NASA Contractor Report 181919

ICASE Report No. 89-62

ICASE

AN ITERATIVE METHOD FOR SYSTEMS OF NONLINEAR HYPERBOLIC EQUATIONS

Jeffrey S. Scroggs

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

AN ITERATIVE METHOD FOR SYSTEMS
OF NONLINEAR HYPERBOLIC EQUATIONS

Jeffrey S. Scroggs[1]

ABSTRACT

An iterative algorithm for the efficient solution of systems of nonlinear hyperbolic equations is presented. Parallelism is evident at several levels. In the formation of the iteration, the equations are decoupled, thereby providing large grain parallelism. Parallelism may also be exploited within the solves for each equation. Convergence of the iteration is established via a bounding function argument. Experimental results in two-dimensions are presented.

1. INTRODUCTION. An iterative algorithm suitable for the solution of a system of nonlinear hyperbolic partial differential equations in multiple dimensions is discussed. Convergence is established analytically in the continuous case when the solutions are smooth; however, numerical experiments with a system of nonlinar conservation laws demonstrate that convergence is achieved in the discrete case even when the solutions have discontinuities (shocks). This method is an extension of an iterative method for one-dimensional scalar equations [11, 6] to systems of equations in multiple dimensions.

This method decouples the PDEs by linearizing the convection coefficient for a space-time domain. This provides opportunity to exploit large-grain parallelism. In addition, the linearization allows the treatment of some terms in the equations as source theorems, providing more freedom to choose from a wider variety of numerical methods. This could be exploited, for example, when extending the algorithm that couples the method of characteristics with cellular automata [8] from a scalar equation to a system of equations [7]. Cellular automata and the method of characteristics are both sources of parallelism.

The iteration is presented in Section 3. Convergence of the method is established analytically in Section 4, and numerical experiments demonstrate that the solution improves by more than a digit in each iteration in Section 6.

2. PROBLEM. Consider the following initial-boundary value problem

$$(1) \qquad \frac{\partial U}{\partial t} + \sum_{i=1}^{n} F_i(t, x, U) \frac{\partial U}{\partial x_i} + R(t, x, U)U = 0,$$

where $U = (u_1, u_2, ..., u_m)$ is a vector of $m \geq 1$ components. Assume that the spatial domain can be specified by a real valued function $d(x)$ as $\Pi = \{x \mid d(x) < 0\}$. The range of the temporal variable is $0 < t < T$, and the complete domain is denoted $\Omega = (0, T) \times \Pi = \{(t, x) \mid \Delta(t, x) < 0 \}$, where $\Delta(x, t) = td(x)$. The coefficients R and F_i, for $i = 1$ to n, are matrix-valued functions in $\mathbb{R}^{m \times n}$.

Denote by $F_{i,j}$ the jth row of the matrix F_i and denote the lth component of $F_{i,j}$ by $f_{i,j,l}$. Similarly, $r_{j,l}$ denotes the component of R that is in the jth row and l^{th} column.

The data

$$(2) \qquad u_j(t, x) = \alpha_j(t, x)$$

serve as the initial guess when $t = 0$ and $x \in \Pi$, and as the boundary data for the portion of $\partial \Pi$ such that

$$(3) \qquad \psi_j(t, x, U) := F_{i,j}(t, x, U) \bullet \mathrm{grad}(\Delta) < 0.$$

2

Here, grad(Δ) is the gradient of Δ with respect to the spatial variables. This is analogous to the inflow boundary conditions of fluid dynamics. Other types of boundary conditions may be possible [2], but are not studied here.

3. ITERATION. From Equation (1), each component of the solution satisfies

$$(4) \qquad \frac{\partial u_j}{\partial t} + \sum_i^n f_{i,j,j}(t,x,U)\frac{\partial u_j}{\partial x_i} + r_{j,j}(t,x,U)u_j =$$

$$-\sum_i^n \sum_{l \neq j}^m f_{i,j,l}(t,x,U)\frac{\partial u_l}{\partial x_i} - \sum_{l \neq j}^m r_{j,l}(t,x,U)u_l.$$

Let $U^k = (u_1^k,...,u_n^k)$ be the kth iterate. The equation for the jth component is linearized by using the most recent iterate only in the places where the the jth component is differentiated. This results in the equation

$$(5) \qquad \frac{\partial u_j^{k+1}}{\partial t} + \sum_i^n f_{i,j,j}(t,x,U^k)\frac{\partial u_j^{k+1}}{\partial x_i} + r_{j,j}(t,x,U^k)u_j^{k+1} =$$

$$-\sum_i^n \sum_{l \neq j}^m f_{i,j,l}(t,x,U^k)\frac{\partial u_l^k}{\partial x_i} - \sum_{l \neq j}^m r_{j,l}(t,x,U^k)u_l^k.$$

Thus, the equations for each of the iterates have been decoupled. This is analogous to the Jacobi iteration for systems of linear equations. Each u_j^k satisfies (2) for the portion of $\partial\Omega$ such that $\psi_j(t,x,U^{k-1}) < 0$.

Forms of the iteration other than (5) are possible and may be desirable. For example, if parallelism is not important, then it may be advantageous to linearize but not decouple the equations.

4. CONVERGENCE. The convergence of the iteration has its roots in the existence and uniqueness proofs of partial differential equations [1, 9, 4]. However, existence and uniqueness are not the topics of this paper. The iteration defined by Equation (5) is shown to be a contraction provided the domain is restricted to regions where the solution is sufficiently smooth and continuous. The numerical experiments of Section 6 demonstrate that the restriction on the smoothness of U is only for the sake of the analysis, and the iteration will converge under less strict conditions. In the statement of the theorem, $\overline{\Omega}$ is the closure of Ω, and

$$||h(t,x)||_T = \left[\int_0^T \int_\Pi (h(t,x))^2 dx dt\right]^{1/2}$$

3

denotes the L_2 norm in space.

THEOREM 4.1. *Assume that F_i and R depend continuously on their arguments. If the data α_j of Equation (2) and their first derivatives are continuous and the initial guess U^0 is Lipschitz continuous on $\overline{\Omega}$, then for each k there are constants $C^k > 0$ and $\lambda^k > 0$*

$$
(6) \qquad ||u_j^{k+1} - u_j^k||_T < C^k(e^{t\lambda^k} - 1)||u_j^k - u_j^{k-1}||_T
$$

when $0 < t < T$ and for $1 \leq j \leq m$ and finite $k \geq 1$, and U^k is the kth iterate of Equation (5).

First, the boundedness and continuity of the iterates is established.

LEMMA 4.2. *Assume that F_i and R depend continuously on their arguments. Suppose the data α_j of Equation (2) and their first derivatives are continuous, and the iterate U^k is Lipschitz continuous on $\overline{\Omega}$, then U^{k+1} is Lipschitz continuous for $0 < t < T$.*

To establish this lemma the equation governing the iterates will be simplified. Then the iterate will be shown to be bounded, and the continuity of the iterates will be established.

Proof. Equation (5) may be written

$$
(7) \qquad \frac{\partial u_j^{k+1}}{\partial t} + g_{j,j}^k u_j^{k+1} + \sum_{l=1, l \neq j}^{m} g_{l,j}^k u_l^k = 0,
$$

where

$$
g_{l,j}^k = r_{l,j}(t, x, U^k) + \sum_{i=1}^{n} f_{i,l,j}(t, x, U^k) \frac{\partial}{\partial x_i}.
$$

The Lipschitz continuity of U^k and the continuity of the boundary data results in the coefficient $g_{j,j}^k$ and the summation term in the equation being bounded. Since the domain is bounded, iterate U^{k+1} will also be bounded.

An equation governing $\partial u_j^{k+1}/\partial x_i$ may be derived by taking the partial derivative of Equation (7) with respect to x_i to obtain

$$
(8) \qquad \frac{\partial}{\partial t}\left(\frac{\partial u_j^{k+1}}{\partial x_i}\right) + g_{j,j}^k\left(\frac{\partial u_j^{k+1}}{\partial x_i}\right) + S = 0,
$$

where the source term S and the coefficient $g_{j,j}^k$ of this equation are functions of u_j^{k+1} and the values and first derivatives of R, F_i, and U^k. The boundedness of u_j^{k+1} has been

4

established, and the boundedness of the remaining terms follow from the assumptions made for this lemma. Thus, the boundedness of S and $g_{j,j}^k$ follow. From the boundedness of these terms the boundedness of $\partial u_j^{k+1}/\partial x_i$ (and hence the Lipschitz continuity of u_j^{k+1}) follow. $\square$

To establish Theorem 4.1, an equation governing the difference between iterates is derived. Since the iterates are Lipschitz continuous and the domain is bounded, appropriate upper and lower bounds for U^{k+1} will be determined. Then the difference is shown to be bounded with solutions of ordinary differential equations.

Proof (of Theorem 4.1). The components of the difference $Z^k = (z_1^k, ..., z_m^k)$, where $z_j^k = u_j^{k+1} - u_j^k$ are governed by the equation

$$(9) \qquad \frac{\partial z_j^k}{\partial t} + \sum_{l \neq j}[g_{l,j}^k z_l^{k-1} + (g_{l,j}^k - g_{l,j}^{k-1})u_l^{k-1}] + g_{j,j}^k z_j^k + (g_{j,j}^k - g_{j,j}^{k-1})u_j^k = 0.$$

The initial data is $z_j^k(0, x) = 0$. By the Mean Value Theorem

$$g_{l,j}^k - g_{l,j}^{k-1} = Z^{k-1} \bullet G_{l,j}^k,$$

where

$$G_{l,j}^k = \nabla_u r_{l,j}(t, x, \hat{U}) + \sum_{i=1}^n \nabla_u f_{i,l,j}(t, x, \hat{U})\frac{\partial}{\partial x_i}.$$

Here, $\hat{U}$ is some function bounded above and below by $max(U^{k-1}, U^k)$ and $min(U^{k-1}, U^k)$, respectively. Thus, Equation (9) may be written as

$$(10) \qquad \frac{\partial z_j^k}{\partial t} + g_{j,j}^k z_j^k = -Z^{k-1} \bullet \left[G_{j,j}^k u_j^k + \sum_{l \neq j} G_{l,j}^k u_l^{k-1} \right] - \sum_{l \neq j} g_{l,j}^k z_l^{k-1}.$$

Using the lemma, the absolute value of the terms on the right hand side of Equation (10) are bounded by $C_j^k\|Z^{k-1}\|$, for some constant C_j^k. In addition, the coefficient of z_j^k is bounded above (below) by some constant $\overline{\lambda^k}$ ($\underline{\lambda^k}$). Thus, the functions

$$\overline{\omega_j} = tC_j^k\|Z^{k-1}\|_T e^{\overline{\lambda^k}t}$$

and

$$\underline{\omega_j} = tC_j^k\|Z^{k-1}\|_T e^{\underline{\lambda^k}t}$$

are upper and lower bounds, respectively, for $\|z_j^k\|$. With $C^k = max_j C_j^k$ and $\lambda^k = max_j \lambda_j^k$, the theorem is established. $\square$

5

Theorem 4.1 did not establish the boundedness of C^k and λ^k as k goes to infinity. If these constants did grow unbounded for large k, then the size of the temporal partition for which the iteration would converge will tend to zero. In practice, the iteration converged in a finite number of iterations, and the effect of unbounded growth in C^k and λ^k was not experienced. When these constants are assumed to be bounded, as in the corollary below, the convergence of the iteration follows immediately from the theorem.

COROLLARY 4.3. *Assume that F_i and R depend continuously on their arguments. Also Assume that the data α_j of Equation (2) and their first derivatives are continuous and the initial guess U^0 is Lipschitz continuous on $\overline{\Omega}$. If there are bounds $C > C^k$ and $\lambda > \lambda^k$ for all $k >= 1$, then then the iteration defined by (5) converges to a solution of (1)*.

Proof. From Theorem 4.1,

$$\|u_j^{k+1} - u_j^k\|_T < C(e^{t\lambda} - 1)\|u_j^k - u_j^{k-1}\|_T.$$

It is possible to choose a temporal bound $\hat{T}$ such that $C(e^{t\lambda} - 1) < 1$ when $T < \hat{T}$, and the iteration is a contraction.

Since the iteration is a contraction, it must converge to a fixed point $U^\infty = U^k = U^{k+1}$. Putting U^∞ into Equation (5), clearly $U = U^\infty$ is a solution to (1)-(2). □

Thus, there are reasonable conditions under which the iteration will converge. Stronger results would likely be possible, especially considering that the experiments presented later in the paper do not satisfy all of the conditions of the analytic results.

5. ALGORITHM. An algorithm based on the iteration is presented here. The algorithm is a combination of the iteration presented in Section 3 with a temporal partitioning and a stopping criteria.

The temporal variable may need to be partitioned into several regions. For example, it is possible for the method to diverge if T is too large. This is manifested by the coefficient $C^k(e^{t\lambda^k} - 1)$ of Equation (6) in Theorem 4.1 becoming larger than unit. In addition, the iteration requires that the solution be stored for the entire temporal region, possibly requiring too much memory. These problems are resolved by partitioning time into Q sections $0 < T_0 < T_1 < ... < T_Q = T$. The iteration will be performed on the partition $\Omega_q = (T_{q-1}, T_q] \times \Pi$ of the domain, using the solution at time $t = T_{q-1}$ from the iteration on Ω_{q-1} as the initial condition.

The stopping criteria is based on the norm of the difference between iterates at time $t = T_q$. When the norm is less than some user specified tolerance, then the iteration is

assumed to have converged.

I. Initialize.
    A. Set temporal partition counter to $q = 1$.
    B. Apply initial data (2) to the solution.
II. Determine the solution on temporal partition q.
    A. Doall $j = 1$ to n (j is the index for the elements of the solution).
        1. Determine initial guess u_j^0 for $(T_{q-1}, T_q) \times \Pi$.
    B. Initialize iteration counter $k = 1$.
    C. Doall $j = 1$ to n (do one iteration).
        1. Solve Equation (5) to obtain u_j^k then
        2. compute the norm of the difference $\kappa_j^k = ||u_j^k - u_j^{k-1}||$.
    D. If $K^k = \sum_j \kappa_j^k > \text{TOL}$
        1. Then increment k and go to Step C.
        2. Else finished with this temporal partition.
            a. Restart by using the solution at $t = T_{q-1}$ as initial conditions.
            b. Increment q and goto Step II.

Algorithm 1 Iteration with Restart.

The algorithm is independent of the the discretization used in Step II.C.1 and of the initial guess used in Step II.A.1 (see Algorithm 1). Since the coefficients have nonlinear behavior, discretizations should be chosen that are appropriate for nonlinear problems. Notice that if an explicit discretization is used with a temporal partition that contain only a single time step, then the algorithm is no different than applying the explicit discretization directly to the linearization of Equation (1). Thus, each temporal partition should contain at least two time steps.

There is potential to exploit parallelism on several levels. Large-grain parallelism is available through the decoupling of the equations, and may be further enhanced by using domain decomposition techniques (see [10]). Smaller-grain parallelism may be exploited by using appropriate data structures for each of the subdomains [3, 5], and by choosing an appropriate numerical scheme to solve the PDEs. Exploitation of parallelism is an area of further research.

6. NUMERICAL EXPERIMENTS. In this section, Algorithm 1 is used to solve a system of nonlinear hyperbolic equations. These numerical experiments are a demonstration of the convergence of the algorithm. The implementation of the algorithm has not been optimized or parallelized; thus only the convergence results are presented. Experiments also examine the effects of the size of the temporal partition on the number of iterations. All the experiments were performed on an Ardent Titan, and required less than 5 minutes execution time for each run.

7

The equations solved are a two-dimensional form of Burgers' equation

(11)
$$u_t + uu_x + vu_y = 0$$

(12)
$$v_t + uv_x + vv_y = 0,$$

where time has the range $0 < t < 1$ and the spatial domain is the unit cube $\Pi = \{(x,y) \mid 0 > \Delta(x,y) = t(max[(x-1/2)^2, (y-1/2)^2] - 1/4)\}$. The equations are capable of simulation of some of the physical phenomena that arise in computational fluid dynamics. Namely, with initial and boundary data

$$u = 2.0(.5 - y),$$

$$v = .1 + .9x,$$

Equations (11-12) will be used to model shocks. The data are applied for $t = 0$ and for the inflow portion of Π which is defined by $y = 0$, $y = 1$, and $x = 1$. The solution is smooth initially, and develops a shock (see Figures 1-2). Thus, the examples satisfy all
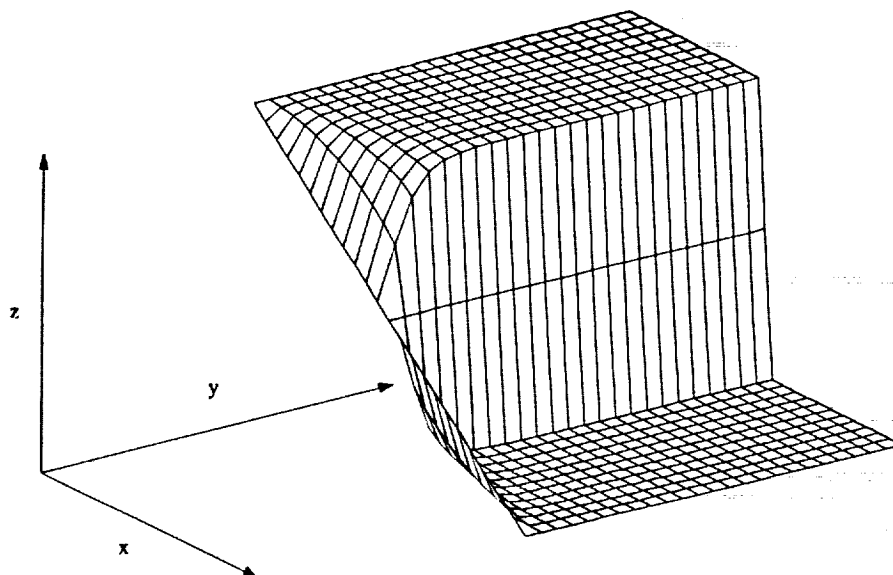

FIG. 1. *U at t = .8.*

of the conditions of the theoretical results for the initial partition Ω_1, and evolves into
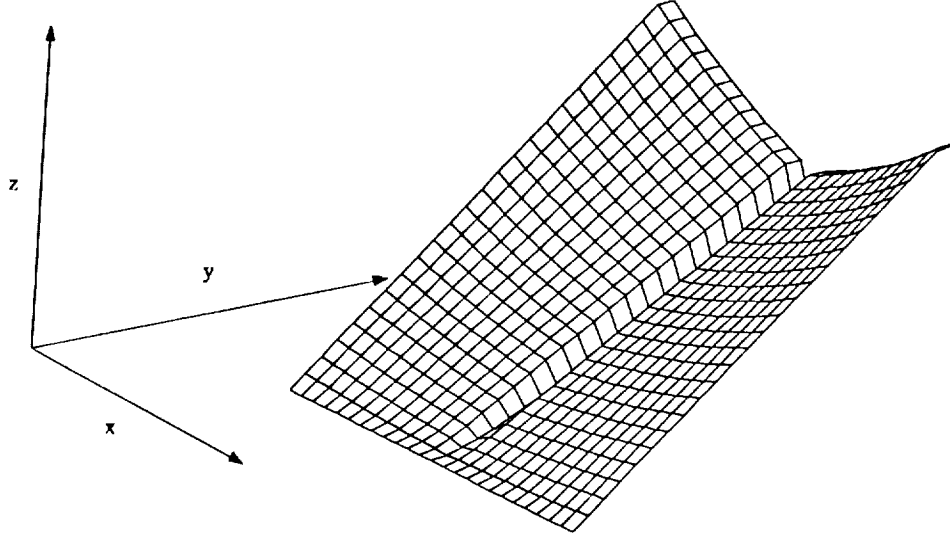
8

FIG. 2. V at $t = .8$.

a nonsmooth function that violates these conditions. Thus, the experimental results will demonstrate that the algorithm works even when some of the conditions used to established the theory are violated.

The experiments were performed using two different explicit discretizations, a second-order MacCormack scheme [2], and a first-order upwind scheme. The upwind scheme consists of applying a backward or forward difference based on the sign of the coefficient of the spatial derivative being differenced.

To improve computational efficiency, the discrete analog of the L_1 norm at time $t = T_q$ is used in place of the L_2 norm in Step II.C.2. Thus,

$$(13) \qquad \kappa_j^k = \frac{1}{P} \sum_{p=1}^{P} |u_{j,p}^k - u_{j,p}^{k-1}|,$$

where P is the number of spatial grid points and $u_{j,p}^k$ is the discrete value of component j of iterate k. There were 25 unknowns in each direction including boundary points; thus, the total number of points was $P = 625$. The time step was $\Delta t = .002$.

The initial guess on Ω_q is

$$u^0(t, x, y) = u^k(T_{q-1}, x, y)$$

9

and

$$v^0(t, x, y) = v^k(T_{q-1}, x, y),$$

where (u^k, v^k) was the last iterate for partition Ω_{q-1}. This resulted in a good initial guess as measured by the norm K^1 of the difference between the initial guess and the first iterate (see Table 1).

Table 1. Average Number of Iterations

Number of	Upwind		MacCormack	
Time Steps	K^1	Num Iter	K^1	Num Iter
20	9.94×10^{-3}	3.8	2.93×10^{-2}	4.1
50	2.43×10^{-2}	4.7	7.12×10^{-2}	6.7
100	4.69×10^{-2}	5.6	1.38×10^{-1}	8.6

The results in Table 1 show the effects of varying the temporal partition size on the convergence of the method with the initial guess as described above. The data are the average over the total number of temporal partitions. Thus, corresponding to the number of time steps of 20, 50, and 100 are partition sizes $T_q - T_{q-1} = .04, .1,$ and $.2$, respectively. The results in the Number of Iterations columns are the data averaged over all of the temporal partitions required to take the 500 time steps necessary to obtain the solution at $T = 1$. This means that the numbers are averaged over 50, 10, and 5 values for the rows with 20, 50, and 100 number of time steps, respectively. Experiments using a constant initial guess confirm that the additional iterations required for the larger partitions are not due to the larger beginning norm K^1. Further work in this area is necessary to establish the reason for this trend; however, the larger computational domain resulting from the large temporal partition is a likely cause.

Linear convergence is demonstrated by the data from numerical experiments in Table 2. A steady shock has formed in the solution by time $t = .8$; thus, the data in the

Table 2. Reduction in Norm for all iterations

Iteration	Num Stopped	Ave Reduction
2	0	6.99×10^{-2}
3	3	5.48×10^{-2}
4	1	5.87×10^{-2}
5	3	4.09×10^{-2}
6	3	2.88×10^{-2}

table reflect that the reduction in the difference was still linear when the solution has

large gradients that violate some of the conditions of the theorems. The tolerance TOL for the stopping criteria was set to 2×10^{-7}, slightly lower than machine precision. The iteration on each partition used the TOL stopping criteria. This resulted in iterations on various temporal partitionings stopping at different iterates. The number of iterates on a partition that stopped at the iteration number in the left column is reported in the 'Num Stopped' column. The average reduction column is thus the relative reduction K^k/K^{k-1} as measured over the number of times the iteration made it that far.

The experiments confirm and go beyond the analytic results. The reduction in the error for each component of the solution in each iteration is monotonic, as predicted by Theorem 4.1 was confirmed. The experiments went beyond the analytic results by showing a linear reduction of slightly more than a digit of accuracy on a model problem with a shock. Larger temporal partitions required more iterations. This indicates that the larger partitions will require more computations, but may require fewer synchronizations. To reach any conclusions, this issue could be studied once the algorithm has been implemented in a parallel processing environment.

7. CONCLUSION. An iterative algorithm for the efficient solution of systems of nonlinear hyperbolic equations has been discussed. The method is simple to implement, and has parallelism that can be exploited on several levels. In this paper, convergence was established analytically for continuous solutions. Numerical experiments demonstrated the monotone convergence predicted by the analysis, both when the solution was smooth, and when the problem had shocks resulting from the nonlinearities.

REFERENCES

[1] D. L. BERNSTEIN, *Existence Theorems in Partial Differential Equations*, Princeton University Press, Princeton, New Jersey, 1950.

[2] D. A. ANDERSON, J. C. TANNEHILL, AND R. H. PLETCHER, *Computation Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corp., McGraw-Hill, New York, 1984.

[3] D. GANNON AND J. VAN ROSENDALE, *On the structure of parallelism in a highly concurrent pde solver*, J. Par. Dis. Comp., 3 (1986), pp. 106–135.

[4] G. S. LADDE, V. LAKSHMIKANTHAM, AND A. S. VATSALA, *Monotone Iterative Techniques for Nonlinear Differential Equations*, Pitman Publishing, Inc., Marshfield, Massachusetts, 1985.

[5] W. D. GROPP, *Local uniform mesh refinement with moving grids*, SIAM J. Sci. Stat. Comp., 8 (1987), pp. 292–304.

[6] J. S. SCROGGS, *A physically motivated domain decomposition for singularly perturbed equations*, SIAM Journal on Numerical Analysis, submitted (1989).

[7] M. GARBEY, *Private communications*, 1989.

[8] M. GARBEY AND D. LEVINE, *Massively parallel computation of conservation laws*, Preprint MCS-P85-0689, MCS, Argonne National Lab, June 1989.

[9] R. COURANT AND D. HILBERT, *Methods of Mathematical Physics, Volumes I and II*, Interscience, New York, 1962.

[10] R. GLOWINSKI, G. H. GOLUB, G. A. MEURANT, AND J. PERIAUX, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadel-

phia, PA, 1988.

[11] 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.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA CR-181919 ICASE Report No. 89-62	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle AN ITERATIVE METHOD FOR SYSTEMS OF NONLINEAR HYPERBOLIC EQUATIONS		5. Report Date August 1989
		6. Performing Organization Code
7. Author(s) Jeffrey S. Scroggs		8. Performing Organization Report No. 89-62
		10. Work Unit No. 505-90-21-01
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225		11. Contract or Grant No. 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: Submitted to Applied
Richard W. Barnwell   Numerical Analysis

Final Report

16. Abstract

An iterative algorithm for the efficient solution of systems of nonlinear hyperbolic equations is presented Parallelism is evident at several levles. In the formation of the iteration, the equations are decoupled, thereby providing large grain parallelism. Parallelism may also be exploited within the solves for each equation. Convergence of the iteration is established via a bounding function argument. Experimental results in two-dimensions are presented.

17. Key Words (Suggested by Author(s)) nonlinear, hyperbolic, iteration, parallel computing	18. Distribution Statement 64 - Numerical Analysis Unclassified - Unlimited		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 13	22. Price A03

NASA FORM 1626 OCT 86