Acta Numerica (1995), *pp.* 105–158

Introduction to Adaptive Methods for Differential Equations

Kenneth Eriksson
Mathematics Department,
Chalmers University of Technology,
412 96 Göteborg
kenneth@math.chalmers.se

Don Estep
School of Mathematics,
Georgia Institute of Technology,
Atlanta GA 30332
estep@pmath.gatech.edu

Peter Hansbo
Mathematics Department,
Chalmers University of Technology,
412 96 Göteborg
hansbo@math.chalmers.se

Claes Johnson
Mathematics Department,
Chalmers University of Technology,
412 96 Göteborg
claes@math.chalmers.se

Knowing thus the Algorithm of this calculus, which I call Differential Calculus, all differential equations can be solved by a common method (Gottfried Wilhelm von Leibniz, 1646–1719).

When, several years ago, I saw for the first time an instrument which, when carried, automatically records the number of steps taken by a pedestrian, it occurred to me at once that the entire arithmetic could be subjected to a similar kind of machinery so that not only addition and subtraction, but also multiplication and division, could be accomplished by a suitably arranged machine easily, promptly and with sure results.... For it is unworthy of excellent men to lose hours like slaves in the labour of calculations, which could safely be left to anyone else if the machine was used.... And now that we may give final praise to the machine, we may say that it will be desirable to all who are engaged in computations which, as is well known, are the managers of financial affairs, the administrators of others estates, merchants, surveyors, navigators, astronomers, and those connected with any of the crafts that use mathematics (Leibniz).

CONTENTS

1 Leibniz's vision 106
2 Computability and predictability 108
3 The finite-element method 110
4 Adaptive computational methods 111
5 General framework for analysis of Fem 112
6 Piecewise-polynomial approximation 118
7 An elliptic model problem in one dimension 119
8 Basic time-dependent model problems 126
9 Nonlinear systems of ordinary differential equations 137
10 An elliptic model problem in two dimensions 143
11 Adaptive algorithms 149
12 The heat equation 150
13 References 153
14 Conclusion, open problems 154
15 How to obtain Femlab and 'Introduction to Numerical Methods for Differential Equations' 155
References 155

1. Leibniz's vision

Newton and Leibniz invented calculus in the late 17th century and laid the foundation for the revolutionary development of science and technology up to the present day. Already 300 years ago, Leibniz sought to create a 'marriage' between calculus and computation, but failed because the calculator he invented was not sufficiently powerful. However, the invention of the modern computer in the 1940s started a second revolution and today, we are experiencing the realization of the original Leibniz vision. A concrete piece of evidence of the 'marriage' is the rapid development and spread of mathematical software such as Mathematica, Matlab and Maple and the large number of finite-element codes.

The basic mathematical models of science and engineering take the form of differential equations, typically expressing laws of physics such as conservation of mass or momentum. By determining the solution of a differential equation for given data, one may gain information concerning the physical process being modelled. Exact solutions may sometimes be determined through symbolic computation by hand or using software, but in most cases this is not possible, and the alternative is to approximate solutions with numerical computations using a computer. Often massive computational effort is needed, but the cost of computation is rapidly decreasing and new possibilities are quickly being opened. Today, differential equations modelling complex phenomena in three space dimensions may be solved using desktop workstations.

As a familiar example of mathematical modelling and numerical solution, consider weather prediction. Weather forecasting is sometimes based on solving numerically a system of partial differential equations related to the Navier–Stokes equations that model the evolution of the atmosphere beginning from initial data obtained from measuring the physical conditions – temperature, wind speed, etc. – at certain locations. Such forecasts sometimes give reasonably correct predictions but are also often incorrect. The sources of errors affecting the reliability are data, modelling and computation. The initial conditions at the start of the computer simulation are measured only approximately, the set of differential equations in the model only approximately describe the evolution of the atmosphere, and finally the differential equations can be solved only approximately. All these contribute to the total error, which may be large. It is essential to be able to estimate the total error by estimating individually the contributions from the three sources and to improve the precision where most needed. This example contains the issues in mathematical modelling that are common to all applications.

In these notes, we present a framework for the design and analysis of computational methods for differential equations. The general objective is to achieve reliable control of the total error in mathematical modelling including data, modelling and computation errors, while making efficient use of computational resources. This goal may be achieved using adaptive methods with feedback from computations. The framework we describe is both simple enough to be introduced early in the mathematical curriculum and general enough to be applied to problems on the frontiers of research. We see a great advantage in using this framework in a mathematics education program. Namely, its simplicity suggests that numerical methods for differential equations could be introduced even in the calculus curriculum, in line with the Leibniz idea of combining calculus and computation. In these notes, we hope to reach a middle ground between mathematical detail and ease of understanding. In Eriksson, *et al.* (1994), we give an even more simplified version aimed at early incorporation in a general mathematical curriculum. These notes, together with the software Femlab implementing the adaptive methods for a variety of problems, are publicly available through the Internet; see below. In the textbook Eriksson, *et al.* (in preparation), we develop the framework in detail and give applications not only to model problems, but also to a variety of basic problems in science including, for example, the Navier–Stokes equations for fluid flow.

We begin by discussing the basic concepts of predictability and computability, which are quantitative measures of the accuracy of prediction from the computational solution of a mathematical model consisting of differential equations.

In the next part, we present an abstract framework for discretization, error estimation and adaptive error control. We introduce the fundamental concepts of the framework: reliability and efficiency, a priori and a posteriori error estimates, accuracy and stability. We then recall the basic principles underlying the Galerkin finite-element method (Fem), which we use as a general method of discretization for all differential equations. We then describe the fundamental ingredients of error estimates for Galerkin discretization, including stability, duality, Galerkin orthogonality and interpolation. We also discuss data, modelling, quadrature and discrete-solution errors briefly.

In the last part, we apply this framework to a variety of model problems. We begin by recalling some essential facts from interpolation theory. We next consider a collection of model problems including stationary as well as time-dependent, linear and non-linear, ordinary and partial differential equations. The model problems represent a spectrum of differential equations including problems of elliptic, parabolic and hyperbolic type, as well as general systems of ordinary differential equations. In each case, we derive a posteriori and a priori error bounds and then construct an adaptive algorithm based on feedback from the computation. We present a sample of computations to illustrate the results. We conclude with references to the literature and some reflections on future developments and open problems.

2. Computability and predictability

Was man mit Fehlerkontrolle nicht berechnen kann, darüber muss mann schweigen (Wittgenstein).

The ability to make predictions from a mathematical model is determined by the concepts of *computability* and *predictability*. We consider a mathematical model of the form

$$A(u) = f, \tag{2.1}$$

where A represents a differential operator with specified coefficients (including boundary and initial conditions) on some domain, f is given data and u is the unknown solution. Together, A and f define the mathematical model. We assume that by numerical and/or symbolic computation an approximation U of the exact solution u is computed, and we define the *computational error* $e_c \equiv u - U$. The solution u, and hence U, is subject to perturbations from the data f and the operator A. Letting the unperturbed form of (2.1) be

$$\hat{A}(\hat{u}) = \hat{f}, \tag{2.2}$$

with unperturbed operator $\hat{A}$, data $\hat{f}$ and corresponding solution $\hat{u}$, we define the *data-modelling error* $e_{\mathrm{dm}} \equiv \hat{u} - u$. In a typical situation, the unperturbed problem (2.2) represents a complete model that is computationally too complex to allow direct computation, and (2.1) a simplified model that is actually used in the computation. For example, (2.2) may represent the full Navier–Stokes equations, and (2.1) a modified Navier–Stokes equations that is determined by a turbulence model that eliminates scales too small to be resolved computationally.

We define the total error e as the sum of the data-modelling and computational errors,

$$e \equiv \hat{u} - U = \hat{u} - u + u - U \equiv e_{\mathrm{dm}} + e_{\mathrm{c}}. \tag{2.3}$$

Basic problems in computational mathematical modelling are: (i) estimate quantitatively the total error by estimating both the data-modelling error e_{dm} and the computational error e_{c}, and (ii) control any components of the data-modelling and computational errors that can be controlled. Without some quantitative estimation and even control of the total error, mathematical modelling loses its meaning.

We define the solution $\hat{u}$ of the unperturbed model (2.2) to be *predictable* with respect to a given norm $\|\cdot\|$ and tolerance $TOL > 0$ if $\|e_{\mathrm{dm}}\| \leq TOL$. We define the solution u of the perturbed model (2.1) to be *computable* with respect to a given norm $\|\cdot\|$, tolerance TOL and computational work, if $\|e_{\mathrm{c}}\| \leq TOL$ with the given computational work. Note that the choice of norm depends on how the error is to be measured. For example, the L^2 and L^∞ norms are appropriate for the standard goal of approximating the values of a solution. Other norms are appropriate if some qualitative feature of the solution is the goal of approximation. We note that the predictability of a solution is quantified in terms of the norm $\|\cdot\|$ and the tolerance TOL, whereas the computability of a solution is quantified in terms of the available computational power, the norm $\|\cdot\|$ and the tolerance TOL. There is a natural scale for computability for all models, namely, the level of computing power. The scale for predictablity depends on the physical situation underlying the model. The relevant level of the tolerance TOL and the choice of norm depend on the particular application and the nature of the solution $\hat{u}$.

A mathematical model with predictable and computable solutions may be useful since computation with the given model and data may yield relevant information concerning the phenomenon being modelled.

If the uncertainty in data and/or modelling is too large, individual solutions may effectively be non-predictable, but may still be computable in the sense that the computational error for each choice of data is below the chosen tolerance. In such cases, accurate computations on a set of data may give useful information of a statistical nature. Thus, models with non-predictable but computable solutions may be considered partially deterministic, that is, deterministic from the computational point of view but non-deterministic from the data-modelling point of view. One may think of weather prediction again, in which it is not possible to describe the initial state as accurately as one could enter the data into a computation. Finally, models for which solutions are non-computable do not seem to be useful from a practical point of view.

The computational error e_{c} is connected to perturbations arising from discretization through a certain stability factor S_{c} measuring the sensitivity of u to perturbations from discretization. The computability of a problem may be estimated quantitatively in terms of the stability factor S_{c} and a quantity Q related to the nature of the solution u being computed and the tolerance level. The basic test for computability reads: if

$$S_{\mathrm{c}} \times Q \leq P \tag{2.4}$$

where P is the available computational power, then the problem is numerically computable, whereas if $S_{\mathrm{c}} \times Q > P$, then the problem is not computable. In this way we may give the question of numerical computability a precise quantitative form for a given exact solution, norm, tolerance and amount of computational power. Note that $S_{\mathrm{c}} \times Q$ is related to the complexity of computing the solution u, and an uncomputable solution has a very large stability factor S_{c}. This occurs with pointwise error control of direct simulation of turbulent flow without turbulence modelling over a long time, for example.

Similarly, the sensitivity of $\hat{u}$ to data errors may be measured in terms of a stability factor S_{d}. If $S_{\mathrm{d}} \times \delta$ is sufficiently small, where δ measures the error in the data, then the problem is predictable from the point of view of data error. In addition, some kind of modelling errors can be associated to a stability factor S_{m} and if $S_{\mathrm{m}} \times \mu$ is sufficiently small, where μ measures the error in the model, then the problem is predictable from the point of view of modelling.

Different perturbations propagate and accumulate differently, and this is reflected in the different stability factors. The various stability factors are approximated by numerically solving auxiliary linear problems. In the adaptive algorithms to be given, these auxiliary computations are routinely carried out as a part of the adaptive algorithm and give critically important information on perturbation sensitivities.

3. The finite-element method

$(Fe)^m$: Finite elements, For everything, For everyone, 'For ever' $(m = 4)$.

Fem is based on

- Galerkin's method for discretization,
- piecewise-polynomial approximation in space, time or space/time.

In a Galerkin method, the approximate solution is determined as the member of a specified finite-dimensional space of trial functions for which the residual error is orthogonal to a specified set of test functions. The residual error, or simply the residual, is obtained by inserting the approximate solution into the given differential equation. The residual of the exact solution is zero, whereas the residual of approximate solutions deviates from zero. Applying this method for a given set of data leads to a system of equations that is solved using a computer to produce the approximate solution. In Fem, the trial and test functions are piecewise polynomials. A piecewise polynomial is a function that is equal to a polynomial, for example, a linear function, on each element of a partition of a given domain in space, time or space-time into subdomains. The subdivision is referred to as a mesh and the subdomains as elements. In the simplest case, the trial and test space are the same.

If the trial functions are continuous piecewise polynomials of degree q and the test functions are continuous or discontinuous, we refer to the resulting methods as continuous Galerkin or cG(q) methods. With discontinuous piecewise polynomials of degree q in both trial and test space, we obtain discontinuous Galerkin methods or dG(q) methods.

4. Adaptive computational methods

The goal of the design of any numerical computational method is

- *reliability,*
- *efficiency.*

Reliability means that the computational error is controlled on a given tolerance level; for instance, the numerical solution is guaranteed to be within 1 per cent of the exact solution at every point. Efficiency means that the computational work to compute a solution within the given tolerance is essentially as small as possible.

The computational error of a Fem has three sources:

- Galerkin discretization,
- quadrature,
- solution of the discrete problem.

The Galerkin discretization error arises because the solution is approximated by piecewise polynomials. The quadrature error comes from evaluating the integrals arising in the Galerkin formulation using numerical quadrature, and the discrete-solution error results from solving the resulting discrete systems only approximately, using Newton's method or multigrid methods, for example. It is natural to seek to balance the contribution to the total computational error from the three sources.

To achieve the goals of reliability and efficiency, a computational method must be adaptive with feedback from the computational process. An *adaptive method* consists of a discretization method together with an *adaptive algorithm.* An adaptive algorithm consists of

- a *stopping criterion* guaranteeing error control to a given tolerance level,
- a *modification strategy* in case the stopping criterion is not satisfied.

The adaptive algorithm is used to optimize the computational resources to achieve both reliability and efficiency. In practice, optimization is performed by an iterative process, where in each step an approximate solution is computed on a given mesh with piecewise polynomials of a certain degree, a certain quadrature and a discrete-solution procedure. If the stopping criterion is satisfied, then the approximate solution is accepted. If the stopping criterion is not satisfied, then a new mesh, polynomial approximation, set of quadrature points and discrete-solution process are determined through the modification strategy and the process is continued. To start the procedure, a coarse mesh, low-order piecewise-polynomial approximation, set of quadrature points and discrete-solution procedure are needed.

Feedback is centrally important to the optimization process. The feedback is provided by the computational information used in the stopping criteria and the modification strategy.

We now consider the Galerkin discretization error. Adaptive control of this error is built on *error estimates.* The control of quadrature and discrete-solution errors is largely parallel, but each error has its own special features to be taken into account.

5. General framework for analysis of Fem

5.1. Error estimates

Error estimates for Galerkin discretizations come in two forms:

- *a priori error estimates,*
- *a posteriori error estimates.*

An a priori estimate relates the error between the exact and the approximate solution to the regularity properties of the exact (unknown) solution. In an a posteriori estimate, the error is related to the regularity of the approximation.

The basic concepts underlying error estimates are

- *accuracy,*
- *stability.*

Accuracy is a measure of the level of the discretization at each point of the domain, while stability is a measure of the degree to which discretization errors throughout the domain interact and accumulate to form the total error. These properties enter in different forms in the a posteriori and a priori error estimates. The a posteriori version may be expressed conceptually as follows:

$$\text{small residual} + \text{stability of the continuous problem} \Longrightarrow \text{small error}, \quad (5.1)$$

where the continuous problem refers to the given differential equation. The a priori version takes the conceptual form

$$\text{small interpolation error} + \text{stability of the discrete problem} \Longrightarrow \text{small error}, \quad (5.2)$$

where the interpolation error is the difference between the exact solution and a piecewise polynomial in the Fem space that interpolates the exact solution in some fashion. Note that the a posteriori error estimate involves the stability of the continuous problem and the a priori estimate the stability of the discrete problem. We see that accuracy is connected to the size of the residual in the a posteriori case, and to the interpolation error in the a priori case.

Both the residual and the interpolation error contribute to the total error in the Galerkin solution. The concept of stability measures the accumulation of the contributions and is therefore fundamental. The stability is measured by a multiplicative *stability factor*. The size of this factor reflects the computational difficulty of the problem. If the stability factor is large, then the problem is sensitive to perturbations from the Galerkin discretization and more computational work is needed to reach a certain error tolerance.

In general, there is a trade-off between the norms used to measure stability and accuracy, that is, using a stronger norm to measure stability allows a weaker norm to be used to measure accuracy. The goal is to balance the measurements of stability and accuracy to obtain the smallest possible bound on the error. The appropriate stability concept for Galerkin discretization methods on many problems is referred to as *strong stability* because the norms used to measure stability involve derivatives. Strong stability is possible because of the orthogonality built into the Galerkin discretization. For some problems, Galerkin's method needs modification to enhance stability.

The stopping criterion is based solely on the a posteriori error estimate. The modification strategy, in addition, may build on a priori error estimates. The adaptive feature comes from the information gained through computed solutions.

5.2. A posteriori error estimates

The ingredients of the proofs of *a posteriori* error estimates for Galerkin discretization are:

1 Representation of the error in terms of the residual of the finite-element solution and the solution of a continuous (linearized) dual problem.
2 Use of Galerkin orthogonality.
3 Local interpolation estimates for the dual solution.
4 Strong-stability estimates for the continuous dual problem.

We describe (1)–(4) in an abstract situation for a symbolic linear problem of the form

$$Au = f, \tag{5.3}$$

where $A\colon V \to V$ is a given linear operator on V, a Hilbert space with inner product $(\cdot,\cdot)$ and corresponding norm $\|\cdot\|$, and $f \in V$ is given data. The corresponding Galerkin problem is: find $U \in V_h$ such that

$$(AU, v) = (f, v), \quad \forall v \in V_h,$$

where $V_h \subset V$ is a finite-element space. In many cases, $V = L^2(\Omega)$, where Ω is a domain in $\mathbb{R}^n$. We let $e \equiv u - U$.

1 Error representation via a dual problem:

$$\|e\|^2 = (e, e) = (e, A^*\varphi) = (Ae, \varphi) = (f - AU, \varphi) = -(R(U), \varphi),$$

where φ solves the dual problem

$$A^*\varphi = e,$$

with A^* denoting the adjoint of A, and $R(U)$ is the residual defined by

$$R(U) \equiv AU - f.$$

2 Galerkin orthogonality: Since $(Ae, v) = -(R(U), v) = 0, \quad \forall v \in V_h$,

$$\|e\|^2 = -(R(U), \varphi - \pi_h\varphi),$$

where $\pi_h\varphi \in V_h$ is an interpolant of φ.

3 Interpolation estimate:

$$\|h^{-\beta}(\varphi - \pi_h\varphi)\| \leq C_{\mathrm{i}}\|D^\beta\varphi\|,$$

where C_{i} is an interpolation constant, h is a measure of the size of the discretization and $D^\beta\varphi$ denotes derivatives of order β of the dual solution φ. Such estimates follow from classical interpolation theory when the solution is smooth.

4 Strong-stability estimate for the dual continuous problem:

$$\|D^\beta\varphi\| \leq S_{\mathrm{c}}\|e\|,$$

where S_{c} is a strong-stability factor.

Combining (1)–(4), we obtain

$$\|e\|^2 = (R(U), \pi_h\varphi - \varphi) \leq S_{\rm c}C_{\rm i}\|h^\beta R(U)\| \; \|e\|,$$

which gives the following a posteriori error estimate

$$\|u - U\| \leq S_{\rm c}C_{\rm i}\|h^\beta R(U)\|. \tag{5.4}$$

The indicated framework for deriving a posteriori error estimates for Galerkin methods is very general. In particular, it extends to problems $A(u) = f$, where A is a nonlinear operator. In such a case, the operator A^* in the dual problem is the adjoint of the Fréchet derivative of A (linearized form of A) evaluated between u and U. Details are given below in the context of systems of nonlinear ordinary differential equations.

5.3. A priori error estimates

Proofs of *a priori* error estimates have similar ingredients:

1 Representation of the error in terms of the exact solution and a discrete linearized dual problem.

2 Use of Galerkin orthogonality to introduce the interpolation error in the error representation.

3 Local estimates for the interpolation error.

4 Strong-stability estimates for the discrete dual problem.

We give more details for the above abstract case.

1 Error representation via a discrete dual problem:

$$\|e_h\|^2 = (e_h, e_h) = (e_h, A^*\varphi_h) = (Ae_h, \varphi_h),$$

where $e_h \equiv \pi_h u - U$, for $\pi_h u$ an interpolant of u in V_h, and the discrete dual problem with solution $\varphi_h \in V_h$ is defined by

$$(v, A^*\varphi_h) = (v, e_h), \quad \forall v \in V_h.$$

2 Galerkin orthogonality: Using $(Ae, v) = 0$, $\forall v \in V_h$, gives

$$\|e_h\|^2 = (A(\pi_h u - U), \varphi_h) = (\pi_h u - u, A^*\varphi_h).$$

3 Interpolation estimate:

$$\|u - \pi_h u\| \leq C_{\rm i}\|h^\alpha D^\alpha u\|,$$

where $C_{\rm i}$ is an interpolation constant.

4 Strong-stability estimate for the discrete dual problem:

$$\|A^*\varphi_h\| \leq S_{{\rm c},h}\|e_h\|,$$

where $S_{{\rm c},h}$ is discrete strong-stability factor.

Combining (1)–(4), we get an a priori error estimate:

$$\|u - U\| \leq C_{\mathrm{i}}(S_{\mathrm{c},h} + 1)\|h^{\alpha} D^{\alpha} u\|. \tag{5.5}$$

The interplay between the 'strong' norm, used in the strong stability involving A^*, and the corresponding 'weak' norm, used to estimate the interpolation error, is crucial.

Note that the stability of a continuous dual problem is used in the a posteriori error analysis whereas the stability of a discrete dual problem is used to prove the a priori error estimate. In both cases, the stability of the dual problems reflect the error accumulation and propagation properties of the discretization procedure.

5.4. Adaptive algorithms

Suppose the computational goal is to determine an approximate solution U such that $\|u - U\| \leq TOL$, where TOL is a given tolerance. The corresponding stopping criterion reads:

$$S_{\mathrm{c}} C_{\mathrm{i}} \|h^{\beta} R(U)\| \leq TOL, \tag{5.6}$$

which guarantees the desired error control via the a posteriori error estimate (5.4). The strategy of adaptive error control can be posed as a constrained nonlinear optimization problem: compute an approximation U satisfying (5.6) with minimal computational effort. In the case of Galerkin discretization, the control parameters are the local mesh size h and the local degree of the piecewise polynomials q and we seek h and q that minimize computational effort. We solve this problem iteratively, where the modification strategy indicates how to compute an improved iterate from the current iterate. The modification strategy is based on both the a posteriori error estimate (5.4) and the a priori error estimate (5.5). The mesh modification itself requires a mesh generator capable of generating a mesh with given mesh sizes. Mesh modification may also involve stretching and orientating the mesh. Such mesh generators in two and three dimensions are available today.

Adaptive algorithms using (5.6) as stopping criterion are reliable in the sense that by (5.4) the error control $\|u - U\| \leq TOL$ is guaranteed. The efficiency of the adaptive algorithm depends on the quality of the mesh-modification strategy.

5.5. The stability factors and interpolation constants

The stability factor S_{c} and the interpolation constant C_{i} in the a posteriori error estimate defining the stopping criterion have to be computed to give the error control a quantitative meaning.

The stability factor S_{c} depends in general on the particular solution being approximated, since it is defined in terms of the linearized continuous dual problem. In some cases, all solutions have the same stability factors. For example, for typical elliptic problems with analysis in the energy norm, $S_{\mathrm{c}} = S_{\mathrm{c},h} = 1$ with a suitable definition of norms. In general, this is not true. Analytic upper bounds on S_{c} often are too crude to be useful for quantitative error control. Hence, in the adaptive algorithms the stability factors S_{c} are approximated by solving the linearized continuous dual problem numerically. The amount of work required to compute S_{c} with sufficient accuracy is problem and solution dependent and depends on the degree of reliability desired. For complex problems, complete reliability cannot be reached, but the degree of reliability may be increased by spending more on the computation of the S_{c}.

The interpolation constants C_{i} depend on the shape of the elements, the local order of the polynomial approximation and the choice of norms, but not on the particular solution being approximated or the mesh size. Bounds for the interpolation constants C_{i} may be determined analytically or numerically from interpolation theory. Alternatively, once stability factors have been computed, the interpolation constants may be determined through calibration by numerically solving problems with known exact solutions.

5.6. *Error estimates for quadrature, discrete-solution, data and modelling errors*

A posteriori estimates of quadrature, discrete-solution and data-modelling errors are performed in a similar fashion to the analysis of the Galerkin-discretization error. The key difference is due to the fact that different perturbations accumulate at different rates. In particular, perturbations satisfying an orthogonality relation are connected to strong stability. For example, orthogonality is the basis of the Galerkin discretization and multigrid methods for discrete solutions. In general, weak stability must be used. A typical weak-stability estimate for the dual continuous problem takes the form

$$\|\varphi\| \leq \tilde{S}_{\mathrm{c}}\|e\|,$$

where the dual solution φ is estimated in terms of the data e and the weak-stability factor. The corresponding a posteriori error estimate takes the form

$$\|u - U\| \leq \tilde{S}_{\mathrm{c}} C_{\mathrm{i}} \|R(U)\|. \tag{5.7}$$

Note that the factor h^{β} that resulted from the use of strong stability is missing.

6. Piecewise-polynomial approximation

In this section, we review results on piecewise-polynomial interpolation that are used below. For the sake of simplicity, we limit ourselves to discontinuous piecewise-constant approximation and continuous piecewise-linear approximation on an interval $I \equiv (a, b)$ in space or time. Higher-order results are similar. Two-dimensional analogues are given below.

Assuming first that $I \equiv (0,1)$ is a space interval, let $0 \equiv x_0 < x_1 < x_2 < \cdots < x_{M+1} \equiv 1$ be a subdivision of I into subintervals $I_j \equiv (x_{j-1}, x_j)$ of length $h_j \equiv x_j - x_{j-1}$. We use the notation $T_h \equiv \{I_j\}$ for the corresponding mesh and define its mesh function $h(x)$ by $h(x) = h_j$ for x in I_j.

We also consider the situation in which I is a time interval, for example, $I \equiv (0, \infty)$. In this case, the mesh is given by a sequence of discrete time levels $0 \equiv t_0 < t_1 < \cdots < t_n < \cdots$, with corresponding time intervals $I_n \equiv (t_{n-1}, t_n)$, time steps $k_n \equiv t_n - t_{n-1}$ and mesh function $k(t)$ defined by $k(t) = k_n$ for t in I_n. We denote the corresponding mesh by T_k.

We let W_h denote the space of discontinuous piecewise-constant functions $v = v(x)$ on I, that is, v is constant on each subinterval I_j. We define the interpolant $\pi_h v \in W_h$ of an integrable function v by

$$\int_{I_j} (v - \pi_h v)\, \mathrm{d}x = 0, \tag{6.1}$$

that is,

$$\pi_h v = \frac{1}{h_j} \int_{I_j} v(x)\, \mathrm{d}x \text{ on } I_j \tag{6.2}$$

is the average of v on each element. It is easy to show that for $1 \le p \le \infty$,

$$\|v - \pi_h v\|_p \le \|hv'\|_p, \tag{6.3}$$

where $\|\cdot\|_p$ denotes the usual $L^p(I)$ norm.

We let V_h denote the space of functions that are continuous on I and linear on each subinterval I_j, $j = 1, ..., M+1$. We denote by V_h^0 the subspace of functions $v \in V_h$ satisfying $v(0) = v(1) = 0$. For a continuous function v on I, we define the nodal interpolant $\pi_h v$ in V_h by

$$\pi_h v(x_j) = v(x_j),\ j = 0, \ldots, M+1. \tag{6.4}$$

Then, there are constants $C_{\mathrm{i},k}$ such that for $1 \le p \le \infty$,

$$\|h^{-2}(v - \pi_h v)\|_p \le C_{\mathrm{i},1} \|v''\|_p, \tag{6.5}$$

$$\|h^{-1}(v - \pi_h v)\|_p \le C_{\mathrm{i},2} \|v'\|_p, \tag{6.6}$$

$$\|h^{-1}(v - \pi_h v)'\|_p \le C_{\mathrm{i},3} \|v''\|_p. \tag{6.7}$$

Remark 1 The interpolation constants $C_{\mathrm{i},k}$ have the values 1/8, 1 and 1/2 for $k = 1, 2, 3$ respectively.

Remark 2 Below, we use weighted L^2 norms. Given a continuous positive function $a(x)$, a weighted L^2 norm is defined by

$$\| \cdot \|_a \equiv \| \sqrt{a}(\cdot) \|_2.$$

The interpolation results (6.3), (6.5)–(6.7) hold in the weighted norm with C_i depending on $\max_j\,(\max_{I_j} a / \min_{I_j} a)$.

7. An elliptic model problem in one dimension

As a model case, we consider a two-point boundary-value problem: find $u(x)$ such that

$$\begin{aligned} -(a(x)u')' + b(x)u' + c(x)u &= f(x), \quad x \in I = (0,1), \\ u(0) &= 0, \quad u(1) = 0, \end{aligned} \tag{7.1}$$

where $a(x)$, $b(x)$ and $c(x)$ are given coefficients with $a(x) > 0$, and $f = f(x)$ is a given source term. This is a model for a stationary diffusion–convection–absorption process in one dimension. If $|b|/a$ is not large, then this problem has elliptic character, while if $|b|/a$ is large then the character is hyperbolic. We first consider the elliptic case with $b = 0$ for simplicity, and comment on the hyperbolic case with $|b|/a$ large in Remark 3 below.

In the elliptic case, we first assume $c = 0$. The variational formulation of (7.1) with $b = c = 0$, resulting from integration by parts, takes the form

$$\int_I a u' v' \,\mathrm{d}x = \int_I f v \,\mathrm{d}x, \quad \forall v \in V_h^0. \tag{7.2}$$

The cG(1) method for (7.1) reads: find $U \in V_h^0$ such that

$$\int_I a U' v' \,\mathrm{d}x = \int_I f v \,\mathrm{d}x, \quad \forall v \in V_h^0. \tag{7.3}$$

This expresses the Galerkin orthogonality condition on the residual error which is apparent upon subtracting (7.3) from (7.2) to obtain

$$\int_I a(u - U)' v' \,\mathrm{d}x = 0, \quad \forall v \in V_h^0. \tag{7.4}$$

Representing the finite-element solution as

$$U(x) = \sum_{j=1}^{M} \xi_j \varphi_j(x), \tag{7.5}$$

where $\{\varphi_j\}_{j=1}^M$ is the set of basis functions associated with the interior nodes, we find that (7.3) is equivalent to a matrix equation for the vector $\xi = (\xi_j)$:

$$A\xi = b, \tag{7.6}$$

where $A = (a_{ij})$ is the $M \times M$ stiffness matrix with coefficients

$$a_{ij} \equiv \int_I a \varphi_j' \varphi_i' \,\mathrm{d}x, \tag{7.7}$$

and $b = (b_i)$ is the load vector with elements

$$b_i \equiv \int_I f\varphi_i \, \mathrm{d}x. \tag{7.8}$$

The system matrix A is positive definite and tridiagonal and is easily solved by Gaussian elimination to give the approximate solution U.

The basic issue is the size of the error $u - U$. We first prove an a posteriori error estimate and then an a priori error estimate.

7.1. A posteriori error estimate in the energy norm

We prove an a posteriori estimate of the error $e \equiv u - U$ in the *energy norm* $\|\cdot\|_E$ defined for functions v with $v(0) = v(1) = 0$ by

$$\|v\|_E = \|v'\|_a \equiv \left(\int_I a(v')^2 \, \mathrm{d}x\right)^{\frac{1}{2}}.$$

Using Galerkin orthogonality (7.3) by choosing $v = \pi_h e \in V_h^0$, we obtain the error representation:

$$\begin{aligned} \|e'\|_a^2 = & \int_I ae'e' \, \mathrm{d}x = \int_I au'e' \, \mathrm{d}x - \int_I aU'e' \, \mathrm{d}x = \int_I fe \, \mathrm{d}x - \int_I aU'e' \, \mathrm{d}x \\ = & \int_I f(e - \pi_h e) \, \mathrm{d}x - \int_I aU'(e - \pi_h e)' \, \mathrm{d}x \\ = & \int_I f(e - \pi_h e) \, \mathrm{d}x - \sum_{i=1}^{M+1} \int_{I_j} aU'(e - \pi_h e)' \, \mathrm{d}x. \end{aligned} \tag{7.9}$$

In this case, the solution of the dual problem is the error itself. We integrate by parts over each subinterval I_j in the last term, and use the fact that all the boundary terms disappear, to get

$$\|e'\|_a^2 = \int_I R(U)(e - \pi_h e)\mathrm{d}x \le \|hR(U)\|_{\frac{1}{a}} \|h^{-1}(e - \pi_h e)\|_a,$$

where $R(U)$ is the residual defined on each subinterval I_j by

$$R(U) \equiv f + (aU')'.$$

Recalling (6.6) for the weighted L^2 norm, one proves:

Theorem 1 The finite-element solution U satisfies

$$\|u - U\|_E \le C_{\mathrm{i}} \|hR(U)\|_{\frac{1}{a}}. \tag{7.10}$$

7.2. An adaptive algorithm

We design an adaptive algorithm for automatic control of the energy-norm error $\|u - U\|_E$ using the a posteriori error estimate (7.10) as follows:

1 Choose an initial mesh $T_{h^{(0)}}$ of mesh size $h^{(0)}$.

2 Compute the corresponding cG(1) finite-element solution $U^{(0)}$ in $V_{h^{(0)}}$.

3 Given a computed solution $U^{(m-1)}$ in $V_{h^{(m-1)}}$ on a mesh with mesh size $h^{(m-1)}$, stop if

$$C_{\mathrm{i}}\|h^{(m-1)}R(U^{(m-1)})\|_{\frac{1}{a}} \leq TOL. \tag{7.11}$$

4 If not, determine a new mesh $T_{h^{(m-1)}}$ with mesh function $h^{(m-1)}$ of maximal size such that

$$C_{\mathrm{i}}\|h^{(m-1)}R(U^{(m-1)})\|_{\frac{1}{a}} = TOL \tag{7.12}$$

and continue.

We note that (7.11) is the stopping criterion and (7.12) defines the mesh-modification strategy. By Theorem 1, it follows that the error $\|u - U\|_E$ is controlled to the tolerance TOL if the stopping criterion (7.11) is reached with $U = U^{(m-1)}$. The relation (7.12) defines the new mesh size $h^{(m-1)}$ by maximality, that is, we seek a mesh function $h^{(m-1)}$ as large as possible (to maintain efficiency) such that (7.12) holds. In general, maximality in $\|\cdot\|$ is obtained by the 'equidistribution' of error such that the error contributions from the individual intervals I_j are kept equal.

Equidistribution of the error results in the equation

$$a(x_j)^{-1}(h_j^{(m)}\|R(U^{(m-1)})\|_{L_\infty I_j^{(m)}})^2 h_j^{(m)} = \frac{TOL^2}{N^{(m)}}, \quad j = 1, \ldots, N^{(m)}, \tag{7.13}$$

where $N^{(m)}$ is the number of intervals in $T_{h^{(m)}}$. The equation reflects the fact that the total error is given by the sum of the errors from each interval, and so the error on each interval must be a fraction of the total error. In practice, this nonlinear equation is simplified by replacing $N^{(m)}$ by $N^{(m-1)}$.

Example 1. Consider problem (7.1) with $a(x) = x + \varepsilon$, $\varepsilon = 0.01$, $b = c = 0$ and $f(x) \equiv 1$. Because the diffusion coefficient a is small near $x = 0$, the solution u and its derivatives u' and u'' there change rapidly with x; see Figure 1a. To compute u, we use the code Femlab which contains an implementation of the adaptive algorithm just described. Figure 1b shows the residual $R(U)$ of the computed solution U, and Figure 1c shows, the local mesh size $h(x)$ when the stopping criterion with $TOL = 0.05$ was reached. Note that the mesh size is small near $x = 0$ where the residual is large.

7.3. A priori error estimate in the energy norm

We now prove an a priori energy-norm error estimate for (7.3) by comparing the Galerkin-discretization error to the interpolation error. Using Galerkin

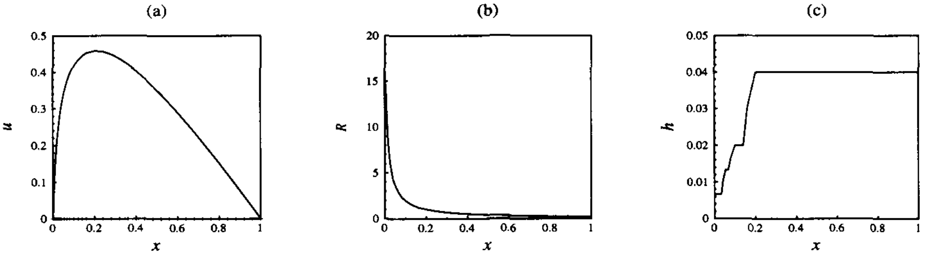

Fig. 1. Solution, residual and mesh size for Example 1

orthogonality (7.4) with $v = U - \pi_h u$, we obtain

$$\int_I a(u-U)'(u-U)'\,\mathrm{d}x = \int_I a(u-U)'(u-\pi_h u)'\,\mathrm{d}x \le \|u'-U'\|_a \|(u-\pi_h u)'\|_a,$$

so

$$\|u' - U'\|_a \le \|(u - \pi_h u)'\|_a \tag{7.14}$$

This shows that the Galerkin approximation is optimal because its error in the energy norm is less than the error of the interpolant. Together with (6.7), this proves:

Theorem 2 The finite-element solution U satisfies

$$\|u' - U'\|_a \le C_{\mathrm{i}} \|hu''\|_a. \tag{7.15}$$

Remark 3 It is easy to show that

$$C_{\mathrm{i}} \|hR(U)\|_{\frac{1}{a}} \le CC_{\mathrm{i}} \|hu''\|_a,$$

with C a constant depending on a, indicating that the a posteriori energy error estimate is optimal in the same sense as the a priori estimate.

7.4. A posteriori error estimate in the L^2 norm

We prove an a posteriori error estimate in the L^2-norm, allowing the absorption coefficient c in (7.1) to be nonzero. The extension of (7.3) to this case is direct by including $\int_I cUv\,\mathrm{d}x$ on the left-hand side. We introduce the dual problem

$$\begin{aligned} -(a\varphi')' + c\varphi &= e, \quad x \in I, \\ \varphi(0) &= 0, \quad \varphi(1) = 0, \end{aligned} \tag{7.16}$$

which takes the same form as the original problem (7.1). We use Galerkin orthogonality (7.3), by choosing $v = \pi_h e \in V_h^0$, to get

$$\begin{aligned}\|e\|_2^2 &= \int_I e(-(a\varphi')' + c\varphi)\,\mathrm{d}x = \int_I (ae'\varphi' + ce\varphi)\,\mathrm{d}x \\ &= \int_I (au'\varphi' + cu\varphi)\,\mathrm{d}x - \int_I (aU'\varphi' + cU\varphi)\,\mathrm{d}x \\ &= \int_I f\varphi\,\mathrm{d}x - \int_I (aU'\varphi' + cU\varphi)\,\mathrm{d}x \\ &= \int_I f(\varphi - \pi_h\varphi)\,\mathrm{d}x - \sum_{i=1}^{M+1} \int_{I_j} (aU'(\varphi - \pi_h\varphi)' + cU(\varphi - \pi_h\varphi))\,\mathrm{d}x.\end{aligned}$$

We now integrate by parts over each subinterval I_j, using the fact that all the boundary terms disappear, to get

$$\|e\|_2^2 \leq \|h^2 R(U)\|_2 \|h^{-2}(\varphi - \pi_h\varphi)\|_2,$$

where $R(U) \equiv f + (aU')' + cU$ on each subinterval. Using (6.3) and defining the strong-stability factor S_c by

$$S_c \equiv \max_{g \in L^2(I)} \frac{\|\psi_g''\|_2}{\|g\|_2}, \tag{7.17}$$

where ψ_g satisfies

$$\begin{aligned} -(a\psi_g')' + c\psi_g &= g, \quad x \in I, \\ \psi_g(0) &= 0, \quad \psi_g(1) = 0, \end{aligned} \tag{7.18}$$

we obtain:

Theorem 3 The finite-element solution U satisfies

$$\|u - U\|_2 \leq S_c C_i \|h^2 R(U)\|_2. \tag{7.19}$$

Example 2. In Figure 2a, we plot the computed solution in the case $a = 0.01$, $c = 1$ and $f(x) = 1/x$ with L^2 error control based on (7.19) with $TOL = .01$. The residual and mesh size are plotted in Figures 2b and 2c. In this example, there are two sources of singularities in the solution. First, because the diffusion coefficient a is small, the solution may have boundary layers; second, the source term f is large near $x = 0$. The singularity in the data f enters only through the residual, while the smallness of a enters both through the residual and through the stability factor S_c. The adaptive algorithm computes the stability factor S_c by solving the dual problem (7.16) with e replaced by an approximation obtained by subtracting approximate solutions on two different grids. In this example, $S_c \sim 37$.

7.5. A priori error estimate in the L^2 norm

We now prove an a priori error estimate in the L^2 norm, assuming for simplicity that the mesh size h is constant and that $c = 0$.

Theorem 4 The finite-element solution U satisfies

$$\|u - U\|_2 \leq C_i S_c \|h(u - U)'\|_2 \leq C_i S_c \|h^2 u''\|_2, \tag{7.20}$$

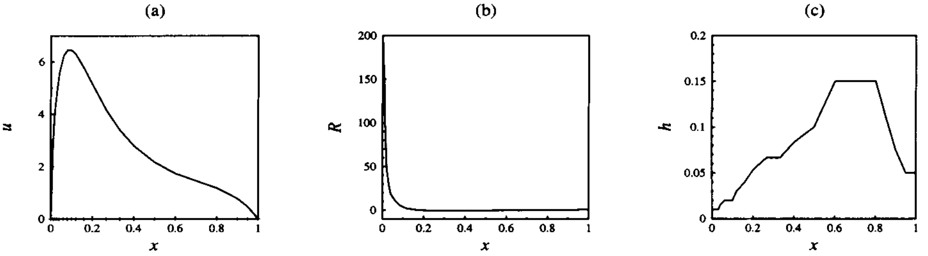

Fig. 2. Approximation, residual and mesh size for Example 2

where $S_c \equiv \max_{g \in L^2(I)} \|\psi_g''\|_a / \|g\|_2$, with ψ_g satisfying (7.18).

Proof. By (7.4) and (6.7), for φ satisfying (7.16) with $c = 0$, we have

$$\begin{aligned} \|e\|_2^2 = &\ \textstyle\int_I ae'\varphi' \, \mathrm{d}x = \int_I ae'(\varphi - \pi_h\varphi)' \, \mathrm{d}x \\ \leq &\ \|he'\|_a \|h^{-1}(\varphi - \pi_h\varphi)'\|_a \leq C_{\mathrm{i}} \|he'\|_a \|\varphi''\|_a. \end{aligned}$$

The proof is finished by noting that multiplying the energy-norm error estimate by h gives

$$\|he'\|_a \leq C_{\mathrm{i}} \|h^2 u''\|_a. \tag{7.21}$$

□

This estimate generalizes to the case of variable h assuming that the mesh size h does not change too rapidly from one element to the next, (cf. Eriksson (1994)).

7.6. Data and modelling errors

We make an a posteriori estimate of data and modelling errors. Suppose that $a(x)$ and $f(x)$ in (7.3) are approximations of the correct coefficient $\hat{a}(x)$ and data $\hat{f}(x)$ and let $\hat{u}$ be the corresponding correct solution. We seek an a posteriori error estimate of the total error $\hat{e} \equiv \hat{u} - U$ including Galerkin-discretization, data and modelling errors. We start from a modified form of the error representation (7.9),

$$\begin{aligned} \|(\hat{u} - U)'\|_{\hat{a}}^2 &= \int_I f(\hat{e} - \pi_h\hat{e}) \, \mathrm{d}x - \sum_{j=1}^{M+1} \int_{I_j} aU'(\hat{e} - \pi_h\hat{e})' \, \mathrm{d}x \\ &\quad + \int_I (\hat{f} - f)\hat{e} \, \mathrm{d}x - \sum_{j=1}^{M+1} \int_{I_j} (\hat{a} - a)U'\hat{e}' \, \mathrm{d}x \\ &\equiv I + II - III, \end{aligned}$$

with the obvious definition of I, II and III. The first term I is estimated as above. For the new term III, we have

$$III \leq C_{\mathrm{i}} \|(\hat{a} - a)\hat{U}'\|_{\frac{1}{\hat{a}}} \|e'\|_{\hat{a}}.$$

Similarly, integration by parts gives

$$II \leq \|F - \hat{F}\|_{\frac{1}{\hat{a}}} \|e'\|_{\hat{a}},$$

where $F' = f$, $\hat{F}' = \hat{f}$ and $F(0) = \hat{F}(0) = 0$. Altogether, we obtain:

Theorem 5 The finite-element solution U satisfies

$$\|\hat{u}' - U'\|_{\hat{a}} \leq C_{\mathrm{i}}(\|hR(U)\|_{\frac{1}{\hat{a}}} + \|F - \hat{F}\|_{\frac{1}{\hat{a}}} + \|(\hat{a} - a)U'\|_{\frac{1}{\hat{a}}}). \qquad (7.22)$$

An adaptive algorithm for control of both Galerkin and data-modelling errors can be based on (7.22). It is natural to assume that $\|\hat{a} - a\|_\infty \leq \mu$ or $\|(\hat{a} - a)\hat{a}^{-1}\|_\infty \leq \mu$, corresponding to an absolute or relative error in $\hat{a}$ on the level μ. In the first case, we obtain $\|(\hat{a} - a)U'\|_{\frac{1}{\hat{a}}} \leq \mu\|U'\|_{\frac{1}{\hat{a}}}$, and in the second case, $\|(\hat{a} - a)U'\|_{\frac{1}{\hat{a}}} \leq \mu\|U'\|_{\hat{a}}$. μ is supplied by the user while the relevant norm on U is computed by the program. For example, for the problem in Example 2 we find that $\|U'\|_{\frac{1}{\hat{a}}} = 13.3531$ while $\|U'\|_{\hat{a}} = 0.5406$.

Remark 4 If $|b|/a$ is large then the problem (7.1) has a hyperbolic character. If $a < h$ then a modified Galerkin method with improved stability properties is used which is called the streamline diffusion method. The modifications consist of a weighted least-squares stabilization that gives extra control of the residual $R(U)$ and a modification of the viscosity coefficient a. L^2 error estimates for this method are derived similarly to the case with $b = 0$. The resulting L^2 a posteriori error estimate has essentially the form (7.19), where the stability constant S_{c} contains a dependence on the viscosity a. In the generic case with a constant, we have $S_{\mathrm{c}} \sim \frac{1}{a}$. The result of using strong stability and Galerkin orthogonality is a factor $\frac{h^2}{a}$ coupled with the residual $R(U)$. In a direct approach that uses weak stability, the result does not contain the factor $\frac{h^2}{a}$. Thus, an improvement results if $a > h^2$. In particular, if $a > h$ then the standard unmodified Galerkin method may be used and the above analysis applies. The condition $a > h$ may be satisfied on the last mesh in the sequence of meshes used in the adaptive process. In this case, the streamline diffusion modification is used only on the initial coarse meshes. Details of this extension to hyperbolic convection–diffusion problems are given in Eriksson and Johnson (1993), (to appear).

Example 3. Consider problem (7.1) with $a(x) = 0.02$, $b(x) = 1$, $c(x) = 0$ and $f(x) = 1$. In Figure 3, we plot the computed solution together with the residual and mesh size obtained using an adaptive algorithm based on an

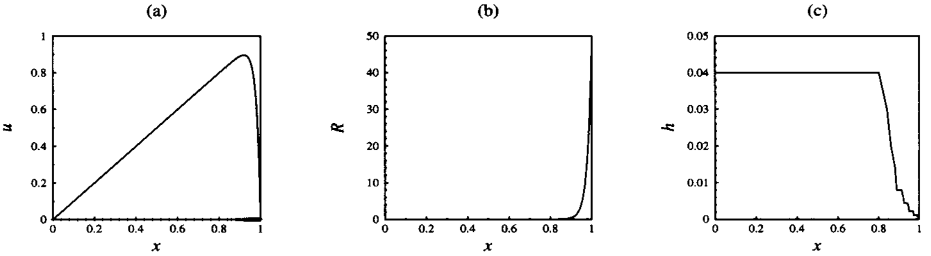

Fig. 3. Solution, residual and mesh size for Example 3

a posteriori error estimate of the form (7.19) with $TOL = .02$. Notice the singularity in u in the boundary layer near $x = 1$.

8. Basic time-dependent model problems

As a first example, we consider the scalar linear initial-value problem: find $u = u(t)$ such that

$$\begin{aligned} u' + a(t)u &= f(t), \quad t > 0, \\ u(0) &= u_0, \end{aligned} \tag{8.1}$$

where $a(t)$ is a given coefficient, $f(t)$ is a given source term and $v' = \frac{dv}{dt}$ now denotes the time derivative of v. The exact solution $u(t)$ is given by the formula

$$u(t) = \mathrm{e}^{-A(t)}u_0 + \int_0^t \mathrm{e}^{-(A(t)-A(s))} f(s)\,\mathrm{d}s, \tag{8.2}$$

where $A' = a$ and $A(0) = 0$, from which we can draw some conclusions about the dependence of u on a. In general, the exponential factors may become large with time. However, if $a(t) \geq 0$ for all t, then $A(t) \geq 0$ and $A(t) - A(s) \geq 0$ for all $t \geq s$, and both u_0 and f are multiplied by quantities that are less than or equal to one. We shall see that if $a(t) \geq 0$, then Galerkin-discretization errors accumulate in such a way that accurate long-time computation is possible. The problem (8.1) with $a(t) \geq 0$ is a model for a class of parabolic problems that includes generalizations of (8.1) with the coefficient a replaced by $-\nabla \cdot (\alpha \nabla)$ with $\alpha \geq 0$. The analysis for the case $a \geq 0$ allowing long-time integration without error accumulation extends directly to this more complex case.

For the sake of simplicity, we consider the dG(0) method, which reads: find U in W_k such that for all polynomials v of degree 0 on I_n,

$$\int_{I_n} (U' + a(t)U)v\mathrm{d}t + [U_{n-1}]v_{n-1}^+ = \int_{I_n} fv\,\mathrm{d}t, \tag{8.3}$$

where $[v_n] = (v_n^+ - v_n^-)$, $v_n^\pm = \lim_{s\to\pm 0} v(t_n + s)$ and $U_0^- = u_0$. We note that (8.3) says that the 'sum' of the residual $U_t' + a(t)U - f$ in I_n, and the 'jump'

$[U_{n-1}]$ is orthogonal to all discrete test functions. Since U is a constant on I_n, $U' \equiv 0$ on I_n.

If U_n denotes the constant value of U on the time interval I_n, then the dG(0) method (8.3) satisfies

$$U_n - U_{n-1} + U_n \int_{I_n} a \,\mathrm{d}t = \int_{I_n} f \,\mathrm{d}t, \qquad n = 1, 2, \ldots, \tag{8.4}$$

where $U_0 = u_0$. The classical backward Euler method is thus the dG(0) method with the rectangle rule applied to the integrals. We assume that if $a(t)$ is negative, then the time step is small enough that $|\int_{I_n} a \,\mathrm{d}t| < 1$, in which case (8.4) defines U_n uniquely. We use the notation $\|v\|_I \equiv \max_{t\in[0,T]} |v(t)|$, where $I \equiv (0, T)$ is a given time interval.

8.1. An a posteriori error estimate

To derive an a posteriori error estimate for the error $e_N \equiv u(t_N) - U_N$, $N \geq 1$, we introduce the continuous dual 'backward' problem,

$$\begin{aligned} -\varphi' + a(t)\varphi &= 0, \quad t \in (0, t_N), \\ \varphi(t_N) &= e_N, \end{aligned} \tag{8.5}$$

with solution given by $\varphi(t) = \mathrm{e}^{A(t)-A(t_N)} e_N$. Integration by parts over each subinterval I_n gives

$$\begin{aligned} e_N^2 = {} & e_N^2 + \textstyle\sum_{n=1}^N \int_{I_n} e(-\varphi' + a\varphi) \,\mathrm{d}t \\ = {} & \textstyle\sum_{n=1}^N \int_{I_n} (e' + ae)\varphi \,\mathrm{d}t + \sum_{n=1}^{N-1} [e_n]\varphi_n^+ + (u_0 - U_0^+)\varphi_0^+ \\ = {} & \textstyle\sum_{n=1}^N (\int_{I_n} (f - aU)\varphi \,\mathrm{d}t - [U_{n-1}]\varphi_{n-1}^+), \end{aligned} \tag{8.6}$$

where in the last step we use the facts that $U' = 0$ on each I_n and $U_0^- = u_0$. Now we use Galerkin orthogonality (8.3) by taking $v = \pi_k\varphi$ with

$$\int_{I_n} (\varphi - \pi_k\varphi) \,\mathrm{d}t = 0, \quad n = 1, \ldots, N,$$

to obtain the error representation formula:

$$e_N^2 = \sum_{n=1}^N \left(\int_{I_n} (f - aU)(\varphi - \pi_k\varphi) \,\mathrm{d}t - [U_{n-1}](\varphi - \pi_k\varphi)_{n-1}^+ \right).$$

Using (6.3), we obtain

$$\begin{aligned} e_N^2 \leq {} & \textstyle\int_0^{t_N} |\varphi'| \mathrm{d}t \ \max_{n=1,\ldots,N}(|[U_{n-1}]| + \|k(f - aU)\|_{I_n}) \\ \leq {} & S_\mathrm{c}(t_N)(|[U_{n-1}]| + \|k(f - aU)\|_{I_n})|e_N|, \end{aligned} \tag{8.7}$$

where $S_\mathrm{c}(t_N)$ is the stability factor defined by

$$S_\mathrm{c}(t_N) \equiv \max_{\varphi(t_N)} \frac{\int_0^{t_N} |\varphi_t| \mathrm{d}t}{|\varphi(t_N)|}. \tag{8.8}$$

To complete the proof of the a posteriori error estimate, we need to estimate $S_c(t_N)$. The following lemma presents such a stability estimate in both the general case and the dissipative case when $a(t) \geq 0$ for all t. We also state an estimate for φ itself.

Lemma 1 If $|a(t)| \leq \mathbb{A}$ for $t \in (0, t_N)$, then φ satisfies for all $t \in (0, t_N)$:

$$|\varphi(t)| \leq \exp(\mathbb{A}t_N)|e_N|, \tag{8.9}$$

and

$$S_c(t_N) \leq \mathbb{A}t_N \exp(\mathbb{A}t_N). \tag{8.10}$$

If $a(t) \geq 0$ for all t, then φ satisfies for all $t \in (0, t_N)$:

$$|\varphi(t)| \leq |e_N|, \tag{8.11}$$

and

$$S_c(t_N) \leq 1. \tag{8.12}$$

Proof. The first and second estimates follow from the boundedness assumption on a. The third estimate follows from the fact that $A(t_N) - A(t)$ is non-negative for $t \leq t_N$. Further, since a is non-negative,

$$\begin{aligned} \textstyle\int_I |\varphi'| \mathrm{d}t = &\ |e_N| \textstyle\int_I a(t) \exp(A(t_N) - A(t)) \mathrm{d}t \\ = &\ |e_N|(1 - \exp(A(0) - A(t_N))) \leq |e_N|, \end{aligned}$$

which completes the proof. □

We insert the strong-stability estimates (8.10) or (8.12) into (8.7) and obtain the a posteriori error estimate:

Theorem 6 The finite-element solution U satisfies for $N = 1, 2, \ldots$

$$|u(t_N) - U_N| \leq S_c(t_N)|kR(U)|_{(0,t_N)},$$

where

$$R(U) \equiv \frac{|U_n - U_{n-1}|}{k_n} + |f - aU|_{I_n}, \quad t \in I_n.$$

8.2. An a priori error estimate

The a priori error estimate for (8.3) reads as follows:

Theorem 7 If $|a(t)| \leq \mathbb{A}$ for all t, then there is a constant $C > 0$ such that U satisfies for $N = 1, 2, \ldots$.

$$|u(t_N) - U_N| \leq C\mathbb{A}t_N \exp(C\mathbb{A}t_N)|ku'|_I,$$

and if $a(t) \geq 0$ for all t, then for $N = 1, 2, \ldots,$

$$|u(t_N) - U_N| \leq |ku'|_{(0,t_N)}. \tag{8.13}$$

We note the optimal nature of the estimate compared to interpolation in the case $a(t) \geq 0$.

Proof. We introduce the discrete dual backward problem: find $\Phi \in W_k$ such that for $n = N, N-1, \dots, 1$,

$$\int_{I_n} (-\Phi' + a(t)\Phi)v \mathrm{d}t - [\Phi_n]v_n^- = 0, \quad \forall v \in W_k, \tag{8.14}$$

where $\Phi_N^+ = (\pi_k u - U)_N^-$. It suffices to estimate the 'discrete' error $\bar{e} \equiv \pi_k u - U$ in W_k since $u - \pi_k u$ is already known. With the choice $v = \bar{e}$, the Galerkin orthogonality allows U to be replaced by u and we obtain the following representation:

$$\begin{aligned}
|\bar{e}_N^-|^2 &= \sum_{n=1}^{N} \int_{I_n} (-\Phi' + a(t)\Phi)\bar{e} \mathrm{d}t - \sum_{n=1}^{N-1} [\Phi_n]\bar{e}_n^- + \Phi_N^- \bar{e}_N^- \\
&= \sum_{n=1}^{N} \int_{I_n} (-\Phi' + a(t)\Phi)(\pi_k u - u) \mathrm{d}t \\
&\quad - \sum_{n=1}^{N-1} [\Phi_n](\pi_k u - u)_n^- + \Phi_N^-(\pi_k u - u)_N^- \\
&= -\int_I (a\Phi(u - \pi_k u)) \mathrm{d}t + \sum_{n=1}^{N-1} [\Phi_n](u - \pi_k u)_n^- - \Phi_N^-(u - \pi_k u)_N^-,
\end{aligned}$$

where we use $\Phi' = 0$ on each time interval. Recalling (6.3), we get the desired result follows from a lemma expressing the weak and strong stability of the discrete dual problem (8.14). □

Lemma 2 When $|a(t)| \leq \mathbb{A}$ for all t, then there is a constant $C > 0$ such that the solution of the discrete dual problem (8.14) satisfies

$$|\Phi_n^-| \leq \exp(C\mathbb{A}t_N)|\bar{e}_N^-|, \tag{8.15}$$

$$\sum_{n=1}^{N-1} |[\Phi_n]| \leq C\mathbb{A}t_N \exp(C\mathbb{A}t_N)|\bar{e}_N^-|, \tag{8.16}$$

$$\sum_{n=1}^{N} \left| \int_{I_n} a|\Phi_n| \mathrm{d}t \right| \leq C\mathbb{A}t_N \exp(C\mathbb{A}t_N)|\bar{e}_N^-|. \tag{8.17}$$

If $a(t) \geq 0$ for all t, then

$$|\Phi_n^-| \leq |\bar{e}_N^-|, \tag{8.18}$$

$$\sum_{n=1}^{N-1} |[\Phi_n]| \leq |\bar{e}_N^-|, \tag{8.19}$$

$$\sum_{n=1}^{N}\left|\int_{I_n} a|\Phi_n|\mathrm{d}t\right| \le |\bar{e}_N^-|. \tag{8.20}$$

Proof. The discrete dual problem (8.14) takes the form

$$\begin{aligned} -\Phi_{n+1}+\Phi_n+\Phi_n\int_{I_n} a(t)\mathrm{d}t &= 0, \quad n=N,N-1,\dots,1,\\ \Phi_{N+1} &= \bar{e}_N^-, \end{aligned}$$

where Φ_n denotes the value of Φ on I_n, so

$$\Phi_n = \prod_{j=n}^{N}\left(1+\int_{I_j} a\,\mathrm{d}t\right)^{-1}\Phi_{N+1}.$$

In the case where a is bounded, the results follow from standard estimates. When a is nonnegative, this proves (8.18) immediately. To prove (8.19), we assume without loss of generality that Φ_{N+1} is positive, so the sequence Φ_n decreases when n decreases, and

$$\sum_{n=1}^{N}|[\Phi_n]| = \sum_{n=1}^{N}[\Phi_n] = \Phi_{N+1}-\Phi_1 \le |\Phi_{N+1}|.$$

Finally, (8.20) follows from the discrete equation. □

We note that the a priori error estimate (8.13) is optimal compared to interpolation in the case $a \ge 0$.

Remark 5 It is important to compare the general results of Theorem 6 and Theorem 8.13, when a is only known to be bounded, to the result for dissipative problems with $a \ge 0$. In the first case, the errors can accumulate at an 'exponential' rate, and, depending on $\mathbb{A}$, $S_c(t_N)$ can become so large that controlling the error is no longer possible. In the case $a \ge 0$, there is no accumulation of error so accurate computation is possible over arbitrarily long times. Note that we do not require $a(t)$ to be positive and bounded away from zero; it is enough to assume that a is non-negative.

Example 4. Consider the dissipative problem $u' + u = \sin t$, $u(0) = 1$ with solution $u(t) = 1.5\mathrm{e}^{-t} + .5(\sin t - \cos t)$. We compute with dG(0) and plot the solution and the approximation in Figure 4a. The approximation is computed with an error tolerance of .001. In Figure 4b, we plot $S_c(t)$ versus time. Note that $S_c(t)$ tends to 1 as t increases, indicating that the numerical error does not grow significantly with time, and accurate computations can be made over arbitrarily long time intervals.

Example 5. We now consider the problem $u' - u = 0$, $u(0) = 1$ with solution $u(t) = \mathrm{e}^t$. We compute with dG(0) keeping the error below .025. Since the

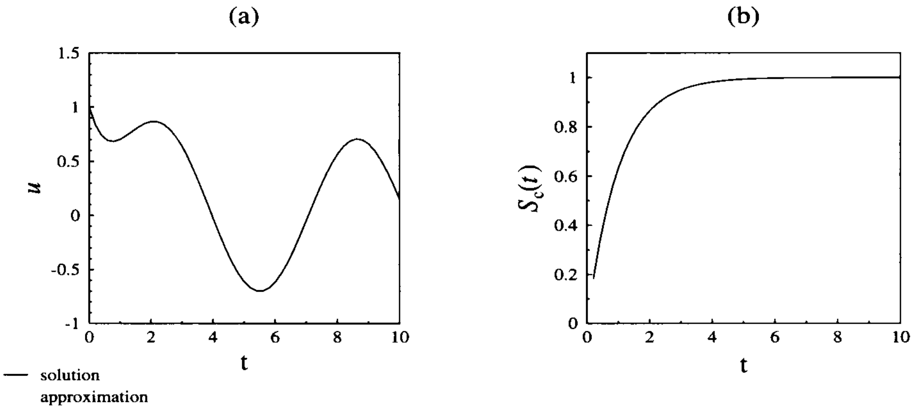

Fig. 4. Solution, approximation and stability factor for Example 4

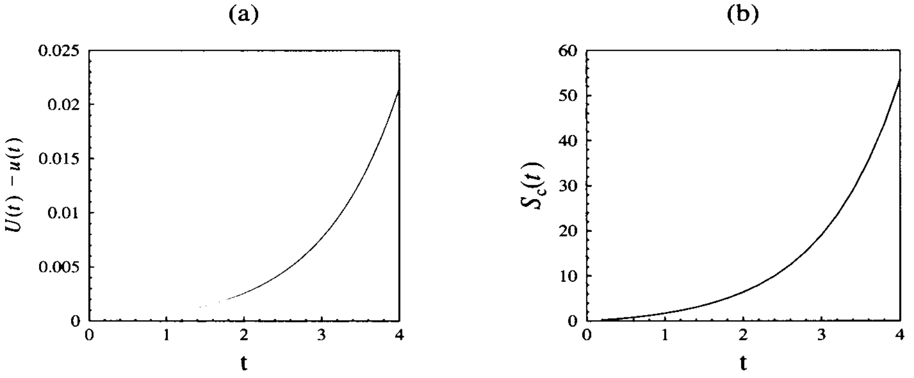

Fig. 5. Error and stability factor for Example 5

problem is not dissipative, we expect to see the error grow. The difference $U(t) - u(t)$ is plotted in Figure 5a and the exponential growth rate is clearly visible. Given a certain amount of computational power, for example, a fixed precision or a fixed amount of computing time, there is some point in time at which accurate computation is no longer possible. $S_c(t)$ is plotted in Figure 5b, and we note that it reflects the rate of instability precisely.

8.3. Adaptive error control

An adaptive algorithm based on the a posteriori error estimate takes the form: determine the time steps k_n so that

$$\hat{S}_c(t_N)(|U_n - U_{n-1}| + k_n|f - aU|_{I_n}) = TOL, \quad n = 1, ..., N,$$

where $\hat{S}_c(t_N) \equiv \max_{1 \le n \le N} S_c(t_n)$. This guarantees that

$$|u(t_n) - U_n| \le TOL, \quad n = 1, \cdots N.$$

As mentioned above, $\hat{S}_c(t_N)$ is approximated in an auxiliary computation solving the backward problem with chosen initial data; see below and [25] and [8] for more details.

Example 6. We consider a more complicated problem,

$$\begin{aligned} &u' + (.25 + 2\pi \sin(2\pi t))u = 0, \qquad t > 0, \\ &u(0) = 1, \end{aligned}$$

with solution

$$u(t) = \exp(-.25t + \cos(2\pi t) - 1).$$

The unstable solution oscillates as time passes, but the oscillations dampen. In Figure 6a, we plot the solution together with the dG(0) approximation computed with error below .12. In Figure 6b, we plot the time steps used for the computation. We see that the steps are adjusted for each oscillation and in addition that there is an overall trend to increasing the steps as the size of the solution decreases.

In addition, the solution has changing stability characteristics. In Figure 7a, we plot the stability factor versus time, and it is evident that the numerical error decreases and increases in alternating periods of time. If a crude 'exponential' bound on the stability factor is used instead of a computational estimate, then the error is greatly overestimated with the consequence that the computation can only be done over a much smaller interval. To demonstrate the effectiveness of the a posteriori estimate for error control, we plot the ratio of the true error to the computed bound versus time in Figure 7b. The ratio quickly settles down to a constant, which means that the bound is predicting the behaviour of the error in spite of the fact that the error oscillates a good deal.

8.4. Quadrature errors

We now consider the error arising from computing the integrals in the dG(0) method (8.3) approximately using quadrature. We focus on the error from computing $\int_{I_n} f \, \mathrm{d}t$ using quadrature. To illustrate essential aspects, we consider the midpoint rule,

$$\int_{I_n} f \, \mathrm{d}t \approx k_n f(t_{n-\frac{1}{2}}), \qquad t_{n-\frac{1}{2}} = \frac{1}{2}(t_{n-1} + t_n), \tag{8.21}$$

and also the rectangle rule,

$$\int_{I_n} f \, \mathrm{d}t \approx k_n f(t_n). \tag{8.22}$$

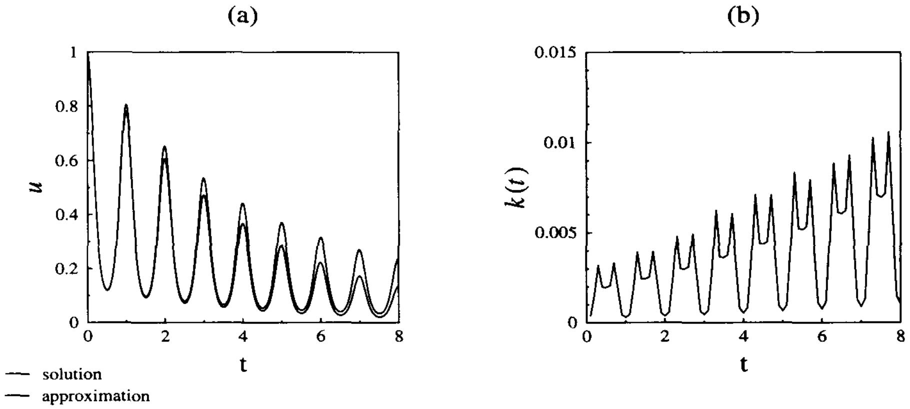

Fig. 6. Solution, approximation and step sizes for Example 6

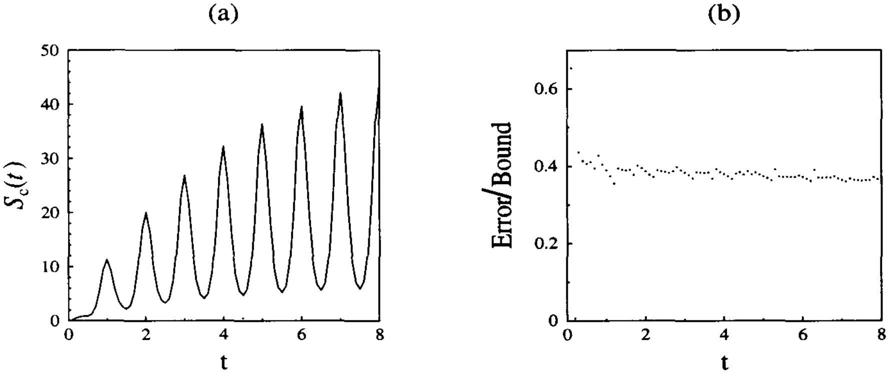

Fig. 7. Stability factor and error/bound ratio for Example 6

We recall that the backward Euler scheme is generated by using the rectangle rule. We compare dG(0) approximations computed with the two quadratures (8.21) and (8.22) and conclude that the classical choice (8.22) is less accurate for many problems. The analysis shows the advantage of separating the Galerkin and quadrature errors since they accumulate differently.

For the midpoint rule (8.21), the quadrature error on a single interval is bounded by

$$\left| \int_{I_n} f \, \mathrm{d}t - k_n f(t_{n-\frac{1}{2}}) \right| \le \min\left\{ \int_{I_n} |kf'| \, \mathrm{d}t, \frac{1}{2} \int_{I_n} |k^2 f''| \, \mathrm{d}t \right\}. \qquad (8.23)$$

The corresponding error estimate for the rectangle rule reads

$$\left| \int_{I_n} f \, \mathrm{d}t - k_n f(t_n) \right| \leq \int_{I_n} |kf'| \, \mathrm{d}t. \tag{8.24}$$

We notice that the midpoint rule is more accurate unless $|f''| >> |f'|$, while the cost of the two rules is the same.

We now determine the effect of quadrature on the final error $u(t_N) - U_N$ after N steps. We start with the modified form of the the error representation

$$\begin{aligned} e_N^2 &= \sum_{n=1}^{N} \left(\int_{I_n} (\hat{f} - aU)(\varphi - \pi_k\varphi) \, \mathrm{d}t - [U_{n-1}](\varphi - \pi_k\varphi)_{n-1} \right. \\ &\quad \left. + \int_{I_n} (f - \hat{f})\varphi \, \mathrm{d}t \right). \end{aligned} \tag{8.25}$$

where for $t \in I_n$ we define $\hat{f}(t) \equiv f(t_{n-\frac{1}{2}})$ for the midpoint rule and $\hat{f}(t) \equiv f(t_n)$ for the rectangle rule. Introducing the weak-stability factor

$$\tilde{S}_c(t_N) \equiv \frac{\int_0^{t_N} |\varphi| \mathrm{d}t}{|\varphi(t_N)|},$$

we obtain a modified a posteriori error estimate that includes the quadrature errors.

Theorem 8 U satisfies for $N = 1, 2, \ldots,$

$$|u(t_N) - U_N| \leq S_c(t_N) |k\hat{R}(U)|_{(0,t_N)} + \tilde{S}_c(t_N) C_{qj} \int_0^{t_N} k^j |f^{(j)}| \, \mathrm{d}t,$$

where

$$\hat{R}(U) = \frac{|U_n - U_{n-1}|}{k_n} + |\hat{f} - aU|_{I_n}, \quad t \in I_n,$$

and $j = 1$ for the rectangle rule, $j = 2$ for the midpoint rule, $C_{q1} = 1$, $C_{q2} = 1/2$, $f^{(1)} = f'$ and $f^{(2)} = f''$.

We note that this estimate includes the factor $\int_0^{t_N} k^j |f^{(j)}| \, \mathrm{d}t$ that grows linearly with t_N if the integrand is bounded. This linear growth in time, representing the accumulation of quadrature errors, is also present in the case $a \geq 0$ when $\tilde{S}(t_N) \leq 1$. For long-time integration in the case $a \geq 0$, it is thus natural to use the midpoint rule, since the accumulation of quadrature error can be compensated by the second-order accuracy.

In general, since the computational cost of the quadrature is usually small compared to the Galerkin computational work (which requires the solution of a system of equations), the precision of the quadrature may be increased if needed without significantly increasing the overall work. This illustrates the

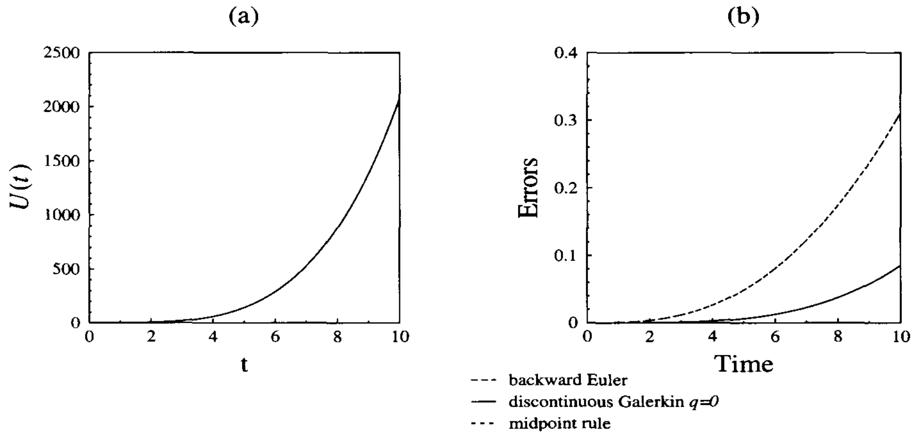

Fig. 8. Approximation and errors for Example 7

importance of separating Galerkin-discretization and quadrature errors since they accumulate at different rates. These errors should not be combined as happens in the classic analysis of difference schemes, leading to non-optimal performance.

Example 7. We consider the approximation of $u' - .1u = t^3$, $u(0) = 1$. We compute using the dG(0) method, the backward Euler scheme (rectangle rule quadrature), and the midpoint rule, with accuracies plotted below. The approximation is plotted in Figure 8a; the problem is not dissipative, so we expect error accumulation. We plot the errors of the three computations in Figure 8b. The dG(0) and the dG(0) with midpoint rule approximations are very close in accuracy, while the backward-Euler computation errors accumulate at a much faster rate.

8.5. A 'hyperbolic' model problem

We consider the ordinary differential equation model for a 'hyperbolic' problem: find $u = (u_1, u_2)$ such that

$$\begin{aligned} &u_1' + au_2 = f_1, \quad t > 0, \\ &u_2' - au_1 = f_2, \quad t > 0, \\ &u_1(0) = u_{10}, \quad u_2(0) = u_{20}, \end{aligned} \tag{8.26}$$

where the $a = a(t)$ is a given bounded coefficient with $|a| \leq \mathbb{A}$, and the f_i and u_{i0} are given data. This is a simple model for wave propagation.

We study the application of the cG(1) method to (8.26), where V_k is the set of continuous piecewise-linear functions $v = (v_1, v_2)$ on a partition T_k. This method takes the form: find $U = (U_1, U_2)$ in V_k such that for $n = 1, 2, \ldots,$

$$\begin{array}{rcl} \int_{I_n} (U_1' + aU_2)\, dt & = & \int_{I_n} f_1\, dt, \\ \int_{I_n} (U_2' - aU_1)\, dt & = & \int_{I_n} f_2\, dt, \\ U_1(0) = u_{10}, \quad U_2(0) & = & u_{20}, \end{array} \tag{8.27}$$

corresponding to using piecewise-constant test functions on each interval I_n. We use piecewise-constant test functions because there are only first-order derivatives in (8.26), in contrast to the elliptic problem discussed above. In the case where a is constant, with the notation $U_{i,n} = U_i(t_n)$, the method (8.27) reduces to:

$$\begin{array}{rcl} U_{1,n} - U_{1,n-1} + k_n a(U_{2,n} + U_{2,n-1})/2 & = & \int_{I_n} f_1\, dt, \\ U_{2,n} - U_{2,n-1} - k_n a(U_{1,n} + U_{1,n-1})/2 & = & \int_{I_n} f_2\, dt, \\ U_1(0) = u_{10}, \quad U_2(0) & = & u_{20}, \end{array} \tag{8.28}$$

from which the classical Crank–Nicolson method can be obtained by an appropriate choice of quadrature. The method cG(1) has less dissipation and better accuracy than dG(0), and it is advantageous to use it in this problem since the solution is smooth.

8.6. An a posteriori error estimate

To derive an a posteriori error estimate for the error $e_N = u(t_N) - U_N$, $U_N \equiv U(t_N)$, we introduce the dual problem: find $\varphi = (\varphi_1, \varphi_2)$ such that

$$\begin{array}{rcl} -\varphi_1' + a\varphi_2 & = & 0, + \in (0_1 + N), \\ -\varphi_2' - a\varphi_1 & = & 0, + \in (0_1 + N), \\ \varphi(t_N) & = & e_N. \end{array} \tag{8.29}$$

Again using Galerkin orthogonality, we obtain an error representation formula:

$$\|e_N\|^2 = -\int_0^{t_N} R \cdot (\varphi - \pi_k \varphi)\, dt,$$

where

$$R_1 = U_1' + aU_2 - f_1, \quad R_2 = U_2' - aU_1 - f_2$$

and π_k is the nodal interpolation operator into V_k. Multiplying the equations by φ_1 and φ_2 respectively and using the cancellation of the terms $\pm a\varphi_1\varphi_2$, we obtain the following stability estimates:

$$\max_{0 \le t \le t_N} \|\varphi(t)\| \le \|e_N\|$$

and

$$\max_{0 \le t \le t_N} \|\varphi'(t)\| \le \mathbb{A}\|e_N\|.$$

Combining the error representation with the strong-stability estimate and using the interpolation estimate (6.3) we have proved:

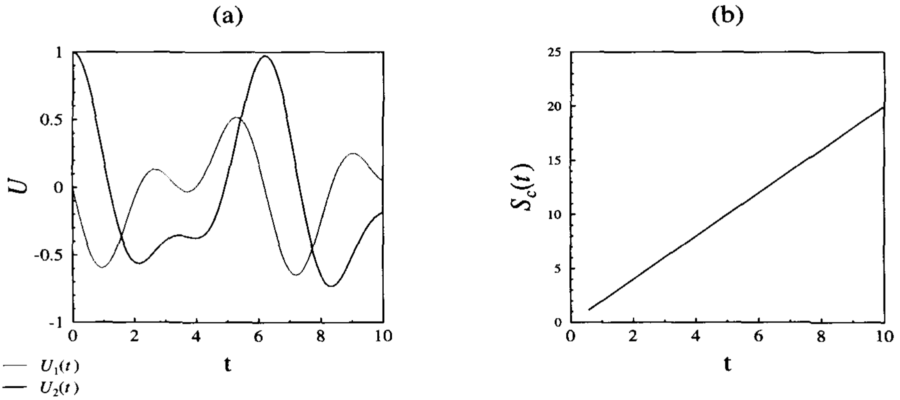

Fig. 9. Approximation and stability factors for Example 8

Theorem 9 U satisfies for $N = 1, 2, \ldots,$

$$\|u(t_N) - U_N\| \leq \mathbb{A} \int_0^{t_N} k\|R\| \, dt.$$

8.7. An a priori error estimate

The corresponding a priori error estimate takes the form:

Theorem 10

$$\|u(t_N) - U_N\| \leq \mathbb{A} \int_0^{t_N} \|k^2 u''\| \, dt \leq \mathbb{A} t_N \|k^2 u''\|_{(0,t_N)}.$$

We note the linear growth of error with time that is characteristic of a hyperbolic problem.

Example 8. We compute for the problem:

$$\begin{array}{l} u_1' + 2u_2 = \cos(\pi t/3), \quad t > 0, \\ u_2' - 2u_1 = 0, \quad t > 0, \\ u_1(0) = 0, \quad u_2(0) = 1, \end{array}$$

using the cG(1) method with error below .07. The two components of the approximation are plotted in Figure 9a. This demonstrates that different components of a system of equations may behave differently at different times. The error control discussed here must choose the time steps to maintain accuracy in all components simultaneously. In Figure 9b, we plot the stability factor, and the linear growth of error is evident.

9. Nonlinear systems of ordinary differential equations

The framework for a posteriori error analysis described above directly extends to initial-value problems for nonlinear systems of differential equations in $\mathbb{R}^d, d \geq 1$ (or more generally a Hilbert space). The ease of the extension depends on the definition of the stability factors occurring in the a posteriori analysis. In the adaptive algorithms built on the a posteriori error estimates, the stability factors are estimated by computation and not by analysis. Thus the essential computational difficulty is the approximation of the stability factors and the essential mathematical difficulty is the justification of this process. We return to this issue after presenting the extension.

We consider the computation of solutions $u = u(t)$ of the following initial-value problem:

$$\begin{aligned} u' + f(t,u) &= 0, \quad t > 0, \\ u(0) &= u_0, \end{aligned} \tag{9.1}$$

where $f(t,\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is a given function and u_0 given initial data. We assume that f and u_0 are perturbations of correct $\hat{f}$ and $\hat{u}_0$, and denote by $\hat{u}$ the corresponding exact solution satisfying

$$\begin{aligned} \hat{u}' + \hat{f}(t,\hat{u}) &= 0, \quad t > 0, \\ \hat{u}(0) &= \hat{u}_0. \end{aligned} \tag{9.2}$$

We seek an a posteriori error bound for the complete error $\hat{e} \equiv \hat{u} - U$, where U is the dG(0) approximate solution of (9.1) defined by: find U in W_k such that for all constant vectors v,

$$\int_{I_n} (U' + f(t,U)) \cdot v \mathrm{d}t + [U_{n-1}] \cdot v_{n-1}^+ = 0, \tag{9.3}$$

where $[v_n] \equiv v_n^+ - v_n^-$, $v_n^\pm \equiv \lim_{s\to\pm 0} v(t_n + s)$ and $U_0^- = \hat{u}_0$. With the notation $U_n \equiv U|_{I_n}$, the dG(0) method (9.3) takes the form

$$U_n - U_{n-1} + \int_{I_n} f(t,U_n)\,\mathrm{d}t = 0, \quad n = 1,2,..., \tag{9.4}$$

where $U_0 = \hat{u}_0$. Again, this is an improved variation of the classical backward Euler method with exact evaluation of the integral with integrand $f(t,U)$.

9.1. An a posteriori error estimate

To derive an a posteriori error estimate for e_N for $N \geq 1$ including data, modelling and Galerkin-discretization errors, we introduce the continuous dual 'backward' problem

$$\begin{aligned} -\varphi' + \hat{A}(t)^*\varphi &= 0, \quad t \in (0,t_N), \\ \varphi(t_N) &= \hat{e}_N, \end{aligned} \tag{9.5}$$

where

$$\hat{A}(t) \equiv \int_0^1 \hat{f}_u(t, su + (1-s)U)\,\mathrm{d}s,$$

where $\hat{f}_u(t,\cdot)$ denotes the Jacobian of $\hat{f}(t,\cdot)$ and $*$ denotes the transpose. Note that

$$\hat{A}(t)e = \int_0^1 \hat{f}_u(t, s\hat{u} + (1-s)U)\hat{e}\,\mathrm{d}s$$

and

$$\int_0^1 \frac{\mathrm{d}}{\mathrm{d}s}\hat{f}(t, s\hat{u} + (1-s)U)\,\mathrm{d}s = \hat{f}(t,\hat{u}) - \hat{f}(t,U).$$

Integrating by parts, with $\|\cdot\|$ denoting the Euclidean norm, we get

$$\begin{aligned}
\|e_N\|^2 &= \|e_N\|^2 + \sum_{n=1}^{N}\int_{I_n} e\cdot(-\varphi' + \hat{A}^*\varphi)\,\mathrm{d}t \\
&= \sum_{n=1}^{N}\int_{I_n}(e' + \hat{A}(t)e)\cdot\varphi\,\mathrm{d}t + \sum_{n=0}^{N-1}[e_n]\cdot\varphi_n^+ + e_0^-\cdot\varphi(0) \\
&= -\sum_{n=1}^{N}\left(\int_{I_n}(U' + f(t,U))\cdot\varphi\,dt + [U_{n-1}]\cdot\varphi_{n-1}^+\right) \\
&\quad + e_0^-\cdot\varphi(0) + \sum_{n=1}^{N}\int_{I_n}(f(t,U) - \hat{f}(t,U))\cdot\varphi\mathrm{d}t.
\end{aligned}$$

Now we use Galerkin orthogonality (9.3) to insert $\pi_k\varphi$ in the first term on the right, and we obtain, since $U' = 0\,\mathrm{on}\,I_n$, the following error representation formula:

$$\begin{aligned}
\|e_N\|^2 &= -\sum_{n=1}^{N}\left(\int_{I_n}(U' + f(t,U))\cdot(\varphi - \pi_k\varphi)\,\mathrm{d}t + [U_{n-1}]\cdot(\varphi - \pi_k\varphi)_{n-1}^+\right) \\
&\quad + e_0^-\cdot\varphi(0) + \sum_{n=1}^{N}\int_{I_n}(f(t,U) - \hat{f}(t,U))\cdot\varphi\mathrm{d}t \\
&= -\int_0^{t_N} f(t,U)\cdot(\varphi - \pi_k\varphi)\,\mathrm{d}t - \sum_{n=0}^{N-1}[U_n]\cdot(\varphi - \pi_k\varphi)_n^+ + e_0^-\cdot\varphi(0) \\
&\quad + \int_0^{t_N}(f(t,U) - \hat{f}(t,U))\cdot\varphi\mathrm{d}t.
\end{aligned}$$

Recalling the interpolation estimate (6.3), we see that

$$\|e_N\|^2 \le \int_0^{t_N}\|\varphi'\|\mathrm{d}t \max_{1\le n\le N}(\|[U_{n-1}]\| + k_n\|f(\cdot,U_n)\|_{I_n})$$

$$+\|e_0^-\|\|\varphi(0)\| + \int_0^{t_N}\|\varphi\|\mathrm{d}t \max_{1\le n\le N}\|f(\cdot,U_n) - \hat{f}(\cdot,U_n)\|_{I_n}.$$

Finally, we define the strong-stability factor $S_{\mathrm{c}}(t_N)$ by

$$S_{\mathrm{c}}(t_N) \equiv \frac{\int_0^{t_N} \|\varphi'\| \mathrm{d}t}{\|\varphi(t_N)\|}, \tag{9.6}$$

and the data and modelling stability factors by

$$S_{\mathrm{d}}(t_N) \equiv \frac{\|\varphi(0)\|}{\|\varphi(t_N)\|}, \qquad S_{\mathrm{m}}(t_N) \equiv \frac{\int_0^{t_N} \|\varphi(s)\| \mathrm{d}s}{\|\hat{\varphi}(t_N)\|},$$

and arrive at an a posteriori error estimate for data, modelling and Galerkin-discretization errors.

Theorem 11 U satisfies for $N = 1, 2, \dots,$

$$\begin{aligned} \|u(t_N) - U_N\| \le\ & S_{\mathrm{c}}(t_N) \max_{1 \le n \le N} k_n R(n, U) \\ & + S_{\mathrm{d}}(t_N) \|\hat{u}_0 - u_0\| \\ & + S_{\mathrm{m}}(t_N) \max_{1 \le n \le N} \|\hat{f}(\cdot, U_n) - f(\cdot, U_n)\|_{I_n}, \end{aligned}$$

where

$$R(n, U) \equiv \|U_n - U_{n-1}\|/k_n + \|f(\cdot, U_n)\|_{I_n}.$$

Remark 6 There is a corresponding a priori result with stability factors related to discrete dual problems.

9.2. Computational evaluation of stability factors

To give the a posteriori estimate concrete meaning, the stability factors have to be determined. Accurate analytic estimates are possible only in a few special cases, and in general we resort to numerical integration of the dual linearized problems. The critical mathematical issue is the reliability of this evaluation, since this directly translates into the reliability of the adaptive algorithm. The basic sources of error in the computational evaluation of stability factors are (i) the choice of linearization and (ii) the choice of data, and (iii) the numerical solution of the dual linearized problem. In practice, the problem is linearized around an approximation rather than the mean value that involves the unknown exact solution used in the definition of the dual problems. Moreover, the current error is unknown, and hence the true initial data for the dual problems cannot be used. Finally, the resulting problem must be approximated numerically. The reliability of the computation of stability factors related to (i) and (ii) may be guaranteed for certain classes of problems but in the general case, little is known. Reliability with respect to (iii) seems to be an issue of smaller magnitude.

In many experiments, (see Estep (to appear), Estep and French, (to appear), Eslep and Johnson (1994)), we have seen that the choice of initial data in the dual problem is often immaterial provided the time interval is

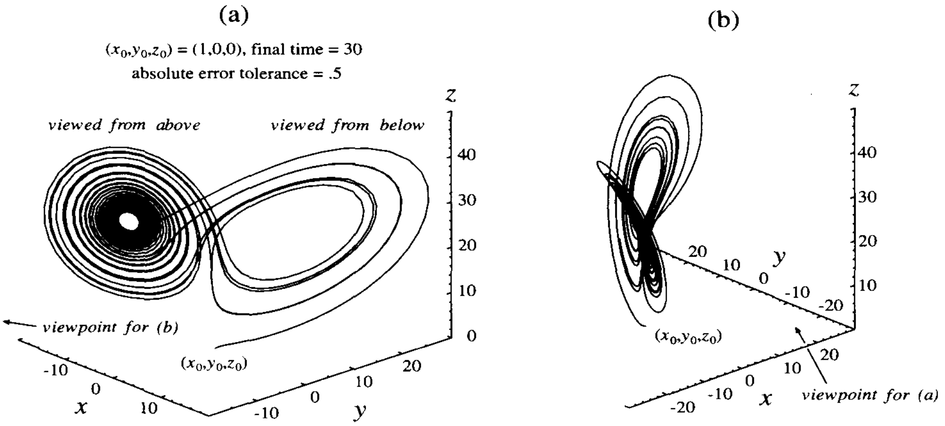

Fig. 10. Two views of a solution of the Lorenz system

sufficiently long. Otherwise, computing dual problems using several different initial values improves reliability. Moreover, unless grossly inaccurate, approximate trajectories seem to provide reasonably accurate stability factors.

Example 9. In the early 1960s, the meteorologist E. Lorenz presented a simple model in order to explain why weather forecasts over more than a couple of days are unreliable. The model is derived by taking a three-element Fem space discretization of the Navier–Stokes equations for fluid flow (the 'fluid' being the atmosphere in this case) and simply ignoring the discretization error. This gives a three-dimensional system of ODE's in time:

$$\begin{aligned} &x' = -\sigma x + \sigma y, && t > 0,\\ &y' = -rx - y - xz, && t > 0,\\ &z' = -bz + xy, && t > 0,\\ &x(0) = x_0, y(0) = y_0, z(0) = z_0, && \end{aligned} \tag{9.7}$$

where σ, r and b are positive constants. These were determined originally as part of the physical problem, but the interest among mathematicians quickly shifted to studying (9.7) for values of the parameters that make the problem *chaotic*.

A precise definition of chaotic behaviour seems difficult to give, but we point out two distinguishing features: while confined to a fixed region in space, the solutions do not 'settle down' into a steady state or periodic state; and the solutions are *data sensitive*, which means that perturbations of the initial data of a given solution eventually cause large changes in the solution. This corresponds to the 'butterfly effect' in meteorology in which small causes may sometimes have large effects on the evolution of the weather. In such situations, numerical approximations always become inaccurate after

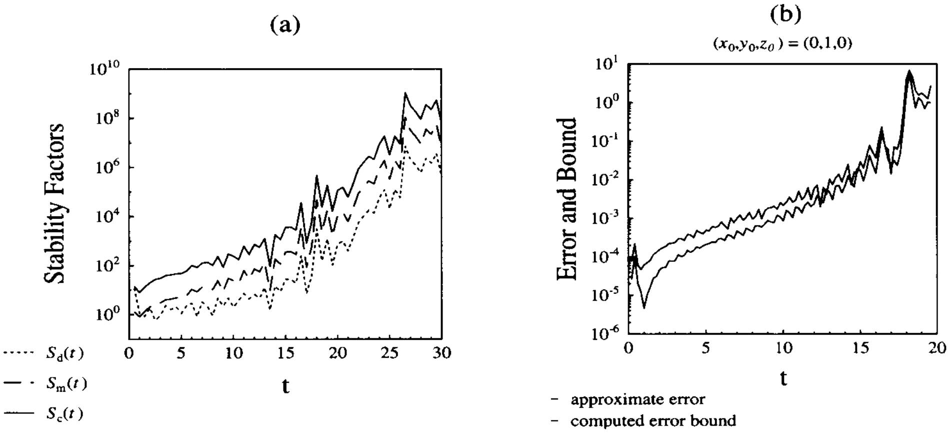

Fig. 11. Stability factors and error bound for the Lorentz system

some time. An important issue is to determine this time, since, for example, it is related to the maximal length of a weather forecast.

We choose standard values $\sigma = 10$, $b = 8/3$ and $r = 28$, and we compute with the dG(1) method. In Figure 10, we plot two views of the solution corresponding to initial data $(1, 0, 0)$ computed with an error of .5 up to time 30. The solutions always behave similarly: after some short initial time, they begin to 'orbit' around one of two points, with an occasional 'flip' back and forth between the points. The chaotic nature of the solutions is this flipping that occurs at apparently random times. In fact, accurate computation can reveal much detail about the behaviour of the solutions; see Eriksson *et al.* (1994*b*).

Here, we settle for demonstrating the quality of the error control explained in these notes. In Figure 11a, we plot the approximate stability factors on a logarithmic scale. The data sensitivity of this problem is reflected in the overall exponential growth of the factors, and it is clear that any computation becomes inaccurate at some point. The error control allows this time to be determined. Note, however, that the factors do not grow uniformly rapidly and there are periods of time with different data sensitivity. It is important for the error control to detect these to avoid gross overestimation of the error. To test this, we do an experiment. We compute using two error tolerances, one 10^{-5} smaller than the other, and then we subtract the less accurate computation from the more accurate computation. This should be a good approximation to the true error (which is unknown of course). In Figure 11b, we plot this approximate error together with the error bound predicted by the error control based on a posteriori estimates as we have described. There is remarkable agreement.

Finally, in Figure 12a, we plot the S_c for the various trajectories computed

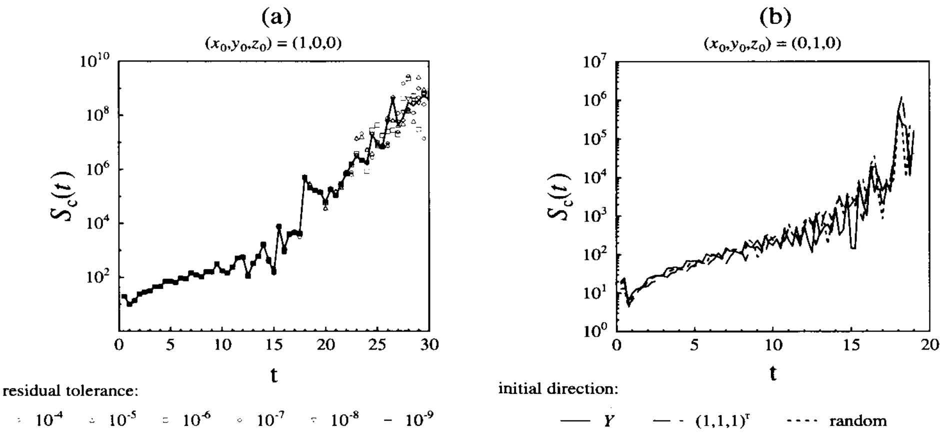

Fig. 12. Stability factors for the Lorentz system

with different tolerances. Overall, the stability factors are roughly the same order of magnitude for all trajectories. The stability factors agree as long as the trajectories are near each other, but variations occur as some trajectories enter more data-sensitive areas than others at the same time. In Figure 12b, we plot the approximation to S_c computed for three different choices of initial data for the dual problem (9.5).

10. An elliptic model problem in two dimensions

In this section, we consider Fem for Poisson's equation in two dimensions. We discuss a priori and a posteriori error estimates for the Galerkin-discretization error and also the discrete-solution error for a multigrid method, and design corresponding adaptive methods. The analysis is largely parallel to that of the one-dimensional model problem, though the analysis of the multigrid method is more technical.

Consider the Poisson equation with homogeneous Dirichlet boundary conditions: find $u = u(x)$ such that

$$\begin{aligned} -\Delta u &= f, \quad x \in \Omega, \\ u &= 0, \quad x \in \partial\Omega, \end{aligned} \tag{10.1}$$

where Ω is a bounded domain in $\mathbb{R}^2$ with boundary $\partial\Omega$, $x \equiv (x_1, x_2)$, Δ is the Laplace operator and $f = f(x)$ is given data. The variational form of (10.1) reads: find $u \in H_0^1(\Omega)$ such that

$$(\nabla u, \nabla v) = (f, v), \quad \forall v \in H_0^1(\Omega), \tag{10.2}$$

where $(w, v) \equiv \int_\Omega wv\,dx$, $(\nabla w, \nabla v) \equiv \int_\Omega \nabla w \cdot \nabla v\,dx$ and $H_0^1(\Omega)$ is the Sobolev space of square-integrable functions with square integrable derivatives on Ω that vanish on $\partial\Omega$. We recall that $\|\nabla \cdot\|_2$ is a norm in $H_0^1(\Omega)$

that is equivalent to the $H^1(\Omega)$ norm. The existence of a unique solution of (10.2) follows by the Riesz representation theorem if $f \in H^{-1}(\Omega)$, where $H^{-1}(\Omega)$ is the dual space of $H_0^1(\Omega)$ with norm

$$\|f\|_{H^{-1}(\Omega)} \equiv \sup_{v \in H_0^1(\Omega), \|\nabla v\|_2 = 1} (f, v).$$

We recall strong-stability (or elliptic regularity) estimates for the solution of (10.2) to be used below. The estimates are termed 'strong' because derivatives of the solution u are estimated. We use the notation $D^0 v \equiv v$, $D^1 v \equiv |\nabla v|$ and $D^2 v \equiv (\sum_{i=1}^2 (\frac{\partial^2 v}{\partial x_i \partial x_j})^2)^{\frac{1}{2}}$. Further, we use $\|\cdot\| = \|\cdot\|_\Omega$ to denote the $L^2(\Omega)$ norm.

Lemma 3 The solution u of (10.2) satisfies

$$\|\nabla u\| \leq \|f\|_{H^{-1}(\Omega)}. \tag{10.3}$$

Furthermore, if Ω is convex with polygonal boundary, or if $\partial\Omega$ is smooth, then there is a constant S_c independent of f, such that

$$\|D^2 u\| \leq S_c \|f\|. \tag{10.4}$$

If Ω is convex, then $S_c = 1$.

10.1. Fem for Poisson's equation

The simplest Fem for (10.1) results from applying Galerkin's method to the variational formulation (10.2) using a finite-dimensional subspace V_h $H_0^1(\Omega)$ based on piecewise-linear approximation on triangles. For simplicity, we consider the case of a convex polygonal domain. Let $T_h = \{K\}$ be a finite-element triangulation of Ω into triangles K of diameter h_K with associated set of nodes $N_h = \{N\}$ such that each node N is the corner of at least one triangle. We require that the intersection of any two triangles K' and K'' in T_h be either empty, a common triangle side or a common node.

To the mesh T_h we associate a mesh function $h(x)$ satisfying, for some positive constant c_1,

$$c_1 h_K \leq h(x) \leq h_K, \quad \forall x \in K, \quad \forall K \in T_h. \tag{10.5}$$

We further assume that there is a positive constant c_2 such that

$$c_2 h_K^2 \leq 2|K|, \quad \forall K \in T_h. \tag{10.6}$$

This is a 'minimum angle' condition stating that angles of triangles in T_h are bounded from below by the constant c_2. As usual, C_i denotes an interpolation constant related to piecewise-linear interpolation on the mesh T_h. In this case, C_i depends on c_1 and c_2, but not on h otherwise.

With $V_h \subset H_0^1(\Omega)$ denoting the standard finite-element space of piecewise-linear functions on T_h, the Fem for (10.1) reads: find $u_h \in V_h$ such that

$$(\nabla u_h, \nabla v) = (f, v), \quad \forall v \in V_h. \tag{10.7}$$

Galerkin orthogonality for (10.7), resulting from (10.2) and (10.7), takes the form:

$$(\nabla(u - u_h), \nabla v) = 0, \quad \forall v \in V_h. \tag{10.8}$$

We write $u_h = \sum_{i=1}^M \xi_i \varphi_i$, where $\{\varphi_i\}$ is the usual basis for V_h associated to the set of nodes $N_h^0 = \{N_i\}_{i=1}^M$ in the interior of Ω and $\xi_i = u_h(N_i)$. Then, (10.7) is equivalent to the linear system of equations

$$A\xi = b, \tag{10.9}$$

where $\xi = (\xi_i)_{i=1}^M$, $A = (a_{ij})_{j,i=1}^M$ is the $M \times M$ stiffness matrix with elements $a_{ij} \equiv (\nabla\varphi_i, \nabla\varphi_j)$ and $b = (b_j)_{j=1}^M$ is the load vector with $b_j \equiv (f, \varphi_j)$. We use a multigrid method to solve the discrete system (10.9), producing an approximation $\tilde{u}_h \in V_h$ of the exact discrete solution u_h.

We require an error estimate for interpolation by piecewise-linear functions, where the piecewise-linear nodal interpolant $\pi_h w \in V_h$ of a given function $w \in H_0^1(\Omega) \cap H^2(\Omega)$ is defined by $\pi_h w(N) = w(N)$, $\forall N \in N_h^0$. We also need an analogous estimate for a 'quasi-interpolant' of $w \in H_0^1(\Omega)$ that requires less regularity where the 'quasi-interpolant' interpolates local mean values of w over neighbouring elements. We use the same notation for the nodal interpolant and the quasi-interpolant. The basic interpolation estimate is:

Lemma 4 For $s \leq m + 1$, $m \in \{0, 1\}$, there are constants $C_{\rm i}$ depending only on c_1 and c_2 such that for $w \in H_0^1(\Omega) \cap H^{m+1}(\Omega)$

$$\|h^{-m-1+s} D^s(w - \pi_h w)\| + \left(\sum_{K \in T_h} h_K^{-2m-1} \|w - \pi_h w\|_{\partial K}^2 \right)^{1/2} \leq C_{\rm i}\, \|D^{m+1} w\|.$$

10.2. The discrete and continuous residuals

We shall prove an a posteriori error estimate for the total error $e \equiv u - \tilde{u}_h = u - u_h + u_h - \tilde{u}_h$ including the Galerkin-discretization error $u - u_h$ and the discrete-solution error $u_h - \tilde{u}_h$. The a posteriori error estimate involves both a discrete residual $R_h(\tilde{u}_h)$ related to solving the discrete system (10.7) approximately and an estimate $R(\tilde{u}_h)$ of the residual related to the continuous problem (10.1).

To define $R_h(\tilde{u}_h)$, we introduce the L^2-projection $P_h \colon L^2(\Omega) \to V_h$ defined by $(u - P_h u, v) = 0, \quad \forall v \in V_h$, and the 'discrete Laplacian' $\Delta_h \colon V_h \to V_h$ on V_h defined by $(\Delta_h w, v) = -(\nabla w, \nabla v), \quad \forall v, w \in V_h$. We may then write

(10.7) equivalently as $R_h(u_h) = 0$, where for $w \in V_h$ the discrete residual $R_h(w)$ is defined as

$$R_h(w) \equiv \Delta_h w + P_h f. \tag{10.10}$$

For the approximate solution $\tilde{u}_h$, we have $R_h(\tilde{u}_h) \neq 0$.

Remark 7 Letting $\tilde{U}_h$ denote the nodal-valued vector of the approximate solution $\tilde{u}_h$, we define the 'algebraic' residual $r_h(\tilde{U}_h)$ by $r_h(\tilde{U}_h) \equiv b - A\tilde{U}_h$. By the definition of $R_h(\tilde{u}_h)$, it follows that $r_h(\tilde{U}_h) = M_h \hat{R}_h(\tilde{u}_h)$, where $M_h = (m_{ij})_{i,j=1}^M$ is the mass matrix with elements $m_{ij} = (\varphi_i, \varphi_j)$ and $\hat{R}_h(\tilde{u}_h)$ is the nodal-valued vector of $R(\tilde{u}_h)$. Thus, $R_h(\tilde{u}_h)$ is computable from the algebraic residual $r_h(\tilde{U}_h)$ by applying M_h^{-1}. In practice, M_h may be replaced by a diagonal matrix corresponding to 'mass lumping'.

The estimate $R(\tilde{u}_h)$ for the continuous residual is defined on each element $K \in T_h$ by

$$R(\tilde{u}_h) \equiv |f + \Delta \tilde{u}_h| + D_h^2 \tilde{u}_h, \quad x \in K, \tag{10.11}$$

where for $v \in V_h$

$$D_h^2 v|_K \equiv \frac{1}{2\sqrt{h_K}} \|h_K^{-1}[\nabla v]\|_{\partial K}, \tag{10.12}$$

where $[\nabla v]$ denotes the jump in ∇v across ∂K. Note that D_h^2 resembles a second derivative in the case of piecewise-linear approximation when ∇v is constant on each element K. The factor $1/2$ arises naturally because the jump is associated to two neighbouring elements. We note that $D_h^2 v$ is a piecewise-constant function and thus in particular belongs to $L^2(\Omega)$.

The residual function $R(\tilde{u}_h)$ also belongs to $L^2(\Omega)$ and has two parts: the 'interior' part $|f + \Delta \tilde{u}_h|$ and the 'boundary' part $D_h^2 \tilde{u}_h$. The boundary part can be made to vanish in the one-dimensional problems considered above, because the interpolation error vanishes at inter-element boundaries. In the present case with piecewise-linear approximation, $R(u_h) = |f| + D_h^2 u_h$, $x \in K$, while in the case of higher-order polynomials, Δu_h no longer vanishes on each triangle and has to be taken into account.

In the proofs below, we use the following crucial estimate:

Lemma 5 For $m \in \{0, 1\}$, there is a constant $C_{\rm i}$ such that $\forall v \in H_0^1(\Omega) \cap H^{m+1}(\Omega)$,

$$|(f, v - \pi_h v) - (\nabla \tilde{u}_h, \nabla(v - \pi_h v))| \leq C_{\rm i} \|h^{m+1} R(\tilde{u}_h)\| \|D^{m+1} v\|. \tag{10.13}$$

Appropriate values of the constant $C_{\rm i}$ in (10.13) can be calculated analytically or numerically. If $c_1 \sim 1$ and $c_2 \sim 1$, then $C_{\rm i} \sim 0.2$ for $m = 0, 1$ (cf. Johnson and Hansbo (1992*b*), Becker *et al.* (1994)).

Proof. By integration by parts, observing that $v - \pi_h v$ is continuous, we have

$$\begin{aligned}(f, v - \pi_h v) &- (\nabla \tilde{u}_h, \nabla(v - \pi_h v)) \\ &= \sum_{K \in T_h} \{(f + \Delta \tilde{u}_h, v - \pi_h v)_K - (\partial_n \tilde{u}_h, v - \pi_h v)_{\partial K}\} \\ &= \sum_{K \in T_h} \left\{(f + \Delta \tilde{u}_h, v - \pi_h v)_K - \frac{1}{2}([\nabla \tilde{u}_h], v - \pi_h v)_{\partial K}\right\},\end{aligned}$$

with $[\nabla \tilde{u}_h]$ denoting the jump of $\nabla \tilde{u}_h$ across the element edges, from which the result follows by Lemma 4. □

10.3. A priori estimates of the Galerkin-discretization error

We first give an a priori error estimate of the Galerkin-discretization error $u - u_h$ in the energy norm.

Theorem 12 There exists a constant $C_{\rm i}$ depending only on c_1 and c_2 such that

$$\|\nabla(u - U)\| \leq \|\nabla(u - \pi_h u)\| \leq C_{\rm i} \|h D^2 u\|. \tag{10.14}$$

Proof. In (10.8), we choose $v = U - \pi_h u$ and use Cauchy's inequality to get

$$\begin{aligned}\|\nabla e\|^2 &= (\nabla e, \nabla(u - U)) = (\nabla e, \nabla(u - U)) + (\nabla e, \nabla(U - \pi_h u)) \\ &= (\nabla e, \nabla(u - \pi_h u)) \leq \|\nabla e\| \, \|\nabla(u - \pi_h u)\|,\end{aligned} \tag{10.15}$$

from which the desired result follows from Lemma 4. □

We next give an a priori error estimate in the L^2 norm.

Theorem 13 There exists a constant $C_{\rm i}$ only depending on c_1 and c_2 such that

$$\|u - U\| \leq S_{\rm c} C_{\rm i} \|h \nabla(u - U)\|, \tag{10.16}$$

where

$$S_{\rm c} \equiv \max_{g \in L^2(\Omega)} \frac{\|D^2 \varphi\|}{\|g\|},$$

with $\varphi \in H_0^1(\Omega)$ satisfying $-\Delta \varphi = g$ in Ω. Further, if $|\nabla h(x)| \leq \mu$, $x \in \Omega$, with μ a sufficiently small positive constant, then

$$\|h \nabla(u - U)\| \leq C_{\rm i} \|h^2 D^2 u\|, \tag{10.17}$$

where $C_{\rm i}$ now depends also on μ.

Proof. The proof of (10.16) uses duality in a manner similar to that of the proof of Theorem 4. Note that (10.17) follows directly from the energy-norm error estimate if h is constant. □

10.4. A posteriori error estimates of Galerkin and discrete-solution errors

We now turn to a posteriori error estimates, including the discrete-solution error in the case of multigrid methods. Let T_j, $j = 0, 1, 2, \dots, k$, be a hierarchy of successively refined meshes with corresponding nested sequence of finite-element spaces V_j and mesh functions h_j, where the final mesh T_k corresponds to the mesh T_h in the above presentation. We seek to compute an approximation $\tilde{u}_k \in V_k$ of the finite-element solution $u_k \in V_k$ on the final mesh T_k, using a multigrid algorithm based on the hierarchy of meshes. For $j \in \{0, \dots, k\}$, define the residual $R_j(\tilde{u}_k) \in V_j$ related to the mesh T_j by the relation

$$R_j(\tilde{u}_k) \equiv P_j(f + \Delta_k \tilde{u}_k), \tag{10.18}$$

where P_j is the L^2-projection onto V_j and $\Delta_k : V_k \to V_k$ is the discrete Laplacian on V_k.

The multigrid algorithm consists of a sequence of smoothing operations $V_j \to V_j$ (e.g. Jacobi, Gauss–Seidel or ILU iterations) on the different meshes T_j, which are together with grid transfer operations (prolongations and restrictions). The objective of the multigrid algorithm is to make the residual $R_k(\tilde{u}_k)$ on the final mesh T_k small, which is realized in a hierarchical process that also makes the residuals $R_j(\tilde{u}_k)$ small for $j = 0, 1, \dots, k-1$. We assume that $R_0(\tilde{u}_k) = 0$, which corresponds to solving the discrete equations exactly on the coarsest mesh. The details of the multigrid method are immaterial for the a posteriori error estimate to be given.

We now state and prove the a posteriori error estimate and then briefly discuss a corresponding adaptive algorithm.

Theorem 14 For $m \in \{0, 1\}$, there are constants C_{i} and S_{c} such that, if u is the solution of (10.1) and $\tilde{u}_k \in V_k$ is an approximate finite-element solution with $R_0(\tilde{u}_k) = 0$, then

$$\|D^m(u - \tilde{u}_k)\| \le S_{\mathrm{c}} C_{\mathrm{i}} \left\{ \|h_k^{2-m} R(\tilde{u}_k)\| + \sum_{j=1}^{k} \|h_{j-1}^{2-m} R_j(\tilde{u}_k)\| \right\}. \tag{10.19}$$

If $m = 1$, or if $m = 0$ and Ω is convex, then $S_{\mathrm{c}} = 1$.

Proof. For $m = 0$ or $m = 1$, let $\varphi \in H_0^1(\Omega)$ be the solution to the dual continuous problem

$$(\nabla v, \nabla \varphi) = (D^m v, D^m e), \quad \forall v \in H_0^1(\Omega).$$

Taking $v \equiv e$, we obtain the error representation

$$\|D^m e\|^2 = (\nabla e, \nabla \varphi) = (f, \varphi) - (\nabla \tilde{u}_k, \nabla \varphi) \equiv \langle r(\tilde{u}_k), \varphi \rangle.$$

For $j \leq k$, we have the telescoping identity

$$\langle r(\tilde{u}_k), \varphi\rangle = \langle r(\tilde{u}_k), \varphi - \pi_k\varphi\rangle + \sum_{j=1}^{k}\langle r(\tilde{u}_k), \pi_j\varphi - \pi_{j-1}\varphi\rangle + \langle r(\tilde{u}_k), \pi_0\varphi\rangle\,,$$

where π_j denotes the interpolation operator into V_j related to the mesh T_j. Observing that for $v \in V_j$, since $V_j \subset V_k$,

$$\langle r(\tilde{u}_k), v\rangle = (P_j(f + \Delta_k\tilde{u}_k), v) = (R_j(\tilde{u}_k), v),$$

and that $R_0(\tilde{u}_k) = 0$ by assumption, we reduce this to

$$\langle r(\tilde{u}_k), \varphi\rangle = \langle r(\tilde{u}_k), \varphi - \pi_k\varphi\rangle + \sum_{j=1}^{k}(R_j(\tilde{u}_k), \pi_j\varphi - \pi_{j-1}\varphi)\,.$$

Hence, we obtain using Lemma 4 and Lemma 5

$$\|D^m e\|^2 \leq C_{\mathrm{i}}\left\{\|h_k^{2-m}R(\tilde{u}_k)\|\|D^{2-m}\varphi\| + \sum_{j=1}^{k}\|h_{j-1}^{2-m}R_j(\tilde{u}_k)\|\|D^{2-m}\varphi\|\right\},$$

from which the assertion follows using Lemma 3. □

Remark 8 For the exact solution u_h of the finite-element equation (10.7), the a posteriori error estimate has the familiar form

$$\|D^m(u - \tilde{u}_h)\| \leq S_{\mathrm{c}}C_{\mathrm{i}}\|h_k^{2-m}R(\tilde{u}_h)\|. \qquad (10.20)$$

Remark 9 The effect of round-off in the computation of the discrete solution $\tilde{u}_k$ may be taken into account as follows: Suppose the multigrid computation is carried out in single precision. The a posteriori error estimate is valid if the residuals $R(\tilde{u}_k)$ and $R_j(\tilde{u}_k)$, $j = 0, 1, \ldots, k$, are evaluated exactly. In practice, this means in double precision. We can also add the term $\|R_0(\tilde{u}_k)\|$ to take into account that $R_0(\tilde{u}_k) = 0$ in single precision only. If the a posteriori error estimators evaluated in single and double precision differ by more than the chosen tolerance, then the entire computation should be redone in double precision.

11. Adaptive algorithms

The stopping criterion of an adaptive algorithm based on Theorem 14 takes the form

$$S_{\mathrm{c}}C_{\mathrm{i}}\max\left(\sum_{j=1}^{k}\|h_{j-1}^{2-m}R_j(\tilde{u}_k)\|, \|h_k^{2-m}R(\tilde{u}_k)\|\right) \leq \frac{1}{2}TOL, \qquad (11.1)$$

which ensures that the Galerkin-discretization and the discrete-solution errors are equilibriated. The form of the stopping criterion for the discrete-solution error suggests how to monitor the smoothing process on the different levels to make the different residuals $R_j(\tilde{u}_k)$ appropriately small.

11.1. A posteriori error estimates in the $L^\infty(\Omega)$ norm

We now give an a posteriori error estimate for the Galerkin-discretization error $u - u_h$ in the $L^\infty(\Omega)$ norm $\|\cdot\|_\infty$.

Theorem 15 There is a constant $C_{\rm i}$ such that

$$\|u - u_h\|_\infty \leq S_{\rm c} L_h C_{\rm i} \|h^2 R(u_h)\|_\infty, \tag{11.2}$$

with $S_{\rm c} \equiv \max_{y\in\bar\Omega} \|\log(|-y|)^{-1} D^2 \psi_y\|_1$, where ψ_y is the Green's function for Δ on Ω with pole at $y \in \Omega$, and $L_h \equiv (1 + \log(1/h_{\rm min}))$, where $h_{\rm min}$ is the minimal mesh size of T_h. There is a constant C such that $S_{\rm c} \leq C$ for all polygonal domains Ω of diameter at most one.

Proof. The proof is based on the following error representation:

$$(u - u_h)(y) = \int_\Omega f(\psi_y - \pi_h \psi_y)\,{\rm d}x - \int_\Omega \nabla u_h \cdot \nabla(\psi_y - \pi_h \psi_y)\,{\rm d}x,$$

from which the desired estimate follows by arguments analogous to those used above. □

Example 10. We now present results obtained using adaptive algorithms based on Theorem 15 for L^∞ control and Theorem 14 for energy-norm control with $m = 1$ and $S_{\rm c} = 1$, where Ω is the L-shaped domain $(-1,1) \times (-1,1) \setminus (0,1) \times (-1,0)$. We consider a case with an exact solution u with a singularity at the nonconvex corner, given by $u(r,\theta) = r^{\frac{2}{3}} \sin(\frac{2}{3}\theta)$ in polar coordinates.

In the case of maximum-norm control, the stability factor $S_{\rm c}$ is determined by computing approximately ψ_y for some sample points y. In this case apparently, a few choices of y are sufficient. The interpolation constant is set to $C_{\rm i} = 1/8$. In Figure 13, we present the initial mesh (112 nodes and 182 elements) and the level curves of the exact solution. In Figure 14, we show the final mesh (217 nodes and 382 triangles) produced by the adaptive algorithm with $TOL = 0.005$. Figure 15 shows the variation of the efficiency index and the stability constant S_c as functions of the number of refinement levels. The efficiency index, defined as the ratio of the error to the computed bound, increases slightly from the coarsest to the finest grid. In Figure 16, we show the final mesh (295/538) using energy-norm error control with $TOL = 0.005$. Note that the final meshes in the two cases are considerably different.

12. The heat equation

We briefly consider the extension of the results for the scalar equation $u' + au = f$ with $a \geq 0$ given above, to the heat equation, the standard model

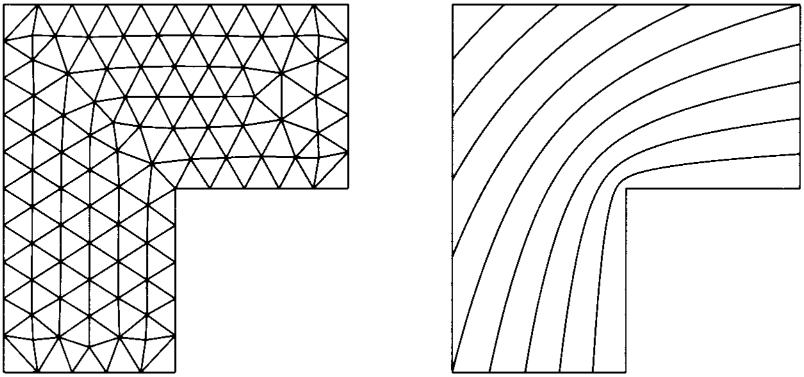

Fig. 13. Original mesh and isolines of the solution on a fine mesh

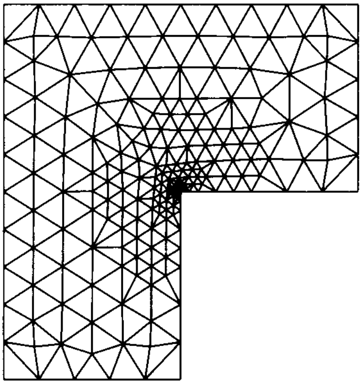

Fig. 14. Maximum-norm control of the error

problem of parabolic type: find $u = u(x,t)$ such that

$$\begin{aligned} &u_t - \Delta u = f, && (x,t) \in \Omega \times I, \\ &u = 0, && (x,t) \in \partial\Omega \times I, \\ &u = u_0, && x \in \Omega, \end{aligned} \tag{12.1}$$

where Ω is a bounded polygonal domain in $\mathbb{R}^2$, $I = (0,T)$ is a time interval, $u_t \equiv \partial u/\partial t$ and the functions f and u_0 are given data.

For discretization of (12.1) in time and space we use the dG(r) method based on a partition $0 \equiv t_0 < t_1 < \cdots < t_n < \cdots < t_N \equiv T$ of I and associate with each time interval $I_n \equiv (t_{n-1}, t_n]$ of length $k_n \equiv t_n - t_{n-1}$ a triangulation T_n of Ω with mesh function h_n and the corresponding space

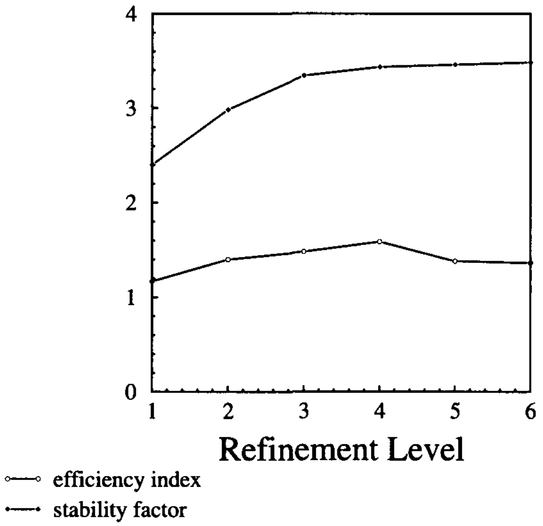

Fig. 15. Stability constant and efficiency index on the different refinement levels

$V_n \in H_0^1(\Omega)$ of piecewise-linear functions as in Section 5. Note that we allow the space discretizations to change with time. We define

$$V_{rn} \equiv \left\{ v : v = \sum_{j=0}^{r} t^j \varphi_j, \quad \varphi_j \text{ in } V_n \right\},$$

and discretize (12.1) as follows: find U such that for $n = 1, 2, \ldots$, $U|_{\Omega \times I_n} \in V_{rn}$ and

$$\int_{I_n} \{(U_t, v) + (\nabla U, \nabla v)\} \mathrm{d}t + ([U]_{n-1}, v_{n-1}^+) = \int_{I_n} (f, v) \mathrm{d}t, \quad \forall v \in V_{rn}, \quad (12.2)$$

where $[w]_n \equiv w_n^+ - w_n^-$, $w_n^{+(-)} \equiv \lim_{s \to 0^{+(-)}} w(t_n + s)$ and $U_0^- = u_0$.

As above, if $r = 0$, then (12.2) reduces to a variant of the Euler backward method, and for $r = 1$ it reduces to a variant of the subdiagonal Padé scheme of order (2,1), that is third-order accurate in U_n^- at the nodal points t_n.

The a posteriori error estimate in the case $r = 0$ has the form

$$\|u(t_N) - U_N\|_2 \le C_{\mathrm{i}} L_N \max_{n=1,\ldots,N} (\|h_n^2 R(U_n)\| + \|[U_{n-1}]\| + \|h_n^2 k_n^{-1} [U_{n-1}]\|^*), \quad (12.3)$$

where $R(U)$ is defined by (10.11), $L_N \equiv \max_{n=1,\ldots,N} (1 + \log(\frac{t_n}{k_n}))^{\frac{1}{2}}$ is a logarithmic factor and the starred term is present only if the space mesh changes at time t_{n-1}. The analogous a priori error estimate assuming $h_n^2 \le k_n$

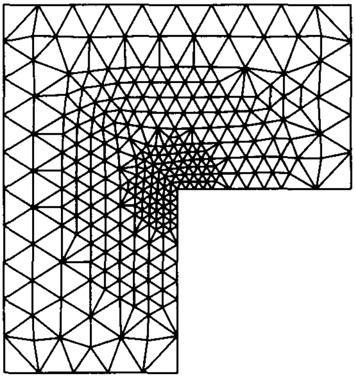

Fig. 16. Energy-norm control of the error

takes the form

$$\|u(t_N) - U_N\| \le C_{\mathrm{i}} L_N \max_{n=1,\dots,N} (\|h_n^2 D^2 u\|_{I_n} + \|k_n u_t\|_{I_n}), \tag{12.4}$$

An adaptive algorithm may be based on (12.3). We note the optimal character of (12.4) and (12.3), that in particular allows long-time integration without error accumulation.

13. References

We give here a brief account of the current status of the development of the framework for adaptive approximation of differential equations that we have described. We also give some references to the extensive literature on adaptive methods that are of particular relevance for our work.

Adaptive methods for linear elliptic problems with energy-norm control were first developed by Babuška *et al.* (see Babuška (1986) and references therein) and Bank *et al.* (see Bank (1986)). In both cases, a posteriori error estimates were obtained by solving local problems with the residual acting as data. Residual-based a posteriori energy-norm error estimates were also derived for Stokes's equations in Verfürth (1989).

The basic approach we use for adaptive methods for linear elliptic problems, including a priori and a posteriori error estimates in H^1, L^2 and L^∞ norms, is presented in Eriksson and Johnson (1991) and Eriksson (to appear). Extensions to adaptive control of the discrete-solution error using multigrid methods is developed in Becker *et al.* (1994). Nonlinear elliptic problems including obstacle and plasticity problems are considered in Johnson (1992*a*), Johnson and Hansbo (1992*a*) and Johnson and Hansbo

(1992*b*). Recently, applications to eigenvalue problems have been given in Nystedt (in preparation).

Early a posteriori error analysis for ordinary differential equations was used in Cooper (1971) and Zadunaisky (1976). These approaches are quite different from ours. We develop adaptive global error control for systems of ordinary differential equations in Johnson (1988), Estep (to appear), Estep and French (to appear), Estep and Johnson (1994), and Estep and Williams (in preparation). Lippold (1988) had an influence on our early work.

The series Eriksson and Johnson (1991), Eriksson and Johnson (1994*a, b, c, d*), (to appear), Eriksson, Johnson and Larsson (1994) develops adaptive Fem for a class of parabolic problems in considerable generality including space-time discretization that is variable in space-time, and applications to nonlinear problems.

Adaptive Fem for linear convection–diffusion problems is considered in Johnson (1990), Eriksson and Johnson (1993) and Eriksson and Johnson (to appear). Extensions to the compressible Euler equations are given in Hansbo and Johnson (1991) and Johnson and Szepessy (to appear). Extensions to the Navier–Stokes equations for incompressible flow are given in Johnson, et al. (to appear), Johnson and Rannacher (1993) and Johnson, *et al.* (1994). Second order wave equations are considered in Johnson (1993).

The presented framework also applies to Galerkin methods for integral equations. An application to integral equations is given in Asadzadeh and Eriksson (to appear). The potential of the framework is explored in Carstensen and Stephan (1993).

14. Conclusion, open problems

The framework for deriving a posteriori error estimates and designing adaptive algorithms for quantitative error control may be applied to virtually any differential equation. The essential difficulties are (i) the computational estimates of stability factors and (ii) the design of the modification strategy. The reliability depends on the accuracy of the computed stability factors and may be increased by increasing the fraction of the total work spent on stability factors. Optimization of computations of stability factors is an important open problem. Optimal design of the modification criterion is also largely an open matter for complex problems. Thus, the contours of a general methodology for adaptive error control seem to be visible, but essential concrete algorithmic problems connected mainly with (i) and (ii) remain to be solved. The degree of difficulty involved depends on the features of the underlying problem related to, for example, stability and nonlinearities.

The concept of computability as a measure of computational complexity is central. A basic problem in mathematical modelling is to develop mathematical models for which solutions are computable. A basic problem of this form is turbulence modelling. Isolating computational errors from modelling errors gives the possibility of evaluating and improving the quality of mathematical models.

To sum up, it appears to be possible to develop reliable and efficient adaptive computational software for a large class of differential and integral equations arising in applications, which could be made available to a large group of users from calculus students to engineers and scientists. If such a program can be successfully realised, it will open up entirely new possibilites in mathematical modelling.

15. How to obtain Femlab and 'Introduction to Numerical Methods for Differential Equations'

Femlab contains software for solving: (i) one dimensional, two point boundary value problems (Femlab-1d); (ii) initial value problems for general systems of ordinary differential equations (Femlab-ode); and two dimensional boundary valve problems (Femlab-2d). Femlab, together with the educational material Eriksson et al. (1994), can be obtained over the Internet. Femlab-ode can be obtained by anonymous ftp to

ftp.math.gatech.edu.

Change to directory */pub/users/estep* and get *femlabode.tar.* This tar file contains the codes and a brief user's manual. In that same directory is *intro.ps.Z*, a compressed postscript version of *Eriksson et al.* [1994].

To obtain Femlab-1d and Femlab-2d open to the WWW (World Wide Web) address

http://www.math.chalmers.se/ kenneth

using (for instance) the Mosaic program. There is a README file located there that gives further instructions.

Femlab-1d consists of a number of Matlab script files and M-files. You can import these files using the 'save as ...' command under the 'file' menu. To run the code, you then just start your local Matlab program and give the command adfem, calling the script file adfem.m. For more details, see the README file.

REFERENCES

M. Asadzadeh and K. Eriksson (1994), 'An adaptive finite element method for a potential problem', *M3AS*, to appear.

I. Babuška (1986), 'Feedback, adaptivity and *a posteriori* estimates in finite elements: Aims, theory, and experience', in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, (I. Babuška, O. C. Zienkiewicz, J. Gago and E. R. de A. Oliveira, eds.), Wiley, New York, 3–23.

R. Bank (1986), 'Analysis of local a posteriori error estimate for elliptic equations', in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations* (I. Babuška, O. C. Zienkiewicz, J. Gago and E. R. de A. Oliveira, eds.), Wiley, New York.

R. Becker, C. Johnson and R. Rannacher (1994), 'An error control for multigrid finite element methods', Preprint #1994-36, Department of Mathematics, Chalmers University of Technology, Göteborg.

C. Carstensen and E. Stephan (1993), 'Adaptive boundary element methods', Institute für Angewandt Mathematik, Universität Hannover.

G. Cooper (1971), 'Error bounds for numerical solutions of ordinary differential equations', *Num. Math.* **18**, 162–170.

K. Eriksson (1994), 'An adaptive finite element method with efficient maximum norm error control for elliptic problems', *M3AS*, to appear.

K. Eriksson, D. Estep, P. Hansbo and C. Johnson (1994*a*), *Adaptive Finite Element Methods*, North Holland, Amsterdam, in preparation.

Eriksson, *et al* (1994) K. Eriksson, D. Estep, P. Hansbo and C. Johnson (1994*b*), 'Introduction to Numerical Methods for Differential Equations', Department of Mathematics, Chalmers University of Technology, Göteborg.

K. Eriksson and C. Johnson (1988), 'An adaptive finite element method for linear elliptic problems', *Math. Comput.* **50**, 361–383.

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

K. Eriksson and C. Johnson (1993), 'Adaptive streamline diffusion finite element methods for stationary convection–diffusion problems', *Math. Comp.* **60**, 167–188.

K. Eriksson and C. Johnson (1994*a*), 'Adaptive finite element methods for parabolic problems II: Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$', *SIAM J. Numer Anal.* to appear.

K. Eriksson and C. Johnson (1994*b*), 'Adaptive finite element methods for parabolic problems III: Time steps variable in space'. in preparation.

K. Eriksson and C. Johnson (1994*c*), 'Adaptive finite element methods for parabolic problems IV: Non-linear problems', *SIAM J. Numer Anal.* to appear.

K. Eriksson and C. Johnson (1994*d*), 'Adaptive finite element methods for parabolic problems V: Long-time integration', *SIAM J. Numer Anal.* to appear.

K. Eriksson and C. Johnson (1994*e*), 'Adaptive streamline diffusion finite element methods for time-dependent convection–diffusion problems', *Math. Comp.* to appear.

K. Eriksson, C. Johnson and S. Larsson (1994), 'Adaptive finite element methods for parabolic problems VI: Analytic semigroups', Preprint, Department of Mathematics, Chalmers University of Technology, Göteborg.

D. Estep (1994), 'A posteriori error bounds and global error control for approximations of ordinary differential equations', *SIAM J. Numer. Anal.* to appear.

D. Estep and D. French (1994), 'Global error control for the continuous Galerkin finite element method for ordinary differential equations', *RAIRO M.M.A.N.* to appear.

D. Estep and C. Johnson (1994*a*), 'The computability of the Lorenz system', Preprint #1994-33, Department of Mathematics, Chalmers University of Technology, Göteborg.

D. Estep and C. Johnson (1994*b*), 'An analysis of quadrature in Galerkin finite element methods for ordinary differential equations', in preparation.

D. Estep and S. Larsson (1993), 'The discontinuous Galerkin method for semilinear parabolic problems', *RAIRO M.M.A.N.* **27**, 611–643.

D. Estep and A. Stuart (1994), 'The dynamical behavior of Galerkin methods for ordinary differential equations and related quadrature schemes', in preparation.

D. Estep and R. Williams (1994), 'The structure of an adaptive differential equation solver', in preparation.

P. Hansbo and C. Johnson (1991), 'Adaptive streamline diffusion finite element methods for compressible flow using conservation variables', *Comput. Methods Appl. Mech. Engrg.* **87**, 267–280.

C. Johnson (1988), 'Error estimates and adaptive time step control for a class of one step methods for stiff ordinary differential equations', *SIAM J. Numer. Anal.* **25**, 908–926.

C. Johnson (1990), 'Adaptive finite element methods for diffusion and convection problems', *Comput. Methods Appl. Mech. Engrg.* **82**, 301–322.

C. Johnson (1992*a*), 'Adaptive finite element methods for the obstacle problem, *Math. Models Methods Appl. Sci.* **2**, 483–487.

C. Johnson (1992*b*), 'A new approach to algorithms for convection problems based on exact transport + projection', *Comput. Methods Appl. Mech. Engrg.* **100**, 45–62.

C. Johnson (1993*a*), 'Discontinuous Galerkin finite element methods for second order hyperbolic problems', *Comput. Methods Appl. Mech. Engrg.* **107**, 117–129.

C. Johnson (1993*b*), 'A new paradigm for adaptive finite element methods', in *Proc. Mafelap 93, Brunel Univ.*, Wiley Comput. Methods Appl. Mech. Engrg. vol. 107, Wiley, New York, 117–129.

C. Johnson and P. Hansbo (1992*a*), 'Adaptive finite element methods for small strain elasto-plasticity', in *Finite Inelastic Deformations – Theory and Applications* (D. Besdo and E. Stein, eds.), Springer, Berlin, 273–288.

C. Johnson and P. Hansbo (1992*b*), 'Adaptive finite element methods in computational mechanics', *Comput. Methods Appl. Mech. Engrg.* **101**, 143–181.

C. Johnson and R. Rannacher (1994), 'On error control in CFD', in *Proc from Conf. on Navier–Stokes Equations Oct 93, Vieweg*, to appear.

C. Johnson, R. Rannacher, and M. Boman (1994*a*), 'Numerics and hydrodynamic stability: Towards error control in CFD', *SIAM J. Numer. Anal.* to appear.

C. Johnson, R. Rannacher, and M. Boman (1994*b*), 'On transition to turbulence and error control in CFD', Preprint #1994-26, Department of Mathematics, Chalmers University of Technology, Göteborg.

C. Johnson and A. Szepessy (1994), 'Adaptive finite element methods for conservation laws based on a posteriori error estimates', *Comm. Pure Appl. Math.* to appear.

G. Lippold (1988), 'Error estimates and time step control for the approximate solution of a first order evolution equation', preprint, Akademie der Wissenschaften der Karl Weierstrass Institut für Mathematik, Berlin.

C. Nystedt (1994), 'Adaptive finite element methods for eigenvalue problems', Licentiate Thesis, Department of Mathematics, Chalmers University of Technology, Göteborg in preparation.

R. Verfürth (1989), 'A posteriori error estimators for the Stokes equations', *Numer. Math.* **55**, 309–325.

P. Zadunaisky (1976), 'On the estimation of errors propagated in the numerical integration of ordinary differential equations', *Numer. Math.* **27**, 21–39.