

AD-A186 534

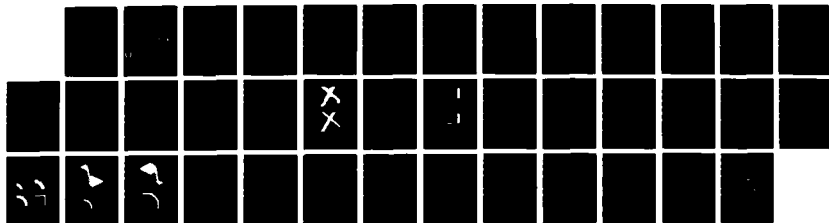
ADAPTIVE FINITE ELEMENT METHODS FOR PARABOLIC SYSTEMS
IN ONE- AND TWO-SPAL. (U) ARMY ARMAMENT RESEARCH
DEVELOPMENT AND ENGINEERING CENTER WAT..
S ADJERID ET AL. SEP 87 ARCCB-TR-87024

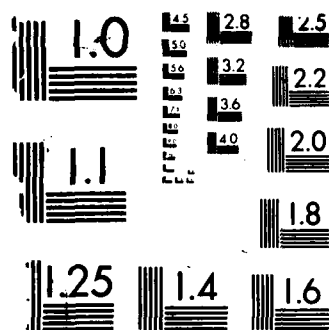
1/1

UNCLASSIFIED

F/G 12/1

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

DTIC FILE COPY

12

AD

TECHNICAL REPORT ARCCB-TR-87024

ADAPTIVE FINITE ELEMENT METHODS FOR PARABOLIC SYSTEMS IN ONE- AND TWO-SPACE DIMENSIONS

AD-A186 534

SLIMANE ADJERID
JOSEPH E. FLAHERTY

DTIC
ELECTE
OCT 22 1987
S D

SEPTEMBER 1987



US ARMY ARMAMENT RESEARCH, DEVELOPMENT
AND ENGINEERING CENTER

CLOSE COMBAT ARMAMENTS CENTER
BENÉT WEAPONS LABORATORY
WATERVLIET, N.Y. 12189-4050

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARCCB-TR-87024	2. GOVT ACCESSION NO. AD-A186	3. REPORTS CATALOG NUMBER 534
4. TITLE (and Subtitle) ADAPTIVE FINITE ELEMENT METHODS FOR PARABOLIC SYSTEMS IN ONE- AND TWO-SPACE DIMENSIONS		5. TYPE OF REPORT & PERIOD COVERED Final
7. AUTHOR(s) Slimane Adjerid and Joseph E. Flaherty (CONT'D ON REVERSE)		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army ARDEC Benet Weapons Laboratory, SMCAR-CCB-TL Watervliet, NY 12189-4050		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AMCMS No. 6111.02.H600.0 PRON No. 1A6DZ602NMSC
11. CONTROLLING OFFICE NAME AND ADDRESS US Army ARDEC Close Combat Armaments Center Picatinny Arsenal, NJ 07806-5000		12. REPORT DATE September 1987
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office)		13. NUMBER OF PAGES 32
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Presented at the Fourth Army Conference on Applied Mathematics and Computing, Cornell University, Ithaca, New York, 27-30 May 1986. Published in the Proceedings of the Conference. (CONT'D ON REVERSE)		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Adaptive Methods Finite Element Methods Parabolic Partial Differential Equations Local Mesh Refinement Mesh Moving Techniques Method of Lines		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) We discuss adaptive finite element methods for solving initial-boundary value problems for vector systems of parabolic partial differential equations in one- and two-space dimensions. One-dimensional systems are discretized using piecewise linear finite element approximations in space and a backward difference code for stiff ordinary differential systems in time. A spatial error estimate is calculated using (CONT'D ON REVERSE)		

7. AUTHORS (CONT'D)

Slimane Adjerid
Department of Computer Science
and
Center for Applied Mathematics and Advanced Computation
Rensselaer Polytechnic Institute
Troy, NY 12180-3590

Joseph E. Flaherty
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
and
US Army ARDEC
Close Combat Armaments Center
Benet Weapons Laboratory
Watervliet, NY 12189-4050

18. SUPPLEMENTARY NOTES (CONT'D)

This research was partially supported by the U.S. Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant Number AFOSR 85-0156 and the U.S. Army Research Office under Contract Number DAAL 03-86-K-0112. This work was used to partially fulfill the Ph.D. requirements of the first author at the Rensselaer Polytechnic Institute.

20. ABSTRACT (CONT'D)

piecewise quadratic approximations that employ nodal superconvergence to increase computational efficiency. This error estimate is used to move and refine the finite element mesh in order to equidistribute a measure of the total spatial error and to satisfy a prescribed error tolerance. Ordinary differential equations for the spatial error estimate and the mesh motion are integrated in time using the same backward difference software that is used to determine the finite element solution.

Two-dimensional systems are discretized using piecewise bilinear finite element approximations in space and backward difference software in time. A spatial error estimate is calculated using piecewise cubic approximations that take advantage of nodal superconvergence. This error estimate is used to locally refine a stationary finite element mesh in order to satisfy a prescribed spatial error tolerance.

Some examples are presented in order to illustrate the effectiveness of our error estimation technique and the performance of our adaptive algorithm.

UNCLASSIFIED

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. ONE-DIMENSIONAL ADAPTIVE PROCEDURES	3
II.1. Discrete System	3
II.2. Adaptive Algorithms	6
II.3. Computational Examples	10
III. TWO-DIMENSIONAL ADAPTIVE PROCEDURES	14
III.1. Discrete System	16
III.2. Local Refinement Algorithms	17
III.3. Computational Example	21
IV. DISCUSSION OF RESULTS AND CONCLUSIONS	25
REFERENCES	28

TABLES

1. Number of space-time cells on $0 \leq t \leq 1.2$, spatial discretization error at $t = 1.2$, and effectivity index at $t = 1.2$ as functions of error tolerance using stationary and moving mesh methods to solve Example 1.	12
2. Number of space-time cells as a function of error tolerance to solve Example 2 on $0 \leq t \leq 0.5$ using stationary and moving meshes.	14

LIST OF ILLUSTRATIONS

1. Top-level description of an adaptive finite element procedure with mesh motion and/or local mesh refinement/coarsening.	8
2. Mesh trajectories used to solve Example 1 with a tolerance of $1/8$ on stationary (upper) and moving (lower) meshes.	13
3. Mesh trajectories used to solve Example 2 with a tolerance of 0.1 on stationary (upper) and moving (lower) meshes.	15
4. A coarse mesh with four elements numbered 1 to 4 (top) and the resulting mesh after refining element 1 (bottom).	19

Evaluation	
Out	Final or Special
A-1	

	<u>Page</u>
5. Tree representation of the mesh shown in the lower portion of Figure 4.	20
6. Meshes that were used for Example 3 at $t = 0.2867$ (upper left), 0.2979 (upper right), 0.3055 (lower left), and 1 (lower right).	22
7. Surface (top) and contour (bottom) plots of calculated temperature for Example 3 at $t = 0.28674$.	23
8. Surface (top) and contour (bottom) plots of calculated temperature for Example 3 at $t = 0.3115$.	24

I. INTRODUCTION.

Adjerid and Flaherty [1-3]* developed adaptive finite element methods for solving m -dimensional vector systems of partial differential equations having the form

$$M(\mathbf{x}, t)u_t + f(\mathbf{x}, t, u, \nabla u) = \sum_{k=1}^d [D^k(\mathbf{x}, t, u)u_{x_k}]_{x_k}, \quad \mathbf{x} \in \Omega, \quad t > 0, \quad (1a)$$

subject to the initial and boundary conditions

$$u(\mathbf{x}, 0) = u^0(\mathbf{x}), \quad \mathbf{x} \in \Omega \cup \partial\Omega, \quad (1b)$$

$$\text{either } u_i(\mathbf{x}, t) = c_i(\mathbf{x}, t) \quad \text{or} \quad \sum_{k=1}^d \sum_{j=1}^m D_{ij}^k u_{j_{x_k}}(\mathbf{x}, t) v_k = c_i(\mathbf{x}, t), \\ \text{for } \mathbf{x} \in \partial\Omega, \quad t > 0, \quad i = 1, 2, \dots, m. \quad (1c)$$

They considered problems in one ($d = 1$) and two ($d = 2$) spatial dimensions with $\mathbf{x} = [x_1, \dots, x_d]^T$ denoting a position vector in $\mathbb{R}^d$, t denoting time, and Ω being either a segment of the real line or a rectangle. The subscripts t and x_k denote temporal and spatial partial derivatives, respectively, and $\mathbf{v} = [v_1, \dots, v_d]^T$ denotes the unit outer normal vector to the boundary $\partial\Omega$ of Ω . Problems were assumed to be parabolic and to have an isolated solution; thus, M and D^k , $k = 1, \dots, d$, are positive definite $m \times m$ matrices.

Adjerid and Flaherty discretized Eq. (1) in space using Galerkin's method with a piecewise linear polynomial basis in one dimension and piecewise bilinear polynomials in two dimensions. An a posteriori estimate of the spatial discretization error was calculated using Galerkin's method with piecewise quadratic functions in one dimension and piecewise cubic functions in two dimensions. In each case, a nodal superconvergence property of the finite element method was used to neglect errors at nodes and, thus, improve computational efficiency. The error estimate was used to control global [1] and local [2, 3] refinement procedures that added and/or deleted finite elements to the mesh in order to

* References are listed at the end of this report.

satisfy a prescribed global measure of the spatial discretization error. For one-dimensional problems, the error estimate was further used to move the finite element mesh so as to equidistribute the global error measure. Ordinary differential equations for the finite element solution, error estimate, and, in one dimension, mesh motion were integrated in time using the backward difference code DASSL [18] for stiff differential and algebraic systems.

Initially, a global refinement procedure was used in combination with mesh motion to satisfy prescribed error tolerances in the H^1 norm [1]. This procedure was replaced by a more efficient local mesh refinement strategy and some problem dependent parameters were removed from the mesh moving scheme [2]. In particular, numerical experiments indicated that the performance of the error estimation procedure could deteriorate when the system of equations governing mesh motion was too stiff. Adjerid and Flaherty [2] remedied this defect by limiting the stiffness of the mesh moving equations and using refinement, instead of mesh motion, to equidistribute the error estimate in these situations. They subsequently extended their finite element, error estimation and adaptive local refinement procedures to two-dimensional parabolic problems [3] and proved that the error estimate of References 1 and 2 converged to the true discretization error in H^1 as the mesh is refined for linear one-dimensional parabolic systems [4].

In Section II of this report, we review the one-dimensional adaptive procedures of Adjerid and Flaherty [1, 2], describe some improvements to their mesh refinement scheme, and present some examples that illustrate the relationship and interaction between mesh motion and refinement. The essential details of Adjerid and Flaherty's [3] two-dimensional procedure and the dynamic data structures used in its implementation are summarized in Section III. The results of a nonlinear two-dimensional example are also presented in Section III. Finally, in Section IV, we discuss our results and suggest some future directions.

II. ONE-DIMENSIONAL ADAPTIVE PROCEDURES.

The one-dimensional version of Eq. (1) consists of solving

$$M(x,t)u_t + f(x,t,u,u_x) = [D(x,t,u)u_x]_x, \quad x \in (a,b), \quad t > 0, \quad (2a)$$

$$u(x,0) = u^0(x), \quad x \in [a,b] \quad (2b)$$

$$\begin{aligned} \text{either } u_i(x,t) = c_i(t) \quad \text{or} \quad \sum_{j=1}^m D_{ij} u_{j_x}(x,t) = c_i(t), \\ \text{for } x = a, b, \quad t > 0, \quad i = 1, 2, \dots, m. \end{aligned} \quad (2c)$$

The unit subscripts on x and superscripts on D have been omitted for simplicity.

The procedures for discretizing Eq. (2) and estimating the spatial discretization error of its solution are identical to our earlier work [1, 2] and are briefly summarized in Section II.1. The essential details of our current adaptive procedure are presented in Section II.2 and some examples illustrating its capabilities and the interplay between mesh motion and refinement are presented in Section II.3.

II.1. Discrete System. We construct a weak form of Eq. (2) by assuming $u \in H_E^1$, selecting a test function $v \in H_0^1$, multiplying Eq. (2a) by v , integrating it on $a \leq x \leq b$, and integrating the diffusive term by parts to obtain

$$(v, Mu_t) + (v, f) + A(v, u) = v^T D(x, t, u) u_x \Big|_a^b, \quad \text{for all } v \in H_0^1, \quad t > 0, \quad (3a)$$

where

$$(v, u) = \int_a^b v(x, t)^T u(x, t) dx, \quad A(v, u) = \int_a^b v_x^T D(x, t, u) u_x dx. \quad (3b, c)$$

Recall that the Sobolev space H^1 consists of functions that are square integrable and have square integrable first spatial derivatives. Functions belonging to H_E^1 are further restricted to satisfy any essential (Dirichlet) boundary conditions in Eq. (2c), while functions in H_0^1 must satisfy homogeneous versions of any essential boundary conditions.

Initially u must satisfy

$$(v, u) = (v, u^0), \quad \text{for all } v \in H_0^1, \quad t = 0, \quad (3d)$$

and any natural (Neumann) boundary conditions in Eq. (2c) should be used to replace Du_x in the last term of Eq. (3a).

Finite element solutions of Eq. (3) are constructed by selecting finite dimensional approximations $U \in S_E^N \subset H_E^1$ and $V \in S_0^N \subset H_0^1$ of u and v , respectively, and finding U such that

$$(V, MU_t) + (V, f) + A(V, U) = V^T D(x, t, U) U_x|_a^b, \quad \text{for all } V \in S_0^N, \quad t > 0, \quad (4a)$$

$$(V, U) = (V, u^0), \quad \text{for all } V \in S_0^N, \quad t > 0. \quad (4b)$$

Specifically, we introduce a partition

$$\pi(t, N) := \{ a = x_0(t) < x_1(t) < \dots < x_N(t) = b \} \quad (5)$$

of $[a, b]$ into N moving subintervals $(x_{i-1}(t), x_i(t))$, $i = 1, 2, \dots, N$, $t \geq 0$, and select S_E^N and S_0^N to consist of piecewise linear polynomials with respect to this partition. The system of ordinary differential equations that result from this spatial discretization can be integrated in time using one of the many excellent software packages for solving stiff differential systems. We found that the backward difference code DASSL (cf. Petzold [18]) for differential and algebraic systems best fit our purposes.

The spatial discretization error of the finite element solution

$$e(x, t) = u(x, t) - U(x, t) \quad (6)$$

satisfies Eq. (3) with u replaced by $U + e$, i.e.,

$$(v, M(U_t + e_t)) + (v, f(\cdot, t, U + e, U_x + e_x)) + A(v, U + e) =$$

$$\mathbf{v}^T \mathbf{D}(x, t, \mathbf{U} + \mathbf{e})(\mathbf{U}_x + \mathbf{e}_x)|_a^b, \quad \text{for all } \mathbf{v} \in H_0^1, \quad t > 0. \quad (7a)$$

$$(\mathbf{v}, \mathbf{e}) = (\mathbf{v}, \mathbf{u}^0 - \mathbf{U}), \quad \text{for all } \mathbf{v} \in H_0^1, \quad t = 0. \quad (7b)$$

We approximate $\mathbf{e}$ by a function $\mathbf{E} \in \hat{S}_0^N$, where $\hat{S}_0^N$ is a finite dimensional subspace of H_0^1 consisting of piecewise quadratic functions that vanish on $\pi(t, N)$. We further approximate $\mathbf{v}$ by $\mathbf{V} \in \hat{S}_0^N$ and determine $\mathbf{E}$ as the solution of

$$(\mathbf{V}, \mathbf{M}(\mathbf{U}_t + \mathbf{E}_t)) + (\mathbf{V}, \mathbf{f}(\cdot, t, \mathbf{U} + \mathbf{E}, \mathbf{U}_x + \mathbf{E}_x)) + A(\mathbf{V}, \mathbf{U} + \mathbf{E}) = 0, \\ \text{for all } \mathbf{V} \in \hat{S}_0^N, \quad t > 0, \quad (8a)$$

$$(\mathbf{V}, \mathbf{E}) = (\mathbf{V}, \mathbf{u}^0 - \mathbf{U}), \quad \text{for all } \mathbf{V} \in \hat{S}_0^N, \quad t = 0. \quad (8b)$$

In constructing the error estimate $\mathbf{E}(x, t)$, we assumed the superconvergence of the piecewise linear finite element solution $\mathbf{U}(x, t)$, i.e., we assumed that $\mathbf{U}(x, t)$ converges at a faster rate on $\pi(t, N)$ than elsewhere on $a < x < b$. This superconvergence property was established by Thomee [19] and the convergence of $\mathbf{E}$ to $\mathbf{e}$ has been proven for linear problems by Adjerid and Flaherty [4].

The error estimate $\mathbf{E}(x, t)$ is used to control the refinement/coarsening strategy and the motion of $\pi(t, N)$. We determine mesh motion by solving the ordinary differential system

$$\dot{x}_i(t) - \dot{x}_{i-1}(t) = -\lambda(W_i - \bar{W}), \quad i = 1, 2, \dots, N, \quad (9a)$$

where λ is a non-negative parameter, W_i is an error indicator on the subinterval (x_{i-1}, x_i) , and $\bar{W}$ is the average of W_i , $i = 1, 2, \dots, N$. We shall take W_i to be the square of the local error estimate in H^1 , i.e.,

$$W_i(t) = \|\mathbf{E}\|_{1,i}^2 := \int_{x_{i-1}}^{x_i} [\mathbf{E}^T \mathbf{E} + \mathbf{E}_x^T \mathbf{E}_x] dx; \quad (9b)$$

however, other local measures can be used [2].

When $\lambda > 0$ and $W_i > \bar{W}$, the right-hand side of Eq. (9a) is negative and the nodes x_i and x_{i-1} move closer to each other. Similarly, the nodes $x_i(t)$ and $x_{i-1}(t)$ move apart when $\lambda > 0$ and $W_i < \bar{W}$. Coyle et al. [16] studied the stability of Eq. (9a) with respect to small perturbations from an equidistributing mesh (i.e., one where $W_i(t) = \bar{W}(t)$, $i = 1, 2, \dots, N$, $t \geq 0$) and showed that such perturbations could only grow by a bounded amount when $\lambda > 0$, $W_i > 0$, $i = 1, 2, \dots, N$, and the velocity of the equidistributing mesh remained finite for $t \geq 0$. They further showed that the mesh obtained by solving Eq. (9a) stayed closer to the equidistributing system when λ was large. This, however, introduces stiffness into the system which makes its solution expensive and, as noted, causes some difficulties with our error estimate. Adjerid and Flaherty [2] studied Eq. (9) and developed an adaptive algorithm for selecting λ as a function of t that balanced stiffness and equidistribution. The procedure for selecting λ will not be discussed further, but it has been used in the examples of Section II.3.

In order to maintain sparsity, we eliminate $\bar{W}$ by combining Eq. (9a) on two neighboring intervals and solve the scalar tridiagonal system

$$\dot{x}_{i+1} - 2\dot{x}_i + \dot{x}_{i-1} = -\lambda(W_{i+1} - W_i), \quad i = 1, 2, \dots, N-1, \quad t > 0. \quad (10)$$

The ordinary differential equations resulting from Eq. (8) and Eq. (10) are solved using the same backward difference software that is used to integrate the finite element system Eq. (4).

II.2. Adaptive Algorithms. In addition to controlling mesh motion, the error estimate E described in Section II.1 is used as an error indicator in conjunction with procedures that locally refine or coarsen the mesh. A top-level description of an adaptive local refinement/coarsening algorithm is presented in Figure 1 in a pseudo-PASCAL language. The procedure *adapfem* integrates the system Eqs. (4, 8, 10) from time $t_{initial}$ to t_{final} and attempts to keep the spatial error estimate $\|E\|_1 < TOL$, where TOL is a prescribed tolerance. The time steps that are selected by the temporal integration routine (e.g.,

DASSL) are denoted as $\Delta t[m]$, $m = 1, 2, \dots$, and the corresponding times are $tout[m]$, $m = 0, 1, \dots$, with $tout[0]$ initially set to $tinitial$. The integration is halted every $nstep$ steps or when $tout[m] = tfinal$ and the arrays Δt and $tout$ are recomputed with $tout[0]$ reset to the last computed time, i.e., $tout[nstep]$ or $tfinal$.

The spatial error estimate $\|E\|_1$ is checked whenever the temporal integration is halted. If $\|E\|_1 > TOL$, the last m integration steps are rejected and the mesh is refined by adding

$$N[i] := \max \{ \text{round}_\beta [\|E\|_{1,i} / \bar{E}] - 1, 0 \} \quad (11a)$$

elements uniformly to (x_{i-1}, x_i) , $i = 1, 2, \dots, N$. Here,

$$\text{round}_\beta(x) := \begin{cases} \text{trunc}(x) + 1, & \text{if } x - \text{trunc}(x) \geq \beta \\ \text{trunc}(x), & \text{otherwise,} \end{cases} \quad (11b)$$

where $0 < \beta < 1$, $\text{trunc}(x)$ evaluates the integer part of x , and

$$\bar{E} := 0.9TOL/N. \quad (11c)$$

The choice of $\beta = 0.2$ in Eq. (11) seemed to produce refined meshes that reliably reduced $\|E\|_1$ to approximately TOL the next time that the error estimate was checked. Further justification for this value of β is given in Adjerd and Flaherty [4].

The integration is redone from $tout[0]$ to $tout[m]$, where m is either $nstep$ or such that $tout[m] = tfinal$, on the refined mesh which has

$$N_r = N + \sum_{i=1}^N N[i] \quad (12)$$

elements. This process is repeated until $\|E(\cdot, tout[m])\|_1 \leq TOL$.

Elements can be deleted from a mesh whenever $\|E(\cdot, tout[m])\|_1 < TOL/3$ or whenever refinement was necessary to integrate from $tout[0]$ to $tout[m]$. The need to refine

```

procedure adapfem (tinitial , tfinal , nstep , TOL );

begin
    Calculate the initial conditions and an initial mesh;

                                { Integrate the system from tinitial to tfinal. }

    m := 0;
    tout[0] := tinitial;
    while tout[m] < tfinal do
        begin
            m := m + 1;
            redone := false;
            Integrate Eqs. (4, 8, 10) for one time step  $\Delta t[m]$ ;
            tout[m] := tout[m-1] +  $\Delta t[m]$ ;

                                { Check the error estimate. }

            if (m = nstep) or (tout[m] = tfinal) then
                begin
                    Compute a new value of  $\lambda$ , if necessary;

                                { Refine the mesh. }

                    while  $\|E(\cdot, tout[m])\|_1 > TOL$  do
                        begin
                            Add elements to the mesh;
                            Redo the integration on the refined mesh from
                                 $t = tout[0]$  to tout[m];
                            redone := true
                        end;

                                { Coarsen or regenerate the mesh. }

                        if ( $\|E(\cdot, tout[m])\|_1 < TOL/3$ ) or (redone)
                            then Delete elements from the mesh, if possible
                            else Generate a new mesh, if necessary;

                                tout[0] := tout[m];
                                m := 0
                            end { if m = nstep ... }
                        end { while tout[m] < tfinal }
                    end { adapfem };

```

Figure 1. Top-level description of an adaptive finite element procedure with mesh motion and/or local mesh refinement/coarsening.

often indicates that the spatial error pattern has changed and that fine grids may no longer be needed in some portions of the domain. A mesh is coarsened by uniting successive

pairs of elements, (x_{i-1}, x_i) and (x_i, x_{i+1}) , when $\|E(\cdot, tout[m])\|_{1,j} < TOL/3N$, $j = i, i+1$. This union of elements is only performed when a significant percentage of elements may be removed from a mesh. This strategy avoids the overhead associated with restarting the temporal integration routine.

If $TOL/3 \leq \|E(\cdot, tout[m])\|_1 \leq TOL$, we continue the temporal integration with the existing mesh provided that its speed is not too great and it is not close to equidistributing the local error indicators. A mesh where the error indicators are not equilibrated indicates that mesh motion and/or refinement are being performed in a suboptimal manner and that a new mesh may be more efficient. We use the following indicator to measure the effectiveness of a mesh $\pi(t, N)$ with respect to equidistributing the local error indicators:

$$\mu(\pi(t, N)) = \frac{2}{N\bar{W}} \left[\sum_{i=1}^N \left| \sum_{j=1}^i W_j - i\bar{W} \right| \right]. \quad (13)$$

If π is a mesh that equidistributes W_j , $j = 1, 2, \dots, N$, then $W_j = \bar{W}$, $j = 1, 2, \dots, N$ and $\mu(\pi) = 0$. Larger values of μ indicate increasing departures from equidistribution. For example, suppose all of the error is concentrated in the first element, i.e., $W_1 = N\bar{W}$ and $W_j = 0$, $j = 2, 3, \dots, N$. Then $\mu(\pi) = N$, which we interpret as meaning that at least one element (the last one) will have to cross N elements in order to equidistribute the error indicators. If all of the error were concentrated in the $N/2$ th element, then $\mu(\pi) = N/2$, indicating that one element has to cross $N/2$ elements of π . Whenever the mesh speed is too fast and $\mu(\pi(tout[m], N)) > 0.1N$, we generate a new mesh that approximately equidistributes the error indicators by iteratively removing elements with small error indicators and refining those having large error indicators.

Additional details of our procedures, such as the generation of new initial conditions whenever the number of elements in a mesh changes, are as described in Adjerd and Flaherty [2].

In Section II.3, we present some calculations performed on stationary meshes. These were done by using a code based on adapfem with the mesh moving parameter $\lambda = 0$. Additionally, we only generated new meshes when $\mu(\pi(\text{tout}[m], N)) > 0.4N$ in order to avoid excessive restarting of the temporal integration routine.

II.3. Computational Examples. We conclude this section by presenting some examples that illustrate our adaptive strategies and also attempt to appraise the relative advantages of mesh moving and local refinement. There are several potential reasons why an adaptive procedure that combines mesh moving with refinement would be very efficient. Mesh moving techniques are inexpensive relative to refinement (cf. Arney and Flaherty [6]) and the use of mesh motion should reduce the need for refinement. Mesh motion can also reduce the necessity of restarting the temporal integrator, which is an important consideration in a method of lines approach such as ours. Some refinement is essential, however, since mesh motion alone cannot generally satisfy prescribed error tolerances. Furthermore, rapid mesh motion, e.g., towards an evolving region of high error, can severely restrict time steps and diminish the efficiency of an adaptive procedure (cf. Adjerid and Flaherty [2]). Finally, many more numerical techniques converge at higher rates on uniform meshes than they do on nonuniform moving meshes.

There is, thus, a need to quantify the optimal use of mesh moving with local refinement; however, this is a very difficult problem and there have been very few attempts in this direction. Arney and Flaherty [6] presented some computational results comparing mesh moving and local refinement procedures for two-dimensional hyperbolic systems. Bieterman, Flaherty, and Moore [15] attempted to compare adaptive local refinement and method of lines procedures for one-dimensional parabolic problems and noted the difficulties in finding appropriate performance measures. Herein, we apply a code based on our adaptive procedures to two computational examples and compare results on moving and stationary meshes. We use the total number of space-time cells to integrate the partial differential system from t_{initial} to t_{final} as a measure of

performance. A similar measure of computational complexity was used by Arney and Flaherty [6]. It has several apparent deficiencies, such as not providing an indication of the effort devoted to the various segments of the adaptive algorithm.

Example 1. Consider the linear heat conduction problem

$$u_t + u_x + g(x, t) = u_{xx}, \quad -1 < x < 1, \quad t > 0. \quad (14a)$$

$$u(x, 0) = u^0(x), \quad -1 \leq x \leq 1, \quad (14b)$$

$$u(-1, t) = c_1(t), \quad u(1, t) = c_2(t), \quad t \geq 0. \quad (14c,d)$$

We select g , u^0 , c_1 , c_2 , so that the exact solution of Eq. (14) is

$$u(x, t) = 1 - \frac{1}{2} \{ \tanh[10(x-t+0.8)] + \tanh[20(x+2t-1.6)] \}. \quad (15)$$

Equation (15) represents two wave fronts initially centered at $x = -0.8$ and $x = 1.6$ and moving with speeds 1 and -2, respectively. The center of the fastest front enters the domain $(-1, 1)$ at $x = 1$ and $t = 0.3$.

We solved Eq. (14) for $0 \leq t \leq 1.2$ using tolerances of 2^{-k} , $k = 2, 3, 4, 5$, in H^1 with adaptive procedures on moving and stationary meshes. The total number of space-time cells used on $0 \leq t \leq 1.2$, the exact error $\|e\|_1$ at $t = 1.2$, and the effectivity index

$$\theta := \|E\|_1 / \|e\|_1 \quad (16)$$

at $t = 1.2$ are presented in Table 1. The moving and stationary mesh trajectories that were used to solve Eq. (14) with a tolerance of $1/8$ are shown in Figure 2.

Solutions on moving meshes used less than half of the space-time cells of those on stationary meshes. A larger number of cells are needed with a stationary mesh because the temporal integration must be restarted more often and more time steps must be redone due to a failure to satisfy the error tolerance. In each case, the actual error was less than

the prescribed tolerance and fine meshes were concentrated in high-error regions. The effectivity index is a common method of appraising the performance of an error estimation technique (cf., e.g., Babuska et al. [9]). Ideally, θ should not deviate appreciably from unity and should approach unity as N increases. The results of Table 1 suggest that this is the case. The performance of our error estimate seems to be slightly better on a stationary mesh than on a uniform mesh.

Tol.	Stationary Mesh			Moving Mesh		
	No. Cells $\times 10^{-4}$	$\ e\ _1$	θ	No. Cells $\times 10^{-4}$	$\ e\ _1$	θ
1/4	4.11	0.1803	0.983	1.60	0.1725	0.979
1/8	8.67	0.0848	0.994	2.79	0.0903	0.993
1/16	26.94	0.0413	0.996	9.50	0.0566	0.988
1/32	47.72	0.0230	0.998	20.27	0.0282	0.996

Table 1. Number of space-time cells on $0 \leq t \leq 1.2$, spatial discretization error at $t = 1.2$, and effectivity index at $t = 1.2$ as functions of error tolerance using stationary and moving mesh methods to solve Example 1.

Example 2. Consider the reaction-diffusion system

$$u_t = u_{xx} - Du e^{-\delta T}, \quad LT_t = T_{xx} + \alpha D u e^{-\delta T}, \quad 0 < x < 1, \quad t > 0, \quad (17a,b)$$

$$D = Re^{\delta/\alpha\delta}, \quad (17c)$$

$$u(x,0) = T(x,0) = 1, \quad 0 \leq x \leq 1, \quad (17d,e)$$

$$u_x(0,t) = T_x(0,t) = 0, \quad u(1,t) = T(1,t) = 1, \quad t > 0. \quad (17f,g,h,i)$$

This model was studied by Kapila [17] and used to describe a single one-step reaction ($A \rightarrow B$) of a mixture in the region $0 < x < 1$. The quantity u is the mass fraction of the reactant, T is the reactant temperature, L is the Lewis number, α is the heat release, δ is the activation energy, D is the Damkohler number, and $R > 0.88$ is the reaction rate.

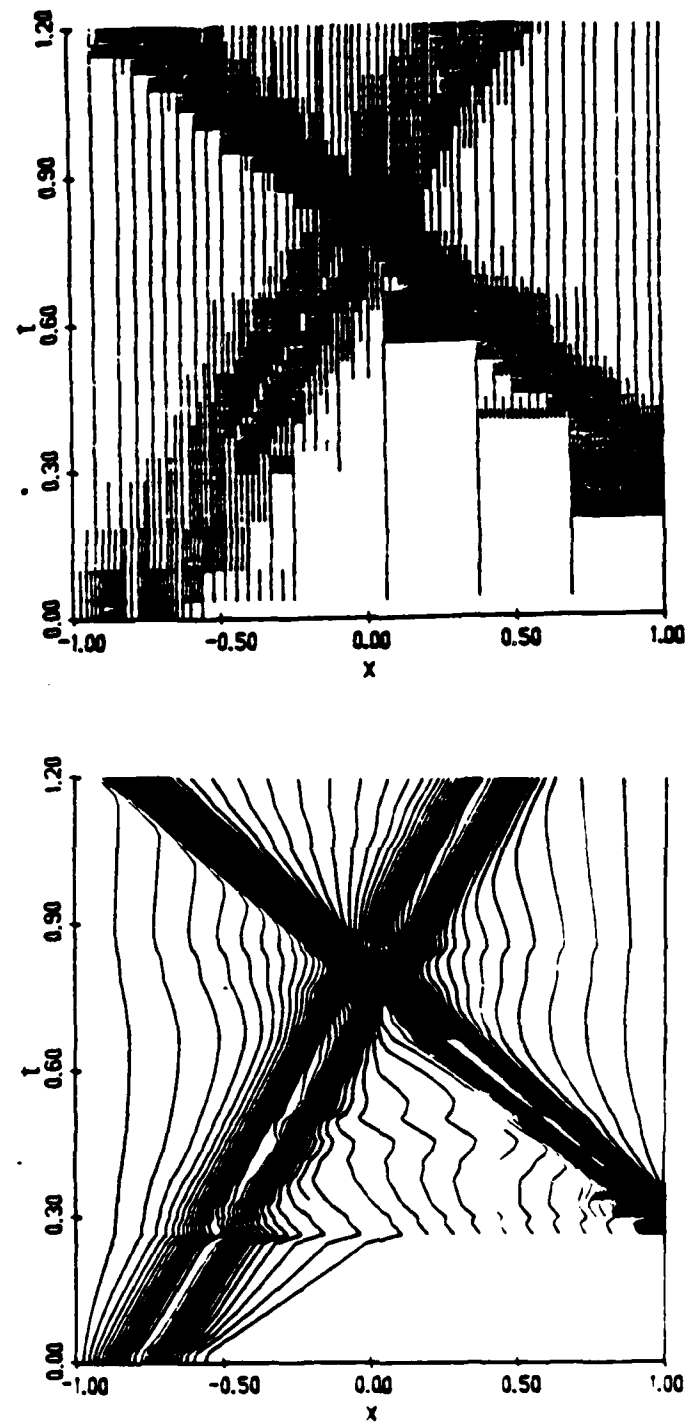


Figure 2. Mesh trajectories used to solve Example 1 with a tolerance of $1/8$ on stationary (upper) and moving (lower) meshes.

When L is near unity, the temperature slowly increases with a "hot spot" forming at $x = 0$. At some time $t > 0$, ignition occurs and the temperature at $x = 0$ jumps rapidly from near unity to near $1 + \alpha$. A steep flame front then forms and propagates towards $x = 1$ with speed proportional to $e^{\alpha\delta/2(1+\alpha)}$. In practical problems, α is about unity and δ is large; thus, the flame front moves exponentially fast after ignition. The solution tends to a steady state once the flame has reached $x = 1$.

We solved Eq. (17) for $0 \leq t \leq 0.5$ with $\alpha = 1$, $\delta = 20$, and $R = 5$ using tolerances of 0.2, 0.1, and 0.05 on stationary and moving meshes. The number of space-time cells needed to solve these problems is presented in Table 2 and the mesh trajectories for both the stationary and moving mesh calculations with a tolerance of 0.1 are shown in Figure 3. As in Example 1, the number of stationary space-time cells is approximately double the number of moving space-time cells.

Tolerance	Number of Space-Time Cells	
	Stationary Mesh	Moving Mesh
0.2	29700	16300
0.1	62300	30300
0.05	170800	118100

Table 2. Number of space-time cells as a function of error tolerance to solve Example 2 on $0 \leq t \leq 0.5$ using stationary and moving meshes.

III. TWO-DIMENSIONAL ADAPTIVE PROCEDURES.

Our finite element, error estimation, and local refinement procedures for two-dimensional partial differential systems closely parallel our one-dimensional methods and are briefly summarized in Section III.1 and III.2. The representation of data and its management are much more complicated in two dimensions, and we use a dynamic tree-data structure to store and retrieve information about the mesh, solution, and error esti-

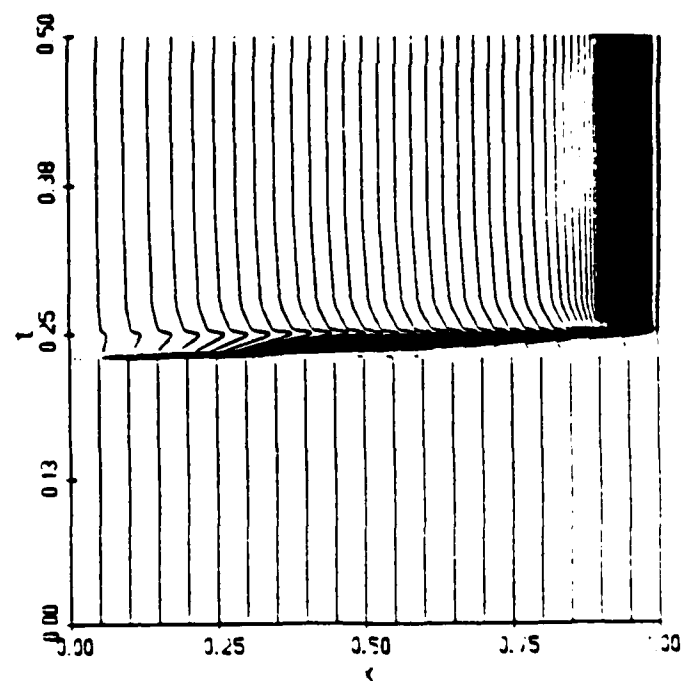
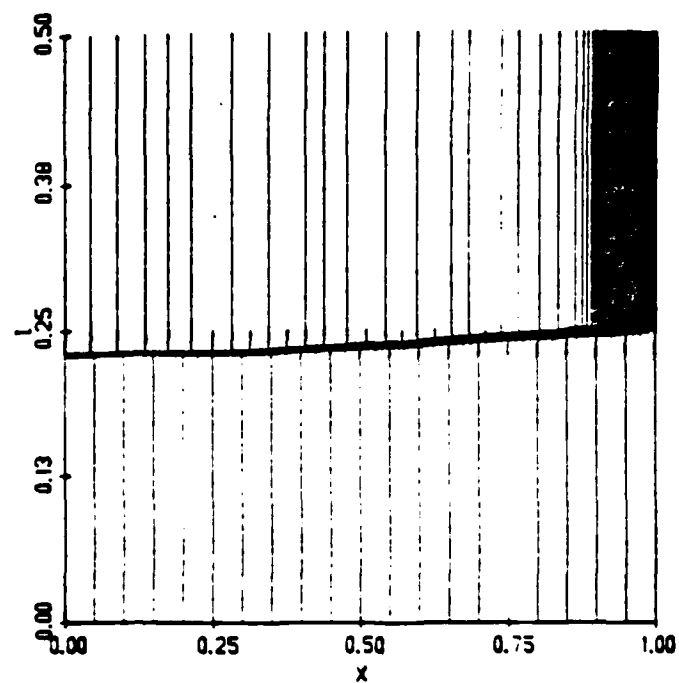


Figure 3. Mesh trajectories used to solve Example 2 with a tolerance of 0.1 on stationary (upper) and moving (lower) meshes.

mate. Similar structures have been used by other investigators (cf., e.g., Babuska et al. [8-10] and Bank et al. [12-14]) to design adaptive procedures for elliptic systems, and they have been shown to be an effective means of reducing storage and access overhead. The essential details of our tree structure are described in Section III.2 and a two-dimensional combustion problem, similar to Example 2, is presented in Section III.3.

III.1. Discrete System. A weak form of Eq. (1) is constructed in the manner described in Section II.1 for one-dimensional problems. Thus, we seek to determine $u \in H_E^1$ such that

$$(v, u_t) + (v, f(\cdot, t, u, \nabla u)) + A(v, u) = \int_{\partial\Omega} [v^T D^1 u_{x_1} v_1 + v^T D^2 u_{x_2} v_2] d\sigma, \quad \text{for all } v \in H_0^1, \quad t > 0, \quad (18a)$$

$$(v, u) = (v, u^0), \quad \text{for all } v \in H_0^1, \quad t = 0, \quad (18b)$$

where

$$(v, u) = \int_{\Omega} v(x, y, t)^T u(x, y, t) dx_1 dx_2, \quad (18c)$$

$$A(v, u) = \int_{\Omega} [v_{x_1}^T D^1(x, t, u) u_{x_1} + v_{x_2}^T D^2(x, t, u) u_{x_2}] dx_1 dx_2. \quad (18d)$$

We have set the mass matrix M in Eq. (1) to the identity matrix for simplicity.

The functions u and v are approximated by $U \in S_E^N \subset H_E^1$ and $V \in S_0^N \subset H_0^1$, respectively, where S_E^N and S_0^N are spaces of bilinear polynomials with respect to a piecewise rectangular partition of the rectangular domain Ω . The finite element solution U is obtained by solving

$$(V, U_t) + (V, f(\cdot, t, U, \nabla U)) + A(V, U) = \int_{\partial\Omega} [V^T D^1 U_{x_1} v_1 + V^T D^2 U_{x_2} v_2] d\sigma, \quad \text{for all } V \in S_0^N, \quad t > 0, \quad (19a)$$

$$(V, U) = (V, u^0), \quad \text{for all } V \in S_0^N, \quad t = 0. \quad (19b)$$

As in the one-dimensional case, the spatial error $e(\mathbf{x}, t) := u(\mathbf{x}, t) - U(\mathbf{x}, t)$ is approximated by $E \in \hat{S}_E^N \subset H_E^1$. In two dimensions, we select the finite dimensional space $\hat{S}_E^N$ to consist of piecewise cubic functions with respect to a piecewise rectangular partition of Ω . The cubic functions are biquadratic polynomials that are missing their quartic terms (i.e., serendipity functions in the terminology of Zienkiewicz [20]) and further vanish at the vertices of each element. Thus, once again we take advantage of nodal superconvergence to simplify our approximation of the discretization error. However, there is very little theoretical justification of the superconvergence property for two-dimensional problems and we are relying on computational evidence [3] and our one-dimensional theory [4].

The approximate error E is determined by replacing u and v in Eq. (18) by $U + E$ and $V \in \hat{S}_0^N \subset H_0^1$, respectively, where $\hat{S}_0^N$ is composed of the same cubic functions as $\hat{S}_E^N$, and solving

$$(V, U_t + E_t) + (V, f(\cdot, t, U + E, \nabla(U + E))) + A(V, U + E) =$$

$$\int_{\partial\Omega} [V^T D^1(U + E)_{x_1} v_1 + V^T D^2(U + E)_{x_2} v_2] d\sigma, \quad \text{for all } V \in \hat{S}_0^N, \quad t > 0, \quad (20a)$$

$$(V, E) = (V, u^0 - U^0), \quad \text{for all } V \in \hat{S}_0^N, \quad t > 0. \quad (20b)$$

The resulting ordinary differential equations (19) and (20) for the solution and error estimate are integrated in time using a code for stiff systems (e.g., DASSL).

III.2. Local Refinement Algorithms. A top-level description of our two-dimensional adaptive procedure closely resembles the one-dimensional algorithm shown in Figure 1, except that we have no mesh moving procedures, as yet. Initially, the domain Ω is partitioned into a "base" mesh of $N \times M$ rectangular elements, which is the coarsest mesh that can be used to solve the problem. Refinement is performed by bisecting the edges of a coarser element, thus, creating four elements where there was previously one. A base mesh having four elements and a refined mesh obtained by bisecting one of them is shown

in Figure 4.

The refinement process may be repeated, i.e., elements may be bisected again to create four new elements. Additionally, quartets of elements that were created by refinement may be subsequently deleted if they are no longer needed to maintain accuracy. Bilinear approximations in S_E^N and S_0^N and cubic approximations in $\hat{S}_E^N$ and $\hat{S}_0^N$ are constrained to be linear and quadratic, respectively, on edges between elements of different levels in order to maintain continuity of U and E on Ω .

The mesh is organized as a tree structure with the domain Ω being the root of the tree and the $N \times M$ elements of the base mesh being offsprings of the root. All nonleaf nodes of the tree, other than the root node, have four offsprings which correspond to the four elements created by refining its parent element. The domain Ω is referred to as level zero of the tree, the elements of the base mesh are level one, and the levels increase as elements are recursively refined. The tree structure for the mesh shown in the lower portion of Figure 4 is displayed in Figure 5.

Each node of the tree contains the following information:

- a. The element number, say k , of the finite element
- b. The level l of the tree
- c. Pointers to the four vertex nodes of element k
- d. Pointers to the four midside nodes of element k , which are needed to represent E
- e. Pointers to the four elements neighboring element k , with a null pointer used when an edge of element k is on the boundary
- f. A pointer to the parent of element k
- g. Pointers to the four sons of element k , with null pointers used when element k is a leaf node of the tree

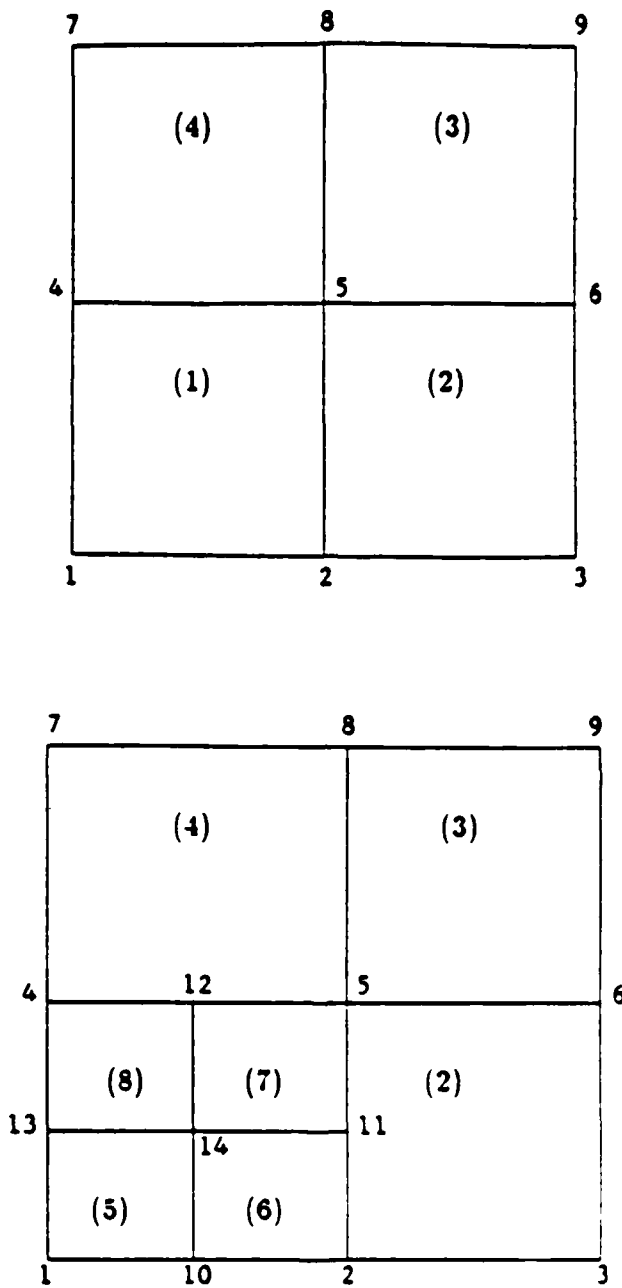


Figure 4. A coarse mesh with four elements numbered 1 to 4 (top) and the resulting mesh after refining element 1 (bottom).

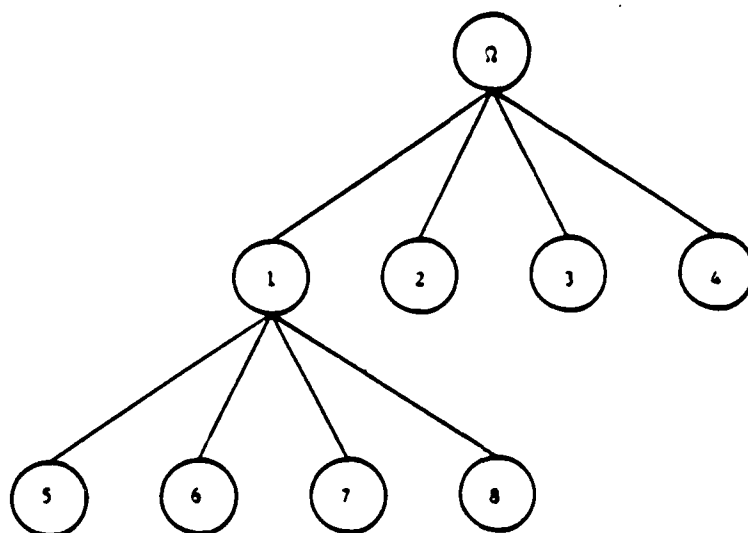


Figure 5. Tree representation of the mesh shown in the lower portion of Figure 4.

As in the one-dimensional algorithm of Figure 1, elements are added to a mesh when $\|E\|_1 > TOL$ and deleted from a mesh when either $\|E\|_1 < TOL/3$ or when refinement was necessary to integrate to the current time. Our refinement and deletion procedures impose the following two rules, which Bank et al. [12-14] found to aid the efficiency and accuracy of their refinement process for elliptic systems:

- a. The *1-irregular* rule, which states that neighboring elements can differ by at most one level of the tree
- b. The *3-neighbor* rule, which states that any element where the number of edges containing elements at a higher level of the tree and the number of boundary edges totals to three or more must be refined.

Refinement is performed by examining the elements of a mesh by levels, proceeding from the root to the leaf nodes of the tree. An element k is refined by dividing it into four subelements whenever $\|E\|_{1,k} > TOL/\sqrt{N_e}$, where N_e is the number of elements in the mesh. Elements are deleted from meshes, other than the base $N \times M$ mesh, by pruning the tree. A quartet of elements having the same parent is deleted if:

- a. Every element in the quartet has no offsprings
- b. Every neighbor of the elements in the quartet are at the same or lower level of the tree
- c. The average error estimate of the four elements is less than $TOL/3\sqrt{N_e}$.

Additional details pertaining to other aspects of our adaptive procedures are presented in Adjerid and Flaherty [3].

III.3. Computational Example. A code based on our two-dimensional local mesh refinement procedure has been written and applied to several problems [3]. Herein, we present the results of a two-dimensional version of the model combustion problem considered in Example 2.

Example 3. Consider the partial differential system on the rectangular domain $\Omega := \{ (x_1, x_2) \mid 0 < x_1, x_2 < 1 \}$

$$T_t = T_{x_1 x_1} + T_{x_2 x_2} + D(1 + \alpha - T)e^{-\delta T}, \quad (x, y) \in \Omega, \quad t > 0, \quad (21a)$$

$$T(x, 0) = 1, \quad x \in \Omega \cup \partial\Omega, \quad (21b)$$

$$T_{x_1}(0, x_2, t) = 0, \quad T(1, x_2, t) = 1, \quad 0 \leq x_2 \leq 1, \quad t > 0, \quad (21c, d)$$

$$T_{x_2}(x_1, 0, t) = 0, \quad T(x_1, 1, t) = 1, \quad 0 \leq x_1 \leq 1, \quad t > 0. \quad (21e, f)$$

All of the parameters are as described in Example 2. The Lewis number L has been set to unity and, in this case, the mass fraction $u = 1 + (1 - T)/\alpha$.

We solved Eq. (21) with $\alpha = 1$, $\delta = 20$, and $R = 5$ using a spatial error tolerance of 0.2. Mesh refinement had to be restricted to a maximum of two levels because of virtual memory restrictions on our computing system. The meshes that were created at $t = 0.2867$, 0.2979, 0.3055, and 1 are shown in Figure 6. Surface and contour plots of the calculated temperatures at $t = 0.28674$ and 0.3115 are presented in Figures 7 and 8, respectively.

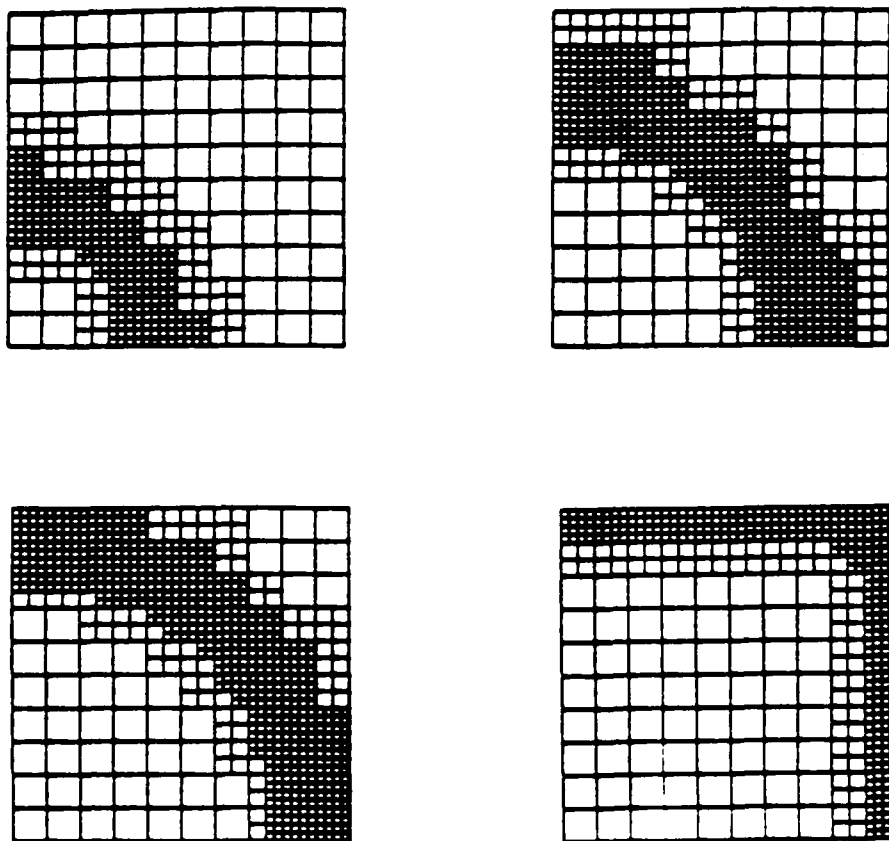


Figure 6. Meshes that were used for Example 3 at $t = 0.2867$ (upper left), 0.2979 (upper right), 0.3055 (lower left), and 1 (lower right).

The temperature slowly increases until ignition occurs at approximately $t = 0.28$. The temperature at the origin then jumps from near unity to near two. A circularly-shaped reaction front forms and moves radially with a speed of approximately 30 towards the

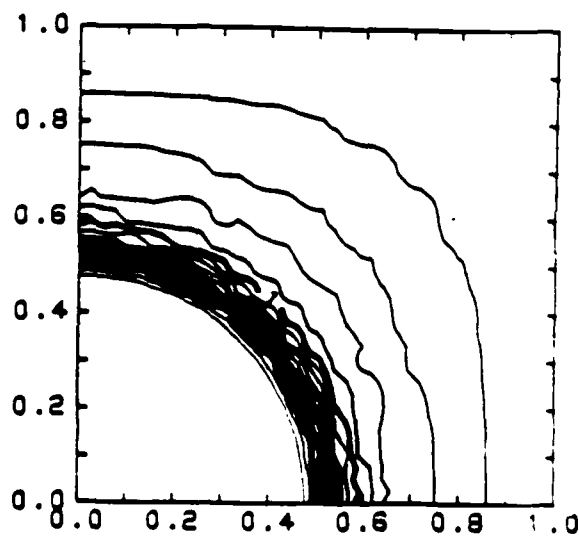
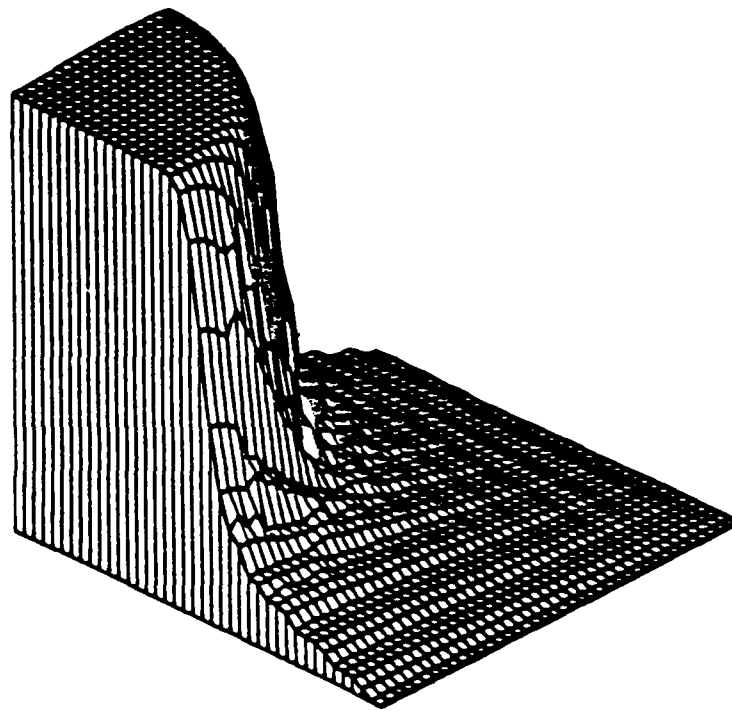


Figure 7. Surface (top) and contour (bottom) plots of calculated temperature for Example 3 at $t = 0.28674$.

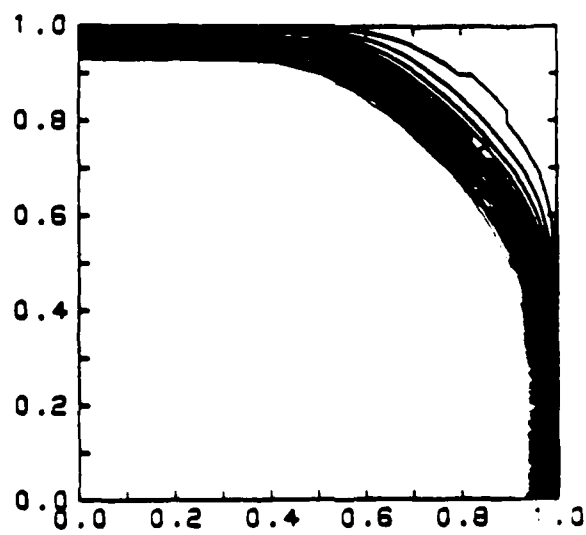
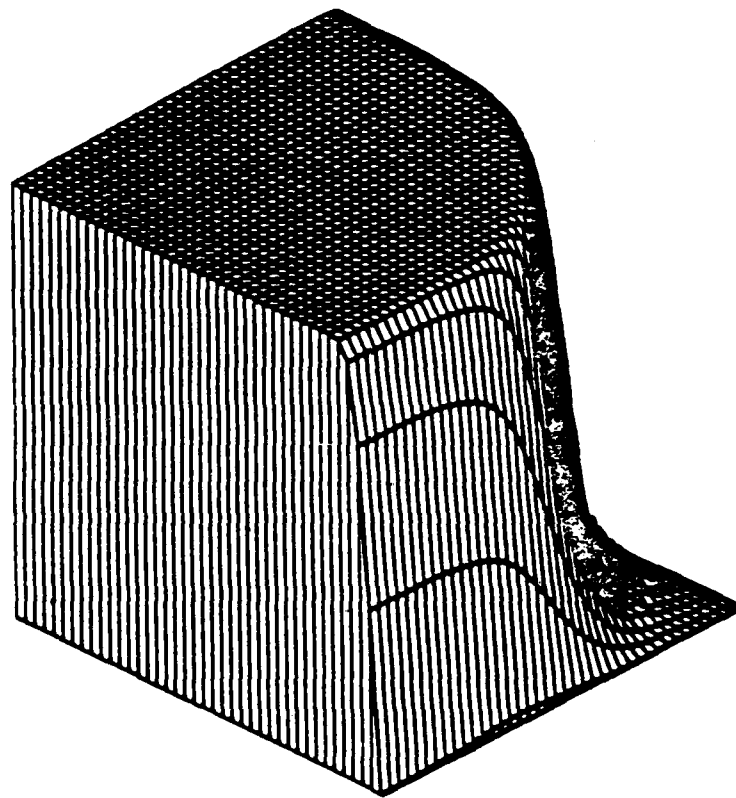


Figure 8. Surface (top) and contour (bottom) plots of calculated temperature for Example 3 at $t = 0.3115$.

boundaries. A steady state is reached at about $t = 0.32$. Refinement is confined to the vicinity of the reaction front. The results of Figures 7 and 8 show some small oscillations in the temperature ahead of the reaction front. At present, we are unsure if these oscillations are caused by interpolation inaccuracies in our plotting routines, inadequate resolution of the finite element solution due to our restricting the number of levels of refinement, or an instability of the reaction front. We plan to explore these matters further using a combination of numerical and asymptotic techniques.

Our results on this difficult nonlinear problem are very encouraging; however, we anticipate that greater efficiency could be achieved by combining local mesh refinement with mesh moving as in the one-dimensional procedures described in Section II.

IV. DISCUSSION OF RESULTS AND CONCLUSIONS.

We have developed adaptive local mesh refinement finite element procedures for solving vector systems of parabolic partial differential equations in one and two dimensions. The nodal superconvergence property of the finite element method on parabolic systems has been used to calculate and estimate the spatial discretization error and mesh motion has been combined with local mesh refinement for one-dimensional problems.

Examples 1 and 2 were designed to illustrate the performance of our one-dimensional procedures and to characterize the importance of mesh motion as an adaptive technique relative to mesh refinement. These experiments indicate that our combination of mesh moving and refinement can obtain solutions with about one-half the total number of space-time cells of calculations performed using only refinement. We emphasize the preliminary nature of these results. Many more experimental and theoretical investigations will be necessary before firm conclusions can be reached regarding the optimal combination of mesh motion and refinement. Appropriate performance measures and optimality conditions are yet to be specified. There is also a strong temptation to compare coded

implementations of procedures and, at this stage, we are interested in more theoretical bounds on an algorithm's performance.

Comparisons of the exact and estimated errors, presented in Example 1 and in References 1 through 4, give us some confidence in the accuracy of our error estimate. Additionally, the results of Example 3 provide an indication of the robustness of our methods. This is a difficult nonlinear two-dimensional problem; yet, we solved it without intervention and a priori knowledge of the solution. No special initial mesh was used, fine meshes were automatically added to the vicinity of the reaction front, and the fine meshes followed the dynamics of the problem. Indeed, our one- and two-dimensional techniques seem to be well-suited for the automatic solution of reaction-diffusion systems.

Despite our preliminary success, there is a great deal more that should be done to justify and improve the performance of our procedures. As noted, rigorous analyses of the convergence of our error estimate to the true error have only been done for one-dimensional linear parabolic problems on stationary meshes [4]. Dimensional, nonlinear, and refinement effects should be included in a complete analysis. This is a difficult task, as very few analyses of two-dimensional time dependent problems with refinement have appeared in the literature.

Several computational procedures in our approach might also be improved. For example, a sparse Gaussian elimination procedure was used to solve the linear algebraic systems associated with the temporal integration of Eqs. (4), (8), and (10) in one dimension and Eqs. (19) and (20) in two dimensions. The solution of linear systems is a significant part of the total computational effort, and it is possible that iterative schemes, such as multigrid methods, could substantially improve performance and reduce storage. Multigrid iteration was used successfully in the adaptive PLTMG package for elliptic systems by Bank et al. [13].

We are also studying the addition of mesh moving capabilities to our two-dimensional algorithm, the use of higher-order finite element approximations, and implementations of our procedures on vector and parallel computers. A simple, stable and explicit mesh moving technique, that may be useful for our purposes, was developed by Arney and Flaherty [5] for two-dimensional hyperbolic systems. This procedure dramatically reduced errors and enhanced the resolution of their solutions (cf. Arney and Flaherty [6]). We are developing procedures for two-dimensional parabolic problems that use piecewise biquadratic finite element approximations as solution spaces and piecewise cubic approximations as error estimates. Babuska [7] has shown that the error associated with even-degree polynomial finite element approximations for elliptic problems is principally due to the error in the interior of the element. Thus, the error on element boundaries may be neglected. Babuska and Yu [11] have implemented procedures for elliptic systems based on this theory and we are studying their utility for parabolic problems. Finally, our tree structure is well-suited for parallel computation and we are exploring its use on a variety of parallel computing systems.

REFERENCES

1. S. Adjerid and J.E. Flaherty, "A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time Dependent Partial Differential Equations," *SIAM J. Numer. Anal.*, 23 (1986), pp. 778-796.
2. S. Adjerid and J.E. Flaherty, "A Moving Mesh Finite Element Method with Local Refinement for Parabolic Partial Differential Equations," *Comp. Meths. Appl. Mech. Engr.*, 56 (1986), pp. 3-26.
3. S. Adjerid and J.E. Flaherty, "A Local Refinement Finite Element Method for Two-Dimensional Parabolic Systems," Tech. Rep. No. 86-7, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, 1986.
4. S. Adjerid and J.E. Flaherty, "Local Refinement Finite Element Methods on Stationary and Moving Meshes for One-Dimensional Parabolic Systems," in preparation.
5. D.C. Arney and J.E. Flaherty, "A Two-Dimensional Mesh Moving Technique for Time Dependent Partial Differential Equations," Tech. Rep. No. 85-9, Department of Computer Science, Rensselaer Polytechnic Institute, Troy, 1985. Also *J. Comput. Phys.*, 67 (1986), pp. 124-144.
6. D.C. Arney and J.E. Flaherty, "An Adaptive Method with Mesh Moving and Refinement for Time-Dependent Partial Differential Equations," *Trans. Fourth Army Conf. Appl. Maths. and Comput.*, ARO Report 87-1, U. S. Army Research Office, Research Triangle Park, NC, (1987), pp. 1115-1142.
7. I. Babuska, personal communication, 1986.
8. I. Babuska and A. Miller, "A Posteriori Error Estimates and Adaptive Techniques for the Finite Element Method," Tech. Note BN-968, Institute for Physical Science and Technology, University of Maryland, College Park, MD, 1981.

9. I. Babuska, A. Miller, and M. Vogelius, "Adaptive Methods and Error Estimation for Elliptic Problems of Structural Mechanics," in *Adaptive Computational Methods for Partial Differential Equations*, I. Babuska, J. Chandra, J.E. Flaherty, eds., SIAM, Philadelphia, 1983, pp. 57-73.
10. I. Babuska and W. Rheinboldt, "Error Estimates for Adaptive Finite Element Computations," *SIAM J. Numer. Anal.*, 15 (1978), pp. 736-734.
11. I. Babuska and D. Yu, "Asymptotically Exact A-Posteriori Error Estimator for Biquadratic Elements," Tech. Note BN-1050, Institute for Physical Science and Technology, University of Maryland, College Park, MD, 1986.
12. R.E. Bank, "PLTMG Users' Guide," June 1981 version, Technical Report, Department of Mathematics, University of California at San Diego, La Jolla, CA, 1982.
13. R.E. Bank and A. Sherman, "An Adaptive Multi-Level Method for Elliptic Boundary Value Problems," *Computing*, 26 (1981), pp. 91-105.
14. R.E. Bank, A.H. Sherman, and A. Weiser, "Refinement Algorithms and Data Structures for Regular Local Mesh Refinement," *Mathematics and Computers in Simulation*, to appear.
15. M. Bieterman, J.E. Flaherty, and P.K. Moore, "Adaptive Refinement Methods for Non-Linear Parabolic Partial Differential Equations," Chap. 19 in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, I. Babuska, O.C. Zienkiewicz, J.R. Gago, and E.R. de A. Olivera, Eds., John Wiley and Sons, Chichester, 1986.
16. J.M. Coyle, J.E. Flaherty, and R. Ludwig, "On the Stability of Mesh Equidistribution Strategies for Time-Dependent Partial Differential Equations," *J. Comput. Phys.*, 62 (1986), pp. 26-39.

17. A.K. Kapila, *Asymptotic Treatment of Chemically Reacting Systems*, Pitman Advanced Publishing Program, Boston, 1983.
18. L.R. Petzold, "A Description of DASSL: A Differential/Algebraic System Solver," Sandia Report No. Sand. 82-8637, Sandia National Laboratory, Livermore, CA, 1982.
19. V. Thomee , "Negative Norm Estimates and Superconvergence in Galerkin Methods for Parabolic Problems," *Math. Comp.*, 34 (1980), pp. 93-113.
20. O.C. Zienkiewicz, *The Finite Element Method: Third Edition*, McGraw Hill, London, 1977.

TECHNICAL REPORT INTERNAL DISTRIBUTION LIST

	NO. OF COPIES
CHIEF, DEVELOPMENT ENGINEERING BRANCH	
ATTN: SMCAR-CCB-D	1
-DA	1
-DC	1
-DM	1
-DP	1
-DR	1
-DS (SYSTEMS)	1
CHIEF, ENGINEERING SUPPORT BRANCH	
ATTN: SMCAR-CCB-S	1
-SE	1
CHIEF, RESEARCH BRANCH	
ATTN: SMCAR-CCB-R	2
-R (ELLEN FOGARTY)	1
-RA	1
-RM	1
-RP	1
-RT	1
TECHNICAL LIBRARY	5
ATTN: SMCAR-CCB-TL	
TECHNICAL PUBLICATIONS & EDITING UNIT	2
ATTN: SMCAR-CCB-TL	
DIRECTOR, OPERATIONS DIRECTORATE	1
ATTN: SMCWV-OD	
DIRECTOR, PROCUREMENT DIRECTORATE	1
ATTN: SMCWV-PP	
DIRECTOR, PRODUCT ASSURANCE DIRECTORATE	1
ATTN: SMCWV-QA	

NOTE: PLEASE NOTIFY DIRECTOR, BENET WEAPONS LABORATORY, ATTN: SMCAR-CCB-TL,
OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST

	<u>NO. OF COPIES</u>		<u>NO. OF COPIES</u>
ASST SEC OF THE ARMY RESEARCH AND DEVELOPMENT ATTN: DEPT FOR SCI AND TECH THE PENTAGON WASHINGTON, D.C. 20310-0103	1	COMMANDER ROCK ISLAND ARSENAL ATTN: SMCRI-ENM ROCK ISLAND, IL 61299-5000	1
ADMINISTRATOR DEFENSE TECHNICAL INFO CENTER ATTN: DTIC-FDAC CAMERON STATION ALEXANDRIA, VA 22304-6145	12	DIRECTOR US ARMY INDUSTRIAL BASE ENGR ACTV ATTN: AMXIB-P ROCK ISLAND, IL 61299-7260	1
COMMANDER US ARMY ARDEC ATTN: SMCAR-AEE	1	COMMANDER US ARMY TANK-AUTMV R&D COMMAND ATTN: AMSTA-DDL (TECH LIB) WARREN, MI 48397-5000	1
SMCAR-AES, BLDG. 321	1	COMMANDER US MILITARY ACADEMY ATTN: DEPARTMENT OF MECHANICS WEST POINT, NY 10996-1792	1
SMCAR-AET-O, BLDG. 351N	1		
SMCAR-CC	1		
SMCAR-CCP-A	1		
SMCAR-FSA	1		
SMCAR-FSM-E	1	US ARMY MISSILE COMMAND REDSTONE SCIENTIFIC INFO CTR ATTN: DOCUMENTS SECT, BLDG. 4484 REDSTONE ARSENAL, AL 35898-5241	2
SMCAR-FSS-D, BLDG. 94	1		
SMCAR-MSI (STINFO)	2		
PICATINNY ARSENAL, NJ 07806-5000			
DIRECTOR US ARMY BALLISTIC RESEARCH LABORATORY ATTN: SLCBR-DD-T, BLDG. 305 ABERDEEN PROVING GROUND, MD 21005-5066	1	COMMANDER US ARMY FGN SCIENCE AND TECH CTR ATTN: DRXST-SD 220 7TH STREET, N.E. CHARLOTTESVILLE, VA 22901	1
DIRECTOR US ARMY MATERIEL SYSTEMS ANALYSIS ACTV ATTN: AMXSY-MP ABERDEEN PROVING GROUND, MD 21005-5071	1	COMMANDER US ARMY LABCOM MATERIALS TECHNOLOGY LAB ATTN: SLCMT-IML (TECH LIB) WATERTOWN, MA 02172-0001	2
COMMANDER HQ, AMCCOM ATTN: AMSMC-IMP-L ROCK ISLAND, IL 61299-6000	1		

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET WEAPONS LABORATORY, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.

TECHNICAL REPORT EXTERNAL DISTRIBUTION LIST (CONT'D)

	<u>NO. OF COPIES</u>		<u>NO. OF COPIES</u>
COMMANDER US ARMY LABCOM, ISA ATTN: SLCIS-IM-TL 2800 POWDER MILL ROAD ADELPHI, MD 20783-1145	1	COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MN EGLIN AFB, FL 32543-5434	1
COMMANDER US ARMY RESEARCH OFFICE ATTN: CHIEF, IPO P.O. BOX 12211 RESEARCH TRIANGLE PARK, NC 27709-2211	1	COMMANDER AIR FORCE ARMAMENT LABORATORY ATTN: AFATL/MNG EGLIN AFB, FL 32542-5000	1
DIRECTOR US NAVAL RESEARCH LAB ATTN: MATERIALS SCI & TECH DIVISION CODE 26-27 (DOC LIB) WASHINGTON, D.C. 20375	1 1	METALS AND CERAMICS INFO CTR BATTELLE COLUMBUS DIVISION 505 KING AVENUE COLUMBUS, OH 43201-2693	1

NOTE: PLEASE NOTIFY COMMANDER, ARMAMENT RESEARCH, DEVELOPMENT, AND ENGINEERING CENTER, US ARMY AMCCOM, ATTN: BENET WEAPONS LABORATORY, SMCAR-CCB-TL, WATERVLIET, NY 12189-4050, OF ANY ADDRESS CHANGES.

END
DATE
FILMED
JAN
1988