}) = \hat{Y}^{p-1,(m)}(T_{p-1}) + C_{p-1}^{(m)} = \hat{Y}_{p-1}^{(m)}$, we have $C_p^{(m)} = \bar{C}_p^{(m)}$. This completes the induction for p , which in turn completes the induction for m and hence the proof. $\square$

4.3.2. A posteriori error analysis—We develop an error representation for the error in the QoI for the solution computed from the Parareal algorithm. The analysis can be carried out at any particular iteration of the Parareal algorithm for which error estimates are needed. Given the coarse and fine primal solutions $\hat{Y}^{p,(m)}$ and $Y^{p,(m)}$ at iteration m of the Parareal algorithm 3, we employ the adjoints $\hat{\phi}$, ϕ^p and $\hat{\phi}^{p,a}$ defined in (3.4), (3.11) and (3.27) respectively. This strategy ensures that the error estimates are computed in the same time complexity as that for the primal solutions $\hat{Y}^{p,(m)}$ and $Y^{p,(m)}$.

Coarse error representation

Theorem 4.2: The error in the QoI in the Stage 1 solution, $\hat{Y}^{p,(m)}$, for the Parareal algorithm 3 is given by,

$$\int_0^T (\psi, \hat{e}^{p,(m)}) dt = \sum_{p=1}^P \left[\left(-\bar{C}_{p-1}^{(m-1)}, \hat{\phi}(T_{p-1}) \right) + \hat{\mathcal{D}}(\hat{Y}^{p,(m)}, \hat{\phi}) \right],$$

where $\hat{\mathcal{D}}(\hat{Y}^{p,(m)}, \hat{\phi})$ is the discretization contribution term defined as

$$\hat{\mathcal{D}}(\hat{Y}^{p,(m)}, \hat{\phi}) = \sum_{\hat{n}=1}^{\hat{N}^p} \int_{\hat{t}_{\hat{n}}}^{\hat{t}_{\hat{n}+1}} (\hat{\mathcal{R}}^{p,m}, \hat{\phi}) dt.$$

Here $\hat{\mathcal{R}}^{p,m}$ is the residual $\hat{\mathcal{R}}^{p,m} = -\dot{Y}^{p,(m)} + f(Y^{p,(m)})$.

Proof: This theorem follows directly from Theorem 3.1 by observing that

$$\hat{z}_{p-1} = \hat{Y}^{p,(m)}(T_{p-1}) = \hat{Y}^{p-1,(m)}(T_{p-1}) + \bar{C}_{p-1}^{(m-1)}. \quad \square$$

Fine error representation

Theorem 4.3: The error in the QoI in the Stage 2 solution, Y^p , for the Parareal algorithm 3 is given by,

$$\int_0^T (\psi, e^p) dt = \sum_{p=2}^P \left[\mathcal{I}(Y^{p-1,(m)}, \hat{Y}^{p,(m)}, \hat{\phi}) + \mathcal{J}(\hat{Y}^{p,(m)}, \hat{Y}^{p-1,(m)}, \hat{\phi}, \phi^p) + \mathcal{A}(\hat{\mathcal{R}}^{p,(m)}, \hat{\phi}^{p,a}) \right] + \sum_{p=1}^P \mathcal{D}(Y^{p,(m)}, \phi^p), \quad (4.15)$$

where the four terms represent the iteration (Iter), coarse scale (Coar), auxiliary (Aux) and discretization (Disc) error contributions defined as,

$$\mathcal{I}(Y^{p,(m)}, Y^{p-1,(m)}, \hat{\phi}) = (\hat{\phi}, Y^{p-1,(m)} - Y^{p,(m)})(T_{p-1}),$$

$$\mathcal{J}(\hat{Y}^{p,(m)}, \hat{Y}^{p-1,(m)}, \hat{\phi}, \phi^p) = (\phi^p - \hat{\phi}, \hat{Y}^{p-1,(m)} - \hat{Y}^{p,(m)})(T_{p-1}),$$

$$\mathcal{A}(\hat{\mathcal{R}}^{p,(m)}, \hat{\phi}^{p,a}) = \sum_{k=2}^{p-1} (\hat{\phi}, \hat{Y}^{k-1} - \hat{Y}^k)(T_{k-1}) + \sum_{k=1}^{p-1} \sum_{\hat{n}=1}^{\hat{N}^k} \int_{t_{\hat{n}-1}^k}^{t_{\hat{n}}^k} (\hat{\mathcal{R}}^{p,(m)}, \hat{\phi}^{p,a}),$$

and

$$\mathcal{D}(Y^{p,(m)}, \phi^p) = \sum_{n=1}^{N^p} \int_{I_n^p} (\mathcal{R}^{p,(m)}, \phi^p) dt.$$

Here $\mathcal{R}^{p,m}$ is the residual $\mathcal{R}^{p,m} = -\dot{Y}^{p,(m)} + f(Y^{p,(m)})$.

Proof: The proof follows directly from Theorem 3.8. $\square$

The discretization error contribution measures the effect of choosing the Stage 2 discretization. The iteration error contribution measures the error due to insufficient convergence of the algorithm and reflects the inherent order of the time variable from past to future as mention in the introduction. The coarse scale error contribution involves the

difference of fine and coarse adjoint solutions scaled by coarse scale initialization differences at the end of subdomains.

4.3.3. Numerical examples—We consider two numerical examples.

Harmonic Oscillator: We show the accuracy of the estimate provided by Theorem 4.3 for the Stage 2 solution of the Parareal algorithm for the harmonic oscillator (4.8) for the implicit Euler, cG(1) and the Crank-Nicolson methods. The QoI is again specified by choosing $\psi(t) = [1 \ 0]$, that is, the weighted average value of the first component. We choose $P = 10$, $T = 3.0$.

Results for Implicit Euler: In this case, the numerical solvers, $\hat{G}^p$ and F^p are chosen to be implicit Euler on the Stage 1 and Stage 2 time meshes on subdomain p respectively. The initial coarse scale error with a time step of 0.01 is computed as $-6.5E-3$. We set an error tolerance of $-6.5E-4$ and employ uniform refinement and the Mesoscale Optimize refinement in § 4.2.2 to construct the Stage 2 discretization.

The uniformly refined mesh for a Stage 2 has time step of 0.001 for a total of 3000 intervals. We show the results for the error and its different contributions for Stage 2 in Table 4.5 for the first four iterations of the Parareal algorithm. The error decreases rapidly initially, but plateaus around the fourth iteration. This is explained by examining the different error contributions. Initially the iteration contribution dominates, as might be expected. By iteration 4, the iteration error is an order of magnitude smaller than the discretization error. Further iterations do not have a significant impact on decreasing the error.

The Stage 2 mesh obtained using the Mesoscale Optimize refinement has 549 intervals which is more than 5 times less than the corresponding number for uniform refinement. The results are shown in Table 4.6. The results are qualitatively similar to those in Table 4.5.

Results for cG(1): In this case, the numerical solvers, $\hat{G}^p$ and F^p are chosen to be the cG(1) method on the Stage 1 and Stage 2 time meshes on subdomain p respectively. The initial coarse scale error with a Stage 1 time step of 0.02 and a total of 150 time intervals, is computed as $7.76E-04$. We set an error tolerance of $(7.76E-04)/80$ and employ uniform refinement and the Mesoscale Optimize refinement in § 4.2.2 to construct the Stage 2 discretization.

The uniformly refined mesh for a Stage 2 has a total of 1350 intervals. We show the results for the error and its different contributions for Stage 2 in Table 4.7 for the first four iterations of the Parareal algorithm. The Stage 2 mesh obtained using the Mesoscale Optimize refinement has 524 intervals which is more than two times less than the corresponding number for uniform refinement. The results are shown in Table 4.8. The results are qualitatively similar to the ones for the implicit Euler methods in Tables 4.5 and 4.6.

Results for Crank-Nicolson: In this case, the numerical solvers, $\hat{G}^p$ and F^p are chosen to be the Crank-Nicolson method on the Stage 1 and Stage 2 time meshes on subdomain p respectively. The initial coarse scale error with a Stage 1 time step of 0.02 and a total of 150

time intervals, is computed as $1:69E-04$. We set an error tolerance of $(1:69E-04)/80$ and employ uniform refinement and the Mesoscale Optimize refinement in § 4.2.2 in order to construct the Stage 2 discretization.

The uniformly refined mesh for Stage 2 has a total of 1350 intervals. We show the results for the error and its different contributions for Stage 2 in Table 4.9 for the first four iterations of the Parareal algorithm. The Stage 2 mesh obtained using the Mesoscale Optimize refinement has 644 intervals which is approximately two times less than the corresponding number for uniform refinement. The results are shown in Table 4.10. The results are qualitatively similar to the ones for the implicit Euler methods in Tables 4.5 and 4.6.

A nonlinear partial differential equation: We consider the time solution of a space discretization of

$$\begin{cases} \dot{u} + \frac{1}{2} \cos(2\pi t) (1+u) u_x = 0, & (x, t) \in [0, 1] \times (0, T], \\ u(x, 0) = \sin(2\pi x), & x \in [0, 1], \end{cases}$$

with periodic boundary conditions. The QoI is the integral $\int_0^T \int_{0.5}^{0.6} u(x, t) dx dt$. A first order finite difference scheme with spatial discretization parameter $h = 0.05$ leads to a system of twenty ODEs. We set $T = 2.0$, $P = 10$ and choose the time step for Stage 1 to be 0.001 for a total of 2000 time intervals. The initial coarse scale error was $-3.98E-04$. We set an error tolerance of $-3.98E-05$ and compute Stage 2 meshes by employing uniform refinement and the Mesoscale Optimization refinement.

The Stage 2 mesh obtained by the uniform refinement has 20000 intervals. The results of four iterations of the Parareal algorithm are shown in Table 4.11. The iteration error contribution is significantly less than the discretization contribution at the end of iteration 2, indicating that the algorithm may be stopped at this point since further iterations will not decrease the error.

The accumulated contributions for the Stage 1 are shown in Figure 4.4, indicating significant error cancellation and potential for efficiency gains using mesoscale refinement. The Stage 2 mesh obtained by the Mesoscale Optimization refinement has 13935 intervals, which is again significantly less than uniform refinement. The results of four iterations of the Parareal algorithm are shown in Table 4.12.

5. Conclusions and future directions

We derive a posteriori error estimates for a quantity of interest for a general class of numerical methods for ordinary differential equations that employ a two stage computation. The estimates utilize different adjoint problems that present trade-offs between accuracy and computational cost. We present applications of the error estimates to dual-weighted error estimation, construction of efficient mesh selection algorithms that take into account sources of error, and for estimation of the error of the Parareal Algorithm. We note that the extension of the analysis to account for the effects of spatial discretization of partial differential equations is immediate, e.g. see [6].

A classical adjoint based error estimate requires communication across the time interval in question, hence is non-parallel. We have addressed this issue in this paper by trying various ways to localize the estimate, hence allowing it to be computed in parallel. Load balancing and parallel efficiency is an interesting area of future research, particularly for the Parareal algorithm, as it involves a tradeoff between iteration and discretization error contributions. A simple strategy for load balancing may be to divide the time intervals evenly among the processors, however, this might lead to large iteration error contribution. On the other hand, load balancing focused on utilizing the mesoscale information to reduce the iteration error contribution may increase the computation time of each iteration of the algorithm. We aim to investigate these issues in future.

Acknowledgments

J. Chaudhry's work is supported in part by the Department of Energy (DE-SC0005304, DE-SC0009279, DE-SC0009324).

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, DE-SC0009279), 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).

S. Tavenner's work is supported in part by the Department of Energy (DE-FG02-04ER25620, INL00120133) and the National Science Foundation (DMS-1016268).

References

1. Ainsworth, M., Oden, T. A posteriori error estimation in finite element analysis. John Wiley-Teubner; 2000.
2. Aubanel E. Scheduling of tasks in the parareal algorithm. *Parallel Computing*. 2011; 37:172–182.
3. Becker R, Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica*. 2001:1–102.
4. Cao Y, Petzold L. A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SISC*. 2004; 26:359–374.
5. Chaudhry J, Estep D, Ginting V, Tavenner S. A posteriori analysis of an iterative multi-discretization method for reaction-diffusion systems. *CMAME*. 2013; 267:1–22.
6. Chaudhry J, Estep D, Ginting V, Shadid John N, Tavenner Simon. A posteriori error analysis of IMEX multi-step time integration methods for advection-diffusion-reaction equations. *CMAME*. 2014; 285:730–751.
7. Chaudry JH, Estep D, Ginting V, Tavenner S. A posteriori analysis for iterative solvers for non-autonomous evolution problems. *SIAM/ASA JUQ*. 2015; 3
8. Collins J, Estep D, Tavenner S. A posteriori error analysis for finite element methods with projection operators as applied to explicit time integration techniques. *BIT*. 2014; doi: 10.1007/s10543-014-0534-9
9. Collins J, Estep D, Tavenner S. A posteriori error estimation for the Lax–Wendroff finite difference scheme. *J Comput Appl Math*. 2014; 263:299–311.
10. Connors J, Banks J, Hittinger J, Woodward C. A method to calculate numerical errors using adjoint error estimation for linear advection. *SINUM*. 2013; 51:894–926.
11. Dahlquist, G. On the control of the global error in stiff initial value problems. Springer; 1982.
12. Delfour M, Hager W, Trochu F. Discontinuous Galerkin methods for ordinary differential equations. *Math Comp*. 1981; 36:455–473.

13. Delfour MC, Dubeau F. Discontinuous polynomial approximations in the theory of one-step, hybrid and multistep methods for nonlinear ordinary differential equations. *Math Comp.* 1986; 47:169–189. S1–S8. With a supplement.
14. Dormand JR, Duckers RR, Prince PJ. Global error estimation with Runge-Kutta methods. *IMAJNA.* 1984; 4:169–184.
15. Emmett M, Minion M. Toward an efficient parallel in time method for partial differential equations. *Comm App Math Comput Sci.* 2012; 7:105–132.
16. Enright WH. Analysis of error control strategies for continuous Runge-Kutta methods. *SINUM.* 1989; 26:588–599.
17. Eriksson K, Johnson C, Logg A. Explicit time-stepping for stiff ODEs. *SISC.* 2004; 25:1142–1157.
18. Estep D. A posteriori error bounds and global error control for approximations of ordinary differential equations. *SINUM.* 1995; 32:1–48.
19. Estep D. A Short Course on Duality. Adjoint Operators, Green's Functions, and A Posteriori Error Analysis. 2004 Unpublished.
20. Estep, D. Error estimates for multiscale operator decomposition for multiphysics models. In: Fish, J., editor. *Multiscale methods: bridging the scales in science and engineering.* Oxford University Press; USA: 2009.
21. Estep D, French D. Global error control for the continuous galerkin finite element method for ordinary differential equations. *MMAN.* 1994; 28:815–852.
22. Estep D, Ginting V, Tavener S. A posteriori analysis of a multirate numerical method for ordinary differential equations. *Computer Methods in Applied Mechanics and Engineering.* 2012; 223:10–27.
23. Estep, Donald, Holst, Michael, Mikulencak, Duane. Accounting for stability: a posteriori error estimates based on residuals and variational analysis. *Comm Num Meth Engring.* 2002; 18:15–30.
24. Estep D, Johnson C. The pointwise computability of the Lorenz system. *Math Models Met Appl Sci.* 1998; 8:1277–1305.
25. Estep D, Larson M, Williams R. Estimating the error of numerical solutions of systems of reaction-diffusion equations. *AMS Memoirs.* 2000; 146
26. Estep D, Målqvist A, Tavener S. Nonparametric density estimation for elliptic problems with random perturbations i: Computational method, a posteriori analysis, and adaptive error control. *SISC.* 2009; 31:2935–2959.
27. Estep D. Nonparametric density estimation for elliptic problems with random perturbations ii: Applications and adaptive modeling. *Int J Numer Methods Engrg.* 2009; 80:846–867.
28. Falgout RD, Friedhoff S, Kolev TzV, MacLachlan SP, Schroder JB. Parallel time integration with multigrid. *LLNL-JRNL-645325.* 2014
29. Gander, M., Hairer, E. *Domain Decomposition Methods in Science and Engineering XVII.* Vol. 60. Springer; Berlin Heidelberg: 2008. Nonlinear convergence analysis for the parareal algorithm; p. 45-56. *Lecture Notes in Computational Science and Engineering*
30. Gander M, Vandewalle S. Analysis of the parareal time-parallel time-integration method. *SISC.* 2007; 29:556–578.
31. Gear, CW. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall; Englewood Cliffs, New Jersey: 1971.
32. Higham D. Global error versus tolerance for explicit Runge-Kutta methods. *IMJNA.* 1991; 11:457–480.
33. Kulikov G, Weiner R. Variable-stepsize interpolating explicit parallel peer methods with inherent global error control. *SISC.* 2010; 32:1695–1723.
34. Lang J, Verwer J. On global error estimation and control for initial value problems. *SISC.* 2007; 29:1460–1475.
35. Lions JL, Maday Y, Turinici G. Résolution d'edp par un schéma en temps pararéel. *CR Acad Sci.* 2001:661–668. *Paris Sér I math*
36. Logg A. Multi-adaptive time integration. *Appl Numer Math.* 2004; 48:339–354.
37. Maday Y. The parareal in time algorithm. 2008

38. Maday Y, Salomon J, Turinici G. Monotonic parareal control for quantum systems. *SINUM*. 2007; 45:2468–2482.
39. Maday Y, Turinici G. A parareal in time procedure for the control of partial differential equations. *Comp Ren Math*. 2002; 335:387–392.
40. Marchuk, GI., Agoshkov, VI., Shutyaev, VP. Adjoint equations and perturbation algorithms in nonlinear problems. CRC Press; Boca Raton, FL: 1996.
41. Minion M. A hybrid parareal spectral deferred corrections method. *Comm Appl Math Comput Sci*. 2011; 5:265–301.
42. Shampine L. Error estimation and control for ODEs. *J Sci Comput*. 2005; 25:3–16.

Fig. 2.1.

Top: Subdomain $[T_{p-1}, T_p]$. Middle: Stage 1 discretization for $[T_{p-1}, T_p]$. Bottom: Stage 2 discretization for $[T_{p-1}, T_p]$

Fig. 4.1.

Left: Local Error Contributions. Right: Accumulated Contributions to the Error. Note how the positive and negative contributions cancel each other to give the total error.

Fig. 4.2.

Upper: Labeling used in §2. Lower: Relabeling for §4.2.

Fig. 4.3.

Left: Solution of (4.8). Right: Accumulated Contributions and mesoscale regions.

Fig. 4.4.
Right: Accumulated Contributions for the Stage 1 solution.

Table 4.1

Comparison of error estimates obtained by different strategies for the Stage 2 primal solution computed using the cG(1) method. The true error is 1.90E-06.

Estimator	Estimated Error	Effectivity Ratio
Stage 1 Adjoint	4.12E-06	2.16
Stage 2 Classical	1.89E-06	0.99
Stage 2 Parallel	1.89 E-06	0.99
Stage 2 Parallel No Auxiliary Terms	2.08E-06	1.09

Table 4.2

Comparison of error estimates obtained by different strategies using the implicit Euler method for the primal solutions. The true error is -0.0014 .

Estimator	Estimated Error	Effectivity Ratio
Stage 2 Parallel	-0.0015	1.07
Stage 2 Parallel No Auxiliary Terms	-0.0031	2.21

Table 4.3

Comparison of allocating resources using the allocation strategy in Theorem 4.1 for mesoscale regions vs. uniform refinement

Refinement	Stage 2 Error
Uniform	-7.04E-04
Theorem 4.1	-1.75E-05

Table 4.4

Comparison of uniform and mesoscale refinement.

Refinement	Stage 2 Error	Number of intervals
Uniform	$-7.04\text{E}-04$	3000
Mesoscale Simple	$2.17\text{E}-04$	874
Mesoscale Optimized	$8.76\text{E}-05$	548

Table 4.5

Error contributions for Parareal algorithm for a uniformly refined Stage 2 mesh and employing the implicit Euler method for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	2.70E-03	1.00	9.78E-06	2.69E-03	0.00E+00	8.56E-08
2	4.70E-05	1.00	9.80E-06	3.72E-05	5.69E-07	-6.00E-07
3	9.75E-06	1.00	9.79E-06	-3.66E-08	5.66E-07	-5.68E-07
4	9.76E-06	1.00	9.79E-06	-3.81E-08	5.67E-07	-5.66E-07

Table 4.6

Error contributions for Parareal algorithm for a Stage 2 mesh obtained by using Mesoscale Optimize refinement and employing the implicit Euler method for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	-0.0034	1.00	-3.75E-04	-3.0E-03	0	9.40E-07
2	4.29E-04	0.99	5.01E-05	3.79E-04	-1.09E-06	8.61E-07
3	1.034E-04	0.97	8.42E-05	1.93E-05	-1.06E-06	8.75E-07
4	8.44E-05	0.96	8.46E-05	3.39E-09	-1.05E-06	8.75E-07

Table 4.7

Error contributions for Parareal algorithm for a uniformly refined Stage 2 mesh and employing the cG(1) method for the problem in 4.3.3. Here Error Est. is given in § (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	2.69E-03	1.00	1.62E-05	2.67E-03	0.00E+00	8.49E-08
2	5.28E-05	1.00	1.62E-05	3.66E-05	5.60E-07	-5.90E-07
3	1.61E-05	0.99	1.62E-05	-3.52E-08	5.57E-07	-5.58E-07
4	1.61E-05	0.99	1.62E-05	-3.69E-08	5.58E-07	-5.57E-07

Table 4.8

Error contributions for Parareal algorithm for a Stage 2 mesh obtained by using Mesoscale Optimize refinement and employing the cG(1) method for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	2.25E-03	1.00	7.27E-06	2.25E-03	0.00E+00	-1.42E-07
2	2.90E-05	1.00	1.01E-05	1.89E-05	5.04E-07	-5.15E-07
3	1.01E-05	0.99	1.00E-05	5.36E-08	4.99E-07	-4.99E-07
4	1.00E-05	0.99	1.00E-05	-1.73E-08	5.00E-07	-4.99E-07

Table 4.9

Error contributions for Parareal algorithm for a uniformly refined Stage 2 mesh and employing the Crank-Nicolson method for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	2.30E-03	1.00	2.27E-06	2.30E-03	0.00E+00	3.91E-07
2	4.01E-05	1.00	2.36E-06	3.78E-05	2.90E-07	-3.23E-07
3	2.25E-06	1.00	2.35E-06	-1.03E-07	2.92E-07	-2.90E-07
4	2.31E-06	1.00	2.35E-06	-3.81E-08	2.93E-07	-2.89E-07

Table 4.10

Error contributions for Parareal algorithm for a Stage 2 mesh obtained by using Mesoscale Optimize refinement and employing the Crank-Nicolson method for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	1.91E-03	1.00	9.91E-07	1.90E-03	0.00E+00	2.52E-07
2	2.08E-05	1.00	2.71E-06	1.78E-05	3.81E-07	-9.44E-08
3	2.94E-06	0.99	2.68E-06	-4.82E-08	3.78E-07	-7.77E-08
4	2.96E-06	0.99	2.68E-06	-1.96E-08	3.79E-07	-7.70E-08

Table 4.11

Error contributions for Parareal algorithm for a Stage 2 mesh obtained by using uniform refinement for the problem in § 4.3.3. Here Error Est. is given in (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	-3.89E-04	1.00	-3.97E-05	-3.50E-04	0	5.42E-07
2	-4.16E-05	1.00	-3.99E-05	-1.75E-06	-1.31E-07	1.95E-07
3	-3.98E-05	1.00	-3.99E-05	8.94E-09	-1.35E-07	1.91E-07
4	-3.99E-05	1.00	-3.99E-05	1.10E-10	-1.35E-07	1.91E-07

Table 4.12

Error contributions for Parareal algorithm for a Stage 2 mesh obtained by using Mesoscale Optimize refinement for the problem in 4.3.3. Here Error Est. is given in § (4.15), Disc, Iter, Coar and Aux are the various contributions to the error by discretization, iteration, coarse scale and auxiliary terms, see (4.15) and §3.3.

m	Error Est.	Effec. Rat.	Disc	Iter	Coar	Aux
1	-3.72E-04	0.99	-3.77E-05	-3.35E-04	0	7.85E-07
2	-3.76E-05	0.99	-3.78E-05	2.56E-07	-3.06E-07	2.89E-07
3	-3.78E-05	0.99	-3.78E-05	4.39E-09	-3.07E-07	2.89E-07
4	-3.78E-05	0.99	-3.78E-05	9.28E-13	-3.07E-07	2.89E-07