JOURNAL OF PROPULSION AND POWER
Vol. 15, No. 1, January–February 1999

Unsteady Flowfield Simulation of Ducted Prop–Fan Configurations

J. Mark Janus,* Howard Z. Horstman,† and David L. Whitfield‡
Mississippi State University, Mississippi State, Mississippi 39762

A technique for the simulation of unsteady flows in and around complex rotating machinery is presented. Domain decomposition mechanisms are presented that extend the range of applicability of software developed for the time-accurate simulation of rotating machinery flowfields. The flow model uses the unsteady three-dimensional Euler equations, discretized as a finite volume method, utilizing a high-resolution approximate Riemann solver for cell interface flux definitions. The numerical scheme is an approximately factored, block lower–upper, implicit, Newton iterative-refinement method. Multiblock domain decomposition is used to partition the field radially and axially, as well as azimuthally into an ordered arrangement of blocks that exhibit varying degrees of similarity. Block–block relative motion is achieved in axial and radial directions using local grid distortion. Block–block interfaces, including dynamic interfaces, are treated to mimic cell communication interior to a block. A general high-order numerical scheme is applied to satisfy the geometric conservation law. Solutions of two configurations are presented, a ducted single-rotation prop–fan and a rotor/deswirl–vane combination that form a single-stage fan. Comparisons are made with other numerical solutions for these geometries and available experimental data.

Nomenclature

A, B, C = flux Jacobians
AR = advance ratio, V_∞/nD
D = blade diameter
e = total energy
$\boldsymbol{F}, \boldsymbol{G}, \boldsymbol{H}$ = flux vectors
J = metric Jacobian
n = blade rotational velocity, revolutions/s
p = pressure
$\boldsymbol{Q}$ = computational solution vector
$\boldsymbol{q}$ = Cartesian solution vector
$\boldsymbol{R}$ = residual vector
t = time, s
U, V, W = contravariant velocities, ξ, η, and ζ components, respectively
u, v, w = flow velocity, x, y, and z components, respectively, ft/s
x, y, z = Cartesian coordinates, ft
α = angle of attack, deg
$\beta^{3/4}$ = 3/4 tip radius blade setting angle, deg
γ = ratio of specific heats, C_p/C_v
$\Delta\tau$ = computational time step
θ, ψ = solver adjustable parameters
ξ, η, ζ = computational coordinates (axial, radial, and circumferential, respectively)
ρ = flow density, slugs/ft^3

Subscripts
i, j, k = spatial indices, ξ, η, and ζ, respectively
∞ = freestream conditions

Superscripts
n = time level index
P = iteration index
$+, -$ = associated with positive and negative eigenvalues, respectively

Introduction

NASA has been active in investigating advanced turboprop propulsion systems and has demonstrated that large savings in fuel consumption are possible using these advanced turboprop systems compared with conventional turbofan systems.[1] For reasons that include acoustics, performance, and passenger acceptability, the interest has shifted toward ducted rather than unducted designs. The computational fluid dynamics (CFD) effort presented herein was initiated to extend flow analysis software developed for the time-accurate simulation of unducted geometries. The extended capability was accomplished by incorporating additional domain decomposition mechanisms to enable the simulation of unsteady, ducted prop–fan flows, i.e., combined external and internal flow.

Considerable effort has been expended in the CFD community toward the computation of flow about prop–fan configurations.[2–17] Various levels of models, i.e., potential,[2,3] Euler,[4–12] and Navier–Stokes,[14–17] have been used in the flowfield description. Some implementations demonstrated only single blade-row capability,[2,5,7,8,10,15,16] whereas others demonstrated the ability to simulate multiple blade-row (multistage) geometries.[3,6,9–14,17] Of these multistage analyses, many were devoted to steady-state computations or computations that had been averaged in some sense.[3,6,9,11,13,17] The interest here, though, is maintaining the ability to perform a time-accurate analysis of the flow within a complex rotating configuration while simultaneously simulating the environment in which such a machine would operate. One goal of the development of unsteady flow analysis capability that includes nonlinear effects is to enable the coupling of software from other disciplines such as structures and/or acoustics to enhance configuration design through the use of interdisciplinary computa-

Presented as Paper 92-0521 at the AIAA 30th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1992; received March 2, 1995; revision received April 1, 1998; accepted for publication June 15, 1998.

*Associate Professor, Department of Aerospace Engineering, National Science Foundation, Engineering Research Center for Computational Field Simulation. Senior Member AIAA.

†Graduate Research Assistant, Department of Aerospace Engineering, National Science Foundation, Engineering Research Center for Computational Field Simulation. Member AIAA.

‡Distinguished Professor, Department of Aerospace Engineering, National Science Foundation, Engineering Research Center for Computational Field Simulation. Member AIAA.

tional tools. This paper outlines the basic computational technology embodied in the turbomachinery application tool referred to as UNCLE_TURBO (unsteady computation of field equations for turbomachinery).

The approach taken in this effort is to further improve a computational tool that has been evolving for the numerical solution of rotating machinery problems on dynamic multiblock grids.[10,12] The present effort uses computational techniques and experience gained in computing unsteady flows about complex geometries using dynamic multiblock grids. Collectively, Refs. 10 and 12 contain detailed discussions of the research issues encountered in the development of such a computational tool. Among these issues are the numerics of the flow solver, which include the equation formulation, the numerical flux at cell faces for a cell-centered scheme, and the implicit-solution algorithm. The dynamic multiblock grid approach, including techniques that have been developed for this particular type of problem involving rotating subdomains, is discussed at length in these references. The result of that effort was a relatively simple and compact flow solver capable of handling any number of blades in any number of blade rows. The development of that computational tool made it possible to consider general unducted prop–fan geometries.

The technological extensions to that tool that enable the modeling of both the external and internal flowfields of a complex ducted rotating configuration are presented here. A more complete description of the extensions can be found in Ref. 18. In addition, to demonstrate the utility of the software package, it is applied toward the simulation of flow in and around various complex rotating configurations. The numerical solutions are compared with those obtained from other researchers and experimentally collected wind-tunnel data. The two configurations selected for comparison are essentially those used in a previous numerical investigation,[13] yet the solutions presented here represent unsteady flowfield simulations that are time averaged for comparison purposes.

One configuration consists of a standard eight-blade, single-rotating prop–fan with SR7 blade design, encircled by a short cowl. Steady-state numerical studies were performed[13] on this configuration along with more recent unsteady, angle-of-attack investigations.[17] The other configuration investigated is that of a single-stage fan (rotor and deswirl vane), experimentally tested by NASA.[19–22] This configuration consists of 12 rotor blades and 32 deswirl vanes (stators) enclosed by a nacelle. It essentially forms a fan stage with a total pressure ratio of 1.15 followed by a flow nozzle. An analysis of this configuration using the average-passage approach was performed by Hall et al.,[13] thus providing an independent numerical source for comparison. The grids used for both configurations were derived from those used in Ref. 13. For the fan stage, to maintain grid continuity, only the circumferential point density was adjusted to attach the rotor to the stator grid.

The next section briefly outlines the mathematical flowfield model developed for a domain whose discrete elements (volumes) are free to move and distort. This is then followed by a brief review of the numerical scheme developed for the solution of the modeling equations. A description is then presented of the approach used to extend the software by incorporating additional domain decomposition mechanisms for accommodating additional component relative motion. This is followed by a discussion of the results of the two rotating configuration simulations. In the final section, conclusions are drawn and recommendations made with regard to software applicability and future development.

Flow Solver

Equation Formulation

The three-dimensional unsteady Euler equations were transformed from Cartesian coordinates to the time-dependent curvilinear coordinates

$$\xi = \xi(x, y, z, t), \quad \eta = \eta(x, y, z, t), \quad \zeta = \zeta(x, y, z, t), \quad \tau = t$$

and then cast into the following semidiscretized integral, or finite volume, form (with cell centers denoted i, j, k):

$$\frac{\partial \boldsymbol{Q}}{\partial \tau} + \delta_i \boldsymbol{F} + \delta_j \boldsymbol{G} + \delta_k \boldsymbol{H} = 0 \tag{1}$$

where $\delta_*(.) = (.)_{*+1/2} - (.)_{*-1/2}$.

$$\boldsymbol{Q} = J \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad \boldsymbol{F} = J \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U(e + p) - \xi_t p \end{bmatrix}$$

$$\boldsymbol{G} = J \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V(e + p) - \eta_t p \end{bmatrix}, \qquad \boldsymbol{H} = J \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W(e + p) - \zeta_t p \end{bmatrix}$$

$$e = \frac{p}{\gamma - 1} + \frac{\rho(u^2 + v^2 + w^2)}{2}$$

Consider the following difference expression encompassing a broad class of difference representations for advancing the solution of Eq. (1) in time[23]:

$$\frac{(1 + \psi)\Delta \boldsymbol{Q}^n - \psi \Delta \boldsymbol{Q}^{n-1}}{\Delta \tau} = (\theta - 1)\boldsymbol{R}^n - \theta \boldsymbol{R}^{n+1} \tag{2}$$

where

$$\boldsymbol{R} = \delta_i \boldsymbol{F} + \delta_j \boldsymbol{G} + \delta_k \boldsymbol{H}$$

$$\Delta \boldsymbol{Q}^n = \boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n = \Delta(J\boldsymbol{q})^n = (J\boldsymbol{q})^{n+1} - (J\boldsymbol{q})^n$$

The linearized form can be written as a Newton method in the form[24]

$$\boldsymbol{L}'(\boldsymbol{q}^{P-1})(\boldsymbol{q}^P - \boldsymbol{q}^{P-1}) = -\boldsymbol{L}(\boldsymbol{q}^{P-1}), \qquad P \geq 1 \tag{3}$$

with $\boldsymbol{q}^0 = \boldsymbol{q}^n$. The sequence of vectors $\{\boldsymbol{q}^P\}$ presumably converges to the solution vector $\boldsymbol{q}^{n+1}$, where $n + 1$ denotes the next time level. The residual vector $\boldsymbol{L}(\boldsymbol{q}^{P-1})$ and the operator $\boldsymbol{L}'(\boldsymbol{q}^{P-1})$ for this broad class of explicit and implicit schemes were derived in Ref. 25, and are given by

$$\boldsymbol{L}(\boldsymbol{q}^{P-1}) = \boldsymbol{q}^{P-1} - \boldsymbol{q}^n + \frac{\boldsymbol{q}^n}{J^{n+1}}\left(\Delta J^n - \frac{\psi}{1 + \psi}\Delta J^{n-1}\right) - \frac{1}{1 + \psi}\left\{\overline{\Delta\tau}[(\theta - 1)\boldsymbol{R}^n - \theta \boldsymbol{R}^{P-1}] + \psi \frac{J^{n-1}}{J^{n+1}}\Delta \boldsymbol{q}^{n-1}\right\} \tag{4}$$

$$\boldsymbol{L}'(\boldsymbol{q}^{P-1}) \equiv \frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{L}(\boldsymbol{q}^{P-1}) = I + \frac{\theta\overline{\Delta\tau}}{1 + \psi}\left[\frac{\partial}{\partial \boldsymbol{q}}\boldsymbol{R}(\boldsymbol{q}^{P-1})\right] = I + \frac{\theta\overline{\Delta\tau}}{1 + \psi}M(\boldsymbol{q}^{P-1}) \cdot \tag{5}$$

where $\overline{\Delta\tau} = \Delta\tau/J^{n+1}$, and

$$M(\boldsymbol{q}^{P-1}) \cdot = \delta_i A(\boldsymbol{q}^{P-1}) \cdot + \delta_j B(\boldsymbol{q}^{P-1}) \cdot + \delta_k C(\boldsymbol{q}^{P-1}) \cdot$$

with the flux Jacobians

$$A \equiv \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{q}}, \qquad B \equiv \frac{\partial \boldsymbol{G}}{\partial \boldsymbol{q}}, \qquad \boldsymbol{C} \equiv \frac{\partial \boldsymbol{H}}{\partial \boldsymbol{q}}$$

Equation (4) is not a particularly common form of the residual vector. For example, the metric Jacobian normally does not explicitly appear. Time-varying Jacobians require the solution of an additional equation, the geometric conservation law (GCL),[26] written here using the same general-difference expression that was used for the flow conservation laws[25]

$$\frac{(1+\psi)\Delta J^n - \psi\Delta J^{n-1}}{\Delta\tau} = (\theta-1)(\delta_i\xi_t + \delta_j\eta_t + \delta_k\zeta_t)^n - \theta(\delta_i\xi_t + \delta_j\eta_t + \delta_k\zeta_t)^m \tag{6}$$

where $m = n$ for $P = 1$, and $m = n + 1$ for $P > 1$.

Numerical Flux

The numerical flux vectors in the residual vector [Eq. (4)] are evaluated using flux-difference-split theory, the details of which can be found in Refs. 12, 27, and 28. To summarize the approach, Roe averaging[29] was used to determine a first-order numerical flux at cell faces. The general scheme of Osher and Chakravarthy[30] was then used for higher-order flux function definition; however, there is some deviation[28] in the construction of the higher-order schemes from that put forth in Ref. 30. For the solutions presented herein, second-order temporal accuracy via three-point backward time differencing ($\theta = 1$, $\psi = 1/2$), and second-order spatial accuracy with the van Leer limiter was used.

Iterative Refinement with Approximate Factorization

In keeping with the upwind nature of evaluating the flux vectors on the right-hand side of Eq. (4), the left-hand side (LHS) solution operator is also upwinded. The operator M can then be written as

$$M^{P-1}\cdot = \delta_i A^+\cdot + \delta_j B^+\cdot + \delta_k C^+\cdot + \delta_i A^-\cdot + \delta_j B^-\cdot + \delta_k C^-\cdot \tag{7}$$

with

$$A^+ = \left(\frac{\partial \boldsymbol{F}^+}{\partial \boldsymbol{q}}\right)^{P-1}, \quad A^- = \left(\frac{\partial \boldsymbol{F}^-}{\partial \boldsymbol{q}}\right)^{P-1}, \quad B^+ = \left(\frac{\partial \boldsymbol{G}^+}{\partial \boldsymbol{q}}\right)^{P-1}, \ldots$$

where $\boldsymbol{F}^+$, $\boldsymbol{F}^-$, $\boldsymbol{G}^+$, ... result from the Steger–Warming flux-vector-split theory. The elements of A^+, A^-, B^+, ... can be found in Ref. 31.

The operator M was approximately factored into two factors using the so-called *modified two-pass* scheme presented in Ref. 32. It can also be viewed as a symmetric Gauss–Seidel relaxation scheme without residual updating prior to the second pass. Consider Eq. (3) written as

$$(L + D + U)z = b \tag{8}$$

where L is a lower block triangular matrix with null matrices on the diagonal, D is a block diagonal matrix formed out of both positive and negative flux Jacobians, and U is an upper block triangular matrix with null matrices on the diagonal. If D is nonsingular, then Eq. (8) can be written as

$$D(I + D^{-1}L + D^{-1}U)z = b \tag{9}$$

Equation (9) can be factored and written as

$$(D + L)D^{-1}(D + U)z = b \tag{10}$$

Equation (10) requires two passes through the computational domain, with each pass requiring a block triangular solution. Notice from Eq. (4) that for steady-state solutions, or unsteady solutions where the vector sequence $\{\boldsymbol{q}^P\}$ converges to $\boldsymbol{q}^{n+1}$ at each time-step, the solution is independent of the solution operator on the LHS of the equation.

Boundary Condition Treatment

The approximate Riemann solver used here for the interface flux formulas and the flux-vector splitting used for flux Jacobian definitions are characteristic-based concepts; hence, the characteristic variable boundary conditions developed in Ref. 33 relative to a three-dimensional time-dependent body-fitted reference frame, are employed where applicable. As in Ref. 33, phantom cells are utilized to implement these boundary conditions. The changes in dependent variables, $\Delta\boldsymbol{q}^{P-1}$ and $\Delta\boldsymbol{q}^*$, are set to zero in the phantom cells for inflow, outflow, and impermeable boundaries.

Dynamic Multiblock Grid Structure

Computer simulations of complex flowfields can involve enormous grid sizes. In addition, the flow solver developed here can require more than 200 pieces of information to be stored in memory for each computational volume or cell. Even with the modern supercomputers available today, many complex problems cannot be completely stored and managed in primary memory. This difficulty is circumvented by breaking the entire flowfield into several smaller and more manageable components, or blocks. The practice of segmenting the flowfield is known as *domain decomposition*, and the gridding technique used to accomplish this is referred to as *composite gridding* the field. In Ref. 25, a dynamic, blocked grid method is developed for the specific case of cylindrical geometries. This method takes advantage of blocking features and restrictions suggested by general cylindrical geometries, and is referred to as *selected-similarity mapped multiblock*. The approach accomplishes domain decomposition by arranging groups of grid partitions with similar features, requiring one-to-one mapping between neighboring partitions, and using a fixed solution path throughout the partitioned domain. As illustrated in Fig. 1, the domain can be partitioned in three ways: radially, axially, and azimuthally.

A partitioning hierarchy exists in this domain decomposition strategy. First, the domain is divided into one or more radial zones. These zones would be strategically placed to aid in solving a complicated flow. For ducted-engine geometries, a natural partition location would be the shroud itself, separating internal and external flows into different zones. Radial partitioning can also be used to separate rotating surfaces such as hubs from stationary surfaces such as shrouds. Zones can have relative motion between each other. Each zone contains one or more axial partitions called *blade rows*. The blade rows can have relative motion between each other, allowing the modeling of multistage configurations. Each zone can have a different number of blade rows, and blade row interfaces do not necessarily have to continue across a radial zone interface. Each blade row is azimuthally partitioned into one or more blocks. These blocks can correspond to blade passages, although a blade row does not necessarily have to contain blades. Blade rows can contain different numbers of blocks, but the blocks have no relative motion within their respective blade row; they rotate together at whatever speed the blade row rotates. Each block is a structured collection of cells (finite volumes).

Some restrictions are placed on the decomposition to achieve an orderly one-to-one mapping between blocks and to aid in algorithm simplicity. Grid line continuity is maintained across all block interfaces. All local block axes follow the global axes of the grid. In other words, there is global similarity among all blocks concerning orientation and direction. The cell indexing limits, NI, NJ, and NK (corresponding to the ξ, η, and ζ directions, respectively), remain constant among all of the blocks within a blade row. Thus, all blocks within a blade row are the same size. While block sizes may vary across blade rows, the blade row azimuthal cell count (BRACC) of all blade rows must be the same. For all blade rows within a zone, NJ is constant, whereas NI and NK may vary. While NJ is constant in a zone, it can vary across zones.

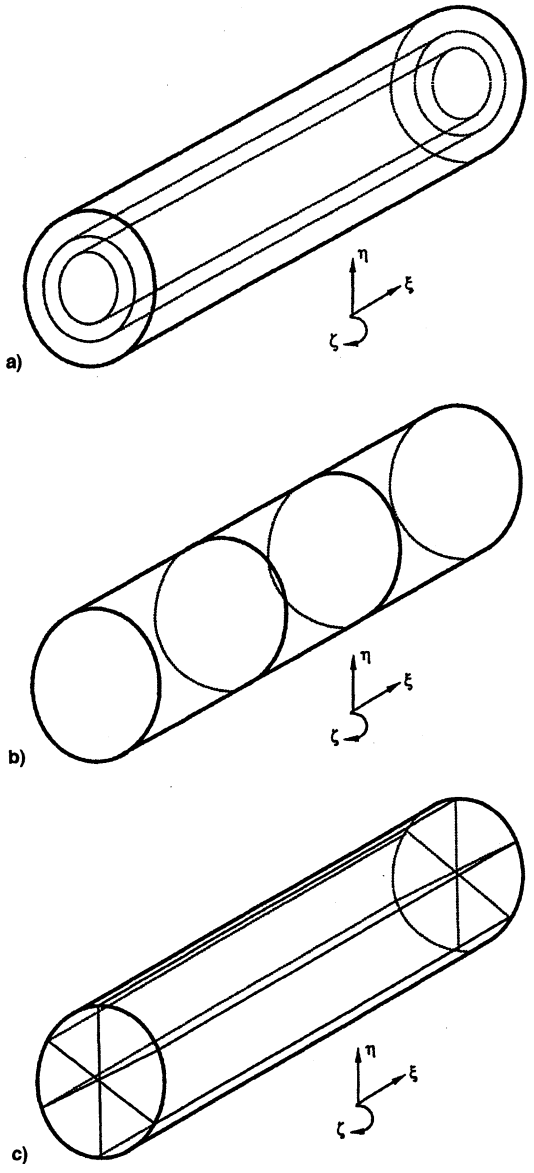

Fig. 1 Domain decomposition: a) radial, b) axial, and c) azimuthal partitioning.

Each zone can have an arbitrary number of blade rows with different NI indexes, but the zone axial cell count (ZACC) for all zones must be the same.

To illustrate radial and axial partitioning of a general grid, such as that shown in Fig. 1, consider Fig. 2. Shown here is a constant ζ-plane cross section of a domain partitioned into three zones. The first zone is then partitioned into three blade rows, the second into two blade rows, and the third into four blade rows. Note that the ZACC of all zones is the same (32 in this case). This is to maintain a one-to-one mapping of information across interfaces.

Figure 3 shows a constant η-plane cross section of further partitioning the first zone into blocks. Blade row one contains three blocks, blade row two contains two blocks, and blade row three contains four blocks. The BRACC for each blade row must be the same, and it can be seen to be the product of NK and the number of blocks in any particular blade row (24 in this case). Figure 3 also illustrates the block indexing used in the code. For example, the indices (1, 3, 2) refer to the second block of blade row 3 of zone 1.

To solve the Euler equations at every cell in a partitioned domain, a fixed solution path is desired that is equivalent to solving an unpartitioned domain. The path used in the flow

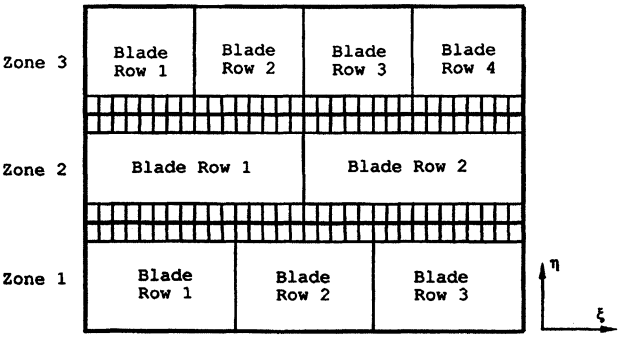

Fig. 2 Radial and axial partitioning.

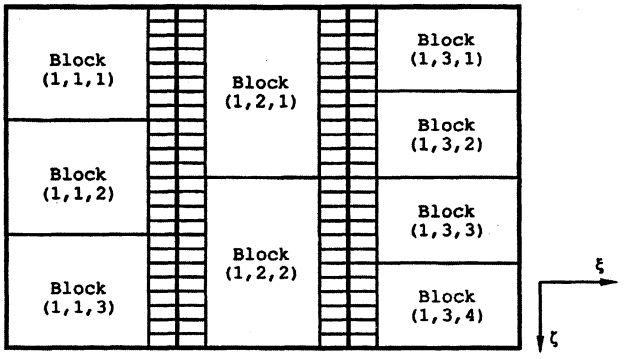

Fig. 3 Azimuthal partitioning.

code is a forward sweep through all of the blocks in the domain, and then a backward sweep retracing the path. All blocks are stored in secondary memory such as Cray's solid-state storage device and brought in-core for processing. This requires two primary to secondary memory swaps per solution cycle. When in-core, each block is processed by sweeping the cells along diagonal planes.[34]

The forward sweep begins at the first block of the first blade row of the first zone and continues through each block of each blade row in that zone. From the last block of the zone, the solution path proceeds to the first block of the first blade row of the next zone. The backward sweep then begins at the last block of the last blade row of the last zone and continues to the first block of the first blade row of the first zone, retracing the path of the forward sweep. At the completion of the backward sweep, the dependent variable vector for each cell in the domain would have been refined (advanced one iteration) toward the goal of converging to q^{n+1}.

Internal Interface Management

As mentioned, each block along the solution path is read into primary memory from secondary memory, processed, and returned to secondary memory. This necessitates a system of internal interface management for each block to communicate with its neighboring blocks. Required information from inside the domain of each block is retained in primary memory when the block is moved to secondary memory, and used to pass information along to neighboring blocks as they are transferred to primary memory for processing. This method is referred to as an *extraction–injection* technique, and the required information is retained in buffer zones.

Buffer zones are areas of shared information between adjacent blocks. The buffer zones are four cells wide, two from each side of the block interface. The code manages this information by establishing three buffer regions in primary memory: one to manage data transfer in the radial direction, one to manage axial data transfer, and one to manage azimuthal data transfer. Azimuthal data transfer is a straightforward process. No relative motion exists between blocks of an individual blade row, and so information extracted from the cells of one

block is simply injected into the neighboring cells of an adjacent block. Axial and radial transfer, however, can involve blocks in relative motion, resulting in interfaces with continually changing communication partners. This necessitates a bookkeeping system that tracks interfaces in relative motion, and chaperones the communication between the cells of one block and those of another. The bookkeeping method used in UNCLE_TURBO involves monitoring the ζ (or k) indices of neighboring blocks.

For configurations involving some degree of symmetry, often only a piece of the full 360-deg geometry is modeled, and then the appropriate data are extracted and imaged to form full buffer regions. The process of data communication across block interfaces is not a trivial process, involving delicate timing when transferring data back and forth from block memory to buffer memory. Care must be taken to ensure proper transfer of data along the solution path.

Localized Grid Distortion

The modeling of geometries with components in relative motion requires a method to control the grid. The method used here is referred to as *localized grid distortion* (LGD).[12] This technique provides a means to relink grid lines across dynamic interfaces while maintaining arbitrary cell spacing and solution time-stepping. This method does not affect the data transfer procedures described earlier.

Distortion regions are established along relative-motion interfaces. Across axial blade-row interfaces, an axial distortion region is designated, and radial distortion regions are established across radial zone interfaces. These distortion regions cannot extend into fixed surface areas of blades, fins, wings, etc., because the physical geometry must remain unaltered. The radial and axial coordinate components remain unchanged; only the azimuthal coordinate changes. Thus, hub and shroud surfaces also remain unaltered.

Grid points are redistributed across distortion regions by means of a simple linear interpolation algorithm. Points on one edge of the distortion region are matched with points on the other edge by using k-index pointers. The pointer is updated (reflecting the interface motion) by computing an azimuthally averaged cell $\Delta\theta$ and incrementing or decrementing the pointer based on whether the relative motion between two blade rows has reached a multiple of $\Delta\theta$. Interface motion is determined by using relative angular velocity and the solution time-step. When a multiple of $\Delta\theta$ is reached, the grid lines change partners or click across an interface. Depending on the time-step, a relative-motion grid interface can click every cycle or undergo distortion for a number of cycles before clicking.

Radial distortion regions along a zone must extend the entire axial length of that zone and may involve several blade rows. For radial interfaces, points in r, θ space are used. Figure 4 illustrates a generic radial grid interface undergoing distortion and, finally, relinking. Across axial interfaces, points in x, θ space are considered. Axial interface relinking occurs in a similar manner. In each case, only the θ coordinate is adjusted. In cases of symmetry, only a portion of the full 360-deg grid is used. The distortion region interface points are then imaged to 360 deg so that the interpolation can proceed as if the full grid were being modeled. Special conditions are required in areas where axial and radial distortion zones overlap. All relative-motion points along the edges of a doubly distorted region must relink or click at the same time.

When dealing with multiblock techniques, an issue of the accuracy necessary for the communication between blocks arises. Ultimately, the best communication possible is that analogous to a singly gridded field. The approach taken here with regard to block–block interface communication is to mimic the cell communication within the blocks, thereby theoretically incurring no block-interface spatial-differencing induced error. Hence, block–block interfaces are maintained with up to third-order spatial accuracy. For the case of reentrant block–block interface communication, periodicity and previous Δq^{P-1} and Δq^* have been found to be satisfactory.

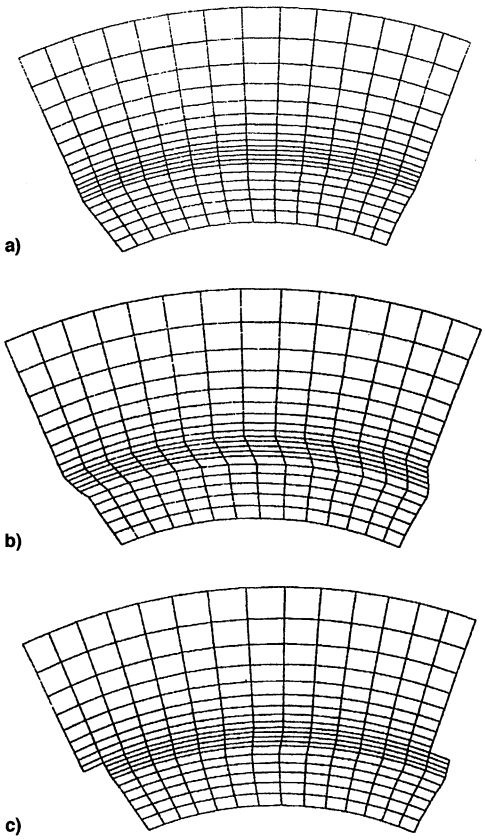

Fig. 4 Dynamic distorting interface: a) initial, b) distorted, and c) clicked grids.

Results

The software developed using the methodology described herein has been optimized for operation on Cray-class vector processors, although it has been ported to cache-based systems such as workstations. The software Cray Y-MP performance statistics are 1) processing = 22,500 computational cells per second and 2) memory usage = 225 words per cell total (150 words per cell in-core).

Ducted SR7 Prop–Fan

The first configuration tested was a ducted SR7 prop–fan.[13] No experimental data were available for such a configuration; however, comparison with two other computer simulations of the same geometry is made here. This geometry was simulated using two different decomposition strategies. Figure 5 illustrates the geometry and grid surfaces of the ducted SR7 prop–fan. The prop–fan consists of eight blades designed with a 41 deg of sweep at the tip and a $\frac{3}{4}$ radius blade setting angle of 60.2 deg. The hub is contoured to eliminate flow choking at the hub. Figure 6 shows the axisymmetric projection of the 69 $\times$ 45 $\times$ 15 H-grid used in the simulation. Because of symmetry, only one blade passage was modeled, and the solution was then imaged to form a full 360 deg. The grid was partitioned into two zones; one consisting of points interior to the shroud radius and the other consisting of points to its exterior. Each zone was made up of a single blade row, containing only one block. Hence, the geometry was only a two-block configuration, which greatly reduced execution time. For comparison purposes, the domain was modeled with and without relative motion between inner and outer zones, i.e., both zones rotating together (no relative motion) and the inner zone rotating with the outer zone stationary (relative motion).

The solution was obtained for a freestream Mach number of 0.7 and an *AR* of 3.06. Characteristic variable boundary conditions were applied to all solid surfaces. The computation time on a Cray Y-MP for three prop revolutions (2688 time steps) was approximately $1\frac{1}{2}$ h using a single Newton iteration. A comparison of convergence histories in Fig. 7 shows a similar, though slightly reduced, convergence rate for the relative motion approach. A slight difference was expected, as discussed in Ref. 25. Virtually no differences in surface pressures resulting from the two approaches were seen. The Euler solution obtained here was compared with another Euler analysis and a frequency-domain panel analysis developed for ducted

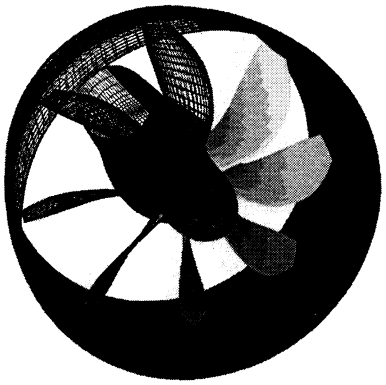

Fig. 5 Ducted SR7 prop-fan geometry and grid surfaces.

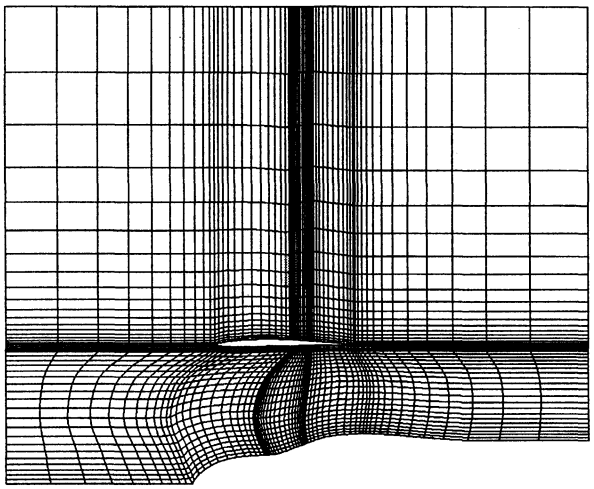

Fig. 6 Axisymmetric projection for ducted SR7 prop-fan.

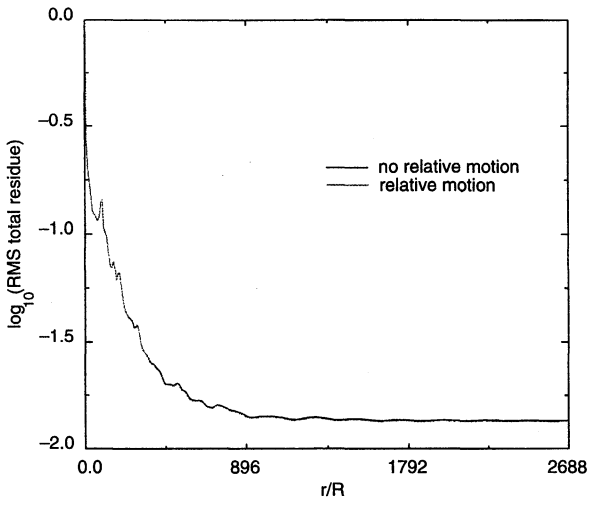

Fig. 7 Ducted SR7 convergence history.

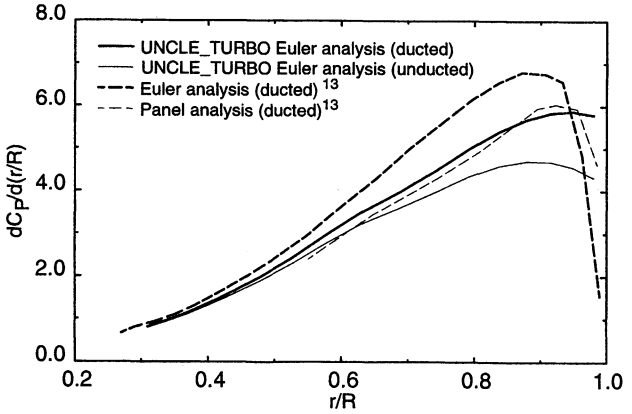

Fig. 8 Predicted blade elemental power coefficient distribution for ducted SR7 prop-fan.

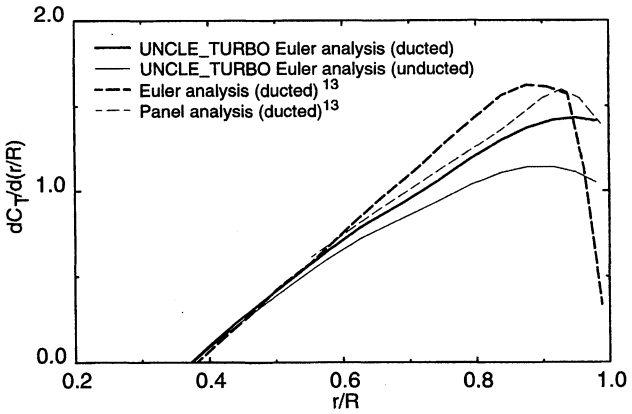

Fig. 9 Predicted blade elemental thrust coefficient distribution for ducted SR7 prop-fan.

prop-fan flow.[13] Figure 8 shows the comparison of predicted spanwise elemental power coefficient loading distributions for the various methods, and the predicted spanwise elemental thrust coefficient loading distributions are shown in Fig. 9. Also shown in Figs. 8 and 9 are the distributions for the prop-fan with a flowthrough (unducted) cowl. The solution indicates that the cowl increases blade loading, particularly near the tips. It should be pointed out that the tip region is rather poorly resolved, in that it is a blunt tip treated with an H-grid, resulting in a slight computational void in the domain. To more accurately model the extreme bluntness of the tip, a tip model utilizing a vena contracta assumption was used.

NASA 1.15 Pressure Ratio Fan Stage

The second configuration tested was a 1.15 total pressure ratio fan stage extensively tested by NASA.[19-22] A technical description of the 25:1 bypass ratio turbofan engine fan stage is shown in Fig. 10. Figure 11 shows the geometry and grid surfaces of the ducted prop-fan. The stage consists of a rotor containing 12 blades and a stator (deswirl vane) containing 36 blades. The actual geometry stator row contains 32 blades; however, this configuration was more computationally expensive to model, thus, the 36-blade configuration was selected instead. A 225 × 52 × 25 H-grid derived from the grid used in Ref. 13 (225 × 52 × 15) was used to model one rotor passage and three stator passages. Figure 12 shows a meridional view of the grid. The grid was partitioned into two zones of three blade rows each. The second blade row contains the rotor and is the only nonstationary blade row. The third blade row contains the stator. Grid distortion regions are established as illustrated by the shaded areas of Fig. 13.

The solution was obtained for a freestream Mach number of 0.75 and an *AR* of 2.86. Imaging was again used to obtain a

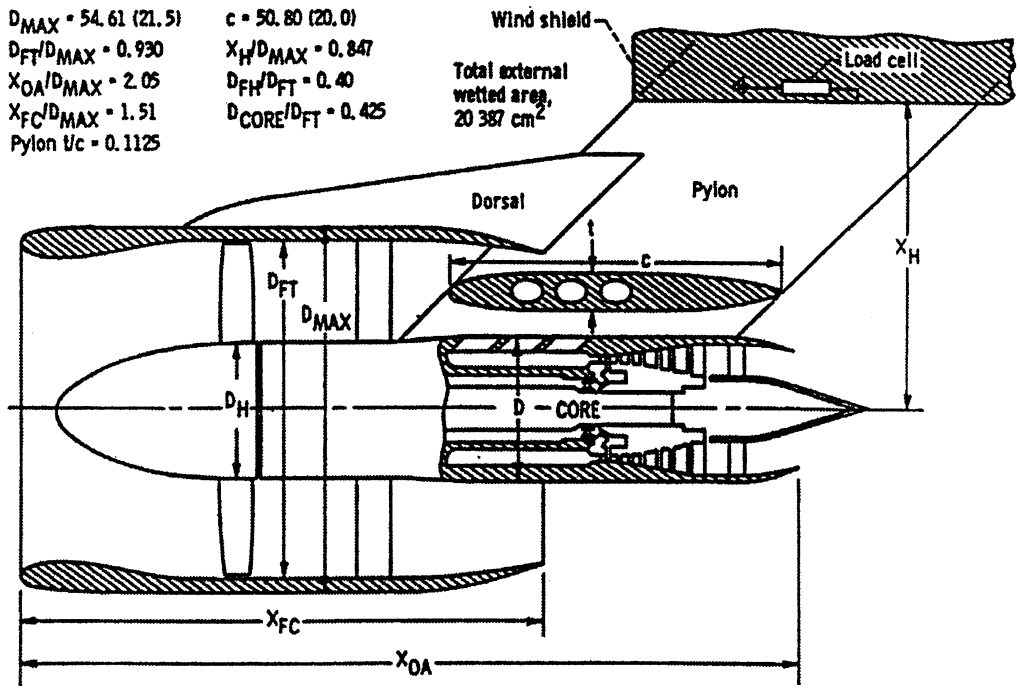

Fig. 10 NASA 1.15 pressure ratio fan stage (dimensions in cm).[19]

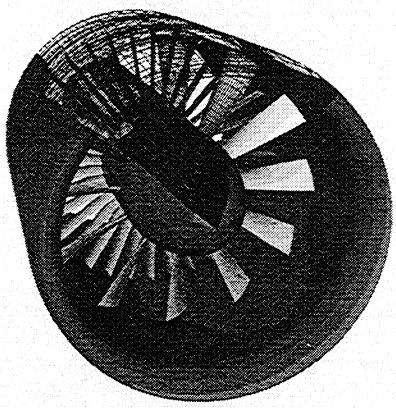

Fig. 11 NASA 1.15 pressure ratio fan stage geometry and grid surfaces.

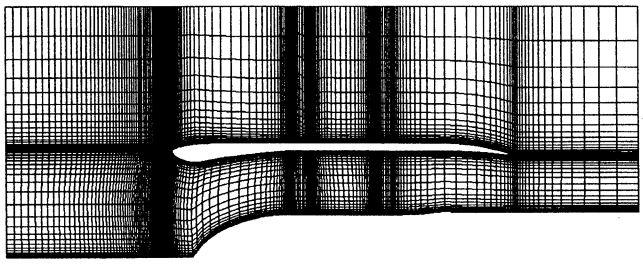

Fig. 12 NASA 1.15 pressure ratio fan stage axisymmetric plane projection.

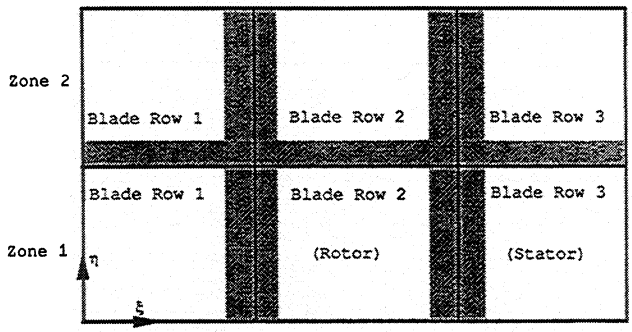

Fig. 13 NASA 1.15 pressure ratio fan stage distortion regions.

complete 360-deg solution. Characteristic variable boundary conditions were used on all surfaces except the blades where zero-pressure gradient boundary conditions were used instead. Computation times on a Cray Y-MP were approximately 10 h for each rotor revolution (2880 time steps), using a single Newton iteration, with seven revolutions used for the results shown here (note, on the final revolution three Newton iterations were used).

Figure 14 shows the locations of compared pressure data in Figs. 15–17. Figure 15 compares the predicted cowl leading-edge surface static pressure coefficients (azimuthally and time averaged) to experimental data and to the Euler analysis of Ref. 13. Note the nondissipative nature of the present algorithm as seen at the leading edge of the cowl, where the static pressure coefficient changes abruptly. Also, the solution seems to indicate a weak shock forming along the upper surface of the cowl. The critical pressure coefficient for a freestream Mach number of 0.75 is approximately −0.6. Figure 15 shows that the flow has expanded to a pressure coefficient far beyond this value, thus indicating a relatively high supersonic velocity near the leading edge.

Figure 16 shows the nozzle inlet radial distribution of total pressure. The total pressure ratio was numerically sampled for a complete period of numerical simulation at a location that was aligned with the center of the stator passage. The solution shows good agreement over the profile, except near the walls where viscous effects dominate the experimental data. The shaded region indicates the extent of unsteady fluctuations in total pressure ratio as predicted here. Note, the total pressure predicted here is higher than experiment. This is consistent with the lack of viscous dissipation (an Euler simulation is presented here) and lack of numerical dissipation from the upwind algorithm used in this simulation. Thus, the addition of viscous dissipation (a Navier–Stokes simulation) will move the solution into closer agreement with experiment.

The nozzle boat-tail surface static pressure coefficient distribution (azimuthally and temporally averaged) is given as a function of projected area ratio in Fig. 17. As seen in the fig-

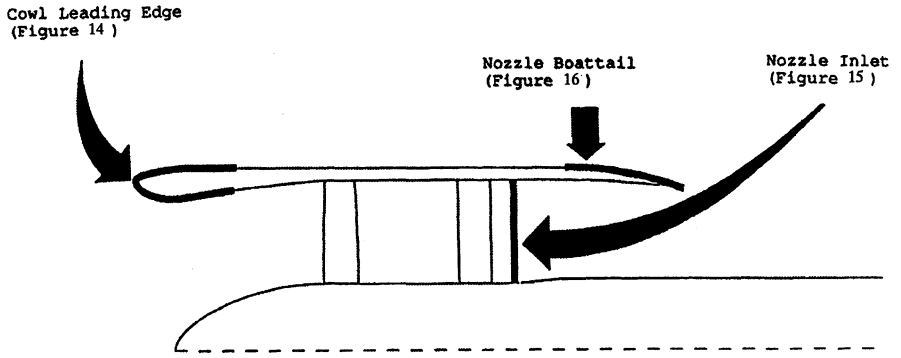

Fig. 14 NASA 1.15 pressure ratio fan stage predicted and experimental data locations.

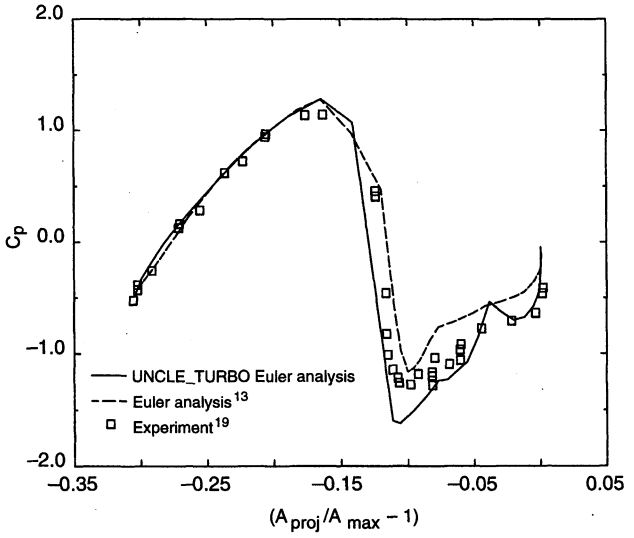

Fig. 15 Cowl leading-edge static pressure coefficient.

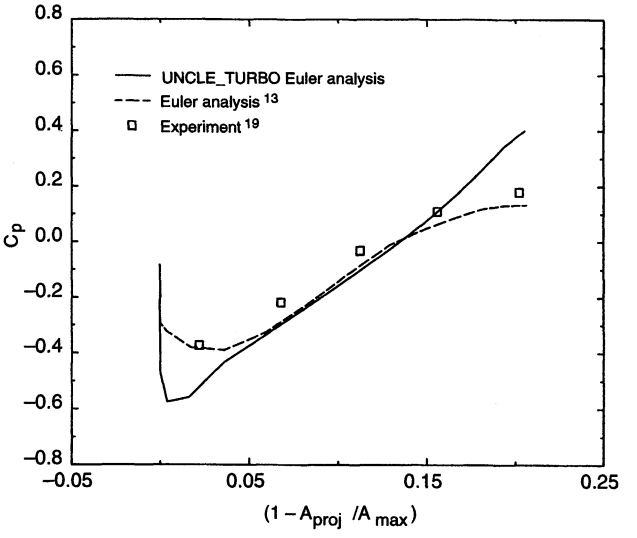

Fig. 17 Nozzle boat-tail static pressure coefficient.

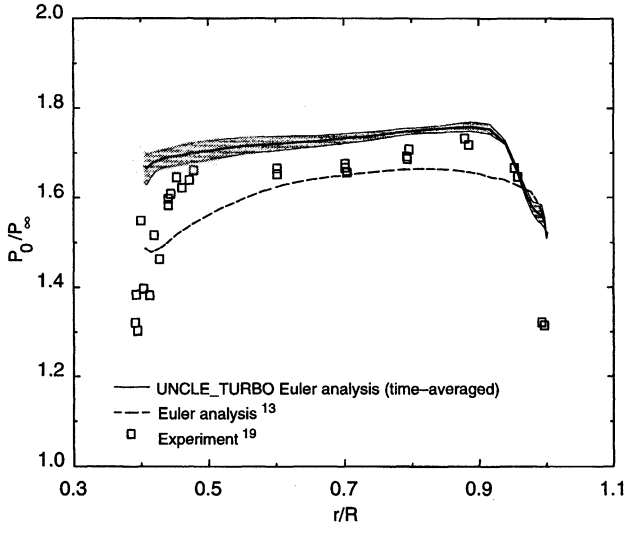

Fig. 16 Nozzle inlet total pressure ratio radial distribution (shaded area represents unsteady envelope).

ure, there is a slight discrepancy in the solution presented here. Located in the aft end of the machine, the boat-tail is affected by a number of things; namely, the amount of ingested mass (capture ratio), amount of work done internally on the fluid (rotational speed and efficiency), and the geometric shape of the boat-tail. Potential issues regarding the geometry shape and mass capture ratio have essentially been ruled out. It has been established that the correct geometry is being modeled. In addition, rather than attempting to match ingested mass (a computationally costly proposition), the experimental advance ratio was specified in this simulation. An integration of the mass flow entering the fan stage revealed that the ingested mass is in agreement with experiment for the case studied. The disagreement seen along the boat-tail region is likely a result of the higher values of total pressure predicted in the nozzle (Fig. 16). The slightly elevated total pressure in the nozzle will result in a higher energy fluid issuing from the nozzle exit. This fluid entrains the fluid in the boat-tail region, resulting in a lower pressure coefficient over the beginning of the boat-tail. This is followed by a higher pressure coefficient at the end of the boat-tail region, when the flow coming off the trailing edge of the boat-tail encounters the high-energy flow exiting the nozzle of the machine.

Conclusions

The basic inviscid computational technology embodied in the turbomachinery application tool referred to as UNCLE_TURBO has been presented. This enhanced software implementation can simulate axisymmetric or nonsymmetric multistage geometries with single-rotating, counterrotating, or rotor–stator configurations with or without shrouds or ducts, with arbitrary blade numbers and angles of attack. To facilitate a relatively simple and compact software package, certain restrictions regarding domain decomposition are imposed. The algorithm was tested on two ducted fan configurations and was shown to provide reasonable solutions to the flowfields about these geometries. Additional effort will be focused on improving relative-motion interface software flexibility while maintaining solution integrity. The flow model in the released version of UNCLE_TURBO can operate with or without viscous effects.

Acknowledgments

The research reported herein was performed by Mississippi State University under contract for NASA Lewis Research Center. Funding was provided by NASA Grant NAG-3-767. The authors would like to acknowledge several individuals who throughout the course of this endeavor have been extremely helpful, namely John Adamczyk and Dennis Huff of NASA Lewis Research Center, and Ed Hall of Allison Gas Turbine. In addition, appreciation is extended to Larry Bober and John Groeneweg of NASA Lewis Research Center for their continued support of prop–fan CFD analysis.

References

[1]Whitlow, J. B., Jr., and Sievers, G. K., "Fuel Savings Potential of the NASA Advanced Turboprop Program," NASA TM-83736, Sept. 1984.

[2]Jou, W. H., "Finite Volume Calculation of Three-Dimensional Flow Around a Propeller," AIAA Paper 82-0957, June 1982.

[3]Valarezo, W. O., "Calculation of Isolated and Installed Multiple Rotor Flows Using a Surface Panel Method," AIAA Paper 89-2214, Aug. 1989.

[4]Denton, J. D., and Singh, U. K., "Time Marching Methods for Turbomachinery Flow Calculation," *Application of Numerical Methods to Flow Calculations in Turbomachines*, von Kármán Inst. for Fluid Dynamics, VKI-LEC-SER-1979-7, Belgium, 1979.

[5]Bober, L. J., Chaussee, D. S., and Kutler, P., "Prediction of High Speed Propeller Flow Fields Using a Three-Dimensional Euler Analysis," AIAA Paper 83-0188, Jan. 1983.

[6]Adamczyk, J. J., "Model Equation for Simulating Flows in Multistage Turbomachinery," NASA TM-86869, Nov. 1984.

[7]Barton, J. M., Yamamoto, O., and Bober, L. J., "Inviscid Analysis of Advanced Turboprop Propeller Flow Fields," AIAA Paper 85-1263, July 1985.

[8]Yamamoto, O., Barton, J. M., and Bober, L. J., "Improved Euler Analysis of Advanced Turboprop Propeller Flows," AIAA Paper 86-1521, June 1986.

[9]Celestina, M. L., Mulac, R. A., and Adamczyk, J. J., "A Numerical Simulation of the Inviscid Flow Through a Counter-Rotating Propeller," NASA TM-87200, June 1986.

[10]Whitfield, D. L., Swafford, T. W., Janus, J. M., Mulac, R. A., and Belk, D. M., "Three-Dimensional Unsteady Euler Solutions for Propfans and Counter-Rotating Propfans in Transonic Flow," AIAA Paper 87-1197, June 1987.

[11]Mulac, R. A., Schneider, J. C., and Adamczyk, J. J., "Average-Passage Simulation of Counter-Rotating Propfan Propulsion Systems as Applied to Cruise Missiles," AIAA Paper 89-2943, July 1989.

[12]Janus, J. M., and Whitfield, D. L., "Counter-Rotating Propfan Simulations Which Feature a Relative-Motion Multiblock Grid Decomposition Enabling Arbitrary Time Steps," AIAA Paper 90-0687, Jan. 1990.

[13]Hall, E. J., Delaney, R. A., and Bettner, J. L., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems," NASA CR-185217, April 1990.

[14]Kobayakawa, M., and Nakao, M., "Numerical Solutions for the Flowfield Around a Counter-Rotating Propeller," *Journal of Aircraft*, Vol. 26, No. 5, 1989, pp. 417–422.

[15]Matsuo, Y., Saito, S., Arakawa, C., and Kobayashi, H., "Navier-Stokes Computations for Flowfield of an Advanced Turboprop," AIAA Paper 88-3094, July 1988.

[16]Saito, S., Nakamura, Y., Matsuo, Y., and Kobayashi, H., "Predicted Flow Field Around the Advanced Propeller at Take-Off," AIAA Paper 88-3151, July 1988.

[17]Hall, E. J., and Delaney, R. A., "Time-Dependent Aerodynamic Analysis of Ducted and Unducted Propfans at Angle of Attack," American Society of Mechanical Engineers, Paper 91-GT-190, June 1991.

[18]Horstman, H. Z., "Incorporation of Radial Partitioning in a Turbomachinery Flow Solver for Unsteady Ducted Applications," M.S. Thesis, Mississippi State Univ., Mississippi State, MS, Aug. 1991.

[19]Steffen, F. W., "Cruise Performance of an Isolated 1.15 Pressure Ratio Turbofan Propulsion Simulator at Mach Numbers from 0.6 to 0.85," NASA TM X-3064, June 1974.

[20]Wesoky, H. L., Abbott, J. M., Albers, J. A., and Dietrich, D. A., "Low-Speed Wind Tunnel Tests of a 50.8 Centimeter (20 in.) 1.15 Pressure Ratio Fan Engine Model," NASA TM X-3062, June 1974.

[21]Wesoky, H. L., and Steffen, F. W., "Wind Tunnel Tests of a 20 in. Diameter 1.15 Pressure Ratio Fan Engine Model," NASA TM X-71445, Nov. 1973.

[22]Osborn, W. M., and Steinke, R. J., "Performance of a 1.15 Pressure Ratio Axial-Flow Fan Stage with a Blade Tip Solidity of 0.5," NASA TM X-3052, Aug. 1974.

[23]Beam, R. M., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, No. 4, 1978, pp. 393–402.

[24]Ortega, J. M., and Rheinboldt, W. C., *Iterative Solution of Nonlinear Equations in Several Variables*, Academic, New York, 1970.

[25]Janus, J. M., "Advanced 3-D CFD Algorithm for Turbomachinery," Ph.D. Dissertation, Mississippi State Univ., Mississippi State, MS, May 1989.

[26]Thomas, P. D., and Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1978, pp. 1030–1037.

[27]Simpson, L. B., "Unsteady Three-Dimensional Thin-Layer Navier-Stokes Solutions on Dynamic Blocked Grids," Ph.D. Dissertation, Mississippi State Univ., Mississippi State, MS, Dec. 1988.

[28]Whitfield, D. L., Janus, J. M., and Simpson, L. B., "Implicit Finite Volume High Resolution Wave-Split Scheme for Solving the Unsteady Three-Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids," Mississippi State Univ., MSSU-EIRS-ASE-88-2, Mississippi State, MS, Feb. 1988.

[29]Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.

[30]Osher, S., and Chakravarthy, S., "Very High Order Accurate TVD Schemes," Inst. for Computer Applications in Science and Engineering, Rept. 84-44, Sept. 1984.

[31]Belk, D. M., "Unsteady Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids," Ph.D. Dissertation, Mississippi State Univ., Mississippi State, MS, Aug. 1986.

[32]Whitifeld, D. L., "Newton-Relaxation Schemes for Nonlinear Hyperbolic Systems," Mississippi State Univ., MSSU-EIRS-ASE-90-3, Mississippi State, MS, Oct. 1990.

[33]Janus, J. M., "The Development of a Three Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications," M.S. Thesis, Mississippi State Univ., Mississippi State, MS, Aug. 1984.

[34]Janus, J. M., and Whitfield, D. L., "Advanced 3-D Viscous SSME Turbine Rotor-Stator CFD Algorithms," NASA CR-178997, Sept. 1986.