A Comparison of Adaptive Refinement Techniques for Elliptic Problems

WILLIAM F. MITCHELL
University of Illinois, Urbana

Adaptive refinement has proved to be a useful tool for reducing the size of the linear system of equations obtained by discretizing partial differential equations. We consider techniques for the adaptive refinement of triangulations used with the finite element method with piecewise linear functions. Several such techniques that differ mainly in the method for dividing triangles and the method for indicating which triangles have the largest error have been developed. We describe four methods for dividing triangles and eight methods for indicating errors. Angle bounds for the triangle division methods are compared. All combinations of triangle divisions and error indicators are compared in a numerical experiment using a population of eight test problems with a variety of difficulties (peaks, boundary layers, singularities, etc.). The comparison is based on the L-infinity norm of the error versus the number of vertices. It is found that all of the methods produce asymptotically optimal grids and that the number of vertices needed for a given error rarely differs by more than a factor of two.

Categories and Subject Descriptors: G.1.0 **[Numerical Analysis]**: General—*numerical algorithms*; G.1.8 **[Numerical Analysis]**: Partial Differential Equations—*elliptic equations*; *finite element methods*; G.4 **[Mathematics of Computing]**: Mathematical Software

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Adaptive refinement, error estimate, triangle division

1. INTRODUCTION

The adaptive refinement of discretization grids has proved to be a very successful tool for decreasing the size of the linear systems that arise when solving partial differential equations. Over the years, several methods of adaptive refinement have been developed. Each of these methods has been fully described and analyzed in other papers [2–7, 9, 12–15, 17–20]. Additionally, the author usually presents some numerical evidence that indicates the success of the method. However, the different approaches have never been compared with each other. In this paper we briefly describe some of the more popular approaches for the refinement of triangulations for piecewise linear functions used with the finite element method. The methods are then compared in an extensive numerical experiment to determine which of the methods requires the fewest nodes for a given accuracy.

This work was supported in part by the U.S. Department of Energy grant DOE DEF G02-87ER25026.
Author's current address: GE Advanced Technology Laboratories, Bldg. 145-2 Moorestown Corporate Center, Moorestown, NJ 08057.

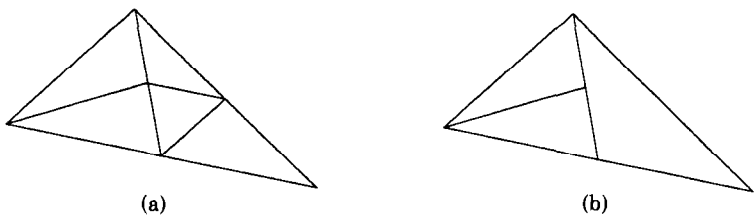


Fig. 1. Examples of (a) compatible triangulation and (b) incompatible triangulation.

We consider the elliptic problem

$$\begin{aligned} Lu &= (p(x, y)u_x)_x + (q(x, y)u_y)_y + r(x, y)u = f(x, y) && \text{in } \Omega \\ u &= g(x, y) && \text{on } \partial\Omega \end{aligned}$$

where Ω is a polygonal domain in $\mathbf{R}^2$.

We are particularly interested in situations in which there is a singularity, rapid variation in the solution, or some other property where adaptive refinement may increase the accuracy of the approximate solution. We assume that one wishes to solve the equation using the finite element method with piecewise linear polynomials over triangles.

To obtain an adaptively refined grid for such a problem, we begin with an initial triangulation, T_0, of the domain Ω. $T_0 = \{t_1, t_2, \ldots, t_{N_0}\}$ where t_i is a triangle, $\cup\, t_i = \Omega$, and, for $i \neq j$, $t_i \cap t_j$ is either empty, a common side of t_i and t_j, or a common vertex of t_i and t_j. Such a triangulation is said to be *compatible* or *conforming*. Figure 1 illustrates both a compatible and an incompatible triangulation. In the case of Figure 1(b), Rivara [12] calls the large triangle a $\frac{1}{2}$-*nonconforming triangle*, the edge a $\frac{1}{2}$-*nonconforming side*, and the midpoint a $\frac{1}{2}$-*nonconforming point*.

Given such an initial triangulation, T_0, we generate a sequence of nested compatible triangulations, $T_k = \{t_{1,k}, \ldots, t_{N_k,k}\}$. When the meaning is clear, we omit the second subscript. To get T_{k+1} from T_k, the error on each triangle $t_{i,k}$ is estimated and used to determine a set of triangles, S_k, that are to be refined. The selection of triangles for S_k is such that T_{k+1} is finer than T_k only in those regions where the refinement makes a significant reduction in the error. We refine the triangles in S_k and also those necessary to ensure that T_{k+1} is compatible. The triangles in T_0 are said to have *generation* 0. Triangles formed by dividing a triangle of generation i have *generation* $i + 1$ and are called the *children* of the triangle divided.

It is important that the triangles be refined in such a way that the angles remain bounded away from 0 and π. It is shown by Babuška and Aziz [1] that as the maximum angle approaches π, the interpolation error grows. Moreover, they show that the interpolation error does not grow if a triangle has only one arbitrarily small angle. However, Fried [8] shows that the condition number of the stiffness matrix grows like $1/\sin(\theta)$ where θ is the minimum angle, so it is important to bound the angles away from 0 as well.

Given an initial triangulation T_0, we use the following algorithm for solving the partial differential equation with adaptive refinement:

```
k ← 0
solve the PDE on triangulation T_0
estimate errors
while the maximum error estimate is bigger than a given error tolerance τ do
  determine a set of triangles to divide, S_k ⊆ T_k
  divide triangles in S_k and those necessary for compatibility to get T_{k+1}
  solve the PDE on triangulation T_{k+1}
  estimate errors
  k ← k + 1
endwhile
```

There are four main tasks to be performed here:

(1) solving the PDE
(2) determining a set of triangles to divide
(3) dividing triangles and maintaining compatibility
(4) estimating errors

As stated earlier, the PDE is solved using the usual finite element method with piecewise linear basis functions. The set of triangles to be divided, S_k, is determined by the method of Babuška and Rheinboldt [2] which has been widely used in adaptive refinement algorithms. In this method one approximates, using local convergence properties, what the largest error estimate would be after a single refinement of each triangle and refines those triangles whose error estimate is larger than that value. The purpose of this paper is to compare various methods for the other two tasks, dividing triangles and estimating errors.

The remainder of the paper is organized as follows. In Section 2 we present four methods for dividing triangles and maintaining compatibility. Minimum and maximum angle bounds are given for each method. Some other properties and relationships of the methods are also examined, including an extensive comparison of angle bounds. In Section 3 we present eight methods for estimating errors. The methods considered fall into two classes: those based on a bound on the error in some norm and those obtained by solving an auxiliary problem. In Section 4 a numerical experiment is used to compare the methods. A population of eight problems is solved by each method. These problems represent several forms of difficulty for which adaptive refinement is appropriate. Every effort is made to avoid a bias toward any method. In Section 5 we present our conclusions based on the numerical experiment and other considerations.

2. METHODS FOR DIVIDING TRIANGLES

There are two basic methods for dividing triangles used in practice: *regular division* and *bisection*. Both of these exist in commonly used software. PLTMG [3, 5] uses regular division and 2DEPEP [18, 19] uses bisection. The bisection method also has several variations other than that used in 2DEPEP.

To divide a triangle by regular division, one connects the midpoints of the sides of the triangle to obtain four triangles similar to the original, as in Figure 2(a). When only some of the triangles are divided by regular division,

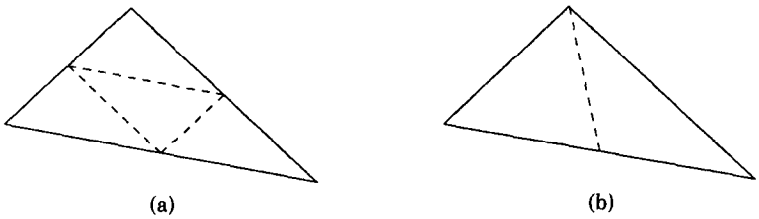


Fig. 2. Division of a triangle by (a) regular division and (b) bisection.

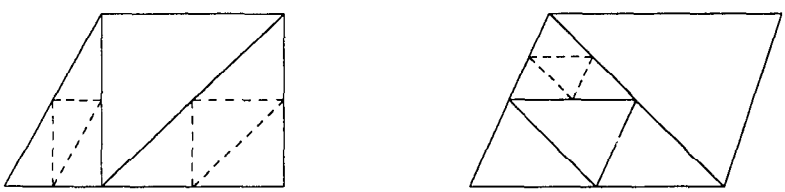


Fig. 3. Incompatibilities resulting from regular division.

incompatibilities surely arise. Bank and Sherman [5] developed a way of "cleaning up" the triangulation to remove these incompatibilities. First, the triangles in S_k are divided by regular division. Then, any triangles that have two $\frac{1}{2}$-nonconforming sides or two nonconformities on the same side, as in Figure 3, are refined by regular division. If necessary, this process is repeated until every triangle is either compatible or has one $\frac{1}{2}$-nonconforming side. Then, the triangles that have a $\frac{1}{2}$-nonconforming side are divided by a "green" division, which connects the opposite vertex to the $\frac{1}{2}$-nonconforming point as in Figure 2(b). This creates a compatible triangulation. The "green" refinements are removed before the next refinement. Since every triangle in T_{k+1} is either similar to or a bisection of some triangle in T_0, the angles are clearly bounded away from 0 and π. Since the "green" refinements are removed before continuing to the next triangulation, the family of triangulations created by this algorithm is not nested; however, the sets of vertices are nested.

The second approach to dividing triangles is bisection. Here a triangle is divided by connecting a vertex to the midpoint of the opposite side, as in the "green" divisions used for "clean up" in the regular division method. Several variations of bisection method result according to which vertex or edge is to be divided.

One choice is to divide the longest edge. This choice is intuitively reasonable since it divides the largest angle, which should prevent the angles from getting too small. Indeed, it has been proven by Rosenberg and Stenger [16] that by always bisecting the longest edge, the smallest angle possible is half of the smallest angle in T_0.

After dividing all the triangles in S_k, Rivara [13] maintains compatibility by a longest edge bisection of all the triangles with a $\frac{1}{2}$-nonconforming side. This may create new incompatibilities or may not remove all the old ones (see Figure 4).

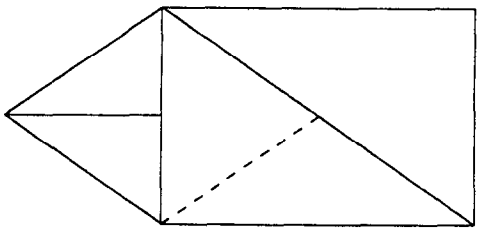


Fig. 4. Failure of a single longest edge bisection to fix incompatibilities.

Fig. 5. Propagation of the peak with newest node bisection.

The process is repeated until there are no incompatibilities. Rivara has shown that this terminates after a finite number of steps.

A second possible choice is to divide the newest node. In this choice, an angle is never divided twice, which intuitively should prevent the angles from getting too small. The newest node is called the *peak* and the opposite side, the *base* of a triangle. One begins by selecting a peak for every triangle. After dividing a triangle by connecting the peak to the midpoint of the base, the new node created is assigned as the peak of each of the two children triangles, as in Figure 5. Sewell [17] showed that all the descendants of an original triangle fall into four similarity classes (see Figure 6) and hence the angles are bounded away from 0 and π.

Sewell [17] enforces compatibility by only dividing triangles that are compatibly divisible. (*Note*: This is not the approach used in PDE/PROTRAN [19].) A triangle is called *compatibly divisible* if its base is also the base of the triangle that shares that edge. Thus both triangles can be divided as a pair. One might expect that any triangle eventually becomes compatibly divisible, so that any triangle can eventually be divided. However, this need not be the case, as the following example shows:

Consider approximating, via interpolation, a function on the triangle of Figure 7(a) with the initial triangulation and peak selection shown. Suppose the function to be approximated is continuous, 0 on t_1, nonzero on the interior of t_2, and has some property that requires a finer grid near the top of t_2. Since the approximation is exact on t_1, the error there is 0 and t_1 will never be in S_k. Thus the first refinement divides t_2 as in Figure 7(b). Now, the behavior of the solution dictates that $t_{2,1}$ be refined. But $t_{2,1}$ can never be compatibly divisible because t_1 can never be divided. In fact, for any refinement the only compatibly divisible triangle is the lower right triangle, and the refinement is as in Figure 7(c). Note that this triangulation has been refined in the wrong place.

To avoid this problem, we have developed a new method that enforces compatibility by recursively dividing neighboring triangles. If a triangle is not compatibly divisible, then after a single division of the neighbor opposite the

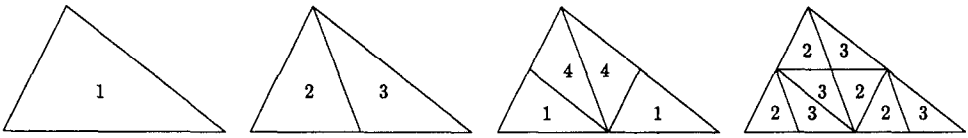


Fig. 6. Four similarity classes of triangles generated by the newest node bisection.

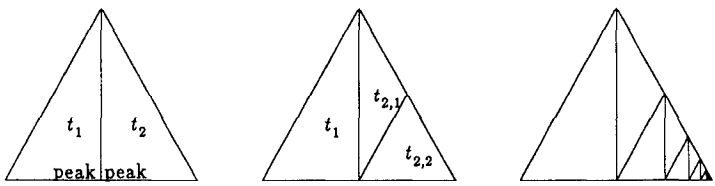


Fig. 7. Example of refinement in the wrong place.

peak, it will be. Of course, it may be possible that the neighboring triangle is not compatibly divisible, so we recursively refine the neighboring triangle by newest node bisection until a compatibly divisible triangle is found. The recursion occurs before the division, so we always divide pairs of triangles (except near the boundary). Here there is no "clean up" process after dividing the triangles in S_k. Instead, compatibility is maintained during the division process.

It is easily shown that this recursion is finite if the peaks of T_0 are chosen such that every triangle is compatibly divisible. In this case, each triangle in the recursion is of one generation less than the triangle before it. Thus the length of the recursion is bounded by the generation number of the starting triangle. One can also show that given any compatible initial triangulation T_0, there exists a choice of peaks such that every triangle is compatibly divisible, so this assumption is reasonable.

Rivara [13] has proposed a bisection algorithm that is a hybrid of the longest edge and newest node approaches. In this method the triangles in S_k are bisected by the longest edge, and compatibility is enforced by repeatedly bisecting the $\frac{1}{2}$-nonconforming triangles, but the $\frac{1}{2}$-nonconforming triangles are not always divided by the longest edge. If the triangle is in T_k, it is divided by the longest edge. If it is a descendant of a triangle in T_k, it is divided by the newest node. In general, this generates fewer triangles during the "clean up" process than the longest edge. Rivara has shown that the bound on the angles is the same as that achieved by always bisecting the longest edge and that the process is finite.

In many cases the three bisection methods are the same. This happens whenever the longest edge is always opposite the newest node. To be specific, we refer to Figure 8 and assume that $\alpha \leq \beta \leq \gamma$. It is easily shown that the three bisection methods are the same if and only if

(1) γ is the peak for newest node bisection
(2) $\epsilon_2 \geq \beta$
(3) $\epsilon_2 \geq \delta_2$, and
(4) $\xi \geq \delta_2$

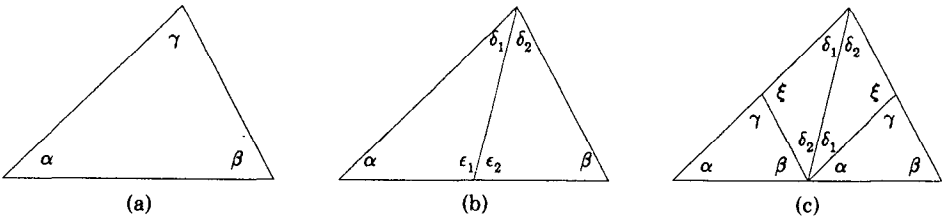


Fig. 8. Angles that arise during bisection.

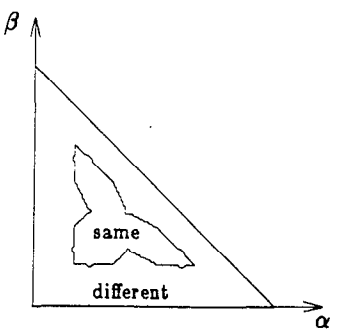


Fig. 9. Region in which the three bisection methods are the same.

The equivalent expression in terms of α and β are extremely complicated. We use Figure 9 to illustrate, in terms of α and β, when the three methods are the same, assuming that the largest angle is chosen as the initial peak for the newest node bisection.

A common example of the methods coinciding occurs when the triangulation consists entirely of isosceles right triangles and the initial peaks are chosen to be at the right angles. Then the peaks are always at the right angles and the hypotenuse (the longest edge) is opposite the peak.

Such a triangulation turns out to be optimal, in terms of angle conditions, for bisection methods. If we consider the bisection of an arbitrary triangle, as in Figure 8(b), one of the angles ϵ_1 or ϵ_2 must be at least $\pi/2$. Also, one of the angles α, β, δ_1, or δ_2 must be less than or equal to $\pi/4$. Thus, when using a bisection method, the smallest maximum angle bound possible is $\pi/2$ and the largest minimum angle bound possible is $\pi/4$. The above triangulation satisfies both of these bounds since any refinement of it consists only of isosceles right triangles and hence is optimal.

The optimal type of triangulation for regular refinement is one consisting entirely of equilateral triangles. A triangle obtained by regular division is similar to a triangle in the initial grid. When these are equilateral triangles all the angles are equal to $\pi/3$, which is the furthest away from 0 and π as possible. A triangle obtained by a "green" division can have any of the initial angles divided. Thus the smallest initial angle should be as large as possible and that is obtained by equilateral triangles. The minimum and maximum angles are then $\pi/6$ and $\pi/2$, respectively.

For newest node bisection and regular refinement we can determine all the angles that might occur in the children of an initial triangle with angles α, β, and γ, where γ is the initial peak for newest node bisection. Figure 8 illustrates the possible angles. It can be shown that the size of these angles is

$$
\begin{aligned}
&\alpha \text{ given}\\
&\beta \text{ given}\\
&\gamma \text{ given}\\
&\epsilon_1 = \tan^{-1}[2 \sin\alpha \sin\beta/\sin(\alpha - \beta)]\\
&\epsilon_2 = \pi - \epsilon_1\\
&\delta_1 = \epsilon_2 - \alpha\\
&\delta_2 = \epsilon_1 - \beta\\
&\xi = \pi - \gamma
\end{aligned}
$$

It is easily seen that the smallest of these angles is either α, β, δ_1, or δ_2 and the largest is either γ, ξ, ϵ_1, or ϵ_2. For regular refinement, the angle ξ does not arise since "green" refinements are removed before continuing the refinement process. But any of the three angles may be the angle divided. Thus there are three versions of the ϵ's and δ's. Let the superscript α, β, or γ denote which angle was divided, for example, $\epsilon_1^\gamma = \epsilon_1$, and ϵ_1^α is the equivalent of ϵ_1 when α is the angle divided. We are now in a position to give explicit bounds on the minimum and maximum angles. Let θ_{bisect}, Θ_{bisect}, θ_{reg}, and Θ_{reg} be the minimum and maximum angle bounds for the newest node bisection and regular refinement, respectively. Then we have

$$
\begin{aligned}
\theta_{\text{bisect}} &= \min\{\alpha, \beta, \delta_1^\gamma, \delta_2^\gamma\}\\
\Theta_{\text{bisect}} &= \max\{\gamma, \xi, \epsilon_1^\gamma, \epsilon_2^\gamma\}\\
\theta_{\text{reg}} &= \min\{\alpha, \beta, \delta_1^\alpha, \delta_2^\alpha, \delta_1^\beta, \delta_2^\beta, \delta_1^\gamma, \delta_2^\gamma\}\\
\Theta_{\text{reg}} &= \max\{\gamma, \epsilon_1^\alpha, \epsilon_2^\alpha, \epsilon_1^\beta, \epsilon_2^\beta, \epsilon_1^\gamma, \epsilon_2^\gamma\}
\end{aligned}
$$

We note that $\theta_{\text{bisect}} \geq \theta_{\text{reg}}$ for any triangle. There is no analogous statement for Θ, that is, the method with the smaller maximum bound depends on the shape of the initial triangle. Figure 10 illustrates, in terms of α and β, for which triangle shapes the smaller maximum bound is achieved for each method, assuming that the largest angle is the initial peak for newest node bisection.

For longest edge and hybrid bisection we cannot give such explicit bounds on the angles. This is because we cannot, for an arbitrary triangle, determine the sequence of divisions that would occur. For a specific triangle this can be done for the longest edge bisection, but it is still impossible for hybrid bisection. So the best we can do is use the earlier stated bound on the minimum angle of $\alpha/2$ where α is the smallest angle of the initial triangle. It immediately follows that the maximum angle is bounded by $\pi-\alpha$. In general, both of these bounds are tight since we have equality for an equilateral triangle where the minimum and maximum angles achieved are $\pi/6$ and $2\pi/3$, respectively. However, for specific cases the bounds may be pessimistic, as in the case of isosceles right triangles.

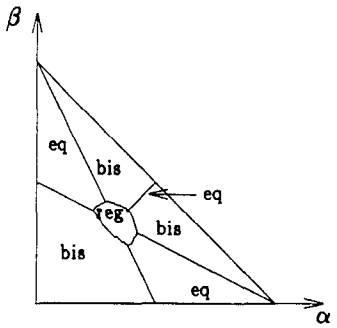


Fig. 10. Regions in which (bis) bisection has the smaller maximum angle, (reg) regular division has the smaller maximum angle, and (eq) maximum angle bounds are equal.

3. ERROR INDICATORS

In this section we present eight methods for estimating the error on each triangle, or rather, indicating which triangles have large errors. The distinction between an error estimate and an error indicator is made by Zienkiewicz et al. [20]. They outline a set of requirements that a practical *error estimate* should satisfy. The most important of these are that the estimate should be determined *a posteriori* from local information and that the effectivity index (the ratio of the error estimate to the exact norm of the error) should be greater than 1, but not too large, and asymptotically approach 1. Clearly, a computed value need not satisfy the conditions on the effectivity index to be a good indication of which triangles should be divided. A value that indicates *which* triangles have the largest error (without necessarily telling *what* the error is) is referred to as an *error indicator.* In this study, we are not concerned with having a good estimate of the error, only in having an error indicator that produces an optimal finite element mesh.

The error indicators presented here fall into two classes: those based on a bound on the error in some norm and those obtained by solving an auxiliary problem. We consider first some indicators based on error bounds.

Sewell [19] used a bound on the L_∞ norm of the interpolation error to suggest that for a near optimal grid

$$\int_{t_i} D^2_{\max} u$$

should be approximately equal for all triangles t_i, where

$$D^2_{\max} u(x, y) = \max\{|u_{xx}(x, y)|, |u_{xy}(x, y)|, |u_{yy}(x, y)|\}$$

This is an error indicator if we can compute an approximation to $D^2_{\max} u$. The program 2DEPEP [18] requires that the user provide a function to approximate $D^2_{\max} u$. (Actually, 2DEPEP needs $D^3_{\max} u$ since it uses piecewise quadratics.) For our test problems, where we know the true solution, we use the exact $D^2_{\max} u$. This, of course, could not be done when solving practical problems but gives a best case for this type of error indicator for comparison with other error indicators.

Sewell does provide a method for approximating $D^2_{\max} u$ from the approximate solution u_T [17]. Since u_T is piecewise linear, $D^2_{\max} u_T \equiv 0$ on each triangle. But,

$D_x u_T$ and $D_y u_T$ are both constant on each triangle. We define a continuous piecewise linear function $ED_x u_T$ on T by assigning the nodal values to be the average of $D_x u_T$ in each triangle around the node. Similarly, we define $ED_y u_T$. Then we approximate $D^2_{\max} u$ by

$$\max\{D_x ED_x u_T,\ D_y ED_y u_T,\ \tfrac{1}{2}(D_x ED_y u_T + D_y ED_x u_T)\}$$

Another norm-based error indicator is an estimate of a bound on the energy norm of the error, where the energy norm is defined by

$$\| u \|^2 = \int p u_x^2 + q u_y^2 + r u^2$$

and p, q, and r are the coefficient functions in the differential operator L. This error estimate is used by Bank and Sherman in the July 1979 version of the program PLTMG [4]. For $p = q$ in the differential operator L, the estimate is

$$C_1(p, t_i) h_{t_i}^2 \int_{t_i} (Lu_T - f)^2 + C_2(p, t_i) h_{t_i} \int_{\partial t_i} J^2$$

where J is the jump in the normal derivative in u_T across ∂t_i, h_{t_i} is the diameter of t_i, and C_1 and C_2 are constants that depend on the coefficient function p and the geometry of t_i but are independent of the size of t_i. We use the C_1 and C_2 that were used in PLTMG. A nearly identical error estimate for bilinear rectangular elements is derived in [20].

Babuška and Rheinboldt [2] introduced the idea of computing error indicators by solving an auxiliary problem. The auxiliary problem takes the form of approximately solving a differential equation to obtain a solution that is locally more accurate (or equivalently an equation whose solution approximates the error). The domain for the auxiliary problem is the union of a small number of triangles from T_k so that the problem is computationally much easier to solve than the original PDE. The boundary conditions may be Dirichlet, Neumann, or mixed. To increase the order of accuracy, one may either use higher order polynomials on the triangles of T_k or use the same order polynomial and refine the triangles of T_k.

Bank and Weiser [6] developed three error estimates of this type. They use Neumann boundary conditions to define an auxiliary problem on t_i. The accuracy of the solution is improved by using quadratic basis functions. We consider here only the third estimate, which clearly outperformed the others in the numerical results presented in [6]. This is the error indicator used in Version 4.0 of PLTMG [3]. Let $\partial/\partial n$ denote the normal derivative, and $[v]_J$ denote the magnitude of a jump discontinuity of v, and let the functions p and q of the differential operator be equal. Let $\hat{e}$ be the approximate solution to

$$Le = f - Lu_T \qquad \text{in} \quad t_i$$

$$p \frac{\partial e}{\partial n} = \frac{1}{2}\left[p \frac{\partial u_T}{\partial n} \right]_J \qquad \text{on} \quad \partial t_i$$

obtained by the usual finite element method using a three-function quadratic basis, each basis function having the value 1 at the midpoint of one side and 0 at

the midpoints of the other sides and vertices. The boundary conditions are modified slightly for triangles that have a side on $\partial\Omega$. The error estimate is the energy norm of $\hat{e}$.

The other auxiliary problem error indicators considered use Dirichlet boundary conditions to define the problem and a refinement of the triangles to increase the accuracy of the solution. We have not seen these error indicators used before. We consider these methods only for bisection divisions. It becomes clear that these methods could also be defined for regular divisions, but it would be more complicated.

Let t_1, with vertices v_1, v_2, and v_3, be the triangle on which the error is to be estimated, and let t_2 with vertices v_1, v_2, and v_4 be the neighboring triangle opposite the peak v_3 as in Figure 11(a). For now, suppose that such a triangle t_2 exists (i.e., the base of t_1 is not part of $\partial\Omega$) and that the peak of t_2 is v_4 (that is, that t_1 is compatibly divisible). We solve the auxiliary problem on the domain $t_1 \cup t_2$ with the same grid that we would have if we were to divide the triangle pair t_1, t_2 by bisection. Specifically, let $\hat{v}$ be the approximate solution to

$$\begin{aligned} Lv &= f \qquad \text{in} \quad t_1 \cup t_2 \\ v &= u_T \qquad \text{on} \quad \partial(t_1 \cup t_2) \end{aligned}$$

obtained by applying the usual finite element method with piecewise linear elements using a grid with four triangles as shown in Figure 11(a). Then $\hat{v}$ should be a better approximation to the solution than u_T, and we approximate the error by $\| \hat{v} - u_T \|$. Here the norm is computed only over $t_1 \cup t_2$, and one can use either the L_∞ norm or the energy norm. We refer to these as 1-*point local Dirichlet* methods. Note the L_∞ norm is just the value at the middle node and the energy norm is that value times the energy norm of the basis function at that node. This is actually an error indicator for the *pair* of triangles.

If the triangle t_2 does not exist (the base of t_1 is on the boundary), then the new node is on the boundary. In this case, we use the boundary conditions to define $\hat{v}$.

If t_1 is not compatibly divisible, we introduce a fictitious node, $\hat{v}_4$, on the base of t_2, as in Figure 11(b). The value of u_T at $\hat{v}_4$ is the average of the values at v_1 and v_4. $\hat{v}$ is then defined as above using t_1 and the adjacent child of t_2 as the domain.

The energy norm version of this indicator is, with appropriate modifications, the same as that proposed by Zienkiewicz et al. [20] for bilinear rectangular elements.

It may be the case that this does not provide a sufficiently accurate approximation to the error. In computing this estimate, we assume that the values at the neighboring four nodes of the new node remain constant. In fact, the values at these, and every, node change with the addition of the new node, but the amount of change that occurs at a node diminishes as the distance from the new node increases. A better approximation may be obtained by allowing the four neighboring nodes to vary also. Then we are solving the auxiliary Dirichlet problem on the domain and triangulation of Figure 12. This involves the solution of a 5×5 symmetric positive definite system. Again, the error can be measured

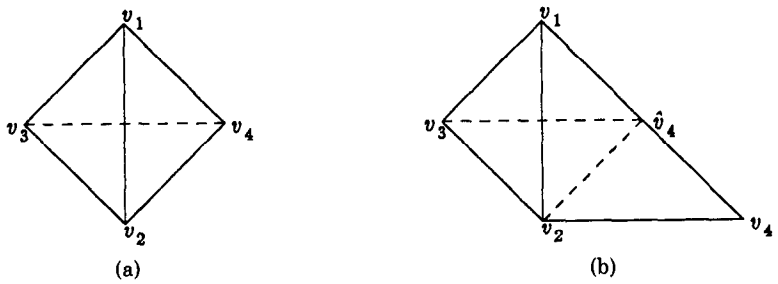


Fig. 11. Domain for 1-point local Dirichlet problem. (a) v_4 is the peak of t_2; (b) v_4 is not the peak of t_2.

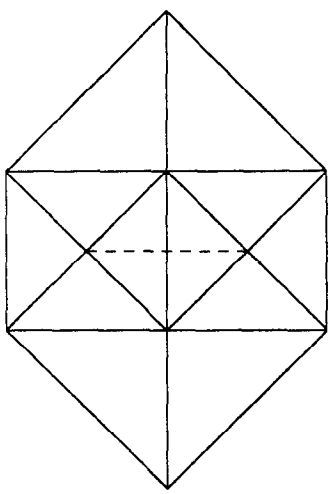


Fig. 12. Domain for 5-point local Dirichlet problem.

in either the L_∞ or energy norm. We refer to these methods as 5-*point local Dirichlet* methods.

The local Dirichlet methods have an interpretation other than that of using a more accurate approximation to estimate the error. Since the grid used for the solution of the auxiliary problem is the grid that would be obtained if the triangle t_i were to be divided, the value of the "error" at the new node is actually an approximation of how much the solution changes if we were to divide t_i. Thus, when we divide the triangles with the largest error indicators, we are dividing the triangles that make the most change in the solution. Presumably, this change is toward the true solution of the differential equation, so we are dividing the triangles which reduce the error the most. We also remark that the inner products computed for the local Dirichlet indicators are precisely those needed for the global stiffness matrix. By saving these values, the matrix can be obtained without computing any further inner products.

4. NUMERICAL COMPARISON OF ADAPTIVE REFINEMENT METHODS

In this section we present a numerical comparison of the methods of Sections 2 and 3. We consider all combinations of the error indicators with the triangle

divisions except for the local Dirichlet error indicators with regular refinement. Although we could have included the excepted case, it is much more difficult to implement the local Dirichlet error indicator for regular refinement owing to the fact that the division of a triangle adds three new vertices rather than one. Additionally, we consider, for comparison, the use of the true error instead of an error indicator and the use of a uniform grid instead of adaptive refinement. We refer to the methods by the following names:

Dividing Triangles:

(1) regular: regular division with "green" patches,
(2) longest edge bisection: bisection of longest edge,
(3) newest node bisection: bisection at newest node with recursion for compatibility,
(4) hybrid bisection: bisection with hybrid of longest edge and newest node.

In cases where the three bisection methods coincide we refer to the method simply as bisection.

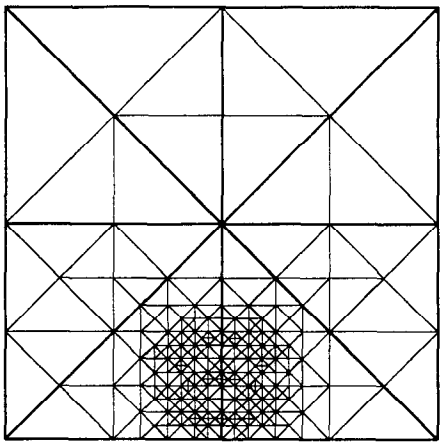
Error Indicators:

(1) Sewell-exact: $\int D^2_{\max} u$ using the exact $D^2_{\max} u$,
(2) Sewell-approximate: $\int D^2_{\max} u$ using an estimate of $D^2_{\max} u$ from the approximate solution,
(3) Bank–Sherman: estimate of energy norm,
(4) Bank–Weiser: energy norm of solution to a Neumann auxiliary problem,
(5) LD 1 pt maximum: L_∞ norm of solution to 1-point local Dirichlet problem,
(6) LD 1 pt energy: energy norm of solution to 1-point local Dirichlet problem,
(7) LD 5 pt maximum: L_∞ norm of solution to 5-point local Dirichlet problem,
(8) LD 5 pt energy: energy norm of solution to 5-point local Dirichlet problem,
(9) True-maximum: L_∞ norm of the true error,
(10) True-energy: energy norm of the true error,
(11) Uniform: uniform grid, no error indicator needed.

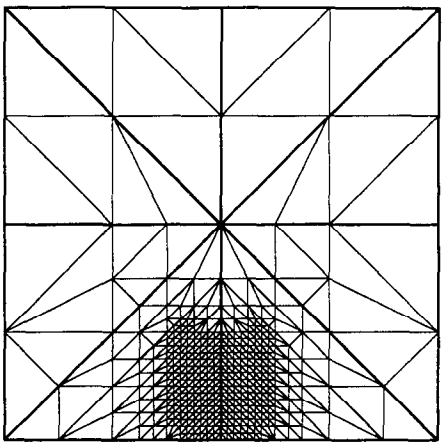
Using the algorithm of Section 1, each of the test problems is solved by each method using several values for the termination error tolerance τ. These computations were performed on a Pyramid 90x with floating point accelerator operating under the Pyramid Technology OSx 3.1 Operating System. This is a dual port of System V Release 2.0 of AT&T Bell Laboratories and the 4.2BSD of the University of California at Berkeley. The Pyramid Technology Optimizing FORTRAN 77 compiler which has about 7 decimal digits, was used with single precision.

The object of adaptive refinement is to limit the size of the linear system while producing a small discretization error. Thus the criteria by which we compare the methods is the number of vertices versus L_∞ norm of the error. The L_∞ norm is approximated by measuring the error at the vertices, the centers of the triangles, and the three points halfway between the center and the vertices.

The complete results of the experiment are presented in [10]. (*Note*: In [10] we used an L_1 norm-based error indicator of Sewell [17, 18] that performed



Problem # 1a
Refinement: Bisection
Error indicator: LD 5pt max
408 triangles 215 vertices 10 levels



Problem # 1a
Refinement: Regular
Error indicator: Bank–Weiser*
736 triangles 387 vertices 5 levels

Fig. 13. Sample triangulation plots.

poorly in the L_∞ norm measurements. When considering the "Sewell" error indicators, one should use the results in this paper rather than those of [10].) In Appendix 1 of [10] we present the population of eight test problems used as a basis for comparing the methods. The solutions of the equations exhibit various difficulties that adaptive refinement is suited to deal with: peaks, boundary layers, singularities, and so forth. Two of the problems are solved on two domains so as to give both bisection and regular division their respective optimal shape of triangles. In Appendix 2 of [10] we present sample triangulations using the best error indicator for each triangle division method. The initial triangulation is indicated by dark lines. For the first problem, we also present triangulations from some other error indicators to demonstrate the effect of the error indicator on the triangulation obtained. In Appendix 3 of [10] we use log-log plots to present the error as a function of the number of vertices. In Appendix 4 of [10] the methods are ranked for each problem. For "nice" problems, the L_∞ norm of the error is $O(N^{-1})$ for a linear approximation, where N is the number of vertices. One would expect that, for problems that are nearly singular, adaptive refinement would preserve this order of convergence while uniform refinement may need a very fine grid before the proper order of convergence is observed. Thus, the data of each method is fit with a least squares approximation of the form error $= c * N^{-1}$. The methods are ranked according to the value of c. Smaller values of c indicate that fewer vertices are required for a given error. We also present the value b from a least squares fit of the form error $= a * N^{-b}$ to demonstrate the actual rate of convergence obtained. The last column gives the ratio c/c_{best} where c_{best} is the value of c for the best method. A value of 2, for example, indicates that this method takes twice as many vertices as the best method to achieve the same error.

We present here a summary of the results from [10]. Appendix A contains the set of eight test problems used. Figure 13 shows sample triangulations for

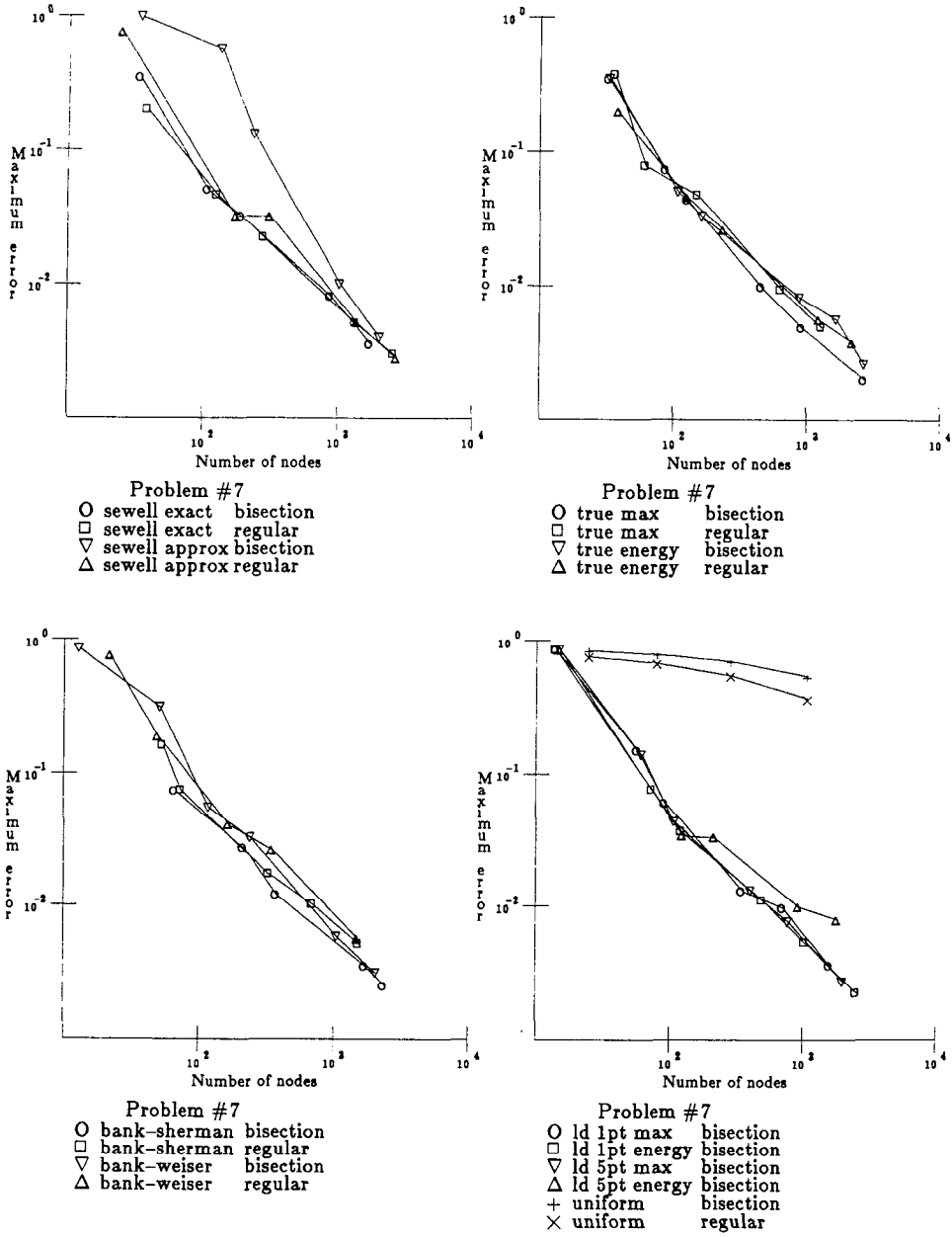


Fig. 14. Results for problem #7.

bisection and regular division. The solution for this problem has a peak centered at (.5, .117). Figure 14 and Table I contain the complete results for newest node bisection and regular division for problem #7. Figure 14 shows the log-log plots of error versus the number of vertices, and Table I contains the information of Appendix 4 of [10]. This is a very difficult problem as indicated by the poor

Table I. Ranking of Methods for Problem #7

Triangle division	Error indicator	c	Rate of convergence	c/c_{best}
Bisection	Bank–Sherman	5.444	0.942	1.00
Bisection	True-maximum	6.062	1.159	1.11
Bisection	LD 1pt energy	6.135	1.106	1.13
Bisection	LD 5pt maximum	6.704	1.156	1.23
Bisection	LD 1pt maximum	6.900	1.139	1.27
Regular	Bank–Sherman	6.919	0.980	1.27
Regular	True-energy	6.998	0.971	1.29
Regular	Sewell-exact	7.001	0.981	1.29
Regular	True-maximum	7.182	1.117	1.32
Bisection	Sewell-exact	7.206	1.095	1.32
Bisection	True-energy	7.501	1.018	1.38
Bisection	Bank–Weiser	8.432	1.146	1.55
Bisection	LD 5pt energy	8.784	0.960	1.61
Regular	Sewell-approximate	9.115	1.166	1.67
Regular	Bank–Weiser	9.642	1.128	1.77
Bisection	Sewell-approximate	23.610	1.449	4.34
Regular	Uniform	90.481	0.190	16.62
Bisection	Uniform	112.444	0.116	20.65

performance of the uniform grids. Finally, Table II contains, for all problems, the c/c_{best} ratio used to rank the methods. One should not take this ranking as a strict ordering of the methods. With a small number of data points and no more than 3000 vertices, the least squares fit with a slope of -1 may not accurately represent the asymptotic behavior. Any methods for which the values of c are close, say within 10 percent, should be considered to be equivalent.

In any numerical experiment of this nature one must be careful not to make conclusions that are too broad. Of necessity, only a small sample of problems can be examined, with the hope that they are representative of some larger class of problems for which the same behavior would be observed. We believe the following observations are likely to hold for a much larger class of problems:

(1) In most cases, the two forms of Sewell's error indicator are fairly close. This indicates that the given approximation of $D^2_{\max} u$ is sufficiently accurate for the purposes of this error indicator.

(2) Longest edge and hybrid bisection are essentially equivalent, rarely differing by more than a few percent. Since longest edge bisection is a simpler algorithm than the hybrid bisection, we see no reason to use hybrid bisection.

(3) When using the true error as an error indicator there is no preference for either the energy or L_∞ norm. This suggests that error indicators may be based on either norm.

(4) The true error does not usually produce the best grid. This indicates that the best error estimates, which are close to the error, might not make the best error indicators. The local Dirichlet indicators, which are close to the amount of change that the refinement would cause, are usually among the best methods.

Table II. c/c_{best} Values

Triangle division	Error indicator	Problem number									
		1a	1b	2	3	4	5	6a	6b	7	8
Newest node bisection	Bank–Weiser	1.11	1.56	1.27	1.24	1.43	1.92	1.11	1.43	2.25	6.61
Newest node bisection	Sewell-exact	1.43	1.96	1.41	1.27	3.70	2.07	1.24	1.98	1.92	4.20
Newest node bisection	Sewell-approximate	1.71	2.16	1.81	1.21	4.57	2.69	1.87	2.42	6.30	5.67
Newest node bisection	True-maximum	1.11	1.85	1.01	1.06	1.55	2.00	1.00	1.41	1.62	2.36
Newest node bisection	True-energy	1.18	1.60	1.19	1.15	1.40	2.12	1.23	1.51	2.00	3.46
Newest node bisection	Bank–Sherman	1.31	1.69	1.38	1.18	1.67	1.41	1.44	1.19	1.45	2.52
Newest node bisection	LD 1pt maximum	1.62	1.12	1.01	1.84	1.47	1.86	1.62	1.42	1.84	2.84
Newest node bisection	LD 1pt energy	1.75	1.14	1.00	1.65	1.27	1.44	1.65	1.39	1.64	2.98
Newest node bisection	LD 5pt maximum	1.00	1.00	1.02	1.22	1.43	1.48	1.34	1.39	1.79	3.04
Newest node bisection	LD 5pt energy	1.08	1.13	2.38	1.00	1.35	1.89	1.40	1.81	2.34	2.50
Newest node bisection	Uniform	8.04	15.96	2.91	1.17	4.66	3.68	12.27	11.88	30.01	3.42
Regular	Bank–Weiser	1.64	1.62	1.63	1.71	1.66	2.12	1.94	1.00	2.57	3.05
Regular	Sewell-exact	1.79	1.43	1.69	2.12	2.31	2.64	1.95	1.42	1.87	1.54
Regular	Sewell-approximate	1.83	2.31	2.53	2.19	3.72	2.33	2.85	2.13	2.43	3.06
Regular	True-maximum	1.62	2.52	1.52	1.53	1.32	3.03	1.99	3.48	1.92	1.00
Regular	True-energy	1.54	1.84	1.55	1.64	1.54	2.05	1.98	1.12	1.87	1.44
Regular	Bank–Sherman	1.78	1.98	1.93	1.77	1.61	2.59	2.28	5.50	1.85	1.44
Regular	Uniform	10.81	9.98	3.61	1.45	4.12	4.35	12.54	10.94	24.15	1.20
Longest edge bisection	Bank–Weiser	1.11	1.56	1.27	1.24	1.13	1.23	1.11	1.43	2.03	6.61
Longest edge bisection	Sewell-exact	1.43	1.96	1.41	1.27	2.80	1.27	1.24	1.98	1.92	4.20
Longest edge bisection	Sewell-approximate	1.71	2.16	1.81	1.21	2.36	1.86	1.87	2.42	5.70	5.67
Longest edge bisection	True-maximum	1.11	1.85	1.01	1.06	1.00	1.78	1.00	1.41	1.38	2.36
Longest edge bisection	True-energy	1.18	1.60	1.19	1.15	1.13	1.16	1.23	1.51	1.65	3.46
Longest edge bisection	Bank–Sherman	1.31	1.69	1.38	1.18	1.52	1.02	1.44	1.19	1.00	2.52
Longest edge bisection	LD 1pt maximum	1.62	1.12	1.01	1.84	1.10	1.08	1.62	1.42	1.91	2.84
Longest edge bisection	LD 1pt energy	1.75	1.14	1.00	1.65	1.11	1.08	1.65	1.39	1.81	2.98
Longest edge bisection	LD 5pt maximum	1.00	1.00	1.02	1.22	1.07	1.02	1.34	1.39	1.61	3.04
Longest edge bisection	LD 5pt energy	1.08	1.13	2.38	1.00	1.20	1.92	1.40	1.81	2.49	2.50
Longest edge bisection	Uniform	8.04	15.96	2.91	1.17	4.00	3.30	12.27	11.88	48.62	3.42
Hybrid bisection	Bank–Weiser	1.11	1.56	1.27	1.24	1.11	1.23	1.11	1.43	2.02	6.61
Hybrid bisection	Sewell-exact	1.43	1.96	1.41	1.27	1.65	1.27	1.24	1.98	1.92	4.20
Hybrid bisection	Sewell-approximate	1.71	2.16	1.81	1.21	1.82	1.86	1.87	2.42	5.70	5.67
Hybrid bisection	True-maximum	1.11	1.85	1.01	1.06	1.11	1.55	1.00	1.41	1.39	2.36
Hybrid bisection	True-energy	1.18	1.60	1.19	1.15	1.32	1.16	1.23	1.51	1.65	3.46
Hybrid bisection	Bank–Sherman	1.31	1.69	1.38	1.18	1.48	1.02	1.44	1.19	1.00	2.52
Hybrid bisection	LD 1pt maximum	1.62	1.12	1.01	1.84	1.39	1.06	1.62	1.42	1.99	2.84
Hybrid bisection	LD 1pt energy	1.75	1.14	1.00	1.65	1.11	1.06	1.65	1.39	1.76	2.98
Hybrid bisection	LD 5pt maximum	1.00	1.00	1.02	1.22	1.18	1.00	1.34	1.39	1.68	3.04
Hybrid bisection	LD 5pt energy	1.08	1.13	2.38	1.00	1.22	1.90	1.40	1.81	2.27	2.50
Hybrid bisection	Uniform	8.04	15.96	2.91	1.17	4.00	4.27	12.27	11.88	48.45	3.42

(5) The ranking of the triangle division methods usually agrees with the angle bounds. On the basis of this observation and Figure 10 we conclude that it is best to use regular refinement if all the initial triangles are close to equilateral, and bisection otherwise.

(6) All methods perform well. The rates of convergence are close to 1, even in the presence of singularities, whereas the uniform grid is not yet fine enough to achieve this rate. The c/c_{best} ratio is, except for uniform grids, always small (usually less than 2).

Our main conclusion is that all of the methods considered are satisfactory for general purpose PDE solvers with adaptive refinement. In terms of accuracy for a given number of vertices, there is no significant difference between the methods considered. The selection of an adaptive refinement strategy should be based on other considerations, such as compatibility with the method for solving the resulting linear system or the accuracy of the error estimator. In the absence of other considerations, we would suggest the simplest, cheapest method. For triangle division this is newest node bisection, which does not require any computations to determine which node or edge to divide, has no "clean up" process after the refinement phase, and does not have "green" refinements that prevent the triangulations from being nested and must be removed before the next refinement phase. For an error indicator, the simplest method is the 1-point local Dirichlet indicator using the L_∞ norm. This uses only six multiplications and five additions each time the indicator for a pair of triangles is computed. It does, in addition, require the computation of six integrals the first time it is computed, but these are the same integrals needed for the new row of the matrix when the triangles are divided.

5. CONCLUSION

In this paper we considered several methods for the adaptive refinement of triangulations used for the solution of elliptic PDE's by the finite element method. Each method was briefly described and some properties of the methods were presented. In particular, the angle bounds of the triangle division methods were carefully examined and compared. A numerical experiment was conducted to compare the performance of the methods on the basis of the achieved accuracy per number of vertices.

It was found that adaptive refinement is superior to uniform refinement except for "nice" problems with smooth solutions. The optimal rate of convergence is regained even for our sample problems with singularities. The added effort of adapting the grid to the solution is unquestionably worthwhile. The methods performed nearly the same. The number of vertices required for a given error is usually within a factor of 2 for any two given methods. The only method we cannot recommend is the hybrid bisection method for dividing triangles which is more complicated than longest edge bisection and has essentially the same performance.

We conclude that the choice of an adaptive refinement method should be based on other considerations, such as compatibility with the method for solving the resulting linear system or the accuracy of the error estimator. In the absence of

other considerations, we suggest the use of newest node bisection with the L_∞ norm 1-point local Dirichlet error indicator, which is the simplest, cheapest method.

APPENDIX A. The Population of Test Problems

(1) The solution of this problem has a sharp peak. Two variations of the problem are run on different domains so as to give each type of refinement its optimal shape of triangles.

source	[11]
equation	Poisson

(a) peak at (.5, .117)

domain	unit square
initial triangles	isosceles right
angle bounds	regular—minimum 18.43°; maximum 116.57° bisection—minimum 45°; maximum 90°
solution	$x(x-1)y(y-1)e^{-100((x-.5)^2+(y-.117)^2)}$

(b) peak at (0, 0)

domain	hexagon with corners (1, 0), $(\frac{1}{2}, \sqrt{3}/2)$, $(-\frac{1}{2}, \sqrt{3}/2)$, (−1, 0), $(-\frac{1}{2}, -\sqrt{3}/2)$, and $(\frac{1}{2}, -\sqrt{3}/2)$
initial triangles	equilateral
angle bounds	regular—minimum 30°; maximum 90° bisection—minimum 30°; maximum 120°
solution	$(x+1)(x-1)(y+1)(y-1)e^{-100(x^2+y^2)}$

(2) The solution of this problem has a boundary layer along the lines $x = 1$ and $y = 1$.

source	[3], [11], and [14]
equation	$\nabla^2 u - 100u = f$
domain	unit square
initial triangles	isosceles right
angle bounds	regular—minimum 18.43°; maximum 116.57° bisection—minimum 45°; maximum 90°
solution	$\dfrac{\cosh(10x) + \cosh(10y)}{2 \cosh 10}$

(3) The solution of this problem has four mild peaks. It is fairly smooth so that a uniform grid should do nearly as well as adaptive grids. The equation has a nonconstant coefficient.

source	[11]
equation	$\nabla^2 u - (100 + \cos 2\pi x + \sin 3\pi y)u = f$

domain	unit square
initial triangles	isosceles right
angle bounds	regular—minimum 18.43°; maximum 116.57° bisection—minimum 45°; maximum 90°
solution	$-.31(5.4 - \cos 4\pi x)(\sin \pi x)(y^2 - y)$ $\cdot (5.4 - \cos 4\pi y)\left(\frac{1}{1 + \phi^4} - .5\right)$ $\phi = 4(x - .5)^2 + 4(y - .5)^2$

(4) The solution of this problem has a wavefront along the lines $x = .5$, $0 < y < .5$ and $y = .5$, $0 < x < .5$. The initial triangulation consists of tall isosceles triangles. The newest node bisection is run with two choices of initial peaks. In the first, the peaks are selected to be at one of the equal angles. In the second, the peaks are chosen to be at the nonequal angle.

source	[11]
equation	Poisson
domain	hexagon with corners $(1, \frac{1}{2})$, $(\frac{7}{8}, 1)$, $(\frac{1}{8}, 1)$, $(0, \frac{1}{2})$, $(\frac{1}{8}, 0)$, and $(\frac{7}{8}, 0)$
initial triangles	isosceles height $\frac{1}{2}$ width $\frac{1}{4}$
angle bounds	regular—minimum 14.04°; maximum 129.09° bisect(1) minimum 22.83°; maximum 129.09° bisect(2) minimum 14.04°; maximum 151.93°
solution	$\phi(x)\phi(y)$ $\phi(x) = 1$ for $x \leq .4$ $\phi(x) = 0$ for $x \geq .6$ ϕ is a quintic polynomial for $.4 \leq x \leq .6$ such that ϕ has two continuous derivatives

(5) The solution of this problem is an eighth-degree harmonic polynomial with moderate peaks at the four corners of the domain. The initial triangulation contains 3 shapes of triangles with some relatively small angles.

source	[11]
equation	Laplace
domain	$(-1, 1) \times (-1, 1)$
initial triangles	3 types of isosceles triangles
angle bounds	regular—minimum 9.46°; maximum 143.97° bisection—minimum 18.43°; maximum 143.13°
solution	$1.1786 - .1801p + .006q$ $p = x^4 - 6x^2y^2 + y^4$ $q = x^8 - 28x^6y^2 + 70x^4y^4 - 28x^2y^6 + y^8$

(6) The solution of this problem has a singularity at the origin, which is a reentrant corner of the domain. As in Problem 1, two variations of this

problem are used for different initial triangles. The strength of the singularity is different for the two versions.

	source	[2], [4], [5], [14], and [15]
	equation	Laplace
(a)	domain	L-shaped $(-1, 1) \times (-1, 1) \backslash (0, 1) \times (-1, 0)$
	initial triangles	isosceles right
	angle bounds	regular—minimum 18.43°; maximum 116.57° bisection—minimum 45°; maximum 90°
	solution	$r^{2/3} \sin \frac{2\theta}{3}$ (polar coordinates)
(b)	domain	hexagon as in 1b with a slit along the line $y = 0, x > 0$
	initial triangles	equilateral
	angle bounds	regular—minimum 30°; maximum 90° bisection—minimum 30°; maximum 120°
	solution	$r^{1/4} \sin \frac{\theta}{4}$

(7) The solution of this problem is harmonic, but drops very sharply near (.01, 0). If the domain were extended to $x = 0$, there would be a jump discontinuity in the boundary condition.

equation	Laplace
domain	$(.01, 1) \times (-1, 1)$
initial triangles	right triangles with legs of length 1 and .495
angle bounds	regular—minimum 12.43°; maximum 135.29° bisection—minimum 26.34°; maximum 127.33°
solution	$\tan^{-1} \frac{y}{x}$

(8) The coefficient function in the operator of this equation is discontinuous. $a(x, y)$ is piecewise constant with the values 1 and 100 on alternate triangles of the initial triangulation. The solution is continuous, but the first derivative has a jump discontinuity where a is discontinuous.

source	[3]
equation	$\nabla a \nabla u = 0$ where a is piecewise constant as described above
domain	hexagon as in 1b
initial triangles	equilateral
angle bounds	regular—minimum 30°; maximum 90° bisection—minimum 30°; maximum 120°
solution	$\frac{y(3x^2 - y^2)}{a}$

REFERENCES

1. BABUŠKA, I., AND AZIZ, A. K. On the angle condition in the finite element method. *SIAM J. Numer. Anal. 13* (1976), 214–226.
2. BABUŠKA, I., AND RHEINBOLDT, W. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal. 15* (1978), 736–754.
3. BANK, R. E. PLTMG User's Guide, Edition 4.0. Tech. Rep., Dept. of Mathematics, Univ. of California at San Diego, La Jolla, Calif., 1985.
4. BANK, R. E., AND SHERMAN, A. H. The use of adaptive grid refinement for badly behaved elliptic partial differential equations. In *Advances in Computer Methods for Partial Differential Equations III*, R. Vichnevetsky and R. S. Stepleman, Eds. IMACS, New Brunswick, 1979, pp. 33–39.
5. BANK, R. E., AND SHERMAN, A. H. An adaptive multilevel method for elliptic boundary value problems. *Computing 26* (1981), 91–105.
6. BANK, R. E., AND WEISER, A. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comput. 44* (1985), 283–301.
7. DE, J. P., GAGO, S. R., KELLY, D. W., ZIENKIEWICZ, O. C., AND BABUŠKA, I. A posteriori error analysis and adaptive processes in the finite element method: Part II—Adaptive mesh refinement. *Int. J. Numer. Meth. Eng. 19* (1983), 1621–1656.
8. FRIED, I. Condition of finite element matrices generated from nonuniform meshes. *AIAA J. 10* (1972), 219–221.
9. KELLY, D. W., DE, J. P., GAGO, S. R., ZIENKIEWICZ, O. C., AND BABUŠKA, I. A posteriori error analysis and adaptive processes in the finite element method: Part I—Error analysis. *Int. J. Numer. Meth. Eng. 19* (1983), 1593–1619.
10. MITCHELL, W. F. A comparison of adaptive refinement techniques for elliptic problems. Tech. Rep. UIUCDCS-R-87-1375, Dept. of Computer Science, Univ. of Illinois, Urbana, 1987.
11. RICE, J. R., HOUSTIS, E. N., AND DYKSEN, W. R. A population of linear, second order elliptic partial differential equations on rectangular domains. *Math. Comput. 36* (1981), 475–484.
12. RIVARA, M. C. Mesh refinement processes based on the generalized bisection of simplices. *SIAM J. Numer. Anal. 21* (1984), 604–613.
13. RIVARA, M. C. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Int. J. Numer. Meth. Eng. 20* (1984), 745–756.
14. RIVARA, M. C. Design and data structure of fully adaptive multigrid, finite-element software. *ACM Trans. Math. Softw. 10*, 3 (Sept. 1984), 242–264.
15. RIVARA, M. C. Adaptive finite element refinement and fully irregular and conforming triangulations. 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, 1986, pp. 359–369.
16. ROSENBERG, I. G., AND STENGER, F. A lower bound on the angles of triangles constructed by bisecting the longest side. *Math. Comput. 29* (1975), 390–395.
17. SEWELL, E. G. Automatic generation of triangulations for piecewise polynomial approximation. Ph.D. dissertation, Purdue Univ., West Lafayette, Ind., 1972.
18. SEWELL, E. G. A finite element program with automatic user-controlled mesh grading. In *Advances in Computer Methods for Partial Differential Equations III*, R. Vichnevetsky, and R. S. Stepleman, Eds. IMACS, New Brunswick, 1979, pp. 8–10.
19. SEWELL, E. G. *Analysis of a Finite Element Method: PDE/PROTRAN.* Springer-Verlag, New York, 1985.
20. ZIENKIEWICZ, O. C., KELLY, D. W., GAGO, J., AND BABUŠKA, I. Hierarchical finite element approaches, error estimates and adaptive refinement. In *The Mathematics of Finite Elements and Applications IV*, J. R. Whiteman, Ed. Academic Press, Orlando, Fla., 1982, pp. 313–346.

Received December 1987; revised November 1988; accepted December 1988