RECENT ADVANCES IN ERROR ESTIMATION
AND ADAPTIVE IMPROVEMENT OF
FINITE ELEMENT CALCULATIONS

J.T. ODEN, T. STROUBOULIS,
PH. DEVLOO, and M. HOWE

Texas Institute for Computational Mechanics,
The University of Texas at Austin
Austin, Texas 78712

Abstract. We collect in this article a synopsis of methods and results on adaptive finite element methods. We outline methods for constructing a-posteriori error estimates for linear and nonlinear problems in mechanics. Adaptive methods are described and a variety of numerical results are given on applications to problems in fluid mechanics.

1. INTRODUCTION

How good are the answers? What can be done to improve them? These questions arise with increasing frequency among users of modern computational mechanics codes. They are fundamental, in that such questions relate to the basic goals of computational mechanics: the use of computational methods and devices to simulate mechanical phenomena. Yet much of contemporary research in computational mechanics is concerned with a myriad of other issues which, important as they may be, do not consciously and directly focus on those primitive and fundamental questions. When one does focus on those queries, a sequence of natural constraints are met that have a profound effect on the way one approaches the development of modern codes, numerical schemes, algorithms, and data management techniques for computational mechanics applications:

Modulo natural deficiencies in the ability of the mathematical model itself to capture real physical behavior, we translate the first question into one that can be managed in mathematical terms: how accurate are the numerical solutions? The only plausible and general approach toward answering this question is to construct a-posteriori error estimates; i.e. to use the results of an initial calculation to estimate the local error in a finite element / finite difference approximation.

Having obtained an indication of "how good the answers are," one can proceed to the second question: what can be done to improve them? The answer is clearly to use adaptivity of the approximation in some way: to change the structure of the approximation to improve accuracy, where by "structure" we mean the basic mesh topology, the number and location of nodes and cells, the local order of the approximations, etc.

As is well known, there has emerged in the literature several methods for effectively altering
this structure: h-methods, in which the mesh is automatically refined to reduce error; p-methods, in which the local polynomial degree is increased; r-methods, in which a fixed number of nodal points are redistributed to reduce error; and combined methods, in which h - p, r - h, r - p, -- combinations are employed. A survey of the recent literature on such adaptive methods has been compiled by Oden and Demkowicz [20].

What is especially significant about these answers to the basic question is that they have a great impact on the design of computational mechanics codes. To implement a rational adaptive scheme one must obey the following criteria in designing a programming strategy:

1. **Mesh Independence.** Since the mesh itself may well be changing as the solution evolves, it is necessary to have schemes which can be implemented on arbitrary unstructured or quasi-structured meshes. This first criterion makes obsolete most existing most existing body-fitted coordinate schemes common in finite difference literature.

2. **Robustness.** Since the structure of the approximation is continually changing in an adaptive scheme, adaptive methods must be very stable under changes in mesh size, under mesh distortions, etc.

3. **Mathematical Basis.** Since a -posteriori error estimates are necessary for an effective adaptive scheme, it is necessary that a solid mathematical basis exist for the adaptive methods.

4. **Geometry Independence.** Modern computational methods, adaptive or not, must be able to cope with solution domains of arbitrary, complex geometry. The "real world" problems encountered in applications seldom have simple geometries for which many classical methods work well.

5. **Supercomputing.** The significant data management problems inherent in adaptive strategies must lend themselves to supercomputing strategies-vectorization, parallelism, etc.

6. **Efficiency.** Hopefully, when all features of an adaptive strategy are optimized in a program/computer structure, an efficient analysis tool will emerge. It is not necessary that the final product be capable of analyzing a given discretization as "fast" as possible; rather, the objective is overall optimization: to produce the best possible answers (in some sense) for a fixed level of computational effort.

In our opinion, it is very clear that only finite element methodologies can fulfill all of these criteria.

In this paper, we shall outline several recent advances in developments of the basis components of adaptive methods. We do not attempt to provide a thorough review of the literature, as this has already been the subject of a recent paper [20]. Rather, we provide summary comments in a few areas that we think stand out as important advances in the field. Naturally we are most familiar with our own efforts in this field, so we comment more fully on some of our own results.

Following this Introduction, we give a brief summary of a few recent advances in adaptive finite elements. This is followed by several sections on general ideas behind a-posteriori error estimation, h-method data management, algorithms for fluid-mechanics applications, and some new results on numerical experiments with our adaptive codes. Finally, we comment on future directions of research in this field.

### 2. RECENT ADVANCES

The state-of-the-art in adaptive finite element methods is adequately summarized in the volume of collected works and presentations made at the Lisbon conference of 1985. These have recently appeared under the editorship of Babuska, Zienkiewicz, Gago, and Oliveira [1]. Here one will find
information of the basic ideas of h, p, r methods together with numerous applications to problems in solid mechanics and fluid mechanics.

More recently a number of significant advances have been made which should be brought to view in the area of elliptic problems, we mention the important theoretical work by Guo and Babuska [14] on h-p methods. It is known that one can generally achieve a faster increase in local accuracy using p methods than h methods. By this it is meant that greater accuracy can be achieved with fewer degrees of freedom by increasing the local order p of the polynomial than by refining the mesh. This does not necessarily mean that the p methods offer a superior approach to solving elliptic problems, for one must add to this equation the significant factor of a data management scheme, which is often the life and death of an adaptive method. Babuska and his co-workers have shown, however, that the best possible approach to the accuracy problem, one leading to exponential convergence, is to simultaneously refine both h and p. The h-p methods have shown, in certain example problems, to produce exceptionally accurate results. At this writing, most of these results have been confined to one-dimensional problems and to linear elliptic problems in two-dimensions. There would appear to be some computational difficulties in extending these methods to time dependent problems, since there one must cope with the difficult issue of consistent mass matrices, stability and space-time approximation. However, it is possible that these difficulties may also be overcome with additional research.

A production finite element code based on p methods is now being promoted and sold. This is the PROBE code, and its successful implementation of the p method has already an impact on the design of linearly elastic structures, see [25]. The simple r methods produced by Diaz and Kikuchi, and Taylor [12] have been used effectively in classes of problems in which one wants to keep the number of degrees of freedom more or less constant. In particular, in problems such as metal forming simulations, where one must solve a large number of nonlinear partial differential equations, it is natural to try to achieve the best possible accuracy for a fixed number of nodal points. Some simple moving mesh algorithms have been proposed which are easy to implement and which apparently work well in two and three-dimensional problems. These have proved to be very effective for nonlinear problems in plasticity in nonlinear solid mechanics.

In general, moving mesh methods suffer from one defect: for a fixed number of nodes and fixed degree polynomial within each element, there is an inherent threshold of error which cannot be eliminated. Thus, with the exception of the work of Miller on moving finite element methods and the work mentioned above by Diaz and Kikuchi on r methods, most of the recent work on adaptive methods has focused on h-methods and p-methods.

Perhaps the most significant recent advances in adaptive finite element methods have come in the area of time dependent problems. We mention in this regard the important work of Flaherty and his co-workers (see, for example, [8]) who have developed effective numerical methods for certain classes of parabolic problems. Additional references on this subject can be found in these papers. We also mention the construction of adaptive characteristic Petrov-Galerkin methods by Demkowicz and Oden [9,10] which involves not only the construction of the local a-posteriori estimates but also the construction of near optimal schemes for nonlinear convection diffusion problems with small diffusion coefficients. These results have recently been extended to solve Euler equations in two-dimensions.

One area in which adaptive methods appear to be making some in-roads is in supersonic gas dynamics and general fluid mechanics. Several effective numerical schemes have been proposed by Löhner, Morgan and Zienkiewicz [16,17,18], and the authors [19,21]. These schemes have been used effectively to solve two-dimensional steady state and transient problems in compressible fluid mechanics.

More recently, Oden, Strouboulis and Devloo [19,23,24] have extended these methods to fluid mechanics in which moving domains are encountered. In particular, adaptive schemes have been developed for classes of problems in which flow interaction occurs due to the motion of one body or another through a flow field. Initial results on the application of adaptive methods to supersonic rotor-stator problems have produced some impressive results, some of which are
outlined later in this paper. These include examples in which adaptivity has resulted in a mesh with nearly 70 percent fewer elements than the uniform fine mesh required to produce equivalent accuracy.

We comment on some of the components of an effective adaptive scheme below.

3. A-POSTERIORI ERROR ESTIMATION

The great majority of results on a-posteriori error estimation that have appeared in the literature in recent years is restricted to linear elliptic problems; however, a great deal of precision and depth of results has been possible for problems in this class. In [21,24,9], we have described a general method for a-posteriori error estimation that is applicable to broad classes of linear and nonlinear problems, including parabolic and hyperbolic problems. Successful use of our method in determining error estimates for the finite element approximation of the Navier-Stokes equations has also been made [24].

An outline of the general method is provided by the abstract linear problem:

Find $u \in V$ such that $a(u,v) = f(v)$ \quad $\forall \ v \in V$ \quad (3.1)

where $a(\cdot, \cdot)$ is a bilinear form on $V \times V$, $V$ being a Banach space, and $f$ is a linear functional on $V$. The Galerkin approximation of (3.1) in a finite-dimensional subspace $V_h$ of $V$ is characterized by the problem,

Find $u^h \in V^h$ such that $a(u^h, v^h) = f(v^h)$ \quad $\forall \ v^h \in V^h$ \quad (3.2)

We suppose that $V \rightarrow H = H^* \rightarrow V^*$, the inclusions being dense and continuous, for a Hilbert (pivot) space $H$, $V^*$ being the dual of $V$, etc. If $\langle \cdot, \cdot \rangle$ denotes duality pairing on $V^* \times V$, then we generally have

$$a(u, v) = \langle Au, v \rangle$$

If $V_A = \{ v \in V \mid A v \in H \}$, then we also have $\langle Au,v \rangle = (A u,v)_H$; $u,v \in V_A$ where $(\cdot, \cdot)$ is the inner product on $H$ and $A \in \mathcal{L}(V_A, H)$.

In general, for finite element approximations, the form $a(\cdot, \cdot)$ can be expressed as the sum of contributions from an assembly of $E$ subdomains:

$$u,v \in V: a(u,v) = \sum_e \left[ (A u,v)_e + \Gamma_e(u,v) \right]$$

where $(\cdot, \cdot)$ denotes the $H$-inner product defined on restrictions of $u$ and $v$ to subdomain (element) $e$ and $\Gamma_e(u,v)$ is the bilinear concomitant associated with boundary terms on the boundary of subdomain $e$.

Let $e^h = u - u^h$ denote the error and suppose that

$$\|v\|_A^2 \overset{\text{def}}{=} a(v,v)$$
\[
\|w\|_A = \sup_{v \in V} \frac{a(w, v)}{\|v\|_A}
\]

Then
\[
\|e^h\|_A = \sup_{v \in V} \frac{a(e^h, v)}{\|v\|_A}
\]

\[
= \sup_{v \in V} \|v\|_A^{-1} \{ \sum_e (r^h, v)_e + \Gamma_e (e^h, v) \} \tag{3.3}
\]

where \(r^h = \Delta u - \Delta u^h = f - \Delta u^h\) is the local residual. To eliminate \(r^h\), we construct a local auxiliary problem, for a function \(\theta_e\) defined by
\[
a(\theta^e, v) = R_e (v); \quad e = 1, 2, \ldots, E \tag{3.4}
\]

where \(R_e (v) = (r^h, v)_e + \Gamma_e (e^h^*, v) e^h^*\) being some appropriate approximation of \(e^h\) on the boundary. Setting \(A_i\) the restriction of \(A\) over \(\Omega_e\) and

\[
\|\theta^e\|_{A_i}^2 = a_e (\theta^e, \theta^e) = <A_i \theta^e, \theta^e>
\]

introduce (3.4) into (3.3) to arrive at the a-posteriori estimate,

\[
\|e^h\|_A \leq \{ \sum_e \|\theta^e\|_{A_i} \}^{1/2} \tag{3.5}
\]

The functions \(\theta^e\) are local error indicators. Of course we do not wish to solve the \(E\) equations to obtain the \(\theta^e\). We are, thus, content to construct an approximate solution to (3.4) over some enriched subclass \(V_e^h\) of functions so as to produce approximations \(\theta^h_e\) of \(\theta^e\). Several different methods of a-posteriori error estimation may result from different schemes for approximating (3.4). Alternatively, if one can derive local a-priori bounds such as \(\|\theta^e\|_A \leq C \|R^e\|_{A_i}\), then (3.5) can be rewritten in terms of the residual functional \(R^e\).

In many nonlinear problems, a step such as (3.3) may not hold, and instead, we bound the residual. For example

\[
\|r^h\|_{\ast} = \sup_{v \in V} \frac{<\Delta u - \Delta u^h, v>}{\|v\|} = \sup_{v \in V} \|v\|^{-1} \{ \sum_e (r^h, v)_e + \Gamma_e (e^h, v) \}
\]

\[
\leq \{ \sum_e \|\theta\|_{A_i}^2 \}^{1/2}
\]

We conclude this section with several remarks.

1. These examples provide global a-posteriori error (or residual) bounds in terms of local error indicators. By a special construction of test functions, truly local error estimates can be
obtained. For example, Demkowicz and Oden [9] studied a special Petrov-Galerkin method for the problem - ε u" + u = f, and showed that the local error must satisfy the sharp a-posteriori estimate

\[ \| e^h \|_{L^2(\Omega_e)} \leq \frac{h^2}{h^2 + \varepsilon \pi^2} \| r^h \|_{L^2(\Omega_e)} \]

where \( r^h \) is, again, the element residual.

2. For a time-dependent problem, such as

\[ \int_\Omega (\partial u/\partial t + A(u)) \psi \, dx \, dy = \int_\Omega f \psi \, dx \, dy \]

for arbitrary test functions and \( \psi \), and linear \( A \), the fact that the error must be the function \( e^h = u - u^h \) leads, by direct substitution, to the evolution equation,

\[ \int_\Omega (\partial e^h/\partial t + A(e^h)) \psi \, dx \, dy = - \int_\Omega r^h \psi \, dx \, dy \]

Thus, using a higher order approximation \( E^h \) of \( e^h \) than that used in approximating \( u^h \), we arrive naturally at a system of equations for the evolution of error,

\[ M E + K E = R \quad (3.6) \]

Various dynamic error estimators can be constructed depending on how one constructs the approximation \( E^h \) of \( e^h \). In (3.6), \( M \) is the usual mass matrix associated with the approximation \( E^h = \sum_j E_j(t) \psi_j(x), E \) is the vector of nodal errors \( E_j \), \( K \) is the stiffness matrix, and \( R \) the residual vector.

3. For certain classes of problems, it is possible (or, at least, it may be assumed to be possible) to obtain an estimate,

\[ \| \theta^e - \theta^h \|_{A,e} \leq C \| \theta^e - \psi^h \|_{A,e} \quad \forall \psi^h \in \psi^e \]

where \( \psi^e \) is the special class of local test functions used in approximating the local auxiliary problems (3.4). Then \( \| \theta^e - \psi^h \|_{A,e} \) may, in turn, be estimated using standard results from finite element interpolation theory (see Oden and Carey [22]). In particular, if \( \theta^e \) is the interpolant of \( \theta^e \) over \( \Omega_e \) obtained using polynomials of degree \( \leq k \), for an \( n \)-dimensional problem with quasi-uniform mesh refinements,

\[ \| \theta^e - \theta^h \|_{m,q,\Omega_e} \leq C h^{n/q - n/p + k + 1 - m} \| \theta^e \|_{k+1,p,\Omega_e} \quad (3.7) \]

with \( \| \cdot \|_{m,q,\Omega_e} \) the \( W^{m,q}(\Omega_e) \) seminorm \( 0 \leq p \leq \infty \), and \( q = p/(p - 1) \). For the case \( m = 0, k = 1, p = q = 2 \), we obtain

\[ \| \theta^e - \theta^h \|_{L^2(\Omega_e)} \leq C h^2 \| \theta^e \|_{2,2,\Omega_e} \]

and for \( m = 0, k = 0, q = 1, p = \infty \), we have
These estimates can judge the quality of the approximations of the local indicators, provided a means for computing estimates of the seminorms \( |\theta_{e}|_{k+1,p,c} \) is developed.

4. FEATURES OF AN ALGORITHM APPROPRIATE FOR ADAPTIVE FEM

Earlier in this paper we tested criteria for the development of adaptive finite element codes for complex problems in solid and fluid mechanics. In this section, we summarize features of an adaptive code we have developed for two dimensional problems in compressible gas dynamics in which we have attempted to meet most of these criteria.

4.1 Preliminaries

We consider the motion of a perfect gas flowing through a two-dimensional domain \( \Omega \subset \mathbb{R}^2 \). If \( U = U(x,t) \) is the vector of conservation variables with \( \rho \) the mass density, \( m \) the linear momentum and \( e \) the total energy, it satisfies the following weak initial -boundary value problem:

Find \( U \in V \) such that

\[
\int_{\Omega} \int_{0}^{T} (U^T \frac{\partial \phi}{\partial t} + Q(U) : \nabla \phi) \, d\Omega \, dt + \int_{\Omega} U_T^0 \phi(\cdot, 0) \, d\Omega = \int_{\partial \Omega} F^T \phi \, ds \, dt
\]

\( \forall \phi \in W \)

Here \( Q(U) \) is the Euler flux tensor,

\[
Q(U) = \begin{bmatrix}
    m_1 & m_2 \\
    \rho^{-1} m_1^2 + p(U) & \rho^{-1} m_1 m_2 \\
    \rho^{-1} m_1 m_2 & \rho^{-1} m_1^2 + p(U) \\
    \rho^{-1} m_1 (e + p(U)) & \rho^{-1} m_2 (e + p(U))
\end{bmatrix}
\]

\( p(U) = (\gamma - 1)(e - \rho^{-1} m \cdot m / 2) \)

where \( p \) is the thermodynamic pressure and \( \gamma \) is the ratio of specific heats.

Moreover,

\[
V = \{ \, v = (v_1, v_2, v_3, v_4)^T \mid v_i = v_i(x, t) \in L^\infty(0, T; L^1(\Omega)); i = 1,2,3,4 \} \quad (4.3)
\]

\[
W = \{ \, w = (w_1, w_2, w_3, w_4)^T \mid w_i \in C^1[\Omega, T]), w_i(x, T) = 0 \, ; \, i = 1,2,3,4 \} \quad (4.4)
\]

\( F \) is the actual prescribed flux through the boundary \( \partial \Omega \) and the following notation is used...
Let us now consider an arbitrary time interval \([\tau_1, \tau_2]\) \(\subset [0,T]\) and modify the space of test functions to include functions which do not vanish at the final time, namely:

\[
W^{\tau_1, \tau_2} = \{ w = (w_1, w_2, w_3, w_4)^T | w_i \in C^1(\Omega \times [\tau_1, \tau_2]) ; \ i = 1,2,3,4\}
\]

Then we can state the weak-statement of the conservation laws over the space time subdomain \(\Omega \times [\tau_1, \tau_2]\) as follows:

Find \(U \in V^{\tau_1, \tau_2}\) such that

\[
\int_{\Omega} (U^T(\cdot, \tau_2) \phi (\cdot, \tau_2)) \, d\Omega = \int_{\Omega} (U^T(\cdot, \tau_1) \phi (\cdot, \tau_1)) \, d\Omega
\]

\[
+ \int_{\tau_1}^{\tau_2} \int_{\Omega} (U^T \frac{\partial \phi}{\partial t} + Q : \nabla \phi) \, d\Omega \, dt
\]

\[
- \int_{\tau_1}^{\tau_2} \int_{\Omega} F^T \phi \, dy \, dt \quad \forall \phi \in W^{\tau_1, \tau_2} \ (4.5)
\]

Here \(V^{\tau_1, \tau_2}\) is appropriately defined as the solution space over the strip \(\Omega \times [\tau_1, \tau_2]\).

### 4.2 Solution Algorithm

We obtain a finite element approximation of (4.1) by partitioning the space-time domain \(\Omega \times [0,T]\) into subdomain \(\Omega \times [t_n, t_{n+1}]\) (with \(0 = t_0 < t_1 < ... < t_n < t_{n+1} < ... t_N = T\)) by discretizing each subdomain and by employing (4.5) using the discrete spaces of test and trial functions defined by the discretization. Moreover, by approximating the space-time integrals using numerical integration we get the following scheme [19]:

**I. First Step:**

For each element \(\Omega_e\), compute \(U_e^{n+1/2}\) such that,

\[
U_e^{n+1/2} \int_{\Omega_e} d\Omega = \int_{\Omega_e} U_h^n d\Omega - \Delta t/2 \int_{\Omega_e} \text{div} \, Q(U_h^n) \, d\Omega \quad (4.6)
\]

**II. Second Step:**

Calculate \(U_h^{n+1}\) such that,

\[
\int_{\Omega} \phi_h^T U_h^{n+1} d\Omega = \int_{\Omega} \phi_h^T U_h^n d\Omega + \Delta t \int_{\Omega} Q(U_h^{n+1/2}) : \nabla \phi_h d\Omega
\]
Here we assumed $d<\phi_h/dt = 0$ (i.e. the spatial grid remains fixed), we let $F = Q \phi_h$ and denote $V_n = V_{n+1}$. Equations (4.6), (4.7) define a two-step TG/FELW (Taylor-Galerkin/Finite-Element-Lax Wendroff) method which has been introduced by Donea [13], studied by Baker et al. [2], refined by Löhner et al. [17] and others ([3], [19]). The second step of the scheme, as given in (4.7) involves a global calculation of the form:

$$M \{ U \}_{n+1} = \{ R \}$$

Here $M$ denotes the consistent mass matrix, $\{ R \}$ the load vector whose definition can be easily deduced from (4.7) and $\{ U \} = \{ U_1, U_2, U_n, U_{n+1}, \ldots, U_n \}^T$ is the global vector of nodal unknowns. The inversion of the mass matrix can be performed by a Jacobi iteration [17] or a preconditioned Jacobi Conjugate Gradient [19].

The TG/FE-LW method provides us with a fast, multi-dimensional time stepping algorithm with a high resolution (high order of accuracy) in smooth regions of flow and which applies to unstructured adaptive grids. It is well known [17] that the algorithm suffers from a phenomenon of non-linear instability. To overcome this deficiency, artificial diffusion is added to stabilize the scheme in the presence of discontinuities ([18], [19]).

4.3 Flux-Corrected Transport

The theory of Flux Corrected Transport has been developed by Boris, Book and others ([4], [5], [6]) and it involves an attempt to systematically correct finite-difference transport schemes in order to avoid non-physical oscillations in the solution. Fully multi-dimensional FCT schemes have been constructed by Zalesak [26]. Recently Löhner et al. [18] presented a flux-correction procedure of the TG/FELW scheme for systems of conservation laws. In this section we give a short exposition of the FCT - TG/FE-LW algorithm which we employed in some of our adaptive calculations.

The FCT procedure consists of solving equation (4.9) by using a diffusion and an antidiffusion step. In the diffusion step a "strong" diffusion term is added to obtain a "transported and diffused" solution which is free of non-physical oscillations. In the antidiffusion step part, a "limited" amount of diffusion is subtracted from the right hand side (4.9) in order to steepen the solution at discontinuities and increase the accuracy in "smooth" regions of flow.

In particular, we have:

**Step I: "Diffusion" Step**

Compute $\{ U_d^{n+1} \}$ from

$$M \{ U_d^{n+1} \} = \{ R \} + \{ V \}$$

Here $\{ V \}$ denotes the vector of added diffusion with nodal contributions of the form:
\[ V_i = \int_{\Omega} (D_x \frac{\partial \phi_{hi}}{\partial x} \partial U^n + D_y \frac{\partial \phi_{hi}}{\partial y} \partial U^n ) \, d\Omega \]  

(4.11)

For a mesh of quadrilaterals we let

\[ D_x = D_y = c \, A_e \]

where \( c \) is a constant and \( A_e \) denotes the area of element \( \Omega_e \).

**Step II**: "Antidiffusion" Step.

Compute \( \{ U_{n+1} \} \) as the limit of the sequence of iterates \( \{ U_{n+1} \} \), \( i = 1, 2, 3, \ldots \) defined by:

\[ M_L \{ U_{n+1} \}_{[i+1]} - U_{n+1}^{[i]} = I (F) \]

\[ F = (M_L - M) \Delta U_{n+1}^{[i]} - V \]

(4.12)

Here \( M_L \) denotes the lumped mass matrix and \( I \) denotes the flux limiting function which may be defined appropriately in order to prevent oscillations in the solution. In our applications we used the strategy of Zalesak [26] and Löhner et al. [18] to compute \( I (F_{[i]}) \).

### 4.4 An h Refinement / Unrefinement Strategy for Steady-State Solutions of High-Speed Compressible Flow

An adaptive procedure for steady-state solutions of equations of compressible gas dynamics involves the following steps:

For a given domain a coarse finite element mesh is defined which contains only a number of elements sufficient to model the basic geometric features of the flow domain (see Figure 1a). Each element in the initial mesh is assigned a "level" equal to zero. Then a finer mesh is generated by a bisection process, indicated in Figure 1b, in order to obtain an initial grid with the "group" structure. Note that when an element is refined a group of 4 elements is defined and each of the 4 new elements has a level one unit higher than the "parent" element.

1. For a given finite element grid determine the steady-state solution.

2. Compute error indicators \( \theta_e \) over all \( M \) elements in the grid. Let

\[ \theta_{\text{MAX}} = \max_{1 \leq e \leq M} \theta_e \]

3. We scan groups of 4 elements and compute

\[ \theta_{\text{GROUP}} = \sum_{k=1}^{4} \theta_{m_k} \]
where $m_k$ is the $k$-th element in group $m$.

4. Error tolerances are given by two real numbers, $0 < \alpha, \beta < 1$.

If $\theta_e \geq \beta \theta_{\text{MAX}}$

we refine element $\Omega_e$ by bisecting it into four new elements.

If $\theta_{\text{GROUP}} \leq \alpha \theta_{\text{MAX}}$

we unrefine group $m$ by replacing this group with a single new element with the nodes coincident with the corner nodes of the group.

5. Go to step 1.

4.5 Numerical Examples

In this section we present examples of adaptive calculations of steady-state solutions of problems in high speed compressible flow. The error indicator employed in the numerical examples is given by the normalized gradient of the density:

$$
\theta_e = A_e \frac{\max_{i=1,2} \left| \frac{\partial \rho_h}{\partial x_i} \right|}{\rho_h}
$$

where $\rho_h$ denotes an average value of the density of element $\Omega_e$.

4.5.1 Supersonic Flow Over a 20° Ramp

We consider the problem of a Mach 3 flow (with $\gamma = 1.40$) over a 20° ramp. The gas enters with uniform flow conditions through the left boundary of the domain and develops an oblique shock at the root of the ramp.

A coarse initial mesh with the computed pressure contours are illustrated in Fig. 2. Adaptive mesh results are shown in Figures 3 and 4 with one and two levels of refinement respectively. The constants for the adaptive scheme were chosen $\alpha = 0.05$, $\beta = 0.15$. The FCT version of the time-stepping algorithm was employed with $c = 0.125$. The results compare well with the exact solution except for some small disturbances downstream which are due to the artificial stagnation point at the tip of the corner. A three-dimensional view of the pressure is shown in Figure 5.

4.5.2 Supersonic Flow in Expansion Corner

In this example, the steady supersonic flow through a 10° expansion is studied. The inflow Mach number was selected $M_{\infty} = 6$ with $\gamma = 1.38$. Figures 6 through 8 show the meshes
Figure 1. 
(a) A coarse initial mesh consisting of 4-element groups.
(b) The refinement and unrefinement of a group of elements.
Figure 2. Supersonic flow over a 20° ramp. Initial mesh and pressure contours.
Figure 3.  Supersonic flow over a 20° ramp.  
Mesh and pressure contours obtained with one level of refinement.
Figure 4. Supersonic flow over a $20^\circ$ ramp. Mesh and pressure contours obtained with two levels of refinement.
Figure 5. Supersonic flow over a 20° ramp. Three-dimensional view of the converged pressure function obtained with two levels of refinement.
Figure 6. Supersonic expansion around a 10° corner. Initial mesh and density contours.
Figure 7. Supersonic expansion around a 10° corner. Mesh and density contours obtained with one level of refinement.
Figure 8. Supersonic expansion around a 10° corner. Mesh and density contours obtained with two levels of refinement.
employed in the calculation with the corresponding density contours. The results were obtained with the FCT scheme with \( \alpha = 0.05, \beta = 0.15 \) and \( c = 0.125 \). Striking improvement in the solution is seen to result from the refinement procedure.

### 4.5.3 Asynchronous Time-Stepping Procedures

In the algorithms described in the previous paragraph the global timestep \( \Delta t \) is determined as the minimum allowable time step in the grid, namely:

\[
\Delta t = \min_{e=1,...,M} \Delta t_e; \quad \Delta t_e = \frac{C \sqrt{A_e}}{\text{lul} + c}
\]

(4.14)

Here \( C \) denotes the C.F.L. number, \( c \) is the local speed of sound in the element and \( \text{lul} = u_1^2 + u_2^2 \).

From the definition (4.14) we see that since \( \Delta t \sim C \sqrt{A_e} \sim h_e \) the timestep may be governed by the smallest element in the mesh. This choice of \( \Delta t \) guarantees stability and time-accuracy of the scheme. For steady-state calculations however time-accuracy is not important and it may be more economical to employ asynchronous time-stepping by prescribing local time-steps.

Let us denote by \( \Delta t_{\text{node}}^j \) the nodal timestep of node \( j \) which is computed by the minimum of the time-steps of the elements which are connected to node \( j \). Then, an Asynchronous TG/FE-LW scheme may be employed as follows:

**I. First Step:**

For each element \( \Omega_e \), compute \( U_e^{n+1} \) such that:

\[
U_e^{n+1} \int_{\Omega_e} d\Omega = \int_{\Omega_e} U_h^n d\Omega - \frac{\Delta t_e}{2} \int_{\Omega_e} \text{div} \left( Q \left( U_h^n \right) \right) d\Omega
\]

(4.15)

**II. Second Step:**

Calculate \( U_{h}^{n+1} = \sum_{j=1}^{N} U_{h}^{i,j,n+1} \phi_i \) such that,

\[
\sum_{i=1}^{N} \left( \int_{\Omega} \phi_j^T \phi_i d\Omega \right) U_{h}^{i,n+1} = \sum_{i=1}^{N} \left( \int_{\Omega} \phi_j^T \phi_i d\Omega \right) \\
+ \Delta t_{\text{node}}^j \int_{\Omega} Q \left( U_h^{n+1/2} \right) : \nabla \phi_j d\Omega \\
- \Delta t_e \int_{\partial\Omega} \phi_j^T \left( Q \left( U_h^{n+1/2} \right) - Q \left( U_h^n \right) \right) n d\gamma \\
+ \Delta t_{\text{node}}^j \int_{\partial\Omega} \phi_j^T Q \left( U_h^n \right) n d\gamma \\
j = 1, 2, \ldots, N
\]

(4.16)
We now demonstrate some of the features of the asynchronous time-stepping scheme using two numerical examples:

### 4.5.3.1. The Reflecting Shock Problem:

The statement of the problem is given in [19]. Figures 9 and 10 show the steady-state density contours obtained with the time-accurate and asynchronous algorithms, respectively. We note that the steady state was obtained after 130 time steps with the time-accurate scheme and after only 100 time-steps with the asynchronous scheme, which represents 30% of savings in computational effort.

### 4.5.3.2. NACA 0012 Airfoil in Supersonic Wind Tunnel:

We also considered the problem of a NACA 0012 airfoil in a supersonic wind tunnel with inflow Mach number $M_{\infty} = 3$, $\gamma = 1.4$ [19]. Figure 11 presents a comparison between the steady-state density contours obtained with the two schemes. The time-accurate scheme requires 585 time-steps to converge while the asynchronous scheme converged after 496 time-steps.

### 4.5.4. Elevon Cove Problem

The Elevon Cove problem has to do with supersonic flow past a complex swan-like geometry of a portion of the space shuttle elevon. The problem is described in [3]. Figures 12 and 13 show a preliminary calculation of the problem with our adaptive Euler code. The mesh shown does not correspond to a later unrefined mesh. This mesh is not yet optimal, since the program was still attempting to compute a new mesh at the time calculations were stopped.

### 5. Features for an Adaptive Finite Element Algorithm for Transient Calculations

We now present an example of an h-refinement / unrefinement strategy for transient calculations. The basic steps of the algorithm are:

a) Advance the solution N time steps.

b) Do the following until no more elements can be refined:

   1. Compute the element error indicators $\theta_e$.
   2. Refine all elements with $\theta_e \geq \beta \theta_{MAX}$
   3. Integrate the last N time steps with the updated (refined) mesh
   4. Go to (1).

c) Compute the element error indicators $\theta_e$ and unrefine all groups with $\theta_{GROUP}^m \leq \alpha \theta_{MAX}$

d) Go to a).

We note that the "do loop" in step b) converges when no more elements can be refined (the maximum level of refinement is fixed). Although the iteration in step b) guarantees a "fully updated" mesh it may lead to an expensive scheme if more than a few passes are required for convergence of the "do loop". A cheaper alternative is presented by the following "two-pass" scheme:

a) Advance the solution N time steps.
b) Compute the element error indicators $\theta_e$.

c) Refine all elements with $\theta_e \geq \beta \theta_{\text{MAX}}$

d) Integrate the last N time steps with the refined mesh obtained in c)

e) Compute the element error indicators $\theta_e$ and

1) Unrefine all groups with $\theta_{\text{GROUP}} \leq \alpha \theta_{\text{MAX}}$

2) Refine all elements with $\theta_e \leq \beta \theta_{\text{MAX}}$

f) Go to step a).

In the following, we present two examples of adaptive refinement for transient problems.

5.1 Rotating Cone Problem [11]

We consider the following advection problem:

$$\frac{\partial U}{\partial t} + \text{div} (a U) = 0$$

$$U(x,y,0) = \begin{cases} 
0, & r \geq 150 \\
250 \left[ 1 + \cos \frac{\pi r}{150} \right], & r < 150 
\end{cases}$$

Here, $r^2 = x^2 + (y - 250)^2$ is given by the vector $a(R, \theta) = (R \cos \theta, -R \sin \theta)$ where $R, \theta$ are the polar coordinates indicated in Figure 14.

This problem has been solved by many authors and it is considered as a benchmark problem for algorithms for advection problems ([15], [11]). Here we show some results obtained with an adaptive SUPG algorithm [11]. Figure 15 shows some "fully updated" meshes which are obtained with the scheme outlined in the beginning of this section. For more details the reader should refer to [11].

5.2 A Problem of Supersonic Rotor-Stator Interaction

We applied the "two-pass" adaptive algorithm to a problem of supersonic rotor-stator interaction. We consider now two rows of doubly-parabolic airfoils with thickness to length ratio equal to 0.08. Figure 16 shows some of these airfoils and the initial finite element discretization of the domain. We assume that the stator and the rotor have the same number of airfoils and we perform the computation on domains corresponding in one rotor and one stator airfoil while the presence of the remaining airfoils is simulated by periodic boundary conditions. In the figures the domain of the rotor airfoil is drawn twice.

In the Figs. 17 through 25 we give the results of a supersonic calculation obtained with a dynamically adapted grid. The distance between consecutive airfoils of the rotor (and stator) is assumed equal to twice the airfoil length while the distance between the tail of the stator and the front tip of the rotor airfoil is taken equal to 0.2 of the airfoil length. We impose boundary conditions of supersonic inflow on the left boundary of the stator with the dependent variables equal to

$$\rho = 1.4, \rho u = 4.2, \rho v = 0, \rho e = 8.8.$$
The inflow boundary conditions correspond to a free stream Mach number equal to three. Boundary conditions of supersonic outflow were assumed on the right boundary of the domain of the rotor. The steady-state solution which is obtained by keeping the airfoil fixed was used as an initial condition.

We have chosen $\beta = 0.19$, $\alpha = 0.06$ and we defined the group error indicator to be equal with the maximum element error indicator of the elements in the group. We did not specify $N$ but instead we revised the mesh every time the fifth nodes [23,24] of the rotor mesh coincided with corner nodes of the stator mesh (this resulted in mesh revisions every 10-12 time steps). We also note that all interface elements have been refined beforehand with the maximum level of refinement to facilitate the application of the sliding interface algorithm.

In order to capture shocks of variable strength we used the "normalized" error indicators given in (4.13). It becomes clear from the numerical results that variable shocks are captured well and the mesh evolves dynamically to adapt to the solution of the rotor-stator problem.

Results are shown in Figures 17 - 25. The initial mesh is that shown in Fig. 16. The first adaptive calculation for a steady-state initial condition is shown in Fig. 17a. The corresponding computed pressure profiles are shown in Fig. 17b. Note the symmetry of the shock lines, the continuous pressure fields across the mesh interface, and the fact that both unrefinement and refinement of the mesh were required to achieve the accuracy limits specified. The rotor blades are then allowed to move with unit speed and the mesh is dynamically refined. Plots are shown of calculated adaptive meshes and pressure profiles for 1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8 and 1 cycle (period) of the motion, during which a rotor blade makes a complete revolution from its initial position in Fig. 17 back to the same position.

Several features of the computed meshes and solutions are noteworthy. In the initial steady-state case, only 16 unrefined elements appear. The size of these large elements, indicating small local error, is limited in the present calculations by the distance from the tips of the rotor and stator blades: two elements in the present case since there must exist a sliding interface between them. A minor program modification could allow much larger elements in regions of small error. For the transient case, the number of larger elements (indicating substantial unrefinement) increases, and these regions of low error migrate over the mesh as solution evolves in time. Conversely, substantial refinement of the mesh is indicated at the interface and along shock lines. The method successfully captures shock interactions and the increasing density of pressure profiles downstream from the moving blades. The ratio of the number of elements in the adaptive mesh to that in the uniform fine mesh varies in time, but is typically 4,000 / 12,500, a reduction of 68 percent! The initial coarse mesh of Fig.16 contains around 4000 cells and is incapable of delivering the required accuracy, a fact not easily realized without an expensive computation.

6. Future Directions

We believe adaptive finite element methods will have a significant impact on computational fluid dynamics and computational structural mechanics in the future. These techniques, together with the modern parallel and array processors, will make obsolete many of the more popular methods in numerical analysis in use today. In particular, use of the body-fitted coordinate techniques, splitting methods such as ADI, etc. will probably lose some of their popularity since they are not well suited for problems with unstructured meshes.

It is likely that large gains are to be made in three-dimensional problems. Here more than anywhere else, one needs to do computations on a near optimal mesh where only a minimum number of degrees-of-freedom is required to produce a given level of accuracy.

It is likely that new advances in parallel and array processing will bring the p-methods and h-p methods to the forefront, since, at least from a theoretical point of view, array processors may have
Figure 9. Reflecting shock problem. Adaptive grid and density contours obtained with the time-accurate scheme.
Figure 10. Reflecting shock problem.
Adaptive grid and density contours obtained with the asynchronous time-stepping scheme.
Figure 11. NACA 0012 airfoil in supersonic wind tunnel.
(a) Steady-state density contours obtained with the time-accurate scheme.
(b) Steady-state density contours obtained with the asynchronous time-stepping scheme.
Figure 12. Elevon Cove Problem.
Adaptive Finite Element Grid.
Figure 13. Elevon Cove Problem.
Density contours of the computed solution.