

NASA Technical Memorandum 106823  
ICOMP-94-32; AIAA-95-0577

11-24  
E. J. ...  
15P

# Parallel Navier-Stokes Computations on Shared and Distributed Memory Architectures

M. Ehtesham Hayder  
*Institute for Computational Mechanics in Propulsion*  
*Lewis Research Center*  
*Cleveland, Ohio*

D.N. Jayasimha  
*Ohio State University*  
*Columbus, Ohio*

(NASA-TM-106823) PARALLEL  
NAVIER-STOKES COMPUTATIONS ON  
SHARED AND DISTRIBUTED MEMORY  
ARCHITECTURES (NASA, Lewis  
Research Center) 15 p

N95-18941

Unclass

and

G3/34 0035002

Sasi Kumar Pillay  
*Lewis Research Center*  
*Cleveland, Ohio*

Prepared for the  
33rd Aerospace Sciences Meeting and Exhibit  
sponsored by the American Institute of Aeronautics and Astronautics  
Reno, Nevada, January 9-12, 1995



National Aeronautics and  
Space Administration





# Parallel Navier-Stokes Computations on Shared and Distributed Memory Architectures

M. Ehtesham Hayder

Institute for Computational Mechanics in Propulsion  
Ohio Aerospace Institute  
NASA Lewis Research Center, Cleveland, OH

D. N. Jayasimha\*

Department of Computer Science  
The Ohio State University, Columbus, OH

Sasi Kumar Pillay

Chief, Computer Graphics Branch  
Computer Services Division  
NASA Lewis Research Center, Cleveland, OH

## Abstract

We study a high order finite difference scheme to solve the time accurate flow field of a jet using the compressible Navier-Stokes equations. As part of our ongoing efforts, we have implemented our numerical model on three parallel computing platforms to study the computational, communication, and scalability characteristics. The platforms chosen for this study are a cluster of workstations connected through fast networks (the LACE experimental testbed at NASA Lewis), a shared memory multiprocessor (the Cray YMP), and a distributed memory multiprocessor (the IBM SP1). Our focus in this study is on the LACE testbed. We present some results for the Cray YMP and the IBM SP1 mainly for comparison purposes. On the LACE testbed, we study: (a) the communication characteristics of Ethernet, FDDI and the ALLNODE networks and (b) the overheads

induced by the PVM message passing library used for parallelizing the application. We demonstrate that clustering of workstations is effective and has the potential to be computationally competitive with supercomputers at a fraction of the cost.

**Key words:** Navier-Stokes computations, parallel computing, cluster computing.

## 1. Introduction

Numerical simulations can play an important role in examining the physical processes associated with many important practical problems. The suppression of jet exhaust noise is one such problem which will have a great impact on the success of the High Speed Civil Transport plane (HSCT). The radiated sound emanating from the jet can be computed by solving the full (time-dependent) compressible Navier-Stokes equations. This computation can, however, be very expensive and time consuming. The difficulty can be partially overcome by limiting the solution domain to the near field where the jet is nonlinear and then using acoustic analogy [e.g., Lighthill<sup>1</sup>] to relate the far-field noise to the near-field sources. The latter technique requires obtaining the time-dependent flow field. In this study we concentrate on such flow fields near the nozzle exit.

---

\*Most of this work was performed while the author was in residence at Ohio Aerospace Institute, NASA Lewis Research Center, Cleveland, OH.

Copyright ©1995 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein or Governmental purposes. All other rights are reserved by the copyright owner.

The advent of massively parallel processors and clustered workstations gives the opportunity for scientists to parallelize their computationally intensive codes and reduce turnaround time. Recognizing this, a number of researchers<sup>2 3 4 5</sup> have studied CFD applications on specific parallel architectures. Our goal is to implement a particular CFD application code which uses a high order finite difference scheme on a variety of parallel computing platforms to study the computational, communication, and scalability characteristics. The platforms chosen for this study are a cluster of workstations connected through fast networks (the LACE<sup>6</sup> experimental testbed at NASA Lewis), a shared memory multiprocessor (the Cray YMP), and a distributed memory multiprocessor (the IBM SP1). Clustered workstations such as the LACE are becoming increasingly popular because they show promise as a low cost alternative to expensive supercomputers and massively parallel processors. We have followed a pragmatic approach by parallelizing only those segments of the algorithm which are computationally intensive (identified using a profiler). Such an approach can reduce the time for parallelizing "dusty deck" sequential codes. Our distributed memory implementation on the LACE uses PVM<sup>7</sup>, a portable message passing library developed at the Oak Ridge National Laboratory.

In the next two sections we briefly discuss the governing equations and the numerical model. Section 4 has a discussion of the parallel architectures used in the study and the parallelisation schemes. In Section 5 we present an analytical model for computation and communication costs and discuss simulation results on the three platforms used in this study.

## 2. Governing Equations

We solve the Navier-Stokes and the Euler equations to compute flow fields of an axisymmetric jet. The Navier-Stokes equations for such flows in polar coordinates can be written as

$$LQ = S$$

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial r} = S$$

where

$$Q = r \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}$$

$$F = r \begin{pmatrix} \rho u \\ \rho u^2 - \tau_{xx} + p \\ \rho uv - \tau_{xr} \\ \rho uH - u\tau_{xx} - v\tau_{xr} - \kappa T_x \end{pmatrix}$$

$$G = r \begin{pmatrix} \rho v \\ \rho uv - \tau_{xr} \\ \rho v^2 - \tau_{rr} + p \\ \rho vH - u\tau_{xr} - v\tau_{rr} - \kappa T_r \end{pmatrix}$$

$$S = \begin{pmatrix} 0 \\ 0 \\ p - \tau_{\theta\theta} \\ 0 \end{pmatrix}$$

$F$  and  $G$  are the fluxes in the  $x$  and  $r$  directions respectively, and  $S$  is the source term that arises in the cylindrical polar coordinates, and  $\tau_{ij}$  are the shear stresses. One obtains the Euler equations from the above equations, by setting  $\kappa$  and all  $\tau_{ij}$  equal to zero.

## 3. Numerical Model

We use the fourth order MacCormack scheme, due to Gottlieb and Turkel<sup>8</sup>, to solve the Navier-Stokes and the Euler equations. For the present computations, the operator  $L$  in the equation  $LQ = S$  or equivalently  $Q_t + F_x + G_r = S$  is split into two one-dimensional operators and the scheme is applied to these split operators. For the one dimension model/split equation  $Q_t + F_x = S$ , we express the predictor step with forward differences as

$$\bar{Q}_i = Q_i^n + \frac{\Delta t}{6\Delta x} \{7(F_{i+1}^n - F_i^n) - (F_{i+2}^n - F_{i+1}^n)\} + \Delta t S_i$$

and the corrector step with backward differences as

$$Q_i^{n+1} = \frac{1}{2}[\bar{Q}_i + Q_i^n + \frac{\Delta t}{6\Delta x} \{7(\bar{F}_i - \bar{F}_{i-1}) - (\bar{F}_{i-1} - \bar{F}_{i-2})\} + \Delta t S_i]$$

This scheme becomes fourth-order accurate in the spatial derivatives when alternated with symmetric variants<sup>8</sup>. We also use the standard MacCormack scheme in our tests. This scheme is second order accurate in both space and time and can be written as

$$\bar{Q}_i = Q_i^n + \frac{\Delta t}{\Delta x} (F_{i+1}^n - F_i^n) + \Delta t S_i$$

$$Q_i^{n+1} = \frac{1}{2}[\bar{Q}_i + Q_i^n + \frac{\Delta t}{\Delta x} (\bar{F}_i - \bar{F}_{i-1}) + \Delta t S_i]$$

We define  $L_1$  as a one dimensional operator with a forward difference in the predictor and a backward difference in the corrector. Its symmetric variant  $L_2$  uses a backward difference in the predictor and a forward difference in the corrector. For our computations, the one dimensional sweeps are arranged as

$$Q^{n+1} = L_{1z} L_{1r} Q^n$$

$$Q^{n+2} = L_{2r} L_{2z} Q^{n+1}$$

This scheme is used for the interior points. In order to advance the scheme near boundaries the fluxes are extrapolated outside the domain to artificial points using a cubic extrapolation to compute the solution on the boundary. At the outflow boundary, we then solve the following set of equation to get the solution at the new time for all boundary points.

$$p_t - \rho c u_t = 0$$

$$p_t + \rho c u_t = R_2$$

$$p_t - c^2 \rho_t = R_3$$

$$v_t = R_4$$

where  $R_i$  is determined by which variables are specified and which are not. Whenever, the combination is not specified,  $R_i$  is just those spatial derivatives that come from the Navier-Stokes equations. Thus,  $R_i$  contains viscous contributions even though the basic format is based on inviscid characteristic theory. This framework of outflow boundary condition implementation is discussed by Hayder and Turkel<sup>9</sup>. Due to the presence of a singularity, special care is needed to treat the centerline. Here, we briefly outline one of the formulations for the centerline used in three dimensional calculations. In this treatment, equations corresponding to the Cartesian coordinates are solved at the centerline. Away from the centerline variables are solved in polar coordinates. We use  $r^2 = x^2 + y^2$ ,  $\tan\theta = y/x$ ,  $x = r\cos\theta$  and  $y = r\sin\theta$  to derive the equations at the centerline. The conservation equation for the Cartesian variables is

$$W_t + F_x + G_y + H_z = 0$$

or

$$W_t + \frac{(Fy_\theta - Gx_\theta)_r}{r} + \frac{(Gx_r - Fy_r)_\theta}{r} + H_z = 0$$

or

$$(rW)_t + (r\hat{F})_r + \hat{G}_\theta + (rH)_z = 0$$

where  $\hat{F} = F\cos\theta + G\sin\theta$  and  $\hat{G} = G\cos\theta - F\sin\theta$ . Because both F and G are single valued,  $F_\theta$  and  $G_\theta$  are zero at the centerline. Then using the L'Hospital rule at  $r=0$  we get

$$W_t + (F + G_\theta)_r \cos\theta + (G - F_\theta)_r \sin\theta + H_z = 0$$

The above equation is solved at the centerline in three dimensional calculations. For axisymmetric calculations,

we have options to use the L'Hospital rule or extrapolations to get the quantities at the centerline. Our numerical model is three dimensional. However, to maintain low computational costs, in this study we obtained only two dimensional results. Further discussion of our numerical model is given in Hayder et al<sup>10</sup> and Mankbadi et al.<sup>11</sup>

## 4. Parallel Implementations

### 4.1 Network of Workstations

The LACE (Lewis Advanced Cluster Environment) is a cluster of 33 IBM RS6000 workstations (node0 – node32). Node 0 is the file server. All the nodes are connected through two Ethernet networks (10 Mbits/sec). In addition, nodes 00–16 are connected through the much faster FDDI interface (100 Mbits/sec) and nodes 17–32 are connected through IBM's ALLNODE prototype switch (capable of a peak throughput of 512 Mbits/sec). The ALLNODE switch is an Omega interconnection network capable of providing multiple contentionless paths between the nodes of the cluster.

Communication among the processors is through *message passing*. We used the popular PVM (Parallel Virtual Machine) message passing library (version 3.2.2) to implement our parallel program. We partitioned the domain along the  $z$  direction only. Our partitioning scheme was static since there was no explicit need to keep long vectors given the architecture of the RS6000. The LACE test bed is upgraded periodically. Our results are for RS6000/560 processors. We will report results with upgraded hardware and software in subsequent studies.

### 4.2 Shared Memory Architecture

We used the Cray YMP/8, which has eight vector processors, for this study. The Cray YMP/8 has a peak rating of approximately 2.7 GigaFLOPS. It offers a *single address space* and the communication between processes executing on different processors is through shared variables. Using DOALL directives, we partitioned the domain along the orthogonal direction of the sweep to keep the vectors lengths large and to avoid non-stride access for most of the variables.

### 4.3 Distributed Memory Multiprocessor

In this study, we also used the IBM SP1 which is a distributed memory, message passing multiprocessor. We would like to point out that our test bed is the machine at NASA Lewis in which each processor is a RS6000/370. This system has the original SP1 processors and software upgrades for SP2. The nodes of the SP1 are interconnected through a variation of the Omega network. This network permits multiple contentionless paths between nodes like the ALLNODE switch. We used the MPL library for parallel implementation of our numerical model on the SP1.

## 5. Results

### 5.1 Analytical Model

We will first provide a model to estimate the computation and communication costs. Note that in one time step there is one corrector and one predictor step in each of the coordinate directions. Each of the predictor and corrector steps has a forward and a backward sweep. Let us examine the communication cost for two dimensional computations with  $P_z$  partitions in the  $z$  direction. We divide the cost of data communications across partition boundaries into two parts,  $C_s$  and  $C_f$ .  $C_s$  is associated with the communication needed to compute stresses or heat fluxes at the boundary points. This will be zero for the Euler calculations.  $C_f$  is associated with the transfer of fluxes at the domain boundaries, as required by the forward and the backward sweeps. This would be same for both the Euler and the Navier-Stokes calculations. For this partitioning, we get

$$C_{s,z} = 2n_s[N_r t_t + t_s]$$

$$C_{f,z} = n_v[S_c N_r t_t + t_s]$$

$$C_{s,r} = 2(n_s - 1)[N_r t_t + t_s]$$

$$C_{f,r} = 0$$

where  $n_s$  is the number of neighboring variables needed for stress and heat flux computations. Subscripts  $z$  and  $r$  in the above set of equations refer the direction of the sweep. While velocities at neighboring points in both directions are needed for stress computations, temperatures in only the sweep direction are needed for heat flux computations. Thus, for two dimensional computations,  $n_s$  is three in the sweep direction and two in the other direction. Let  $N_r$  be the number of grid points in the radial direction,  $t_s$  and  $t_t$  be the startup and vector transfer rates (time per element) respectively,  $n_v$  be the number of conservation variables (4 in two dimensions), and  $S_c$  be the width of the stencil measured from the point where variables are updated. This is one for the

standard (second order) MacCormack scheme and 2 for the fourth order extension.  $S_c$  would increase for higher order schemes. The coefficient two in above equations for  $C_s$  comes from the fact that data exchanges in both directions across the partition boundary are needed for stress computations. On the contrary, since the algorithm uses one sided differences of fluxes, the communications of fluxes are only one direction at any time. The total communication times for one time step per processor is then

$$T_{cm} = 2[(4n_s - 2)(N_r t_t + t_s) + n_v(S_c N_r t_t + t_s)]$$

This is expected to rise as the surface area in the partitions increase. It is easily seen that the total computation time per processor is

$$T_{cp} = \frac{N_z}{P_z} N_r T_c$$

where  $T_c$  is the computation time per grid point and  $N_z$  is the total number of grid points in the  $z$  direction. It should be noted that as the size of the problem changes,  $T_c$  could also change based on the architecture of the machine. As the size of the problem increases, there will be an increased amount of cache transfers and misses. This would increase the overhead associated with the computation. The total time spent per processor is then

$$T < T_{cm} + T_{cp}$$

The inequality is a result of some overlapped computations and communications. As the number of processor ( $P_z$ ) increases,  $T_{cm}$  will become significant compared to  $T_{cp}$ .

### 5.2 Numerical results

We consider a jet with the mean inflow profile

$$\bar{U}_r = U_\infty + (U_c - U_\infty)g_r$$

$$\bar{T}_r = T_c + (T_\infty - T_c)g_r + \frac{\gamma - 1}{2} M_c^2 (1 - g_r)g_r$$

$$g_r = \frac{1}{2} \left[ 1 + \tanh\left(\frac{\frac{1}{r} - r}{4\theta}\right) \right]$$

where  $\theta$  is the momentum thickness. The subscripts  $c$  and  $\infty$  refer to the centerline and free stream values respectively. At inflow, we assume the radial velocity is zero and the static pressure is constant. The standard size of our computational domain is 50 radii in the axial direction and 5 radii in the radial direction. We excite the inflow profile at location  $r$  and time  $t$  as

$$U(r, t) = \bar{U}(r) + \epsilon Re(\hat{U} e^{i\pi S_c t})$$

$$\begin{aligned}
P(r, t) &= \bar{P}(r) + \epsilon Re(\hat{P}e^{i\pi S_t t}) \\
\rho(r, t) &= \bar{\rho}(r) + \epsilon Re(\hat{\rho}e^{i\pi S_t t}) \\
V(r, t) &= \epsilon Re(\hat{V}e^{i\pi S_t t})
\end{aligned}$$

$\hat{U}$ ,  $\hat{V}$ ,  $\hat{\rho}$  and  $\hat{P}$  are the eigenfunctions of the linearized equations with the same mean flow profile,  $\epsilon$  is the excitation level and  $S_t$  is the Strouhal number.

We consider a case with  $\frac{U_\infty}{U_c} = \frac{1}{4}$ ,  $\frac{T_\infty}{T_c} = \frac{1}{2}$ , momentum thickness,  $\theta = \frac{1}{8}$  and Strouhal number,  $S_t = \frac{1}{8}$ . The jet center Mach number is 1.5 while the Reynolds number based on the jet diameter is 1.2 million. Our present mean flow and eigen function profiles are same as those in Scott et al.<sup>12</sup> In Figure 1 we show a contour plot of axial momentum from the solution of the Navier Stokes equations. A grid of size 100x250 was used in this computation. This result was obtained after about 16,000 time steps. For all other results in this paper, we simulated the same physical problem on the same grid, but ran for 5000 time steps.

We optimized our code on the LACE to reduce computation time. In the process of optimization, we developed several versions of our code. We will refer the original version of the code as version-1. In version-2, we replaced exponentiations in the code by multiplications (i.e.  $a^2$  is replaced by a times a, etc.). Then we rearranged dimensions of the arrays to ensure *stride-1* access whenever possible. We call the modified code version-3. We also reduced number of divisions and used a single COMMON block instead of many COMMON blocks. We will refer these versions as version-4 and version-5 respectively. In Figure 2 we compare the execution times of different versions of our numerical models on a single processor. It can be easily noted that *Stride-1* access was the single most important optimization and it reduced the computation time considerably. Our partitioning scheme resulted in a good load balance among the processors. In Figure 3, we show the computation times in different processors for our solutions of the Navier-Stokes equations on sixteen processors of the IBM SP1. Results shown in this figure include the computation time and the execution times for the MPL library calls, but exclude any communication wait or transfer time. To study communication characteristics in distributed computing, we developed two additional versions of our code. In version-6 some computations are overlapped with communications. And finally, we divided interprocessor messages in shorter packets to avoid bursty traffic. This is our version-7. In Figure 4 we compare execution times on multiple processors connected by different networks. It is noted that as the number of processor increases, performance of the cluster degrades for the Ethernet due to traffic congestions. For large numbers of processors, in fact execution time increases as the number of processors increases. The

break even point for the Ethernet was about 8-10 processors. Both the ALLNODE switch and the FDDI network show sublinear speedups. We observe that the high speed of FDDI is balanced by slower multiple contention free links in the ALLNODE switch. In Figure 5, we analyze the communication characteristics of different networks. The communication time we report is really the sum of the actual communication time, which is not overlapped with the computation time, and the waiting time by a processor for messages. Communication times for the ALLNODE and the FDDI did not change very much for the number of processors used in this study. On the other hand, Ethernet shows a significant increase of communication times with a large number of processors. For the Navier-Stokes computations with the ALLNODE switch with 16 processors, communication time is comparable to the computation and PVM setup time (PVM set up time includes execution times for all PVM library calls, including packing and unpacking. This time does not include any communication wait and transfer time). Relative communication times are lower for the Euler computations. In Figure 6, we show similar results for the standard (second order) MacCormack scheme. Since the stencil size is now three points wide in any particular direction, instead of five points for the fourth order extension, there is less communication. As a result, Ethernet performed relatively better. However, we note that communication times are comparable to the computation and PVM setup time for more than eight processors. We should point out that all our results in the study except those in Figure 6, were obtained with using the fourth order MacCormack scheme.

In version-6, we tried to optimize our numerical model further by doing some computations overlapped with interprocessor communications. Such an approach should reduce communication times. However, this modification also increases the complexity of the numerical model and overhead associated with loop setups. Our results with overlapped computation and communication along with the results with version-7 are shown in Figure 7. We observe that overlapped computation and communication did not improve the performance of our numerical model. With shorter packet size (in version-7), the Ethernet performed better, while we notice degradation for simulations using the ALLNODE switch. The apparent ineffectiveness of the optimizations in versions 6 and 7 is likely due to the nature of communication requirements of our numerical model, i.e., our algorithm. We used our numerical model of version 4 to obtain execution times on the IBM-SP1. As indicated earlier, we used the MPL library for our parallel implementations. Our results are shown in Figure 8. For 16 processors, the speed up was about 13 for the Euler and 14 for the

Navier-Stokes computations. We would like to point out that processors in the SP1 are not exactly same as those in the LACE cluster. As shown in this figure, communication times were negligible compared to execution times. Because of the fast communication, we expect to see significant speedups with additional processors.

On the Cray-YMP, initially we ran the model with only the Autotasking option on and obtained speedups of slightly above one. We then identified the computationally intensive portions of the code using a profiler, performed subroutine inlining, and partitioned data structures explicitly into shared and private variables. With these modifications, using eight processors, we obtained speedups of more than 6 for both the Euler and the Navier Stokes computations. Our results on the Cray YMP are shown in Figure 9.

We compare results of three computing platforms in Figure 10.1 and 10.2. For the LACE cluster, we show results with the ALLNODE switch. Execution times in both the LACE cluster and the IBM SP1 are likely to go down with additional processors. Because of the fast network, speed ups for the IBM SP1 are likely to be higher. The LACE cluster with 16 processors is about one and a half to two times slower than a single processor Cray YMP. The performance of the IBM SP1 appear to be similar. Both the LACE cluster and the IBM SP1 are going through upgrades and performances are going to improve. Parallel computers, specially the network of workstations are becoming viable and cost effective alternatives to traditional supercomputing. For cluster computing, one should try to use a fast network. Otherwise the communication times can become significant and in fact the execution time can rise as processors are added beyond a certain number. In our tests with large number of processors, the ALLNODE performed well and the FDDI was also comparable. On the other hand, the Ethernet performed poorly for such cases. We plan to follow up our present work with studies of other parallel computing platforms including the Cray T3D and examinations of other numerical models.

### Acknowledgements

The centerline formulation for three dimensional calculations presented in this paper was implemented by the first author following suggestions of Prof. Eli Turkel. We would like to thank Mr. Dale Hubler for his assistance in using the LACE and the IBM-SP1. We are grateful to Dr. Lou Povinelli for his support for this research in the ICOMP program.

### 6. References

- <sup>1</sup>Lighthill, M. J., (1952), "On Sound Generated Aerodynamically, Part I, General Theory", *Proc. Roy. Soc. London*, Vol 211, pp 564-587.
- <sup>2</sup>Hayder, M. E., Flannery, W. S., Littman, M. G., Nosenchuck, D. M. and Orszag, S. A., (1988), "Large Scale Turbulence Simulations on the Navier-Stokes Computer", *Computers and Structures*, Vol 30, No. 1/2, pp 357-364.
- <sup>3</sup>Landsberg, A. M., Young, T. R. and Boris, J. P., (1994), "An Efficient, Parallel method for Solving Flows in Complex Three Dimensional Geometries", AIAA paper 94-0413.
- <sup>4</sup>Morano, E. and Mavriplis, D., (1994), "Implementation of a Parallel Unstructured Euler Solver on the CM-5", AIAA paper 94-0755.
- <sup>5</sup>Venkatakrishnan, V., (1994), "Parallel Implicit Unstructured Grid Euler Solvers", AIAA paper 94-0759.
- <sup>6</sup>Horowitz, J. G., (1993), "Lewis Advanced Cluster Environment", Proceeding of the Distributed Computing for Aerosciences Applications Workshop, NASA Ames Research Center, October 1993.
- <sup>7</sup>Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V., (1993), "PVM 3 User's Guide and Reference Manual", ORNL/TM-12187, Oak Ridge National Laboratory, Oak Ridge, TN.
- <sup>8</sup>Gottlieb, D. and Turkel, E. (1976), "Dissipative Two-Four Methods for Time Dependent Problems", *Math. Comp.*, Vol 30, pp 703-723.
- <sup>9</sup>Hayder, M. E. and Turkel, E., (1993), "High Order Accurate Solutions of Viscous Problems", *AIAA paper* 93-3074, NASA TM 106267.
- <sup>10</sup>Hayder, M. E. Turkel, E. and Mankbadi, R. R., (1993), "Numerical Simulations of a High Mach Number Jet Flow", *AIAA paper* 93-0653, NASA TM 105985.
- <sup>11</sup>Mankbadi, R. R., Hayder, M. E. and Povinelli, L. A., (1994), "The Structure of Supersonic Jet Flow and Its Radiated Sound", *AIAA Journal*, Vol 32, No. 5, pp 897-906.
- <sup>12</sup>Scott, J. N., Mankbadi, R. R., Hayder, M. E. and Hariharan S. I. (1993), "Outflow Boundary Conditions for the Computational Analysis of Jet Noise", *AIAA paper* 93-4366.



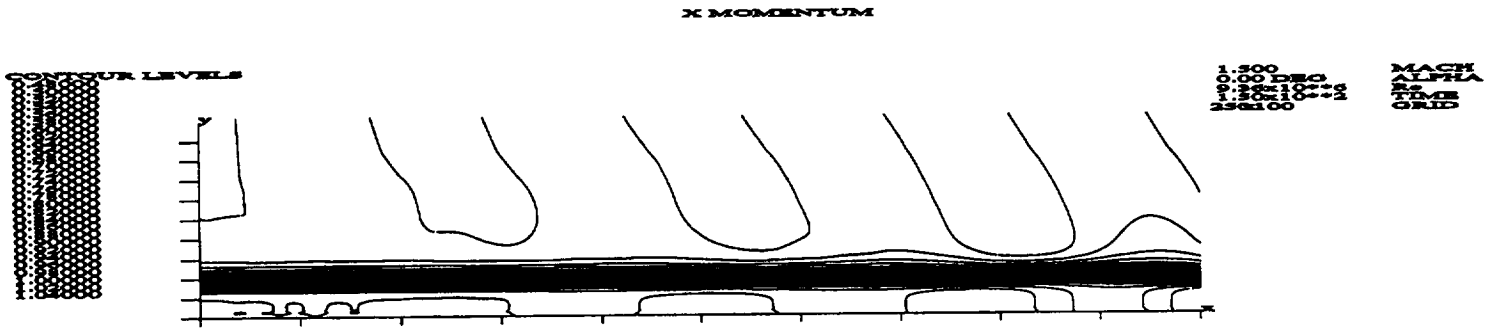


Figure 1: Axial momentum in an excited axisymmetric jet

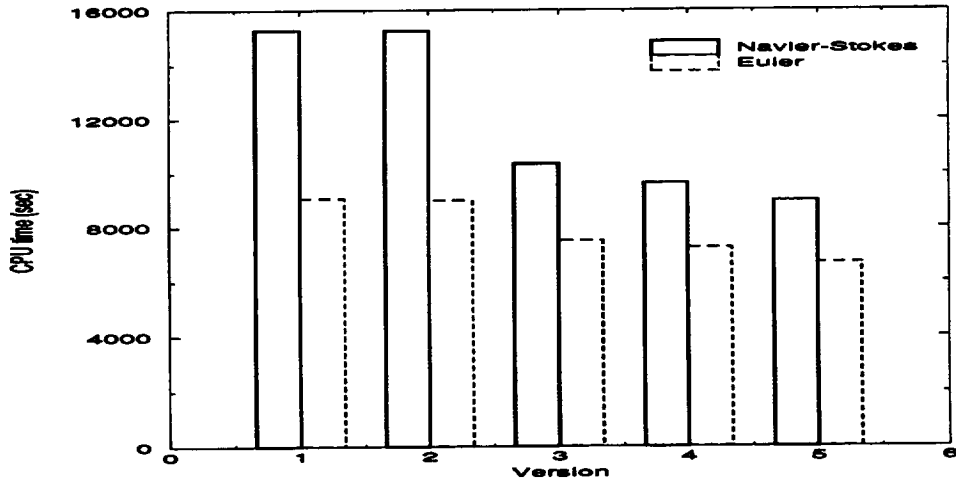


Figure 2: Execution time on a single processor RS6000/560

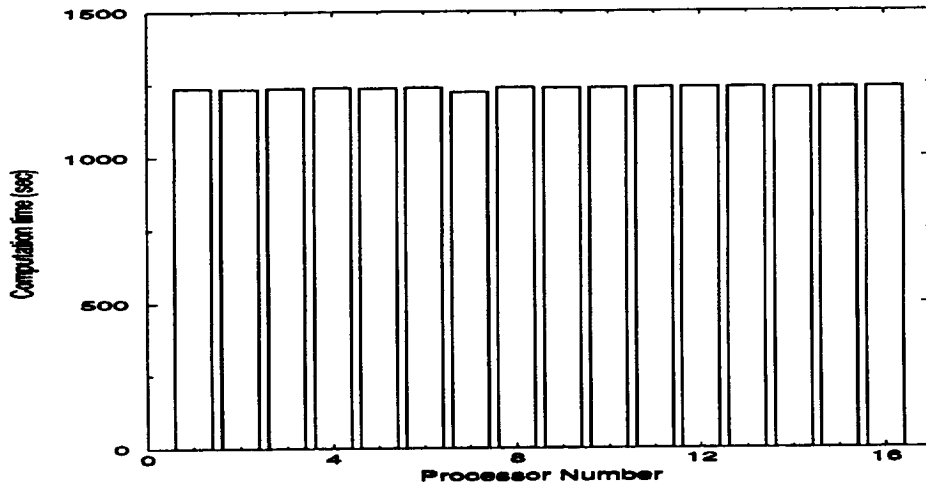


Figure 3: Computation times on sixteen RS6000/370 processors IBM SP-2

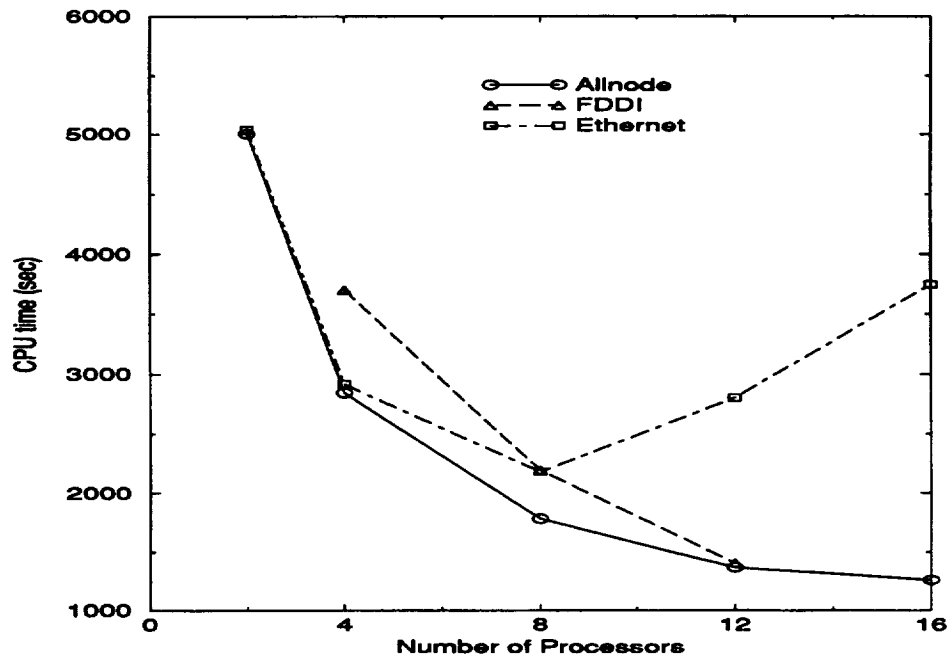


Figure 4.1: Navier Stokes execution time on the LACE

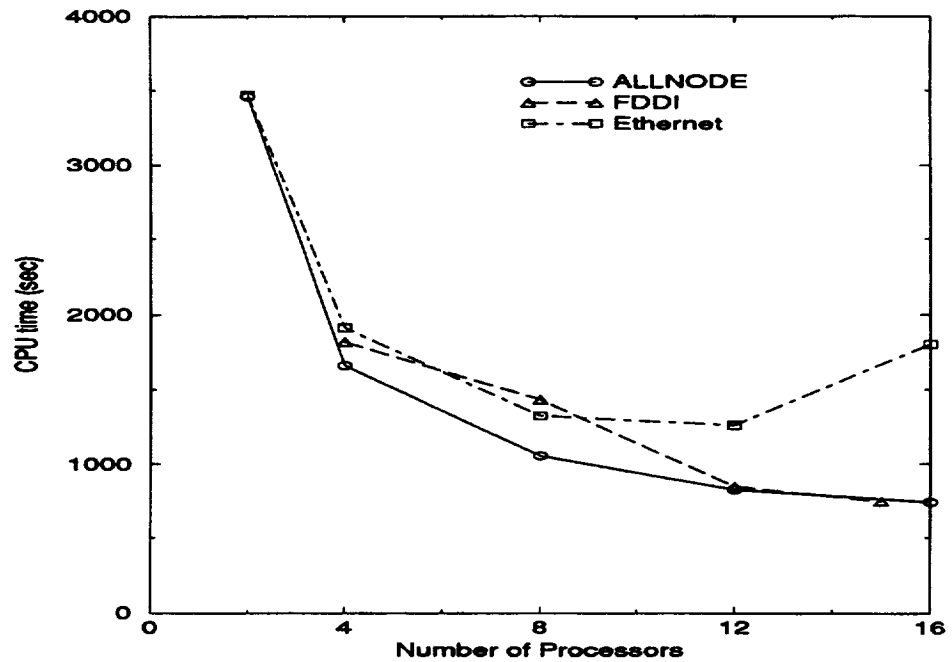


Figure 4.2: Euler execution time on the LACE

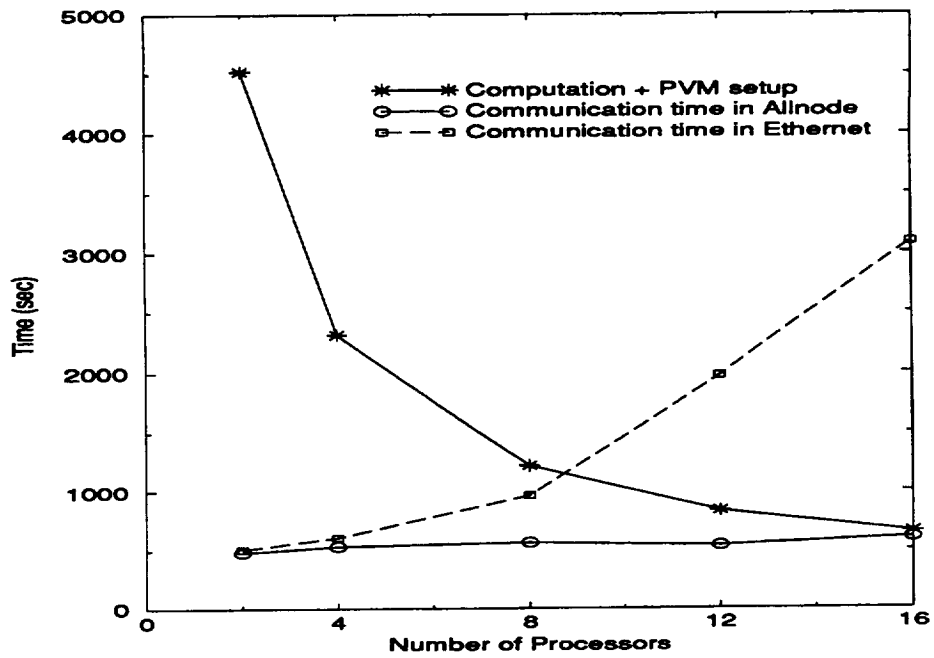


Figure 5.1: Navier Stokes computations on the LACE

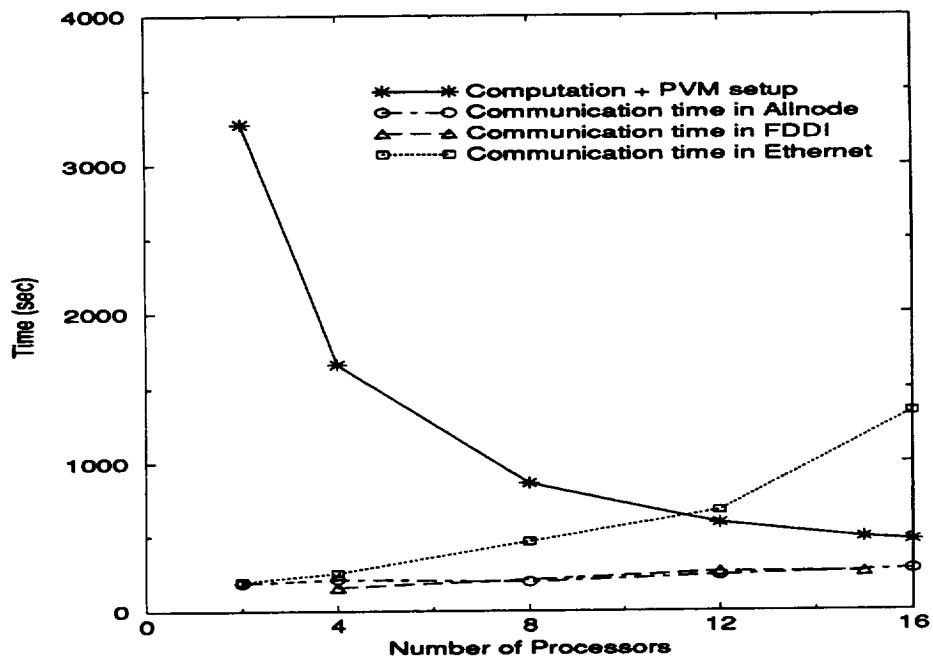


Figure 5.2: Euler computations on the LACE

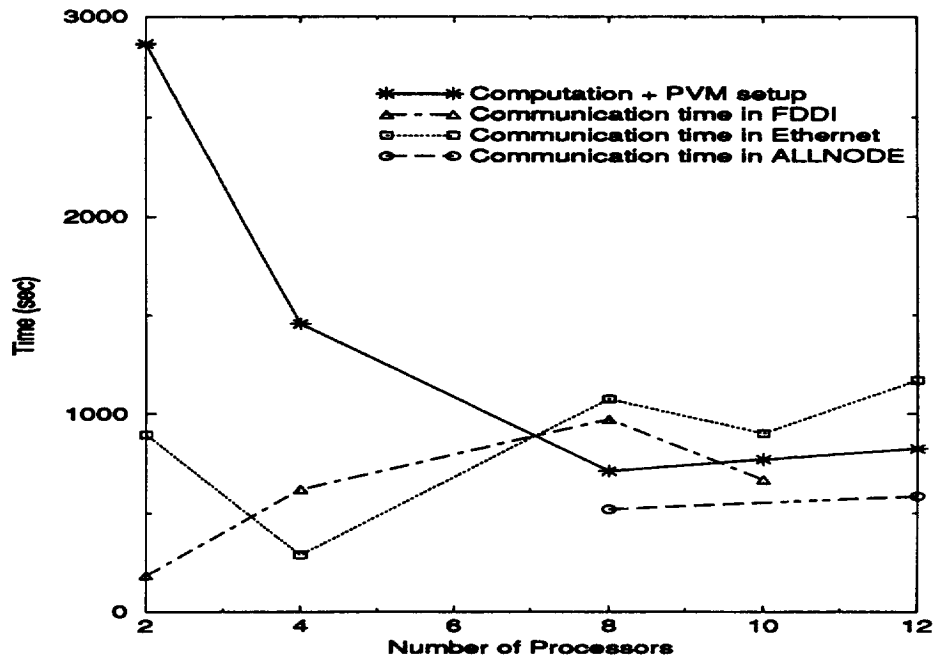


Figure 6.1: Navier Stokes computations by the standard MacCormack scheme

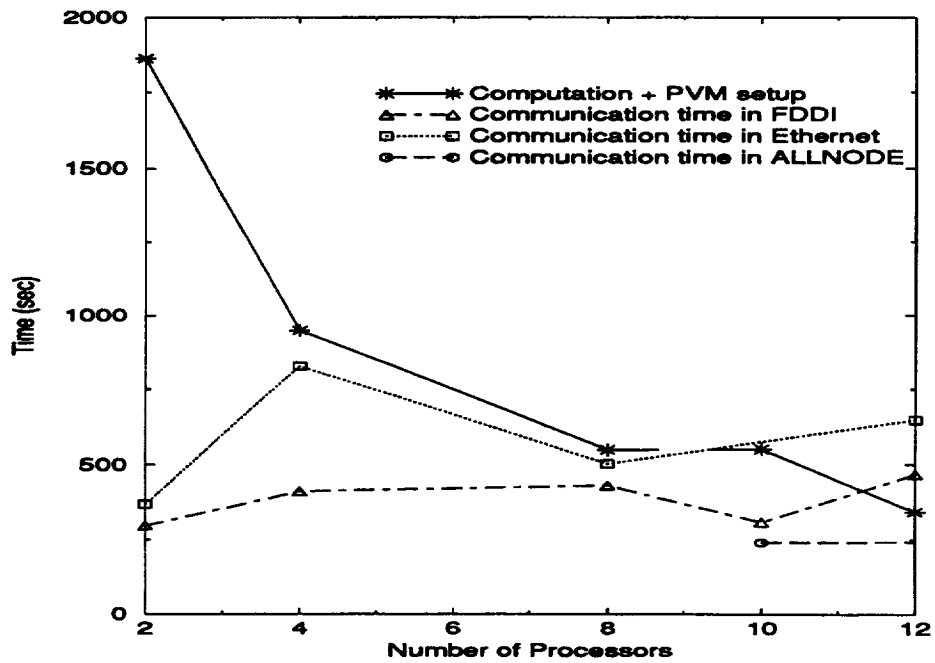


Figure 6.2: Euler computations by the standard MacCormack scheme

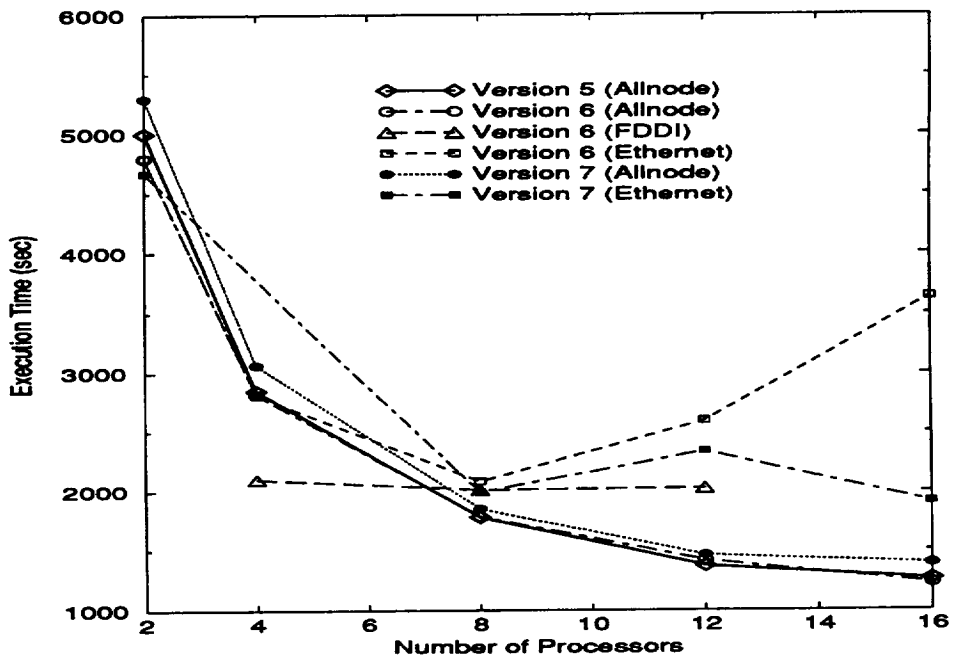


Figure 7.1: Communication optimisations for the Navier Stokes computations

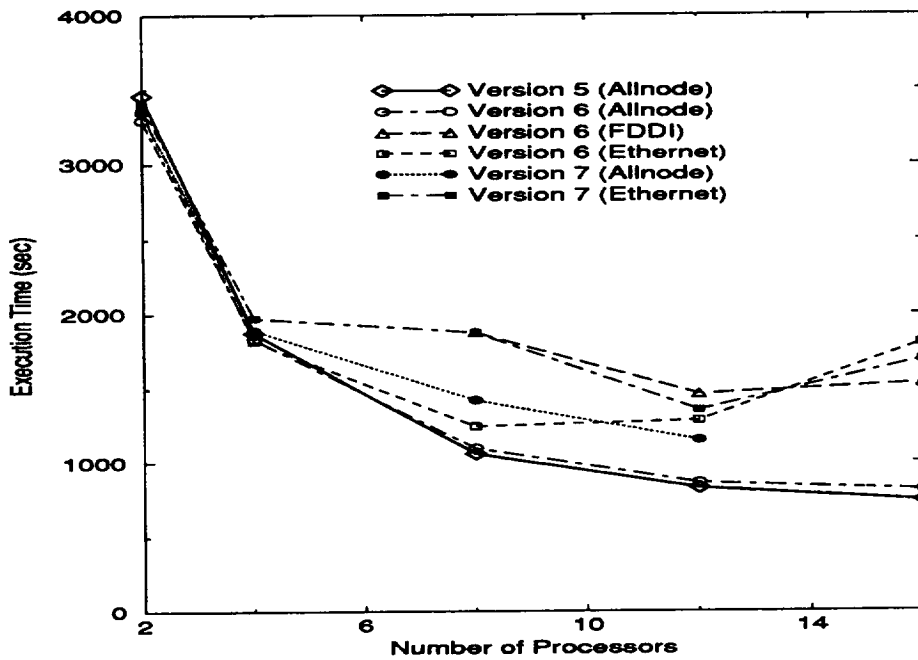


Figure 7.2: Communication optimisations for the Euler computations

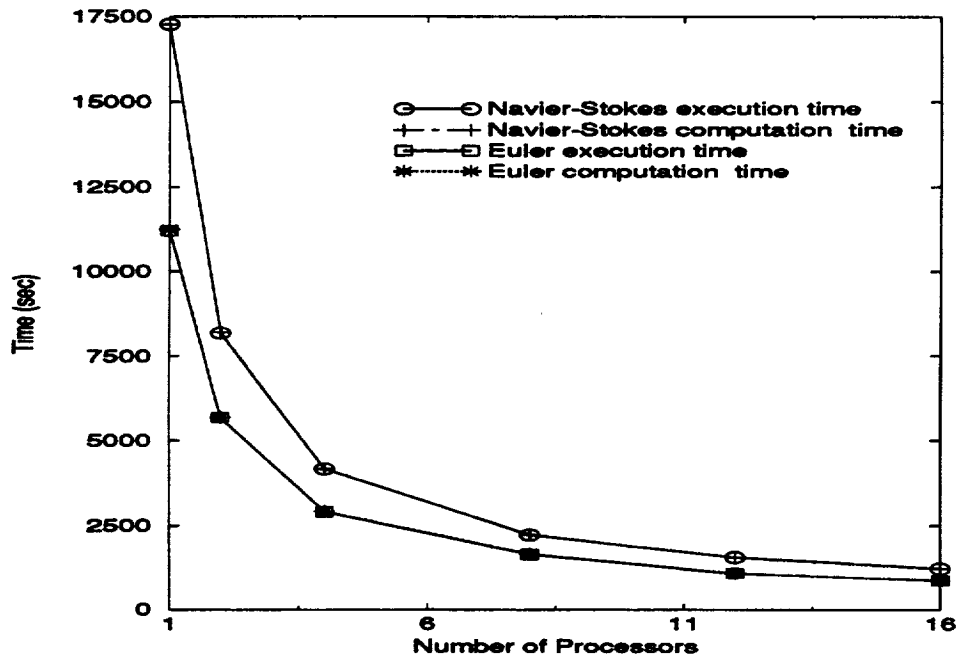


Figure 8: Execution times on the IBM-SP2 (with RS6000/370 processors)

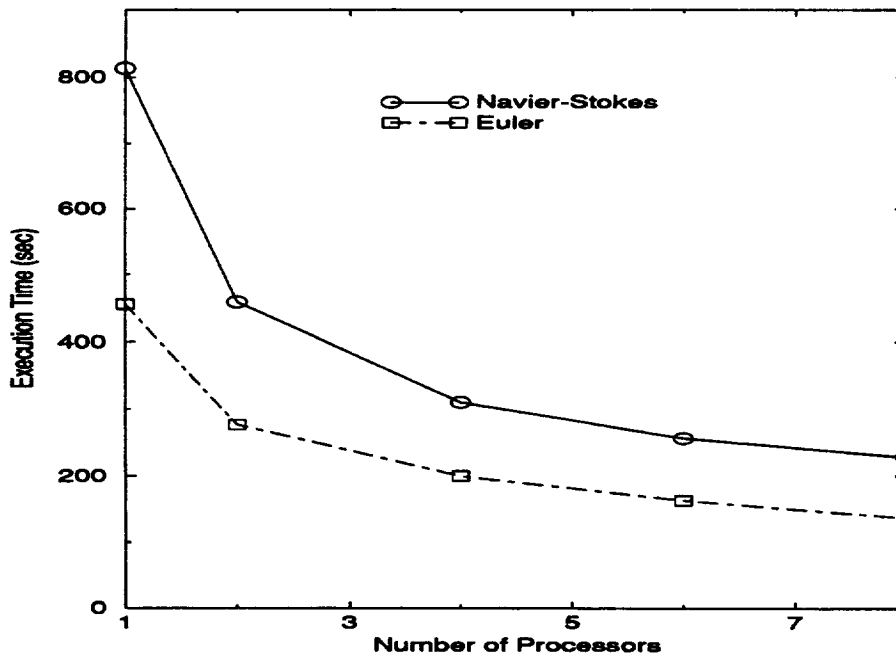


Figure 9: Execution times on the Cray YMP

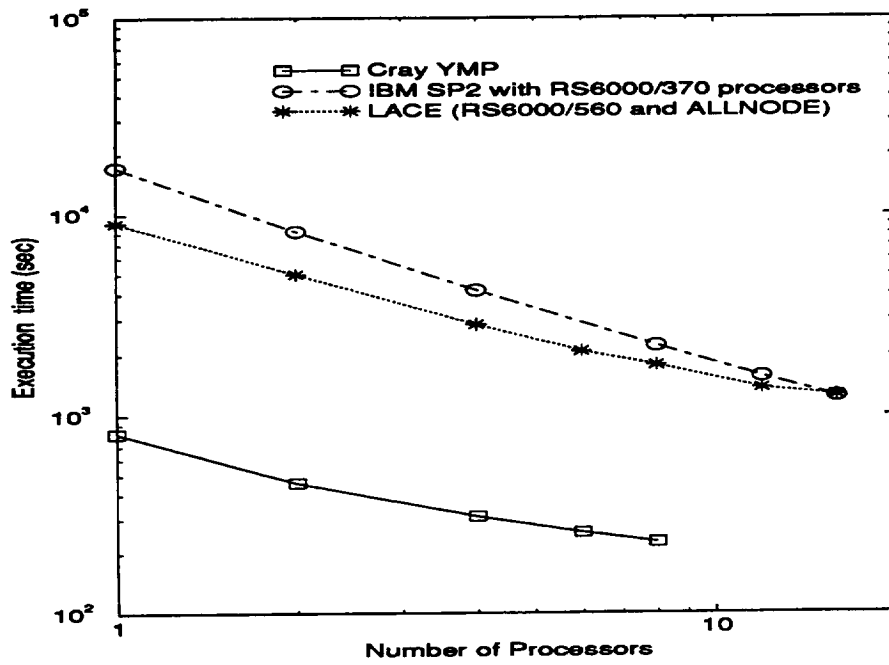


Figure 10.1: Navier-Stokes execution times

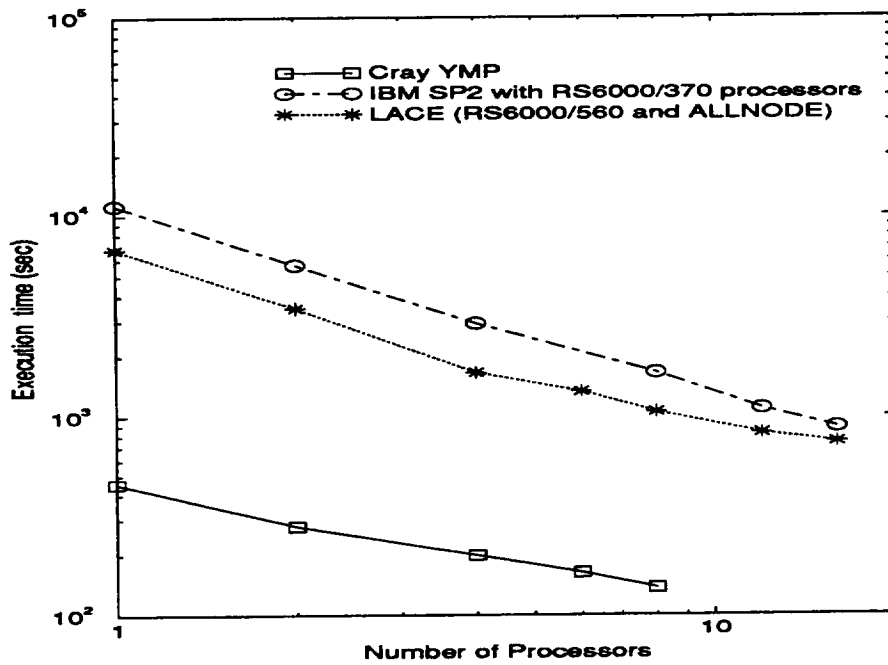


Figure 10.2: Euler execution times

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 1995	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Parallel Navier-Stokes Computations on Shared and Distributed Memory Architectures		5. FUNDING NUMBERS  WU-505-90-5K	
6. AUTHOR(S)  M. Ehtesham Hayder, D.N. Jayasimha, and Sasi Kumar Pillay		8. PERFORMING ORGANIZATION REPORT NUMBER  E-9369	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191		10. SPONSORING/MONITORING AGENCY REPORT NUMBER  NASA TM-106823 ICOMP-94-32 AIAA-95-0577	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Washington, D.C. 20546-0001		11. SUPPLEMENTARY NOTES Prepared for the 33rd Aerospace Sciences Meeting and Exhibit sponsored by the American Institute of Aeronautics and Astronautics, Reno, Nevada, January 9-12, 1995. M. Ehtesham Hayder, Institute for Computational Mechanics in Propulsion, Lewis Research Center (work funded under Cooperative Agreement NCC3-233); D.N. Jayasimha, Ohio State University, Columbus, Ohio 43210; and Sasi Kumar Pillay, NASA Lewis Research Center, ICOMP Program Director, Louis A. Povinelli, organization code 2600, (216) 433-5818.	
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified - Unlimited Subject Categories 34, 61, and 64  This publication is available from the NASA Center for Aerospace Information, (301) 621-0390.		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  We study a high order finite difference scheme to solve the time accurate flow field of a jet using the compressible Navier-Stokes equations. As part of our ongoing efforts, we have implemented our numerical model on three parallel computing platforms to study the computational, communication, and scalability characteristics. The platforms chosen for this study are a cluster of workstations connected through fast networks (the LACE experimental testbed at NASA Lewis), a shared memory multiprocessor (the Cray YMP), and a distributed memory multiprocessor (the IBM SP1). Our focus in this study is on the LACE testbed. We present some results for the Cray YMP and the IBM SP1 mainly for comparison purposes. On the LACE testbed, we study: (a) the communication characteristics of Ethernet, FDDI and the ALLNODE networks and (b) the overheads induced by the PVM message passing library used for parallelizing the application. We demonstrate that clustering of workstations is effective and has the potential to be computationally competitive with supercomputers at a fraction of the cost.			
14. SUBJECT TERMS  Navier-Stokes computations; Parallel computing; Cluster computing		15. NUMBER OF PAGES 15	
17. SECURITY CLASSIFICATION OF REPORT Unclassified		16. PRICE CODE A03	
18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	