JOURNAL OF COMPUTATIONAL PHYSICS 62, 1–25 (1986)

# Two-Dimensional Fully Adaptive Solutions of Solid–Solid Alloying Reactions*

M. D. SMOOKE

*Department of Mechanical Engineering, Yale University, New Haven, Connecticut 06520*

AND

M. L. KOSZYKOWSKI

*Sandia National Laboratories, Livermore, California 94550*

Received January 17, 1984; revised September 10, 1984

Solid solid alloying reactions occur in a variety of pyrotechnical applications. They arise when a mixture of powders composed of appropriate oxidizing and reducing agents is heated. The large quantity of heat evolved produces a self-propagating reaction front that is often very narrow with sharp changes in both the temperature and the concentrations of the reacting species. Solution of problems of this type with an equispaced or mildly nonuniform grid can be extremely inefficient. In this paper we develop a two-dimensional fully adaptive method for solving problems of this class. The method adaptively adjusts the number of grid points needed to equidistribute a positive weight function over a given mesh interval in each direction at each time level. We monitor the solution from one time level to another to ensure that the local error per unit step associated with the time differencing method is below some specified tolerance. The method is applied to several examples involving exothermic, diffusion-controlled, self-propagating reactions in packed bed reactors. 

## 1. INTRODUCTION

Solid solid, diffusion-controlled alloying reactions are important in numerous pyrotechnical and metallurgical applications. Problems of this type arise when a suitable mixture of powders composed of oxidizing and reducing agents is heated. Due to the large quantity of heat evolved, a self-propagating reaction front can develop. In the unreacted zone ahead of the front there is essentially no physical change in the powders, while behind the front, the reaction is complete and the temperature is very high due to the released chemical energy. Although reactions of this type are highly exothermic, large initial quantities of energy are often required to initiate a self-propagating front. In addition, one component of the system often melts before the chemical heat release becomes large [1]. The increase in the reac-

* Work supported by DOE Office of Basic Energy Sciences.



0021-9991/86 $3.00

tion rate due to melting can be attributed to the larger liquid–solid as opposed to solid–solid diffusion rates. One-dimensional aspects of this problem, such as burn velocity and its dependence upon various parameters, have been reported previously in [2].

The model we develop is physically consistent with the above considerations and the developing reaction front is often very narrow with regions of high spatial activity (sharp peaks and steep fronts) in both the temperature and the species concentrations. These high-activity regions move as the front advances in time. In addition, the time scales in the problem can change by several orders of magnitude before and after the melting process. Before the melting occurs, the time rate of change of the temperature and the species concentrations is small since the process is controlled by thermal diffusion. After melting, the chemical term becomes dominant due to the large reaction rates and hence the time scale in the problem decreases substantially.

The complexity of the governing equations prevents exact analytical solutions from being obtained. Therefore, numerical methods must be used. Traditional finite difference methods attempt to solve problems of this type by using an equispace or mildly nonuniform grid held fixed for the entire calculation. Application of such methods in two dimensions can require an excessively large number of grid points to resolve accurately the regions of high spatial activity. As was the case in one-dimensional problems [2], we anticipate such an approach to be computationally inefficient when compared to methods that place grid points adaptively in such regions.

The equations for this problem can be case in the general form of two-dimensional mixed initial-boundary value problems.

$$\begin{aligned} u_t &= f(x, y, t, u, u_x, u_y, u_{xx}, u_{yy}), && (x, y) \in \Omega,\ t > 0, \\ g(x, y, t, u, u_x, u_y) &= 0, && \text{on } \partial\Omega,\ t > 0, \\ u(x, y, 0) &= r(x, y), && (x, y) \in \Omega, \end{aligned} \tag{1.1}$$

where $u$, $f$, $g$, and $r$ are $N$ vectors and the computational domain $\Omega$ is the rectangular region enclosed by $0 \leqslant x \leqslant 1$, $0 \leqslant y \leqslant 1$. The boundary of $\Omega$ is denoted by $\partial\Omega$. Problems similar in form to (1.1) can also arise in a variety of other physical problems.

In this paper we develop an algorithm that obtains fully adaptive (space and time) solutions to equations of the form (1.1). We generalize to two dimensions the fully adaptive algorithm developed in [2] for one-dimensional mixed initial-boundary value problems. By discretizing the time derivatives in (1.1) we obtain a nonlinear elliptic boundary value problem at each time level. We solve these problems by a finite difference procedure. Grid points are chosen adaptively such that positive weight functions in both the $x$ and $y$ directions are equidistributed over each subinterval. This variable node static rezone approach has the advantage of automatically adjusting the number of grid points as the numerical solution is

advanced in time. As in the one-dimensional case, however, there are problems in which the time evolution of the solution is such that we may not have to increase or decrease the number of grid points. In such cases we move a fixed number of nodes by extrapolating their positions from previous time levels. We also monitor the solutions of the boundary value problems from one time level to another to ensure that the local error per unit step associated with the time differencing method is below some specified tolerance. In the next section we present the solid–solid alloying model and in Section 3 we develop the fully adaptive algorithm. The method is applied to several example problems in Section 4.

## 2. Problem Formulation

We model solid–solid, diffusion-controlled alloying reactions by assuming we have a mixture of two powders, e.g., aluminum and palladium (for a discussion of this system, its parameters, and typical reactor design see Birnbaum [3, 4]). We further assume that at a given temperature one of the metals melts and coats the spherical particles of the second material. The unmelted material is assumed to be of uniform size and to have a radius $R$. The melted material then diffuses into the unmelted sphere at which point it reacts and releases chemical energy. The thermodynamic properties of the reacted and unreacted materials are assumed to remain constant over the temperature range considered. We model this process with a nonlinear energy equation coupled with a time-dependent source term. We do not allow for macroscopic mass transport. With these approximations the governing equations can be written (see, e.g., Booth [5]):

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + \Delta h \frac{\partial z}{\partial t}, \tag{2.1}$$

$$\frac{\partial z}{\partial t} = \frac{D \exp(-E/T)}{R^2} \cdot \frac{(1-z)^{1/3}}{1-(1-z)^{1/3}}, \tag{2.2}$$

where $T$ is the temperature and $z$ the fraction of the product formed. The initial conditions are given by

$$T(x, y, 0) = T_0(x, y), \qquad (x, y) \in \Omega, \tag{2.3}$$

$$z(x, y, 0) = z_0(x, y), \qquad (x, y) \in \Omega, \tag{2.4}$$

and the boundary conditions by

$$g_1(0, y, t, T) = 0, \qquad 0 \leqslant y \leqslant 1, \quad t > 0, \tag{2.5}$$

$$g_2(1, y, t, T) = 0, \qquad 0 \leqslant y \leqslant 1, \quad t > 0, \tag{2.6}$$

$$g_3(x, 0, t, T) = 0, \qquad 0 \leqslant x \leqslant 1, \quad t > 0, \tag{2.7}$$

$$g_4(x, 1, t, T) = 0, \qquad 0 \leqslant x \leqslant 1, \quad t > 0. \tag{2.8}$$

In the applications we consider $T_0$ and $z_0$ will be constant throughout the domain $\Omega$. In addition to the quantities already defined, $\Delta h$ denotes the heat of reaction, $k$ the thermal conductivity, $c$ the specific heat, $\rho$ the density, $D$ a reaction rate, and $E$ the activation energy.

Several other conceptually as well as functionally simple models for the source term have been suggested. Hardt and Phung [1] have discussed a model that views the reactor as being made up of alternating slabs of reactants. They assume the chemistry is fast so the source term is limited by linear diffusion in the slabs. The resulting expression is

$$\frac{\partial z}{\partial t} = \frac{D \exp(-E/T)}{w a_0^2 z}, \tag{2.9}$$

where $z$ is the fraction of the product formed, $D$ is a rate constant, $E$ an activation energy, and $wa_0^2$ is related to the slab thickness.

A second model has been discussed by Margolis [6]. He uses asymptotic methods to investigate instabilities of the planar solution. In this model the chemistry gives rise to the source term and is assumed to have the usual Arrhenius form. The general bimolecular Arrhenius expression can be written

$$\frac{\partial z}{\partial t} = k[A][B] \exp(-E/T), \tag{2.10}$$

where $[A]$ and $[B]$ are the reactant concentrations, $k$ is the rate constant, and $E$ is the activation energy.

We have chosen to focus on the model in (2.1)–(2.8) where the source term is limited by spherical diffusion. Although the third model has been successful in predicting experimentally observed pulsating modes of propagation, it does not address the effects of particle size. In addition, it is not applicable to a class of reactions that are known to be diffusion controlled. The first model does address the question of particle size and diffusion; however, we found it produced pulsating burn fronts for even low activation energies. We point out that care must be taken to assure that the rate terms in (2.2) and (2.9) remain bounded for $z = 0$. In practice, the initial value of $z_0$ is set equal to a small positive number. Furthermore, the rate expression in (2.9) is positive even after the reaction has finished and the burn fraction is equal to one. The expression in (2.2), however, approaches zero smoothly as the burn fraction approaches one.

## 3. Method of Solution

In this section we develop a method that obtains fully adaptive solutions of (1.1). We point out that a variety of approaches has been developed recently for solving time-dependent partial differential equations with adaptive spatial grids. In one

dimension, for example, there is the moving finite element work of Miller and Miller [7] and Gelinas, Doss, and Miller [8], the adaptive finite element method of Davis and Flaherty [9], the arclength approach of White [10], and the fully adaptive method of Smooke and Koszykowski [2]. Other methods have been considered by Bolstad [11], Dwyer, Kee, and Sanders [12], and Winkler, Norman, and Newman [13]. Much less work has been published for two-dimensional problems. We mention, in particular, the moving finite element work of Djomehri and Miller [14], the local mesh refinement method of Berger, Gropp, and Oliger [15] (see also Berger [16] and Gropp [17]) for hyperbolic problems, and the finite difference method by Dwyer, Smooke, and Kee [18].

These adaptive algorithms can be interpreted in terms of equidistributing a positive weight function over a given interval at each time level. (For an interesting discussion of these ideas see Herbst, Mitchell, and Schoombie [19].) The major differences in the methods center around the choice of the weight function and whether or not the mesh is coupled with the calculation of the solution. With the exception of the explicit time stepping methods of Bolstad in one dimension and of Berger, Gropp, and Oliger in two dimensions and the implicit method of Smooke and Koszykowski in one dimension, all of the adaptive methods mentioned above move a fixed number of grid points. While we believe that one does not have to couple the calculation of the solution components with the nodes to obtain accurate resolution of high-activity regions, we do believe that the adaptive mesh algorithm must be able to adjust automatically the number of nodes needed to maintain a specified degree of accuracy in the numerical solution. This is particularly important in unstable combustion problems.

Much of our development parallels that of [2]. We therefore abbreviate our discussion on those common features and focus our attention instead on those issues pertinent to the two-dimensional generalization. As was the case in [2], we are particularly interested in problems that occur in combustion. Such problems require the use of implicit time discretization methods. In the discussion that follows we develop the adaptive algorithm using a backward-Euler approximation to the time derivatives in (1.1). As we pointed out in [2], a higher-order backward differentiation formula, for example, could be used as well.

Our goal is to obtain a numerical solution of (1.1) at the time levels $0 = t^0 < t^1 < t^2 < \cdots < t^J = \mathscr{T}$, for some finite time $\mathscr{T}$. If for a function $g$ we define $g^n(x, y) = g(x, y, t^n)$, $n = 0, 1, \ldots, J$, then, upon replacing the time derivatives in (1.1) by a backward-Euler approximation and upon neglecting the time discretization error, we have the following semidiscrete approximation to (1.1),

$$f(x, y, t^{n+1}, \tilde{v}^{n+1}, \tilde{v}_x^{n+1}, \tilde{v}_y^{n+1}, \tilde{v}_{xx}^{n+1}, \tilde{v}_{yy}^{n+1}) - \frac{\tilde{v}^{n+1}}{\tau^{n+1}} = -\frac{\tilde{v}^n}{\tau^{n+1}}, \qquad (x, y) \in \Omega,$$

$$g(x, y, t^{n+1}, \tilde{v}^{n+1}, \tilde{v}_x^{n+1}, \tilde{v}_y^{n+1}) = 0, \qquad \text{on } \partial\Omega, \qquad (3.1)$$

$$\tilde{v}^0(x, y) = r(x, y),$$

$n=0, 1,..., J-1$, where the time step $\tau^{n+1}=t^{n+1}-t^n$. We point out that solution of the original mixed initial-boundary value problem is reduced to solving a nonlinear elliptic boundary value problem at each time level. We solve the boundary value problems by applying a finite difference method.

*Newton's Method*.

We want to obtain a discrete solution of (3.1) on the mesh $\mathscr{M}^{n+1}$ where the nodes are formed by the intersection of the lines

$$\mathscr{M}_x^{n+1}=\{0=x_0^{n+1}<x_1^{n+1}<\cdots<x_{M_x^{n+1}}^{n+1}=1\}, \tag{3.2}$$

and

$$\mathscr{M}_y^{n+1}=\{0=y_0^{n+1}<y_1^{n+1}<\cdots<y_{M_y^{n+1}}^{n+1}=1\}, \tag{3.3}$$

where $h_{x,i}^{n+1}=x_i^{n+1}-x_{i-1}^{n+1}$, $i=1, 2,..., \mathscr{M}_x^{n+1}$, and $h_{y,j}^{n+1}=y_j^{n+1}-y_{j-1}^{n+1}$, $j=1, 2,..., \mathscr{M}_y^{n+1}$. We include a superscript on the mesh since, in general, the number and/or the locations of the grid points in both the $x$ and $y$ directions can differ from one time level to another.

The spatial derivatives in (3.1) are approximated by finite differences. Omitting the time level superscript, we can write

$$\frac{\partial}{\partial x}\left(a(x, y)\frac{\partial g}{\partial x}\right)_{x_i, y_j} \approx \partial_x^2(a_{i,j}\, g_{i,j})$$

$$=\left(\frac{2}{x_{i+1}-x_{i-1}}\right)(a_{i+1/2,j}\partial_x g_{i+1,j}-a_{i-1/2,j}\partial_x g_{i,j}), \tag{3.4a}$$

and

$$\left(\frac{\partial g}{\partial x}\right)_{x_i, y_j} \approx \partial_x g_{i,j}, \tag{3.4b}$$

where

$$g_{i+1/2,j}=\frac{(g_{i+1,j}+g_{i,j})}{2}, \tag{3.5a}$$

$$\partial_x g_{i+1,j}=\frac{(g_{i+1,j}-g_{i,j})}{h_{x,i+1}}, \tag{3.5b}$$

and where we denote $g_{i,j}=g(x_i, y_j)$, $i=0, 1,..., M_x$, and $j=0, 1,..., M_y$. Similar expressions can be derived for derivatives with respect to $y$.

If the differential operators in (3.1) are replaced by expressions similar to those in (3.4)–(3.5), the problem of finding an analytic solution of (3.1) is converted into one of finding an approximation to it at each point of the mesh $\mathscr{M}^{n+1}$. If we denote

this approximation by $v_{i,j}^{n+1}$, we seek the solution $V^{n+1} = (v_{0,0}^{n+1}, v_{1,0}^{n+1}, \ldots, v_{M_x^{n+1}, M_y^{n+1}}^{n+1})^T$ of the system of nonlinear difference equations

$$\mathscr{F}(V^{n+1}) = F(V^{n+1}) - \left(\frac{V^{n+1} - V^n}{\tau^{n+1}}\right) = 0, \qquad n \geqslant 0. \tag{3.6}$$

The nonlinear equations in (3.6) can be solved by Newton's method. We write

$$\left(J(V_k^{n+1}) - \frac{I}{\tau^{n+1}}\right)(V_{k+1}^{n+1} - V_k^{n+1}) = -\lambda_k \mathscr{F}(V_k^{n+1}), \qquad n \geqslant 0,\ k = 0, 1, \ldots, \tag{3.7}$$

where $V_k^{n+1}$ denotes the $k$th solution iterate, $\lambda_k$ the $k$th damping parameter $(0 < \lambda \leqslant 1)$, $I$ the identity matrix, and $J(V_k^{n+1}) = \partial F(V_k^{n+1})/\partial V^{n+1}$ the steady-state Jacobian matrix. As in the one-dimensional problem, a linear set of equations is solved for corrections to the previous solution vector.

In a number of combustion problems we have found the cost of forming and then factoring the Jacobian matrix to be a significant part of the cost of a full Newton step (see, e.g., [21]). In such problems it is natural to apply the modified Newton method in which the Jacobian matrix is only periodically reevaluated. We employ the error estimate derived in [20] to determine whether the sequence of successive modified Newton iterates is converging at a fast enough rate. If the rate of convergence is too slow we revert to a full Newton method.

*Adaptive Gridding*

As in the one-dimensional case, if the boundary value problems in (3.1) admit solutions that exhibit a high degree of spatial activity, it is important for reasons of accuracy and computational efficiency that grid points be place in these regions. In two-dimensional problems the use of a fixed equispaced or mildly nonuniform grid can require an excessively large number of points to obtain a solution to a preset level of accuracy.

We determine the grid points of the two-dimensional mesh $\mathscr{M}^{n+1}$ by generalizing to two dimensions the concept of equidistributing a positive weight function over a given interval (see, e.g., [22]). In one-dimensional problems we say that a mesh $\mathscr{M}$ (with $M$ points) is equidistributed on the interval $0 \leqslant x \leqslant 1$ with respect to the nonnegative function $W$ and constant $C$ if

$$\int_{x_i}^{x_{i+1}} W\, dx = C, \qquad i = 0, 1, \ldots, M-1. \tag{3.8}$$

In two-dimensional problems we attempt to equidistribute the mesh $\mathscr{M}_x$ with respect to the nonnegative function $W_x$ and constant $C_x$ for each of the $M_y + 1$ horizontal grid lines. We write

$$\int_{x_i}^{x_{i+1}} W_x\, dx = C_x, \qquad i = 0, 1, \ldots, M_x - 1, \tag{3.9}$$

for $j=0, 1,..., M_y$. Similarly, we attempt to equidistribute the mesh $\mathcal{M}_y$ with respect to $W_y$ and $C_y$ for each of the $M_x+1$ vertical grid lines. We have

$$\int_{y_j}^{y_{j+1}} W_y\, dy = C_y, \qquad j=0, 1,..., M_y-1, \tag{3.10}$$

for $i=0, 1,..., M_x$. The choice of the weight functions is somewhat flexible. They should be chosen, however, such that grid points are placed in regions of high spatial activity with the goal of reducing the local spatial discretization error (and hopefully the global error). We equidistribute the difference in the components of the solution and its gradient between adjacent mesh points. Upon denoting the vector $\tilde{v}=[\tilde{v}_1, \tilde{v}_2,..., \tilde{v}_N]^T$, we seek to obtain a mesh $\mathcal{M}_x$ such that

$$\int_{x_i}^{x_{i+1}} \left|\frac{d\tilde{v}_k}{dx}\right| dx \leqslant \alpha\, |\max_{\Omega} \tilde{v}_k - \min_{\Omega} \tilde{v}_k|, \qquad \begin{matrix} i=0, 1,..., M_x-1, \\ k=1, 2,..., N, \end{matrix} \tag{3.11}$$

and

$$\int_{x_i}^{x_{i+1}} \left|\frac{d^2\tilde{v}_k}{dx^2}\right| dx \leqslant \beta \left|\max_{\Omega} \frac{d\tilde{v}_k}{dx} - \min_{\Omega} \frac{d\tilde{v}_k}{dx}\right|, \qquad \begin{matrix} i=1, 2,..., M_x-1, \\ k=1, 2,..., N, \end{matrix} \tag{3.12}$$

for each of the $M_y+1$ horizontal grid lines, and we seek to determine a mesh $\mathcal{M}_y$ such that

$$\int_{y_j}^{y_{j+1}} \left|\frac{d\tilde{v}_k}{dy}\right| dy \leqslant \gamma\, |\max_{\Omega} \tilde{v}_k - \min_{\Omega} \tilde{v}_k|, \qquad \begin{matrix} j=0, 1,..., M_y-1, \\ k=1, 2,..., N, \end{matrix} \tag{3.13}$$

and

$$\int_{y_j}^{y_{j+1}} \left|\frac{d^2\tilde{v}_k}{dy^2}\right| dy \leqslant \delta \left|\max_{\Omega} \frac{d\tilde{v}_k}{dy} - \min_{\Omega} \frac{d\tilde{v}_k}{dy}\right|, \qquad \begin{matrix} j=1, 2,..., M_y-1, \\ k=1, 2,..., N, \end{matrix} \tag{3.14}$$

for each of the $M_x+1$ vertical grid lines. The quantities $\alpha$, $\beta$, $\gamma$, and $\delta$ are small constants less than one to be chosen by the user.

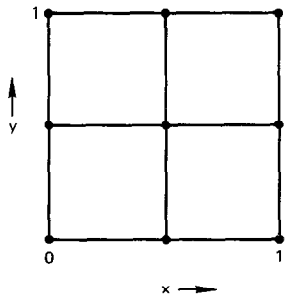


FIG. 1. Example of a two-dimensional grid.

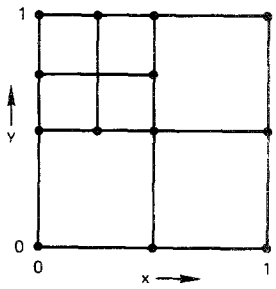


FIG. 2. Local refinement of the two-dimensional grid.

In implementing the adaptive mesh algorithm we first solve the boundary value problems in (3.1) on a coarse mesh. The maximum and minimum values of $\tilde{v}_k$, $d\tilde{v}_k/dx$, and $d\tilde{v}_k/dy$ are obtained and we test the inequalities in (3.11) and (3.12) one $x$ subinterval at a time for all the "j" grid lines. If either of the inequalities is not satisfied a grid point is inserted at the midpoint of the $x$ subinterval in question for $j=0, 1,..., M_y$. Once this procedure has been carried out in the $x$ direction we check the inequalities in (3.13) and (3.14) one $y$ subinterval at a time for all "i" grid lines. If either of the inequalities is not satisfied a grid point is inserted at the midpoint of the subinterval in question for $i=0, 1,..., M_x$. For example, suppose we have a two-dimensional mesh as in Fig. 1. Suppose further that the inequalities in (3.11)–(3.14) imply that the mesh should be refined as in Fig. 2. Since we add grid points globally in each direction, the mesh we actually obtain is illustrated in Fig. 3. We observe that each grid line is connected to two sides of the region $\Omega$. While there can be a loss of efficiency due to the introduction of "unwanted" nodes, this type of grid refinement algorithm does not require a sophisticated data structure and it enables the Jacobian matrix in (3.7) to be evaluated efficiently on vector machines (see below).

An equidistribution procedure such as that in (3.9) and (3.10) may produce a mesh that is not smoothly varying. We have found that this can affect the convergence properties of the method as well as its accuracy. As in the one-dimensional case, we also require that the mesh be locally bounded. We require that the ratio of

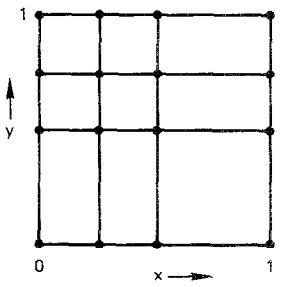


FIG. 3. Global refinement of the two-dimensional grid.

adjacent mesh intervals in the $x$ and $y$ directions be bounded above and below by constants, i.e.,

$$\frac{1}{A_x} \leqslant \frac{h_{x,i}}{h_{x,i-1}} \leqslant A_x, \qquad i = 2, 3, \ldots, M_x, \tag{3.15}$$

and

$$\frac{1}{A_y} \leqslant \frac{h_{y,j}}{h_{y,j-1}} \leqslant A_y, \qquad j = 2, 3, \ldots, M_y, \tag{3.16}$$

where $A_x$ and $A_y$ are constants $\geqslant 1$. Values of $A_x$ and $A_y$ close to one produce very dense quasiuniform grids in each direction. Large values of these contants often produce convergence difficulties. Based upon our numerical experiments, we have found values of $A_x$ and $A_y$ between two and six to be adequate. In the calculations described in Section 4, $A_x$ and $A_y$ were set equal to four. Such a two-dimensional buffering tends to smooth out rapid changes in the size of the mesh intervals. Once the equidistribution conditions are checked, we determine whether grid points must be inserted to satisfy (3.15) and (3.16). Once (3.11)–(3.14) and (3.15)–(3.16) are satisfied, the previously converged numerical solution is interpolated onto the new mesh and the result serves as an initial solution estimate for Newton's method on this finer grid. The process is continued until the inequalities are satisfied.

*Interpolation*

The boundary value problems in (3.2) are inhomogeneous. Their solution requires a knowledge of the solution of the boundary value problem from the previous time level. If the same spatial grid were used for every time level, then this solution information would be available. The number and the locations of the grid points would not change as the calculation was advanced in time. In our case, however, we may have to determine adaptively the mesh every time the boundary value problems are solved. As a result, we anticipate that the number as well as the locations of the grid points may change from one time level to another. When a grid point at time level $n+1$ does not have a point whose location is identical to a point at time level $n$, we form $v_{i,j}^n$ by interpolation. Suppose that the relation of the point $P = (x_i^{n+1}, y_j^{n+1})$ to grid points at the $n$th level is as illustrated in Fig. 4. We assume that the $x$ and $y$ coordinates of $P$ can be bounded above and below by points from the $n$th level, that is, we assume

$$x_k^n \leqslant x_i^{n+1} \leqslant x_{k+1}^n, \tag{3.17}$$

and

$$y_l^n \leqslant y_j^{n+1} \leqslant y_{l+1}^n. \tag{3.18}$$

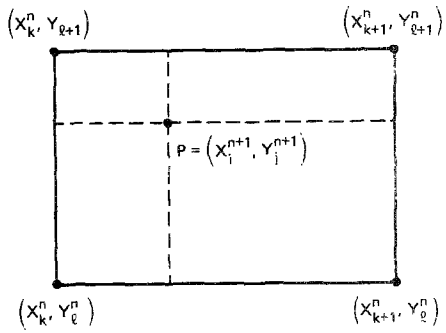


FIG. 4. Illustration of grid points at time levels $n$ and $n+1$. The physical location of point $P$ does not coincide with points at level $n$.

We obtain $v_{i,j}^n$ by forming

$$v_{i,j}^n = v_{k,l}^n + \left(\frac{v_{k+1,l}^n - v_{k,l}^n}{x_{k+1}^n - x_k^n}\right)(x_i^{n+1} - x_k^n) + \left(\frac{v_{k,l+1}^n - v_{k,l}^n}{y_{l+1}^n - y_l^n}\right)(y_j^{n+1} - y_l^n)$$
$$+ \left[\frac{\left(\frac{v_{k+1,l+1}^n - v_{k,l+1}^n}{x_{k+1}^n - x_k^n}\right) - \left(\frac{v_{k+1,l}^n - v_{k,l}^n}{x_{k+1}^n - x_k^n}\right)}{y_{l+1}^n - y_l^n}\right](x_i^{n+1} - x_k^n)(y_j^{n+1} - y_l^n). \tag{3.19}$$

As in the one-dimensional problem, an additional spatial discretization error is introduced into our algorithm as a result of the interpolation procedure. In particular, for a function $g(x, y, t)$ that is three times continuously differentiable in the domain $(\Omega \times [O, T])$, we can write

$$\frac{g_{i,j}^{n+1} - g_{i,j}^n}{\tau^{n+1}} = \left.\frac{\partial g_{i,j}}{\partial t}\right|_{t^{n+1}}$$
$$+ O\left(\frac{\tau^{n+1}}{2} + \frac{h_{x,k+1}^n (x_i^{n+1} - x_k^n)}{\tau^{n+1}} + \frac{h_{y,l+1}^n (y_j^{n+1} - y_l^n)}{\tau^{n+1}}\right), \tag{3.20}$$

where $g_{i,j}^n$ is similar in form to (3.19). The discretization error associated with the time derivative term has an additional error $O(h_{x,k+1}^n (x_i^{n+1} - x_k^n)/\tau^{n+1} + h_{y,l+1}^n (y_j^{n+1} - y_l^n)/\tau^{n+1})$ that modifies the way in which the adaptive time steps are chosen (see below).

*Grid Propagation*

By solving the boundary value problems adaptively, we have the ability to increase or decrease the number of grid points required to equidistribute the weight functions at every time level. As our one-dimensional results indicate, this is impor-

tant in unstable burn front problems [2]. The method is not without cost, however; since the time differencing method is implicit, we must solve a system of nonlinear equations at each time level. The use of Newton's method implies that a Jacobian matrix must be formed and the Newton equations solved at each time level. If at some time during the calculation the solution can be determined accurately with a fixed number of (possibly moving) grid points in each direction (e.g., a stable front), then the Jacobian can be held fixed; its factorization can be stored and the cost of propagating the solution will be less than if the variable node approach had been used.

Most of the adaptive methods that have been used to solve both one- and two-dimensional time-dependent problems move a fixed number of grid points. In our approach we begin the calculation by applying a variable node static rezone method—the boundary value problems are solved adaptively at each time level. Once the number of nodes required to satisfy (3.11)–(3.16) remains constant in each direction for several time levels, we predict the locations of the grid points by solving

$$\frac{dx_i}{dt} = \frac{x_i^n - x_i^{n-1}}{\tau^n}, \qquad i = 1, 2, \ldots, M_x - 1, \tag{3.21}$$

$$x_i(0) = x_i^{n-1},$$

and

$$\frac{dy_j}{dt} = \frac{y_j^n - y_j^{n-1}}{\tau^n}, \qquad j = 1, 2, \ldots, M_y - 1, \tag{3.22}$$

$$y_j(0) = y_j^{n-1}.$$

These equations can be integrated to yield

$$x_i(t) = \left(\frac{x_i^n - x_i^{n-1}}{\tau^n}\right) t + x_i^{n-1}, \qquad i = 1, 2, \ldots, M_x - 1, \tag{3.23}$$

and

$$y_j(t) = \left(\frac{y_j^n - y_j^{n-1}}{\tau^n}\right) t + y_j^{n-1}, \qquad j = 1, 2, \ldots, M_y - 1. \tag{3.24}$$

Both the $x$ and $y$ nodes are extrapolated linearly from their positions at time levels $n$ and $n-1$. If the extrapolation procedure moves the nodes out of the region of high spatial activity, the buffering of the mesh produced by (3.15) and (3.16) absorbs some of the inaccuracy of the grid points' locations. Nevertheless, it is still very important to recheck the equidistribution conditions in (3.11)–(3.14) to make certain that the extrapolation has followed the regions of high spatial activity and that no new nodes have to be added. The potential problems of node tangling and

extrapolation out of the computational domain are handled in a manner analogous to the approaches used in the one-dimensional case (see [2]).

*Time Stepping*

Our objective is to choose the time steps $\tau^n$, $n = 1, 2, \ldots, J$, such that the local error per unit step associated with the time differencing scheme in (3.1) is below some specified value $\varepsilon$. The local error (l.e.) at the point $P = (x_i^{n+1}, y_j^{n+1})$ can be estimated by

$$\text{l.e.} \approx a \frac{(\tau^{n+1})^2}{2} + 2bh^n_{x,k+1}(x_i^{n+1} - x_k^n) + 2ch^n_{y,l+1}(y_j^{n+1} - y_l^n), \tag{3.25}$$

where $a$, $b$, and $c$ are related to bounds on the quantities $\partial^2 u/\partial t^2$, $\partial^2 u/\partial x^2$, and $\partial^2 u/\partial y^2$, respectively, in the rectangular region illustrated in Fig. 4. In practice these partial derivatives are evaluated by finite differences.

At the point $P$ we want to determine the value of $\tau^{n+1}$ such that

$$\text{l.e.} \leqslant \theta \varepsilon \tau^{n+1}, \tag{3.26}$$

where, as in the one-dimensional problem, $\theta$ is a safety factor designed to account for the fact that the expression in (3.25) is only approximate. We determine the proper value of the time step by following an argument similar to the one developed in [2]. If we denote the last accepted time step by $\tau_{\text{old}}$, the step we want to calculate by $\tau_{\text{new}}$, and the contribution to the local error arising from the interpolation process by $\varepsilon_{\text{int}}$, then we can write

$$\frac{\alpha(\tau_{\text{new}}^{n+1})^2}{2} \leqslant \theta \varepsilon \tau_{\text{new}}^{n+1} - \varepsilon_{\text{int}}, \tag{3.27}$$

and

$$\frac{\alpha(\tau_{\text{old}}^{n+1})^2}{2} \leqslant \text{l.e.} - \varepsilon_{\text{int}}. \tag{3.28}$$

If we eliminate $\alpha$ and then solve the resulting expression for the largest root of $\tau_{\text{new}}^{n+1}$, we find

$$\tau_{\text{new}}^{n+1} \leqslant \frac{\dfrac{(\tau_{\text{old}}^{n+1})^2 \theta \varepsilon}{\text{l.e.} - \varepsilon_{\text{int}}} + \sqrt{\dfrac{(\tau_{\text{old}}^{n+1})^4 \theta^2 \varepsilon^2}{(\text{l.e.} - \varepsilon_{\text{int}})^2} - \dfrac{4(\tau_{\text{old}}^{n+1})^2 \varepsilon_{\text{int}}}{(\text{l.e.} - \varepsilon_{\text{int}})}}}{2}, \tag{3.29}$$

where, to ensure that the discriminant is positive,

$$\varepsilon \geqslant \frac{\sqrt{4\varepsilon_{\text{int}}(\text{l.e.} - \varepsilon_{\text{int}})}}{\tau_{\text{old}}^{n+1}\theta}. \tag{3.30}$$

We carry out this procedure for each of the $(M_x^{n+1}-1)(M_y^{n+1}-1)$ interior nodes and choose $\tau^{n+1}$ as the smallest calculated value.

As was the case in the one-dimensional problem, if $\varepsilon_{\text{int}}=0$, then the expression for $\tau_{\text{new}}^{n+1}$ reduces to

$$\tau_{\text{new}}^{n+1} \leqslant \frac{(\tau_{\text{old}}^{n+1})^2\,\theta\varepsilon}{\text{l.e.}}, \tag{3.31}$$

as one would expect for the backward-Euler method. The restriction on $\varepsilon$ arises from the fact that part of the local error comes from the interpolation process. In affect, the expression in (3.30) implies that the interpolation errors cannot be too large for a given value of $\varepsilon$. If the expression in (3.30) is not satisfied we must reduce the interpolation error—for example, by increasing the number of spatial nodes. The above analysis can be generalized to accommodate higher-order interpolation and time discretization methods.

*Computational Considerations*

Selection of the damping parameter $\lambda_k$ has been studied in depth by Deuflhard [23]. While we have implemented a variation of his method for nonsingular Jacobians, we almost always take full Newton steps. We terminate the Newton iteration when

$$\|\Delta V_{k+1}^{n+1}\|_2 = \|V_{k+1}^{n+1} - V_k^{n+1}\|_2 \leqslant \text{TOL}\,\sqrt{NM_x^{n+1}M_y^{n+1}}, \qquad k=0,\,1,\ldots, \tag{3.32}$$

where we typically take $\text{TOL} \leqslant 10^{-5}$.

The converged numerical solution at the $n$th time level provides an excellent initial approximation to the solution at level $n+1$. We implement this approach when the number of grid points remains fixed from one time level to another. However, in a two-dimensional calculation, such an approach can increase rapidly the number of grid points when used with a variable node static rezone technique. The converged solution at level $n+1$ would contain all the nodes from the $n$th level plus those added at the current level. To avoid this possibility, we generally use one-half the number of grid points from the $n$th level as our starting mesh for time level $n+1$. This procedure restricts the unwanted growth in the number of nodes during the variable node part of a calculation.

The initial time step used at the current time level is the last value calculated from (3.29). If using this step we are unable to obtain convergence in our Newton iteration, we halve the time step and restart the calculation at the current time level. If Newton's method converges, we calculate a new value of the time step using (3.29). If we find that the calculated step is smaller than the value we have used, we redo the calculation at the current time with the new value of $\tau^{n+1}$. Otherwise, we proceed to the next level and continue the process with the calculated value from (3.29).

*Numerical Jacobian*

The Jacobian matrix can be written in block pentadiagonal form. We have found that in a number of combustion problems (see, e.g., [21]) it is often more efficient to evaluate the Jacobians by finite difference procedures as opposed to analytical means. The method we implement generalizes ideas outlined by Curtis, Powell, and Reid [24]. The idea is to form several columns of the Jacobian simultaneously using vector function evaluations and the Jacobian's given sparsity structure. We point out that, while the Jacobian can be assembled one node at a time, such a procedure is not as efficient computationally when implemented on a vector machine as forming the Jacobian with vector function evaluations (the total number of component function evaluations being the same in both cases). As a result, even though the global grid refinement strategy adds extra points compared to a local refinement procedure, some of the efficiency loss is made up by the fact that the Jacobian formation can be vectorized. A variable node local grid refinement strategy, however, may ultimately be the method of choice for this class of problems.

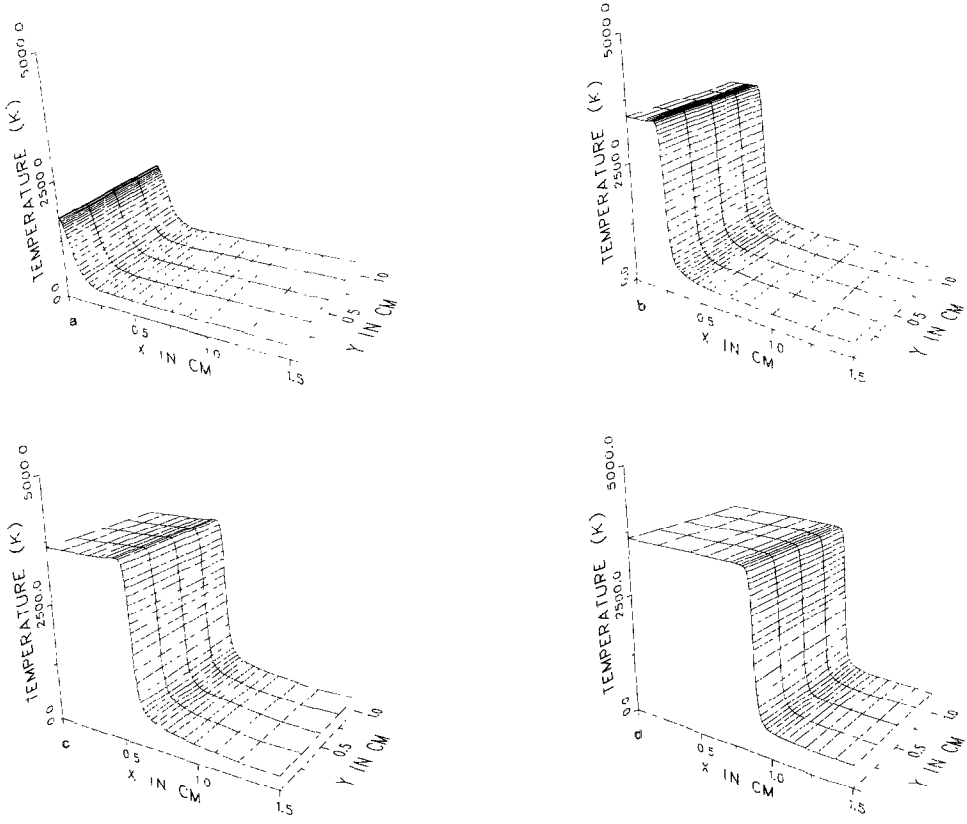


FIG. 5. Temperature profiles for test Problem 1. The burn front is quasi-one-dimensional.

If to each column of the Jacobian we associate the $i$ and $j$ values of the node corresponding to the column's diagonal block, then all columns of the Jacobian having the same value of the parameter

$$\alpha = (i + 3j) \bmod(5), \tag{3.33}$$

can be evaluated simultaneously. Ideas along these lines have been explored in a more abstract setting by Newsam and Ramsdell [25] and Coleman and More [26]. Once the Jacobian is formed, we solve the Newton equations with a block-line SOR method.

## 4. Numerical Results.

In this section we illustrate application of the method in the solution of three solid–solid alloying problems. We apply the variable node static rezone method with grid extrapolation to an aluminum–palladium system. While a detailed numerical study similar to the one performed in [2] should ultimately be undertaken to compare the merits of various two-dimensional adaptive algorithms, this is not our intention here. Our goal instead is to illustrate the effectiveness of our adaptive algorithm in such problems.

We consider three problems. All of the calculations are performed on a reactor of length 1.5 cm and width 1 cm (in the calculations, however, the lengths of both

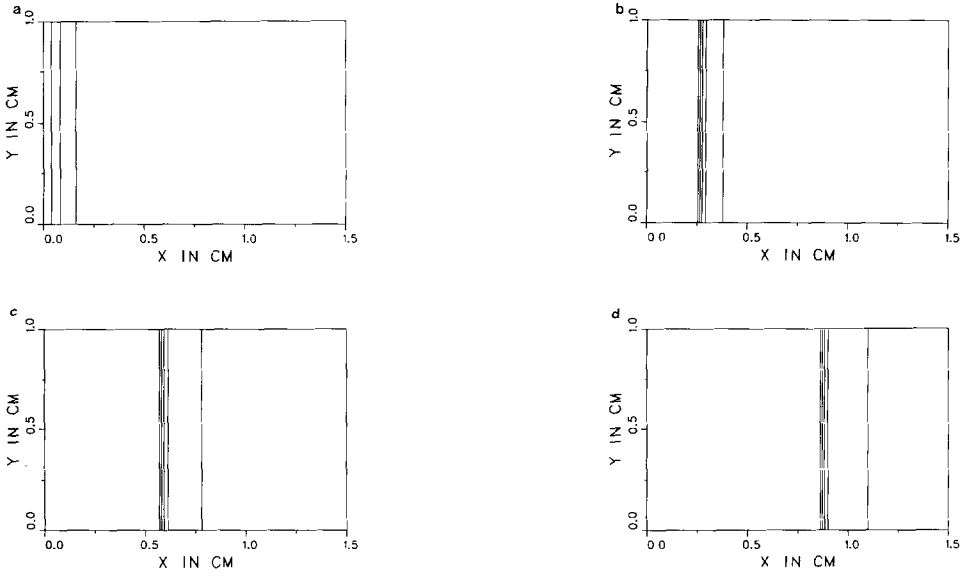


Fig. 6. Isotherms for test Problem 1. The isotherms increase from right to left in 500 K increments (500, 1000,..., 2500 K).

sides and the temperature are scaled between 0 and 1). The first is a quasi-one-dimensional problem in which the temperature of one of the reactor walls is linearly ramped up to 2000K. Since there is no heat loss in the $y$ direction, the problem is effectively one dimensional. In the second problem one of the reactor walls is initially heated with a Gaussian temperature ramp. In one case not enough heat is supplied and the reaction is eventually extinguished. When the peak input temperature is doubled, a symmetric two-dimensional burn front develops. Heat is also removed from the system along the walls of the reactor. Finally, we consider a reactor in which the distribution of particles is nonuniform. The smallest particles are located along the top and bottom walls of the reactor. As a result, the front propagates fastest down these walls. The values of the rate and transport parameters used in the calculations were obtained from Birnbaum [3, 4]. In particular, we set $k/\rho c = 2.5 \times 10^{-5}$, $\Delta H/\rho c = 2.8 \times 10^{-3}$, $D/R^2 = 5.8 \times 10^{3}$, and $E = 1.1 \times 10^{4}$. The adaptive grid quantities ($\alpha$, $\beta$, $\gamma$, and $\delta$) were set equal to 0.1, which when combined with a time tolerance of $10^{-3}$ assured us of three places of accuracy in the velocity of propagation. All of the calculations were performed on a CRAY-1S computer.

PROBLEM 1. The first example we consider is a uniform packed reactor ignited along the wall $y = 0$. The initial conditions for the problem are given by

$$T(x, y, 0) = 300, \tag{4.1}$$

$$z(x, y, 0) = 10^{-3}, \tag{4.2}$$

and the boundary conditions for $t \leqslant 0.05$ by

$$T(0, y, t) = 3.4 \times 10^{4} t + 300, \qquad 0 \leqslant y \leqslant 1, \tag{4.3}$$

$$\frac{\partial T}{\partial x}(1.5, y, t) = 0, \qquad 0 \leqslant y \leqslant 1, \tag{4.4}$$

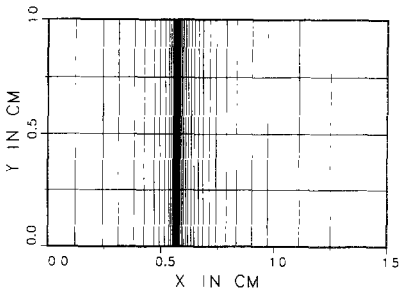


FIG. 7. A typical two-dimensional adaptive grid for test Problem 1.

$$\frac{\partial T}{\partial y}(x, 0, t) = 0, \qquad 0 \leqslant x \leqslant 1.5, \tag{4.5}$$

$$\frac{\partial T}{\partial y}(x, 1, t) = 0, \qquad 0 \leqslant x \leqslant 1.5. \tag{4.6}$$

For $t > 0.05$ the boundary conditions remain the same except for the heated wall. In this case we have

$$\frac{\partial T}{\partial x}(0, y, t) = 0, \qquad 0 \leqslant y \leqslant 1. \tag{4.7}$$

Since there are no initial variations in the $y$ direction and since no heat is either added or removed in the $y$ direction, the problem is effectively one-dimensional. We use it, however, to illustrate the computational method and to test it against an

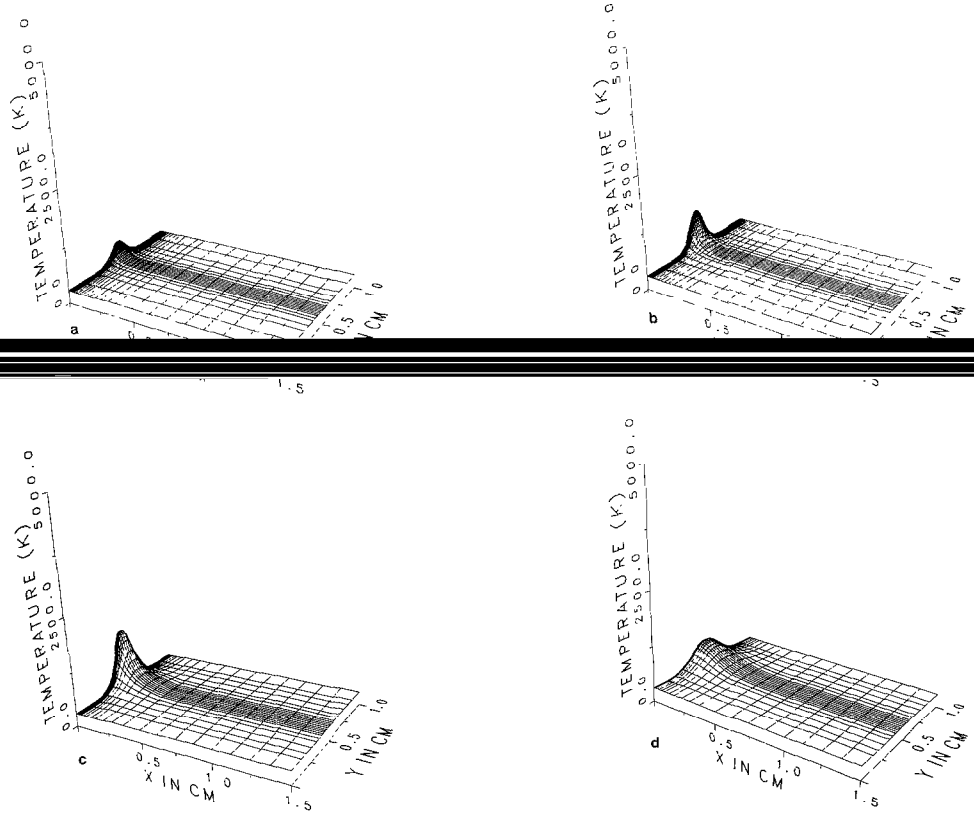


FIG. 8. Temperature profiles for test Problem 2a. Not enough heat has been supplied to the reactor and the reaction goes out.

equispaced calculation. The calculation is begun on a grid with 5 equispaced nodes in both the $x$ and $y$ directions. As the integration proceeds, nodes are added or removed as dictated by the spatial error tolerances. The maximum number of nodes used in the $x$ direction was 49 and in the $y$ direction 5. The minimum grid spacing in the $x$ direction was 0.003 cm, which implies that 500 equispaced $x$ nodes would be required to obtain a solution with comparable accuracy. An equally spaced calculation was run using only 100 $x$ nodes and the burn front failed to propagate. In fact, we were unable to obtain a self-propagating front until several hundred nodes were used. We have found (see also [2]) that, when too coarse a spatial grid is used in problems of this type, heat is often diffused away faster than it is produced. As a result, the reaction is extinguished.

In Fig. 5 we illustrate a sequence of temperature profiles obtained with the adaptive calculation. The profiles show the bed being heated and the subsequent propagation of the burn front. The resolution of the burn front with the adaptive

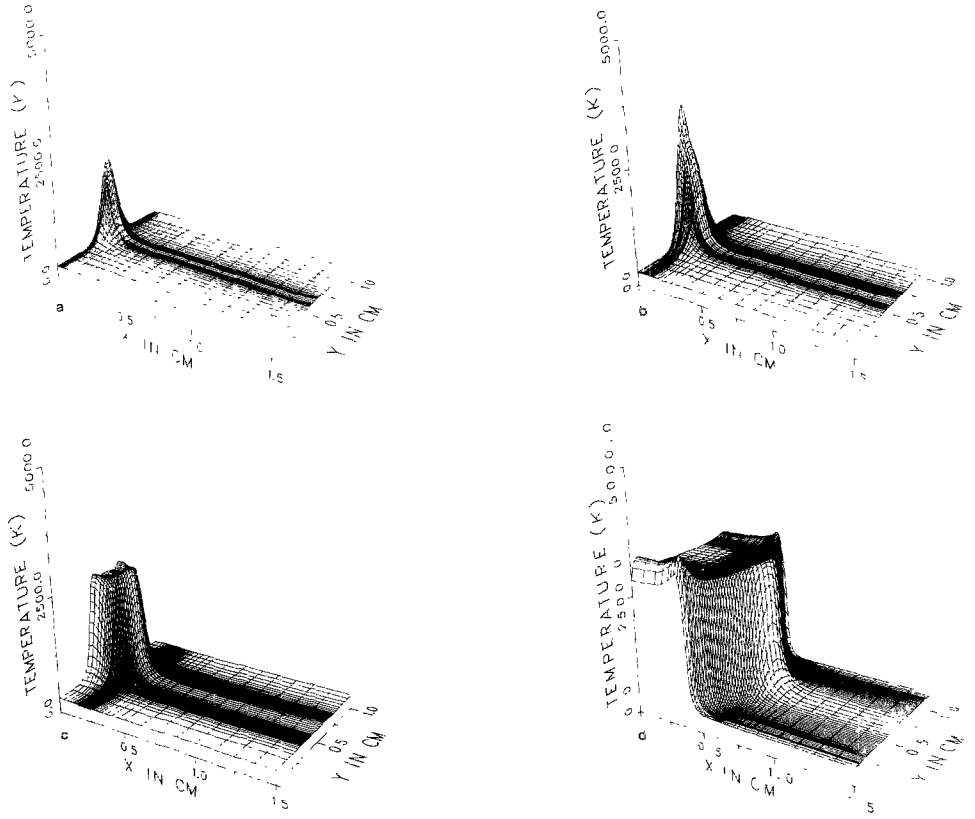


FIG. 9. Temperature profiles for test Problem 2b. After the source term is turned off, a symmetric, two-dimensional burn front develops.

grid is evident. In Fig. 6 we illustrate the isotherms corresponding to the profiles in Fig. 5. The contours increase from right to left in 500K increments (500, 1000,..., 2500K). Finally, in Fig. 7 we show a typical grid for this calculation.

Problem 2. *Case a.* The second problem we consider involves the initiation of the powders with a Gaussian heat pulse in the center of the left wall ($y=0$). The initial conditions for this problem are identical to those in Problem 1 but the boundary conditions for $t \leqslant 0.05$ are

$$T(0, y, t) = [(3.4 \times 10^4) \exp(-10^6(y-0.5)^2)]\, t + 300, \qquad 0 \leqslant y \leqslant 1, \tag{4.8}$$

$$\frac{\partial T}{\partial x}(1.5, y, t) = 0, \qquad 0 \leqslant y \leqslant 1, \tag{4.9}$$

$$\frac{\partial T}{\partial y}(x, 0, t) = 500(T-300), \qquad 0 \leqslant x \leqslant 1.5, \tag{4.10}$$

$$\frac{\partial T}{\partial y}(x, 1, t) = -500(T-300), \qquad 0 \leqslant x \leqslant 1.5. \tag{4.11}$$

For $t > 0.05$ the boundary conditions remain the same except for the heated wall. In this case we have

$$\frac{\partial T}{\partial x}(0, y, t) = 0, \qquad 0 \leqslant y \leqslant 1. \tag{4.12}$$

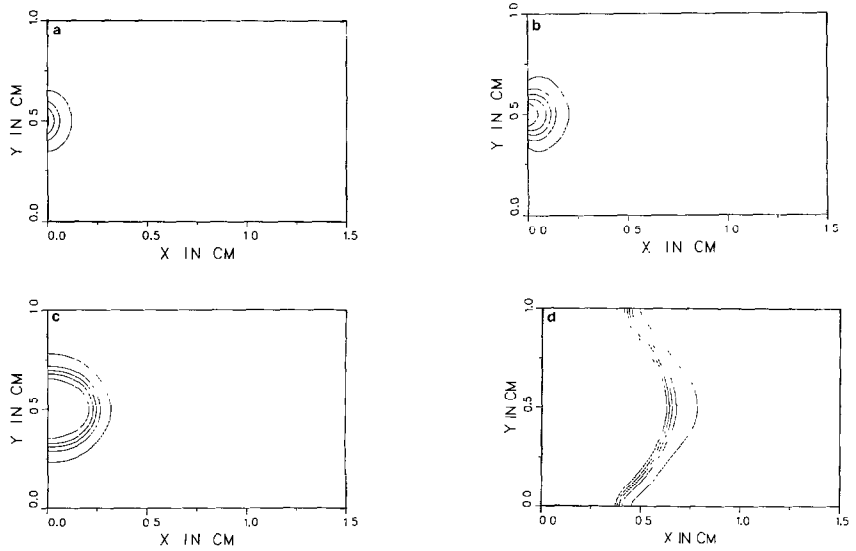


FIG. 10. Isotherms for test Problem 2b. The isotherms increase from right to left in 500K increments (500, 1000,..., 2500K).

In this problem the center of the left wall is heated up most rapidly. Once the temperature reaches its peak value, the heat source is turned off and the wall becomes insulated. The value of the heat loss in (4.10) and (4.11) was chosen to assure wall temperatures several hundred degrees cooler than the interior of the bed. In practice, the heat loss parameters would be determined from the reactor material and the environment surrounding the reactor. In Fig. 8 we illustrate the temperature profiles for this set of initial and boundary conditions. After the initial heat pulse, the source is turned off and the reaction fails to propagate; the heat pulse simply diffuses into the reactor. It is worthwhile to point out that initiation can be a sensitive probe into the kinetics of pyrotechnic processes. We anticipate that this type of calculation combined with detailed experiments could prove to be a useful tool in such situations.

*Case b.* We next doubled the peak temperature of the heat source. In this case the reaction began to propagate. The resulting temperature profiles are illustrated in Fig. 9. The temperature at the left wall increases until the peak temperature is reached. The heat source is then turned off and, as heat begins to diffuse in both the $x$ and $y$ directions, a symmetric two-dimensional burn front develops. As the front propagates down the reactor, it eventually reaches the top and bottom walls ($y=0$ and $y=1$). We then obtain a quasiplanar burn front. The temperature profiles in Fig. 9 clearly illustrate the narrow reaction region and the ability of the adaptive grid to resolve the front. The corresponding temperature contours are illustrated in Fig. 10. Again, they increase from right to left in 500K increments. A sample grid is shown in Fig. 11.

The minimum numbers of nodes used in the $x$ and $y$ directions were 13 and 9, respectively. The maximum number was 63 for the $x$ direction and 99 for the $y$ direction. In addition, the minimum mesh spacings were 0.005 and 0.006 for the $x$ and $y$ directions, respectively. This implies that to obtain similar results with an equispaced grid we would need well over 40,000 grid points. We performed a calculation with 10,000 (the limit of our capacity) equispaced nodes and found that the burn front did not propagate.

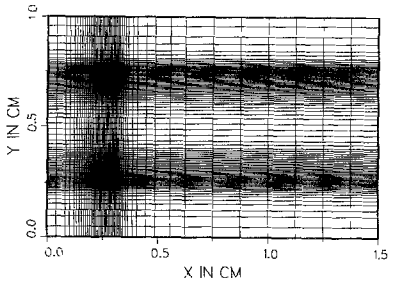


FIG. 11. A typical two-dimensional adaptive grid for test Problem 2b.

PROBLEM 3. The final example we consider has a reactor bed in which the distribution of particle sizes is nonuniform. We use the same initial and boundary conditions as in Problem 1. The central part of the bed ($0.2 \leqslant y \leqslant 0.8$) is packed with the largest particles. They are then linearly ramped down by a factor of four at both walls ($y=0$ and $y=1$). The density and thermal conductivity are held constant based upon the results of Dletz [27]. Since the source term is proportional to $1/R^2$, the sections of the reactor with the smallest particles ignite first and the reaction starts to propagate along the reactor's top and bottom walls.

The calculation was performed with an initial grid consisting of 10 equispaced points in both the $x$ and $y$ directions. In Fig. 12 we illustrate the temperature profiles. We see the early initiation of the reaction near the two walls. In addition, before the source term has been turned off (the temperature in the center of the bed

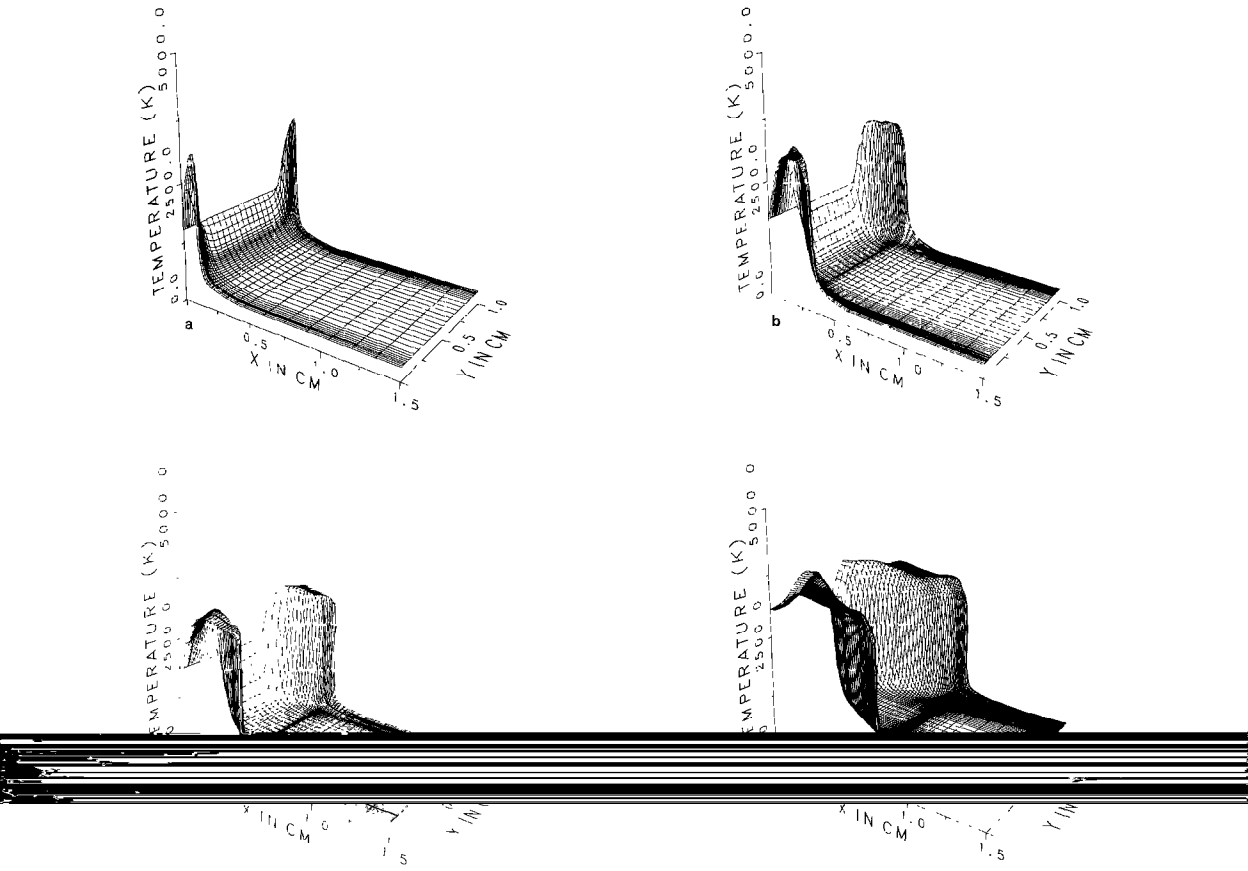


FIG. 12. Temperature profiles for test Problem 3. The reactor has a nonuniform distribution of particle sizes.

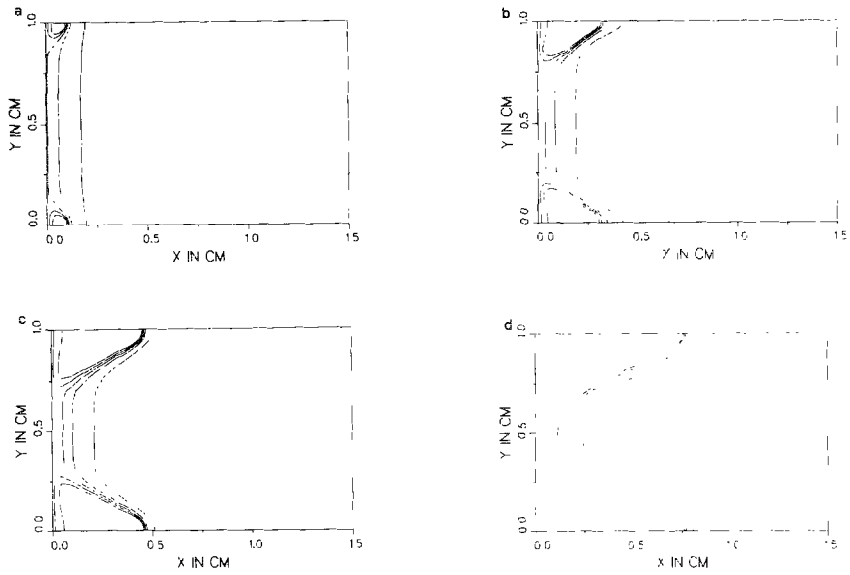


FIG. 13. Isotherms for test Problem 3. The isotherms increase from right to left in 500 K increments (500, 1000,.... 2500 K).

is still rising), the front has propagated a considerable distance down the sides of the reactor. This behavior is also seen in the isotherms in Fig. 13. We observe a "parabolic" burn front due to the faster burn velocity near the walls. The minimum number of $x$ and $y$ grid points was 10 while the maximum used was 98 in the $x$ direction and 86 in the $y$ direction. The minimum mesh spacings were 0.0004 cm in the $x$ direction and 0.006 cm in the $y$ direction. Such a grid distribution would correspond to an equispaced calculation with over 600,000 nodes. The in-core storage of the mesh alone could be far too large for many mainframe computers. Once again a calculation with a total of 10,000 equispaced grid points failed to propagate. A typical grid for this problem is illustrated in Fig. 14.

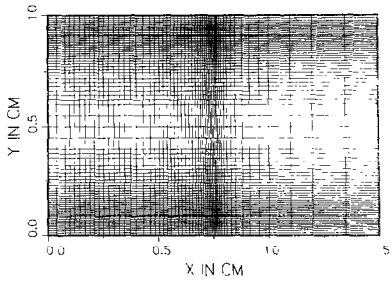


FIG. 14. A typical two-dimensional adaptive grid for test Problem 3.

## 5. SUMMARY

We have developed a variable node adaptive algorithm for solving two-dimensional mixed initial-boundary value problems. The method adaptively adjusts the number of grid points in each direction such that positive weight functions are equidistributed approximately. In addition, we chose the time steps such that the local error per unit step is below some tolerance. When the solution can be propagated with a fixed number of points, we use grid extrapolation. This reduces the number of Jacobian evaluations that must be performed.

We have applied the method to the solution of several problems involving the burning of metallic powders. In each example we have obtained smooth burn fronts in both space and time. The adaptive selection of grid points has enabled these problems to be solved with a fraction of the number of points required by an equispaced method. In addition, for two of the three test examples, we could not get the reactions to propagate with the maximum number of in-core equispaced nodes allowable on our mainframe. The source term used in each example was derived on the assumption of infinitely fast chemical reaction rates and the diffusion of heat into a sphere. While other source terms have been suggested, we have found the model proposed in this paper to be realistic in its ability to predict initiation and the effect of burn velocity on particle size distribution. The method developed in this paper could be applied equally well to a number of other physical problems.

## REFERENCES

1. A. P. HARDT AND P. V. PHUNG, *Combust. Flame* **21** (1973), 77.
2. M. D. SMOOKE AND M. L. KOSZYKOWSKI, *SIAM J. Sci. Statist. Comput.*
3. M. BIRNBAUM, "Determination of Palladium/Aluminum Reaction Propagation Rates and Temperatures," Sandia National Laboratories Report, SAND78-8503, 1978.
4. M. BIRNBAUM, "Studies of Palladium/Aluminum Powders," Sandia National Laboratoriezs Report, SAND78-0486, 1978.
5. F. BOOTH, *Trans. Faraday Soc.* **44** (1953), 790.
6. S. B. MARGOLIS, *SIAM J. Appl. Math.* 43 (1983), 351.
7. K. MILLER AND R. MILLER, *SIAM J. Numer. Anal.* **18** (1981), 1019.
8. R. J. GELINAS, S. K. DOSS, AND K. MILLER, *J. Comput. Phys.* **40** (1981), 202.
9. S. F. DAVIS AND J. E. FLAHERTY, *SIAM J. Sci. Statist Comput.* **3** (1982), 6.
10. A. B. WHITE, *SIAM J. Numer. Anal.* **19** (1982), 683.
11. J. H. BOLSTAD, "An Adaptive Finite Difference Method for Hyperbolic Systems in One Space Dimension," Ph.D. thesis, Stanford University, 1982.
12. H. A. DWYER, R. J. KEE, AND B. R. SANDERS, *AIAA J.* **18** (1980), 1205.
13. K. A. WINKLER, M. L. NORMAN, AND M. J. NEWMAN, Adaptive mesh techniques for fronts in star formation, presented at the International Conference on Fronts, Interfaces and Patterns, Los Alamos National Laboratory, 1983.
14. J. DJOMEHRI AND K. MILLER, "A Moving Finite Element Code for General Systems of PDE's in 2-D," Technical Report, Center for Pure and Applied Mathematics, University of California, Berkeley, 1981.
15. M. BERGER, W. D. GROPP, AND J. OLIGER, *in* "Numerical Grid Generation Techniques," NASA Conference Publication 2166, 1981.

16. M. BERGER, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," Ph.D. thesis, Stanford University, 1982.
17. W. D. GROPP, *SIAM J. Sci. Statist. Comput.* **1** (1980), 191.
18. H. A. DWYER, M. D. SMOOKE, AND R. J. KEE, *in* "Numerical Grid Generation" (J. F. Thompson, Ed.), Elsevier, New York, 1982.
19. B. M. HERBST, A. R. MITCHELL, AND S. W. SCHOOMBIE, "Equidistributing Principles Involved in Two Moving Finite Element Methods," Technical Report, University of the Orange Free State, South Africa, 1981.
20. M. D. SMOOKE, *J. Optim. Theory Appl.* **39** (1983), 489.
21. M. D. SMOOKE, *J. Comput. Phys.* **48** (1982), 72.
22. J. KAUTSKY AND N. K. NICHOLS, *SIAM J. Sci. Statist. Comput.* **1** (1980), 499.
23. P. DEUFLHARD, *Numer. Math.* **22** (1974), 289.
24. A. R. CURTIS, M. J. POWELL, AND J. K. REID, *J. Inst. Math. Appl.* **13** (1974), 117.
25. G. N. NEWSAM AND J. D. RAMSDELL, "Estimation of Sparse Jacobian Matrices," Harvard University Report TR-17-81, 1981.
26. T. F. COLEMAN AND J. J. MORE, "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems," Argonne National Laboratory Report ANL-81-39, 1981.
27. P. W. DLETZ, *Ind. Eng. Chem. Fundam.* **18** (1979), 283.