NASA Contractor Report 181730

ICASE REPORT NO. 88-60

ICASE

AN ASYMPTOTIC INDUCED NUMERICAL METHOD FOR THE CONVECTION-DIFFUSION-REACTION EQUATION

Jeffrey S. Scroggs

Danny C. Sorensen

Contract No. NAS1-18605
October 1988

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

AN ASYMPTOTIC INDUCED NUMERICAL METHOD FOR THE CONVECTION-DIFFUSION-REACTION EQUATION

JEFFREY S. SCROGGS * AND DANNY C. SORENSEN†

ICASE
NASA Langley Research Center
Hampton, Virginia 23665-5225

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, Illinois 60439

ABSTRACT

A parallel algorithm for the efficient solution of a time dependent reaction convection diffusion equation with small parameter on the diffusion term will be presented. The method is based on a domain decomposition that is dictated by singular perturbation analysis. The analysis is used to determine regions where certain reduced equations may be solved in place of the full equation. Parallelism is evident at two levels. Domain decomposition provides parallelism at the highest level, and within each domain there is ample opportunity to exploit parallelism. Run-time results demonstrate the viability of the method.

* Research conducted while in residence at the Center for Supercomputing Research and Development, University of Illinois supported in part by the National Science Foundation under Grant No. US NSF PIP-8410110, the U.S. Department of Energy under Grant No. US DOE-DE-FG02-85ER25001, the Air Force Office of Scientific Research under Grant No. AFOSR-85-0211, the IBM Donation to CSRD, and by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48. Research was also partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while in residence at the Institute for Computer Applications in Science and Engineering (ICASE).

† Work supported in part by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U.S. Department of Energy under Contracts W-31-109-Eng-38, DE-AC05-840R21400.

1. INTRODUCTION. In this paper, a new approach to solving partial differential equations which model fluid flow is discussed and demonstrated. The algorithm is appropriate for modeling laminar transonic flow, such as through a duct of variable width. The method is an asymptotics-induced numerical method suitable for parallel processors which represent the state of the art in scientific computers. The contents of this paper concentrate on a description of the method and computational results. The complete theoretical basis for the algorithm has been developed in [32] and will appear separately.

Competition between convection, diffusion, and reaction is crucial to the understanding of fluid flow. When modeling transonic flow, except in regions of rapid variation such as in shocks and boundary layers, convection and reaction dominate over diffusion. A novel aspect of this method is the use of asymptotic analysis to exploit these physical properties, providing the theoretical basis for a domain decomposition. The analysis identifies the following two types of subdomains: regions where the solution is smooth, where a reduced equation may be solved; and regions of rapid variations, such as in a neighborhood of a shock, where the full equation must be solved. Domain decomposition provides large-grain parallelism. The domain decomposition is independent of the choice of numerical schemes for the subdomains; thus, schemes may be chosen which are a source of smaller-grain parallelism. Even though large grain parallelism is not exploited in the implementation, significant speedups are demonstrated. In addition to dictating the domain decomposition, asymptotics also provides a means of approximating solutions to the problem. In this way, a set of simplified problems is obtained that is better conditioned for numerical computations; hence, they may be solved by conventional techniques. The use of asymptotic analysis to precondition the computations is a new aspect of this method.

The techniques presented herein are applicable to Computational Fluid Dynamics (CFD) in the transonic and supersonic regimes, in physical settings such as laminar flow through a nozzle (duct) and laminar flow around airfoils and other bodies. The gasdynamic equations, including viscous effects, are used as a model in these settings. Except for very simple geometries and boundary conditions there is no analytic solution to these gasdynamic equations, and a numerical solution is difficult to obtain. For these reasons new algorithms are usually developed and tested on a more tractable canonical equation. The convection-diffusion-reaction equation

$$\frac{\partial u}{\partial t} + A(x,t,u)\frac{\partial u}{\partial x} - \epsilon\frac{\partial^2 u}{\partial x^2} - r(x)u = 0, \tag{1}$$

is such a canonical equation and will be the focus of this paper. The flows considered in this paper are not reacting fluids. Here, the reaction term arises from the effects of a variable cross sectional area in a duct. When the equation is nondimensionalized [24], the diffusion coefficient ϵ is inversely proportional to the Reynolds number. Based on free-stream conditions in transonic flow, the Reynolds number for this problem is large. Asymptotic analysis exploits the smallness of the positive parameter ϵ and involves study of the solution as ϵ tends to zero ($\epsilon \downarrow 0$). This equation contains many of the properties that make the gasdynamic equations difficult to solve; namely, it is

capable of modeling rapid variations such as shocks and boundary layers. The method is capable of obtaining solutions to (1) when the shock is not stationary, which extends Howes' studies [12,13] into the time-dependent regime.

Asymptotic analysis gives qualitative and quantitative information as $\epsilon \downarrow 0$. The numerical method presented here exploits the analysis to determine an accurate solution for small positive ϵ. The method is in the spirit of matched asymptotic expansions [19,16], but it is not a numerical implementation of matched asymptotics. The asymptotic analysis involves the derivation of analytic upper and lower bounds on the solution, and is performed in the style of Howes [9,10,11]. Initially, bounds are discussed which are valid only in certain subregions. Then the bounds are combined to form a global *a priori* error bound.

Another novel feature of the method is the availability of extensive error information in the form of both *a priori* error bounds and reliable *a posteriori* error estimates. Reliable *a posteriori* error estimates are obtained using the error analysis which accompanies the numerical schemes used to solve the sub-problems. In addition, *a priori* error bounds are provided through the use of asymptotic analysis. The error analysis is based on the physical mechanisms associated with the problem; hence, it is based on accurate information (see [23]), not on the truncation of a Taylor series of a poorly behaved function.

The method is an iterative technique. A linearized version of the original problem is solved in each step of the iteration. Theorems establishing the convergence of the method are presented, the proofs will appear in a subsequent paper. Computational experiments show that in just a few steps of the iteration, the solution to the nonlinear equation may be obtained. The iterative algorithm as well as the theorems associated with it are novel.

In the next section, some of the ideas behind multiple scales asymptotic analysis are discussed. In addition, an introduction into how the asymptotic analysis and the numerical analysis are blended to form a computational method is presented. In Section 3 the problem is presented. Asymptotic analysis specific to this problem is discussed with the theorems supporting the method in Section 4. The iteration and method for detection of the subdomain boundary is discussed in Section 5. The numerical schemes used in the method are presented in Section 6. The method is stated in algorithmic form in Section 7. In Section 8 computational results on an Alliant FX/8 are presented.

2. MULTIPLE SCALES. Many problems of scientific interest have multiple scales. These problems are characterized by the presence of distinguishable physical mechanisms, each associated with a temporal or spatial gauge or scale. When modeling a shock in a duct, for example, the width of the duct provides one scale, and the thickness of the shock layer provides another. The resolution of these scales is frequently required to determine the physics of interest. Asymptotic analysis provides analytic tools to identify and utilize the multiple scales. The relative importance of

any two physical processes in a given domain may be measured by the ratio of the corresponding scales; thus, the various scales may be ranked by a set of dimensionless parameters, the ratios of scales. When the ratio of two scales is a large or a small number, then it often happens that one of the competing mechanisms is dominant in most of the domain. For example, in laminar duct flow with large Reynolds number the effects of viscosity may be ignored except in a neighborhood of the shock and boundary layers. The scales of the various competing processes (and, therefore, the relative magnitudes of the dimensionless parameters) usually change as the phenomenon evolves. Consider the behavior of the solution of the nonlinear parabolic equation,

$$P[u] := u_t + uu_x - \epsilon u_{xx} - ru = 0, \tag{2}$$

where ϵ is a small positive parameter. This equation may be used as a model for shocks and boundary layers. For example, if $r(x) = -a'(x)/a(x)$, where $a(x)$ is the width of the duct of Figure 1, then this equation is associated with the flow through the duct [13]. There are (at least) two sets of scales appropriate when modeling shocks—the

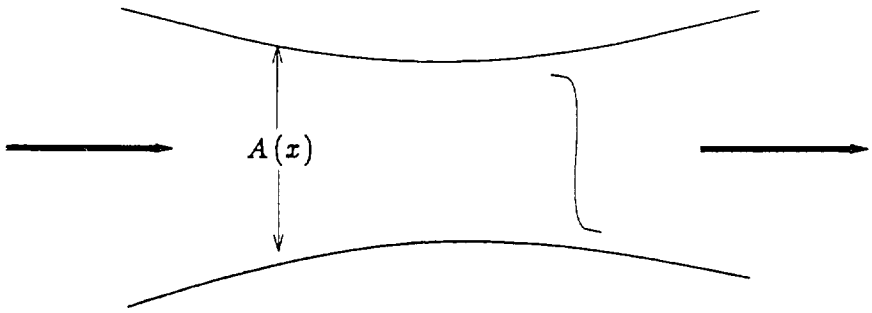

FIG. 1. *Variable width duct.* (From [32].)

scales associated with the original variables (x, t), and the scales appropriate in a small neighborhood of the shock (these are discussed in Section 4).

The most easily tractable multiple-scale problems are those in which there are only a small number of widely separated groups of scales and the motion on the fastest scales has little influence on the smooth part of the solution. An identifying feature of this class is the presence of local regions in which the solution undergoes rapid variation. Such regions are called boundary or internal layers, when located in the neighborhood of a boundary or in the interior of the domain, respectively. These are the problems that are most natural for multitasking because it is easy to break up the domain according to the regions of different local behavior. The method presented here is appropriate for this class of multiple-scale problems.

The decomposition into domains is accomplished using a symbiosis of numerics and asymptotics. The asymptotic analysis identifies the regions where diffusion is

negligible. In these regions, it is sufficient to solve a reduced equation. Solving this reduced equation can significantly reduce the work in the numerical method, and/or increase the potential for parallelism. For example, this allows the use of the method of characteristics to obtain a good approximation for the solution of (2). The numerics provides a means of solution in the subdomains, and also a feedback mechanism. The numerical scheme can expose regions of unexpected behavior, confirming or correcting the asymptotics-induced subdomain boundaries. This decomposition permits the use of locally refined meshes, allowing the concentration of computational effort in the regions where it is needed most.

There is much literature on multiple-scales problems. Analytic methods for multiple-scales problems are discussed in the books [3,16,20,34]. The theory of multiple-scales analysis is discussed in [17,6,21]. The books [2,25] discuss both the techniques and the theory behind them. Finally, numerical techniques for multiple-scales problems are discussed in the papers [22,26,33]. This list is meant only as an introduction to the literature, and not as a complete list.

3. THE NONLINEAR PROBLEM. The method will be described and demonstrated by solving (2) on the domain

$$D := \{(x,t) | 0 \le x \le b, 0 \le t < T\}, \tag{3}$$

subject to

$$u(x,0) = \gamma(x), \quad 0 < x < b; \tag{4}$$

$$u(0,t) = \alpha(t), \quad 0 < t < T; \text{ and} \tag{5}$$

$$u(b,t) = \beta(t), \quad 0 < t < T. \tag{6}$$

The portion of the boundary along which the data is specified is denoted by

$$\Pi := \{(x,t) | 0 \le x \le b,\ t = 0\} \bigcap \{(x,t) | 0 \le t < T,\ x = 0, b\}.$$

For the sake of simplicity, it is assumed that all boundaries are inflow boundaries, that is, $\alpha(t) \ge \alpha_0 > 0$ and $\beta(t) \le \beta_0 < 0$. The boundary data is assumed to be compatible; thus,

$$\alpha(0) = \gamma(0), \quad \text{and} \quad \gamma(\mathrm{b}) = \beta(0). \tag{7}$$

The coefficient of the forcing term $r(x)$ is bounded with bounded derivatives. In addition, it is assumed that the solution to the reduced equation

$$P_0[\check{u}] := \check{u}_t + \check{u}\check{u}_x - r\check{u} = 0 \tag{8}$$

has continuous derivatives at $(x,t) = (0,0)$, and $(x,t) = (b,0)$. This last restriction prevents the formation of corner layers, and may be expressed as

$$\frac{d\alpha}{dt} + \gamma\frac{d\gamma}{dx} - r\gamma = 0, \quad \text{for } (x,t) = (0,0); \tag{9}$$

$$\frac{d\beta}{dt} + \gamma\frac{d\gamma}{dx} - r\gamma = 0, \quad \text{for } (x,t) = (b,0). \tag{10}$$

Under these conditions, the solution to (2) is uniquely defined [4].

4. ASYMPTOTIC ANALYSIS. Asymptotic analysis is employed to identify the dominant physics, creating an efficient and accurate numerical method. Here we sketch the analysis in the neighborhood of a shock. Readers wishing full detail and proofs of the results are encouraged to consult [32,11,10]. Shocks form in regions of merging characteristics (see Figure 2). Since the boundary conditions imposed on

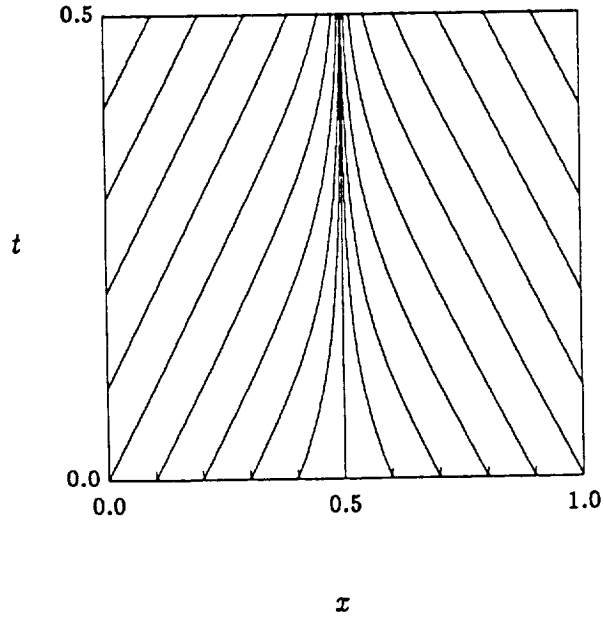

FIG. 2. *Characteristics of steady-state solution* $u = -\tanh[(.5 - x)/2\epsilon]$, *for* $\epsilon = .05$. (From [32].)

the problem are inflow conditions on both the $x = 0$ and $x = b$ boundaries, the characteristics are traveling in the direction of increasing x from $x = 0$, and in the direction of decreasing x from $x = b$. These will merge (become asymptotically close) somewhere inside D, forming a shock. The merging of the characteristics stabilizes the shock, and keeps it from dispersing.

Since the behavior of u as $\epsilon \downarrow 0$ is of interest, it is natural to first study the solution of the reduced equation (8). Weak solutions $\breve{u}$ are sought for (8) with boundary data

(4-6). In order that $\check{u}$ be uniquely defined, it is necessary to impose an entropy condition [18]. Suppose that $\check{u}$ has a single shock. That is, suppose $\check{u}$ is the solution to (8) subject to (4-6) that is discontinuous only along a curve $(x,t) = (\Gamma(t),t)$. For small ϵ, this curve lies in the shock-layer region of the solution to the full problem. The size of this region tends to zero as $\epsilon \downarrow 0$. Analytic methods for choosing Γ are discussed by Whitham [36], Kevorkian and Cole [16], and others. The path of the discontinuity is an analytic tool needed only for the theory. Since Γ is not needed for the computations, methods for choosing Γ will not be discussed here.

The initial and boundary data are assumed to be smooth; thus, the shock does not exist at $t = 0$. Rather, Γ is assumed to be undefined for $t < t^{\Gamma}$, where $t = t^{\Gamma}$ is the time $\check{u}$ becomes discontinuous. It is natural to describe $\check{u}$ in terms of the following functions:

$$\check{u}(x,t) = \begin{cases} \check{u}_0(x,t) & \text{for } 0 < t \leq t^{\Gamma} \\ \check{u}_1(x,t) & \text{for } x < \Gamma^{-1}(t) \text{ and } t \geq t^{\Gamma} \\ \check{u}_2(x,t) & \text{for } x > \Gamma^{-1}(t) \text{ and } t \geq t^{\Gamma}. \end{cases}$$

The shock speed for $\check{u}$ is

$$(\check{u}_1(\Gamma(t),t) + \check{u}_2(\Gamma(t),t))/2), \tag{11}$$

so the entropy condition may be expressed as

$$\check{u}_1(\Gamma(t),t) > (\check{u}_1(\Gamma(t),t) + \check{u}_2(\Gamma(t),t))/2) > \check{u}_2(\Gamma(t),t). \tag{12}$$

Under these conditions,

$$\mu(t) = \check{u}_1(\Gamma(t),t) - \check{u}_2(\Gamma(t),t) \geq 0. \tag{13}$$

The regions where $\check{u}$ is a good approximation to u are defined by presenting functions which bound the difference $\check{u} - u$. These bounds are small except in an asymptotically small neighborhood of the shock. The bounds, based on Howes [11], are reflected in the following theorem.

THEOREM 1. (Howes) *Let $\check{u}$ be the solution to $P_0[\check{u}] = 0$ and u the solution to $P[u] = 0$ on D, each satisfying the the boundary data* (4-6). *The solution to the reduced equation, $\check{u}$, is possibly discontinuous on the curve $(x,t) = (\Gamma,t)$. Assume that the boundary data satisfy the following smoothness conditions: the data* (4-6) *satisfy the compatibility conditions* (7),(9-10), *and α, β, γ, with their first and second derivatives are all bounded. Then for ϵ small enough*

$$|u - \check{u}| = O(\mu \exp[-f^2(x,t)/\epsilon^{1/2}]) + O(\epsilon)$$

when the derivatives of $\check{u}$ are continuous across Γ, and

$$|u - \check{u}| = O(\mu \exp[-f^2/\epsilon^{1/2}]) + O(\epsilon^{1/2}\delta \exp[-f/\epsilon^{1/2}]) + O(\epsilon)$$

in the more general case when the derivatives of $\check{u}$ are not continuous across Γ. Here $f(x,t)$ is a distance function between (x,t) and (Γ,t), and δ is an upper bound on the difference of the normal derivative of $\check{u}$ across Γ.

It is now reasonable to utilize the theorem to make the definitions of the subdomains more precise. The internal layer is the following neighborhood of Γ:

$$D_{IL} = \{(x,t) | (x,t) \in D, |x - \Gamma^{-1}(t)| \leq \Delta(t)\}. \tag{14}$$

Here $\Delta(t) \leq K\eta(t)\epsilon^{1/4}\ln^{1/2}\epsilon$ is the width of the internal layer at time t (K is a constant independent of ϵ). The outer region is the complement of D_{IL} with respect to D, that is,

$$D_{OR} = \{(x,t) | (x,t) \in D, |x - \Gamma^{-1}(t)| > \Delta(t)\}. \tag{15}$$

The upper bound on the size of the internal layer is based on the $\exp[-f^2/\epsilon^{1/2}]$ term in the error bounds of Theorem 1.

Theorem 1 motivates a preconditioning for the problem in D_{OR}. The theorem states that (8) may be solved in place of (2). In addition, the theorem provides an error bound if diffusion (artificial or implicit in the numerical scheme) is incorporated into the solution process of either (8) or (2). Thus, the numerical method for D_{OR} may be chosen from the wide variety of methods designed for hyperbolic equations [1,35,8,28].

The solution in the outer region is used to provide boundary data for the problem in the internal layer. (This is justified in Section 5.1.) Thus, it is possible to have boundary data for the internal-layer problem which is perturbed from the exact solution. The effects of this perturbation are that the height and location of the shock can vary by the same magnitude as the perturbation itself. This results in an error of magnitude $O(1)$ as $\epsilon \downarrow 0$ in an asymptotically small neighborhood of the shock, and is reflected in the error bound of the following theorem [32].

THEOREM 2. *Let u and v be solutions to (2) on the domain D_{IL} with their boundary values satisfying the following smoothness conditions: the data are bounded with bounded derivatives. Assume that the curve defining ∂D_{IL} is smooth.* [1] *Let*

$$\psi(x,t) = u - v, \quad \text{for } (\mathrm{x},\mathrm{t}) \in \partial \mathrm{D}_{\mathrm{IL}}.$$

Then for ϵ and Δ small enough

$$|u - v| = O(\psi) + O((1 + f^2/\epsilon^2)^{-1}), \tag{16}$$

where $f(x,t)$ measures the distance from (x,t) to (Γ,t).

The theorem is the source for a second preconditioner. Namely, a scaled and translated coordinate system based on (16). Let $f = |x - \Gamma(t)|$. Setting $\tilde{x} = (x - \Gamma)/\epsilon$,

[1] The curve is continuous with a continuous tangent.

the second term in the right hand side of (16) will be large when $\tilde{x} = O(1)$ as $\epsilon \downarrow 0$. An analogue of $\tilde{x}$ will be used for the spatial coordinate in D_{IL}, and will be described in Section 6.2. The use of this scaled and translated coordinate system creates a better conditioned numerical problem, thus it is a preconditioning.

The local error bound of Theorem 1 is now used to form a global *a priori* error bound. The bound, as presented in Theorem 3 below, is sharp in D_{OR}; however, the bound reflects the possibility of shock displacement in the internal-layer region.

THEOREM 3. *Let u be the solution to* (2) *satisfying* (4-6). *Suppose v is obtained by first solving* (8) *in D_{OR} subject to* (4-6), *then solving* (2) *on D_{IL} with boundary data v on ∂D_{IL}. Assume the compatibility conditions* (7),(9-10) *obtain, and that the data* (4-6) *are bounded with bounded derivates. If $E = \|u - v\|_1$, then for ϵ small enough*

$$E = O(\epsilon)$$

in D_{OR}, and

$$E = O(\epsilon^{1/4} \ln^{1/2} \epsilon)$$

in D_{IL}.

The computational results were much stronger than the theorem suggests. Both the magnitude of Δ, and the magnitude of the error in the internal-layer subdomain were smaller in the computational results. Thus, the *a priori* error bounds of the asymptotic analysis did not reflect all of the accuracy and behavior of the computational algorithm.

Asymptotics identified two subdomains and provided preconditioners for the problems within the subdomains. The preconditioner for the full equation in D_{IL} is the use of the local scale $\tilde{x} = (x - \Gamma)/\epsilon$ dictated by Theorem 2. This scale allows the diffusion to be modeled accurately, hence the grid is fine enough to resolve the shock. It is reasonable to use this scaling in the method, because computationally the internal-layer subdomain is of width $O(\epsilon)$. The preconditioning in the outer-region subdomain D_{OR} is to solve (8) in place of (2), and was justified in Theorem 2. First, the domain decomposition and preconditionings are combined with a functional iteration to form the computational method. The particular numerical methods discussed herein are not new; however, their combination to form this method is.

5. DISCUSSION OF THE METHOD. An iteration is formed by linearizing the reduced problem. Each step of the iteration requires the solution of (8) in the outer-region subdomain, and (2) in the neighborhood of a shock. This is discussed in Section 5.1. Once the iteration has been described, the boundary detection scheme is presented. The method assumes no *a priori* information about the location of the

internal-layer boundary, and is supported by theory. In Section 5.3, convergence of the method is presented. Numerical details of the method will be presented in Section 6.

5.1. Iteration. In general, each step of the iteration requires the solution of a linear convection-reaction equation in the outer-region subdomain, followed by the solution of a nonlinear convection-diffusion-reaction equation in the internal layer. The convection-reaction equation

$$U_t^{k+1} + U^k U_x^{k+1} - rU^{k+1} = 0 \tag{17}$$

is formed by lagging the convection coefficient of (8). The boundary of the internal-layer subdomain is allowed to change during the iterations. Thus, denote the outer-region subdomain for iterate U^{k+1} by D_{OR}^k, and denote the complement of D_{OR}^k with respect to D by D_{IL}^k. That is, U^{k+1} is obtained by solving (17) in D_{OR}^k, then solving (20) in D_{IL}^k.

The solution of (17) in D_{OR}^k is obtained via a modification of the method of characteristics to account for the forcing term rU^{k+1}. The characteristic transformation $(x,t) \to (\xi,\tau)$ is defined by setting $t = \tau$, and by solving

$$\frac{\partial x^k}{\partial \tau} = U^k(x^k(\xi,\tau),\tau), \tag{18}$$

with initial conditions

$$x^k(0) = \xi, \quad \text{for } b > \xi > 0;$$

$$x^k(y_a^{-1}(\xi)) = 0, \quad \text{for } \xi < 0;$$

$$x^k(y_b^{-1}(\xi)) = 0, \quad \text{for } \xi > b;$$

where, $(\xi,\tau) = (y_a(\tau),\tau)$ is the image of the curve $(x,t) = (0,t)$, and $(\xi,\tau) = (y_b(\tau),\tau)$ is the image of the curve $(x,t) = (b,t)$. Utilizing this transformation, it is a simple task to solve

$$U_\tau^{k+1} = rU^{k+1} \tag{19}$$

in place of (17) along the characteristics defined by (18). This transformation becomes singular in a neighborhood of a shock, hence cannot be applied in the internal-layer subdomain. This fact is the basis of the procedure used to determine ∂D_{IL}^k.

Solutions to the reduced equation are poor approximations to the solution of the full equation in regions of large gradients, such as in the internal-layer subdomain. Thus, the full equation is solved in the internal layer at each iteration. The equation

$$U_t^{k+1} + U^{k+1} U_x^{k+1} - \epsilon U_{xx}^{k+1} - rU^{k+1} = 0 \tag{20}$$

is solved in the internal-layer subdomain for each k. Boundary data for the internal-layer subdomain is provided by the solution of (17) in the outer region. This is justified by observing that for ϵ small enough, the boundary of the internal-layer subdomain will be an inflow boundary.

5.2. Boundary Detection. It is desirable to be able to compute the location of the internal-layer subdomain during the course of the iteration. This has the advantage of requiring less *a prior* information. In addition, if the initial guess provides a poor approximation to the location of the internal-layer subdomain, the method will be able to correct the location of the boundary in the course of the iteration.

The method used to locate the internal-layer subdomain boundary is based on properties of the transformation used to solve (17). The transformation used to solve (17) will become singular (or nearly singular) in the region where characteristics merge (see Figure 2). Thus, the Jacobian

$$J^k = \partial x/\partial \xi, \tag{21}$$

of the transformation (18) will be asymptotically small in a neighborhood of the shock, while it is $O(1)$ in the outer-region subdomain. (For more details on the relationship between the magnitude of J^k and the nature of the transformation, see [31].) This is the measure used to locate the boundary of the internal-layer subdomain. The size of the Jacobian is monitored via the solution of an ODE along each characteristic path of interest. Combining the partial of (18) with respect to ξ and the partial of (21) with respect to τ, the equation

$$\frac{\partial J^k}{\partial \tau} = J^k \frac{\partial U^k}{\partial x}(x(\xi,\tau),\tau) \tag{22}$$

may be derived. The Jacobian is determined by solving this equation subject to $J^k(x_0,t_0) = J_0^k$ for $(x_0,t_0) \in \Pi$. The behavior of the solution inside the domain D is of primary interest; therefore, it is sufficient to monitor $\hat{J}^k := J^k(x_1,t_1)/J_0^k$ in place of J^k to determine the boundary. The ratio is determined using (22) with $\hat{J}^k$ in place of J^k, subject to $\hat{J}^k(x_0,t_0) = 1$ for $(x_0,t_0) \in \Pi$. The curve on which $\hat{J}^k$ becomes nearly singular, that is, where

$$\hat{J}^k(x,t) = TOL, \tag{23}$$

for some small number TOL, is the boundary of the internal layer subdomain for iteration k. The subdomains separated by (23) will, in general, be different than the subdomains used in the theorems. Thus, the subdomains used in the numerical method are

$$\hat{D}_{OR}^k = \{(x,t) | (x,t) \in D,\ \hat{J}^k(x,t) \geq TOL\}, \tag{24}$$

and,

$$\hat{D}_{IL}^k = \{(x,t) | (x,t) \in D,\ \hat{J}^k(x,t) < TOL\}. \tag{25}$$

The theory applies provided

$$D_{IL} \subseteq \hat{D}_{IL}^0 \subseteq \hat{D}_{IL}^1 \subseteq \ldots \subseteq \hat{D}_{IL}^{k-1} \subseteq \hat{D}_{IL}^k; \tag{26}$$

however, convergence was observed when this relation failed, thus the constraint (26) is not a necessary condition for the computational method to converge.

Heuristics, based on both accuracy and efficiency, are used to choose TOL. If TOL is too small, then accuracy will suffer. This is because the internal-layer subdomain will be too small, and the data provided at the boundary of the internal-layer subdomain will have large perturbations as compared to the desired solution. If TOL is too large, the internal-layer subdomain will be too large, and the computational mesh will be refined in regions where the solution is smooth, creating excess work.

5.3. Convergence. An advantage to this method is the availability of extensive error information. A global error bound based on Theorem 3 will be presented in this section. First, convergence of the iteration is established by showing the iteration is a contraction mapping. For more details on these results, see [32].

The convergence of the iteration (17) to a solution of (8) in the outer region will be shown by comparing successive iterates, then establishing a lower bound on the latest time at which the iteration is a contraction. For the sake of the theorem, the boundary of the internal-layer subdomain is assumed to be stationary from iteration to iteration ($\hat{D}_{OR}^{k-1} = \hat{D}_{OR}^{k} = D_{OR}$). With the analysis that follows, this theorem provides a lower bound for the largest time at which the iteration will converge:

THEOREM 4. *Let $\{U^k\}_{k=1}^{\infty}$ be the set of successive iterates of* (17) *in the subdomain D_{OR} satisfying* (4-6) *with initial guess U^0. Assume U^0 satisfies* (4-6) *and is Lipschitz continuous on D. The boundary data are assumed to satisfy the compatiblity conditions* (7),(9-10) *and to have bounded first and second derivatives. Let*

$$\delta = \sup_{D} |U^k - U^{k-1}|.$$

Then

$$|U^{k+1} - U^k| < \delta C e^{-\lambda t}(e^{Rt} - 1) \tag{27}$$

for $(x,t) \in D_{OR}$. Here C, λ and R are known positive constants.

This theorem provides an upper bound on the latest time for which the iteration converges. Apply the infinity norm to (27) to obtain

$$\|U^{k+1} - U^k\|_\infty \leq \hat{C}\|U^k - U^{k-1}\|_\infty.$$

Then the following corollary provides the conditions for convergence.

COROLLARY 5. *Suppose that the conditions of Theorem 4 obtain. Let $T_{\max}$ be the largest positive number such that*

$$\hat{C} = \sup_{0 \leq t \leq T_{\max}} C e^{-\lambda t}(e^{Rt} - 1) \leq 1.$$

If the bound on time in (3) satisfies $0 < T < T_{\max}$, *then the iteration in* D_{OR} *defined by (17) is a contraction mapping; therefore, the sequence of iterates converges to* $v = \lim_{k\to\infty} U^k = U^\infty$, *which is a solution of (8) on* D_{OR} *satisfying (4-6).*

A statement of an *a priori* error bound for the computational method is presented in Corollary 6 below. As with Theorem 3, the bound is sharp in D_{OR}; however, the bound is crude in the region of the shock.

COROLLARY 6. *Let u be the solution to* (2) *satisfying* (4-6). *Suppose each iterate* U^k *is obtained by first solving* (17) *in* D_{OR} *subject to* (4-6), *then solving* (20) *on* D_{IL} *with boundary data* U^k *on* ∂D_{IL}. *Assume that the compatibility conditions* (7),(9-10) *obtain, and that the data* (4-6) *are bounded with bounded derivates. Suppose* $0 < T < T_{\max}$, *and let* $v = U^\infty$. *If* $E = \|u - v\|_1$, *then for* ϵ *small enough*

$$E = O(\epsilon)$$

in D_{OR}, *and*

$$E = O(\epsilon^{1/4} \ln^{1/2} \epsilon)$$

in D_{IL}. *Here* $\Delta = O(\epsilon^{1/4} \ln^{1/2} \epsilon)$.

As with Theorem 3, the computational results reflect that both Δ and the error in the internal-layer subdomain are smaller than the corollary suggests.

6. NUMERICAL DETAILS. The asymptotic analysis has provided a means to precondition the numerical problems. Because the sub-problems are well conditioned, the choice of numerical schemes may be made from a variety of standard methods. This is not usually the case. The class of problems for which the new algorithm is applicable are notoriously difficult to solve, and only a small number of schemes could be employed for its solution (prior to preconditioning). Since the sub-problems are well conditioned, numerical schemes used in the method presented here can be chosen based on criteria such as efficiency or the potential to exploit parallelism.

6.1. Schemes in the Outer Region. The method of characteristics is used to solve the hyperbolic PDE in the outer-region subdomain. This method allows the exploitation of physically motivated parallelism. In addition, the method of characteristics allows handling of the free boundary at $\partial \hat{D}^k_{IL}$ in a straightforward manner.

The method is not limited to using the method of characteristics for the problem in $\hat{D}^k_{IL}$. For example, the schemes for hyperbolic conservation laws [1,8,35] might be modified to account for the term rU^k and used. If this where done, then the method for detection of $\partial \hat{D}^k_{IL}$ could be based on the gradients instead of monitoring the Jacobian.

The method of characteristics scheme involves laying down a characteristic coordinate system, updating solution values, and monitoring the Jacobian. To update the solution and monitor the Jacobian requires a negligible computational cost. The discrete version of (18) is used to determine the characteristic coordinate system. Thus, for each iteration k, a set of characteristics $\{x_i^k\}_{i=1}^{I_\beta}$ are computed and used as the computational grid. Here, the superscript k is the iteration number, and the subscript i identifies the characteristic. To determine a characteristic,

$$\frac{dx_i^k}{dt} = U^k(x_i^k, t) \tag{28}$$

is solved for each $i = 1$ to I_β. All of Π is an inflow boundary for D, thus initial conditions

$$x_i^k(\tau_i) = \xi_i,$$

are specified along all of Π. On the $(x,t) = (0,t)$ portion of Π, the initial condition is $\xi_i = 0$ at $\tau_i = il$, for $i = 1$ to I_α. On the $(x,t) = (x,0)$ portion, the initial condition is $\xi_i = (i - I_\alpha - 1)h$ at $\tau_i = 0$, for $i = I_\alpha + 1$ to I_γ. And on the $(x,t) = (b,t)$ portion, the initial condition is $\xi_i = b$ at $\tau_i = (i - I_\gamma - 1)l$, for $i = I_\gamma + 1$ to I_β. Here l and h are the increments for time and space, respectively. The locations of characteristics are desired at time increments of Δt. It is assumed that τ_i is some integer multiple of Δt; thus, the same temporal points are used for all characteristics for all iterations. Each characteristic is obtained on a Δt interval using the Trapezoid rule to solve (28). The Trapezoid rule is solved via a Newton Iteration. The computed value of $x_i^k(t_n)$ is denoted $x_{i,j}^k$.

As mentioned in Section 5.2, $\hat{J}^k$ is monitored along each characteristic to determine the boundary of the internal-layer subdomain at each iteration. The solution to (22) along characteristic i is

$$\hat{J}^k(x_i^k(t), t) = \exp[S_i^k(t)], \tag{29}$$

where

$$S_i^k(t) = \int_{\tau_i}^{t} U_x^k(x_i^k(\tau), \tau) d\tau. \tag{30}$$

Since computations of (30) are better conditioned than those of (29), S_i^k is monitored in place of $\hat{J}^k$. Denote the computed value of $S_i^k(t_j)$ by $S_{i,j}^k$. The integral is determined using the right-hand rectangle rule

$$S_{i,j+1}^k = S_{i,j}^k + \Delta t U_x^k(x_{i,j+1}^k,\ t_{j+1}) \tag{31}$$

with the initial value $S_i^k(\tau_i) = 0$. The monitoring is performed at a minimal cost. It is necessary to keep only the most recent value of S_i^k, thus minimal storage is required for this technique. In addition, the values of $U_x^k(x_{i,j+1}^k,\ t_{j+1})$ are saved from the Newton iteration, hence very little computational cost is required.

The criteria used to determine the boundary will now be made more precise. The boundary $\partial \hat{D}_{IL}^k$ is defined by $\hat{J}^k = TOL$, hence a characteristic is considered to be in the outer region as long as

$$S_{i,j}^k \geq \ln(TOL).$$

Let $t = t_{\hat{j}} = \hat{j}\Delta t$ be the first time this inequality is violated for characteristic i during iteration k. Then the point $(x,t) = (x_{i,\hat{j}}^k, \hat{j}\Delta t)$ is considered to be inside $\hat{D}_{IL}^k$, and characteristic i is considered to be incident with the boundary at the point $(x,t,) = (x_{i,\hat{j}-1}^k, (\hat{j}-1)\Delta t)$, and is *flagged* as being part of $\hat{D}_{IL}^k$.

After $x_{i,j}^k$ has been determined, the solution U^{k+1} is computed by solving (19) subject to (4-6) using the right-hand rectangle rule

$$U_{i,j+1}^{k+1} = [1 + \Delta t\, r(x_{i,j+1}^k)]U_{i,j}^{k+1}. \tag{32}$$

This formula is used until either $j = T/\Delta t$, or until characteristic i enters $\hat{D}_{IL}^k$, whichever happens first. This formula has minimal computational requirements; however, the iteration requires the storage of the most recent iterate for a portion of D.

6.2. Schemes In the Internal Layer Region. The sub-problem in the internal-layer subdomain requires the solution of a parabolic PDE subject to boundary data provided by the solution in the outer region. There are two major aspects of the computations in the internal-layer subdomain—mesh generation, and the difference technique. The mesh follows the shock, and has been scaled; therefore the variation in the solution is resolved on the new coordinate system. Thus, the mesh provides a preconditioning for the problem in the internal-layer, and the computations are not overly sensitive to the particular difference scheme used to solve the partial differential equation. Russell's Modified Method of Characteristics (MMC) [30], an explicit/implicit finite difference method, was chosen to solve the equation due to the regularity of the linear algebra problems which it generates. As with the schemes in the outer-region subdomain, other methods (see [7,5]) could be employed for the solution in $\hat{D}_{IL}^k$. It is necessary that the boundary of the internal-layer subdomain be identified with respect to the grid in the internal-layer subdomain. This is described first. Then the finite difference technique is reviewed.

The base of the internal-layer subdomain is identified by finding which characteristic (or set of characteristics) has the lowest time at which it is flagged as being part of the internal-layer subdomain boundary. Denote the computed value of t^Γ by $\hat{t}^\Gamma$. For $t > \hat{t}^\Gamma$, the base is taken to be the region between the two outer-most characteristics. This could result in non-flagged characteristics being part of the base, but this is not a problem. Once the base characteristics have been located, it is simple to identify whether a flagged characteristic is on the left or on the right boundary of $\hat{D}_{IL}^k$. A flagged characteristics with an index of lower value than a base characteristic is on the left boundary, and a flagged characteristics with an index of higher value than a base characteristic is on the right boundary.

A description of the method used to locate the left and right boundaries is facilitated by first introducing the coordinate system. The computations will be done on a scaled and translated coordinate system with temporal variable $t^* = t/\epsilon$. (The spatial variable will be described later). The temporal grid is the set of points $\{t_n^*\}$, where $t_n^* = n\Delta t^*$ for $n = \hat{t}^\Gamma/(\epsilon\Delta t^*)$ to $T/(\epsilon\Delta t^*)$. These points are a refinement of $\{t_j\}$, the temporal points on which the characteristics are known. The left and right boundaries of the internal-layer subdomain at time t_n^* are denoted by L_n and R_n, respectively. Algorithm 1 describes the method used to determine R; the procedure used to determine L is symmetric, and hence will not be described.

Do $n = \hat{t}^\Gamma/(\epsilon\Delta t^*)$ to $T/(\epsilon\Delta t^*)$
  If $(t_n^* \neq t_j/\epsilon)$ for any j
    then $R_n := R_{n-1}$
  otherwise
    $j := \epsilon t_n^*/\Delta t$
    $H :=$ number of characteristics incident with the right boundary at t_n^*
    If $H = 0$
      then $R_n := R_{n-1}$
    If $H \geq 1$
      then
        $i :=$ index of right-most characteristic incident
              with the right boundary
        $R_n := x_{i,j}^k$

ALGORITHM 1 Determination of R_n.

It is now appropriate to describe the coordinate system on which the computations in the internal layer are based. Denote the middle of the internal-layer subdomain at time t by $M(t) = [R(t) + L(t)]/2$. Then the spatial coordinate in the internal layer is

$$x^* = [x - M(t)]/\epsilon. \tag{33}$$

The temporal coordinate is also scaled, $t^* = t/\epsilon$. Equation (20), may be written as

$$\bar{U}_{t^*}^{k+1} + \left(\frac{dM}{dt} - \bar{U}^{k+1}\right)\bar{U}_{x^*}^{k+1} - \bar{U}_{x^*x^*}^{k+1} - \epsilon\bar{r}\bar{U}^{k+1} = 0. \tag{34}$$

Here, $\bar{U}^{k+1}(x^*, t^*) = U(M + \epsilon x^*, \epsilon t^*)$, and $\bar{r}(x^*) = r(M + \epsilon x^*)$. This is the form of the equation solved in the internal layer.

Equation (34) on the grid defined above is solved using the MMC. The reader should refer to [27,29,30] for a more complete description of the MMC; however, a review of the method is presented here for the sake of completeness. Denote the the spatial points of the computational grid by $x_i^* = i\Delta x^*$, for $i = -I^{IL}$ to I^{IL}. Since Δx^* is a constant and the width of $\hat{D}_{IL}^k$ varies with time, the number of grid points also varies with time, and $I^{IL} = I^{IL}(t)$. The MMC involves approximation of the

convective term $\bar{U}_{t^*}^{k+1} + \left(\frac{dM}{dt} - \bar{U}^{k+1}\right)\bar{U}_{x^*}^{k+1}$ by a backward Euler approximation along the subcharacteristics of (34) using the following formula

$$\bar{U}_{t^*}^{k+1}(x^*,t^*) + \left(\frac{dM(t^*)}{dt} - \bar{U}^{k+1}(x^*,t^*)\right)\bar{U}_{x^*}^{k+1}(x^*,t^*) \tag{35}$$

$$\simeq [\bar{U}^{k+1}(x^*,t^*) - \bar{U}^{k+1}(\check{x}^*,t^* - \Delta t^*)]/\Delta t^*.$$

Here $\check{x}^* = x^* + \Delta t^*(\frac{dM(t^*-\Delta t^*)}{dt} - \bar{U}^{k+1}(x^*,t^* - \Delta t^*))$ and $\bar{U}^{k+1}(\check{x}^*,t^* - \Delta t^*)$ are determined by linear interpolation between spatial grid points. This linear interpolation is performed element by element, thus there is ample opportunity to exploit parallelism within the method here. Once this quantity has been calculated, the second derivative is approximated using a centered difference. Hence, the full formula in the internal layer is

$$\left[1 - \epsilon\Delta t^*\bar{r}(x^*) + \frac{2}{(\Delta x^*)^2}\Delta t^*\right]\bar{U}^{k+1}(x^*,t^*) - \tag{36}$$

$$\frac{\Delta t^*}{(\Delta x^*)^2}(\bar{U}^{k+1}(x^* + \Delta x^*,t^*) + \bar{U}^{k+1}(x^* - \Delta x^*,t^*)) = \bar{U}^{k+1}(\check{x}^*,t^* - \Delta t^*).$$

There is a domain of dependency requirement on the method [29], which requires the absolute value of the partial with respect to x^* of the convection coefficient to be bounded by $1/\Delta t^*$. Since M is independent of x^*, this requirement is that

$$|\bar{U}_{x^*}^{k+1}(x^*,t^*)| < 1. \tag{37}$$

Extra inner iterations may be necessary to step the solution between the temporal grid lines, t_n^*.

7. OUTLINE AND IMPLEMENTATION OF THE ALGORITHM.

7.1. Algorithm. As a summary, the numerical method is outlined in Algorithm 2 below. The parameters such as TOL and Δt are assumed to have been provided by the user. Several steps are not mentioned. For example, if the domain of dependency requirement is violated in the MMC, it may be necessary to reset Δt^*. However, Algorithm 2 shows the major computational requirements.

The algorithm requires an initial guess. In the theory, the initial guess must satisfy the boundary conditions (4-6), and must be Lipschitz continuous; however, for the computational method, it is only required that some approximation technique which is consistent with the effects of the terms of (2) be used to determine U^0. The MMC was chosen for the initial guess. The computations were done on the original

Do $k = 1$ till converged

I. Solve in $\hat{D}_{OR}^k$

A. Do $i = 1$ to I_β

1. $j := \tau_i / \Delta t$
2. $x_{i,j}^k := \xi_i$
3. Do while $j < T/\Delta t$ and $S_i^k \leq \ln(TOL)$
   a. $j := j + 1$
   b. Step x_i^k from t_{j-1} to t_j
   c. Compute solution value using (32)
   d. Update size of S_i^k with (31)

II. Solve in $\hat{D}_{IL}^k$

A. Determine $\partial \hat{D}_{IL}^k$

1. Find base (determine $\hat{t}^\Gamma$)
2. Find R using Algorithm 1
3. Find L (symmetric to R)

B. Set boundary values along $\partial \hat{D}_{IL}^k$

1. Initial conditions
2. Right boundary values
3. Left boundary values

C. Discretize

1. $t^* := \hat{t}^\Gamma / \epsilon$
2. Determine Δt^* to satisfy (37) as a subdivision of $\{t_j\}$
3. $t^* := t^* + \Delta t^*$
4. Do while $t^* \leq T/\epsilon$
   a. $t^* := t^* + \Delta t^*$
   b. Obtain U^k at t^*
      i. Form right hand side of (36)
      ii. Solve implicit portion of (36)
   c. If (37) is violated, go to step C.1.

ALGORITHM 2 Computational method.

coordinates, x, and t. In order that the domain of dependency requirement be met, artificial diffusion was added instead of restricting the size of Δt.

Use of artificial diffusion could lead to some ill effects, especially in the location of the shock. The size of the diffusion coefficient effects the speed of a nonsteady shock. Thus, using artificial diffusion may result in computing the location of a nonsteady shock incorrectly. This will be shown in the Model Problem II. However, this inaccuracy is acceptable because the method is not sensitive to errors in the initial guess, as long as U^0 is continuous.

7.2. Parallelism. Parallelism may be exploited at several levels in the implementation of Algorithm 2. Consider first the parallelism which may be exploited in the solves for the characteristics. Characteristic x_i^k is obtained at discrete points in time by solving (18) using a Newton iteration. These solves may be scheduled asynchronously, or they may be grouped as vectors. Grouping characteristics and assigning the spatial location of each characteristic in a group to a component of a vector allows the exploitation of vector processing capabilities. Once the location of the characteristic is known for a time step, the value of $\ln(\hat{J}^k)$ may be approximated using (31).

For a large number of processors, a parent process could spawn a task for each characteristic, and allow them to all execute in parallel. In turn, the task solving a particular characteristic could then spawn two tasks, one for the monitoring of the Jacobian, and one for the updating of U^{k+1}. To avoid extra computations, the child task monitoring the Jacobian would need a means of interrupting the parent task computing the characteristic; however, no communication from the task computing U^{k+1} to the parent task is needed. This is reflected by the data dependency graph in Figure 3. The parent routine is labeled OR. The tasks which compute characteristics are labeled C_i, for $i = 1$ to I_β. Each characteristic task spawns two processes, one for the Jacobian monitor, and one to obtain the value of the solution. These are labeled J and U, respectively. The data dependency graph for $\hat{D}_{OR}^k$ is a subgraph of the dependency graph for the whole problem.

For a smaller number of processors, since a large number of characteristics will be computed, the exploitation of the medium grain parallelism outlined above is not needed. Thus, for the implementation on the Alliant FX/8, a single task performs the Newton iteration to determine the new location of x_i^k. Then the same task updates the values of S_i^k and U_i^{k+1} at the new characteristic location.

Other parallelism evident from the description of Algorithm 2 is reflected in Figure 4. Nodes are labeled with the corresponding step number of Algorithm 2. The only sequential step is the location of the base of $\partial\hat{D}_{IL}^k$, and this is a small portion of the overall computations. The major portion of the computations are the characteristic solves in the outer region and the discretization for the internal layer. More efficient methods could possibly be applied in the internal layer, potentially providing more parallelism.

A less obvious source of parallelism is the exploitation of another type of domain

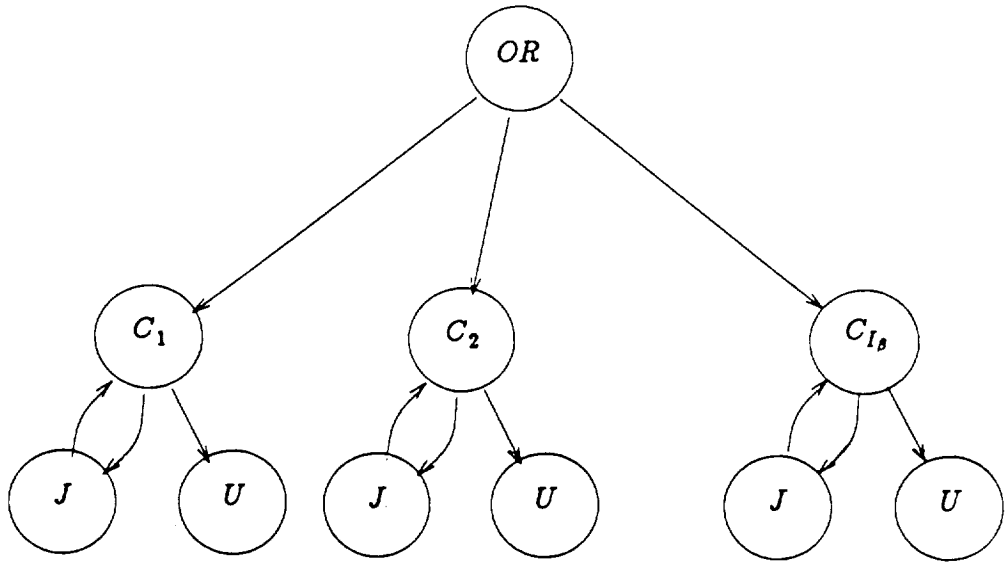

FIG. 3. *Data dependency for characteristic solves.* (From [32].)

decomposition to form a pipeline out of the outer (k) iteration. To obtain U^2 up to time $t = T_1$ requires the knowledge of U^1 for $0 < t \leq T_1$, but not for $t > T_1$. Thus, U^1 for $T_1 < t \leq T_2$ can be computed at the same time that U^2 is determined for $0 < t \leq T_1$. In general, subdivide the domain D in the temporal direction as $0 < T_1 < T_2 \ldots < T_N < T$. While U^1 is being computed for $T_k \leq t \leq T_k + 1$, the computations for U^2 through U^k could be taking place, thus forming a pipeline based on the temporal subdivision of the domain.

7.3. Implementation. Algorithm 2 was implemented on an Alliant FX/8 using a package called Schedule [14]. This package provides a common user interface to the parallel capabilities of a variety of shared memory parallel computers. All of the synchronization required to enforce the data dependencies is automatically provided by the Schedule Package once the graph has been specified correctly. Moreover, there are no machine dependent statements within the user code. All such machine dependencies are internal to Schedule. This provides for transportability of the code between the various machines Schedule has been ported to. The implementation is meant as a demonstration of the viability of the method. Thus, not all of the available parallelism has been exploited. Even so, significant speedups were achieved, and will be discussed in the section on the experiments.

A useful feature of Schedule is the automatic generator of a data flow graph associated with a computation. In Figure 5, the graph Schedule produced for an older version of the code [23] is shown. The nodes represented by circles form the static portion of the data dependency graph (the portion of the graph which does not change for different input data). The rectangular nodes associated with static nodes

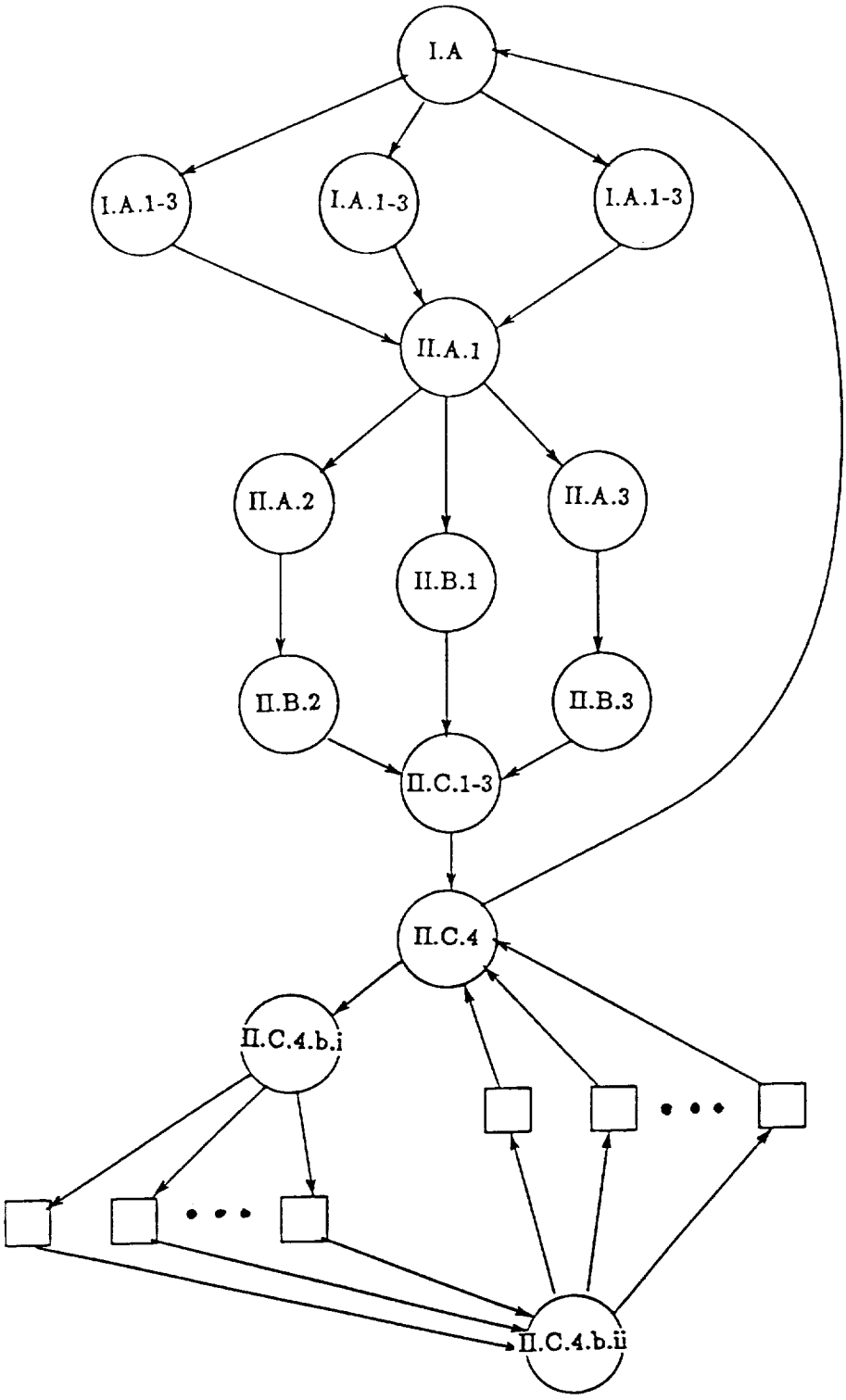

FIG. 4. *Data dependency for algorithm.*

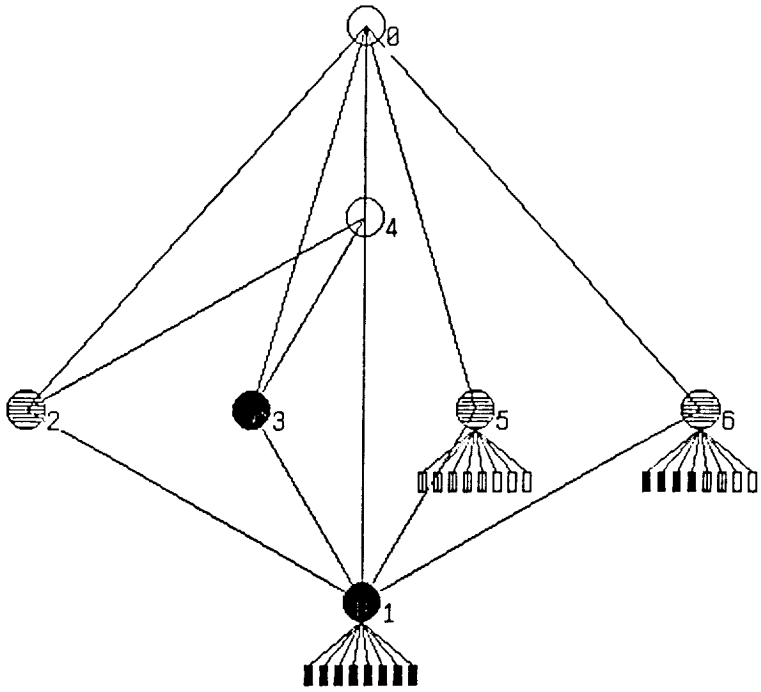

FIG. 5. *Dependency graph generated by Schedule.* (From [23].)

represent dynamically spawned processes which, in this case, are each a characteristic grid line. The graph represents a snapshot of computation shown midway through its execution. The black nodes represent computed processes, the hatched nodes represent active processes, and the white nodes represent processes waiting to execute. Information from this graph and various statistics available with it have been used to improve load balance in this algorithm.

The matrix equation in step II.C.4.b.ii is a symmetric positive definite tridiagonal matrix of about 300 unknowns. Since the number of unknowns is small, the Linpack tridiagonal solver, **sptsl**, [15] was parallelized instead of using a more complicated block scheme. The scheme in **sptsl** uses two dual sweeps of the tridiagonal matrix. The first dual sweep is for the forward elimination, and begins at both the top and bottom of the matrix, then works inward. The backward substitution sweep begins at the middle of the matrix, then works outward in both directions. Thus, the computations can be parallelized for a maximum speedup of two. The implementation used for this method had a speedup of about 1.5 on 2 to 8 Computational Elements (processors) of the FX/8 over the compiler-optimized version of the sequential code.

8. EXPERIMENTS. Two problems are solved, each demonstrating different features of the algorithm. Model Problem I has a steady shock and no forcing term, with an exact solution being known. It demonstrates that the behavior of the error in the computations is the same as the theoretical error estimates of the theorems in the outer regions. Model Problem II has a nonsteady shock. It is used to demonstrate

that the method is not sensitive to the initial guess.

8.1. Model Problem I. The first model problem will demonstrate that the error tends to zero as $\epsilon \downarrow 0$. With no forcing term, (2) is Burgers' equation,

$$u_t + uu_x - \epsilon u_{xx} = 0. \tag{38}$$

The initial and boundary conditions are that

$$u = -2\tanh(x/\epsilon),$$

for $(x,t) \in \Pi$. Under these conditions, the exact solution to (38) is $u = -2\tanh(x/\epsilon)$, hence the computed solution may be compared easily to the exact. The L_1 error is presented for runs with different values of ϵ in Table 1. The error was measured at

Table 1[2] L_1 error

ϵ	Error
10^{-2}	5.0×10^{-02}
10^{-3}	2.2×10^{-03}
10^{-4}	2.3×10^{-04}

time $t = .1$, and remained constant for the remainder of the computational domain. Only one pass of the domain decomposition was needed for this problem. The results presented in Table 1 indicate that the error in the computational method is $O(\epsilon)$.

8.2. Model Problem II. The second model problem has a nonsteady shock, and shows the effect of the forcing term on the location of the shock. In addition, this method demonstrates that the method is not overly sensitive to the accuracy of the initial guess. The equation solved is

$$u_t + uu_x - \epsilon u_{xx} - 4\sin(2\pi x)u = 0,$$

subject to the initial guess

$$u(x,0) = \cos(\pi x), \quad \text{for } 0 < x < 1 = b$$

(see Figure 6). Then the boundary conditions are

$$u(0,t) = 1,\ u(1,t) = -1, \quad \text{for } T > t > 0.$$

For the experiments presented here, $\epsilon = .001$. If there were no forcing term ($r = 0$), then this problem would have a steady shock develop in the center of the domain; however, the forcing term is $r = 4\sin(2\pi x)$. This represents a duct of width $A(x) = \exp[2\cos(2\pi x)/\pi]$, which has the general shape of the one in Figure 1. The effects of the shape of the duct on the location of the shock are reflected by the internal-layer subdomain having a different location as time increases. The movement of the internal

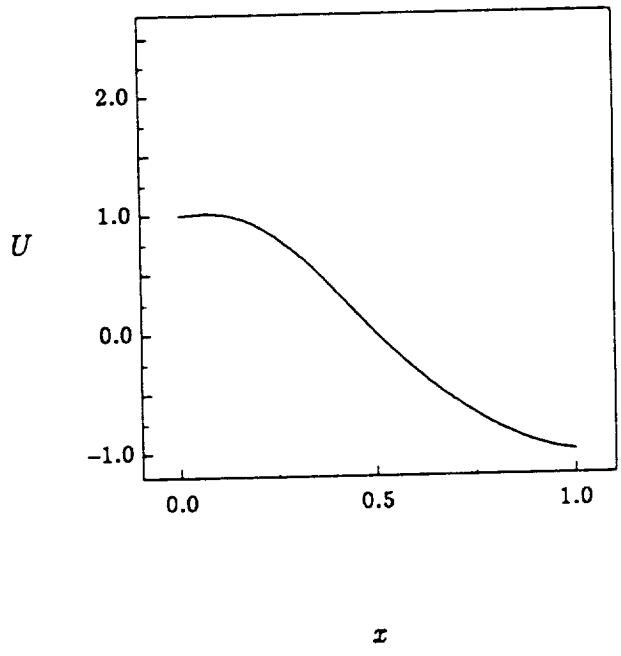

FIG. 6. *Initial guess for iteration.* (From [32].)

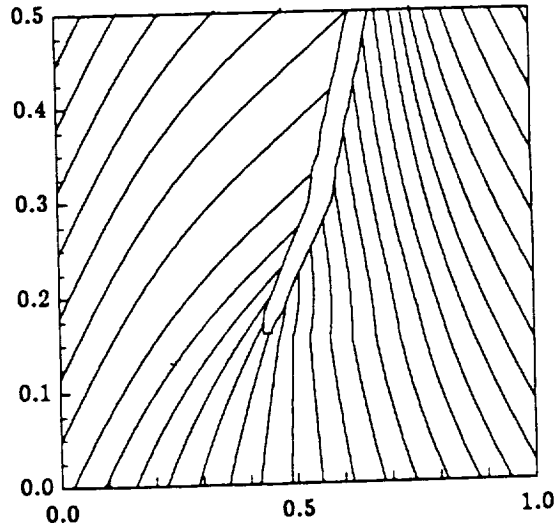

FIG. 7. *Flow lines and domain decomposition.* (From [32].)

layer subdomain can be seen in Figure 7, where the characteristics of the outer-region solution and internal-layer subdomain after four passes of the domain decomposition are shown.

The initial iterate is obtained using the Modified Method of Characteristics on a rectangular grid with 100 spatial points. Figure 8 shows the initial iterate, U^0, at

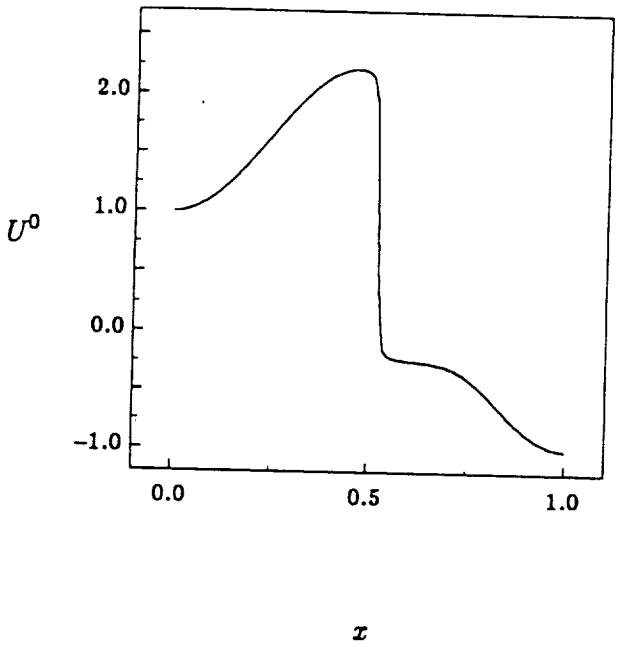

FIG. 8. *Initial guess at* $t = .3$. (From [32].)

time $t = .3$. The use of artificial diffusion resulted in the location and the width of the shock being wrong for the initial iterate. Comparing the initial iterate, U^0, with U^3 in Figure 9, these errors may be seen. The initial guess has the wrong amplitude, and the shock is slightly to the left and wider than the shock in U^3.

The succession of internal-layer subdomains may be seen in Figures 10-13. One of the manifestations of the convergence of the iteration is that the internal-layer subdomain has a much smoother boundary after convergence than before. The boundary for U^4 is smooth up to approximatedly time $t = .3$.

Timings for computing the initial guess and the first iteration are presented in Table 2. Speedup is the execution time for multiple CEs divided by the time for one CE. Efficiency is the speedup divided by the number of CEs. The data indicates that significant speedups were attained, even though a significant portion of the parallelism was not exploited. For example, the pipelining described in Section 7.2 was not used.

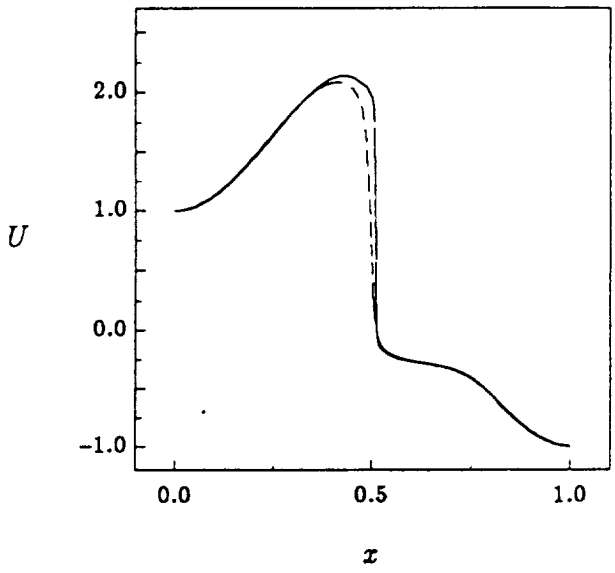

FIG. 9. U^0 *(dotted) and* U^3 *at* $t = .3$. (From [32].)

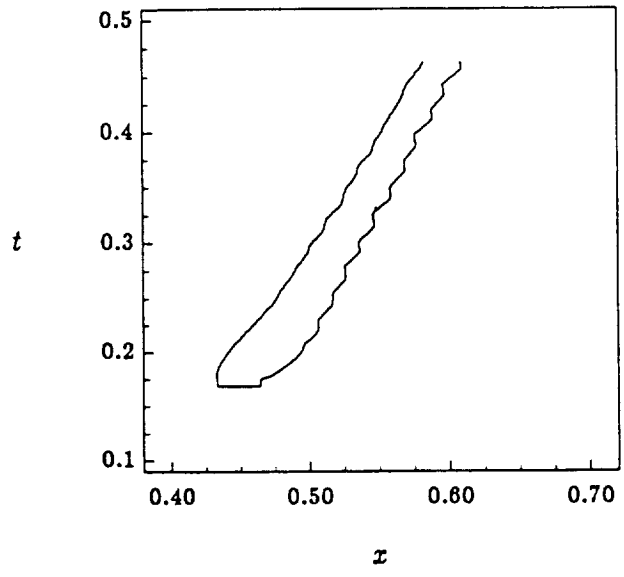

FIG. 10. *Internal-layer boundary for* U^1. (From [32].)

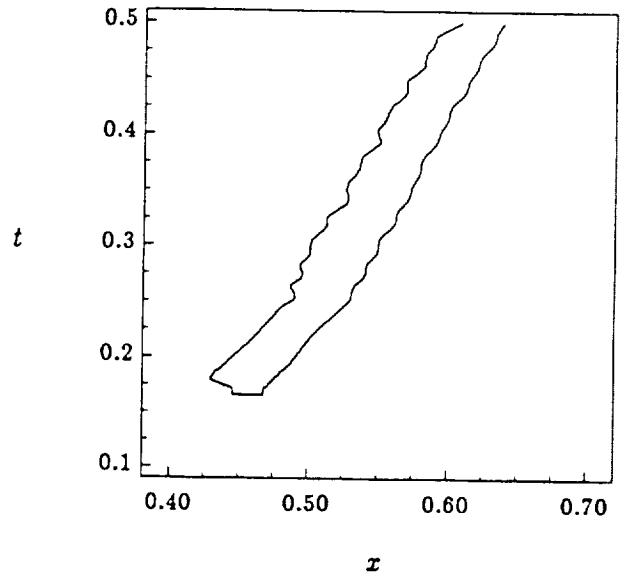

FIG. 11. *Internal-layer boundary for* U^2. (From [32].)

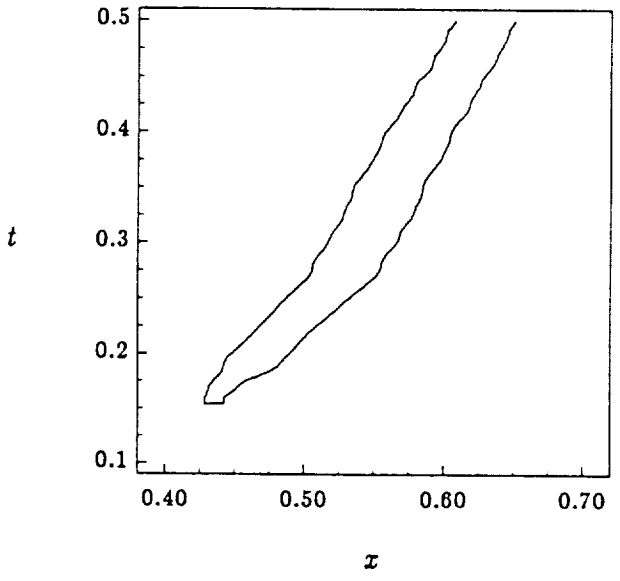

FIG. 12. *Internal-layer boundary for* U^3. (From [32].)

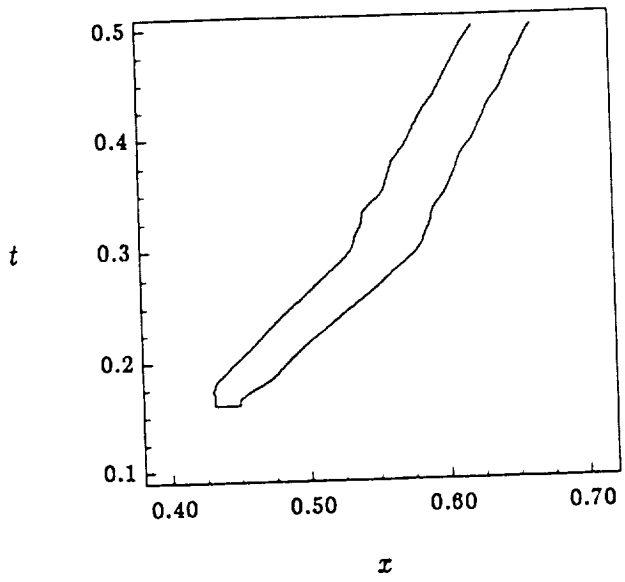

FIG. 13. *Internal-layer boundary for* U^4. (From [32].)

Table 2^3 Timings

CEs (P)	RUN TIME (seconds)	SPEEDUP (S)	EFFICIENCY (E)
1	90.4	1.00	1.00
2	47.5	1.90	0.95
3	34.5	2.61	0.87
4	27.9	3.23	0.81
5	24.3	3.71	0.74
6	21.7	4.16	0.69
7	19.8	4.57	0.65
8	18.5	4.89	0.61

9. CONCLUSION. In this paper, asymptotics and numerics have been blended to form a new computational method. The method has potential to exploit a large amount of parallelism and provides high accuracy. Asymptotic analysis provided a theoretical basis for the domain decomposition, and guided in the derivation of rigorous local and global error bounds.

Two types of subdomains were identified by the asymptotic analysis of this problem: smooth outer regions, and an internal-layer subdomain with a shock. Pipelining the outer iteration provided large-grain parallelism. Smaller-grain parallelism was exploited by using a modification of the method of characteristics in the outer regions, and by blocking the computations in the shock layer.

The method was developed for an important problem in computational fluid dynamics; however, the method is suitable for a wide range of problems in physics and chemistry. Namely, this approach is suitable for problems with internal layers and boundary layers interspersed with regions where the solution is smooth. Examples of such problems other than transonic flow through a duct include the location of stagnation points (flow of zero velocity) where two opposing jets intersect, and combustion fronts.

The availability of estimates and bounds on the error is important in the design of numerical methods. Rigorous *a priori* error bounds were established for the method presented here. In addition, the particular numerical schemes used for the subproblems allowed *a posteriori* error estimation. The *a priori* error bounds were shown to much larger than the errors observed in the computations; thus, sharper error bounds are expected.

ACKNOWLEDGEMENTS.

The authors would like to acknowledge the support of Ahmed Sameh during the research of this project. The development of this method would not have been possible without Gerald Hedstrom, Raymond C. Chin, and Fred Howe's work in the area and collaboration on this project. The authors wish to thank Barbara Stewart for her assistance in the document preparation.

NOTATION

D	(3)	Computational domain.
D_{IL}	(14)	Internal-layer subdomain.
D_{OR}	(15)	Outer-region subdomain.
$\hat{D}_{IL}$	(25)	Computational internal-layer subdomain.
$\hat{D}_{OR}$	(24)	Computational outer-region subdomain.
$L(t)$	Sec. 6.2	Location of the left boundary of $\hat{D}_I L$.
L_n	Sec. 6.2	Computed location of $L(t)$ at t_n.
P	(2)	The full nonlinear operator.
P_0	(8)	The reduced nonlinear operator.
$R(t)$	Sec. 6.2	Location of the right boundary of $\hat{D}_I L$.
R_n	Sec. 6.2	Computed location of $R(t)$ at t_n.
$S_i^k(t)$	(30)	Integral for monitoring Jacobian.
$S_{i,j}^k$	(31)	Discrete values of $S_i^k(t)$.
T	(3)	Upper bound on time t for D.
U^k	(17),(20)	Iterate k.
t^Γ	Sec. 4	Time that the solution to the reduced problem becomes multivalued.
t^*	Sec. 6.2	Scaled and translated internal-layer coordinate.
u	(2)	The solution to the full operator.
$\hat{u}$	(8)	The solution to the reduced operator.
$\tilde{x}$	Sec. 4	Scaled and translated internal-layer coordinate.
x^*	(33)	Spatial coordinate for internal-layer computations.
$\check{x}^*$	(35)	Spatial location used in MMC.
$x^k(t)$	(18)	Characteristic coordinate.
$x_i^k(t)$	Sec (6.1)	Characteristic grid line i for iteration k.
$x_{i,j}^k$	Sec (6.1)	Computed value of x_i^k at time t_n.

REFERENCES

[1] AMIRAM HARTEN, PETER D. LAX, AND BRAM VAN LEER, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Review, 25 (1983), pp. 35–61.

[2] C. C. LIN AND L. A. SEGEL, *Mathematics Applied to Deterministic Problems in the Natural Sciences*, Macmillan, New York, 1974.

[3] C. M. BENDER AND S. A. ORSZAG, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York, 1978.

[4] J. R. CANNON, *The One-Dimensional Heat Equation*, Vol. 23, Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.

[5] DAVID HOFF AND JOEL SMOLLER, *Error bounds for finite-difference approximations for a class of nonlinear parabolic systems*, Math. Comp., 45 (1985), pp. 35–49.

[6] W. ECKHAUS, *Asymptotic Analysis of Singular Perturbations*, North-Holland, Amsterdam, 1979.

[7] D. B. GANNON, *Self adaptive methods for parabolic partial differential equations*, Tech. Rep. PhD. Thesis, University of Illinois, Urbana-Champaign, 1980.

[8] H. C. YEE AND A HARTEN, *Implicit TVD schemes for hyperbolic conservation laws in curvilinear coordinates*, AIAA Journal, 25 (1987), pp. 266–274.

[9] F. A. HOWES, *Perturbed boundary value problems whose reduced solutions are nonsmooth*, Indiana Univ. Math. J., 30 (1981), pp. 267–280.

[10] ——, *Multi-dimensional reaction-convection-diffusion equations*, in Springer Lecture Notes, A. Dold, B. Eckmann, ed., Vol. 1151, Springer-Verlag, New York, 1984, pp. 217–223.

[11] ——, *Multi-dimensional initial-boundary value problems with strong nonlinearities*, Arch. for Rat. Mech. Anal., 91 (1986), pp. 153–168.

[12] ——, *Some stability results for advection-diffusion equations. II*, Studies in Applied Mathematics, 75 (1986), pp. 153–162.

[13] ——, *Asymptotic stability of viscous shock waves*, in Transactions of the Fourth Army Conference on Applied Mathematics and Computing, No. ARO Report 87-1, 1987.

[14] J. J. DONGARRA AND D. C. SORENSEN, *Schedule: Tools for developing and analyzing parallel Fortran programs*, Tech. Rep. ANL/MCS-TM-86, Argonne National Lab, 1986.

[15] J. J. DONGARRA, J. R. BUNCH, C. B. MOLER, AND G. W. STEWART, *LINPACK Users' Guide*, SIAM, Philadelphia, 1979.

[16] J. KEVORKIAN AND J. D. COLE, *Perturbation Methods in Applied Mathematics*, Springer-Verlag, New York, 1981.

[17] K. W. CHANG AND F. A. HOWES, *Nonlinear Singular Perturbation Phenomena: Theory and Applications*, Springer-Verlag, New York, 1984.

[18] P. D. LAX, *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*, in Regional Conference Series in Applied Mathematics, No. 11, 1973.

[19] A. H. NAYFEH, *Perturbation Methods*, Wiley-Interscience, New York, 1973.

[20] A. H. NAYFEH, *Introduction to Perturbation Techniques*, John Wiley and Sons, New York, 1981.

[21] F. W. J. OLVER, *Asymptotics and Special Functions*, Academic Press, New York, 1974.

[22] R. C. Y. CHIN, G. W. HEDSTROM, AND F. A. HOWES, *Moving grid and domain decomposition methods for parabolic partial differential equations*, Tech. Rep. informal report, Lawrence Livermore Natl. Lab., Livermore, CA, 1985.

[23] R. C. Y. CHIN, GERALD W. HEDSTROM, DANNY C. SORENSEN, AND JEFFREY S. SCROGGS, *Parallel computation of a domain decomposition method*, in IMACS 6th International Symposium on Computer Methods for Partial Differential Equations, 1987.

[24] R. C. Y. CHIN, GERALD W. HEDSTROM, JAMES R. MCGRAW, AND F. A. HOWES, *Parallel computation of multiple-scale problems*, in New Computing Environments: Parallel, Vector, and Systolic, SIAM, Philadelphia, 1986, pp. 136–153.

[25] R. E. MEYER AND S. V. PARTER, *Singular Perturbations and Asymptotics*, Academic Press, New York, 1980.

[26] R.C.Y. CHIN AND R. KRASNY, *A hybrid asymptotic-finite element method for stiff two-point*

boundary value problems, SIAM J. Sci. Stat. Comp., 4 (1983), pp. 229–243.

[27] RICHARD E. EWING AND THOMAS F. RUSSELL, *Multistep Galerkin methods along characteristics for convection-diffusion problems*, in Advances in Computer Methods for Partial Differential Equations-IV, R. Vichnevetsky and R. S. Stepleman, eds., IMACS, Rutgers Univeristy, New Brunswick, N.J., 1981, pp. 28–36.

[28] ROBERT D. RICHTMYER AND K. W. MORTON, *Difference Methods for Initial-Value Problems*, Interscience Publischer, John Wiley and Sons, New York, NY, 1967.

[29] T. F. RUSSELL, *Galerkin time stepping along characteristics for Burgers' equation*, in Scientific Computing, R. Stepleman et. al., ed., North-Holland Publishing Company, 1983, pp. 183–192.

[30] ——, *Time stepping along characteristics with incomplete iteration for a Galerkin approximation of miscible displacement in porous media*, SIAM J. Numer. Anal., 22 (1985), pp. 970–1013.

[31] S. MAS-GALLIC AND P. A. RAVIART, *A particle method for first-order symmetric systems*, Numerische Mathematik, 51 (1987), pp. 323–352.

[32] J. S. SCROGGS, *The Solution of a Parabolic Partial Differential Equation via Domain Decomposition: The Synthesis of Asymptotic and Numerical Analysis*, PhD thesis, University of Illinois at U-C, 1988.

[33] V. ERVIN AND W. LAYTON, *On the approximation of derivatives of singularly perturbed boundary value problems*, SIAM J. Sci. Stat. Comp., 8 (1987), pp. 265–277.

[34] M. VAN DYKE, *Perturbation Methods in Fluid Mechanics*, Parabolic Press, Stanford, 1975.

[35] B. VAN LEER, *On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe*, SIAM J. Sci. Stat. Comp., 5 (1984), pp. 1–20.

[36] G. B. WHITHIAM, *Linear and Nonlinear Waves*, John Wiley and Sons, New York, 1974.

NASA National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-181730 ICASE Report No. 88-60		

4. Title and Subtitle	5. Report Date
AN ASYMPTOTIC INDUCED NUMERICAL METHOD FOR THE CONVECTION-DIFFUSION-REACTION EQUATION	October 1988
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Jeffrey S. Scroggs and Danny C. Sorensen	88-60
	10. Work Unit No. 505-90-21-01

9. Performing Organization Name and Address	11. Contract or Grant No.
Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225	NAS1-18605
	13. Type of Report and Period Covered: Contractor Report
12. Sponsoring Agency Name and Address: National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitor:
Richard W. Barnwell

Final Report

Submitted to Mathematics for Large Scale Computing, Julio C. Diaz (ed.)

16. Abstract

A parallel algorithm for the efficient solution of a time dependent reaction convection diffusion equation with small parameter on the diffusion term will be presented. The method is based on a domain decomposition that is dictated by singular perturbation analysis. The analysis is used to determine regions where certain reduced equations may be solved in place of the full equation. Parallelism is evident at two levels. Domain decomposition provides parallelism at the highest level, and within each domain there is ample opportunity to exploit parallelism. Run-time results demonstrate the viability of the method.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
shocks, domain decomposition, asymptotics, parallel processing	34 - Fluid Mechanics and Heat Transfer 64 - Numerical Analysis Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	33	A03