

ADVANCED PARALLEL COMPUTING SYSTEMS APPLICATION TO THE V-22 TILTROTOR

J. C. Narramore

Bell Helicopter Textron, Inc., Fort Worth, Texas, USA

Joseph Vadyak and George Shrewsbury

Lockheed Martin Skunk Works, Palmdale, California, USA

Abstract

Bell Helicopter Textron, Inc. (Bell) in collaboration with Lockheed Martin Skunk Works (LMSW) is conducting research to develop a more accurate, faster, and less costly computational method for the detailed analysis and design of tiltrotor vehicles. The approach is to produce a computational method that is operational on massively parallel computer systems and to use this method to provide solutions for the V-22 tiltrotor that would be difficult if not impossible to obtain using conventional methods. The goals are being accomplished by porting the Euler/Navier-Stokes Three-Dimensional AeroElastic (ENS3DAE) technology, developed by Lockheed for the U.S. Air Force, to the latest high-performance parallel computers and refining this methodology to solve tiltrotor problems. This requires that moving grid capabilities be developed for this code. To date, a superscalable version of the code (ENS3DAE-MPP) has been developed that uses the Intel NX message passing library. The current superscalable version uses a MAP library developed at NASA Ames to permit compute node partitions to communicate during execution. Grid models for the airplane mode and helicopter mode configurations of the V-22 tiltrotor have been developed and initial solutions have been produced with the rotors moving. Correlation of the results of a model without rotors to both V-22 flight test data and wind tunnel test data was conducted to establish initial code validation. Assessments of the computational performance of the method have been generated and code scalability documented.

Notation

α	angle of attack (degrees)
ξ, η, ζ	body conforming coordinates of fixed grid computational space
ϕ, χ, η	body conforming coordinates of rotating grid computational space
ω	blade rotation speed (radians/sec)
BL	distance measured from centerline plane
C_p	pressure coefficient
i, j, k	grid point counters
M	Mach number

Introduction

Computational methods, in recent years, have demonstrated their ability to determine aerodynamic flow

characteristics of complicated configurations. Researchers have begun to apply these CFD methods to the investigation of the complex flow about tiltrotor vehicles. Rotor-alone evaluations have been performed by Narramore (Ref. 1) and Yamauchi (Refs. 2 and 3), confirming that highly twisted rotors can produce higher lift than would be predicted using previous methods. Wing-body computations have been performed by McVeigh (Ref. 4) and Tai (Ref. 5) to investigate the flow characteristics of the tiltrotor at high angles of attack in airplane mode and provide insight into buffet. In addition, Meakin (Refs. 6 and 7) has produced solutions in which rotating blades are combined with the wings and bodies to simulate the unsteady flows that result about the complete tiltrotor vehicle.

All of these methods are moving toward more complex and larger computational simulations in order to capture the critical flow phenomena about the tiltrotor vehicle. Therefore, the computer costs for these computations can be staggering. It is imperative that methods be developed that can produce solutions for large complex problems but which foster lower computing costs and shorter turnaround times. One approach to reduce costs and shorten run times is to distribute computations simultaneously across many low-cost processors.

This paper will describe the development of a general three-dimensional (3-D) multiple grid zone Navier-Stokes flowfield simulation program (ENS3DAE-MPP) designed for efficient execution on massively parallel processor (MPP) computers, and the subsequent application of this method to the prediction of the steady and unsteady viscous flowfields about the V-22 Osprey tiltrotor vehicle. Scaling and timing studies that show the benefits achieved will be discussed.

Objective

The primary objective of this effort is to demonstrate the advancements and improvements provided by advanced parallel computers applied to tiltrotor problems. A reduction in design cycle time and associated costs and an increase in accuracy for tiltrotor design problems is desired. Two tiltrotor problems selected were (1) the forward flight regime, where the rotor influenced aerodynamic flow over the fuselage and wing is difficult to analyze using conventional methods, and (2) the hover condition, where a more accurate determination of the

download associated with the rotor downwash impinging on the wing and fuselage is needed. Research has focused on development of a 3-D Navier-Stokes code for MPP supercomputers. The expected results, due to applying the advanced parallel computers to these problems, are decreased computational costs, increased accuracy of the computational methods, and increased speed of computations.

Approach

The objectives are being accomplished by porting the Euler/Navier-Stokes Three-Dimensional AeroElastic (ENS3DAE) code developed at Lockheed to run on the latest high-performance parallel computers and updating this methodology to solve tiltrotor problems. Bell has used vector/parallel versions of this code successfully for several fuselage force and moment studies including a computation of drag increments due to changes in the cowl shape of a helicopter fuselage (Ref. 8) and a detailed evaluation of forces, moments, and pressure distributions for the M214ST helicopter fuselage (Ref. 9). Additional code validation is being conducted as part of this study by comparing V-22 flight test and wind tunnel test data to ENS3DAE-MPP results.

Concepts for developing grids and algorithm modifications to model the individual rotating blades and a fixed wing/fuselage/empennage were developed and have been implemented. The method is currently operational on the Intel Paragon massively parallel computer. Grid models for the Navier-Stokes code were generated using existing gridding techniques and represent the V-22 FSD configuration that is currently being flight tested at the Patuxent River Naval Air Station in airplane and helicopter modes.

Code timing and optimization studies are being performed on the massively parallel computers. These evaluations assess the computational performance in terms of wall-clock time per solution, execution speed, memory, cost per solution, and disk storage.

Scalability of the code is being demonstrated by determining how memory, wall-clock time, disk storage, and execution performance change as problem size and number of processors are increased.

Computational Grid Generation

The grid system for these models is based on a structured body-fitted 3-D curvilinear computational mesh. The mesh generation programs used rely on numerical grid generation techniques, which are based on solving a system of coupled elliptic or parabolic partial differential equations. Multi-component configurations for the V-22 cases are analyzed using a multi-block grid approach where the global computational grid comprises a series of

subgrids that are patched together along common interface boundaries. An automated batch grid generation method for complete conventional aircraft configurations is provided by using the Lockheed-developed Complete Aircraft Mesh Program (CAMP) (Ref. 10). CAMP constructs wing/fuselage/horizontal tail meshes using a block $H-H$ type grid topology. CAMP uses a combination of algebraic and elliptic/parabolic grid generation techniques. Computational grids for other vehicle components can be produced by additional grid generation computer programs. For example, the DGRID program is used to construct grids for $O-H$ grid topologies used to model the nacelles of the V-22. It uses elliptical grid generation methods. Combinations of these structured grid generation methods were employed to develop final global meshes for the V-22 tiltrotor vehicle for both air-plane and hover modes of operation. The two primary grid generation tools used were the CAMP program and the DGRID program. Other grid utilities were developed to allow for the generation of tiltrotor specific models. A full global grid is comprised of a number of stationary grid blocks that are held fixed to the airframe, and a series of rotor grid blocks that are fixed to the moving rotor/spinner assembly. The rotor grid blocks thereby move with respect to the stationary grid for the time-accurate rotation cases.

Surface coordinates of the V-22 Osprey were used to develop a CAMP input data set. The initial grid produced from this data consisted of a fuselage and wing configuration, but did not include the wingtip-mounted engine nacelles. This grid consisted of four zones, containing the upper wing, lower wing, upper fuselage, and lower fuselage regions, respectively.

To add the nacelles to the configuration, the upper and lower wing zones were subdivided and portions were removed to allow insertion of two polar coordinate block grids for the nacelle and outboard wing. This resulted in an additional upper and lower wing block of the same span as the nacelle/wing block hole, and two outer blocks that contained the upper and lower portions of the original wing blocks outboard of the nacelle/wing block. Grid points on the boundary of the resulting aperture were used as the specified outer boundary points for the program DGRID, which produces meshes of $O-H$ polar coordinate grid topologies. The V-22 nacelle surface coordinate data were used to define the inner boundary geometry for DGRID. DGRID solves a system of coupled elliptical partial differential equations to determine the interior field point distribution. The resulting grid topology is illustrated in Fig. 1. Nacelle grid zones (9 and 11) are polar grids around the nacelle and outboard wing. At the nacelle jet exit, two additional grid zones were constructed, which are constant-area polar coordinate grid blocks extending to the downstream boundary (Zones 10 and 12). These exhaust grids were also created using DGRID.

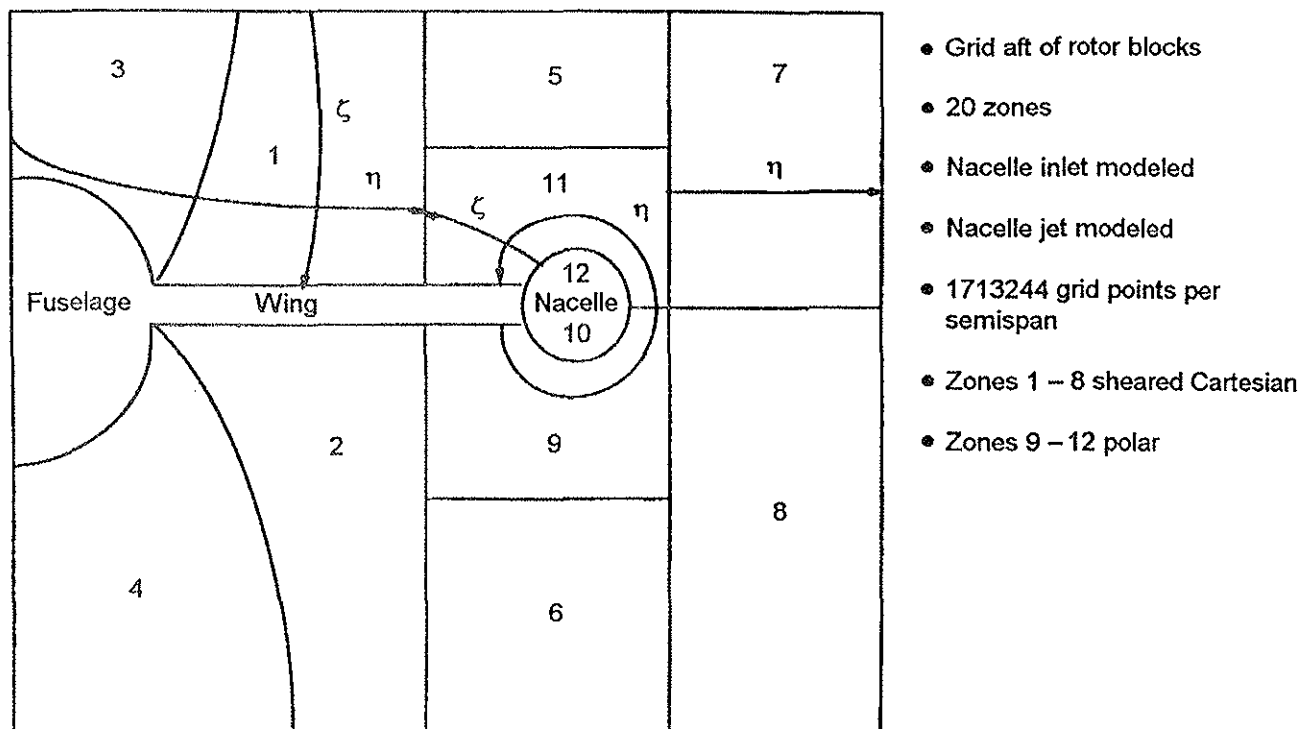


Fig. 1. Front view of zonal grid topology for V-22 vehicle with rotors in airplane mode (aft grid blocks shown).

Two additional grid generation utility codes were developed called CAMP_PROP_SPINNER and FINAL_GRID to allow rotors to be efficiently produced. The rotor grid generation procedure is initiated by first using CAMP to generate a two-zone block H-H mesh about a single rotor blade. This volume grid data is then entered into the CAMP_PROP_SPINNER code, which shears the blade volume grid (but the leaves the surface grid intact) to form a 120-deg sector, adjusts the inboard region to conform to the spinner geometry, and then reflects the single sector grid to form a complete rotor grid assembly consisting of six grid zones containing the three rotor blades and the spinner. An option also exists to reorder the six rotor grid zones into two zones, one on the forward side and one on the aft side of the blades.

At this stage, the rotor grid and the stationary wing/body/nacelle grid are input into the FINAL_GRID code which creates an aperture in the stationary grid for the rotor, generates a polar transition grid block between the rotor grid and the inner boundary of the aperture in the stationary grid, and then adjusts the rotor forward grid blocks to have their outer boundaries coincident with the outer boundary of the stationary grid. The complete semisp. global mesh is then output by FINAL_GRID ready for use by the flow solution code. The application of the CAMP_PROP_SPINNER and FINAL_GRID codes is very automated and requires a minimal amount of user effort. Figs. 2 and 3 depict the resulting grid topologies.

To construct a computational grid for the V-22 in a hover condition, the nacelle surface points were physically rotated about a line which corresponded to the conversion axis of the actual nacelle on the aircraft. To embed the rotated nacelle surface into the global airplane grid, a vertical cutout was removed from the CAMP-generated upper and lower wing grid blocks that extended from the upper grid boundary to the lower grid boundary. Again, DGRID was applied, where the outer surface of the polar coordinate grid was specified as the points corresponding to the vertical cutout, and the inner boundary was specified as the vertical nacelle surface plus the jet exit block. A jet exit grid block was constructed similar to the conventional flight mode grid, except that the centerline of this grid now ran vertically, and extended from the nacelle jet exit to the lower grid boundary surface.

Flow Solution Algorithm Implementation on Massively Parallel Processor Supercomputers and Coding and Communication Strategies

Background of Flow Analysis Methods

Once the computational grid is generated, the flowfield solution can be obtained using a version of the Lockheed-developed ENS3D (Euler/Navier-Stokes in 3-Dimensions) flowfield simulation program by solving either the full 3-D Reynolds-averaged Navier-Stokes equations, the thin-shear-layer Navier-Stokes equations, or the Euler equations. The thin-shear-layer Navier-Stokes equations

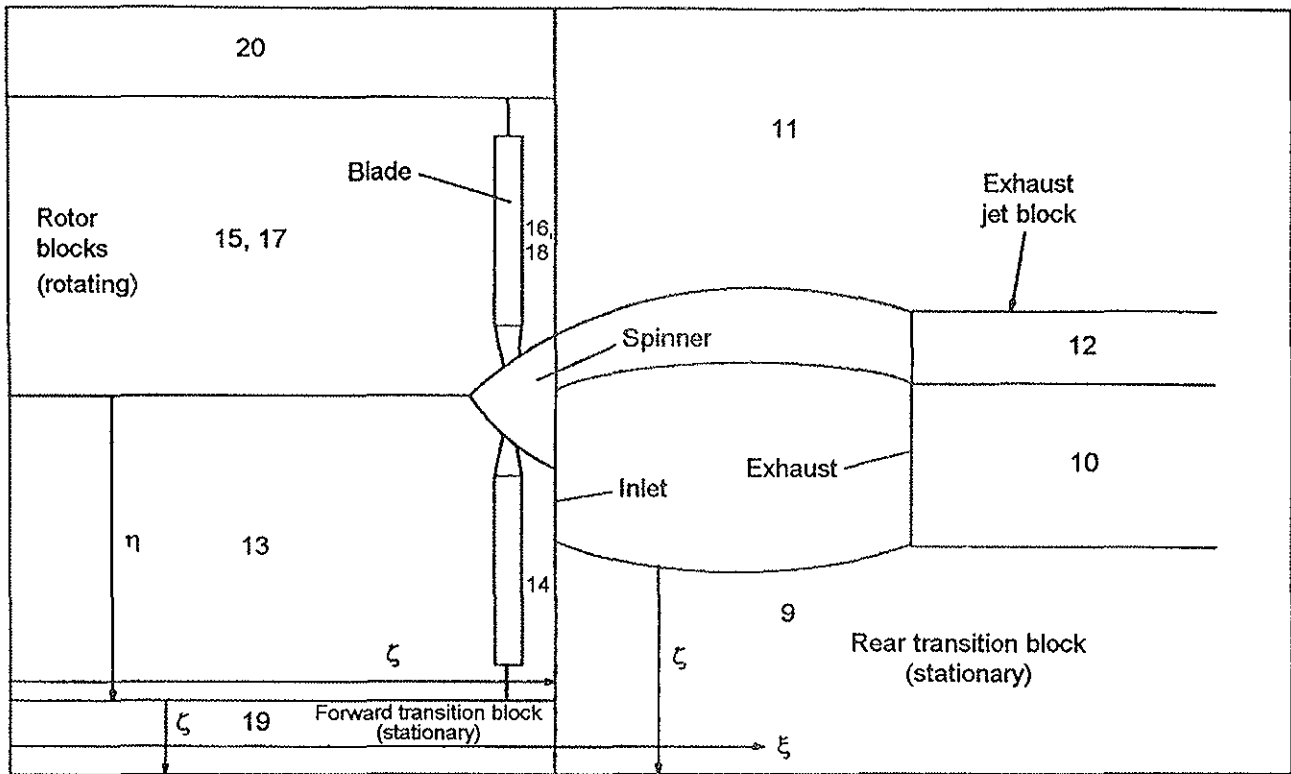
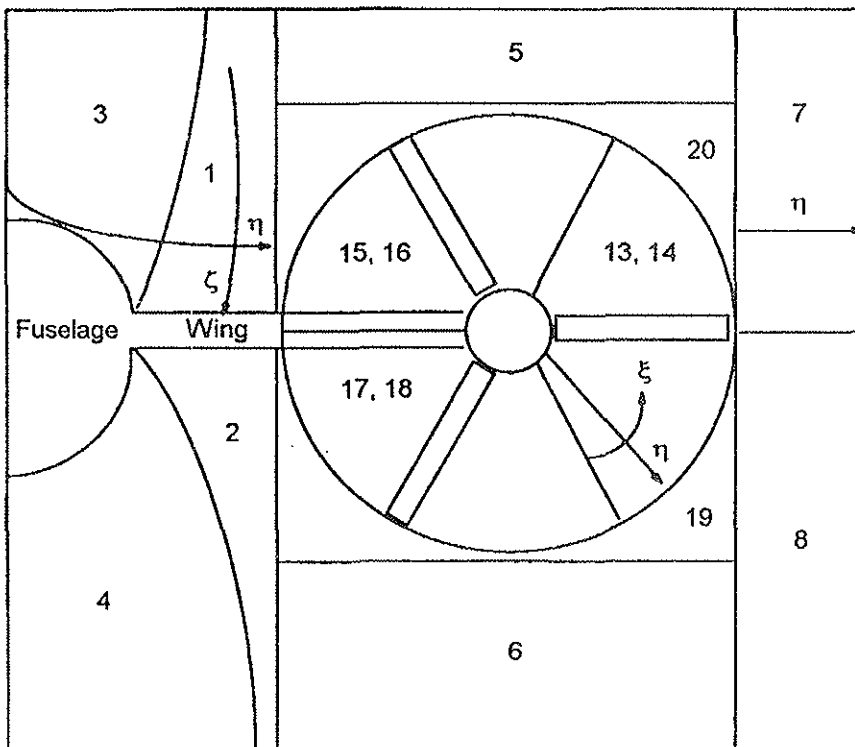


Fig. 2. Side view of zonal grid topology for V-22 vehicle with rotors in airplane mode.

retain the viscous and thermal diffusion terms only in the curvilinear coordinate normal to the body surface. The retained diffusion terms are generally the most dominant, however, and this approximation allows reduced computer execution times to be achieved without, in many cases, neglecting the most salient viscous flow features.

The Euler equations are applicable to inviscid flow modeling.

The governing equations are cast in strong conservation-law form to admit solutions in which shocks are captured. Second-order differencing is used in computing the metric



- Forward portion of grid
- 20 zones
- Zones 13 – 18 are for the rotor
- Zones 19 – 20 are Polar forward transition block grids

Fig. 3. Front view of zonal grid topology for V-22 vehicle with rotors in airplane mode (fore grid blocks shown).

parameters which map the physical domain to the computational domain. A time-marching, fully-implicit approximate factorization scheme (Ref. 11) is used for solution of the finite-difference equations. Either steady-state or time-accurate solutions can be obtained, with second-order spatial accuracy (or fourth-order accuracy for the right hand side) and first- or second-order temporal accuracy. The convective (inviscid) terms in the governing equations can be differenced using either central differencing or TVD upwind differencing using an extension to 3-D viscous flow of Harten's method (Ref. 12). The viscous diffusion terms employ central differencing. The algorithm includes the grid speed terms in the contravariant velocity calculations, thereby permitting the computation of unsteady flows with a time-varying grid. A solution adaptive grid method capability is present that can concentrate mesh points in high gradient regions as the solution progresses for steady flow simulations. This grid adaption scheme is based on sensing density gradients.

Although the interior points are updated implicitly, an explicit boundary condition treatment is employed which allows for the ready adaption of the program to new configurations. To aid convergence, nonreflecting subsonic outflow boundary conditions are employed along with a spatially varying time step for steady-flow solution cases. For the central difference option, the algorithm can use either a constant coefficient artificial dissipation model or a variable coefficient model where the coefficient's magnitude is based on the local pressure gradient. For the upwind differencing option, the algorithm is naturally dissipative. Laminar viscosity is computed for viscous cases using Sutherland's law. For turbulent viscous flows, the effective eddy viscosity can be computed using either the Baldwin-Lomax two-layer algebraic turbulence model (Ref. 13), the Johnson-King one-half equation model (Ref. 14), or the $k-\epsilon$ two-equation transport model (Ref. 15). The $k-\epsilon$ model requires the solution of two additional partial differential equations. For cases with separation, a streamwise eddy viscosity relaxation scheme can also be used in conjunction with the Baldwin-Lomax turbulence model. This accounts for turbulence history effects and improves the simulation of separated flowfields for this particular model.

Porting to the Intel Paragon GP Massively Parallel Processor

The initial target MPP platform used in this study was the 208 GP node Intel Paragon MPP, which was installed at NASA Ames. The flow simulation code was ported to this physically distributed memory platform. Parallelization on the MPP was obtained using extensive explicit message passing. Two main MPP versions were developed, namely the scalable version and the superscalable version. The initial versions of these codes use the Intel NX message-passing library. It was felt that use

of a message-passing approach along with the native message-passing library (NX) would produce the highest performance on this MPP system.

Parallel Coding Approach for the Intel Superscalable Version

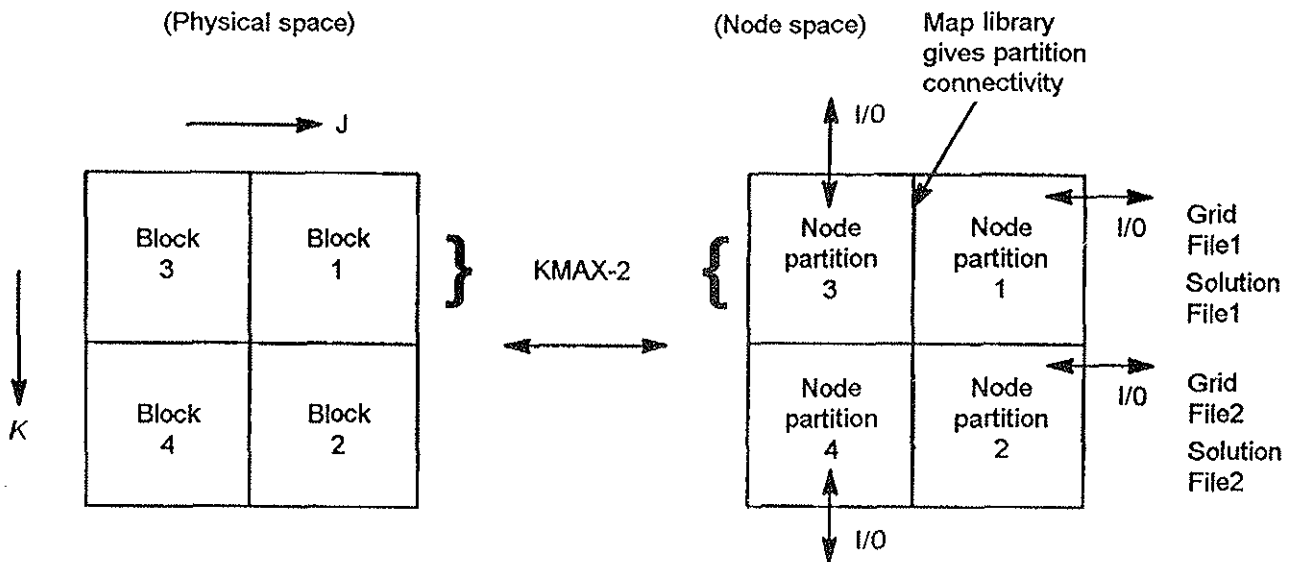
The superscalable version creates a pseudo compute node partition for each grid zone, and maps the entire global mesh topology onto the MPP, as is shown in Fig. 4. A MAP library (Ref. 16), developed at NASA Ames, is used to communicate between the pseudo partitions. This code version allows all the grid zones to be integrated in time concurrently. This allows for operation on hundreds or thousands of compute nodes and greatly reduces wall clock execution time and increases scalability. Because grid and solution data does not have to be rolled in and out of a single compute node partition as in the scalable version, the solution time per given grid zone is also reduced. To minimize the required input/output (I/O) time, nodes within each respective zone partition are used to perform I/O for that grid zone only. Optimum mapping occurs when one processor is used per K -plane per grid zone.

Interblock Communications

As mentioned previously, arbitrary block grid topologies can be analyzed with arbitrary curvilinear coordinate directions in each block. Blocks can be overlapped or have direct abutment for the stationary grid blocks within the global mesh. Generally, for the stationary grid blocks, a point-to-point interface arrangement is employed. Thus the neighboring points from an opposing grid block are known at the outset of the calculation and do not change with time. Indicial information only is needed in these cases to determine grid block connectivity. Explicit message passing is used to perform the block interface updates.

Two options have been programmed for the interblock communication at the time dependent rotor/stationary grid block interfaces. These include overlap with interpolation and direct abutment with interpolation and averaging. The overlap method is depicted in Fig. 5 and the abutment method is depicted in Fig. 6.

The overlap case is in effect a grid-embedding or "chimera" type approach, where the boundary grid point of the rotor block is embedded within the interior field of the adjacent stationary grid block at any point in time. The flow properties at the rotor block boundary point are determined by first searching the grid point neighbors in the stationary block for the nearest point and then using trivariate interpolation of the flow data about this base point in the stationary grid to update the properties. A similar procedure is used to ascertain the stationary block boundary grid point flow data at a given time step.



- Map each grid block to a separate node partition
- Concurrently solve all zones
- Optimum is one processor per K plane per block

Fig. 4. Superscalable MPP version operation.

For the abutment option, as shown in Fig. 6, the grid boundaries for the stationary and rotating grids are coincident. At a given time, a ray is passed through the center of rotation and the boundary grid point of the rotor block (Point C) to determine the nearest interior field grid point in the stationary block. This is used as the base point for interpolation to determine the flow properties at Point A in the stationary grid block. The interpolated property data at Point A and Point B (the first interior grid point in the rotor block along the ray passing through the points)

are averaged to obtain the properties at Point C. A similar procedure is used to obtain the stationary grid point values.

Rotor Grid Block Movement

All rotor grid blocks are moved using a solid body rotation. Therefore, there is no movement of points with respect to one another within a given block. The solution is initiated with the rotor movement off and the given free

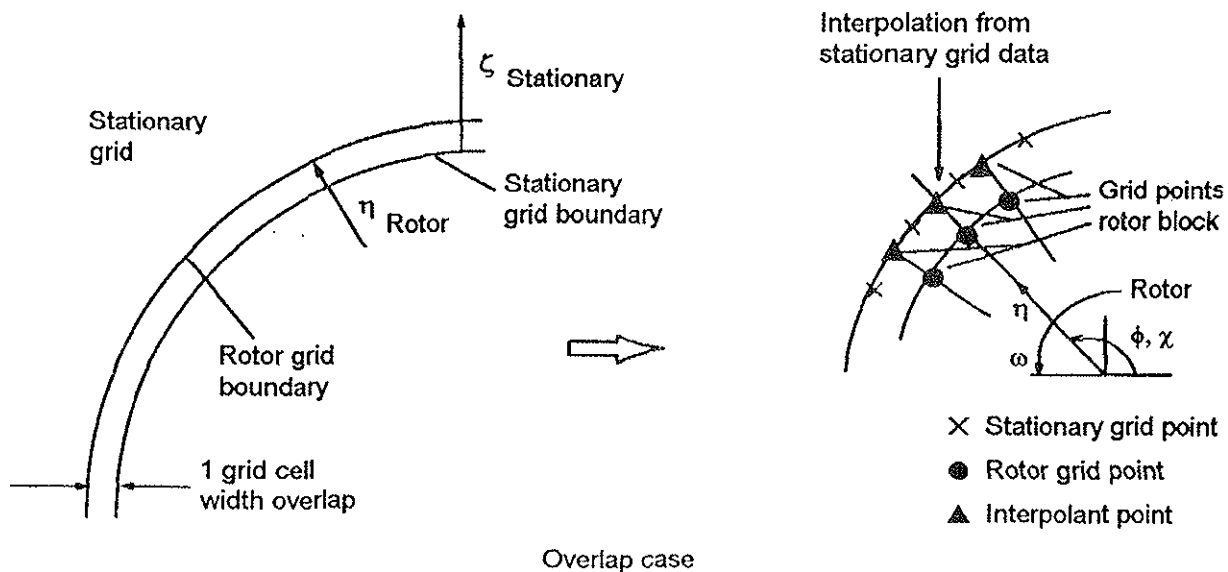


Fig. 5. Stationary / rotor grid interface with overlap option.

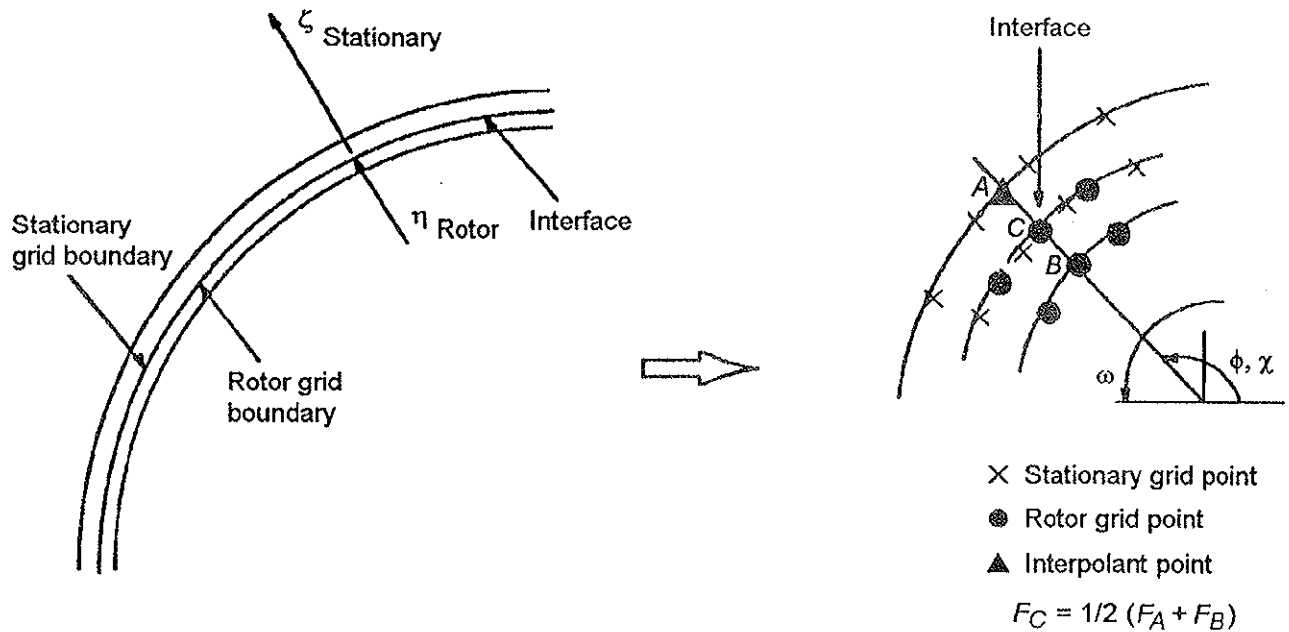


Fig. 6. Stationary / rotor grid interface with abutment option.

stream conditions. Once a steady state converged condition is reached, the rotor advance movement is started and the program is run in the time accurate mode.

Load Balancing Strategy for the Superscalable Version

Load balancing of the superscalable MPP version is achieved by determining the relative size of each grid block in the global mesh and the total available number of compute nodes and then assigning to each partition a number of compute nodes commensurate with number of mesh points in the given grid block. The number of processors assigned to a pseudo-partition is in effect determined by examining the cross plane interior grid dimensions of a zone and assigning compute nodes accordingly. This strategy, as will be shown, has worked well in actual practice. It is planned to investigate a dynamic load balancer also that reassigns nodes during the course of the calculation to optimize processor loading and minimize any idle processor time.

Results, Validation, and Performance Assessment

Results and Validation Studies

In order to determine the accuracy of this Navier-Stokes computational method for determining aerodynamic characteristics of the V-22 tiltrotor, the computational results are being compared to wind tunnel and flight test data. Initially, the validation was performed for a semispan global mesh computational model comprising twelve zones with common interfaces at the zonal boundaries without rotating blades, as is shown in Fig. 7. A fairing over the nacelle inlet was used in these simulations. The

grid topology is depicted in Fig. 8, which shows a front view on the grid interfaces. The mesh consists of a series of sheared Cartesian H-H zones with a polar grid surrounding the nacelle and embedded within the H-H grid. The H-H block grids (Zones 1-8) were generated using CAMP. The elliptical grid generation code DGRID was used for the polar grids (Zones 9-12) around the nacelle. A polar jet grid was embedded within the nacelle grid to simulate the engine exhaust region. Fig. 9 shows the resulting surface grid that represents the V-22 without rotating blades. The global mesh consisted of about 1.1 million grid points per semispan.

A comparison of the flight test measured nacelle loads for the V-22 is compared to results from this Navier-Stokes model and a panel method (VSAERO) in Fig. 10. It

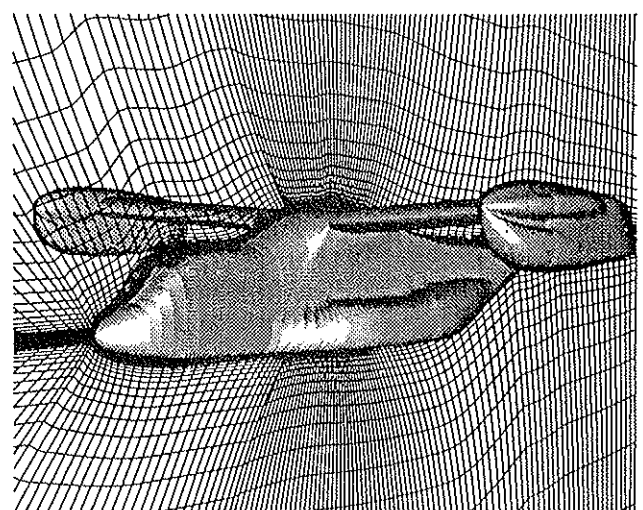


Fig. 7. Solid surface rendering for V-22 vehicle without rotors.

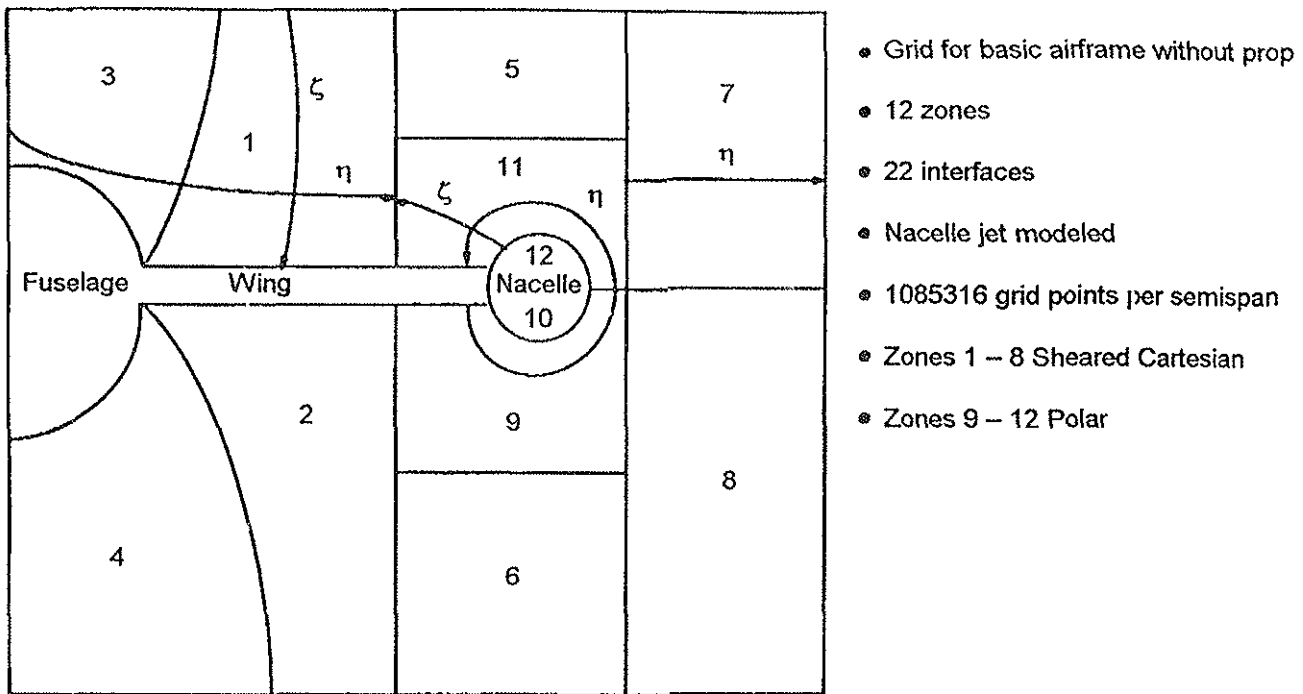


Fig. 8. Front view of zonal grid topology for V-22 vehicle without rotors.

shows that the correlation to the yawing moment of the nacelle is improved significantly when the Navier-Stokes method is used. The measured nacelle loads were determined from forces measured in the links that support the nacelle surface. The surface pressures from the Navier-Stokes solutions were integrated to obtain the computational forces and moments. Since the spinner is not a part of this structure, the spinner and inlet fairing were eliminated from the pressure integration. It should be noted that the panel method produced reasonable results for other critical components of force and moment, but for the yawing moment the results were poor, as is shown in Fig. 10. The Navier-Stokes results produced good results for all of the forces and moments. In Fig. 11, the wind tunnel pressure measurements on the on the aft door ramp are compared to Navier-Stokes results. For this case the

angle of attack is 5.17 deg and the Mach number is 0.21. The results show that the computed pressure distribution from the Navier-Stokes code compares very well with the wind tunnel test measurements. These results indicated that the ENS3DAE code can produce results that correlate well with flight test and wind tunnel test data.

Results are now being generated for which the rotating blade modeling is included in the massively parallel version of the Navier-Stokes code. Solutions from this procedure allow accurate computations of the influence of the rotor on the wing in airplane mode and detailed rotor download computations in helicopter mode. For the airplane mode configuration, 20 and 14 grid zone topologies are currently being used to model the V-22 aircraft and

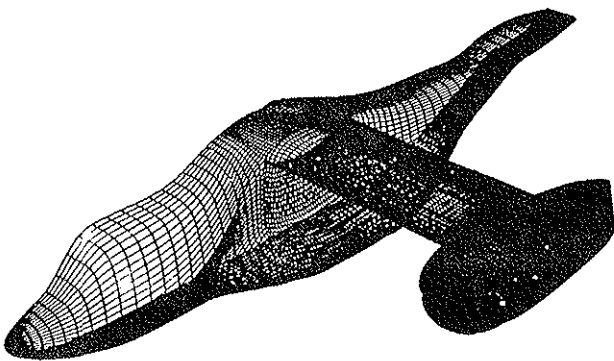


Fig. 9. Surface grid for V-22 vehicle without rotors.

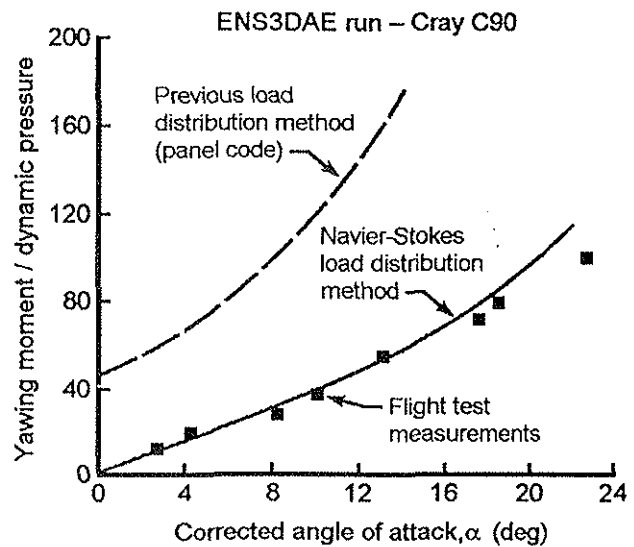


Fig. 10. Comparison of computed and measured nacelle yawing moment vs angle of attack.

V-22 aft door ramp pressures – ENS3DAE results vs. wind tunnel measurements

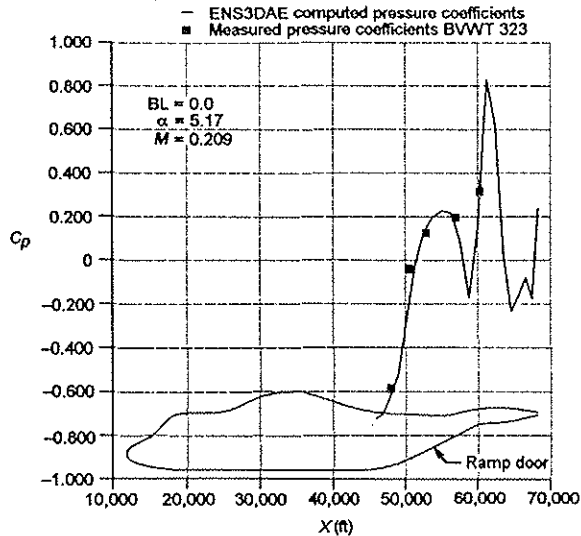


Fig. 11. Comparison of computed and measured fuselage bottom surface centerline pressure distributions.

blades. The final mesh contains 1,713,244 grid points per semispan. In this global grid, Zones 1 through 8 are sheared Cartesian $H-H$ block grids, Zones 9 through 12 are polar grids, Zones 13-18 are for the rotor, and Zones 19 and 20 are polar forward transition block grids.

Fig. 12 presents a solid surface rendering for the V-22 vehicle in airplane mode using the 14-zone global mesh. Fig. 13 shows the surface grid for the airframe and rotors and a portion of the field grid around the rotor. Fig. 14 presents the computed surface pressure distribution for a converged solution for the airplane mode configuration with the blades fixed. This solution is for Mach = 0.36 and 0 deg and serves as the starting condition for the rotating blade execution phase of the computation. The pressures are shown in a grayscale contour format where white denotes high pressure and dark denotes low pressure. All of the grid blocks that represent the rotor/spinner are moved using a solid body rotation. Figs. 15, 16, and 17 show the start of the blade revolution process. They show the Mach number in the grid plane that contains the rotor upper surface. In Fig. 15 the blades have rotated 15 deg, in Fig. 16 the blades have rotated 75 deg, and in Fig. 17 the rotation is about 180 deg. As can be seen from these figures, more blade time steps will be required to produce a converged rotating blade solution. It is anticipated that one or two blade revolutions will be required to produce convergence in the airplane mode.

Fig. 18 depicts a side view of the grid topology used to model the V-22 in hover using a 14-zone-per-semispan grid model. Here, grid Blocks 5 through 8 are sheared Cartesian topology, and Blocks 9, 10, and 11 are polar grids surrounding the vertically mounted nacelle or modeling the exhaust jet. Blocks 13 and 14 represent the rotor grid zones and were obtained by reordering the six individual zones for the rotor into two zones.

The helicopter mode solid surface rendering and surface grid are presented in Figs. 19 and 20. This model will be used to compute results for hover conditions. These results will be compared to wind tunnel test data.

Performance and Scalability Assessment and Estimates of Cost per Solution

Cost per Calculation Estimate versus Cray Y-MP.

Cost studies indicate that the MPP cost per calculation is about 1/4 that of vector/parallel machines using the speed ratio of the code on a Cray Y-MP/2 versus the scalable version on the Intel Paragon with 32 nodes and factoring in the leasing costs available to LMSW of both machines.

Intel Paragon Superscalable Version Scalability Studies (Ideal Load Balance Cases).

Code scalability studies using the superscalable version have been conducted at a series of Intel Paragon MPP sites. In these studies, the number of mesh points has been varied from about 0.5 million to 30 million points and the number of compute nodes has been varied from 30 to 1,024. Three-dimensional Navier-Stokes simulations of complete aircraft were conducted. Intel Paragon MPP machines at NASA Ames Research Center (208 GP nodes, 32MB/node), the Caltech Concurrent Supercomputing Facility (512 GP nodes, 32MB/node), and the Oak Ridge National Laboratory (1,024 MP nodes, 64MB/node) were used in this study.

Table 1 gives computation timing results for a global mesh consisting of five grid zones, each zone having 100 axial stations, 32 spanwise stations, and 32 normal stations, thereby giving a mesh of 512,000 grid points. Shown in this table are timing results for 30, 50, 75, and 150 compute nodes. Wall clock times to compute a global time step (GTS) and to compute the final solution are given. The time per GTS is defined as the wall clock time needed to advance the solution for all points in all grid zones with five local time steps per grid zone being

Table 1. Timing results for a model with 512,000 grid points.

Case	Grid size	Mesh points	Compute nodes	Time per GTS (sec)	Compute time (hr)	Ratio to	
						Actual	Theo
A-1	5x100x32x32	512,000	30	62.40	3.466	1.00	1.00
A-2	5x100x32x32	512,000	50	38.97	2.165	0.62	0.60
A-3	5x100x32x32	512,000	75	28.17	1.565	0.45	0.40
A-4	5x100x32x32	512,000	150	16.87	0.937	0.27	0.20

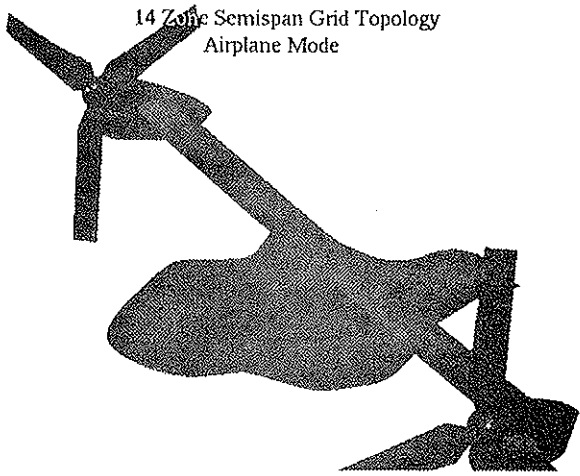


Fig. 12. Solid surface rendering for V-22 with rotors in airplane mode.

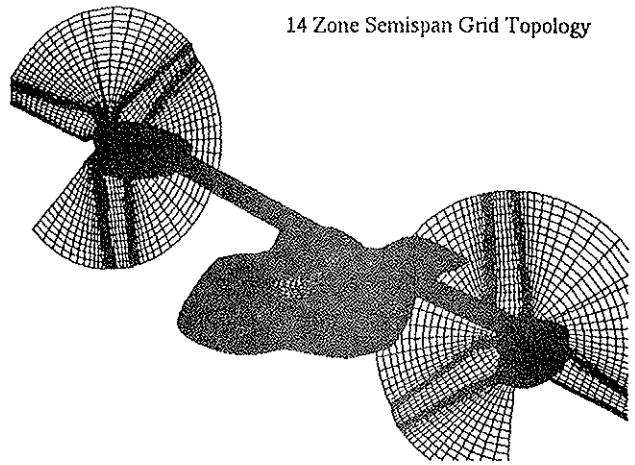


Fig. 13. Surface and partial field grid for V-22 vehicle with rotors in airplane mode.

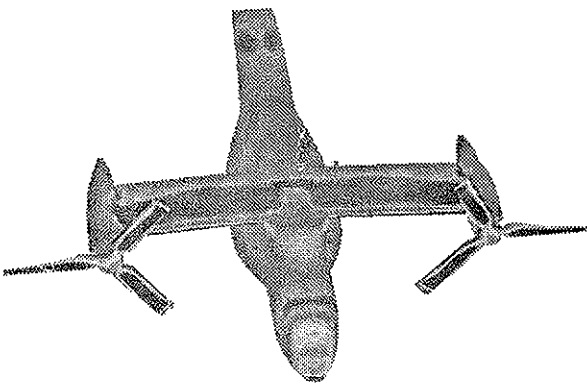


Fig. 14. Computed surface pressure distribution for V-22 vehicle with rotors in airplane mode for Mach 0.36 and $\alpha = 0$ deg.

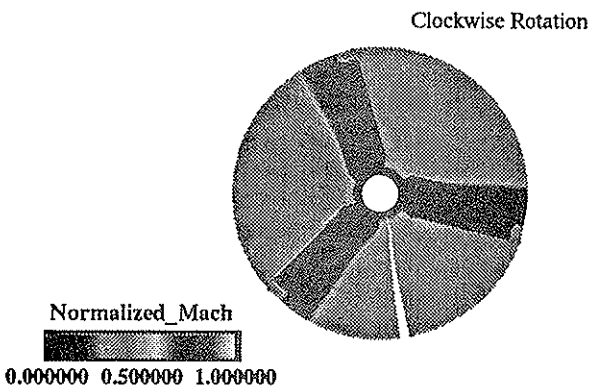


Fig. 15. Computed Mach contours for V-22 rotor in rotation startup in airplane mode ($\psi = 15$ deg).

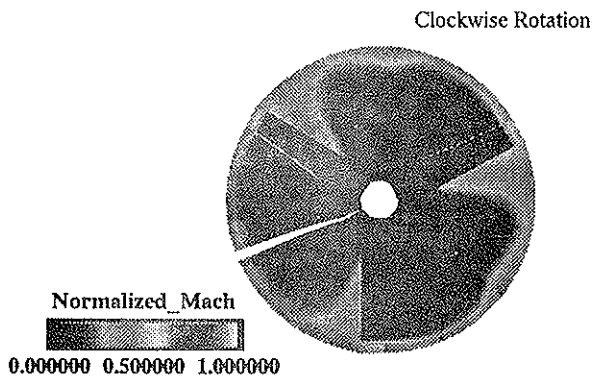


Fig. 16. Computed Mach contours for V-22 rotor in rotation startup in airplane mode ($\psi = 75$ deg).

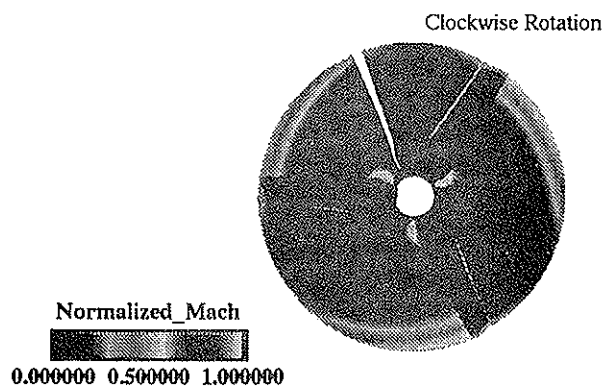


Fig. 17. Computed Mach contours for V-22 rotor in rotation startup in airplane mode ($\psi = 180$ deg).

performed. The total time is the wall clock time needed to advance the solution 200 global time steps or 1,000 effective cycles through the entire mesh. At each grid point, five solution variables are computed which are comprised of the density, three velocity components, and the internal energy. Table 1 shows good scalability using

the Case A-1 simulation on 30 compute nodes as the base. Actual and theoretical linear scalings are presented in the rightmost two columns.

Table 2 presents compute timings for a $8 \times 100 \times 62 \times 62$ grid with 3,075,200 total mesh points being executed on

Table 2. Timing results for a model with 3,075,200 grid points.

Case	Grid size	Mesh points	Compute nodes	Time per GTS	Compute time	Ratio to	
				(sec)	(hr)	Actual	Theo
B-1	8×100×62×62	3,075,200	120	98.62	5.478	1.00	1.00
B-2	8×100×62×62	3,075,200	160	72.91	4.050	0.74	0.75
B-3	8×100×62×62	3,075,200	240	51.32	2.851	0.52	0.50
B-4	8×100×62×62	3,075,200	480	27.15	1.508	0.27	0.25

Table 3. Effect on scalability of doubling the grid size and number of compute nodes.

Case	Grid size	Mesh points	Compute nodes	Time per GTS	Compute time	Ratio to	
				(sec)	3.091	Actual	Theo
C-1	4×100×102×102	4,161,600	400	42.74	2.374	1.00	1.00
C-2	8×100×102×102	8,323,200	800	44.21	2.456	1.03	1.00
C-3	8×100×102×102	8,323,200	400	78.55	4.363	1.84	2.00

120 through 480 compute nodes. With this larger grid size, nearly perfect linear scaling is observed with the wall clock time for a 480-node execution being 1.508 hours.

Table 3 presents Navier-Stokes computation times for two meshes. The first semispan mesh had four zones with each zone having 100 axial stations and cross plane grid dimensions of 102×102. This gives a total of 4,161,600 mesh points. The second mesh was used for full-span simulations and had eight grid zones each being 100×102×102. This mesh had 8,323,200 points and is exactly twice as large as the first mesh. Computation times are presented for 400 compute nodes for the first mesh, and for 800 and 400 nodes for the second mesh. Doubling the mesh size and doubling the number of

compute nodes produces nearly linear scaling. Doubling the mesh size and retaining the number of nodes at 400 results in super-linear scaling, in that a 1.84 factor increase in wall clock time was observed with theoretical linear scaling, indicating that a factor of 2.0 increase would have been required. The 8.3 million mesh point case required 2.456 wall clock hours for execution using 800 compute nodes.

Table 4 presents the computation times for three different sizes of mesh flow simulations. The first mesh consisted of eight grid zones with each zone having 80 axial stations and cross plane dimensions of 130×130, giving a total of 10,816,000 points. The second mesh is similar except having 160 axial stations per zone, thereby having

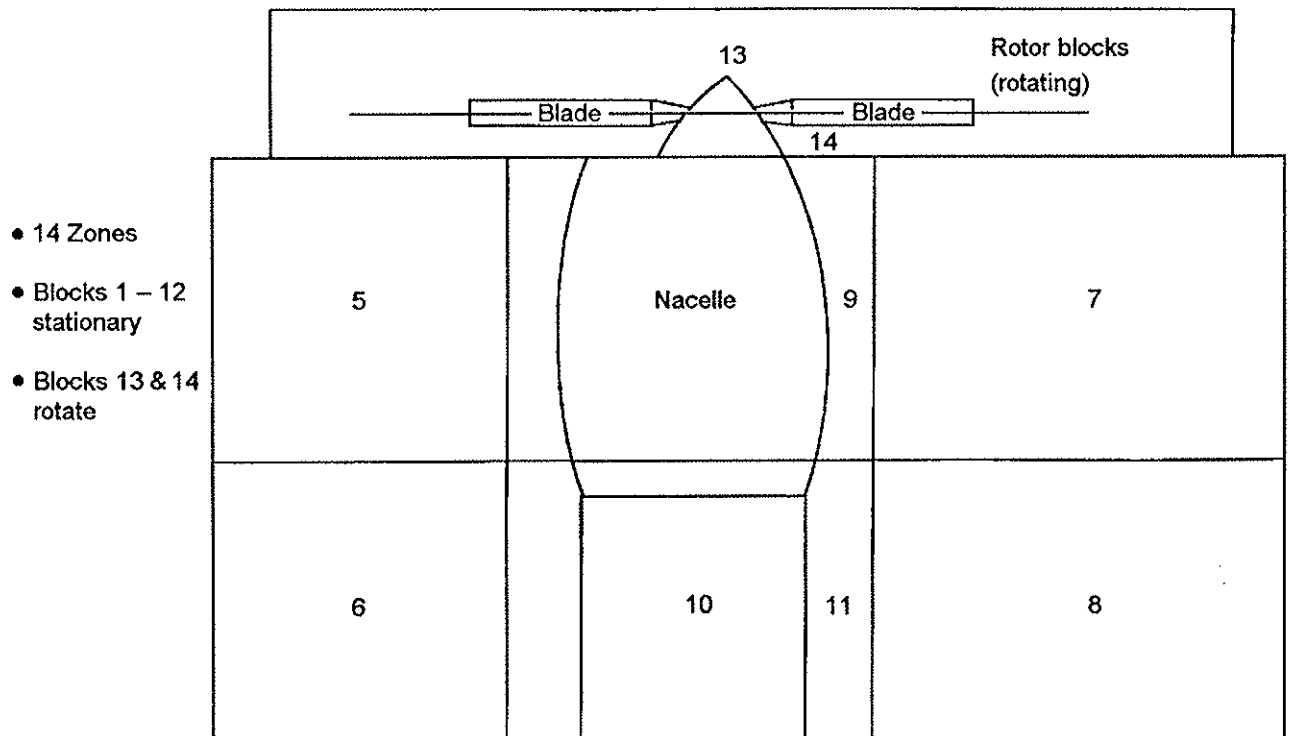


Fig. 18. Side view of zonal grid topology for V-22 vehicle in hover mode.

Table 4. Effect of grid size on the scalability with 1,024 compute nodes.

Case	Grid size	Mesh points	Compute nodes	Time per GTS	Compute time	Ratio to	
				(sec)	(hr)	Actual	Theo
D-1	8×80×130×130	10,816,000	1,024	55.64	3.091	1.00	1.00
D-2	8×160×130×130	21,632,000	1,024	86.71	4.817	1.56	2.00
D-3	8×220×130×130	29,744,000	1,024	124.97	6.940	2.25	2.75

Table 5. Input/output timing results for 8×100×64×64 grid case.

Case	Compute nodes	Time for output using PFS (sec)	Time for output using UFS (sec)
B-1	120	75.5	470.3
B-2	160	91.1	480.0
B-3	240	103.1	—
B-4	480	132.1	779.5

21,632,000 mesh points. This is exactly twice as large as the first mesh. The Case D-2 mesh solution requires the calculation of 5.33×10^8 final solution variables per global time step. Once again, a superlinear scaling was noted in comparing the wall clock times for Cases D-1 and D-2. The 21,632,000 mesh case required only 1.56 times the wall clock time of the 10,816,000 mesh case instead of a linear scaling factor of 2.0. The 10,816,000 mesh case required 3.091 wall clock hours for execution. The 21,632,000 mesh case required 4.817 wall clock hours. Case D-3 presents results for a mesh with $8 \times 220 \times 130 \times 130$ or 29,744,000 points. In this case, a scaling ratio of 2.25 was observed using Case D-1 as the base. Linear scaling gives a factor of 2.75. The superlinear scalings are attributed to the increase in the axial grid dimension parameter which is one of the two primary vector lengths in the original Cray coding. Since the Paragon i860 processor is a vector processor, increasing vector length increases the number of results per clock period and thereby increases the flop rate.

Intel Paragon I/O Parallel File System (PFS) Usage Comparison to the Unix File System. Tables 1 through 4 presented computation timing results excluding I/O times. The only significant I/O is performed at the beginning and end of the execution to load the grid data and to write the resulting flow solution data, respectively. Table 5 presents elapsed times for program output for the 3,075,200 mesh point case whose computation times are given in Table 2. Results are given in Table 5 using the Paragon UNIX File System (UFS) and the Paragon Parallel File System (PFS) for varying number of compute nodes. As discussed earlier, each grid zone employs its own I/O node set. For the 120 compute node case, the Parallel File System required only about 16% of the wall clock time required by the Unix File System to transfer the same amount of flow solution output data for this case.

Intel Paragon Superscalable Version Scalability Studies (Non-Ideal Load Balance Cases). The above scalability studies were conducted for ideal load balance cases in which each grid has the same number of points and hence all compute node partition sizes on the Intel Paragon were of identical size. To address scalability for the V-22 application additional studies were conducted. The 20-zone V-22 airplane mode case had a large variation in grid block size and hence compute node partition size. For the 20-zone case, the smallest grid zone had 6,080 mesh points and the largest had 215,208 mesh points. Table 6 shows the relative times per standard time step when this case was executed on 200 and 330 nodes on the GP node Paragon. Theoretical linear scaling based on the number of nodes is 0.606 and the observed scaling

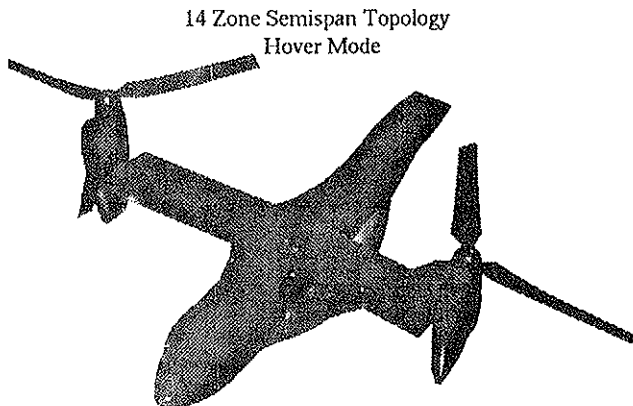


Fig. 19. Solid surface rendering for V-22 with rotors in hover mode.

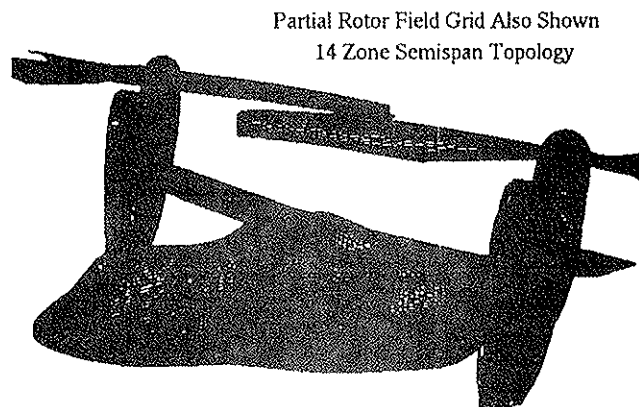


Fig. 20. Surface and partial field grid for V-22 vehicle with rotors in hover mode.

Table 6. Scalability of 20-zone V-22 airplane mode execution.

No. of nodes	Compute time per standard time step
200	10.2*
330	6.4*

• Intel Paragon with GP nodes

• With improved version, 25% further reduction in run time can be anticipated

• 20-zone grid has large variation in number of points per zone

- Minimum = 6,080
- Maximum = 215,208
- 867,000 points in rotor grid blocks

is 0.627. The load balancing strategy incorporated into the code thereby yields good scaling for non-ideal as well as ideal cases.

Conclusions

Based on the results and experiences from this project, it is concluded that advancements and improvements provided by advanced parallel computers applied to tiltrotor problems have been demonstrated. The initial cases for the two tiltrotor problems, including (1) the forward flight regime where aerodynamic flow over the fuselage and wing is influenced by the rotor, and (2) the hover condition where downwash associated with the rotor downwash impinging on the wing and fuselage, are currently executing on the Intel Paragon MPP. Using a 3-D Navier-Stokes code that was developed by Lockheed for the Air Force, many advantages of execution on a massively parallel processor (MPP) computer have been demonstrated. The results that have been produced to date, due to applying the advanced parallel computers to these problems, are: significantly increased computational efficiency, possible benefits in the accuracy of the computational methods, increased speed of computations, and lower cost of computations. An evaluation of the feasibility, practicality, and efficiency of integrating moving grid solutions into the parallel computing environment, the generation of results with improved accuracy, and the use of these methods to improve the design process for tiltrotors is being conducted.

Acknowledgments

Development of the MPP technology for application to the V-22 vehicle flowfield analysis is being sponsored by NASA Ames Research Center under Contract NAS2-14095. Intel Paragon computing time was supplied by NASA Ames, the Caltech Center for Advanced Computing Research (CACR), the Aeronautical Systems Center at Wright-Patterson AFB, Intel Scalable Systems Division,

and the Oak Ridge National Laboratory Center for Industrial Innovation. The authors wish to express their appreciation for this support.

References

1. Narramore, J. C., "Use of a Navier-Stokes Code to Predict Flow Phenomena Near Stall as Measured on a 0.658-scale V-22 Tiltrotor Blade," presented at the AIAA 20th Fluid Dynamics Conference, Buffalo, NY, June 12-14, 1989.
2. Yamauchi, G. K., and Johnson, W., "Navier-Stokes Calculations of a Highly Twisted Rotor Near Stall," presented at the American Helicopter Society Aeromechanics Specialists Conference, Fort Worth, TX, January 19-21, 1994.
3. Yamauchi, G. K., and Johnson, W., "Blade Twist Effects on Hovering Tilt Rotor Flow Fields," presented at the American Helicopter Society 52nd Annual Forum, Washington, D.C., June 4-6, 1996.
4. McVeigh, M. A., Lui, J., and Wood, T., "Aerodynamic Development of a Forebody Strake for the V-22 Osprey," presented at the American Helicopter Society 51st Annual Forum, Fort Worth, TX, May 9-11, 1995.
5. Tai, T. C., "Simulation and Analysis of V-22 Tiltrotor Aircraft Forward Flight Flowfield," presented at the AIAA 33rd Aerospace Sciences Meeting, Reno, NV, January 9-12, 1995.
6. Meakin, R. L., "Unsteady Simulation of the Viscous Flow About a V-22 Rotor and Wing in Hover," presented at the AIAA Atmospheric Flight Mechanics Conference, August, 1995.
7. Meakin, R. L., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," presented at the AIAA 11th Computational Fluid Dynamics Conference, July 6-9, 1993.
8. Narramore, J. C., "Navier-Stokes Computations of Fuselage Drag Increments," presented at the 1994 American Helicopter Society Aeromechanics Specialists Conference, San Francisco, California, January 19-21, 1994.
9. Narramore, J. C. and Brand, A. G., "Navier-Stokes Correlations to Fuselage Wind Tunnel Test Data," presented at the 48th Annual Forum of the American Helicopter Society, Washington, DC, June 3-5, 1992.
10. Schuster, D. M., and Smith, M. J., "Flight Loads Prediction Methods for Fighter Aircraft Vol. II - Complete Aircraft Mesh Program (CAMP)," WRDC/FIBRC,

Wright-Patterson Air Force Base, WRDC-TR-89-3104, Nov. 1990.

11. Vadyak, J., "Simulation of External Flowfields Using a Three-Dimensional Euler/Navier-Stokes Algorithm," AIAA Paper 87-0484, 1987.

12. Yee, H.C., and Harten, A., "Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates," *AIAA Journal*, Vol. 25, No. 2, 1987.

13. Baldwin, B.S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, 1978.

14. Johnson, D.A., and King, L.S., "A New Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers," *AIAA Journal*, Vol. 23, No. 11, 1985.

15. Gorski, I.J., "A New Near-Wall Formulation for the k- ϵ Equations of Turbulence," AIAA Paper 86-0556, 1986.

16. Fineberg, S.A., "The Design of a Flexible Group Mechanism for the Intel Paragon XP/S," Computer Sciences Corp., NAS, NASA Ames Research Center, Moffett Field, CA.