ACCOUNTING FOR STABILITY: *A POSTERIORI* ERROR ESTIMATES BASED ON RESIDUALS AND VARIATIONAL ANALYSIS

DONALD ESTEP *, MICHAEL HOLST †, AND DUANE MIKULENCAK ‡

Abstract. *A posteriori* error estimates have had a major impact on adaptive error control for the finite element method. In this paper, we review a relatively new approach to *a posteriori* error estimation based on residuals and a variational analysis. This approach is distinguished by a direct attempt to account for the effects of stability on the propagation of error. We illustrate properties of this approach using several examples.

Key words. *a posteriori* error estimate, adaptive error control, computational error estimation, dual problem, finite element method, multi-scaled problems, residual, stability, stability factor, variational analysis

1. Introduction. The search for reliably accurate numerical tools for multi-scaled differential equations has become increasingly urgent in recent years. Not the least because multi-scaled problems arise in a wide range of applications and computing accurate numerical solutions often strains the limits of computational resources. One way to obtain accurate solutions of multi-scaled problems is through computational error estimation and adaptive error control. Computational error estimation is directed towards determining the kind of information that can be accurately obtained from a particular computation and estimating the accuracy of said information. This is an important goal both in general scientific terms as well as in terms of adaptive error control, that is for deciding how to use computational resources to achieve a desired accuracy.

Over the last two decades, there has been significant progress in computational error estimation and adaptive error control arising out of developments in *a posteriori* error analysis. In *a posteriori* analysis, the error of a numerical solution is estimated as much as possible in terms of computable quantities that depend on the numerical solution and in particular the estimate is computed after the numerical solution has been computed. The progress in *a posteriori* analysis has resulted in important advances in the reliably accurate solution of multi-scaled problems. As a consequence, *a posteriori techniques* have found widespread use in engineering and mathematics.

There are several different approaches to *a posteriori* error analysis. In this paper, we review a relatively new approach based on residuals and variational analysis involving the dual, or adjoint, problem to the original equation. This approach is distinguished by a direct attempt to account for the effects of propagation and accumulation of errors by computational means. We present a formal description of this approach and describe the application to elliptic and parabolic differential equations. We also illustrate various aspects of this approach using a set of numerical examples.

2. Stability and estimating the error of numerical solutions. Since one of our main concerns is the *accurate* evaluation of the effects of stability on the error of a numerical solution, we begin by discussing stability and numerical error. First consider the standard notion of stability used in the derivation of the classical *a priori* convergence error bound. This kind of error analysis begins with the introduction of a local measure of discretization error, such as truncation error, which is not typically computable. The local measure is related to the error of the numerical solution using stability properties of the solution operator of the differential equation. There is an assumption of a generic stability property like continuous dependence on data, or well-posedness, and the error bound is obtained by an argument that assumes that errors always accumulate. The resulting bound reflects the worst possible rate of accumulation of errors in a general class of problems satisfying the well-posedness condition. For example in a time (t) dependent problem, the bound resulting from a Gronwall argument typically has an exponential factor e^{Lt}, where L is almost always positive and very large. This makes the classic *a priori* error bound uselessly inaccurate for a particular computed solution of a particular problem. The fact is that the notion of well-posedness is simply too crude to be useful in the vast majority of physical problems.

*Department of Mathematics, Colorado State University, Fort Collins, Colorado 80523. (estep@math.colostate.edu). The research of D. Estep is partially supported by the National Science Foundation, DMS 9506519.

†Michael Holst, Department of Mathematics, Room 5739, AP&M Building, University of California, San Diego, 9500 Gilman Drive, Dept. 0112, La Jolla, CA 92093-0112 USA. (mholst@math.ucsd.edu). This author was supported in part by NSF CAREER Award 9875856 and in part by a UCSD Hellman Fellowship.

‡School of Chemical Engineering, Georgia Tech, Atlanta, Georgia 30332. (duane@vortex.che.gatech.edu).

We illustrate this claim with the well-known bistable (Allen-Cahn) problem $\dot{u} - \epsilon\Delta u = u - u^3$ posed on the unit interval with Neumann boundary conditions. This is used to model the motion of domain walls in a ferromagnetic material and also as a prototypical example of *metastability*. The problem has two attracting steady state solutions 1 and -1. Generic solutions eventually converge to one of these two steady state solutions, but in doing so remain nearly stationary for long periods of time. This is called metastability. We show the evolution of numerical approximation of a typical metastable solution in Fig. 2.1. Generic data

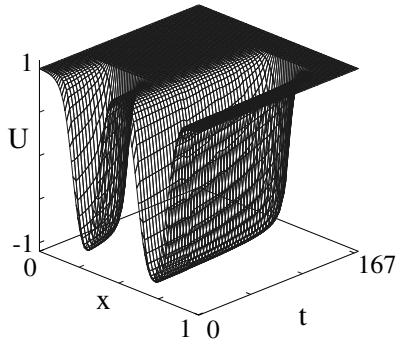

FIG. 2.1. *A metastable solution of the bistable problem with* $\epsilon = .0009$.

rapidly forms a pattern of layers between the values of -1 and 1 and then the layers move "horizontally" until they become sufficiently close and there is a rapid transient to another metastable state or the final stable state. In the solution in Fig. 2.1, there are two metastable periods $[0, 44]$ and $[44, 144]$. The convergence to a steady state solution typically takes a long time as the timescale of metastable periods increases exponentially in $1/\sqrt{\epsilon}$ as the diffusion coefficient ϵ decreases.

The question is whether or not the numeric picture presented in Fig. 2.1 makes any sense. The classic *a priori* convergence analysis for the error e gives

$$\|e(\cdot, t)\| \leq e^{20t}\big(\Delta t^p + \Delta x^q\big)C(u)$$

where $\|\quad\|$ is some space norm, $\Delta t, \Delta x$ measure the size of the time and space mesh, $C(u)$ depends on unknown derivatives of u of order depending on p and q, and e^{20t} reflects the worst possible rate of accumulation of errors under the well-posedness assumption. If this bound is accurate, then computing accurately to time $t \approx 150$ is clearly impossible. But it is clear that while the solution shown in Fig. 2.1 is exhibiting unstable behavior, we would not expect perturbations to grow continuously at an exponential rate. To accurately estimate the error of the numerical solution shown in Fig. 2.1, we have to estimate the rate that perturbations accumulate in that particular solution.

3. A formal description of one approach to *a posteriori* error analysis. *A posteriori* analysis starts with the introduction of a **computable** measure of discretization error, for example the residual obtained by substituting (in a weak sense) the numeric solution into the differential equation. As above, the analysis relates the local measure to the error through the stability properties of the solution operator, but this relation is typically approximated in some fashion. In the approach described in this paper, a variational analysis involving the dual problem to the original differential equation is used to relate the error to the residual. This approach is simple to explain in formal terms. If $\vec{X}$ denotes the computed solution of a linear continuous problem $A\vec{x} = \vec{b}$, where A is an invertible operator, then the unknown error is $\vec{e} = \vec{x} - \vec{X}$ and the residual is $\vec{R} = A\vec{X} - \vec{b}$. Since the residual of the true solution is zero, the error is related to the residual by the *perturbation relation*

$$A\vec{e} = -\vec{R}. \tag{3.1}$$

Some approaches to *a posteriori* analysis are based on exploiting (3.1) directly or indirectly. Alternatively, we introduce the dual or adjoint problem

$$A^\top \vec{\phi} = \vec{\psi}, \tag{3.2}$$

where $\vec{\psi}$ is a unit vector. A variational argument shows that $\vec{e}\cdot\vec{\psi} = \vec{e}\cdot A^\top\vec{\phi} = A\vec{e}\cdot\vec{\phi} = -\vec{R}\cdot\vec{\phi}$. This gives the *a posteriori* error estimate

$$|\vec{e}\cdot\vec{\psi}| = |\vec{R}\cdot\vec{\phi}| \le \|\vec{R}\|\,\|\vec{\phi}\|. \tag{3.3}$$

Note that (3.3) yields an estimate on a projection of the error. It turns out that this is important in many applications because the practical goal is often to compute some information obtained from the solution rather than the solution itself. We discuss this further below.

If the problem is nonlinear, say $\vec{f}(\vec{x}) = \vec{b}$, then the perturbation relation $\vec{f}(\vec{x}) - \vec{f}(\vec{X}) = -\vec{R}$ is nonlinear. We use the integral mean value theorem to write the perturbation relation as

$$A\vec{e} = \int_0^1 \vec{f}'(s\vec{x} + (1-s)\vec{X})\,ds\;\vec{e} = \vec{f}(\vec{x}) - \vec{f}(\vec{X}) = -\vec{R}$$

where $\vec{f}'$ is the Gateaux derivative of $\vec{f}$. Introducing the linear dual problem associated to the average derivative A, the analysis proceeds as above.

The factor $\|\vec{\phi}\|$ in (3.3) is called the *stability factor* and it measures the accumulation of errors as X is computed. In the context of solving linear systems, the stability factor is related to the condition number of the matrix A. It is well known that the residual of a computed solution of a linear system is nearly always small and that it is necessary to estimate the condition number of a matrix to get a reliable estimate of the error. It turns out that the same is true for differential equations. To obtain an estimate based on (3.3), we both compute the residual and numerically solve the dual problem to obtain the stability factor. By solving the dual problem, we obtain quantitative information on the relevant stability properties of the computed solution.

Adaptive error control is closely tied to computational error estimation because of the need for accurate information about the error. The idealized goal of adaptive error control is to compute some information obtained from the solution with a prescribed accuracy using the least resources. Since the true solution is unknown, the ideal constraint is replaced in practice by a computable acceptance criteria. We use an *a posteriori* error estimate tailored to the desired information for the acceptance criteria.

4. Application to differential equations. We briefly describe the *a posteriori* estimate for a continuous Galerkin space-time finite element method for

$$\dot{u} - \nabla\cdot\epsilon(u,x,t)\nabla u = f(u,x,t). \tag{4.1}$$

Detailed analysis can be found in early papers [3, 7, 8], review article [1], text [2], and monograph [12].

The time axis is partitioned $t_0 < t_1 < t_2 < \cdots$ with steps $k_n = t_n - t_{n-1}$ and intervals $I_n = [t_{n-1}, t_n]$. On each interval, the space domain is triangulated. We use $h = h(x,t)$ and $k = k(x,t)$ to denote the associated piecewise constant mesh functions. The approximation U is a continuous piecewise linear polynomial in time with coefficients in the space of continuous piecewise linear functions V_n associated to the triangulation on I_n. On each interval, U solves

$$\int_{I_n} (\dot{U}, W)\,dt + \int_{I_n} (\epsilon(U)\nabla U, \nabla W)\,dt = \int_{I_n} (f(U), W)\,dt, \tag{4.2}$$

for all $W \in V_n$, where $(\ ,\ )$ denotes the L_2 inner product in space. The data at t_{n-1} is the last value of U from the previous interval projected on the new mesh.

The analysis for a differential equation is complicated by the fact that there are residuals arising from various sources of discretization error: the approximation in space and time of the solution in polynomial spaces; the use of quadrature to compute the finite element solution; and errors in the initial data. Furthermore, each residual is scaled by its own stability factor. In this brief description, we ignore the effects of quadrature and initial error.

Written in "strong" form, the space residuals for (4.1) inside the element K with boundary sides denoted by ∂K, longest side $h(K)$, and area $|K|$ are $\mathcal{R}_x(U) = \dot{U} - \nabla\cdot\epsilon(U)\nabla U - f(U)$ and

$$\mathcal{R}_2(U) = \frac{c}{\sqrt{h(K)|K|}}\,\Big(\int_{\partial K\setminus\partial\Omega} (n_{\partial K}\cdot\epsilon(U)[\nabla U]_{\partial K}/2)^2\;ds\Big)^{1/2},$$

for an appropriate constant c, where $[\nabla U]_{\partial K}$ denotes the difference in ∇U across ∂K. $\mathcal{R}_x$ is computed by substituting U in the differential equation inside each element while $\mathcal{R}_2$ arises from the low regularity of U across element edges. The time residual is $\mathcal{R}_t(U) = \dot{U} - (\nabla \cdot \epsilon(U)\nabla)_h U - f(U)$, where $(\nabla \cdot \epsilon(U)\nabla)_h$ denotes the generalized discrete Laplacian. In "strong" form, the dual problem reads

$$-\dot{\phi} - \nabla \cdot \bar{\epsilon}\nabla\phi + \bar{\beta} \cdot \nabla\phi = \bar{f}\phi + \psi, \quad t_n > t > 0, \tag{4.3}$$

where we have linearized $\bar{\epsilon} = \int_0^1 \epsilon(us + U(1-s))\,ds$, $\bar{\beta} = \int_0^1 \epsilon'(us + U(1-s))\nabla(us + U(1-s))\,ds$, and $\bar{f} = \int_0^1 f'(us+U(1-s))\,ds$. Note the dual problem runs "backwards" in time as a consequence of integrating by parts in time, but the time derivative is multiplied by a "$-$" to compensate. The boundary conditions are the same as for the original problem. The space and time stability factors are

$$\mathcal{S}_x(t_n) = \int_0^{t_n} \|h^{-2}(I-P)\phi\|\,dt, \quad \mathcal{S}_t(t_n) = \int_0^{t_n} \|k^{-1}(I-\pi)P\phi\|\,dt, \tag{4.4}$$

where P and π denote projections into the space and time finite element spaces respectively. Using standard interpolation estimates, we can bound these as

$$\mathcal{S}_x(t_n) \leq C_x \int_0^{t_n} \|D^2\phi\|\,dt, \quad \mathcal{S}_t(t_n) \leq C_t \int_0^{t_n} \|\dot{\phi}\|\,dt,$$

for appropriate constants C_x and C_t. Finally, the strong *a posteriori* estimate reads

$$\left| \int_0^{t_n} (e,\psi)\,dt \right| \leq \mathcal{S}_t(t_n) \max_{0\leq i\leq n} k_i \|\mathcal{R}_t(U)\|_{I_i} + \mathcal{S}_x(t_n) \max_{0\leq i\leq n} \big(\|h^2\mathcal{R}_x(U)\|_{I_i} + \|h^2\mathcal{R}_2(U)\|_{I_i} \big), \tag{4.5}$$

where $\|\ \|_{I_i}$ is the L_2 norm in space and the pointwise maximum norm on the i'th interval in time. For the stationary elliptic problem corresponding to (4.1), the corresponding estimate is

$$|(e,\psi)| \leq \mathcal{S}_x \big(\|h^2\mathcal{R}_x(U)\| + \|h^2\mathcal{R}_2(U)\| \big), \tag{4.6}$$

with $\mathcal{S}_x = \|h^{-2}(I-P)\phi\| \leq C_x\|D^2\phi\|$ and $\mathcal{R}_x(U) = -\nabla \cdot \epsilon(U)\nabla U - f(U)$ elementwise.

As we remarked before, an important feature of this analysis is that **it yields an estimate of the error in a projection of the solution**. Often times, the practical goal of solving a model is to compute specific information obtained from the solution while a globally accurate solution itself is not explicitly required. For example in a flow computation, we might desire accurate values for the lift and drag on the surface of a body. In a computation in a large physical domain, we might require accurate solution values only at specific localized regions in the domain. If in such situations we can express the desired information as a linear functional, i.e. average or projection, of the solution, then we can use this analysis to determine the sensitivity of the particular information to be computed. It turns out that the sensitivity of specific information can be much different than say the sensitivity of the solution as a whole. By computing the sensitivity of the particular desired information, we are essentially attempting to characterize the minimal amount of information needed from the solution to compute that information accurately. This in turn can translate to a tremendous gain in efficiency. We explore this idea below with a concrete example.

This formal description of the *a posteriori* analysis raises several questions. *Are the residuals and stability factors defined and bounded? Can the quantities in the a posteriori error estimate be computed? How should the quantities in the a posteriori error estimate be computed? Is it worth computing the a posteriori error estimate?* We refer to the monograph [12] for a complete discussion and investigation of these issues in the context of reaction-diffusion equations. Here, we just make some observations. First, in practice the residuals and the weights determined by the solution of the dual problem are used only in weak form and norms are kept on the "outside" of the quantities on the right-hand side of the estimate as much as possible, as in the middle term in (3.3). So the error estimate is actually computed by evaluating the right-hand side of the *error representation formula*

$$\int_0^{t_n} (e,\psi)\,dt = \int_0^{t_n} \big((\dot{U}, \pi P\phi - \phi) + (\epsilon(U)\nabla U, \nabla(\pi P\phi - \phi)) - (f(U), \pi P\phi - \phi) \big)\,dt$$

that leads to (4.6). This is important in order to account for the positive effect of the cancellation of errors. The "strong" form (4.6) is used mainly to perform analysis on the estimate and to discuss the stability of a particular solution. Second, since the true solution is unknown we solve the linear problem obtained from (4.3) by substituting a known function for u, most often the numerical solution U. This introduces a "linearization" error into the estimate. Third, since we essentially require information about derivatives of ϕ, we typically solve the dual problem using a higher order method than used for the forward problem.

These questions highlight the shift in paradigm in the mathematical analysis of numerical methods resulting from the *a posteriori* approach. Classical analysis is directed towards estimating the error, which is generally frustrating since we usually turn to numerical computation because analysis is too difficult. In computational error estimation, we use computation to make up for our analytical deficiencies. The *a posteriori* estimates themselves are relatively easy to derive. The bulk of the mathematical analysis is directed towards justifying the process used to produce the computational error estimate as opposed to estimating the size of the error itself.

5. Some examples. We begin by continuing the discussion of the bistable problem. In Fig. 5.1, we show the residuals, stability factors, and the "strong" error estimate (4.5) for the solution shown in Fig. 2.1. First we see that the space residuals decrease after each transient while the time residual is small except

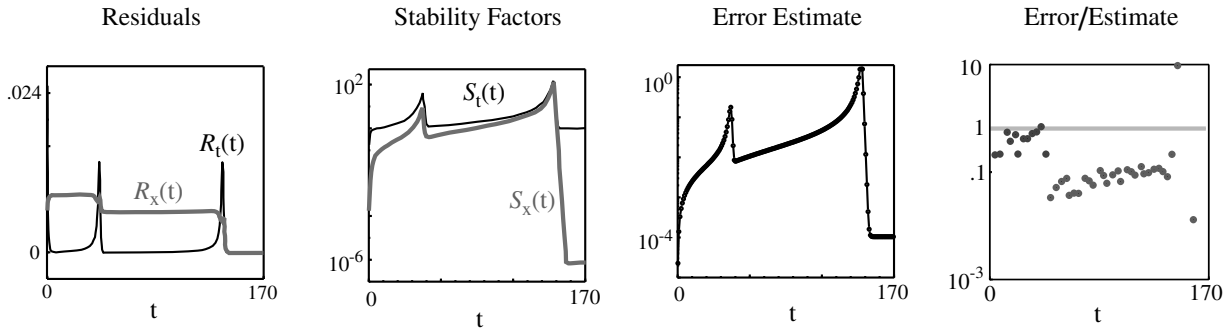

FIG. 5.1. *The error estimate for the solution of the bistable problem shown in Fig. 2.1.*

during the rapid transients. Second we see that the stability factors grow exponentially during the metastable periods, albeit at a very slow rate, but they decrease sharply during the rapid transients between metastable periods. Hence the accumulation of errors that occurs during metastable periods is periodically damped out and accurate solutions of the bistable problem can be computed over long time periods. We show the strong estimate obtained by the plotted residuals and stability factors and it reflects this accumulation/decay cycle.

We perform an experiment to determine the accuracy of the strong estimate and in particular whether the stability factors accurately reflect the accumulation of errors. We compute a very accurate numerical solution $\bar{u}$ and a low accuracy numerical solution U and then plot the ratio of the "error" $\|\bar{u} - U\|$ and the estimate for U. The ratio remains nearly piecewise constant throughout the computation; around .5 through the first transient, then around .1 through the second transient. This shows that the estimate predicts the accumulation of errors very well. The drop in the ratio is an artifact of the uniform discretization. The strong estimate (4.5) at any time involves the maximum residual from all previous times, and the residual at the first transient is an order of magnitude larger than during metastable periods, as can be seen in Fig. 5.1.

It is instructive to consider the solution of the bistable problem in two space dimensions. Again there are two steady stable states 1 and -1 and almost all solutions evolve to one of these. However, the dynamics of the problem are much different because the evolution is governed by motion by mean curvature meaning that the normal velocity of a transition layer is proportional to the sum of the principle curvatures of the layer. Consequently, the time scale for the evolution increases only at an algebraic rate, κ/ϵ, where κ is the mean curvature, as the diffusion coefficient ϵ decreases as opposed to the exponential rate in metastable solutions in one dimension. We solve the bistable problem using initial data consisting of two "mesas" corresponding to the two wells in the first example and using a value of ϵ (= .00003) so that the evolution occurs over the same length of time as the one dimensional example. We show four plots of the solution in Fig. 5.2. We plot the time stability factor $\mathcal{S}_t(t)$ for the one and two dimensional solutions in Fig. 5.3. Though the stability

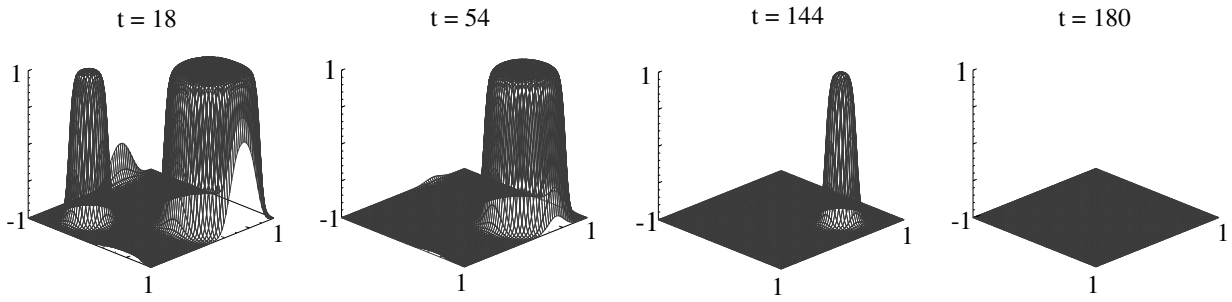

FIG. 5.2. *Some snapshots of a solution of the bistable problem in two dimensions.*

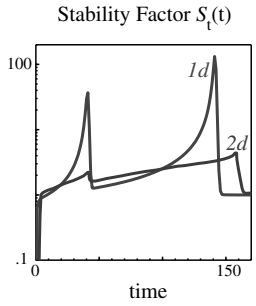

FIG. 5.3. *Time stability factors $S_t(t)$ of the bistable problem in one and two dimensions.*

factors behave similarly, we see that the solution in two dimensions is much less sensitive to accumulation of errors. This is expected since in two dimensions small perturbations smooth out very rapidly because they are associated to a high degree of curvature. This shows that the numerical stability factors are sensitive enough to pick out the crucial change in stability that results from increasing the space dimension in the problem.

The vast majority of computational error estimates are based on evaluation of some "local" measure of discretization error such as truncation error or residuals. However, as in linear algebra, it is generally impossible to get a reliable estimate of the error using a local measure such as residuals. In fact, it is possible to prove that residuals can always be made small regardless of the size of the error under very general assumptions, see [12] for a precise theorem for reaction-diffusion equations. We illustrate this fact and its consequences for computational error estimation using the well-known chaotic Lorenz problem

$$\begin{cases} \dot{x} = -10x + 10y, \\ \dot{y} = 28x - y - xz, \\ \dot{z} = -\frac{8}{3}z + xy. \end{cases} \tag{5.1}$$

One consequence of the chaotic nature of this problem is that the error of any numerical solution grows as time passes. We plot two numerical approximations of the same solution computed with different accuracies in Fig. 5.4. The more accurate solution is computed with an error of 2% on $[0, 30]$ while the inaccurate solution is reasonably accurate until $t \approx 17.8$ but then has a 100% error for $t \geq 17.8$. It turns out that this is typical behavior: the error of a numerical solution generally remains small until some critical time when the error suddenly increases very rapidly. **However, the sudden increase in error is not due to the residual becoming large.** In Fig. 5.4, we plot the residual R_t of the more inaccurate numerical solution versus time. The residual does not become large near $t \approx 17.8$ even though the error suddenly increases there.

The answer can be found by examining the stability factor shown in Fig. 5.4. Despite the chaotic nature of the problem, the approximate $S_t(0, t_n)$ does not grow exponentially at a steady rate. As apparent in the solutions shown in Fig. 5.4, a generic solution spends most of the time orbiting around one of two nonzero fixed points. During this period, errors accumulate at a polynomial rate on average ($S_t(0, t) \sim t^{1.4}$). But

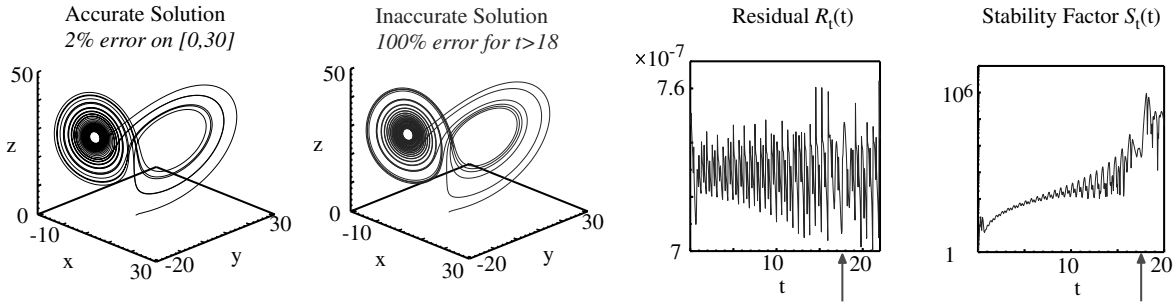

FIG. 5.4. *Two numerical solutions of the Lorenz problem and the residual and stability factor for the inaccurate solution.*

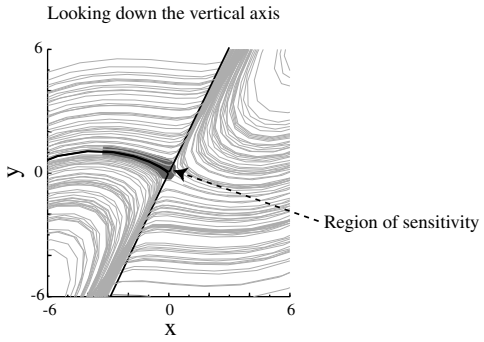

FIG. 5.5. *Many numerical solutions of the Lorenz problem and the region of extreme sensitivity.*

this relatively slow growth is punctuated by short periods of exponential growth ($S_t(0,t) \sim e^{4t}$) during the transition as the solution passes from the neighborhood of one fixed point to a neighborhood of the other fixed point. This can be seen clearly in the plot of the stability factor in Fig. 5.4.

This exponential growth coincides exactly with the time that the trajectory first becomes grossly inaccurate. In other words, the cause of the sudden decrease in accuracy of numerical solutions of the Lorenz system appears to be that trajectories become strongly unstable in a region of phase space near the vertical axis. Generally there is a slow accumulation of errors in a solution unless the trajectory of the solution takes it through this region. The instability in this region reflects the fact that trajectories that are very close as they approach the vertical axis can end up around different fixed points. In Fig. 5.5, we show a plot of many solutions of the Lorenz problem from a viewpoint high on the vertical axis looking straight down. We shade the region of extreme sensitivity associated to the exponential growth of error and we use darker lines to mark the trajectories of two solutions that are extremely close yet end up far apart after a short time.

Given that residuals can always be made small in general, it is natural to wonder about the kinds of stability and instability that can be encountered in practice. In other words, is it always important to account for the accumulation of errors? Stability factors for a variety of problems have been computed, see the references in [12]. There are a couple of examples like the linear homogeneous heat equation in which there is little or no accumulation of errors. There are a few other examples where there is a slow, steady accumulation of errors, as for example can occur in some hyperbolic problems. But such examples appear to be both rare and of limited practical interest and it appears that most problems exhibit much more complicated stability behavior. Even problems where solutions eventually converge to a steady-state like the bistable problem or that have a well-defined average rate of accumulation of error like the Lorenz problem often appear to have periods of relative stability and instability.

It is important to note that stability may vary in space as well as time. To illustrate this, we conclude this section with a couple of stationary elliptic problems. The first is a semilinear elliptic "Bratu" problem

$$-\Delta u = 2\pi^2 \sin(\pi x)\sin(\pi y) + e^{\sin(\pi x)\sin(\pi y)} - e^u \tag{5.2}$$

posed on the unit square with Dirichlet boundary conditions. This problem is constructed with a known solution, $u(x,y) = \sin(\pi x)\sin(\pi y)$, so that we can test the accuracy of the error estimate precisely. In

Fig. 5.6, we plot the mesh and approximation along with contour plots of the local element values of the residual and stability factor for meshes with 32 elements and 512 elements. In this computation, we estimate the average error by choosing the dual data $\psi = 1$. Note the variation in **both** the residuals and dual weights that arises from the use of uniform discretizations. In Fig. 5.7, we demonstrate the accuracy of the estimate

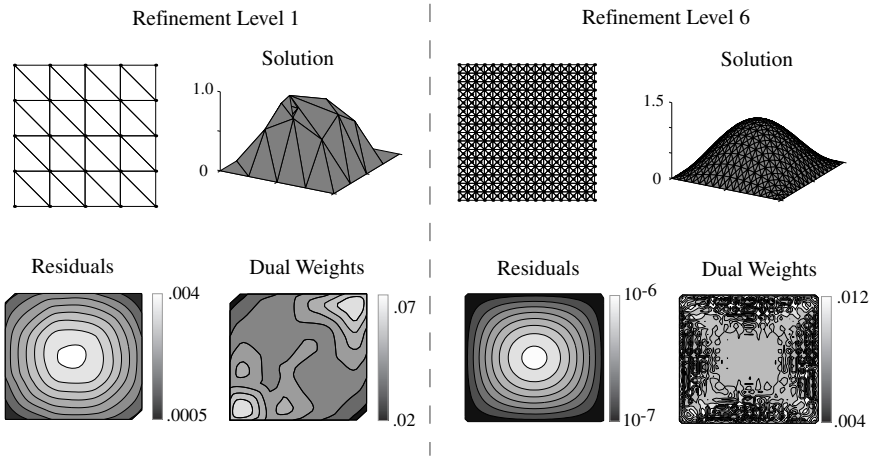

FIG. 5.6. *Solutions of the nonlinear elliptic problem (5.2).*

by plotting the ratio of the error to the estimate for different meshes. The ratio is very close to 1 even on

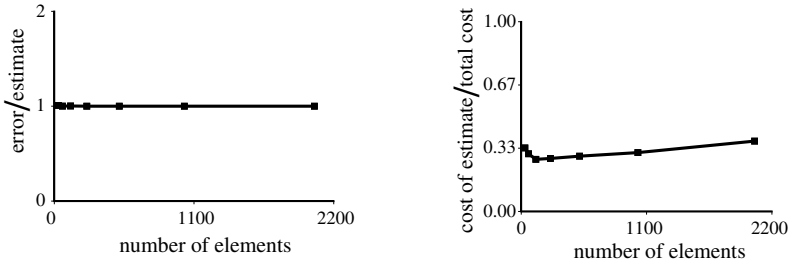

FIG. 5.7. *The accuracy and cost of the error estimate for the numerical solution of the elliptic problem shown in Fig. 5.6.*

the coarsest meshes. To show the cost of the estimate, we plot the ratio of the cost of the estimate versus the total cost of the computation. Since the dual problem is linear, the cost of solving it is roughly the same as the cost of one Newton iteration in the forward problem. **In current implementations, the cost of a single estimate generally runs %20–%30 of the total cost of the solution of a nonlinear problem**.

The next example presents an extreme test of the ability to estimate the effects of cancellation of errors accurately. The problem is

$$-\Delta u = 200 \sin(10\pi x) \sin(10\pi y) \tag{5.3}$$

on the unit square with Dirichlet boundary conditions. We plot the solution $u(x, y) = \sin(10\pi x) \sin(10\pi y)$ in Fig. 5.8. The oscillations in the solution lead to a high degree of cancellation of local discretization errors. In this problem, we estimate the average error, and in Fig. 5.8, we plot the corresponding dual solution.

In the first test of the accuracy of the estimate, we compute solutions on a sequence of successively refined meshes and then estimate the error. We plot the ratio of the true error to the estimates versus the number of elements in the plot on the left in Fig. 5.9. We see that the ratio is always reasonably close to 1 and becomes very close for moderate mesh refinement and finer. Computing on highly refined meshes leads to very accurate estimates of the error on this problem. However obtaining a very accurate estimate of the error in an asymptotic limit has limited practical usefulness. In real world problems, it is rarely the case that we can compute to very high precision and it is rarely desirable to compute to high precisions just to

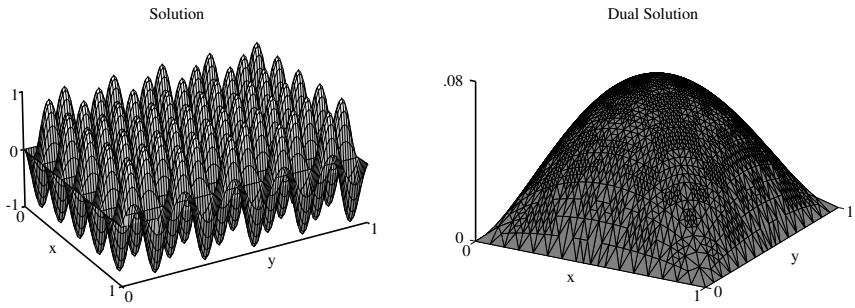

FIG. 5.8. *The solution and dual solution for the highly oscillatory problem (5.3).*

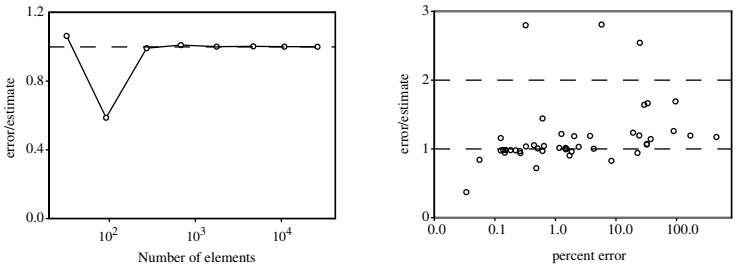

FIG. 5.9. *Ratio of error to estimate for a sequence of successively refined meshes on the left and a large collection of relatively coarse meshes on the right. We gauge the meshes on the right by the error of the corresponding numerical solution.*

obtain accurate error estimates. Therefore it is important to test computational error estimates on relatively coarse discretizations. This example presents an extreme test of the estimator's capability of accounting for the effects of cancellation of errors on coarse meshes. In the plot on the right in Fig. 5.9, we plot the ratio of the error to the estimate for a number of different solutions of accuracies ranging from 1% to 200% (number of elements 15 to 1000). The ratio is between .5 and 3 for all solutions we computed, but rarely falls outside the range of $[.75, 1.5]$.

6. Adaptive error control. To use the *a posteriori* estimate for adaptive error control, we write the estimate as a sum over the space-time elements E of a triangulation of the space-time domain Ω_T

$$\left| \int_{\Omega_T} (e, \psi)\, dt \right| \approx \left| \sum_E \int_E \text{stability factor}|_E \times \text{residual}|_E \, dxdt \right|$$

where stability factor$|_E$, residual$|_E$ are the element contributions to the stability factor and residual. We refine the mesh in order to equidistribute

$$\int_E |\text{stability factor}|_E \times \text{residual}|_E \,|\, dxdt$$

while keeping

$$\left| \sum_E \int_E \text{stability factor}|_E \times \text{residual}|_E \, dxdt \right| \leq \text{TOL}$$

where TOL is the desired accuracy. In this approach, elements may be refined both because the residual is large, i.e. the differential equation is difficult to discretize, and because the stability factor is large, i.e. the solution of the differential equation is particularly sensitive to perturbations.

In particular, the fact that the analysis is tailored to the desired information about a solution can have a profound effect on the discretization required to achieve a given level of accuracy. We illustrate this with an example. The problem is a linear, singularly-perturbed elliptic problem

$$-\nabla \cdot (\frac{1}{2}(1 + \tanh(7(y - .8) + 25|x - .5|))\nabla u) = 1, \tag{6.1}$$

posed on a rectangle $[0,1] \times [0,4]$ with Dirichlet boundary conditions. We plot the diffusion coefficient in the upper left of Fig. 6.1. A reasonably accurate approximation of the solution is shown in the upper middle plot. The singular nature of this problem means that the stability of solutions vary greatly through the domain. We compute two numeric solutions. The goal of the first is to compute a solution whose average error through the domain is smaller than .001 and the goal of the second is to compute a solution whose error at a point $(.5, 3.5)$ far away from the near-singularity is smaller than .001. By choosing suitable data for the dual problem, we can compute this information. For the first, we use $\psi \equiv 1$ and for the second, we choose ψ to be an approximate delta function. We show the solutions after seven adaptive refinements starting with an initial uniform mesh. In the first computation, we have achieved an estimated accuracy of .015 using 5760 elements. In the second case, we have reached an accuracy of 6×10^{-5} using only 210 elements. We plot the corresponding meshes in the lower left-hand corner.

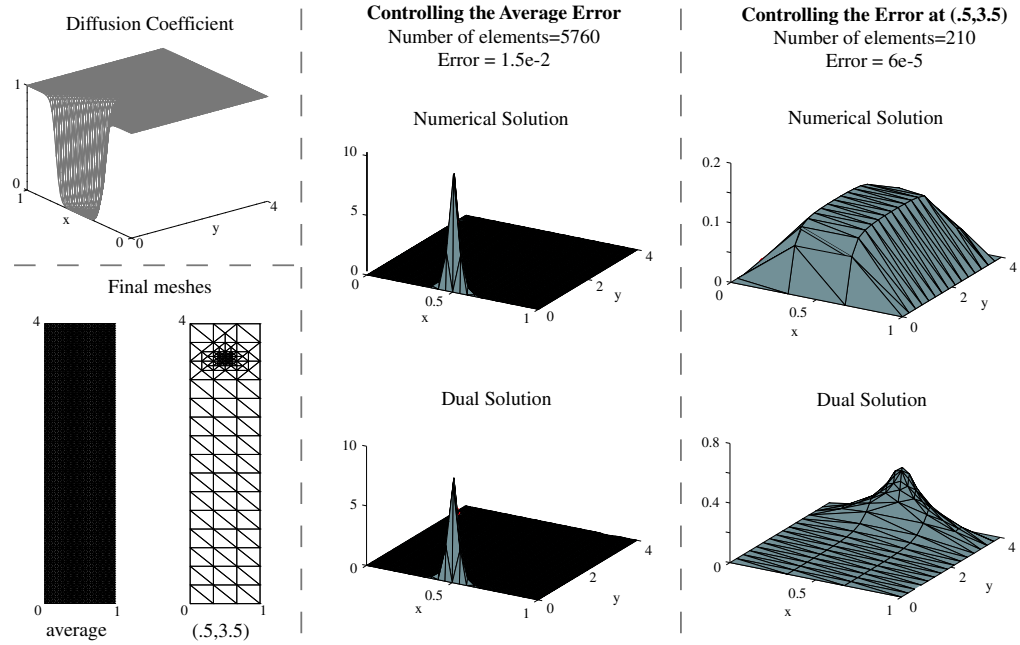

FIG. 6.1. *Numerical results for the singularly perturbed elliptic problem (6.1).*

In the approach to adaptive error control presented in this paper, both time and space discretizations will generally be affected by the stability properties of the solution. This raises some interesting issues for adaptive error control. Considering the Lorenz example again. We know that a solution of the Lorenz problem passing through the region of sensitivity is very susceptible to becoming suddenly inaccurate. However the error of a solution passing through that region is determined not only by the discretization during the time it is in the region but also by the accumulation of errors from the time prior to entering the region. Using small time steps while a solution is in the region of sensitivity can reduce the errors made while in the region but cannot reduce the size of the error the solution inherited from previous times. Thus a purely local adaptive error control strategy cannot reduce the prior error. To reduce the prior error, we have to go back to the beginning and refine the discretization from the start, i.e. we have to employ a global error control strategy. It is common to adopt a global error control strategy in space, for example equidistribution of element contributions is a global strategy, but error control strategies that are global in time are rarely implemented. (See [9] for an example.) There are certainly practical difficulties, such as memory usage, facing true global space-time approaches. Yet local strategies are appropriate only for the relatively few problems like the linear heat equation in which errors do not accumulate. Thus there remain many interesting research questions for adaptive error control for evolutionary problems.

7. Conclusion. In this paper, we discuss a relatively new approach to *a posteriori* error estimation based on residuals and variational analysis. This approach is distinguished by an attempt to account for the precise effects of stability on the accumulation and propagation of errors computationally. After presenting

a formal description of the analysis, we describe the application to a finite element method for nonlinear parabolic and elliptic problems and formulate several key issues that arise in understanding this approach. We illustrate these issues, and the need for consideration of stability in error estimation, using a set of numerical examples. On a variety of difficult problems, the proposed approach to computational error estimation yields accurate estimates at a reasonable cost. It also provides a powerful tool for the analysis of the stability properties of differential equations.

The stationary problems presented in this paper were solved using MCLab [10]. Running under MATLAB, MCLab is an adaptive finite element code that can solve general systems of elliptic equations on general domains in two dimensions. MCLab can solve reasonably large problems and can be used to explore various aspects of the computational error estimation and adaptive error control described in this paper. In particular, it allows different data for the dual problem to be specified, various orders of elements, different adaptive strategies, various refinement criteria, accumulation of statistics on the estimates, plots of various relevant quantities, and so on. It is also written in a "clean" modular fashion and can be executed through a graphical interface that includes extensive help, hence it is well-suited as a teaching tool for courses on adaptive finite element methods. It is freely available for download and use.

REFERENCES

[1] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive methods for differential equations*, Acta Numerica, (1995), pp. 105–158.

[2] ———, *Computational Differential Equations*, Cambridge University Press, New York, 1996.

[3] K. Eriksson and C. Johnson, *Adaptive finite element methods for parabolic problems I: A linear model problem*, SIAM J. Numer. Anal., 28 (1991), pp. 43–77.

[4] ———, *Adaptive finite element methods for parabolic problems. IV. Nonlinear problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1729–1749.

[5] ———, *Adaptive finite element methods for parabolic problems V: Long–time integration*, SIAM J. Numer. Anal., 32 (1995), pp. 1750–1763.

[6] D. Estep, *An analysis of numerical approximations of metastable solutions of the bistable equation*, Nonlinearity, 7 (1994), pp. 1445–1462.

[7] ———, *A posteriori error bounds and global error control for approximations of ordinary differential equations*, SIAM J. Numer. Anal., 32 (1995), pp. 1–48.

[8] D. Estep and D. French, *Global error control for the continuous Galerkin finite element method for ordinary differential equations*, RAIRO Modél. Math. Anal. Numér., 28 (1994), pp. 815–852.

[9] D. Estep, D. Hodges, and M. Warner, *Computational error estimation for a finite element solution of missile trajectory optimization problems*, SIAM J. Sci. Computing, 21 (2000), pp. 1609–1631.

[10] D. Estep, M. Holst, and D. Mikulencak, *MCLAB: manifold code for solving nonlinear elliptic systems in MATLAB*, 1997. can be obtained from http://scicomp.ucsd.edu/∼mholst/codes/mclab/mclab.tar.gz.

[11] D. Estep and C. Johnson, *The computability of the Lorenz system*, Math. Models Meth. Appl. Sci., 8 (1998), pp. 1277–1305.

[12] D. Estep, M. Larson, and R. Williams, *Estimating the error of numerical solutions of systems of reaction-diffusion equations*, Mem. Amer. Math. Soc, 696 (2000), pp. 1–109.

[13] D. Estep and R. Williams, *Accurate parallel integration of large sparse systems of differential equations*, Math. Models Meth. Appl. Sci., 6 (1996), pp. 535–568.