045668

NASA Technical Paper 3686

Efficient Parallel Algorithm for Direct Numerical Simulation of Turbulent Flows

Stuti Moitra and Thomas B. Gatski

December 1997

NASA Technical Paper 3686

Efficient Parallel Algorithm for Direct Numerical Simulation of Turbulent Flows

Stuti Moitra and Thomas B. Gatski
Langley Research Center • Hampton, Virginia

National Aeronautics and Space Administration
Langley Research Center • Hampton, Virginia 23681-2199

December 1997

The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

Available electronically at the following URL address: http://techreports.larc.nasa.gov/ltrs/ltrs.html

Printed copies available from the following:

NASA Center for AeroSpace Information
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

Symbols

$\mathbf{A}, \mathbf{B}, \mathbf{C}$	matrices denoting convective derivatives
C_f	skin friction
c_k	velocity value at *k*th realization
$\mathbf{I}$	identity matrix
I, J, K	grid indices in x, y, and z directions
K	kinetic energy
$\mathbf{M}, \mathbf{N}$	linearized viscous derivative terms
n	total number of points in x direction
P	probability function (see eq. (8))
$\mathbf{P}$	matrix of left eigenvectors (see eq. (7))
P	total number of processors
p	pressure
R, S, T	viscous stress in x, y, and z direction, respectively
T	time scale over which ensemble is averaged (see eq. (8))
t	time
t_0	arbitrary initial time where averaging begins
$\overline{U}$	mean streamwise velocity component
u, v, w	velocity in x, y, and z direction, respectively
$\overline{V}$	mean normal velocity component
α, β	coefficients used in differencing scheme
Δ	forward differences
Δt	grid spacing in x direction
Λ	diagonal matrix containing eigenvalues
Λ^+	matrix of positive eigenvalues
Λ^-	matrix of negative eigenvalues
ρ	density
∇	backward differences
Subscripts:	
i, j, k	grid indices
k	realization in ensemble (see eq. (8))
x, y, z	derivation in x, y, and z direction, respectively
Superscripts:	
n	index of time step
p	iteration index
$+$	forward differences
$-$	backward differences

Abbreviations:

CFD	computational fluid dynamics
DNS	direct numerical simulation
PIOFS	parallel input/output file system
RISC	reduced instruction set computer

Abstract

A distributed algorithm for a high-order-accurate finite-difference approach to the direct numerical simulation (DNS) of transition and turbulence in compressible flows is described. This work has two major objectives. The first objective is to demonstrate that parallel and distributed-memory machines can be successfully and efficiently used to solve computationally intensive and input/output intensive algorithms of the DNS class. The second objective is to show that the computational complexity involved in solving the tridiagonal systems inherent in the DNS algorithm can be reduced by algorithm innovations that obviate the need to use a parallelized tridiagonal solver.

1. Introduction

Computational techniques such as direct numerical simulation (DNS) often have massive associated memory requirements. Storage and computing time on traditional supercomputers are becoming increasingly hard to obtain because of the prohibitive associated costs.

In recent years, computational fluid dynamics (CFD) techniques have been used successfully to compute flows that are related to geometrically complex configurations. Currently, interest in the computation of turbulent compressible flows has been revived to aid in the design of high-speed vehicles and associated propulsion systems. Databases of compressible turbulent flows are not as readily available as those for incompressible turbulent flows. The effort described in this paper is the first parallel application of DNS for a compressible, turbulent, supersonic spatially evolving boundary-layer flow.

A turbulent flow is simulated at moderate Reynolds numbers. A simulation at a moderate Reynolds number can be used as a guide in understanding the fundamental turbulent physics of such flows and can provide relevant Reynolds stress and dissipation rate budgets, which have been proved useful in the incompressible studies.

The computational method used by Rai and Moin (ref. 1) is a high-order-accurate, upwind-biased, implicit finite-difference method. The technique involves the use of upwind-biased differences for the convective terms, central differences for the viscous terms, and an iterative implicit time-integration scheme. The computational method uses the nonconservative form of the governing equations to obtain the solution. This method has been shown in reference 2 to control the aliasing error through the natural dissipation in the scheme but at the expense of some accuracy. However, this loss in accuracy can be overcome by increasing the number of grid points. In the present study, the method presented in reference 2 is extended to solve the compressible form of the Navier-Stokes equations.

The lack of efficient and accurate general tridiagonal solvers is a serious bottleneck in obtaining parallel solutions of problems that involve large tridiagonal systems, such as those in the DNS algorithm. Existing parallel tridiagonal system solvers are limited in their applicability to matrices of particular forms, and their accuracies in the context of general tridiagonal systems are questionable. A significant innovation in the present work is the elimination of the need for parallel tridiagonal solvers. A novel algorithm has been developed to partition the tridiagonal system so that each node can solve its own portion of the system with no sacrifice in accuracy. The scheme and the boundary conditions that are employed herein are described in detail.

2. Numerical Method

The integration method utilized in this work for the compressible form of the Navier-Stokes equations is a high-order-accurate, upwind-biased finite-difference technique that is used in conjunction with an iterative, implicit time-advancement scheme. The convective terms are evaluated as in reference 3 (but with a high-order-accurate differencing technique), and the iterative, implicit technique of reference 4 is used to integrate the equations of motion in time. Essentially, the method is based on an unsteady, compressible, nonconservative formulation of the Navier-Stokes equations in three spatial dimensions as follows:

$$\mathbf{Q}_t + \mathbf{A}\mathbf{Q}_x + \mathbf{B}\mathbf{Q}_y + \mathbf{C}\mathbf{Q}_z = \frac{1}{\rho}(R_x + S_y + T_z) \tag{1}$$

where

$$\mathbf{Q} = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix} \tag{2}$$

In equation (2) ρ is the density, *u*, *v*, and *w* are the velocities in the *x*, *y*, and *z* directions, respectively, and *p* is the pressure.

Before the time-integration method is described, the techniques used to compute the convective and viscous terms should be examined. The convective terms $\mathbf{AQ}_x$, $\mathbf{BQ}_y$, and $\mathbf{CQ}_z$ in equation (1) are evaluated as in reference 3. To illustrate the technique, consider the term $\mathbf{AQ}_x$. The matrix **A** can be written as

$$\mathbf{A} = \mathbf{P}\Lambda\mathbf{P}^{-1} \tag{3}$$

where **P** is the matrix of the left eigenvectors of **A** and Λ is a diagonal matrix that contains the eigenvalues of **A**. The term $\mathbf{AQ}_x$ is evaluated as

$$\mathbf{AQ}_x = \mathbf{A}^+\mathbf{Q}_x^- + \mathbf{A}^-\mathbf{Q}_x^+ \tag{4}$$

where $\mathbf{A}^{\pm} = \mathbf{P}\Lambda^{\pm}\mathbf{P}^{-1}$ and Λ^+ and Λ^- are diagonal matrices that contain the positive and negative eigenvalues of **A**, respectively. The terms $\mathbf{Q}_x^+$ and $\mathbf{Q}_x^-$ are forward and backward differences of the vector **Q**, respectively. In the present study, $\mathbf{Q}_x^+$ and $\mathbf{Q}_x^-$ are computed by using fifth-order-accurate forward- and backward-biased finite differences with a seven-point stencil as

$$\mathbf{Q}_x^- = \frac{-6\mathbf{Q}_{i+2} + 60\mathbf{Q}_{i+1} + 40\mathbf{Q}_i - 120\mathbf{Q}_{i-1} + 30\mathbf{Q}_{i-2} - 4\mathbf{Q}_{i-3}}{120\Delta x} \tag{5a}$$

$$\mathbf{Q}_x^+ = \frac{4\mathbf{Q}_{i+3} - 30\mathbf{Q}_{i+2} + 120\mathbf{Q}_{i+1} - 40\mathbf{Q}_i - 60\mathbf{Q}_{i-1} + 6\mathbf{Q}_{i-2}}{120\Delta x} \tag{5b}$$

on a grid that is equispaced in the *x* direction. The remaining convective terms are evaluated in a similar manner. The viscous terms R_x, S_y, and T_z in equation (1) are computed by using central differences and a five-point stencil (fourth-order accuracy). The fully implicit finite-difference representation of equation (1) in factored form is given by

$$\left[\alpha\mathbf{I} + \beta\Delta t\left(\frac{\mathbf{A}^{+}\nabla_{x}}{\nabla_{x_i}} + \frac{\mathbf{A}^{-}\Delta_{x}}{\Delta_{x_i}}\right)\right]^{p}\left\{\alpha\mathbf{I} + \beta\Delta t\left[\frac{\mathbf{B}^{+}\nabla_{y}}{\nabla_{y_j}} + \frac{\mathbf{B}^{-}\Delta_{y}}{\Delta_{y_j}} - \mathbf{M}\left(\frac{\Delta_{y}}{\Delta_{y_j}} + \frac{\nabla_{y}}{\nabla_{y}}\right) - \mathbf{N}\left(\frac{\Delta_{y}}{\Delta_{y_j}} - \frac{\nabla_{y}}{\nabla_{y}}\right)\right]\right\}^{p}$$

$$\times\left[\alpha\mathbf{I} + \beta\Delta t\left(\frac{\mathbf{C}^{+}\nabla_{x}}{\nabla_{z_k}} + \frac{\mathbf{C}^{-}\Delta_{z}}{\Delta_{z_k}}\right)\right]^{p}(\mathbf{Q}^{p+1} - \mathbf{Q}^{p})$$

$$= -\Delta t\left\{\frac{3\mathbf{Q}^{p} - 4\mathbf{Q}^{n} + \mathbf{Q}^{n-1}}{2\Delta t} + \left(\mathbf{A}^{+}\mathbf{Q}_{x}^{-} + \mathbf{A}^{-}\mathbf{Q}_{x}^{+} + \mathbf{B}^{+}\mathbf{Q}_{y}^{-} + \mathbf{B}^{-}\mathbf{Q}_{y}^{+} + \mathbf{C}^{+}\mathbf{Q}_{z}^{-} + \mathbf{C}^{-}\mathbf{Q}_{z}^{+}\right)^{p}\right.$$

$$\left. - \left[\frac{1}{\rho}(R_x + S_y + T_z)\right]^{p}\right\} \tag{6}$$

where $\alpha = 1.5^{1/3}$, $\beta = 1.5^{-2/3}$, ∇ and Δ are backward and forward difference operations, respectively, the matrices **M** and **N** represent the linearization of the first and second derivatives in the viscous terms, and the superscript p is an iteration index. The factored form of equation (1) results in the systems of block-tridiagonal matrices shown in equation (6). One additional approximation that has been made to the x and z terms on the left-hand side of equation (6) is the use of the diagonal form as given in reference 5. The inversion in the x direction is approximated as

$$\left[\alpha\mathbf{I} + \beta\Delta t\left(\frac{\mathbf{A}^{+}\nabla_{x}}{\nabla_{x_i}} + \frac{\mathbf{A}^{-}\Delta_{x}}{\Delta_{x_i}}\right)\right] \approx \mathrm{P}\left[\alpha\mathrm{I} + \beta\Delta t\left(\frac{\Lambda^{+}\nabla_{x}}{\nabla_{x_i}} + \frac{\Lambda^{-}\Delta_{x}}{\Delta_{x_i}}\right)\right]\mathbf{P}^{-1} \tag{7}$$

where the matrices **P**, $\mathbf{P}^{-1}$, and Λ are defined in equation (3). The approximation results in systems of scalar tridiagonal equations instead of block-tridiagonal equations. The inversion in the z direction is treated in a similar manner.

3. Parallel Hardware Platform

This computation scheme was implemented on the 48-node IBM SP2 computer available at the Langley Research Center. The IBM SP2 is a distributed-memory parallel computer that adopts a message-passing communication paradigm and supports virtual memory. Each processor is functionally equivalent to an IBM reduced instruction set computer (RISC) system/6000 deskside system with 128 Mb of local memory. The key component in all distributed-memory parallel computers is the interconnection network that links the processors together. The SP2 high-performance switch is a multistage packet-switched omega network that provides a minimum of four paths between any pair of nodes in the system.

4. Parallel Algorithm and Implementation

In the present work, the computation domain is split equally among processors along the streamwise direction. The purpose of the data distribution strategy is to balance the load on the processors and minimize communication between processors. In the physical domain, the coordinate directions x (streamwise), y (normal), and z (spanwise) are associated with the indices I, J, and K, respectively. The x direction has the largest number of points associated with it, and more importantly the gradients of the flow quantities are small in this direction. Both the y and z directions have a much smaller number of associated points. Moreover, the y direction, because it is the viscous direction, has very high gradients in its flow quantities. This large gradient can introduce a large error that results from data transfer at the node boundaries. Thus the *x* direction was chosen for the data partitioning.

Partitioning in the I direction results in the assignment of an equal number of J-K planes to each processor. The number of planes assigned to each processor is equal to I_{max}/P, where P is the total number of processors and I_{max} is the total number of points in the I direction. This data-partitioning strategy is shown schematically in figure 1.

The realistic simulation of turbulence and transition necessitates a grid with approximately 17 million reference grid points. (See ref. 5.) For grids of this size, all data planes that are assigned to a node cannot be kept in the core memory of the IBM SP2 computer. The IBM SP2 computer has 128 Mb of memory per processor, of which approximately 72 Mb are currently available for computing at Langley Research Center. Storing of all data assigned to a node in the core memory leads to a prohibitively large volume of paging and, consequently, a serious degradation in performance. The strategy employed here for reducing core memory requirements is to keep only seven planes of data in the core memory at any given time. A minimum of seven planes is necessary to implement the seven-point stencil used in the formulation of the difference equations. This strategy was implemented by using a parallel input/output file system (PIOFS) as a secondary storage device. Planes are read into the nodes from the PIOFS as they are needed. The following procedure is executed by each node to update the planes assigned to the node. To update the first plane, the node reads in the first four planes from PIOFS. These four planes are then shifted three places to the right. Three planes from the preceding node are then brought into the three empty spaces at the left through communication between the nodes. The first plane is then updated. To update each successive plane, the existing planes are shifted one place to the left, and a new plane is read from PIOFS into the vacant space on the right. A schematic presentation of this process is shown in figure 2.

A similar procedure is used to update the last three planes assigned to each node; the only difference is that communication is performed with the succeeding node. The first three planes on the first node and the last three planes assigned to the last node are handled specially because they represent the physical boundary condition.

In this study, the convective derivatives are represented with a seven-point stencil, which evokes a certain pattern of data transfer. Each processor has a total (n/P) I planes, numbered from 0 to (n/P) − 1. To perform the finite differencing with the seven-point stencil, each processor sends planes numbered (n/P) − 1, (n/P) − 2, and (n/P) − 3 to its successor. In addition planes numbered 0, 1, and 2 are sent to the predecessor node.

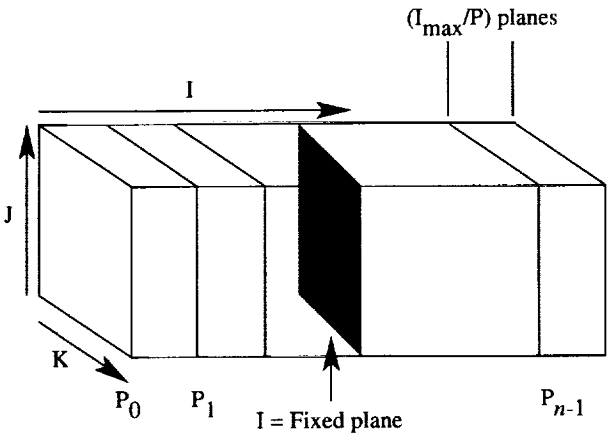
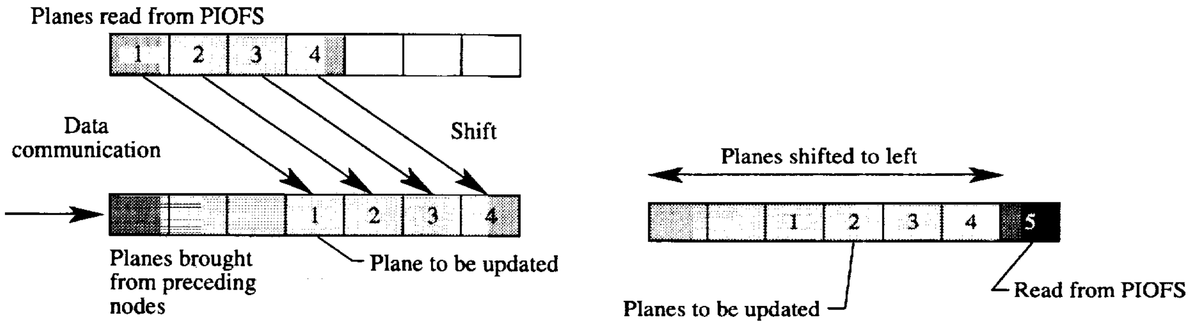


Figure 1. Schematic of data-partitioning strategy.



(a) Initial shifting of planes to right.

(b) Update of successive planes by shifting to left.

Figure 2. Schematic representation of updating process.

The governing systems of equations associated with the x direction are given by equation (7) which gives rise to a scalar tridiagonal system. Equipartition of the data creates smaller tridiagonal systems that reside on the nodes. If the original tridiagonal system is of size $n \times n$, then the smaller tridiagonal system that resides on each node will be of size $n/\mathrm{P} \times n/\mathrm{P}$, where P is the total number of processors.

The difference between the original and the partitioned system is explained schematically below. The linear system of equations $\mathbf{A}x = d$ for the original system has the following representation:

$$\begin{bmatrix} b_0 & c_0 & & & & & & \\ a_1 & b_1 & c_1 & & & & & \\ & \cdot & \cdot & \cdot & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & a_{n-2} & b_{n-2} & c_{n-2} \\ & & & & & & a_{n-1} & b_{n-1} \end{bmatrix}$$

The a_0 in the first row and c_{n-1} in the last row represent the physical boundary conditions. For the partitioned system, the subsystem that resides on each node is represented as

$$\begin{bmatrix} b_0 & c_0 & & & & & & \\ a_1 & b_1 & c_1 & & & & & \\ & \cdot & \cdot & \cdot & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & a_{(n/\mathrm{P})-2} & b_{(n/\mathrm{P})-2} & c_{(n/\mathrm{P})-2} \\ & & & & & & a_{(n/\mathrm{P})-1} & b_{(n/\mathrm{P})-1} \end{bmatrix}$$

For the first node, a_0 is a physical boundary condition; for all other nodes, a_0 must be brought in from the predecessor node through the data communication.

Similarly for the last node, $c_{(n/P)-1}$ is a physical boundary condition; for all other nodes $c_{(n/P)-1}$ must be brought in from the succeeding processor through the data communication. Data needed from neighboring nodes are immediately available because they are brought in through the communication scheme. In the original, undivided problem, the most recent values of the point solutions are used in the back substitution phase. In the distributed case, the node-interface boundary condition values retain their values from the previous time step until a redistribution is accomplished at the next time step. Therefore, a time lag occurs for solutions at the node boundaries. The resulting error can be driven down if necessary by iterations per time step. In the original scheme implemented by Rai and Moin (ref. 1), the performance of additional iterations was necessary at each time step to reduce the linearization error; thus, the number of iterations required to reduce the error at the boundary interfaces does not significantly increase the computational cost.

The algorithmic innovations described have been shown to be an efficient method for the computations of memory-intensive problems such as the DNS formulation. Note that the goal of this parallelization technique is to utilize distributed-memory machines to solve very large problems with considerable memory requirements that exhaust the capabilities of traditional supercomputers. In this context, scaling of the problem has no relevance; thus, timing studies are not warranted. Conventional performance studies that use speedup comparisons are, therefore, not attempted in this work. Verification of the accuracy of the results obtained with the IBM SP2 machine is reported in the following section.

5. Validation

Although the procedure outlined previously has demonstrated a method for executing these large-scale computations, the ultimate test is to verify that equivalent results can be obtained from the CRAY Y-MP and the IBM SP2 computers. Recently, a spatially developing, compressible, flat-plate turbulent boundary-layer flow reference was simulated. (See ref. 5.) The computational domain that was required for this spatial simulation is shown in figure 3. To obtain accurate results, a computational grid of (971 × 55 × 321) reference grid points (ref. 5) in the x, y, and z directions, respectively, was required. These requirements are extensive and clearly indicate the need to advance to distributed-memory machines such as the IBM SP2. With this motivation, the same flow field is simulated on both the CRAY Y-MP and the IBM SP2 computers. A coarser computational mesh was used in these calculations because the physical accuracy of the results was not of interest; rather only the relative equivalence of the two simulations was evaluated.

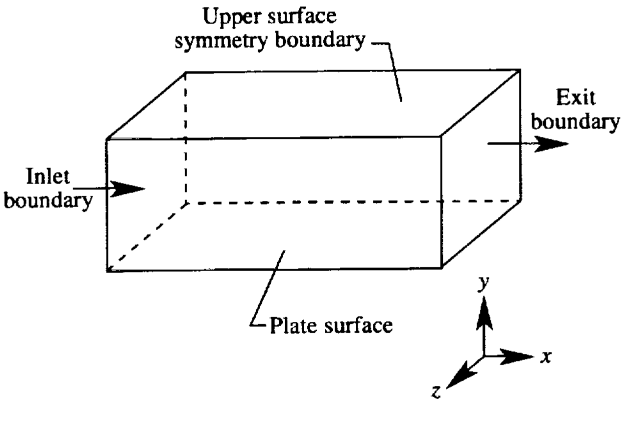


Figure 3. Schematic of computational domain (not to scale).

A fully turbulent flow is a stochastic system that is characterized by its statistical properties. At each spatial location, every time step in the computational procedure is a single realization within a large ensemble of events. Some flows, such as the wall-bounded flow considered here, have two useful properties: they are homogeneous in one spatial direction (*z*) and they are stationary. Thus in the *z* direction a suitable spatial average over several integral scales yields a mean value that is independent of *z*; similar results occur for a temporal average. Thus, effective comparisons between the simulations from each machine can only be accomplished through comparisons of the statistical moments. An explanation of the first two moments, that is, the mean (first moment) and the variance (second central moment), is sufficient.

For illustrative purposes, the instantaneous streamwise velocity is used to demonstrate how these moments can be extracted. The time mean of the instantaneous streamwise velocity *u(x,t)*, for example, is given by

$$\begin{aligned}\bar{U}(x) &= \sum_k U_k(x,t_k) P[t|c_{k-1}(x) \le U(x,t) < c_k(x)] \\ &= \lim_{T \to \infty} \frac{1}{T} \int_{t_0}^{t_0+T} U(x,t)\, dt\end{aligned} \tag{8}$$

where $P[t|c_{k-1}(x) \le U(x,t) < c_k(x)]$ is the probability that the function *U(x,t)* is delimited by $[c_{k-1}(x), c_k(x)]$. Because the process is assumed to be ergodic, the ensemble and time averages shown in equation (8) are equal. Analogous relationships also hold for the homogeneous spatial directions as well. In turbulent flows for which the velocity field is stationary, the flow field customarily is decomposed into its mean and fluctuating parts which are given by

$$U(x,t) = \bar{U}(x) + u(x,t) \tag{9}$$

This decomposition is called a Reynolds decomposition and is interpreted as a partitioning of the flow field into a deterministic mean-flow part and a random turbulent part. The turbulent part, thus, has zero time mean (i.e., $\bar{u} = 0$). Given this property, the second central moment can be obtained in a straightforward manner from the following association with the turbulent part of the flow field:

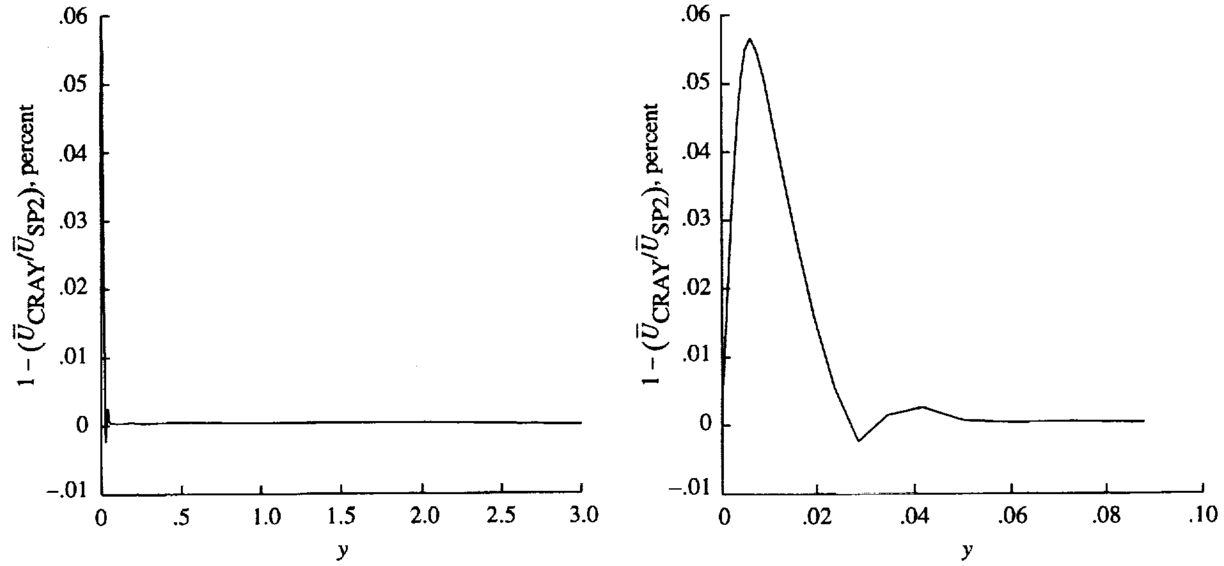
$$u^2(x) = U^2(x,t) - \bar{U}^2(x) \tag{10}$$

In the boundary-layer flow considered here, the mean flow is two-dimensional, with mean velocity components $\bar{U}$ and $\bar{V}$ that correspond to the *x* and *y* directions; the turbulence is always three-dimensional with the components *u*, *v*, and *w* that correspond to the *x*, *y*, and *z* directions. In the analysis of turbulent flows, most comparisons are based either on the mean streamwise velocity and the turbulent kinetic energy or on twice the variance of the total velocity field.

Figure 4 shows a relative comparison of the results obtained from the CRAY Y-MP and the IBM SP2 computers for the normalized (with $\bar{U}_\infty$) streamwise velocity across the boundary layer in the fully turbulent regime. The percent difference across the layer is extremely small; the maximum difference occurs near the wall. This near-wall region is shown in figure 4(b), and it is seen that the maximum difference is approximately 0.06 percent.

The same type of comparison is shown in figure 5 for the second moment quantity, the turbulent kinetic energy *K* (where $2K = u^2 + v^2 + w^2$ is the total variance). With the free-stream kinetic energy used to perform scaling, the figure shows the same behavior as the mean velocity. The difference is similar and the maximum occurs near the wall.

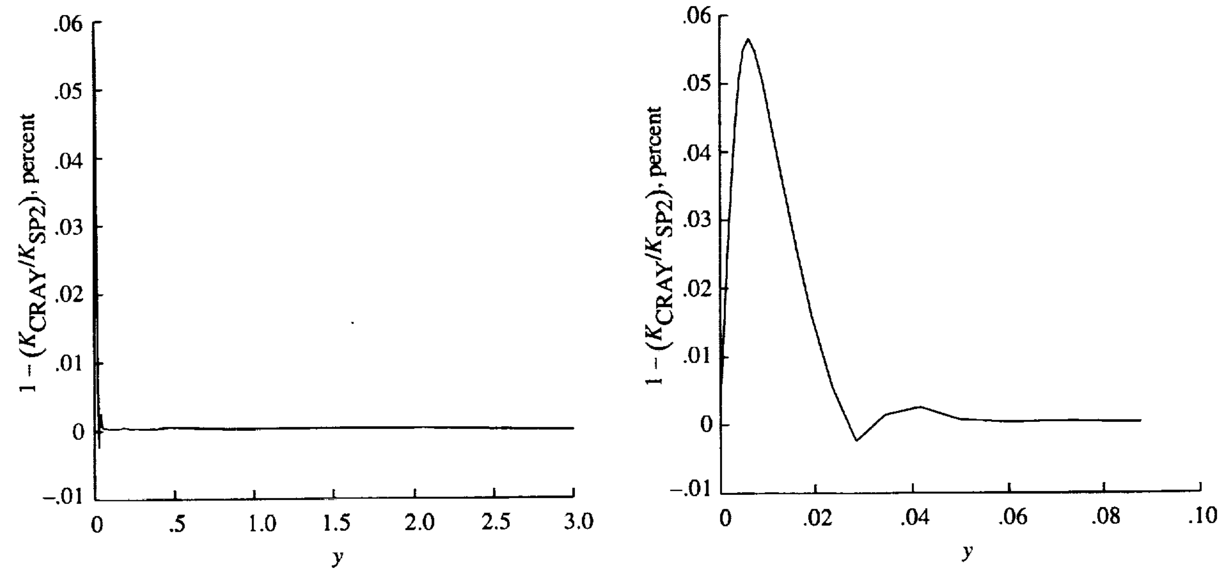
Unlike the previous figures, which compared the mean velocity and turbulent kinetic energy as a function of distance from the wall at a fixed streamwise station, figure 6 shows the variation of the skin friction C_f as a function of streamwise distance along the plate. The origin of the system is at $x_t = 4.0$



(a) Full field.

(b) Near-wall region.

Figure 4. Variation of streamwise mean velocity across boundary layer.



(a) Full field.

(b) Near-wall region.

Figure 5. Variation of turbulent kinetic energy across boundary layer.

and the length of the domain shown in figure 6 is 11 units. The figure shows some interesting features that are not shown in the previous plots where the variation was normal to the wall. In the laminar and transition regions where $0 < x - x_t < 2$, the percentage of variation between the results from the CRAY Y-MP and the IBM SP2 computers is maximum. However, in the region that is delimited by $2 < x - x_t < 5$, the percentage of deviation is small, but high-frequency spikes appear in the distribution. This region is the fully turbulent region in which a broad spectrum of scales (both spatial and temporal) exists. Note, however, that figure 6 represents only a partially converged solution for both machines. The solution scheme that was implemented on the SP2 data communication at the node boundaries introduces a small additional error caused by the time lag of the boundary condition. This error is

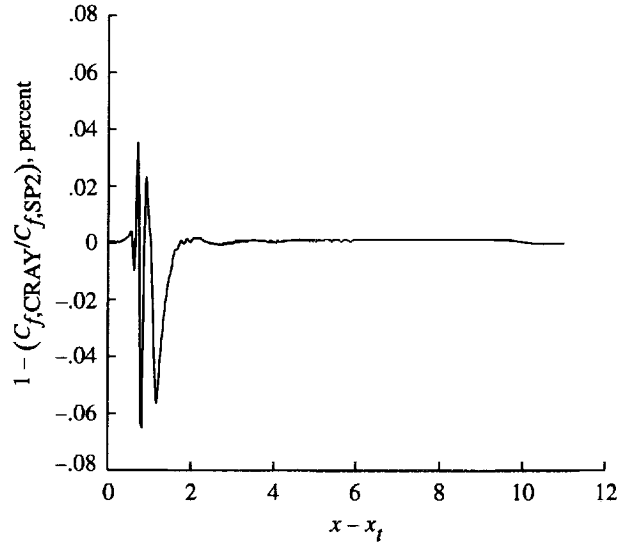


Figure 6. Variation of skin friction as a function of streamwise distance along the plate.

expected to decrease with the number of iterations. Thus magnitude of this error would be negligible for the large number of iterations required for convergence of the solution. Confirmation of this explanation is shown in the region $5 < x - x_t < 11$, where no spikes are seen. In this downstream region, the grid is coarsened to damp the disturbances and to allow for the implementation of outflow boundary conditions. Fewer disturbances mean less time-dependent behavior and less error caused by nodal communication. Although these high-frequency spikes are extremely small and generally would have no real effect on the physics of the flow, they are noteworthy because under resolved simulations or inappropriate parallel algorithms they can have a significant effect on the flow physics.

6. Concluding Remarks

A distributed algorithm for a spatially developing, compressible, flat-plate turbulent boundary-layer flow has been developed and implemented on the IBM SP2 computer at the Langley Research Center. The method does not employ a parallelized tridiagonal system solver and uses a parallel input/output file system (PIOFS) as a secondary storage device to alleviate the considerable memory requirement for this large-scale computational problem. A relative comparison of the accuracy obtained from the CRAY Y-MP and the IBM SP2 computers demonstrates the validity of this method.

NASA Langley Research Center
Hampton, VA 23681-2199
October 23, 1997

7. References

1. Rai, Man Mohan; and Moin, Parviz: Direct Numerical Simulation of Transition and Turbulence in Spatially Evolving Boundary Layer. *J. Comput. Phys.*, vol. 109, no. 2, Dec. 1993, pp. 169–192.

2. Rai, Man Mohan; and Moin, Parviz: Direct Simulations of Turbulent Flow Using Finite-Difference Schemes. AIAA-89-0369, Jan. 1989.

3. Chakravarthy, S. R.; Anderson, D. A.; and Salas, M. D.: The Split Coefficient Matrix Method for Hyperbolic Systems of Gasdynamic Equations. AIAA-80-0268, Jan. 1980.

4. Rai, M. M.; and Chakravarthy, S. R.: An Implicit Form for the Osher Upwind Scheme. *AIAA J.*, vol. 24, May 1986, pp. 735–743.

5. Rai, Man M.; Gatski, Thomas B.; and Erlebacher, Gordon: Direct Simulation of Spatially Evolving Compressible Turbulent Boundary Layers. AIAA-95-0583, Jan. 1995.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1997

3. REPORT TYPE AND DATES COVERED
Technical Paper

4. TITLE AND SUBTITLE
Efficient Parallel Algorithm for Direct Numerical Simulation of Turbulent Flows

5. FUNDING NUMBERS
WU 505-59-50

6. AUTHOR(S)
Stuti Moitra and Thomas B. Gatski

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Langley Research Center
Hampton, VA 23681-2199

8. PERFORMING ORGANIZATION REPORT NUMBER
L-17638

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TP-3686

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited
Subject Category 34
Availability: NASA CASI (301) 621-0390

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A distributed algorithm for a high-order-accurate finite-difference approach to the direct numerical simulation (DNS) of transition and turbulence in compressible flows is described. This work has two major objectives. The first objective is to demonstrate that parallel and distributed-memory machines can be successfully and efficiently used to solve computationally intensive and input/output intensive algorithms of the DNS class. The second objective is to show that the computational complexity involved in solving the tridiagonal systems inherent in the DNS algorithm can be reduced by algorithm innovations that obviate the need to use a parallelized tridiagonal solver.

14. SUBJECT TERMS
Direct numerical simulation (DNS); Parallel input/output file system (PIOFS); Distributed memory; Message-passing

15. NUMBER OF PAGES
15

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT