

Utilization of Parallel Processing in Solving the Inviscid Form of the Average-Passage Equation System for Multistage Turbomachinery

Richard A. Mulac and Mark L. Celestina
Sverdrup Technology, Inc.
(Lewis Research Center Group)
NASA Lewis Research Center
Cleveland, Ohio

John J. Adamczyk
NASA Lewis Research Center
Cleveland, Ohio

Kent P. Misegades and Jef M. Dawson
Cray Research, Inc.
Mendota Heights, Minnesota

Prepared for the
8th Computational Fluid Dynamics Conference
sponsored by the American Institute of Aeronautics and Astronautics
Honolulu, Hawaii, June 9-11, 1987



N88-21160

Unclas
0139989

(NASA-TM-89845) UTILIZATION OF PARALLEL
PROCESSING IN SOLVING THE INVISCID FORM OF
THE AVERAGE-PASSAGE EQUATION SYSTEM FOR
MULTISTAGE TURBOMACHINERY (NASA) 13 p
CSCL 21E G3/07

Utilization of Parallel Processing in Solving the Inviscid Form of the Average-Passage Equation System for Multistage Turbomachinery

Richard A. Mulac and Mark L. Celestina
Sverdrup Technology, Inc.
(Lewis Research Center Group)
NASA Lewis Research Center
Cleveland, Ohio

John J. Adamczyk
NASA Lewis Research Center
Cleveland, Ohio

Kent P. Misegades and Jef M. Dawson
Cray Research, Inc.
Mendota Heights, Minnesota



National Aeronautics and
Space Administration

Lewis Research Center
Cleveland, Ohio 44135

Summary

A procedure is outlined which utilizes parallel processing to solve the inviscid form of the average-passage equation system for multistage turbomachinery along with a description of its implementation in a FORTRAN computer code, MSTAGE. A scheme to reduce the central memory requirements of the program is also detailed. Both the multitasking and I/O routines referred to in this paper are specific to the Cray X-MP line of computers and its associated SSD (Solid-State Disk). Results are presented for a simulation of a two-stage rocket engine fuel pump turbine.

Introduction

The desire of turbomachinery aerodynamicists to increase the performance of multistage machinery significantly beyond the limits of current machinery is forcing them to examine novel three-dimensional design concepts. To assess these concepts, flow models capable of resolving the three-dimensional flow field within a blade-row passage embedded in a multistage environment are needed. The choice of an appropriate flow model is by no means obvious, for a simulation based on the flow model must be computationally compatible with the speed and memory available on today's supercomputers. At the same time it must provide a greater degree of comprehension of the flow field within a blade-row passage than the existing quasi-three-dimensional flow models. A proposed flow model which satisfies these constraints was detailed in reference 1. This flow model, referred to as the "average-passage" flow model, describes the time-averaged flow field within a typical passage of a blade row. It may be derived from first principles through a series of averaging operators applied to the Navier-Stokes equations. Each blade row is thus associated with an "average-passage" flow field which, in addition to representing a time-averaged flow field, is also spatially periodic from blade passage to blade passage. Since nearly all axial flow turbomachinery is designed conceptually around such an averaged flow state, the average-passage model allows us to achieve three-dimensional resolution of a multistage configuration within the computing resources available today.

Based on a closure model for the inviscid form of the average-passage equation system (ref. 2), Celestina and Adamczyk developed a computer program to solve the flow

field associated with a single stage configuration. This code was successful in calculating the inviscid flow through a high-speed counter-rotating propeller and a fan stage (refs. 2 and 3). Their solution strategy consisted of a nested iterative procedure. The inner loop evaluated the three-dimensional flow variables for a specific blade row based on a distribution of body forces, energy sources, energy correlations, and velocity correlations which accounted for the presence of the neighboring blade row. This was performed using a four-stage Runge-Kutta integration procedure (ref. 4), which converged the flow field to within a specified tolerance. Upon convergence of each blade-row simulation, the axisymmetric average of each blade-row's solution was evaluated. The outer loop evaluated the discrepancy between the axisymmetric solutions. According to the closure model, the final axisymmetric solution associated with each blade row should converge to a single axisymmetric solution within a limit set by the computational mesh size. If the difference between the axisymmetric solutions was greater than a set tolerance, then the body forces, energy sources, energy correlations, and velocity correlations were recalculated for each blade row based on their respective estimated average-passage solution. The inner loop was then repeated by using the updated information. When the outer loop convergence criteria was met, the numerical simulation was terminated.

The objective of this paper is to report on the extension of the previous single-stage work to multistage configurations and the application of multitasking to take advantage of multiple processor computer architectures. The present work also includes an improved treatment of the boundary conditions, the use of residual averaging to accelerate convergence, and large high-speed mass storage to ease central memory requirements.

Governing Equations

The three-dimensional average-passage equation system for a multiple blade row configuration can be written in cylindrical (z, r, θ) coordinates as

$$\int (\lambda \mathbf{u}_t) dVol + L(\lambda \mathbf{u}) + \int \lambda S dVol = \int \lambda K dVol \quad (1)$$

The vector \mathbf{u} contains the flow variables density, axial, radial, and angular momenta, and total internal energy. The variable λ is the neighboring blade-row blockage attributed

to its thickness. The value of this parameter ranges between zero and unity, unity being the value associated with a blade of zero thickness. The operator $L(\lambda \mathbf{u})$ balances the mass, axial, radial, and angular momenta, and energy through a control volume. The source term $\{\lambda K dVol$ is due to the cylindrical coordinate system, while $\{\lambda S dVol$ contains the body forces, energy sources, and energy and velocity correlations associated with the neighboring blade rows.

To advance the equations in time, a four-stage Runge-Kutta scheme is used. The scheme employed has been patterned after the work of Jameson, et al (ref. 4). Given information at time level n , the steps to advance to the next level $n + 1$ are

$$\left. \begin{aligned} \mathbf{u}^\alpha &= \mathbf{u}^n - \alpha \Delta t [L(\mathbf{u}^n) + D(\mathbf{u}^n)] \\ \mathbf{u}^\beta &= \mathbf{u}^n - \beta \Delta t [L(\mathbf{u}^\alpha) + D(\mathbf{u}^n)] \\ \mathbf{u}^\gamma &= \mathbf{u}^n - \gamma \Delta t [L(\mathbf{u}^\beta) + D(\mathbf{u}^n)] \\ \mathbf{u}^{n+1} &= \mathbf{u}^n - \Delta t [L(\mathbf{u}^\gamma) + D(\mathbf{u}^n)] \end{aligned} \right\} \quad (2)$$

where $\alpha = 1/4$, $\beta = 1/3$, $\gamma = 1/2$ and $D(\mathbf{u})$ is the dissipation operator. The maximum permissible time-step for this scheme is restricted by the CFL stability limit. To enhance the convergence rate of this scheme, a local time-step is chosen based on the maximum CFL number commensurate with stability. To suppress odd-even point decoupling in the solution, dissipative terms are added to the equations. Jameson (ref. 4) suggested a blend of second and fourth difference smoothing operators. The present work makes use of this operator as adapted to the present equation system. The operator $D(\mathbf{u})$ is decomposed into three spatial operators

$$D(\mathbf{u}) = (D_z + D_r + D_\theta)(\mathbf{u}) \quad (3)$$

such that the dissipation in each direction can be evaluated separately. Further details are given in reference 3.

It is well known that the formulation of the boundary condition for internal flow problems plays a critical role in establishing the convergence rate of the iteration scheme. For the problems of interest in this paper we assume that the absolute flow field entering and exiting the computational domain is subsonic. This implies that four conditions must be specified at the upstream boundary and only one condition at the downstream boundary.

Two models will be proposed for the inlet. The first is for an operating condition in which the relative Mach number is sufficiently high to cause the blade-row passage flow field to become choked. For this case we specify the inlet absolute total temperature distribution. With the total temperature and entropy given, all the remaining total conditions can be evaluated. The mass flow, however, remains free to evolve during the iteration procedure. The second model is only valid for operating conditions in which the passage flow fields are unchoked. Such operating conditions are the rule rather than

the exception for multistage axial flow machinery. For these operating conditions we specify the mass flow per unit area entering the machine and the incoming entropy field. The total conditions are unspecified and evolve during the iteration procedure. This second boundary condition, when applicable, leads to a reduction in the number of iteration cycles required to reach a preassigned convergence level compared to the first model. This difference increases as the inlet Mach number of the flow field is reduced. Based on either model, the axial velocity component and the thermodynamic flow properties at the upstream boundary are recalculated based on a local unsteady one-dimensional flow approximation. The equation associated with this approximation is

$$\frac{\partial C^-}{\partial t} + (v_z - a) \frac{\partial C^-}{\partial z} \quad (4)$$

in which C is the well-known Riemann invariant. It is related to the axial velocity v_z and speed of sound a by the equation

$$v_z - \frac{2a}{\gamma - 1} = C^- \quad (5)$$

The invariant C^- is associated with information coming from the interior of the computational domain. It is recalculated by solving equation (4) in time by using a four-stage Runge-Kutta integration procedure. The axial derivative is approximated by a backward difference operator. This Riemann invariant along with information supplied by the inlet flow models determines the speed of sound and the axial velocity component. The pressure, density, and temperature are updated based on the known value of the incoming entropy. The value of the velocity components parallel to the upstream boundary are assumed known.

At the downstream boundary, simple radial equilibrium is enforced,

$$\frac{\partial p}{\partial r} = \frac{\rho v_r^2}{r} \quad (6)$$

The pressure is specified at the hub boundary and equation (6) is integrated radially toward the shroud by using the trapezoidal rule. The remaining flow variables are extrapolated from the interior. At periodic flow boundaries we require the flow to exhibit a spatial periodicity equal to the pitch of the blade row. Thus, any information required from a cell which lies adjacent to a periodic boundary and outside the computational domain is obtained from a cell which lies adjacent to the opposing periodic boundary but is inside the computational domain.

Since the flux is zero on solid surfaces (hub, shroud, and blade) only the pressure need be known. This can be

determined from the interior by using an adaptation to the present system of equations of a normal pressure gradient condition developed by Rizzi (ref. 5). Further numerical details can be found in references 1 to 3.

The original work of Celestina and Adamczyk (ref. 3) used the simple scheme of local time-stepping to accelerate convergence. In this work we have also included the strategy of residual averaging as suggested by the work of Jameson (ref. 4) to accelerate the iteration process. This procedure permits the use of a local time-step beyond that allowed by the CFL condition. Its effect is to enhance the propagation of error out of the computational domain.

The present solution algorithm requires each blade row to have a unique computational H-mesh describing its respective passage. However, all meshes must share common axial and radial coordinates as shown in figure 1. This eliminates interpolative error when information is passed between the individual flow field calculations. A grid generator satisfying these criteria was employed to create the meshes used by the solution algorithm (ref. 6). The configuration shown in figure 1 is a mesh for a two-stage rocket engine fuel pump turbine. The blade-to-blade view of the mesh at the hub for the first vane, first rotor, second vane, and second rotor is shown in figures 2 to 5, respectively.



Figure 1.—Common axisymmetric mesh.

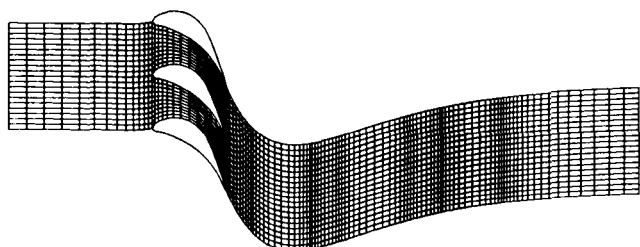


Figure 2.—Blade-to-blade view of first vane mesh.

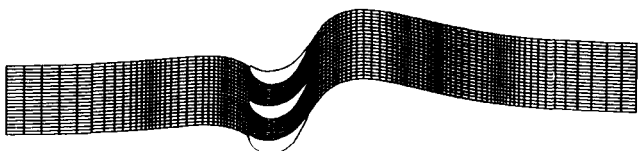


Figure 3.—Blade-to-blade view of first rotor mesh.

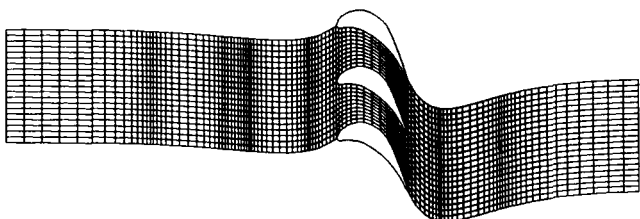


Figure 4.—Blade-to-blade view of second vane mesh.

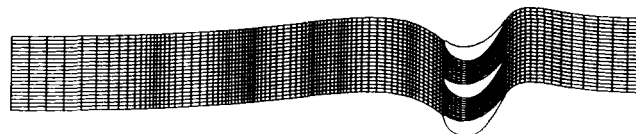


Figure 5.—Blade-to-blade view of second rotor mesh.

Multitasking

The inner loop procedure described previously executes the flow field calculations one blade row at a time. Although the computational time required for a single stage simulation may not be significantly greater than that for an isolated blade row, if one were to attempt a simulation of an N-blade-row configuration, the wall-clock time for such a calculation with a single processor might be impractical. The advent of today's multiprocessor computer alleviates this problem.

When deciding whether a parallel algorithm is suitable for multitasking, one of the determining factors is the level of data dependence the individual tasks have on one another. One primary goal is to have as few variables as possible shared between tasks. The converse of the preceding statement would necessitate variables being "locked" by the first task able to access them. This would result in tasks spending a great deal of time in an idle state waiting for "locked" variables to be released. The inner loop of the solution procedure turns out to be an ideal candidate for parallel processing. Each blade row has its own set of flow variables and a unique computational mesh. The only time data are shared between the individual flow field calculations is immediately before entering the inner loop. Because of this independence, an N-blade-row simulation running on a dedicated N-processor computer should use approximately the same wall-clock time as a single-blade-row simulation since over 97 percent of the computational work is performed in the inner loop.

The parallel algorithm was implemented with Cray multitasking constructs. At the time of this writing, there has been no standard multitasked version of FORTRAN released. The Cray multitasking software contains the FORTRAN callable routine TSKSTART which spins off a task to an available processor and TSKWAIT which returns control to the main program after completion of the task. The program MSTAGE makes use of these routines in three areas. First, after the input parameters are read in by the code, a task is created for each blade row to set up the file containing the mesh, surface area, volume, flow, and body force data which is to be used during the remainder of the program. After this is completed, the two level solution procedure is begun. The inner loop, which calculates the individual flow field solutions, is performed for each blade row as an individual task. This is repeated until the outer loop of the solution procedure has converged to within a certain tolerance. Lastly, output from each of the flow field calculations is performed as concurrent tasks.

Data Storage

Having developed a strategy for reducing the wall-clock time required for an N-blade-row simulation, we find ourselves concerned that such a simulation may extend beyond the memory capacity of today's supercomputers. Since each blade row requires its own set of flow variables and mesh coordinates, one would need memory well above one hundred million words for a serious multistage simulation. One possible solution is to take the flow field and mesh quantities, which were previously kept in three-dimensional arrays, and store them as a series of two-dimensional arrays stacked on one another. In this way only the planes of data which are currently needed by the solution procedure would have to lie in central memory; at other times they could remain out on secondary storage in a random access file.

Because the solution procedure requires that axisymmetric data communicate between the neighboring blade rows, the flow fields are calculated by using H-meshes which share common axial and radial coordinates, but have theta values specific to the blade row being simulated. As the number of blade rows increases, the axial dimension becomes the largest of the three. In order to make use of this long vector length, data is stored on radially indexed planes which run from hub to tip. Storing the data in this manner also permits easy implementation of the periodic boundary condition.

Five files are required by MSTAGE for each blade row. The mesh coordinate file, flow field restart file, body force restart file, random access file, and the printed output file are each assigned one FORTRAN unit number. There are 100 available unit numbers in the current level of Cray FORTRAN, so if 5 are required by each blade row, a maximum of 20 blade rows can be numerically simulated by the code. Data which are stored in the random access file include the current flow field quantities, flow field quantities from the previous iteration, filtered flow field quantities, cell coordinates, volumes, surface areas, local time-step values, body force quantities, and the axisymmetric flow field quantities.

Extensive use of the Cray random access I/O routines READDR and WRITDR are made throughout the code to transfer data planes to and from the random access files on the SSD. These are direct-to-disk routines which deal with unblocked records to eliminate the overhead of blocking and unblocking usually performed by the operating system. To further reduce the overhead of the I/O operations, asynchronous I/O is performed wherever possible. Storage of the data planes in central memory is accomplished through the Cray multi-tasking feature TASK COMMON. This allows each task to create common blocks which are local to the task. Currently the algorithm requires a maximum of three data planes for the mesh coordinates, three data planes for the cell surface areas, and four data planes for the flow field variables, volumes, and local time-steps at each field point. To avoid

having to copy information down the data plane stack as the algorithm sweeps through the radial direction, the POINTER feature of Cray FORTRAN is used. This allows the memory location assigned to a data plane to be changed dynamically.

Results

The results presented in this section are for the rocket engine fuel pump turbine noted previously. This turbine has two stages with 41, 63, 39, and 59 blades in the first, second, third, and fourth blade rows, respectively. The first and third blade rows are stationary, while the second and fourth rotate on a common shaft. The absolute Mach number of the flow entering and leaving the turbine is 0.15 and 0.25, respectively. The exit to inlet static pressure ratio is 0.5671. These operating conditions were inferred from overall engine performance data. Detailed blade-row performance data are not available partly because of the small size of the turbine and the hostile environment in which it operates. The average-passage model provides a means to assess each blade-row's performance including a description of the three-dimensional flow field within each blade-row passage. This information is critical in order to establish any potential turbine design shortcomings which could compromise either the rocket motor performance or its integrity.

The following simulations were made on a Cray X-MP 416 (4 processors, 16 million word memory) with an accompanying 512 million word SSD. The color figures were also produced on the same computer using the graphics package MOVIE.BYU. Each mesh used in the simulations consisted of 110 points in the axial direction, 11 points in the radial direction, and 11 points in the circumferential direction. Each blade surface had 15 points in the axial direction and 11 points in the radial direction. Mesh points were packed at the leading and trailing edges of the blades using a geometric progression formula.

The first simulation assumed a uniform inlet flow field boundary condition. The solution procedure was started by numerically simulating the four blade rows sequentially, while using a uniform flow field as an initial condition. Starting the simulation in this manner insured that the flow entering each blade row was at the proper incidence angle, and thus helped to stabilize the convergence of the solution. Once this was accomplished, the inner loop proceeded as a set of 4 parallel tasks, each performing 750 iterations of its respective blade-row simulation. At the completion of every outer loop iteration, the difference between the axisymmetric components of all blade-row pair combinations was calculated. The magnitude of this difference was measured by constructing an L1 norm for all of the permutations. As noted previously, the model used to estimate the body forces, energy sources and correlations which account for the presence of neighboring

blade rows, limits the magnitude to which these differences can be reduced. This bound is set by the truncation error associated with the artificial dissipation of the iteration procedure. The outer loop convergence history is shown in figure 6. After five iterations of the outer loop, the L1 norm of the residual approaches an asymptotic limit indicating that the simulation is complete. A comparison of the convergence

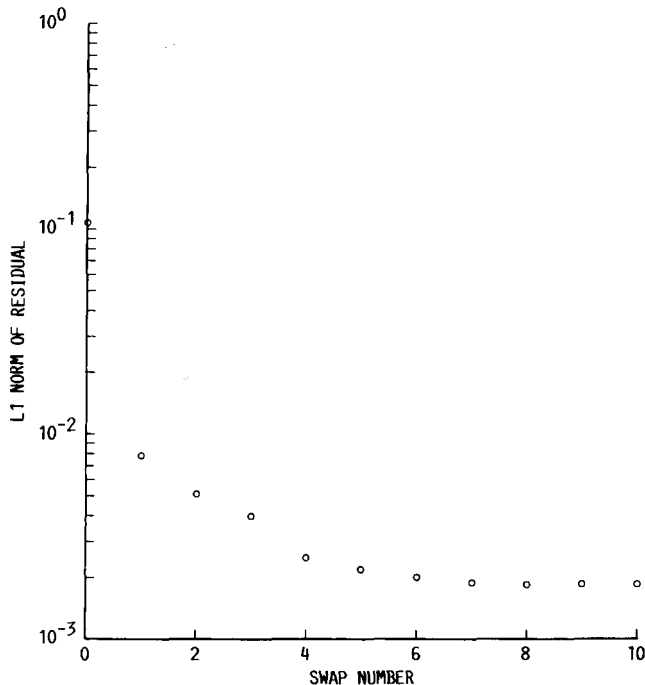


Figure 6.—Outer loop convergence (uniform inlet boundary condition).

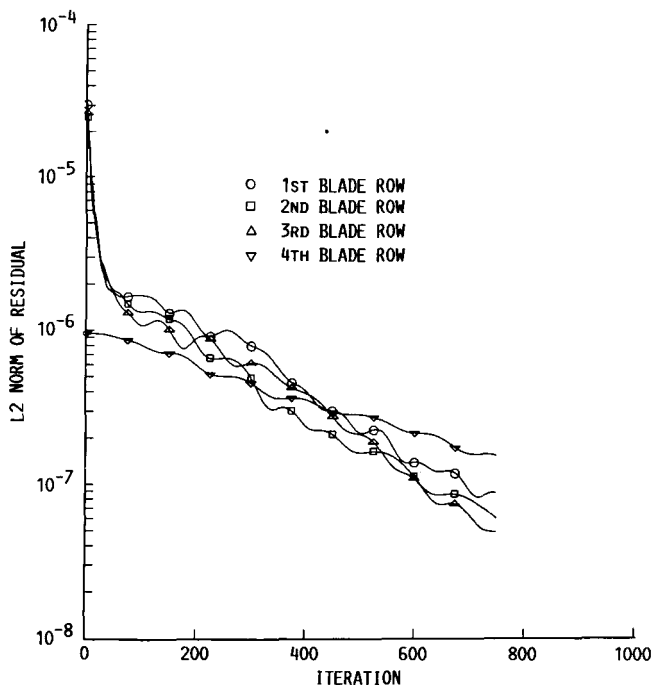


Figure 7.—Inner loop convergence (first swap).

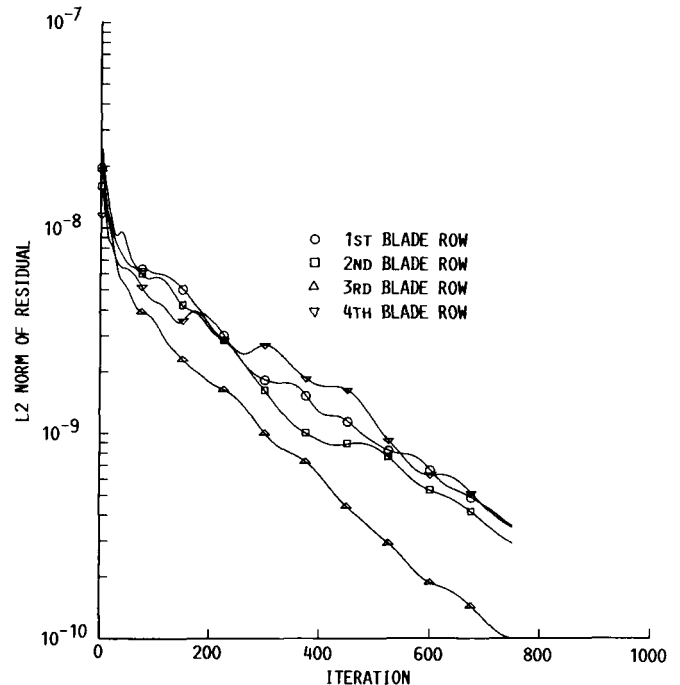


Figure 8.—Inner loop convergence (tenth swap).

history of the inner loop procedure is made in figures 7 and 8. The L2 norms of the individual solution residuals have dropped more than two orders of magnitude by the 10th iteration of the outer loop procedure as compared to the 1st iteration. Figure 9 shows the resulting pressure distribution on each of the four blade rows and the hub surface. The figure is color coded so that red represents regions of high pressure and blue represents regions of low pressure. The pressure distribution on each blade appears to be nearly two-dimensional because variations in the radial direction are slight.

Each iteration of the outer loop procedure required 941 sec with each individual task consuming 235 sec. The total CPU time of the first simulation, consisting of 10 outer loop iterations, was 10 350 sec. In order to analyze the multitasking aspects of the code and deduce algorithm efficiency, an identical run was made with all data stored in central memory to neutralize the effects of I/O to and from the SSD. This was accomplished through the use of a subroutine which mimicked the I/O routines by transferring data between the data planes and a large three-dimensional array which substituted for the SSD. Operating in this mode increased the CPU time of each iteration of the outer loop to 1884 sec and the total simulation CPU time to 20 850 sec because of the pseudo I/O operations. Running in a dedicated mode on the X-MP 416, this in-core simulation required 6.6 million words of main memory. It was completed in 5369 wall-clock sec by using all four processors operating with an efficiency of over 97 percent. Employing the SSD reduced the memory required to 1.7 million words, but increased the wall-clock time for the simulation to 19 482 wall-clock sec; thus the efficiency of processor use was

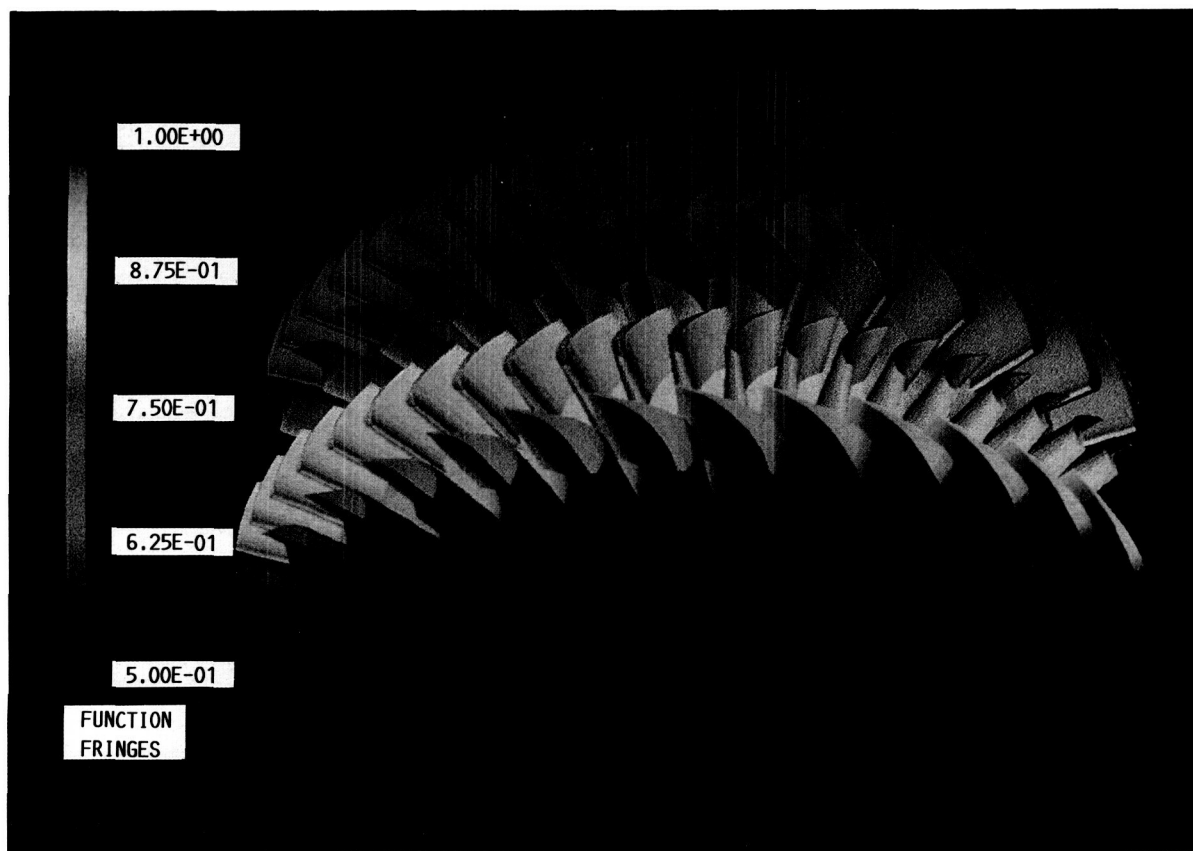


Figure 9.—Two-stage turbine hub and blade surface pressure distribution.

lowered to 14 percent. This was not the fault of the algorithm, but was due to several other factors which included overhead by the operating system during I/O operations, running under a single-threaded operating system, and a limit of four processors and two high-speed channels to the SSD. As future hardware and software products address these limiting factors, the code utilizing high-speed secondary storage should approach the ideal processor efficiency rate. Although the version of the code utilizing the SSD yielded poor processor efficiency on a dedicated machine, there are benefits when running it in a multi-user batch environment. This occurs because of the lower memory required by the program and the ability of other user's tasks to make use of idle processors during I/O operations.

The second simulation run with the code employed a nonuniform inlet boundary condition, consisting of a total temperature profile which was distorted in the radial direction to approximate the exit flow field of a combustor. The solution procedure was started by using the flow field generated by the previous simulation as an initial solution. Once again, as can be seen in figure 10, after 25 iterations of the outer loop procedure the L1 norm of the residual becomes asymptotic

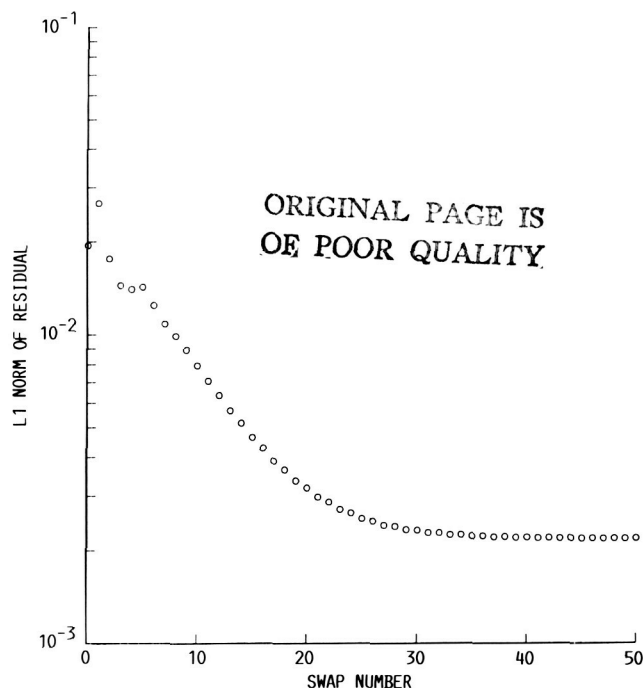


Figure 10.—Outer loop convergence (nonuniform inlet boundary condition).

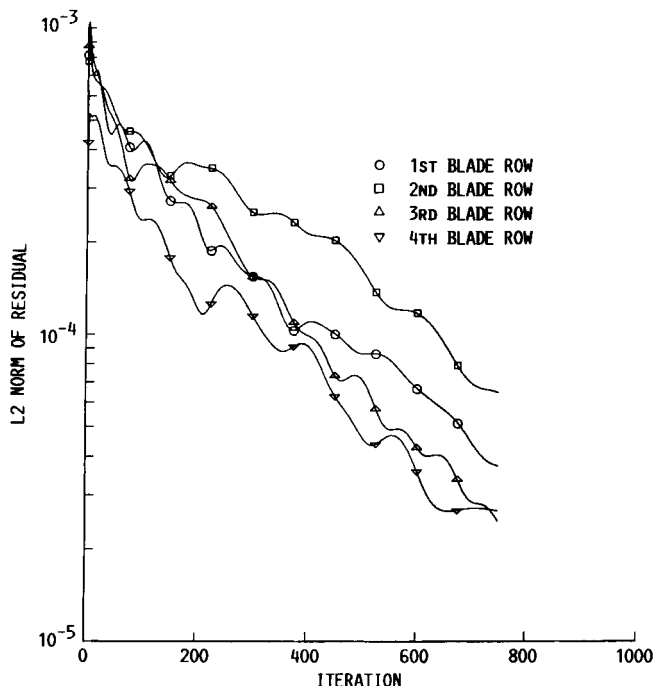


Figure 11.—Inner loop convergence (first swap).

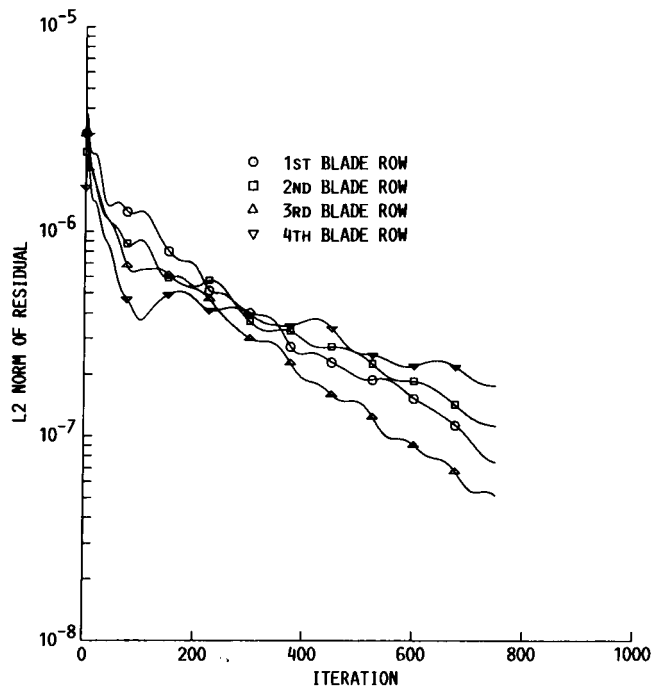


Figure 12.—Inner loop convergence (fiftieth swap).

indicating that the simulation is complete. A comparison of the convergence history of the inner loop procedure is made in figures 11 and 12. The L2 norms of the individual solution residuals have dropped more than two orders of magnitude by the 50th iteration of the outer loop procedure compared to the initial iteration. Total CPU time consumed for the second simulation was 47 990 sec. Both the first and second simulations were run well beyond the time necessary for a complete simulation in order to show the asymptotic nature of the outer loop procedure.

The next set of color figures depicts the evolution of a total temperature distortion through the fuel pump turbine. The total temperature profiles are color coded so that red denotes regions of hot gas and blue denotes regions of cold gas. Figure 13 shows the profile midway through the first vane. The cross-passage profile is nearly independent of tangential position, and is nearly equal to that at the inlet. The lack of any significant variations in the tangential direction is a consequence of the total pressure field being uniform at the inlet, which precludes the development of any secondary or streamwise vorticity. Figure 14 shows the cross-passage total temperature profile midway through the first rotor. Within a rotating passage streamwise vorticity can be generated by a radial gradient of both total temperature and wheel speed. In the tip region these two mechanisms reinforce one another,

whereas in the hub region they oppose each other. The result is that streamwise vorticity is generated in the tip region and suppressed at the hub. The total temperature field is transported by the velocity field set up by the streamwise vorticity field; hence in the tip region the hot gas stream is convected up the pressure surface while the cold gas stream is convected down the suction surface. In the hub, where the generation of secondary vorticity is suppressed, little spanwise transport is observed. Upon exiting the first rotor, the total pressure field is no longer uniform. Any nonuniformity in the radial direction will produce additional streamwise vorticity within the second vane. This will cause a further redistribution of the total temperature field, as shown in figure 15. The cross-passage profile is midway through the second vane. Figure 16 shows the cross-passage profile midway through the second rotor. Once again the streamwise vorticity field causes the hot gas to be convected radially up the pressure surface, while the cold gas is convected radially down the suction surface.

These results show that inviscid mixing as a result of streamwise vorticity generation can produce significant temperature differences between the suction and pressure side of a blade. This difference can lead to local regions of high thermal stress which can cause blade failure. The ability to capture the physics associated with this inviscid mixing process is a key element in increasing the durability of turbine blades.

ORIGINAL PAGE
COLOR PHOTOGRAPH

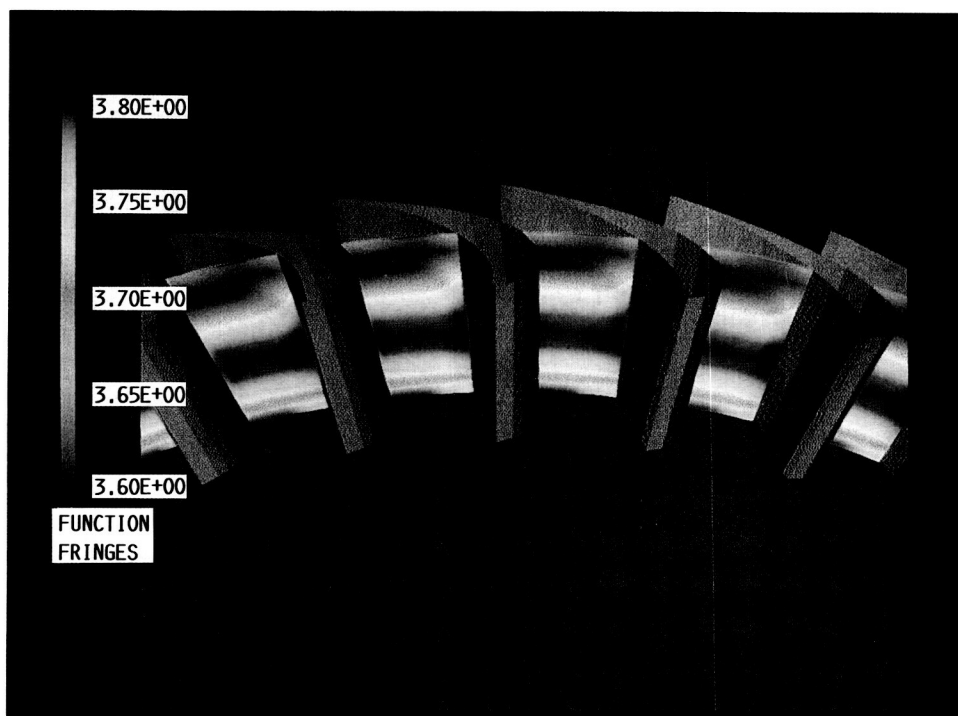


Figure 15.—Total temperature distortion in the second vane at midpassage.

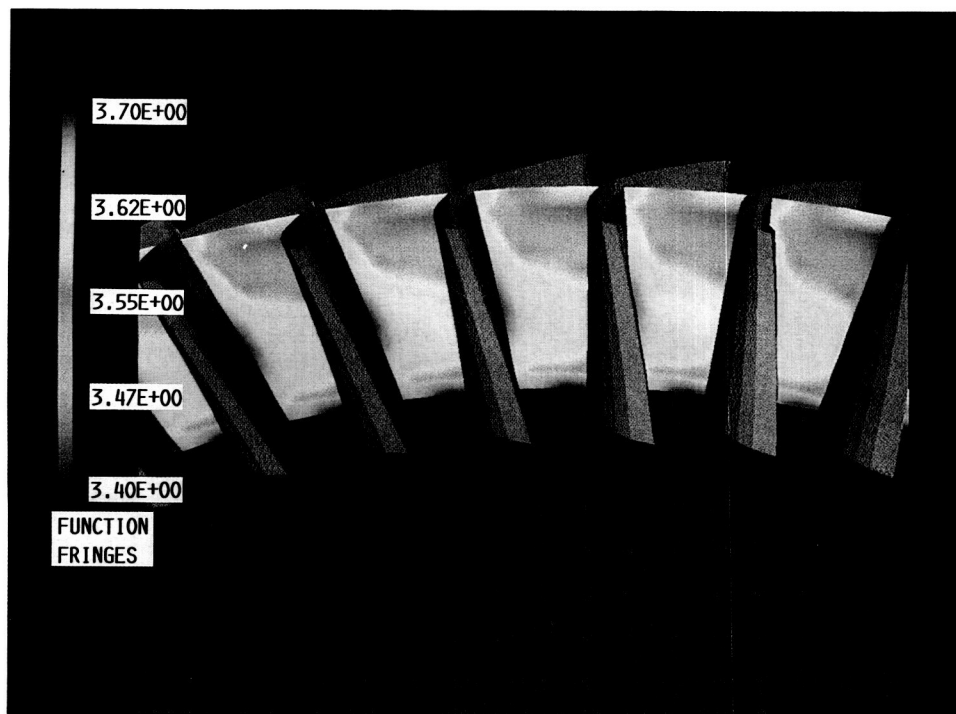


Figure 16.—Total temperature distortion in the second rotor at midpassage.

Figure 14.—Total temperature distortion in the first rotor at midpassage.

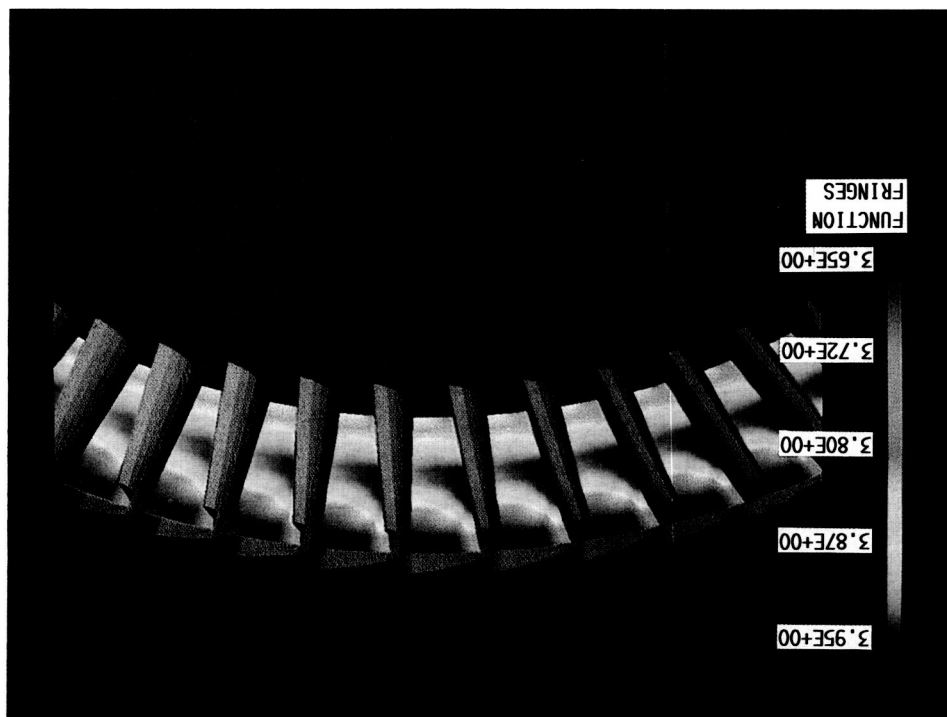
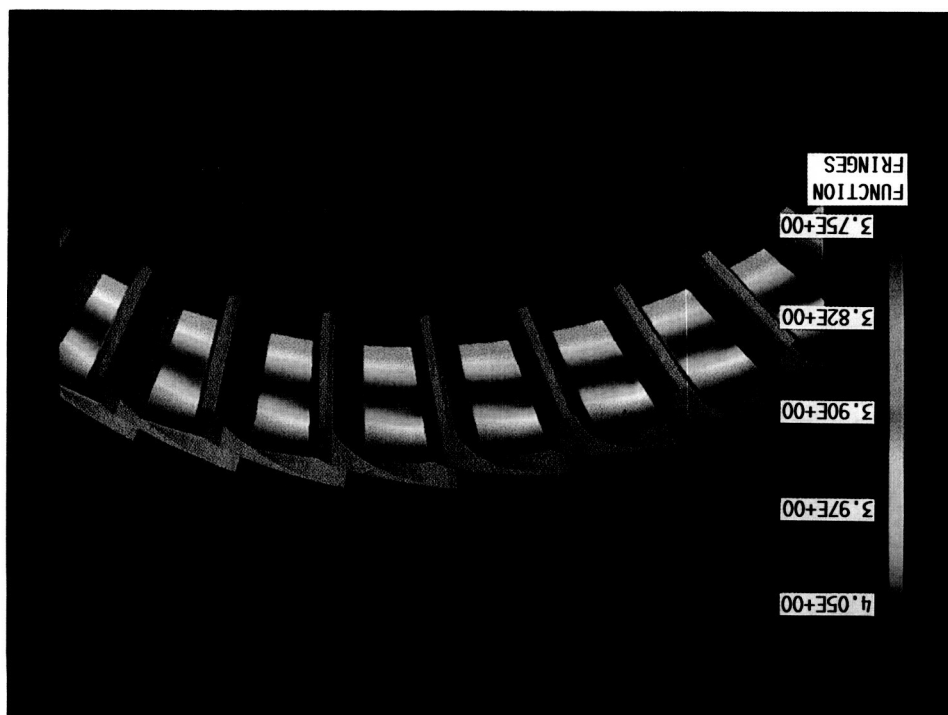


Figure 13.—Total temperature distortion in the first vane row at midpassage.



Conclusions

Three-dimensional flow field simulation of multistage turbomachinery is possible today through the solution of the average-passage equation system on multiprocessor computers. Multitasking of the solution algorithm has eliminated the real-time computational restrictions, and the utilization of modern, high-speed secondary storage puts the problem within the memory limitations of today's supercomputers. A code has been developed and successfully used on a Cray X-MP computer to simulate the inviscid flow through a two-stage turbine.

Acknowledgment

The authors wish to thank John Schwab and Kestutis Civinskas of NASA Lewis for their help in supplying the geometry and operating conditions for the rocket motor fuel pump turbine. Thanks also to Tony Hackenburg and Lucy Geyser of the Computer Services Division at NASA Lewis and the Cray personnel at NASA Lewis and Mendota Heights for their assistance in developing and benchmarking the code.

Special thanks to Mike Long of Cray Research for his time and effort in helping to generate the graphical results presented in this paper.

References

1. Adamczyk, J.J.: Model Equation for Simulating Flows in Multistage Turbomachinery. ASME Paper 85-GT-226, Mar. 1985. (NASA TM-86869).
2. Adamczyk, J.J.; Mulac, R.A.; and Celestina, M.L.: A Model for Closing the Inviscid Form of the Average-Passage Equation System. ASME Paper 86-GT-227, June 1986. (NASA TM-87199).
3. Celestina, M.L.; Mulac, R.A.; and Adamczyk, J.J.: A Numerical Simulation of the Inviscid Flow Through a Counterrotating Propeller. J. Turbomachinery, vol. 108, no. 2, Oct. 1986, pp. 187-193.
4. Jameson, A., et al.: Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. AIAA Paper 81-1259, June 1981.
5. Denton, J.D.; and Singh, U.K.: Time Marching Methods for Turbomachinery Flow Calculations. Application of Numerical Methods to Flow Calculations in Turbomachines, VKI Lecture Series 1979-7, Von Karman Institute for Fluid Dynamics, Rhodes Saint Genese, Belgium, 1979.
6. Mulac, R.A.: A Multistage Mesh Generator for Solving the Average-Passage Equation System. NASA CR-179539, 1988.

1. Report No. NASA TM-89845		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Utilization of Parallel Processing in Solving the Inviscid Form of the Average-Passage Equation System for Multistage Turbomachinery				5. Report Date	
				6. Performing Organization Code	
7. Author(s) Richard A. Mulac, Mark L. Celestina, John J. Adamczyk, Kent P. Misegades, and Jef M. Dawson				8. Performing Organization Report No. E-3508	
				10. Work Unit No. 505-90-01	
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Memorandum	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes Prepared for the 8th Computational Fluid Dynamics Conference, sponsored by the American Institute of Aeronautics and Astronautics, Honolulu, Hawaii, June 9-11, 1987. Richard A. Mulac and Mark L. Celestina, Sverdrup Technology, Inc., (Lewis Research Center Group), NASA Lewis Research Center; John J. Adamczyk, NASA Lewis Research Center; Kent P. Misegades and Jef M. Dawson, Cray Research, Inc., Mendota Heights, Minnesota.					
16. Abstract A procedure is outlined which utilizes parallel processing to solve the inviscid form of the average-passage equation system for multistage turbomachinery along with a description of its implementation in a FORTRAN computer code, MSTAGE. A scheme to reduce the central memory requirements of the program is also detailed. Both the multitasking and I/O routines referred to in this paper are specific to the Cray X-MP line of computers and its associated SSD (Solid-State Disk). Results are presented for a simulation of a two-stage rocket engine fuel pump turbine.					
17. Key Words (Suggested by Author(s)) Turbomachinery Aerodynamics			18. Distribution Statement Unclassified - Unlimited Subject Category 07		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No of pages 12	
				22. Price* A02	