# ADAPTIVE LOCAL GRID REFINEMENT ALGORITHMS FOR FINITE-ELEMENT COLLOCATION

M.B. Allen
M.C. Curran

1989
Journal Article WWRC-89-24

In

Numerical Methods for Partial Differential Equations

Volume 5

M.B. Allen
M.C. Curran

Department of Mathematics
University of Wyoming
Laramie, Wyoming

# Adaptive Local Grid Refinement Algorithms for Finite-Element Collocation

**Myron B. Allen and Mark C. Curran**
*Department of Mathematics, University of Wyoming, Laramie, Wyoming 82071*

An adaptive grid refinement procedure allows accurate solutions to advection-dominated, time-dependent flows using finite-element collocation. The technique relies on a data structure that is readily amenable to parallel computing. The paper discusses computational aspects of the method.

## I. INTRODUCTION

Adaptive gridding offers an important class of strategies for computing accurate solutions to highly advective fluid flows. We present an adaptive local grid refinement scheme for use in finite-element collocation models for such flows. Of special interest here are the algorithmic aspects of the procedure, which is readily amenable to implementation on parallel-architecture computers. We focus on transient flows in one space dimension. The paper has the following structure: Section II briefly reviews finite-element collocation on fixed grids; Section III discusses the grid-refinement algorithm for the linear advection-diffusion equation; Section IV extends the algorithm to nonlinear problems using Burgers' equation as an example; Section V concludes the paper with an examination of the method's performance on a parallel computer.

## II. REVIEW OF FINITE-ELEMENT COLLOCATION

The method of finite-element collocation has its roots in the engineering literature of the 1930s (see [1]), but we owe the modern version to de Boor and Swartz [2] and Douglas and Dupont [3], among others. For purposes of illustration, consider the constant-coefficient advection-diffusion problem posed on the spatial domain $\Omega = (0,L)$:

$$\frac{\partial u}{\partial t} + v\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2} = 0, \qquad (x,t) \in \Omega \times (0,\infty), \tag{1a}$$

$$u(x,0) = u_I(x), \qquad x \in \Omega, \tag{1b}$$

$$u(0,t) = \mathrm{u}_0, \qquad \frac{\partial u}{\partial x}(L,t) = u_N', \qquad t \geq 0. \tag{1c}$$

Here, $v > 0$ represents fluid velocity, $D > 0$ is a diffusion coefficient, and $u = u(x,t)$ stands for an unknown function, say, solute concentration. We shall apply finite-element collocation to the Crank-Nicolson semidiscrete analog

$$u^{n+1} - u^n + k\left(v\frac{du^{n+(1/2)}}{dx} - D\frac{d^2u^{n+(1/2)}}{dx^2}\right) = 0,$$

where the superscripts indicate time level, $(\cdot)^{n+(1/2)} \equiv \frac{1}{2}[(\cdot)^{n+1} + (\cdot)^n]$, and $k$ signifies the time step.

We begin by establishing a spatial grid $\Delta^0 = \{0 = x_0, h = x_1, \cdots, Nh = x_N = L\}$, and call $[x_{i-1}, x_i] = \overline{\Omega}_i$. In later sections, $\Delta^0$ will be the *coarse grid*, and $\overline{\Omega}_i$ will be the $i$th *coarse-grid element*. The space of Hermite piecewise cubics for the grid $\Delta^0$ on $\overline{\Omega} = [0,1]$ is

$$\mathcal{M}_1^3(\Delta^0) = \{f \in C^1(\overline{\Omega}) | f|\overline{\Omega}_i \text{ is cubic}\}.$$

In other words, $f$ is cubic on each subinterval $\overline{\Omega}_i$ and, globally, is continuously differentiable. This order of continuity is the lowest for which one can use collocation on a second-order differential equation (Birkhoff and Lynch [4], p. 200).

The space $\mathcal{M}_1^3(\Delta^0)$ has an interpolating basis $\{H_{i,0}, H_{i,1}\}_{i=0}^N$ in which the support of each function $H_{i,j}(x)$ is a small subset of $\overline{\Omega} = [0,L]$ consisting of at most two adjacent subintervals, $\overline{\Omega}_{i-1} \cup \overline{\Omega}_i$ (Prenter [5], Chapter 3). In terms of this basis, we can write any $f_0 \in \mathcal{M}_1^3(\Delta^0)$ as a linear combination involving values of $f$ and $f'$ at the nodes of $\Delta^0$:

$$f(x) = \sum_{i=0}^{N} [f(x_i)H_{0,i}(x) + f'(x_i)H_{1,i}(x)].$$

In fact, for any $g \in C^1(\overline{\Omega})$, we can define a projection onto $\mathcal{M}_1^3(\Delta^0)$ as

$$(\pi^0 g)(x) = \sum_{i=0}^{N} [g(x_i)H_{0,i}(x) + g'(x_i)H_{1,i}(x)].$$

To solve the semidiscrete analog of the problem (1), we determine a sequence $\{\hat{u}^n\}_{n=0}^{\infty}$ by first imposing initial and boundary conditions:

$$\hat{u}^0(x) = \pi^0 u_I(x) \quad \forall\, x \in \overline{\Omega};$$

$$\hat{u}^n(0) = u_0; \qquad \frac{d\hat{u}^n}{dx}(L) = u_N', \qquad n = 1,2,\cdots.$$

These criteria specify $\hat{u}^0$ completely and determine two of the $2N \times 2$ nodal degrees of freedom for $\hat{u}^1, \hat{u}^2, \cdots$. To determine the remaining $2N$ degrees of freedom at each time level $n+1$, we first form the residual

$$R^{n+1} = \hat{u}^{n+1} - \hat{u}^n + k\left(v\frac{d\hat{u}^{n+(1/2)}}{dx} - D\frac{d^2\hat{u}^{n+(1/2)}}{dx^2}\right).$$

We then pick a collection $\{\overline{x}_1, \cdots, \overline{x}_{2N}\} \subset \Omega$ *of collocation points* and force $R^{n+1}(\overline{x}_k) = 0, k = 1, \cdots, 2N$. Douglas and Dupont [3] show that one can obtain optimal-order error estimates of the form $\|\hat{u}^n - u(\cdot, nk)\|_\infty = \mathcal{O}(h^4)$ by choosing the $\overline{x}_k$ to be the two-point Gauss-quadrature abscissae in each element $\Omega_i$.

Allen and Pinder [6] demonstrate an upstream-weighted technique assigning precisely these collocation points to all terms in the residual except the advection terms $kvd\hat{u}^n/dx$, for which the "collocation" points have the form $\bar{x}_k^* = \bar{x}_k - \zeta h, \zeta > 0$.

Despite the smoothness required of the trial function $\hat{u}^n$, two features of collocation make it an attractive scheme for modeling transient, advection-dominated flows. First, the matrix for the system of collocation equations at each time level has bandwidth five in one space dimension and is therefore sparser than the matrices arising from other fourth-order finite-element schemes. The price paid for this sparseness is a loss of symmetry in the matrix equations approximating self-adjoint problems—a penalty that is irrelevant in advective problems, since they are generally nonself-adjoint. Second, in contrast with classical Galerkin formulations, computing the collocation matrix requires neither the calculation of integrals nor formal assembly of a global matrix from local element matrices. This latter fact makes the method especially useful in transient, nonlinear problems, which typically require the computation of a new matrix at each iteration of each time step.

## III. THE ADVECTION-DIFFUSION EQUATION

Finite element collocation, like other discrete methods, tends to yield unacceptable results for the advection-diffusion equation when the Peclet number $P = vL/D \gg 1$. In its standard $\mathcal{O}(h^4)$ version, collocation yields spuriously oscillatory solutions near sharp fronts unless $h < \sqrt{12}/P$ (Jensen and Finlayson [7]). On the other hand, the upstream collocation scheme just cited smears sharp fronts as a consequence of a numerical diffusion coefficient proportional to $Ph\zeta$ (Allen [8]). Figure 1 illustrates these types of error. When $P \gg 1$, using a uniform grid $\Delta^0$ fine enough to mitigate these errors can be expensive. One way around this dilemma is to adjust $h$ *locally,* so that the grid spacing is small only in regions where the solution exhibits sharp fronts needing fine-scale spatial resoluton. Since the sharp fronts move, it is necessary to refine the grid *adaptively,* so that the refined zone follows the front.

Toward this end, we construct a sequence $\{\Delta^n\}_{n=0}^{\infty}$ of grids, each associated with a time level $n$. For computational convenience we demand that each $\Delta^n \supset \Delta^0$, so that the variables associated with the original coarse grid $\Delta^0$ are present at every time level. Thus at each time level $n$ we construct a mapping $\nu^n: \{1, \cdots, N\} \rightarrow \{0, 1, 2, \cdots\}$ assigning $\nu^n(i)$ new nodes, assumed evenly spaced, to each coarse-grid element $\bar{\Omega}_i = [x_{i-1}, x_i]$ formed by $\Delta^0$. To avoid unnecessary computational effort, we want $\nu^n(i) = 0$ except when $\bar{\Omega}_i$ lies near a sharp front. In these exceptional cases, we determine $\nu^n(i)$ according to a grid-refinement strategy appropriate for the equation being solved. We denote by $Z^n = \sum_{i=1}^{N} \nu^n(i)$ the total number of new nodes added at time level $n$. Also, we associate with each grid $\Delta^n$ a trial space $\mathcal{M}_1^3(\Delta^n)$ and a corresponding projection $\pi^n: C^1(\bar{\Omega}) \rightarrow \mathcal{M}_1^3(\Delta^n)$ mapping continuously differentiable functions onto that trial space. Since the polynomial degree of the finite-element approximation remains constant while the grid spacing changes, this scheme is an example of *h-refinement.*

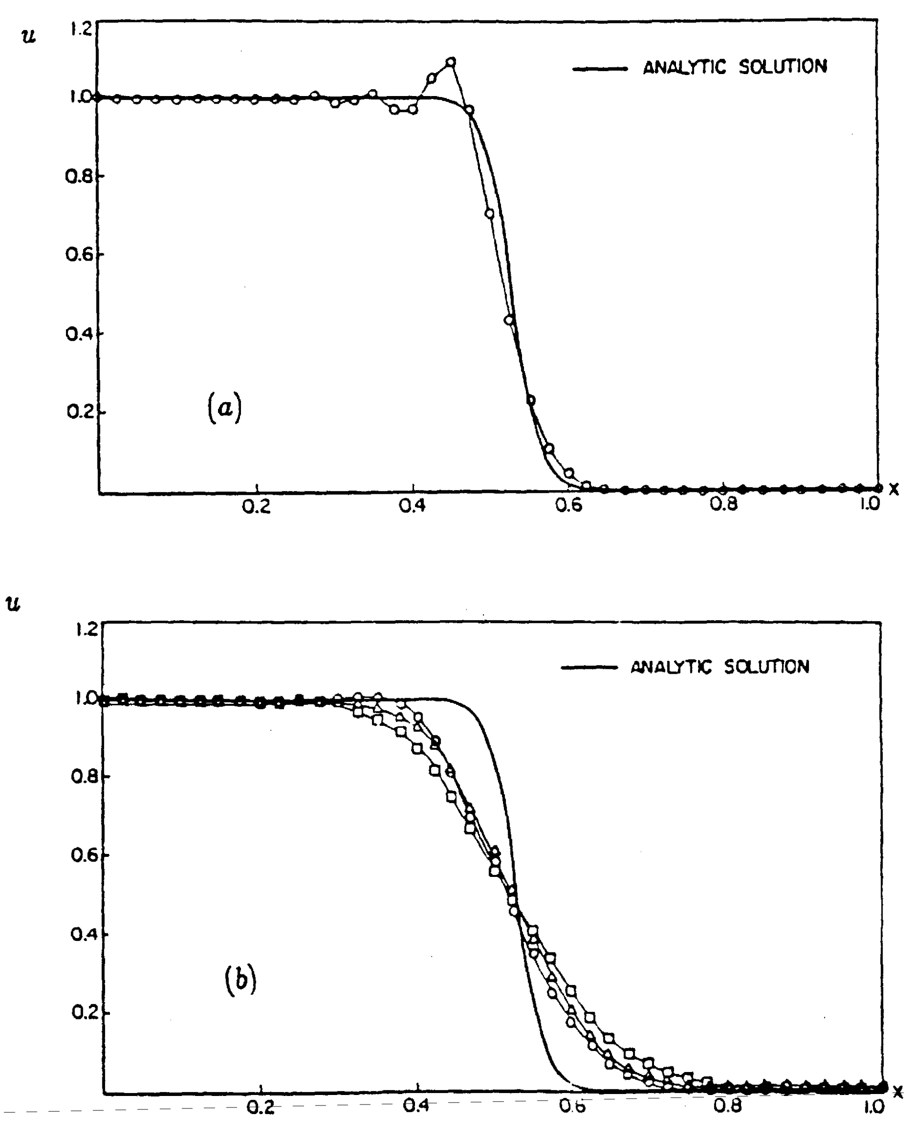

FIG. 1. Spurious oscillations and numerical diffusion associated with (a) standard and (b) upstream-weighted collocation solutions to the advection-diffusion equation. In all cases, $h = 1/40$, $k = 1/20$, $P = 1069$.

We now collocate as before to determine a sequence

$$\{\hat{u} \in \mathcal{M}_1^3(\Delta^0), \hat{u}^1 \in \mathcal{M}_1^3(\Delta^1), \cdots\},$$

using the $2(N + Z^{n+1})$ Gauss abscissae for $\Delta^{n+1}$ as collocation points to solve for the unknown Hermite coordinates of $\hat{u}^{n+1}$. One new wrinkle is that we must project the old solution $\hat{u}^n \in \mathcal{M}_1^3(\Delta^n)$ forward to the new trial space $\mathcal{M}_1^3(\Delta^{n+1})$ to form the residual, getting collocation equations that have the form

$$\hat{u}^{n+1}(\bar{x}_k) + \frac{k}{2}\left[v\frac{d\hat{u}^{n+1}}{dx}(\bar{x}_k) - D\frac{d^2\hat{u}^{n+1}}{dx^2}(\bar{x}_k)\right]$$
$$= (\pi^{n+1}\hat{u}^n)(\bar{x}_k) - \frac{k}{2}\left[v\frac{d}{dx}(\pi^{n+1}\hat{u}^n)(\bar{x}_k) - D\frac{d^2}{dx^2}(\pi^{n+1}\hat{u}^n)(\bar{x}_k)\right].$$

There is another new wrinkle. The addition of $Z^n$ new nodes, and hence $2Z^n$ new unknowns and equations, disrupts the matrix structure associated with collocation on $\Delta^0$. If we have an efficient matrix solver for the structure associated with $\Delta^0$, then it makes sense to decouple the equations associated with newly added nodal parameters of $\hat{u}^{n+1}$, leaving a system having the original structure for the $2N$ coarse-grid unknowns along with a set of smaller systems for the $2Z^{n+1}$ new unknowns the construction of a *p-refinement* scheme for collocation, in which they improve spatial resolution by increasing the local polynomial degree of the aproximation.

We accomplish the decoupling in an elementwise fashion, using sparse row reduction on each of the augmented equation sets associated with refined coarse-grid elements $\overline{\Omega}_i$. At a typical time level $n + 1$, the procedure, which we call *elementwise condensation,* yields a system of the form

$$\begin{bmatrix} \mathbf{A}_0^{n+1} & \mathbf{0} \\ \mathbf{B}^{n+1} & \mathbf{A}_R^{n+1} \end{bmatrix} \begin{bmatrix} \mathbf{u}_0^{n+1} \\ \mathbf{u}_r^{n+1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}}_0^{n+1} \\ \tilde{\mathbf{r}}_r^{n+1} \end{bmatrix}, \tag{2}$$

where $\mathbf{u}_0^{n+1} \in \mathbb{R}^{2N}$ denotes the vector of coarse-grid unknowns; $\mathbf{u}_r^{n+1} \in \mathbb{R}^{2Z^{n+1}}$ denotes the vector of refinement unknowns; $\mathbf{A}_R^{n+1} \in \mathbb{R}^{2Z^{n+1} \times 2Z^{n+1}}$ is an upper bidiagonal matrix multiplying the refinement unknowns, and $\mathbf{B}^{n+1} \in \mathbb{R}^{2Z^{n+1} \times 2N}$ is the matrix coupling new unknowns to the coarse-grid values. In practice, $\mathbf{A}_0^{n+1}$ has the same size and zero structure as the matrix associated with collocation on $\Delta^0$, and $\mathbf{B}^{n+1}$ is sparse, having one $2\nu^{n+1}(i) \times 4$ nonzero block for every refined element $\overline{\Omega}_i$. Figure 2 shows the block structure of Eq. (2) in more detail.

Given this structure, we can solve for the vector $\mathbf{u}_0^{n+1}$ of coarse-grid variables using our efficient coarse-grid solver, then solve for the refinement unknowns essentially using back substitution via the coupling block $\mathbf{B}^{n+1}$. The time-stepping procedure, starting with $\hat{u}^n$ known, is as follows:

1. Compute $\nu^{n+1}(i)$, $i = 1, \cdots, N$, using an adaptive refinement strategy.
2. Form the projection $\pi^{n+1}\hat{u}^n$.

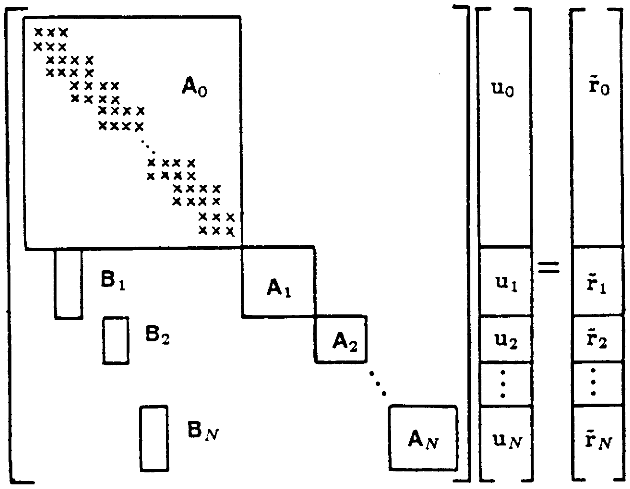

FIG. 2. Block structure of the matrix equation for the locally refined system after elementwise condensation.

3. Compute the matrix entries associated with the refined problem.
4. Use elementwise condensation to construct the system (2).
5. Solve $\mathbf{A}_0^{n+1}\mathbf{u}_0^{n+1} = \bar{\mathbf{r}}_0^{n+1}$ for coarse-grid values.
6. Solve $\mathbf{B}\mathbf{u}_0^{n+1} + \mathbf{A}_R^{n+1}\mathbf{u}_r^{n+1} = \bar{\mathbf{r}}_0^{n+1}$ for variables introduced by the refinement.

Step 6 actually reduces to a set of decoupled problems, each of which has the form

$$\mathbf{B}_i^{n+1}\begin{bmatrix} u_{i-1} \\ u'_{i-1} \\ u_i \\ u'_i \end{bmatrix}^{n+1} + \mathbf{A}_i^{n+1}\mathbf{u}_i^{n+1} = \bar{\mathbf{r}}_i^{n+1}, \qquad (3)$$

for a particular refined coarse-grid element $\overline{\Omega}_i$. Here, $\mathbf{B}_i^{n+1} \in \mathbb{R}^{2\nu^{n+1}(i)\times 4}$ multiplies the coarse-grid unknowns in $\overline{\Omega}_i$, and the upper bidiagonal matrix $\mathbf{A}_i^{n+1} \in \mathbb{R}^{2\nu^{n+1}(i)\times 2\nu^{n+1}(i)}$ multiplies the refinement unknowns in $\overline{\Omega}_i$. Observe that the back substitutions (3) associated with different refined elements $\overline{\Omega}_i$ are independent and therefore are amenable to concurrent processing. Similarly, the elementwise tasks called for in steps 1, 2, 3, and 4 are also parallelizable. We explore this aspect of the method in Section V.

A sample computation demonstrates the effectiveness of this procedure in yielding accurate simulations. Consider the problem (1) on $\Omega = (0, 1)$ with square-wave data,

$$u_I(x) = 0, \qquad u_0 = 1, \qquad u'_N = 0,$$

when $\nu = 0.369$ and $D = 0.001$. If we use a coarse gird $\Delta^0$ having $h = k = 0.05$ and employ upstream weighting with $\zeta = 0.2$, then the numerical solution will exhibit significant smearing, as shown in Figure 3 for $t = 1$. We can virtually eliminate this smearing by forcing $h < 1/P$ globally, but as Figure 3 also shows, we can achieve comparable results by enforcing the same criterion only in zones where $\sup_{x\in\overline{\Omega}_i}|d\hat{u}^n/dx| > (5h)^{-1}$, that is, where the solution is steep. The latter strategy involves solving for at most 180 unknowns per time step, while global refinement requires solving for about 400.

## IV. BURGERS' EQUATION

For nonlinear problems the time-stepping procedure is somewhat more complicated. Here, the use of an implicit scheme for stability forces one to iterate between time steps. Since frontal velocities may be functions of the unknown solution, it is possible that zones needing refinement at a particular time level will be identifiable only in the last few iterations of the time step, when the iterative scheme has nearly converged. We use this reasoning in developing a grid-refinement algorithm for Burgers' equation,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \frac{\mu}{2}\frac{\partial^2 u}{\partial x^2} = 0, \qquad (x,t) \in \Omega \times (0,\infty),$$

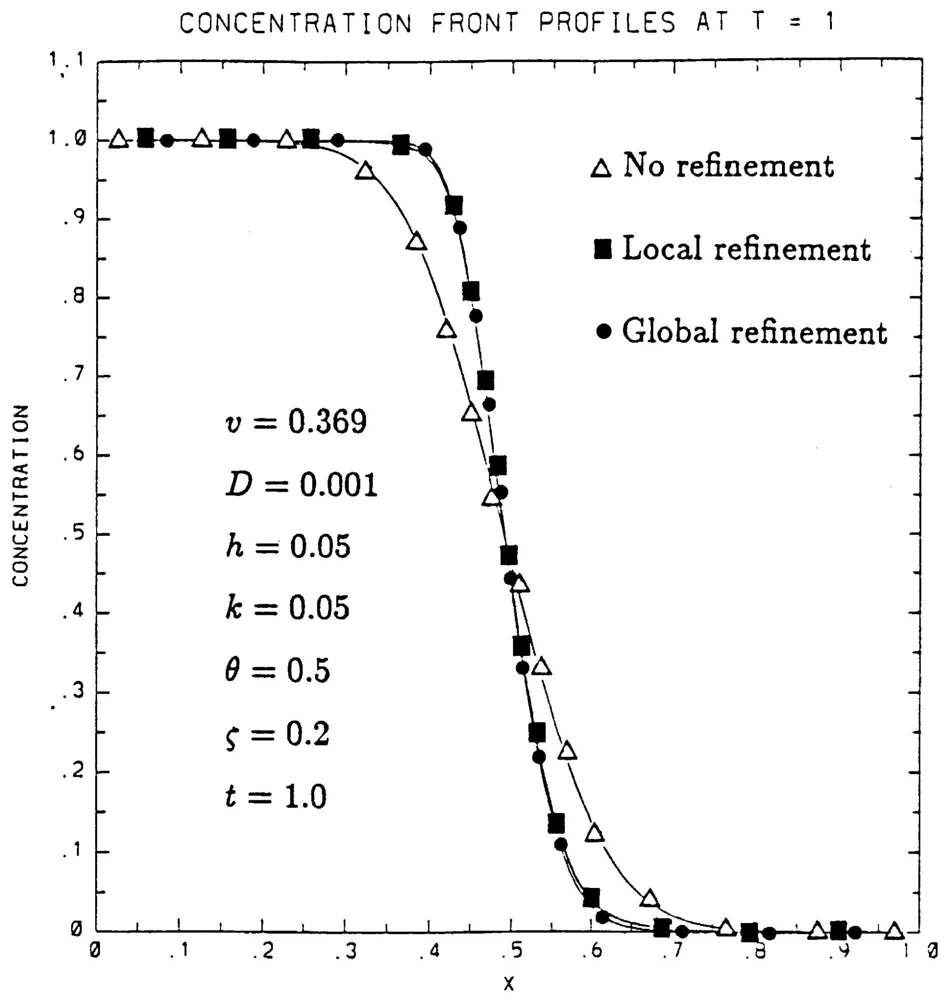

FIG. 3. Upstream-weighted collocation solutions to the advection-diffusion equation using a coarse grid, a globally refined grid, and a locally refined grid.

assuming initial and boundary data having the form

$$u(x,0) = u_I(x), \qquad x \in \Omega = (0,1),$$
$$u(0,t) = u_0, \qquad u(L,t) = u_N.$$

In this equation, $u$ stands for fluid velocity, while $\mu$ represents a fluid viscosity. When $\mu \ll 1$, the equation models nearly inviscid, self-advected flows and has shock-like solutions needing local fine-scale spatial resolution.

In the refined problem on $\Delta^0$, we compute a sequence $\{\hat{u}^n\}_{n=0}^{\infty}$ in $\mathcal{M}_1^3(\Delta^0)$, satisfying the initial and boundary conditions, such that the residual vanishes at each collocation point $\bar{x}_k \in \Omega$. In this case, the residual for the semidiscrete scheme is

$$R^{n+1} = \hat{u}^{n+1} - \hat{u}^n + k\left(\hat{u}^{n+1}\frac{d\hat{u}^{n+1}}{dx} - \frac{\mu}{2}\frac{d^2\hat{u}^{n+1}}{dx^2}\right),$$

which is a nonlinear function of the unknown Hermite coordinates

$$\{(u_0')^{n+1}, (u_1)^{n+1}, (u_1')^{n+1}, \cdots, (u_N')^{n+1}\}$$

for each fixed value of $\bar{x}_k$. To solve this nonlinear problem for $\hat{u}^{n+1}$ in terms of $\hat{u}^n$, we linearize it using Newton's method. Thus we make an initial guess $\hat{u}^{n+1,0} = \hat{u}^n$ and, at each iterative level $m > 0$, solve for a new iterate

$$\hat{u}^{n+1,m+1} = \sum_{i=0}^{N} \{[(u_i)^{n+1,m} + \delta_i]H_{0,i} + [(u_i')^{n+1,m} + \delta_i']H_{1,i}\}.$$

Clearly $\delta_0 = \delta_N = 0$; the boundary values of $\hat{u}^{n+1}$ are known. To compute the vector $\boldsymbol{\delta}$ of remaining increments, we solve the linear system

$$\mathbf{J}^{n+1,m}\boldsymbol{\delta} = -\mathbf{r}^{n+1,m},$$

where the $k$th entry of $\mathbf{r}^{n+1,m}$ is $R^{n+1,m}(\bar{x}_k)$, and $\mathbf{J}^{n+1,m}$ is the Jacobian matrix of $\mathbf{r}^{n+1,m}$ with respect to the unknown Hermite coordinates. Given a tolerance $\tau > 0$, we iterate until $\|\mathbf{r}^{n+1,m+1}\|_\infty < \tau$, then set $\mathbf{u}^{n+1,m+1}$.

In practice this scheme has several nice attributes. First, it is stable for very large time steps, including "Courant" numbers $\|\hat{u}^{n+1}\|_\infty k/h > 100$ that far exceed those required to keep the temporal truncation error reasonably small. Second, it converges rapidly. Using $N = 100$, the scheme reaches $\|\mathbf{r}^{n+1,m+1}\|_\infty < 10^{-7}$ in three or four interations, almost independent of the time step $k$.

To implement adaptive local grid refinement, we adopt a simple "predictor-corrector" strategy in this Newton scheme. This strategy determines the refined grid $\Delta^{n+1}$ only after performing a few Newton iterations on the coarse grid $\Delta^0$. The algorithm runs as follows: for the "predictor" stage, we iterate on $\Delta^0$ to reach a tolerance $\tau_0 > 0$:

1. $\hat{u}^{n+1,0} \leftarrow \tau^0\hat{u}^n$.
2. Solve $\mathbf{J}^{n+1,m}\boldsymbol{\delta} = -\mathbf{r}^{n+1,m}$ on $\Delta^0$ to get iterates $\mathbf{u}^{n+1,m+1} \in \mathcal{M}_1^3(\Delta^0)$. Stop when $\|\mathbf{r}^{n+1,M}\|_\infty < \tau_0$.

At this point we have a crude approximation to the new solution $\hat{u}^{n+1}$, which we use to determine the refined grid:

3. Construct $\Delta^{n+1}$ according to some refinement strategy.

Finally, we perform the "corrector" stage, iterating on $\Delta^{n+1}$ to reach a tolerance $\tau_1 > 0$:

4. $u^{n+1,M+0} \leftarrow \pi^{n+1}u^{n+1,M}$.
5. Solve $\mathbf{J}^{n+1,M+m}\boldsymbol{\delta} = -\mathbf{r}^{n+1,M+m}$ on $\Delta^{n+1}$ to get iterates $\mathbf{u}^{n+1,M+m+1} \in \mathcal{M}_1^3(\Delta^{n+1})$.
6. $\hat{u}^{n+1} \leftarrow \hat{u}^{n+1,M+m+1}$; $n \leftarrow n+1$.

In step 5 we use the elementwise condensation algorithm outlined in the previous section to solve the linear system involving $\mathbf{J}^{n+1,M+m}$.

A sample calculation paralleling that described in Chong [10] illustrates this procedure. Consider problem (3) with $N$-wave data on $\Omega = (0,1)$:

$$u_I(x) = \frac{x}{1 + \exp\left(\dfrac{4x^2-1}{8\mu}\right)}, \qquad u_0 = 0, \qquad u_N = u_I(1),$$

and let $\mu = 10^{-3}$. For the true solution, $|\partial\hat{u}/\partial x| = \mathcal{O}(1)$ except in an interior layer of thickness $\mathcal{O}(\mu)$, in which $|\partial\hat{u}/\partial x| = \mathcal{O}(\mu^{-1})$. If $h$ denotes the coarse-grid mesh, then we insert $\mathcal{O}(\mu^{-1})$ refinement nodes in each coarse-grid element

$\Omega_i$ where $(u_i^n - u_{i-1}^n)/h > 2$. Figure 4 shows the resulting numerical solutions at different time levels, using $h = k = 0.05$, together with a plot of the exact solution for comparison.

## V. IMPLEMENTATION ON A PARALLEL COMPUTER

We have implemented this refinement strategy on an Alliant FX/8 parallel processing computer. The Alliant is a shared-memory machine with optimization capability for both concurrent and vector programming. Five computational elements or processors are available on our machine as currently configured.

The computations associated with each refined coarse-grid element $\overline{\Omega}_i$ are contained in three subroutines. The first routine, called REGRID, constructs the nonsquare system of equations involving variables associated with $\overline{\Omega}_i$. The second, called CNDNS, performs the elementwise condensation and decomposition. The third, BAKSUB, solves for the refinement variables after the solution on the coarse grid is known. These routines are implemented for each refined coarse-grid element $\overline{\Omega}_i$. In each routine, calculations for separate coarse-grid elements are performed concurrently. All computation *inside* each routine must be done sequentially since the processors are in use at this time. However, the sequential calculations in each routine are optimized for vectorization.

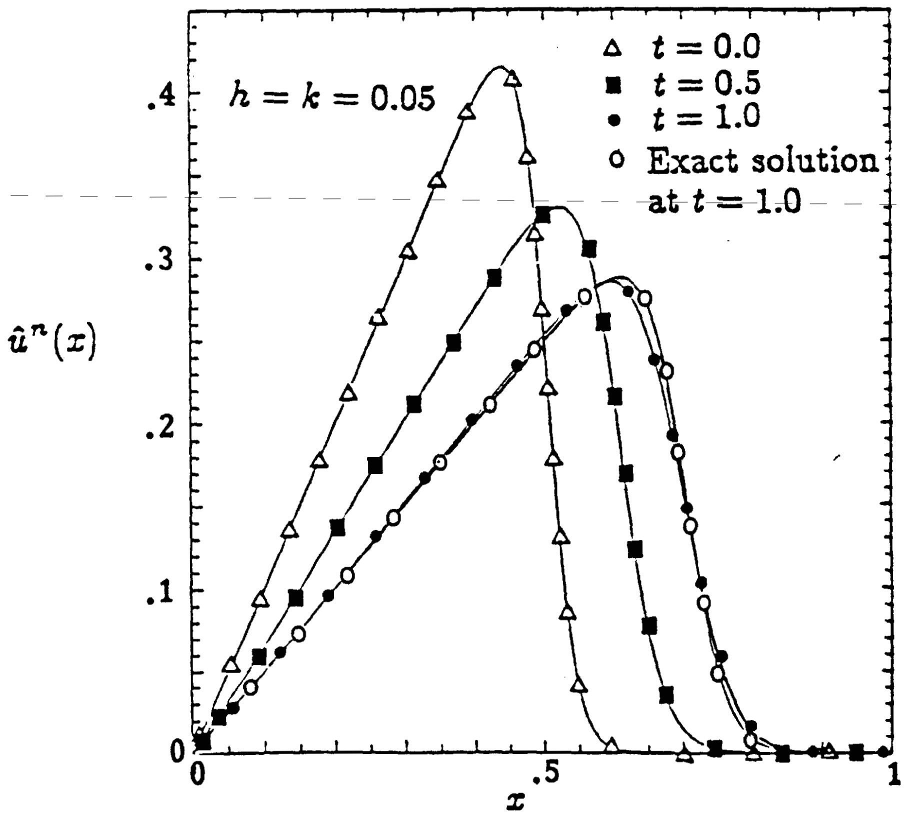

FIG. 4. Solution profiles for Burgers' equation with N-wave data, showing the exact solution for comparison in the last time step.

The machine allows users to control concurrency within a Fortran code through the use of compiler directives. The following is a description of the "corrector" stage of the nonlinear algorithm described in Section IV. The compiler directives themselves begin with the flag CVD$ starting in the first column of code.

```
        Construct the refined grid $\Delta^{n+1}$
        Begin iteration on refined grid until $\|r^{n+1,M+m+1}\| < \tau_1$
          Determine right hand side vector for coarse-grid equations
CVD$L     CNCALL (Compiler directive to permit the concurrent execution of the
          following loop containing a reference to an external procedure.)
          DO for each refined $\overline{\Omega}_i$
            CALL REGRID (Constructs nonsquare systems.)
          END DO
          Check for convergence
          Determine matrix multiplying coarse-grid variables
CVD$L     CNCALL
          DO for each refined $\overline{\Omega}_i$
            CALL CNDNS (Performs condensation and decomposition.)
          END DO
          Solve for coarse-grid variables
CVD$L     CNCALL
          DO for each refined $\overline{\Omega}_i$
            CALL BAKSUB (Solves for refinement variables.)
          END DO
        End Iteration
          .
          .
          .
CVD$R   NOCONCUR (Directive to supress concurrency until the end of the routine.)
        SUBROUTINE REGRID
CVD$R   NOCONCUR
        SUBROUTINE CNDNS
CVD$R   NOCONCUR
        SUBROUTINE BAKSUB
```

One measure of how well the algorithm makes use of the machine's parallel capabilities is the *speedup*. Speedup for $n$ processors is the ratio of the time needed by one processor to the time used by $n$ processors to perform the computation associated with grid refinement. If there were no overhead required to monitor and schedule the various processors, the speedup for $n$ processors would be $n$. Figure 5 shows four speedup curves. These plots represent the speedups achieved by our algorithm for an average of two, four, six, and eight elements refined per time step in the Burgers' equation solver. As expected, for an average of two elements refined per time step, the speedup does not improve for more than two processors and even decreases slightly due to the increased overhead. Similarly, for an average of four elements refined in each time step, speedup does not improve when a fifth processor is used. Figure 6 shows the speedup curve when five elements are refined per time step. Clearly, this amounts to a special case for our machine configuration. The speedup for five processors is 3.51. This result compares with a machine peak of 4.5, observed by Puckett and Schmidt [11] while using a purely parallel algorithm with no

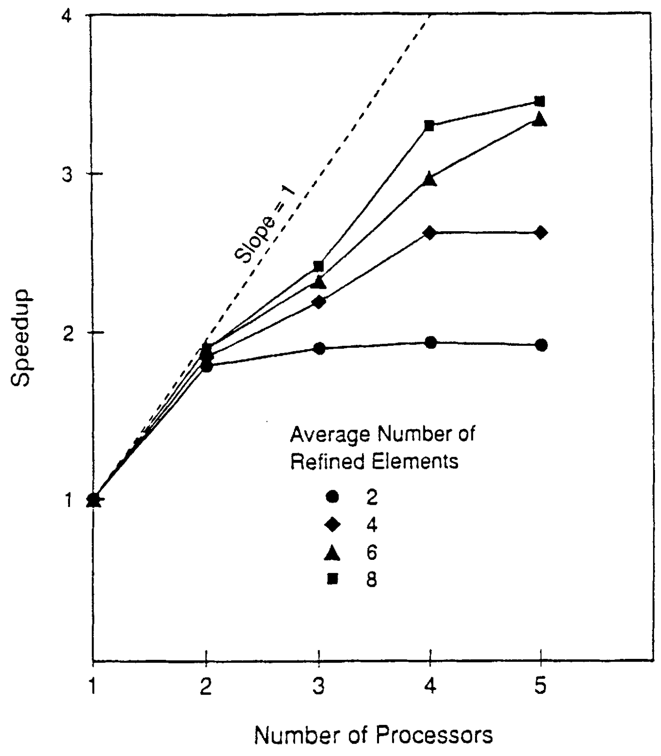

FIG. 5. Speedup plots for the parallel computations in the local gridding algorithm implemented on a five-processor machine with shared memory. Different curves represent different average numbers of coarse-grid elements refined per time step.

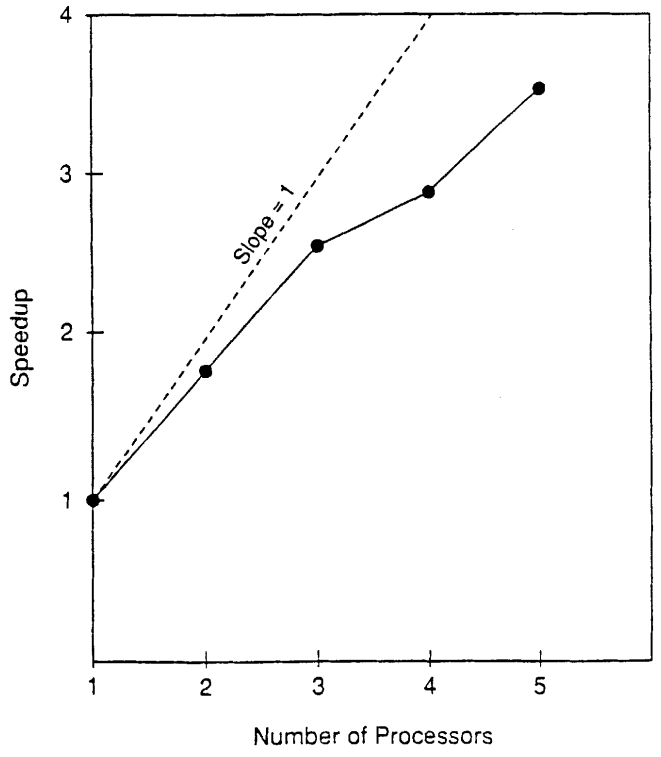

FIG. 6. Speedup plot for the parallel computations in the local gridding algorithm implemented on a five-processor machine with an average of five coarse-grid elements refined per time step.

data sharing among processors and no accumulation of results. Those authors found that peak performance occurs only when the number of iterations in a concurrent loop is quite large: They achieved the speedup of 4.5 in a loop having 3600 iterations.

There are several factors that prevent optimal speedup in our algorithm for grid refinement. First, not every processor has the same computational burden, since the amount of refinement in the coarse-grid elements can vary spatially. Second, the number of iterations performed in each loop is typically small, owing to the local nature of the refinement. A third barrier to the attainment of peak performance is the necessity to accumulate the results of the parallel computations in memory for use in subsequent calculations. These limitations seem inherent in any adaptive gridding procedure for nonlinear, transient flows. With this proviso, our algorithm appears to make good use of the shared-memory parallel architecture.

The National Science Foundation supported this work through grants DMS-8504360 and RII-8610680. The Wyoming Water Research Center also provided support through a grant-in-aid to the authors.

## References

[1] B. A. Finlayson, *The Method of Weighted Residuals and Variational Principles*, Academic Press, New York, 1972.

[2] C. de Boor and B. Swartz, "Collocation at Gaussian points," *SIAM J. Numer. Anal.*, **10**, 582–606 (1973).

[3] J. Douglas and T. Dupont, "A finite element collocation method for quasi-linear parabolic equations," *Math. Comp.*, **27**, 17–28 (1973).

[4] G. Birkhoff and R. E. Lynch, *Numerical Solution of Elliptic Problems*, SIAM, Philadelphia, 1984.

[5] P. M. Prenter, *Splines and Variational Methods*, Wiley, New York, 1975.

[6] M. B. Allen and G. F. Pinder, "Collocation simulation of multiphase porous-medium flow," *Soc. Pet. Eng. J.* 135–142 (1983).

[7] O. K. Jensen and B. A. Finlayson, "Oscillation Limits for Weighted Residual Methods Applied to Convective Diffusion Equations," *Int. J. Numer. Meth. Eng.*, **15**, 1681–1689 (1980).

[8] M. B. Allen, "How upstream collocation works," *Int. J. Numer. Meth. Eng.*, **19**, 1753–1763 (1983).

[9] M. F. N. Mohsen and G. F. Pinder, "Collocation with 'adaptive' finite elements," *Int. J. Numer. Meth. Eng.*, **20**, 1901–1910 (1984).

[10] T. H. Chong, "A variable mesh finite difference method for solving a class of parabolic differential equations in one space variable," *SIAM J. Numer. Anal.* **15**, 835–857 (1978).

[11] J. A. Puckett and R. J. Schmidt, "Finite strip method in a parallel computer environment," preprint, Department of Civil Engineering, University of Wyoming, Laramie, Wyoming, 1988.