Efficient Parallel Communication Schemes for Computational Fluid Dynamics Codes

N. Gopalaswamy,* A. Ecer,† H. U. Akay,† and Y. P. Chien‡
Indiana University—Purdue University at Indianapolis, Indianapolis, Indiana 46202

A scheme for improving the efficiency of communications for the parallel computation of Euler equations is presented. The NPARC (National PARC) code is employed as an example for analyzing the flow through a supersonic inlet. The flowfield is divided into subregions called blocks. The parallel computation of the problem normally requires communication between each block after each time step of an explicit Runge–Kutta integration scheme. In the developed procedure, the boundary conditions are frozen for $k = 10$–20 time steps, and blocks are integrated in time without communication with each other during this period. When the boundary conditions are updated, an oscillatory error is introduced into the solution with a fundamental period of $4k$ time steps, which is then filtered in time. As a result, the communication cost of parallel computing is significantly reduced. Examples for steady and unsteady flows are presented to demonstrate the applicability of the developed procedure.

Introduction

DURING the parallelization of explicit schemes, the efficiency of the communication plays a critical role. Especially for a structured grid, one can develop explicit schemes where computational cost is small in comparison with the communication cost. In the present paper the NPARC (National PARC) code with an explicit Runge–Kutta scheme is chosen as the parallel numerical algorithm to be studied.[1] This code was obtained from NASA Lewis Research Center. Parallelization of this code has already been discussed in a previous paper.[2] It is based on a block-based structure of the data where the solution domain is divided into many subdomains or blocks. The global solution is obtained by integrating the equations for each block separately. The blocks are interconnected through an overlapping region or interface by one grid point. The solution scheme marches in time while exchanging boundary values of each block at each time step. Figure 1 summarizes the arrangement for the case of two neighboring blocks.[2] The numerical integration of the grid points are conducted inside a block solver. At intervals, the block solver updates an interface solver, which then communicates with the interface solver of the neighboring block. Each interface solver also updates its block after receiving information from its neighbor. As can be seen from Fig. 1, each block and its interface solvers are on the same processor. The time intervals for sending and receiving information between the blocks and interfaces can be different and can be chosen based on the local conditions.[2] The distribution of the blocks among a given number of processors can be optimized by distributing the blocks according to their computation and communication requirements.[2,3]

In Ref. 2, based on the local stability conditions, the time intervals for communicating between the blocks and interfaces, as defined in Fig. 1, were selected. The resulting system was then load balanced, and considerable efficiency improvements were obtained, specifically by reducing the communication cost. In the present paper, a brief summary of this procedure and the parallelization tools is provided. A further attempt to reduce the communication cost is presented here because the stability requirements for explicit schemes can be quite stringent. Specifically, the communication time interval is further increased, exceeding the limit suggested by the local stability conditions at the interface. By not updating the boundary of a block at required time intervals, the solution becomes discontinuous between the blocks. An error introduced at each boundary produces high-frequency spatial oscillations inside each block. Based on the study of this error, a filtering scheme, which corrects the boundary conditions and eliminates the high-frequency noise, is developed. By employing this scheme, one can reduce the communication cost by 90% yet maintain the same accuracy. The numerical results presented in this paper demonstrate applications for both steady and unsteady flows.

Explicit Runge–Kutta Algorithm and Stability

The governing Euler equations for inviscid flow are cast in the following conservation form:

$$\frac{\partial Q}{\partial t} + \frac{\partial F_j}{\partial X_j} = 0 \tag{1}$$

where $Q = (\rho, \rho u, \rho v, \rho w, \rho E)^T$ and F_j are the inviscid flux vectors. These equations are transformed into computational coordinates and are solved in strong conservation form by the NPARC code. Additional source terms appear on the right-hand side (RHS) of Eq. (1) for axisymmetric flows. The NPARC code can solve the Euler equations with either an implicit Beam–Warming algorithm or an explicit, multistage Runge–Kutta scheme. In the present paper, a three-stage variant of the Runge–Kutta scheme is considered. The Euler equations are cast in semidiscretized form as follows:

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = A_j \cdot F_j = \mathrm{RHS} \tag{2}$$

where A is the space discretization operator operating on the vector of conservation variables Q. Central differencing is used for the discretization of the spatial domain. The three-stage Runge–Kutta scheme used can be written as follows:

$$Q(1) = Q^n + 0.6\Delta t\,\mathrm{RHS}(0), \qquad Q(2) = Q^n + 0.6\Delta t\,\mathrm{RHS}(1)$$
$$Q^{n+1} = Q^n + \Delta t\,\mathrm{RHS}(2) \tag{3}$$

where Δt is the time step used for the temporal integration. A linearized stability analysis for the one-dimensional Euler equations

Presented as Paper 97-0027 at the AIAA 35th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1997; received March 8, 1997; revision received Jan. 27, 1998; accepted for publication Feb. 5, 1998. Copyright © 1998 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Graduate Research Assistant, Department of Mechanical Engineering, Computational Fluid Dynamics Laboratory, Purdue School of Engineering and Technology, 723 West Michigan Street.

†Professor, Department of Mechanical Engineering, Computational Fluid Dynamics Laboratory, Purdue School of Engineering and Technology, 723 West Michigan Street.

‡Associate Professor, Department of Electrical Engineering, Computational Fluid Dynamics Laboratory, Purdue School of Engineering and Technology, 723 West Michigan Street.

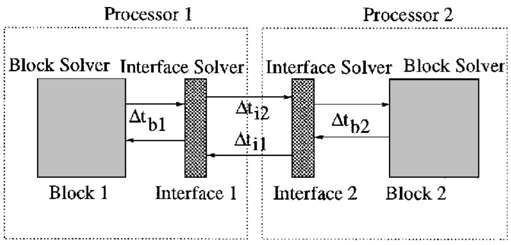

Fig. 1 Blocks and interfaces.

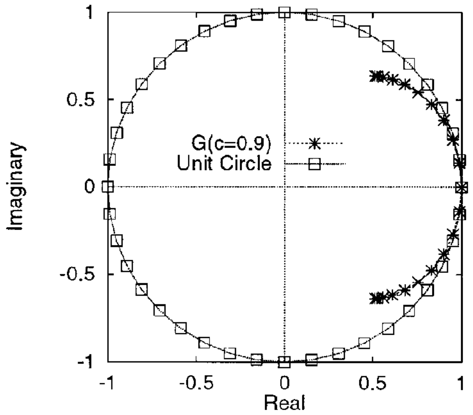

Fig. 2 G for $0 < \theta < 2\pi$ and $c = 0.9$.

in conservation form discretized as defined in Eq. (3) yields the following Courant–Friedrichs–Lewy stability criterion:

$$c = \frac{(u + a)\Delta t}{\Delta x} \leq 1.8 \tag{4}$$

where c is the Courant number in Eq. (4). The amplification factor $G(z)$ can be defined in terms of the characteristic polynomial obtained from the linearized stability analysis[4]:

$$|G| = |Q^{n+1}/Q^n|, \qquad G = 1 - z + 0.6z^2 - 0.36z^3$$
$$z = Ic \sin\theta \tag{5}$$

where θ is the phase angle obtained from a Fourier decomposition in the frequency domain. The region near $\theta = 0$ corresponds to the low-frequency region, whereas the region near $\theta = \pi$ corresponds to the high frequencies. The highest frequency resolvable by the mesh corresponds to a wavelength of $2\Delta x$. Figure 2 contains a plot of the amplification factor G. It can be seen that the amplification factor G is approximately equal to unity for both high- and low-frequency ranges. This implies that the low- and high-frequency spatial waves are not damped by the three-stage Runge–Kutta scheme. Artificial viscosity is normally introduced to damp the high-frequency oscillations. Another thing to be noticed from Fig. 2 is that for high frequencies there is a phase error of π. This implies that high-frequency components of a solution will move in a direction opposite to that of the low-frequency components.

Test Cases

Two test cases are chosen to investigate the effect of the reduced communications. These cases were also employed in the previous study of the NPARC code.[2]

1) Steady flow: An axisymmetric, mixed-compression, variable-diameter centerbody inlet is considered under a supersonic inflow of $M = 2.5$ and a subsonic compressor face outflow Mach number $M = 0.3$ (Ref. 5). The two-dimensional version of the NPARC code has an option to handle axisymmetric flow also. The reference inlet pressure is 117.8 lb/ft^2, and the reference inlet temperature is 395°R. The cowl-tip radius of the inlet, $Rc = 18.61$ in., is used to nondimensionalize the lengths. The grid for this inlet consists of approximately 4500 nodes and is divided into 15 blocks, all of approximately equal size, as shown in Fig. 3. A steady-state solution is sought using local time stepping for all nodes in each block with a uniform Courant number of 0.9 for all nodes. The solution is plotted in the form of density at the midpoint of each interface for all blocks at every iteration. This test case is chosen as an example of a small problem where communication cost is large in comparison with the computation cost.

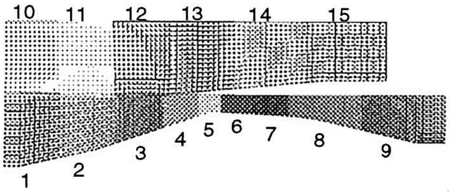

Fig. 3 Axisymmetric engine inlet grid with 15 blocks.

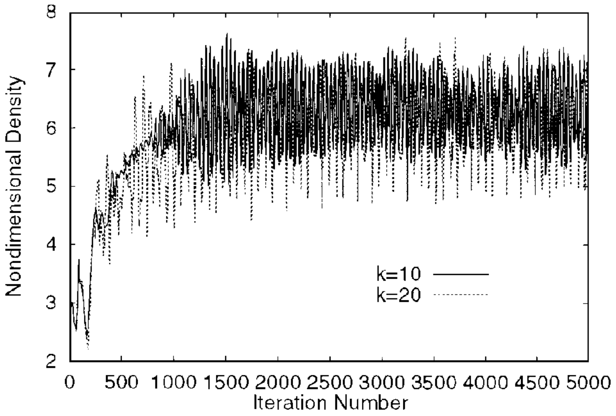

Fig. 4 Density variation at interface for $k = 10$ and 20.

2) Unsteady flow: The same grid illustrated in Fig. 3 is used to study the response to a sinusoidal temperature perturbation applied at the inlet section. The amplitude of the perturbation is 5%, and the frequency is 100 Hz. The density variation is plotted at one of the subsonic interfaces downstream of the normal shock. The steady-state solution is obtained first, and then the temperature perturbation is applied. The reference pressure and temperature are 117.8 lb/ft^2 and 395°R, respectively. Variable time stepping is used inside each block.[2]

Reduced Communications for Explicit Schemes

By using variable time stepping, considerable improvements in efficiency were obtained.[2] To further reduce the communication cost dictated by the stability conditions, one can further increase the interval for updating the interfaces. An experiment was conducted as follows. After grid points on each block and interface have chosen their own time step, based on local stability conditions, the boundary conditions were frozen for 10 time steps. This led to both spatial and temporal oscillations. The magnitude of these oscillations was negligible for supersonic interfaces but significant for subsonic interfaces. In Fig. 4, the variation of density with respect to time is plotted at the subsonic interface between blocks 8 and 9 in Fig. 3 for the steady-flow test case. The solution is stable inside each block because the time step chosen for integration satisfies the local stability condition for the grid points inside the block. However, the solution is polluted by a high-frequency noise emanating from the discontinuity introduced on the boundary. A frequency decomposition of the signal in Fig. 5 shows that the high-frequency oscillations are associated with distinct frequencies. They corresponded to a time period of

$$T = 4k\Delta t \tag{6}$$

and its multiples, where k is the communication interval. The frequency in Fig. 5 is nondimensionalized as follows:

$$\omega^* = \omega(2k\Delta t/\pi) \tag{7}$$

The same behavior was observed when k was increased to 20, as shown in Figs. 4 and 5, although there are many more peaks observed

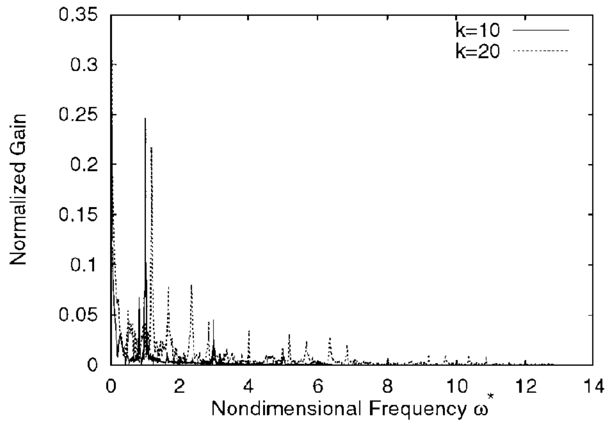

Fig. 5 Frequency response of density variation for $k = 10$ and 20.

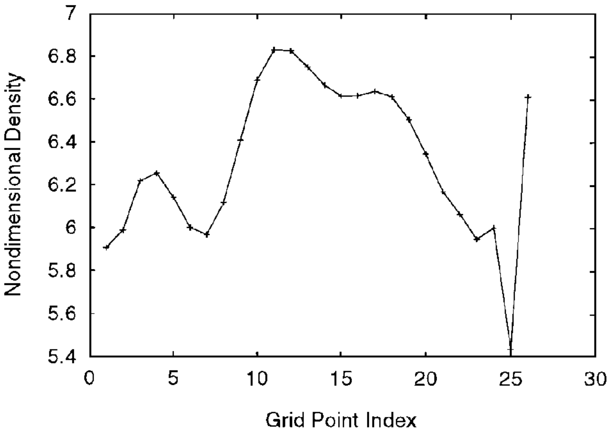

Fig. 6 Instantaneous density variation inside a block for $k = 20$.

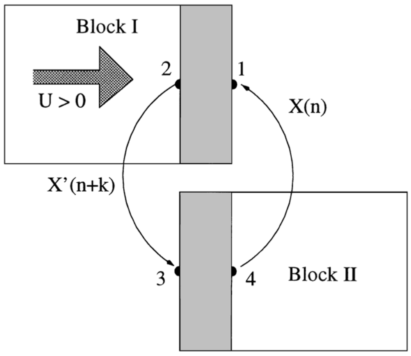

Fig. 7 Model of feedback system.

in the frequency spectrum. This is because the frequencies excited by the communication errors are much lower than in the preceding case and interact with the correct solution. This point will be further discussed later. Figure 6 shows the spatial oscillations developing inside a block due to the error introduced by freezing the boundary conditions for $k = 20$. A frequency decomposition of the signal in Fig. 6 indicates a significant oscillation with a wavelength of $2\Delta x$ near the boundary.

In the following, the source of the preceding errors introduced by reducing the communications is discussed, and a filtering technique is utilized to eliminate the oscillations while maintaining the accurate solution.

Error Analysis

An investigation was carried out to explore the origin of the aforementioned oscillations. The following simple model was defined to study the problem. For the case of two blocks shown in Fig. 7, the flow is assumed to be one dimensional from left to right and subsonic. The interfaces belonging to blocks I and II overlap by only one grid point. Because the flow is subsonic, two waves propagate information downstream with speeds u and $u + a$, whereas one wave propagates information upstream with a speed of $u - a$. The term u is the fluid velocity, and a is the acoustic speed for the fluid.

During the parallel computation, point 1 serves as the downstream boundary condition for block I, and point 3 serves as the upstream boundary condition for block II. Points 2 and 4 are computed as interior points of blocks I and II, respectively. During the time integration, the solution values computed at point 2 overwrite the previous values at point 3 every time communication between the interfaces takes place. Simultaneously, the values computed at point 4 overwrite the previous boundary condition at point 1.

If the communication is halted for a specified interval, then the time integration in blocks I and II proceeds, with the boundary condition remaining frozen at the values received during the past communication step. Hence an error is introduced into the time integration procedure in both blocks. If the semidiscretized Euler equations can be expressed as

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = A \cdot Q \tag{8}$$

for a linearized operator $A\cdot$, the error obeys the same difference equation as the solution. Hence, if we call the error X, the following relation is valid:

$$\frac{\mathrm{d}X}{\mathrm{d}t} = A \cdot X \tag{9}$$

One can trace the propagation of the error through this model. Assume that $X_1(n)$ is an error in the boundary condition, first introduced at a time step n, at point 1. If the boundary conditions are held fixed for k time steps, this error will propagate upstream in block I to point 2. Because the error also obeys the same discretized equation as the solution for a linear operator, the error will be modified by the time it propagates to point 2 to become $X_2(n + k)$ after k time steps. When communication occurs at this instant, $X_3(n + k)$ is replaced by $X_2(n + k)$. Over the next k steps, the error at point 3 propagates to point 4 and also gets modified by the integration process to become $X_4(n + 2k)$. Thus, when communication now occurs at $n + 2k$, $X_1(n + 2k)$ becomes equal to $X_4(n + 2k)$, and this process repeats itself. This can be summarized with the following set of expressions:

$$X_2(n+k) = f_1 \cdot X_1(n), \qquad X_3(n+k) = X_2(n+k)$$
$$X_4(n+2k) = f_2 \cdot X_3(n+k) \tag{10}$$
$$X_1(n+2k) = X_4(n+2k), \qquad X_1(n+2k) = f_1 \cdot f_2 \cdot X_1(n)$$

where $f_1\cdot$ and $f_2\cdot$ are operators representing the integration process inside each block. The last expression in Eq. (10) provides a relationship between the error introduced at time step n and $n + 2k$. It will be shown in the following section that spatial oscillations produce a negative feedback that can be approximated with the following relationship:

$$f_1 \cdot f_2 \cdot \approx -1 \tag{11}$$

Based on this approximation, one can describe the oscillations in time at a boundary point by the following relationship:

$$X_1(n+2k) = -X_1(n) \tag{12}$$

Taking a Z transform of the preceding relation leads to

$$z^{2k}X(z) = -X(z), \qquad (1 + z^{2k})X(z) = 0, \qquad z^{2k} = -1 \tag{13}$$

The solution of the preceding equation provides $2k\theta = 2m\pi + \pi$, $m = 0, 1, 2, 3, \ldots$, where $z = re^{I\theta}$. The fundamental solution is $2k\theta = \pi$, corresponding to $m = 0$. Hence the fundamental frequency of oscillations corresponds to a period of $T = 4k\Delta t$.

Origin of the Negative Feedback

As suggested in Eq. (11), the net effect of the two operators $f_1\cdot$ and $f_2\cdot$ leads to a system with negative feedback. To understand this behavior, one has to study the difference representation of the employed three-stage, explicit Runge–Kutta scheme. For a wave traveling downstream with a wave speed of $u + a$, one can write a difference equation as follows:

$$\frac{\mathrm{d}Q}{\mathrm{d}t} = (u + a)\frac{Q_{i-1} - Q_{i+1}}{2\Delta x} \tag{14}$$

where $(u + a)$ is a constant. The explicit Runge–Kutta difference representation yields

$$\begin{aligned} Q_i^{n+1} &= a_5 Q_{i-3}^n - a_2 Q_{i-2}^n - a_4 Q_{i-1}^n \\ &\quad + (1 + 2a_2) Q_i^n + a_4 Q_{i+1}^n + a_2 Q_{i+2}^n + a_5 Q_{i+3}^n \end{aligned} \tag{15}$$

where

$$\begin{aligned} a_2 &= -0.15c^2, & a_4 &= 0.135c^3 - 0.5c \\ a_5 &= 0.045c^3, & c &= \frac{(u + a)\Delta t}{\Delta x} \end{aligned} \tag{16}$$

The difference equation (15) is then modified near the boundaries and cast in matrix form as follows:

$$\begin{aligned} \{\Delta Q\} &= \left\{Q_i^{n+1} - Q_i^n\right\}, \qquad i = 2, 3, \ldots, N - 1 \\ \{\Delta Q\} &= [B]\left\{Q_i^n\right\} + \{B\}' Q_L + \{B\}'' Q_R \end{aligned} \tag{17}$$

where $\{\Delta Q\}$ denotes the vector of unknowns, N is the total number of grid points, and Q_L and Q_R are the left and right boundary conditions, respectively. The vectors $\{B\}'$ and $\{B\}''$ have the following structure:

$$\{B\}' = \begin{Bmatrix} a_1 \\ -a_2 \\ a_5 \\ \vdots \end{Bmatrix}, \qquad \{B\}'' = \begin{Bmatrix} \vdots \\ -a_5 \\ -a_2 \\ -a_1 \end{Bmatrix} \tag{18}$$

In the preceding, $a_1 = 0.5c - 0.045c^3$; furthermore, a new variable is defined as $a_3 = 0.09c^3 - 0.5c$. One can also express the matrix $[B]$ in the following form:

$$[B] = \begin{bmatrix} a_2 & a_3 & -a_2 & -a_5 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ -a_3 & 2a_2 & a_4 & -a_2 & -a_5 & \cdot & \cdot & \cdot & \cdot & \cdot \\ -a_2 & -a_4 & 2a_2 & a_4 & -a_2 & -a_5 & \cdot & \cdot & \cdot & \cdot \\ a_5 & -a_2 & -a_4 & 2a_2 & a_4 & -a_2 & -a_5 & \cdot & \cdot & \cdot \\ \cdot & a_5 & -a_2 & -a_4 & 2a_2 & a_4 & -a_2 & -a_5 & \cdot & \cdot \\ & & & & \vdots & & & & & \\ \cdot & \cdot & \cdot & a_5 & -a_2 & -a_4 & 2a_2 & a_4 & -a_2 & -a_5 \\ \cdot & \cdot & \cdot & \cdot & a_5 & -a_2 & -a_4 & 2a_2 & a_4 & -a_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & a_5 & -a_2 & -a_4 & 2a_2 & a_3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & a_5 & -a_2 & -a_3 & a_2 \end{bmatrix} \tag{19}$$

An eigenvalue–eigenvector decomposition of the matrix $[B]$ shows that the scheme is stable because the real part of all of the eigenvalues is negative except for one that is equal to zero. The zero eigenvalue corresponds to the highest-frequency spatial oscillation with a wavelength of $\lambda = 2\Delta x$, and hence there is no damping for these high-frequency spatial waves. This behavior was also observed for Euler equations from the linearized stability analysis described in Fig. 2. The introduction of an error Q_L on the left boundary excites the low-frequency waves, which are convected with little damping for positive c. Thus, one can state that $f_1\cdot \approx 1$. On the other hand, the introduction of an error Q_R at the right boundary excites the $2\Delta x$ wave, again with no damping. This results in $f_2\cdot \approx -1$.

The behavior described earlier can be illustrated by a simple one-dimensional example, as shown in Fig. 8. An error of unit magnitude was applied at both boundaries of a block, and $c = 0.9$ was chosen to advance for $n = 30$ time steps. It can be seen from Eq. (18) for $c = 0.9$ that all three nonzero entries in $\{B\}'$ are positive, whereas the signs of the three nonzero entries in $\{B\}''$ alternate. Therefore, a disturbance applied on the left boundary gets convected downstream with little damping, as expected. On the other hand, the one applied on the right boundary produces a high-frequency oscillation of wavelength $2\Delta x$, which travels upstream, again without being damped by the difference scheme.

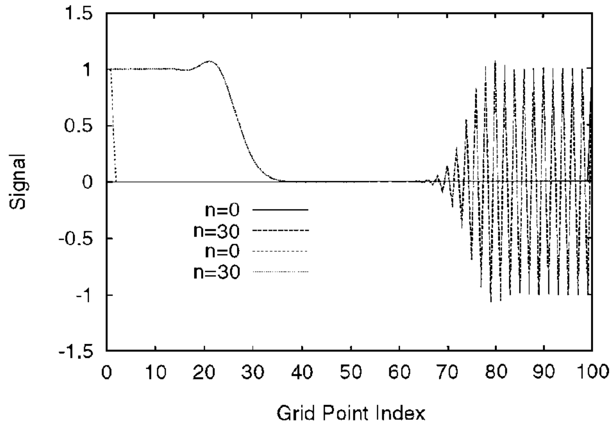

Fig. 8 Convection of a pulse by the Runge–Kutta scheme.

In the preceding model equation, no artificial viscosity was introduced. In the solution of Euler equations, when an error is introduced on the boundary of a subsonic block, after waiting a reasonable number of time steps one can expect that it will appear at one grid point downstream of the boundary with approximately the same magnitude. On the other hand, the same error will appear at one point upstream of the boundary with a negative sign. This behavior is distinctly observed for subsonic flows in the solution of the Euler equations. For supersonic flows, waves traveling upstream are damped by adding numerical viscosity; thus, the feedback and resulting oscillations are negligible.

At this point, one can also comment on the differences observed in the frequency response of the density variation for $k = 10$ and 20 shown Fig. 5. If the communication is delayed too long, there is a coupling between the waves originating from different boundaries as well as waves reflecting back. Thus, for $k = 20$, one observes a more complicated frequency response.

Filtering of the Oscillatory Signals

From the discussion of the preceding sections, it can be deduced that the freezing of the boundary conditions introduces a high-frequency error into the solution with a distinct period of $4k\Delta t$. Because the frequency of the noise is known, one can design a low-pass filter to eliminate the high-frequency noise and to allow the solution to pass through. To design a simple filter, a moving average was employed.[6] As described in Fig. 7, the computed solutions at points 2 and 4 are filtered as follows:

$$\overline{Q_2^n} = \frac{1}{4}\sum_{j=0}^{j=3} Q_2^{n-j\times k}, \qquad \overline{Q_4^n} = \frac{1}{4}\sum_{j=0}^{j=3} Q_4^{n-j\times k} \tag{20}$$

where k is the communication interval. The right-hand side of Eq. (20) involves the raw data calculated at every k time step at points 2 and 4. Intermediate time steps are not utilized in filtering. The left-hand side defines the filtered value of the boundary condition, which is communicated to the neighboring block. This operation corresponds to applying an finite impulse response (FIR) filter, where its Z transform can be expressed as follows:

$$b = \frac{z^{-1} + 2z^{-2} + 3z^{-3} + 4z^{-4} + 3z^{-5} + 2z^{-6} + z^{-7}}{16} \tag{21}$$

which is always stable.

In Fig. 9, the normalized gain of the filter is plotted against the nondimensionalized frequency ω^*. As can be seen, the selected filter

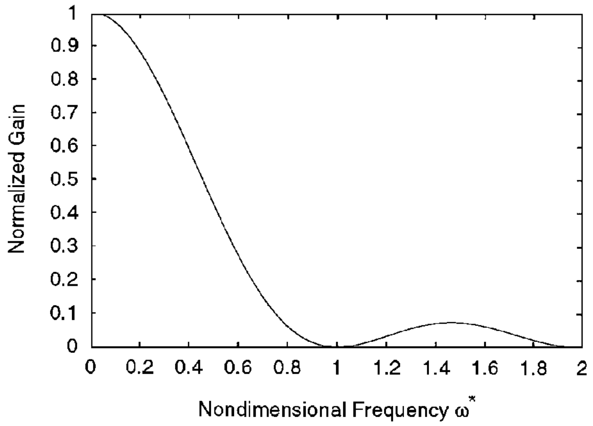

Fig. 9 Frequency response of the FIR filter.

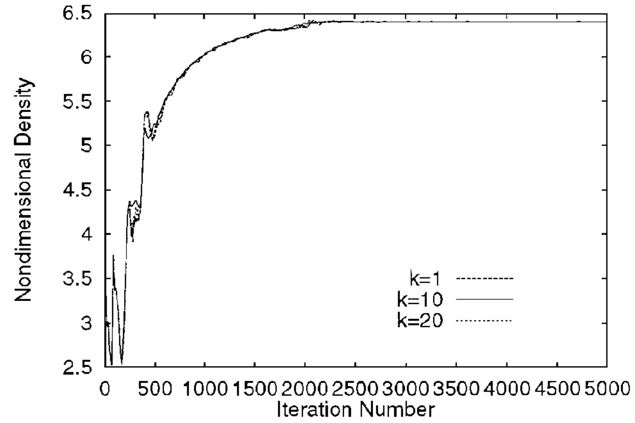

Fig. 10 Solutions with filtering (*k* = 10 and 20) and base case (steady).

provides zero gain for a nondimensionalized frequency of unity or $T = 4k\Delta t$.

Results

Steady Flow

For the steady-flow test case described previously, a base case solution is obtained initially by communicating every time step. Then, first, the communication is frozen for 10 steps, and the resulting solution is filtered at every communication step before being sent to the neighboring interface. Local time stepping is used inside each block for both the base case and the case with the filter. Figure 10 shows the density variation at the midpoint of the subsonic interface in block 9 in Fig. 3. As can be seen, the same steady-state solution is reached after 5000 time steps for both cases. There are some differences in the transient behavior of the solution. Figure 11 shows the frequency spectrum of the same density for both solutions. One can observe that the solutions are accurate within a certain frequency range. Second, communication is frozen for 20 steps, and the solution is again filtered before communication. Figure 10 shows the resulting density variation for the same subsonic interface, and Fig. 12 shows the frequency distribution.

This steady-flow case is solved on a varying number of processors to compare the savings in elapsed time due to the reduced communication. Two systems were used to solve the cases: 1) an IBM SP2 tower with 32 processors using a fast communication network (HPN) located at Poughkeepsie, New York, and 2) an IBM SP1 tower with 16 processors using an Ethernet-based communication network located at NASA Lewis Research Center. The timings obtained are presented in the form of speedup, which is defined as follows:

$$\text{Speedup} = \frac{\text{Elapsed time with 1 processor}(k = 1)}{\text{Elapsed time with } m \text{ processors}} \tag{22}$$

Figure 13 shows the speedup for the steady case for $k = 10$ with filtering and the base case on both types of networks. Speedup is improved considerably because the communication cost is reduced by 90–95%. The speedup improvement is significant, mainly due to the relative importance of the communication cost for the base case. It is also observed that, for a slow network like Ethernet, communication dominates the total elapsed time for the computation of the problem, and hence dramatic improvements are obtained in the speedup when communication is reduced by 90%.

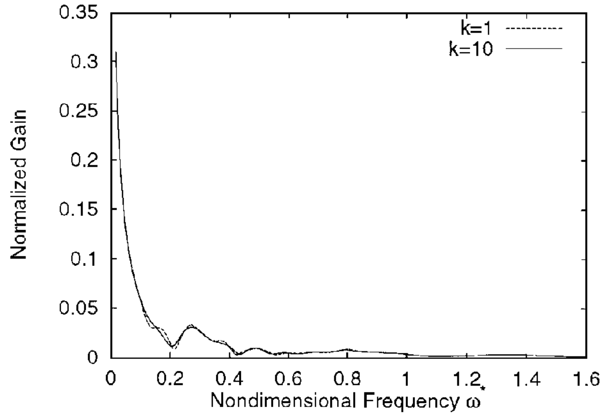

Fig. 11 Frequency response of solutions with filtering (*k* = 10) and base case (steady).

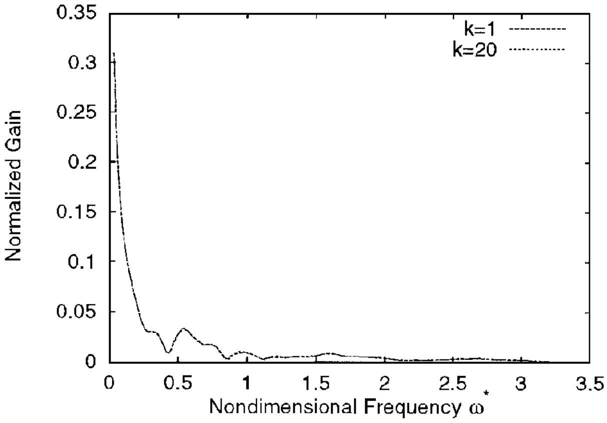

Fig. 12 Frequency response of solutions with filtering (*k* = 20) and base case (steady).

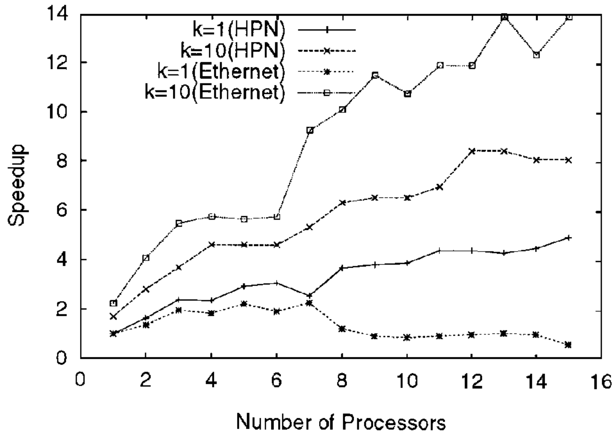

Fig. 13 Speedup for solutions with filtering (*k* = 10) and base case (steady).

Unsteady Flow

An unsteady-flow test case is chosen as described previously. A base case was run by communicating every time step to obtain an accurate solution. Between 400–1800 time steps were employed to integrate over one period of the oscillation for different blocks.

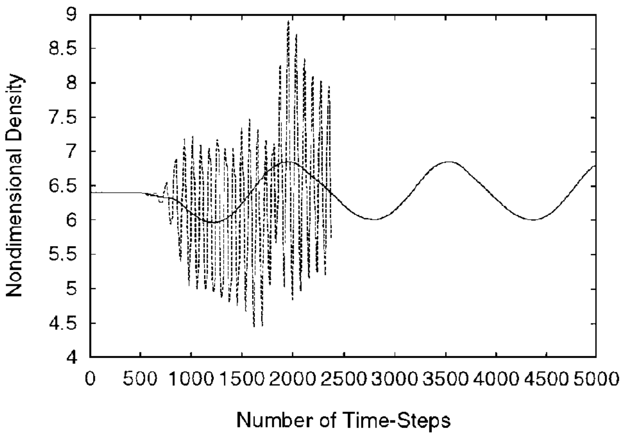

Fig. 14 Solutions for k = 10 and 20 without filtering (unsteady).

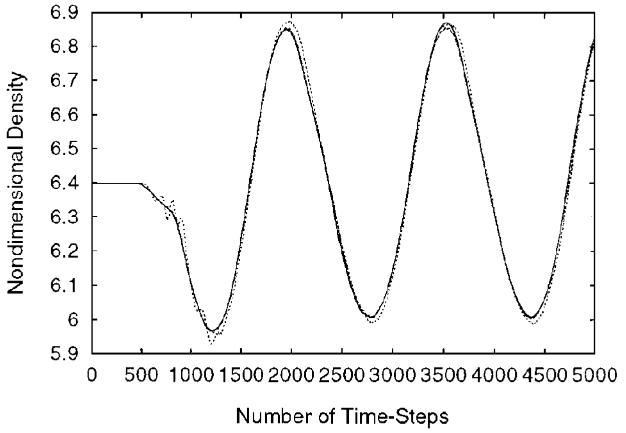

Fig. 15 Solutions for the base case (unsteady) and k = 10 and 20 with filtering.

Figure 14 illustrates the variation of density at the midpoint of the subsonic interface of block 9 in Fig. 3 for $k = 10$ and 20 without filtering. For $k = 10$, the unsteady response is quite accurate. In this case, it was observed that one can freeze the boundaries for $k = 10$ and obtain reasonably accurate solutions even without filtering, as shown in Fig. 14. This may be because the time steps for the unsteady-flow test case are much smaller than those for the steady-flow case. Also in Fig. 14, it can be seen that communicating every 20 time steps introduces an error that eventually causes the solution to diverge. For this case, filtering can be used to eliminate the error and recover the wave of frequency 100 Hz. Figure 15 illustrates the variation of density at the same location in block 9 for two filtered cases with $k = 10$ and 20 in comparison with the base case. As can be seen, the filtering introduces a slight lag for $k = 20$. The design of another filter may eliminate the lag observed in the $k = 20$ case. Also, for the same case, inaccuracies that are associated with the startup transients are observed.

The frequency spectrum of the solution for the base case and for two filtered cases with $k = 10$ and 20 are shown in Figs. 16 and 17. There is very little difference between the frequency contents of these three solutions. Figure 18 shows the speedup for the computation of the unsteady-flow case on a varying number of processors for both types of networks. Again it can be seen that reduction of the communication by 95% contributes to a significant improvement in the speedup. However, the improvement in speedup is not as high as compared with the steady-flow case. This is because of the difference in the time-stepping schemes between the two cases. In the steady-flow case, local time stepping is used, which means for $k = 10$, communication takes place every 10 computation steps for all interfaces. In the unsteady-flow case, each block picks a certain time step, which can be different from other blocks. Hence, for $k = 20$, the number of computation steps before communication occurs can vary from 4 to 20 for various blocks. This can cause communication bottlenecks, which could account for the lower efficiency improvements when compared with the steady-flow case.

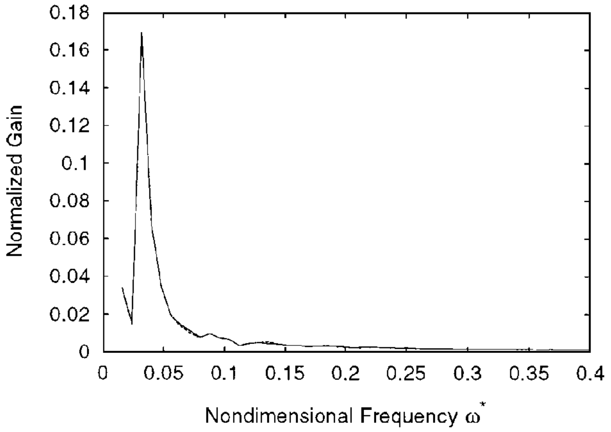

Fig. 16 Frequency response of solutions for base case (unsteady) and k = 10 with filtering.

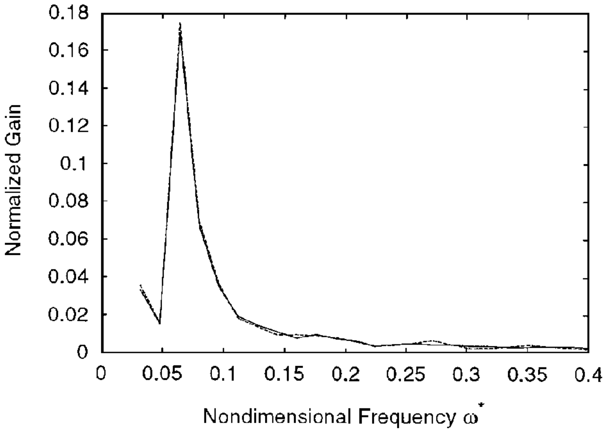

Fig. 17 Frequency response of solutions for base case (unsteady) and k = 20 with filtering.

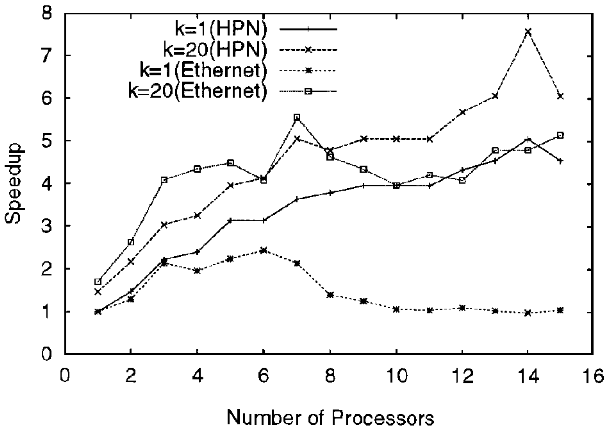

Fig. 18 Comparison of speedup with filtering (k = 20) and base case (unsteady).

At this point, there is also a question as to how much farther k can be increased. It was found that for $k = 30$ the error introduced shifts to the lower end of the frequency spectrum where the majority of the accurate solution is contained for both steady and unsteady cases. Because of this interaction, it is very hard to filter the error without affecting the solution itself. One can determine an upper limit on k by analyzing the solution when using a particular k and verifying whether oscillations with a wavelength of $4k\Delta t$ are filtered. If not, k can be reduced until the filter proves to be effective.

Conclusions

In this paper, a filtering procedure is presented to improve the efficiency of parallel computation of Euler equations using an explicit scheme. It is demonstrated that, in terms of obtaining an accurate solution, the time step chosen by the stability condition for each block may be too restrictive. One can reduce the communication between the blocks by 90–95% and still obtain an accurate solution. The filtering procedure coupled with the variable time-stepping schemes enables efficient utilization of the parallel algorithm for both steady and unsteady flows. It is illustrated that one can communicate with neighboring blocks only when necessary and improve efficiency. Heterogeneity of the flowfield and the computer systems is exploited for this purpose. Study of the interface conditions in the frequency domain provides insight into the problem. Similar filters can be developed for schemes other than Runge–Kutta schemes.

Acknowledgments

This research was funded by the NASA Lewis Research Center under NAG3-1577. The authors appreciate the support provided by Rich Blech, Gary Cole, and Joongkee Chung of the Computational Technologies Branch of NASA Lewis Research Center.

References

[1] Cooper, G. K., and Sirbaugh, J. R., "The PARC Code: Theory and Usage," Arnold Engineering Development Center, TR-89-15, Arnold AFB, TN, 1989.

[2] Gopalaswamy, N., Akay, H. U., Ecer, A., and Chien, Y. P., "Parallelization and Dynamic Load Balancing of NPARC Codes," AIAA Paper 96-3302, July 1996.

[3] Chien, Y. P., Ecer, A., Akay, H. U., Carpenter, F., and Blech, R. A., "Dynamic Load Balancing on a Network of Workstations for Solving Computational Fluid Dynamics Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 119, No. 1, 1994, pp. 17–33.

[4] Hirsch, C., *Numerical Computation of Internal and External Flows*, Vol. 1, Wiley, New York, 1989, Chap. 10.

[5] Chung, J., "Numerical Solution of a Mixed Compression Supersonic Inlet Flow," AIAA Paper 94-0583, Jan. 1994.

[6] Vichnevetsky, R., and Bowles, J. B., "Fourier Analysis of Numerical Approximations of Hyperbolic Equations," *SIAM Studies in Applied Mathematics*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1982, pp. 51–74.

J. Kallinderis
Associate Editor