Large-Scale Domain Decomposition For A Scalable, Three-dimensional Brick Finite Element Based Rotor Dynamic Analysis

Anubhav Datta
ELORET Corporation
AFDD at Ames Research Center
Moffett Field, CA 94035

Wayne Johnson
Aeromechanics Branch
NASA Ames Research Center
Moffett Field, CA 94035

ABSTRACT

This paper implements and analyzes a dual-primal iterative substructuring method for the parallel and scalable solution of a three-dimensional finite element based dynamic analysis of helicopter rotor blades. Scalability and solution times are studied using two prototype problems – one for steady hover (symmetric) and one for transient forward flight (non-symmetric) – carried out on up to 128 processors. Several problem sizes of up to 0.48 million degrees of freedom are considered. A linear speed-up is observed with number of processors up to the point of substructure optimality. Substructure optimality and hence linear speed-up are shown to depend dramatically on the corner based global coarse problem selection. A minimal selection is implemented in this paper consisting only of corner nodes that lie on substructure vertices while the remaining corner nodes on substructure edges are treated as interface nodes with multiple dual variables. The key conclusion is that this minimal selection is key to extending linear speed-up to as high a processor number as possible, and minimizing the solution time for a fixed problem size. It is therefore an essential requirement for the efficient solution of a large-scale 3-D FEM problem.

INTRODUCTION

This paper describes progress in research towards a 3-dimensional (3-D) brick finite element model (FEM) based, parallel and scalable Computational Structural Dynamics (CSD) solver for helicopter rotors. It is envisioned to be a central component of a next generation, High Performance Computing (HPC) based, high fidelity rotorcraft analysis [1]. A research effort was initiated recently by the authors in Ref. [2] towards the development of such a solver.

The state-of-the-art in dynamic analysis of helicopter blades involves a variational-asymptotic reduction of the 3-D nonlinear elasticity problem into a 2-D linear cross-section analysis and a 1-D geometrically exact beam analysis – based on Berdichevsky [3] and pioneered by Hodges et al. [4]. Aeroelastic computations are performed on the beam, followed by a recovery of the 3-D stress field. The method is efficient and accurate – except near end-edges and discontinuities for which a 3-D analysis is still needed – as long as the cross-sectional characteristic dimensions are small compared to the wavelength of deformations along the beam. Modern hingeless and bearingless rotors contain 3-D flexible components near the hub that have short aspect ratios, open sections, and end constraints, and hence cannot be treated as beams. The critical couplings that determine blade dynamics are dominated by these components. Critical stresses often occur in these same components. Moreover, the treatment of blades, depending on their advanced geometry and material anisotropy, require continuous refinements to beam modeling and analysis to accommodate new physics. The objective of the present research is to develop a 3-D FEM based rotor dynamic analysis that can model generic 3-D components and dramatically increase the scope of analysis for modern rotors.

With the emergence of rotorcraft Computational Fluid Dynamics (CFD), physics-based models containing millions of grid points carry out Reynolds Averaged Navier-Stokes (RANS) computations on hundreds of cores, routinely, in a research environment for the rotor, and even for the entire helicopter. Applications today are focused on coupling CFD with relatively simple engineering-level structural models – carried out on

Presented at the American Helicopter Society Specialists' Conference on Aeromechanics, San Francisco, CA, January 20–22, 2010.

a single processor while the remaining processors lie idle. Assessments of the state-of-the-art in loads prediction, however, make it clear that the progress has mostly been in airloads, and much less in the accuracy of structural loads [5, 6]. The intent of this research is to explore the possibility of using 3-D FEM as the physics-based counterpart in the structures domain.

There is no question that such a capability will be powerful. First, it will enable the modeling of critical couplings that occur in hingeless and bearingless hubs with advanced flex structures. Second, it will enable the direct calculation of stresses in these critical load bearing components. Third, it will provide an equal fidelity of representation of the physics of structures and fluids, unlike the CFD/CSD simulations of today which are named so merely for the symmetry of terminology. And finally, even though this research is targeted towards HPC based analysis, it will always provide as a by-product a means (via static analysis) for extracting sectional properties with which efficient lower order beam analyses can be carried out when desired. The key question for such a capability is whether an efficient solution procedure can be found. The primary focus of the present research has therefore been on answering this key question directly.

The work in Ref. [2] demonstrated that a 3-D FEM based dynamic analysis of a rotor can indeed be carried out in a fully parallel and scalable manner. An advanced multi-level iterative substructuring method — the Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) method pioneered by Farhat et al. [7, 8] — was used to implement and study a parallel and scalable solution of a simple 3-D rotary wing structural dynamics prototype. It was concluded that for maximum efficiency, i.e. minimum solution time for a fixed problem size, the FETI-DP solver must be equipped with a minimal coarse problem. A coarse problem is a higher level finite element representation comprised of a selected subset of interface nodes of the partitioned substructures. A minimal set of coarse nodes is expected to maximize the linear speedup range to the highest number of processors for a fixed problem size — an essential requirement for the efficient solution of a large scale problem. The objective of this paper is to implement and study such a coarse problem.

Scope of present work

The main emphasis in this work is on the use of HPC as the enabler and driver of 3-D FEM based rotor structures. Advanced element modeling like locking-free elements, hierarchical elements, nonlinear constitutive laws, and composite material modeling are beyond the scope of this initial work. Realistic 3-D geometry definition and grid generation is not part of this initial endeavor. Simple grids are constructed that are adequate for research purposes. Partitioning is a part of this work due to its unique requirements. Most key elements of a comprehensive rotorcraft analysis are not considered at present: airloads, trim, extraction of periodic dynamics, and multibody dynamics, are all part of future work.

The paper is organized as follows. The first section describes briefly the formulation of the 3-D FEM analysis. The second section presents a brief description of the FETI-DP algorithm. The third section introduces the key components of a 3-D rotor analysis: geometry and grids, partitioning and corner selection, and the hover and forward flight prototypes. The fifth section presents scalability study on prototype problems on up to 128 processors. The last section documents the timings of large-scale problems of size up to 0.48 million degrees of freedom. The key conclusions and future research directions are summarized at the end.

3-D FEM FOR ROTORS

Formulation

The equations of motion are derived using generalized Hamilton's Principle governing the motion of a non-conservative system between times t_1 and t_2

$$\int_{t_1}^{t_2} (\delta U - \delta T - \delta W)\, dt = 0 \tag{1}$$

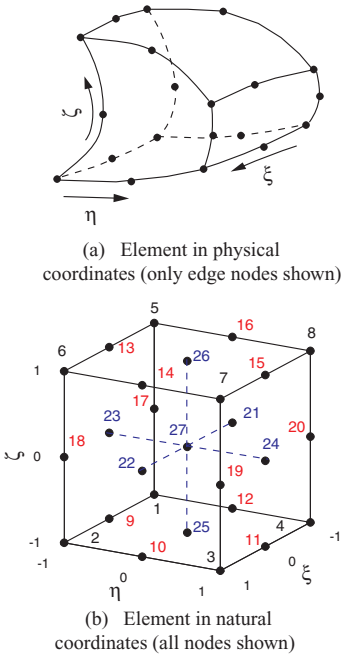
where δU, δT, and δW are variation in strain energy, variation in kinetic energy, and virtual work respectively. The expressions for each of these are derived in Ref. [2]. The formulation uses Green-Lagrange strains and second Piola-Kirchhoff stress measures for strain energy. The non-linear, geometrically exact implementation follows the standard Total Lagrangian based incremental approach [9, 10]. The stress-strain relationship is assumed to be linear.

The analysis of bending dominated problems involving thin structures using 3-D elements suffer from severe artificial stiffening known as locking as the element thickness tends to zero. A simple but effective way to prevent locking is to use higher-order elements containing sufficient number of internal nodes. An isoparametric, hexahedral, quadratic brick element is developed in this study with sufficient internal nodes (Fig. 1). It consists of 8 vertex nodes and 19 internal nodes – 12 edge nodes, 6 face nodes, and 1 volume node. More efficient lower-order locking-free brick elements, based on reduced-integration or Enhanced Assumed Strain methods are beyond the scope of this initial development.

Within isoparametric elements, geometry and displacement solution are both interpolated using the same shape functions. The shape functions are expressed in element natural coordinates ξ, η, and ζ, where $-1 \leq \xi, \eta, \zeta \leq 1$. We consider 2nd order Lagrange polynomials in each direction.

$$H_a(\xi, \eta, \zeta) = L_I^n(\xi)\ L_J^m(\eta)\ L_K^p(\zeta) \tag{2}$$

where H is a shape function and a its node point index. Here $n = m = p = 2$; and I, J, K are node numbers in



(a) Element in physical coordinates (only edge nodes shown)

(b) Element in natural coordinates (all nodes shown)

Figure 1: **27-node isoparametric, hexahedral brick element in curvilinear natural coordinates;** $4\times4\times4$ **Gauss integration points.**

each direction varying as $1, 2, 3$ respectively. Based on the local node ordering shown in Fig. 1(b), we have for example the shape function corresponding to node 11

$$H_{11} = L_2^2(\xi)\; L_3^2(\eta)\; L_1^2(\zeta) = \frac{1}{4}\,\eta\,\zeta(1-\xi^2)\,(\eta+1)\,(\zeta-1)$$

The construction of the finite element matrices then follow as given in Ref. [2].

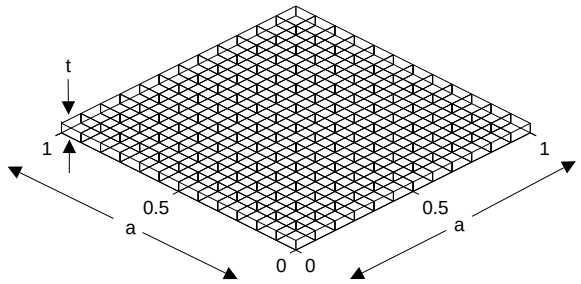


Figure 2: **A cantilevered square plate modeled using a** $(12\times12\times1)$ **grid of 3-D brick finite elements.**

Preliminary verification

A preliminary verification of the 3-D bricks is carried out by reproducing non-rotating thin plate and rotating beam frequencies.

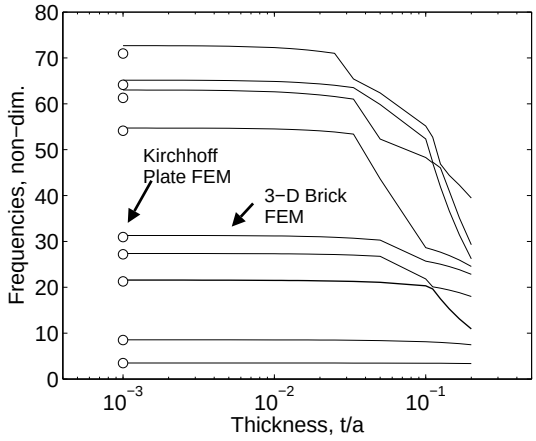


Figure 3: **Plate frequencies using 3-D bricks; symbols are plate FEM results; non-dim. w.r.t.** $\sqrt{D/\rho t a^4}; D = Et^3/12(1-\nu^2)$**, ρ: density, E: Young's modulus, ν: Poisson's ratio.**

Mode	Beam freqs.	3-D freqs.	Type
1	0.679	0.681	L 1
2	1.154	1.155	F 1
3	2.736	2.742	F 2/L 2
4	4.858	4.839	F 3/L 2
5	5.411	5.409	F 3/L 3
6	6.401	6.590	T 1
7	8.483	8.552	F 4/L 4
8	12.704	12.818	F 5/L 4
9	13.255	13.014	F 5/L 4
10	17.883	18.306	F 6/L 6

Table 1: **Blade frequencies for a soft in-plane hingeless rotor at rotation speed of** 27 **rad/s; collective** $20°$**, twist** $-15°$

The shear-locking free behavior of the brick elements is verified by re-producing classical Kirchhoff thin plate frequencies for a square cantilevered plate. The plate is modeled using a single layer of brick elements arranged as a 12×12 rectangular grid (Fig. 2). The variation in predicted frequencies with a gradual reduction in thickness from 20% to 1% (Fig. 3) confirms that the 3-D frequencies approach plate frequencies. The later are obtained using converged rectangular Kirchhoff plate elements that are validated with classical results [11].

Next, the bricks are verified using the rotating frequencies of a slender beam-like geometry of rectangular cross-section, high aspect ratio, and uniform chord. The beam has dimensions of $20c \times c \times c/4$ in span, chord, and thickness directions; a uniform twist of $-15°$ about mid-chord; and is set at a collective pitch angle of $20°$. The rotation axis is along mid-chord. The material modulii are $E = 8.2700 \times 10^7$ Pa and $G = 3.4458 \times 10^7$ Pa ($\nu = 0.2$), density is $\rho = 192.2208$ kg/m^3, and $c = 0.0864$ m. The discretization uses $16 \times 4 \times 2$ elements denoting

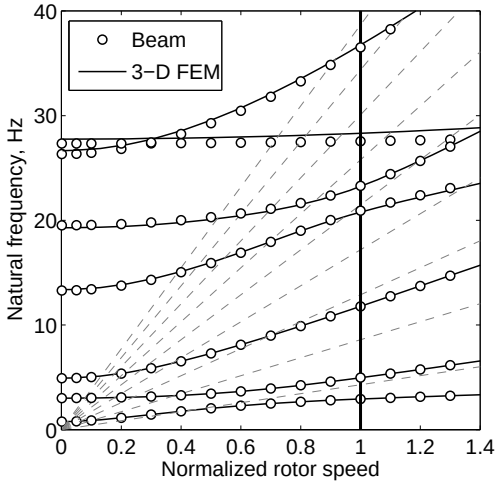


Figure 4: **Blade frequencies vs. normalized rotational speed for a soft in-plane hingeless rotor; collective** $20°$**, twist** $-15°$

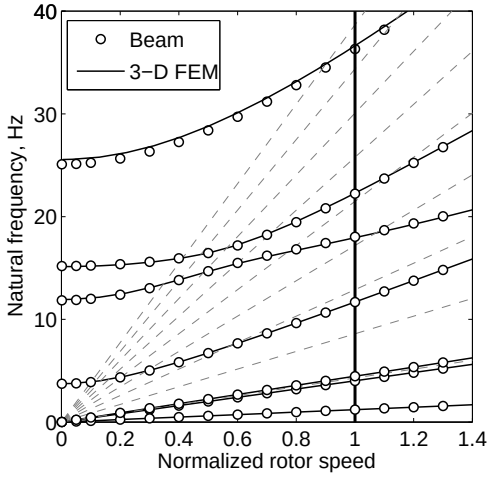


Figure 5: **Blade frequencies vs. normalized rotational speed for a fully articulated rotor; collective** $20°$**, twist** $-15°$

the number of bricks along span, chord, and thickness, respectively. The frequency plot for a hingeless blade is shown in Fig. 4, compared with converged second-order non-linear beam element results (40 elements). The 3-D boundary conditions are zero deflections at all root nodes. The rotor speed is normalized with respect to a baseline value of 27 rad/s. The first ten rotating frequencies at 27 rad/s are tabulated in Tab. 1. A noticeable discrepancy occurs in torsion frequency (3%) compared to its neighboring frequencies. Its exact nature and source is not studied here – it is desired that a cross sectional refinement study be carried out. The frequency plot for a fully articulated blade (5% hinge offset) is shown in Fig. 5. The articulation has zero hinge stiffness. The boundary condition is simply zero deflections at a single articulation node. Incorporating rotational hinge stiffness require special care in 3-D FEM. Unlike beams, there are no rotational states in bricks, and incorporating them require a special formulation that is currently being pursued as part of multibody dynamics research.

PARALLEL NEWTON-KRYLOV SOLVERS

Parallel Newton-Krylov solvers are developed for hover and transient forward flight. Each Newton iteration consists of a fully parallel linear solver based on iterative substructuring.

In iterative substructuring, the substructure interiors are solved using direct factorization. This operation is naturally parallel. The substructure interfaces are solved iteratively, using Krylov updates, the building blocks of which are constructed using fully parallel substructure-by-substructure operations. The building blocks are: (1) residual calculation, (2) preconditioning of the residual, and (3) a matrix-vector multiplication procedure. The Krylov updates – Conjugate Gradient (CG) updates for symmetric systems and Generalized Minimum Residual (GMRES) updates for non-symmetric systems – are constructed using these building blocks.

The goal of iterative substructuring is to construct the building blocks in a scalable manner. This means if the substructures have an average size H, and the finite element mesh within each substructure has an average size h, then the condition number of the preconditioned interface problem must not grow with the number of substructures as long as the mesh within each substructure is refined to keep H/h constant. A large problem will then converge with the same number of Krylov updates (iteration counts) as a small problem. The preconditioner is then called an 'optimal preconditioner' and the solver is said to exhibit 'optimal numerical scalability'.

The FETI-DP algorithm is such an iterative substructuring method. It can be constructed to guarantee optimal numerical scalability for problems governed by PDEs of up to 4th order and with heterogeneous properties.

The FETI-DP algorithm

In the FETI-DP algorithm, the substructure interface is sub-divided into two categories: a selected set of corner nodes and a remaining set of non-corner nodes. The corner nodes are used to formulate a primal interface problem. Hence they are also termed primal nodes. The non-corner nodes are used to formulate a dual interface problem. Hence they are also termed dual nodes. In the primal interface problem, the variables (called primal variables) are the original finite element degrees of freedom. In the dual interface problem, the variables (called dual variables) are a set of auxiliary variables, that are not a direct subset of the original finite element degrees of

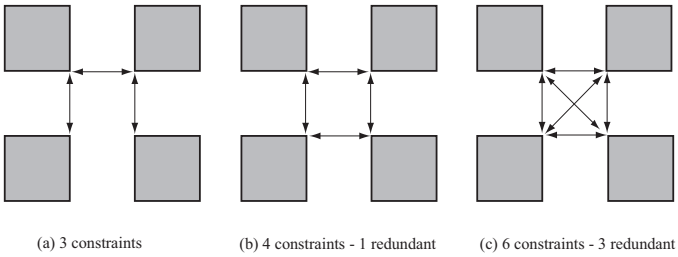


Figure 6: **Each figure is a top view of 4 neighboring substructures; at a dual node common to 4 substructures continuity can be enforced pairwise by (a) a minimum of 3 constraints across 3 pairs of substructures, (b) 4 constraints across 4 pairs and (c) a maximum of 6 constraints across 4 pairs.**

freedom. Each dual variable is used to enforce continuity of the original finite element degrees of freedom across two substructures. The two interface problems are coupled, and the building blocks of the coupled dual-primal interface problem can be constructed in a fully parallel manner requiring communication only between the dual nodes of neighboring substructure — as long as the primal nodes are available in all. The primal problem is therefore solved in every processor and require a global communication between all substructures. The primal nodes or corner nodes are the key to ensuring optimal numerical scalability. These form a coarse finite element representation of the problem, and ensure scalability by propagating local substructure information globally.

Each substructure interface node can be a face, edge, or a vertex node. A node that is common to two and only two substructures is a face node. A node that is common to at least three substructures is an edge node. Of these, those that occur at the end point of edges are vertex nodes. The edge and vertex nodes that are common to more than two substructures can be selected as corner nodes. This selection was used in our earlier work in Ref. [2]. This however leads to a large number of coarse nodes, and because the coarse problem require global communication, they limit the linear speed-up range for a given problem size. In this paper, only the vertex nodes are selected as corner nodes. This is a minimal selection as it excludes all edge nodes. An illustration is given later in the section on 'partitioning and corner selection'.

To implement the minimal coarse problem, the FETI-DP solver must now treat all substructure edge nodes that connect to four substructures as dual nodes, as in Refs. [12, 13]. Each of these dual nodes must then be equipped with sufficient dual variables to enforce continuity of finite element degrees of freedom across, not two, but four substructures. As illustrated in Fig. 6, a minimum of three dual variables per nodal degree of freedom is required for this purpose, each enforcing continuity across a single pair of substructures. However, a maximum of six can be used leading to a set of multiple redundant dual variables.

Parallel CG and GMRES Updates

In addition to the communication required by FETI-DP in constructing the building blocks, the CG and GMRES updates require additional processor synchronization points of their own. These must be minimized to prevent high communication costs diminishing scalability of the parallel implementation regardless of the numerical scalability of the underlying algorithm.

A Conjugate Gradient (CG) update requires three processor synchronization points – vector inner products that require global communication including a norm calculation to determine the stopping criteria. The total number can be reduced to just one using advanced norm estimation techniques [14, 15]. This has not been included at present. The requirement is more severe for the GMRES update and is more relevant to rotary wing structures due to its non-symmetric nature.

A Generalized Minimum Residual (GMRES) update incurs significantly more communication cost than a CG update. At the heart of a GMRES update is the Arnoldi algorithm. To solve $Ax = b$, it constructs m orthonormal basis vectors $V_m = [v_1, v_2, \ldots, v_m]$ spanning the m-dimensional Krylov subspace $K_m(A, r_0) = \text{span}(r_0, Ar_0, \ldots, A^{m-1}r_0)$, where $r_0 = b - Ax_0$ and x_0 is the current estimate of the solution, and a matrix $\bar{H}_m$ of size $(m+1) \times m$ the top $m \times m$ block of which is an upper Hessenberg matrix H_m. The construction of each vector requires orthogonalization with respect to every one of the previous. Traditionally, a Modified Gram-Schmidt procedure is preferred for this orthogonalization step because of its numerical stability over Classical Gram-Schmidt. However it requires as many as m synchronization points compared to only one in Classical Gram-Schmidt. In this study we implement a Reorthogonalized Classical Gram-Schmidt procedure that produces orthogonalization superior to Modified Gram-Schmidt while requiring only two synchronization points [16, 17].

COMPONENTS OF 3-D ROTOR ANALYSIS

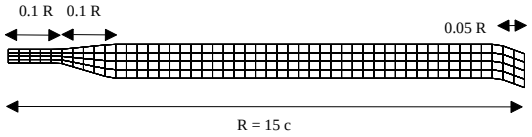


Figure 7: **Planform of a prototype hingeless rotor blade used in this study;** $c = 0.53$ **m.**

3-D geometry and grids

Geometry and grids are critical components of a 3-D rotor analysis, but are not the present focus of this work. It is assumed that suitable geometry and grid generators will be available to the solver from other sources. For

the purposes of solver development, a simple grid generator is developed that can discretize only one continuous structure, assumes that the cross-sectional discretization is same along span, and that all sections are solid. Within these assumptions, it is easy to accommodate arbitrary airfoil shapes, twist, planform, and advanced geometry tips.

Grid	$n_1 \times n_2 \times n_3$	Total DOFs
Small scale		
1	$96 \times 4 \times 2$	25,920
2	$48 \times 4 \times 4$	25,920
3	$64 \times 4 \times 4$	34,560
Large scale		
1	$32 \times 12 \times 12$	120,000
2	$48 \times 12 \times 12$	180,000
3	$64 \times 12 \times 12$	240,000
4	$128 \times 12 \times 12$	480,000

Table 2: **3-D FEM rotor grids**

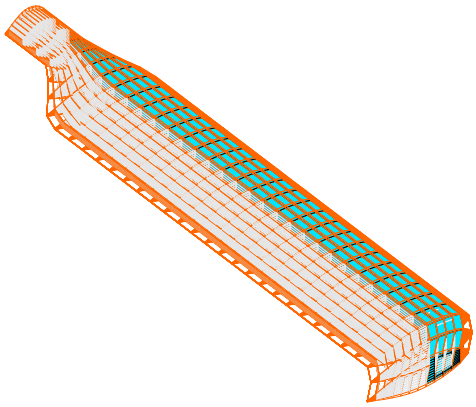


Figure 8: **A hingeless rotor blade prototype with $128 \times 12 \times 12$ elements (every 3 span stations shown); 0.48 M degrees of freedom**

The geometry considered is a hingeless rotor blade (Fig. 7) with a generic, symmetric airfoil with 5% thickness. The planform is generic with a sweep of 20° outboard from 95% span station. Each finite element can accommodate its own material model and ply direction but here we use simple isotropic properties: $E = 73$ GPa; $\nu = 0.3$; and $\rho = 2700$ kg/m^3. The rotational speed is a steady $\Omega = 27$ rad/s. With $c = 0.53$ m, these values generate typical stiffness and inertia of soft in-plane rotors. No attempt is made to place the sectional offsets at quarter-chord.

We consider three small scale problems for the scalability study and four relatively large scale problems for timing study. The problem sizes are listed in Tab. 2. n_1, n_2, and n_3 are numbers of elements along span, chord, and thickness. The largest problem size consists of 0.48 million (M) DOFs. For this size, the discretized blade and the cross section are shown in Figs 8 and 9 respectively.

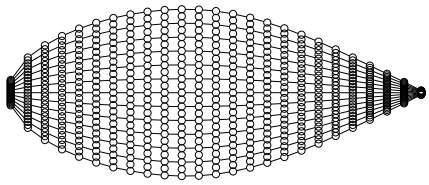


Figure 9: **Cross-section of prototype blade showing 12×12 bricks with 576 nodes; exaggerated vertical scale.**

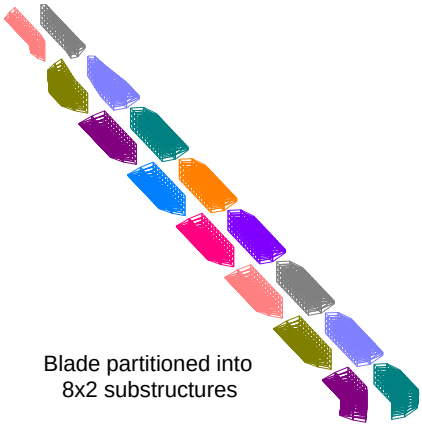


Figure 10: **3-D FEM of a hingeless rotor blade using isoparametric brick elements; blade partitioned into 8×2 substructures for illustration.**

Partitioning and corner selection

Figure 10 shows a rotor blade partitioned into 16 substructures using a 8×2 decomposition in span and chord wise directions. The partitioning requirements are unique in structures. The partitioner performs three tasks: (1) re-orders substructure nodes and element connectivity, (2) selects corner nodes, and (3) constructs substructure to substructure communication maps.

The node re-ordering brings the interior nodes first, followed by interface nodes, and then the boundary nodes. The interface nodes consist of face, edge, and vertex nodes. These are then separated into corner and non-corner nodes for treatment as primal and dual interface nodes respectively.

Selection of corner nodes is the most important requirement and must be performed in an intelligent manner. First, the selection must ensure null kernels in every

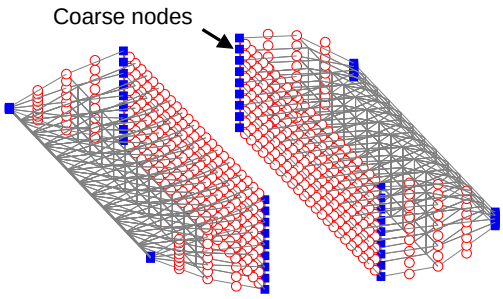


Figure 11: **A typical substructure showing baseline coarse problem selection; circles are dual interface nodes, squares are primal coarse nodes.**

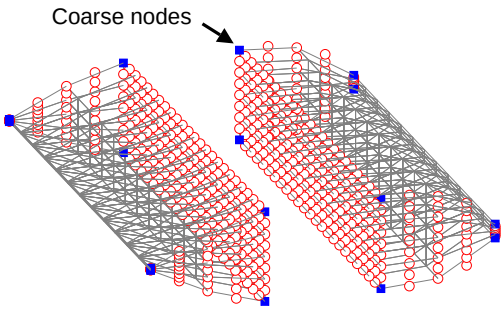


Figure 12: **A typical substructure showing minimal coarse problem selection; circles are dual interface nodes, squares are primal coarse nodes.**

substructure, i.e. constrain rigid body motion by ensuring that the non-corner restriction of the stiffness matrix is invertible. Second, it must be as small as possible, enough just to provide global error propagation but no larger. A selection containing all of the edge and vertex nodes common to more than two substructures (see Fig. 11) was used in our previous study [2] and is referred to in this paper as the baseline coarse problem. The selection studied in this paper contains only a subset of these corner nodes, and consists only of the vertex nodes that lie at the end of the edges (see Fig. 12). This is referred to as the minimal coarse problem. Its size is now independent of the cross sectional grid and is at the most 8 per substructure. Note that the vertices that occur at the boundaries of the structure must also be included, even though they are common to only two substructures, to satisfy the first criteria of null kernels. Otherwise, the substructures at the tip end will contain rigid body rotational modes making them non-invertible.

The substructure to substructure connectivity needs to be calculated only once. Each substructure creates a destination and a reception map. The former contains the substructures to which quantities are to be sent, and the corresponding destination node numbers. The latter contains the substructures from which quantities are to be received, and the corresponding recipient node numbers. The dual nodes that lie on the edges communicate with four neighboring substructures. The dual nodes that lie on the faces communicate only with two neighboring substructures.

Hover and forward flight prototypes

The hover prototype simply solves for steady blade response at a fixed collective with pressure airloads of 100 N/m^2 (418 lb/ft radial distribution) on the top surface. The airloads have the non-linear characteristics of a follower force. The non-linear solution procedure uses Newton-Raphson outer iterations. Within each iteration, the implicit FETI-DP inner solver uses CG updates. A CG update is adequate in ideal hover as the stiffness matrix is symmetric. The initial iterations converge the structural non-linearities associated with rotation. Once converged, the airloads are imposed. The virtual work during each airload iteration is calculated based on the previous iteration deformation state.

The transient forward flight prototype uses a Newmark scheme with a 5° azimuth step. The dynamic stiffness is now non-symmetric, therefore, the inner Krylov solver uses a GMRES update. For purposes of scalability study, the response for a single time step suffices, as the structure of the dynamic stiffness matrix remains same for all. We consider the following dimensions of Krylov subspace: $m = 30$, 40, and 50, deemed more than adequate for large scale problems. Note that increasing m improves efficiency (faster convergence) at the cost of reduced scalability (greater communication).

SCALABILITY OF 3-D ROTOR ANALYSIS

First, the study is conducted on a local unix cluster of 2.2 GHz dual core AMD Opteron processors. This to compare present results consistently with those reported earlier in Ref. [2]. Subsequently all computations are carried out on an Army DoD Supercomputing Resource Center (DSRC) cluster of 3.0 GHz dual core Intel Woodcrest processors. All times are wall clock times.

Consider the problem of size $48 \times 4 \times 4$, partitioned into $n_s = 8 \times 2 = 16$ substructures (as in Fig. 10). The FETI-DP/CG convergence for the baseline and minimal coarse problems are compared in Fig. 13. The minimal coarse problem in this comparison uses the fully redundant set of 6 dual variables per edge corner ($n_\lambda = 6$). This is the most efficient implementation, even though as shown in Fig. 14, using $n_\lambda = 4$ is equally efficient. Clearly, a minimal set of $n_\lambda = 3$ is not desirable. Henceforth, $n_\lambda = 4$ is used, unless otherwise mentioned.

Figure 13 shows that the minimal coarse problem increases the number of FETI-DP iterations required for convergence. This is expected due to the smaller coarse problem size, but the smaller coarse problem also reduces

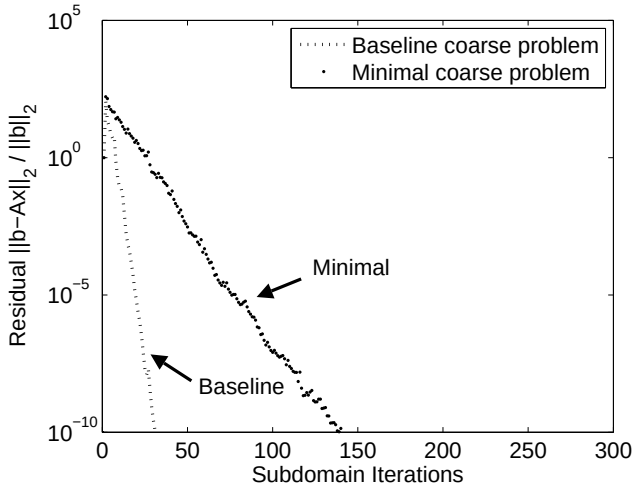


Figure 13: **Typical FETI-DP/CG convergence of baseline coarse problem vs. minimal coarse problem; later uses 6 dual variables per edge corner.**

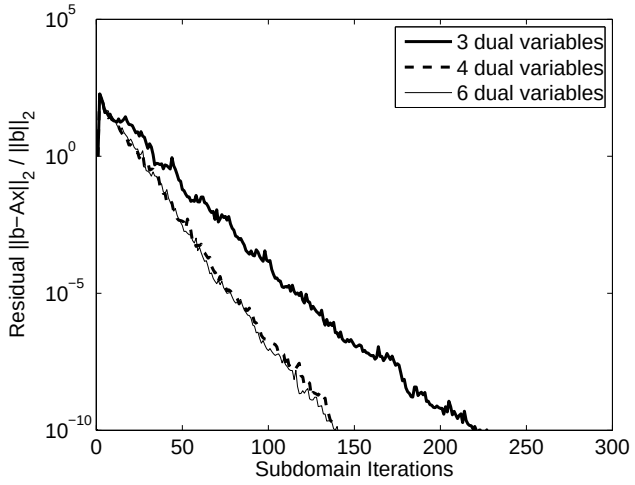


Figure 14: **FETI-DP/CG convergence of minimal coarse problem showing the effect of 3, 4, and 6 dual variables per edge corner.**

the time taken for each iteration, as a result of which the increase in total solver time is only marginal, as will be shown later. The main contribution of the smaller coarse problem, however, is to delay the growth in total solver time to a greater number of substructures.

The solver times for two problems of sizes $48 \times 4 \times 4$ and $96 \times 4 \times 2$ elements are shown in Figs. 15 and 16 respectively, comparing the baseline versus the minimal coarse problem implementations. It is clear that the optimal number of substructures — number of substructures for which the solver time is minimum — is extended by the minimal coarse problem. For the problem of size $48 \times 4 \times 4$ the baseline coarse node selection (as in Fig. 11) produces an optimality at 24 substructures whereas the minimal coarse node selection (as in Fig. 12) produces an optimality at 48 or more substructures. Similarly, for the problem of size $96 \times 4 \times 2$, the optimality is extended from 32 to 64 substructures.

n_s	FE	Sub. LU	Coarse problem	FETI	Solver total
8	198	453	**125**	517	1099
12	197	257	**101**	398	758
16	193	174	**99**	334	611
24	191	101	**149**	258	510
32	190	67	**279**	222	569
48	190	35	**866**	186	1088

Table 3: **Solver time (s) vs. number of substructures n_s with baseline coarse problem; single processor;** $48 \times 4 \times 4$ **elements.**

The reason behind this extension is clear from the detailed break-up of solver timings for the baseline and the minimal coarse problem implementations and are given in Tabs. 3 and 4. In the tables, 'FE' refers

n_s	FE	Sub. LU	Coarse problem	FETI	Solver total
8	198	496	**32**	601	1134
12	198	290	**23**	479	796
16	193	204	**19**	438	664
24	192	124	**15**	346	487
32	191	86	**14**	297	400
48	191	51	**20**	260	333
96	190	20	**94**	546	662

Table 4: **Solver time (s) vs. number of substructures n_s with minimal coarse problem; single processor;** $48 \times 4 \times 4$ **elements.**

to the time taken to construct the structural matrices. 'Solver total' refers to the total solver time. The two together constitute the total simulation time. 'Solver total' consists of three parts: (1) 'Substructure LU' time, which refers to the substructure factorization, (2) 'Coarse problem' time, which refers to the coarse problem factorization, and (3) the 'FETI-DP' time, refers to the Krylov solver time including residual, preconditioner, and matrix-vector multiplies. The tables show that the dramatic reduction in coarse problem time and the delay in its growth leads to a significantly higher substructure optimality for the same problem size. This has important ramifications for scalability and timings for the parallel implementation.

The parallel implementation solves each substructure on a separate processor. To calculate parallel speed-up, the parallel solver time is compared with the serial solver time with the same number of substructures as the parallel solver. This ensures that computations of the same complexity are compared and that the speed-up is not contaminated with the benefits of substructuring itself.

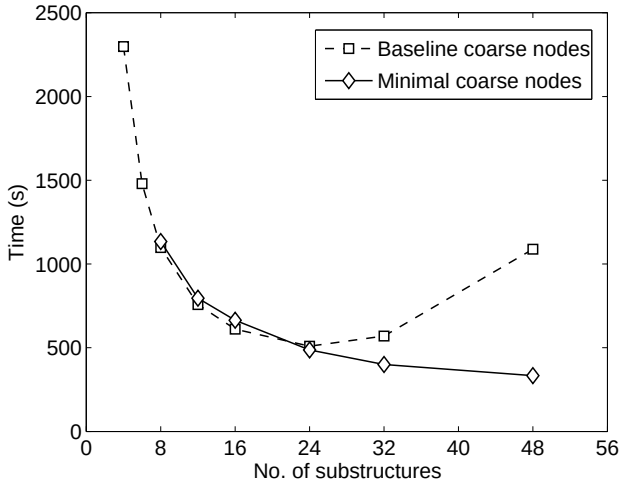


Figure 15: **Solver time (s) vs. number of substructures for calculations on a single processor;** $48 \times 4 \times 4$ **elements; hover.**

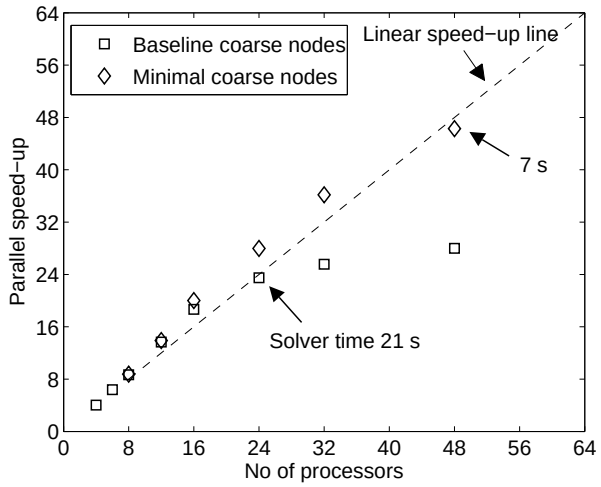


Figure 17: **Parallel speed-up for calculations on multiple processors; each substructure on each processor;** $48 \times 4 \times 4$ **elements; hover.**

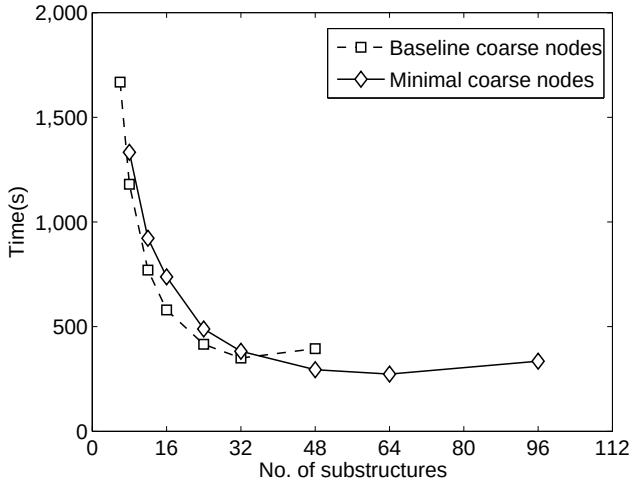


Figure 16: **Solver time (s) vs. number of substructures for calculations on a single processor;** $96 \times 4 \times 2$ **elements; hover.**

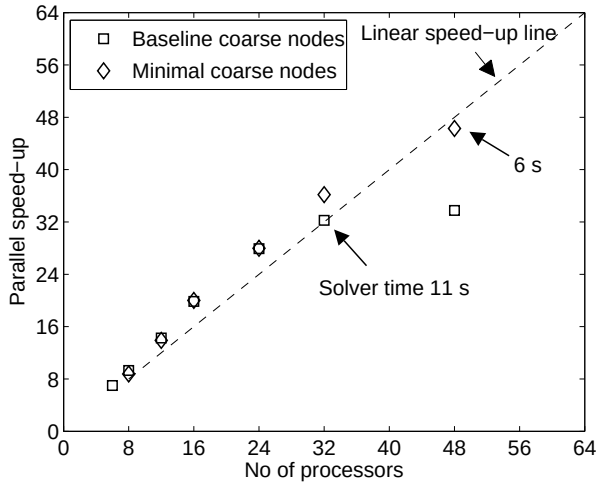


Figure 18: **Parallel speed-up for calculations on multiple processors; each substructure on each processor;** $96 \times 4 \times 4$ **elements; hover.**

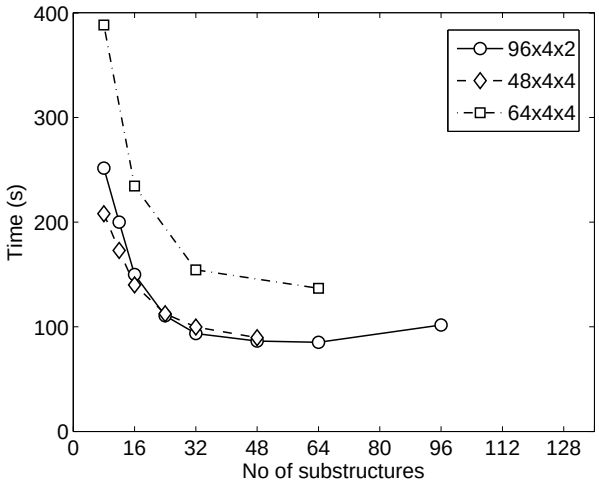


Figure 19: **Solver time (s) vs. number of substructures for calculations on a single processor; three problem sizes; hover.**

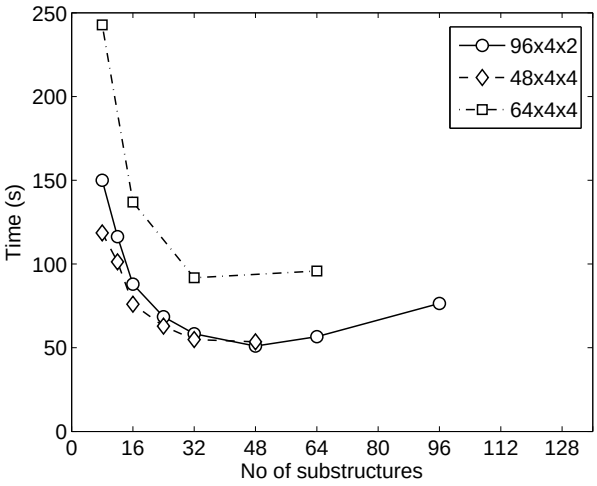


Figure 21: **Solver time (s) vs. number of substructures for calculations on a single processor; three problem sizes; forward flight.**

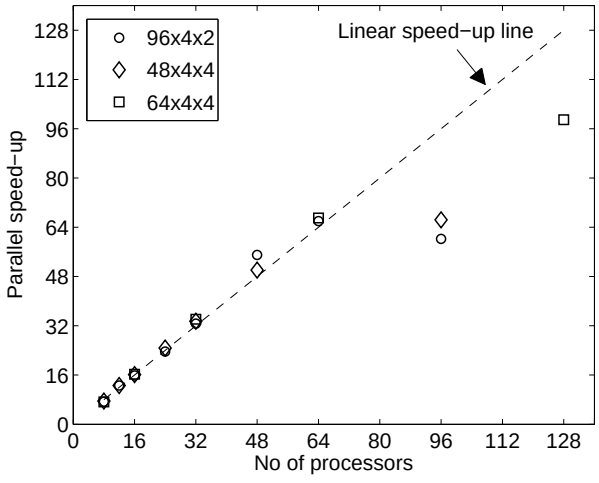


Figure 20: **Parallel speed-up for calculations on multiple processors; three problem sizes; each substructure on each processor; hover.**

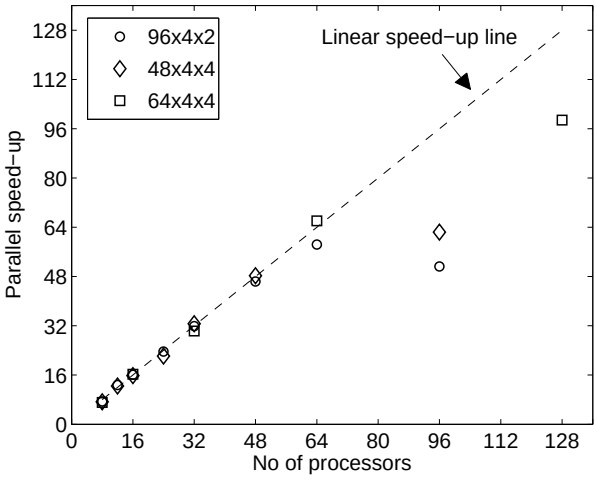


Figure 22: **Parallel speed-up for calculations on multiple processors; three problem sizes; each substructure on each processor; forward flight.**

The parallel speed-up for the two problems are shown in Figs. 17 and 18. In each figure, the speed-up obtained from the two coarse node selections are compared. It is clear that the minimal coarse node selection extends the linear speed-up range to a greater number of processors. Thus, for a given problem size, the minimal selection enables the fastest parallel solver time. From Fig. 17, the problem of size $48 \times 4 \times 4$ that could be solved in 21s using 24 processors, but no faster, can now be solved in 7s using 48 processors. The detailed break-up of the parallel solver times is given in Tab. 5.

n_p	FE	Sub. LU	Coarse problem	FETI	Solver total
8	24	66	4.18	67	137
12	16	26	1.97	34	62
16	12	13	1.19	21	35
24	8.2	5.5	0.68	11	18
32	6.1	2.9	0.54	7.6	11
48	4.2	1.2	0.69	5.2	7

Table 5: **Solver time (s) vs. number of processors n_p with minimal coarse problem;** $48 \times 4 \times 4$ **elements.**

Similarly, from Fig. 18, the problem of size $96 \times 4 \times 2$ that could be solved in 11s is now solved in 6s. However, for this problem the optimality is not yet reached with the available 48 processors. In order to study the full scalability range, all calculations are re-performed on the DSRC cluster, where more processors are available. Henceforth, all studies are conducted on this platform. Figures 19 and 20 show the single processor timings and parallel speed-up respectively of the same problems. An additional problem of size $64 \times 4 \times 4$ elements is considered which could be partitioned into 128 substructures and analyzed on 128 processors. Even though the actual timings are significantly superior on this platform (5–10 times faster), the conclusions on scalability remain the same. The two problems of sizes $96 \times 4 \times 2$ and $64 \times 4 \times 4$ elements that have optimality of 64 show linear speed-up up to 64 processors, the problem of size $48 \times 4 \times 4$ that has optimality of 48 shows linear speed-up up to 48 processors. The solver times for serial and parallel computations for the problem of size $64 \times 4 \times 4$ are documented in Tabs. 6 and 7 respectively.

n_p	FE	Sub. LU	Coarse problem	FETI	Solver total
8	30	79	9.7	298	388
16	24	30	5.2	199	234
32	23	12	3.0	139	154
64	21	5.3	6.8	124	136
128	21	2.7	65.5	349	418

Table 6: **Solver time (s) vs. number of substructures n_s with minimal coarse problem;** $64 \times 4 \times 4$ **elements.**

n_p	FE	Sub. LU	Coarse problem	FETI	Solver total
8	2.5	11.95	1.63	40	53.4
16	1.1	1.59	0.48	12	14.4
32	0.55	0.33	0.16	4	4.52
64	0.27	0.08	0.16	1.8	2.04
128	0.16	0.02	0.78	3.4	4.23

Table 7: **Solver time (s) vs. number of processors n_p with minimal coarse problem;** $64 \times 4 \times 4$ **elements.**

The conclusions drawn on substructure optimality and parallel speed-up using the FETI-DP/CG solver is carried over to the FETI-DP/GMRES solver. Figures 21 and 22 show the single processor timings and parallel speed-up respectively. For these results, the GMRES solver uses a restart parameter of $m = 30$, and a Classical Gram-Schmidt with Re-orthogonalization based Arnoldi algorithm (see Ref. [2]). The actual timings are lower because the convergence criteria is set to 10^{-8}, as compared to 10^{-12} for the CG, due to the oscillatory nature of residual convergence beyond this value.

LARGE SCALE PROBLEMS

The implicit parallel solvers developed in this study using the FETI-DP method of iterative substructuring can solve hover and forward flight response in a scalable manner. As an example, each Newton iteration of a 34,560 DOFs problem could be solved 64 times faster on 64 processors than one a single processor. Realistic blade models will contain millions of DOFs for which, not just scalability but actual solver timings are of equal importance. By extending substructure optimality and linear speed-up to as high a processor number as possible, the minimal coarse problem now enables the benchmarking of actual solver timings on a large scale problem size. In this section, a problem of size 0.48 M DOFs containing $128 \times 12 \times 12$ brick elements is considered. Only the size is realistic, the geometry is still simple without realistic 3-D hub structures or internal construction details of a typical blade.

The discretized blade and cross-sections were shown earlier in Figs. 8 and 9. The cross-section contains 12×12 second order elements with a total of 24×24 nodes. The blade is discretized into $64 \times 2 = 128$ substructures and analyzed on 128 processors. Even though the substructure optimality of this model is expected to be far greater than this number, the partitioner is limited at present to spanwise and chordwise partitions only, with no partitioning across thickness. At this level of decomposition, each substructure contains two layers of bricks each. The FETI convergence criteria for all cases are set to 10^{-6} for the preconditioned residual.

The solver times for a single Newton iteration is shown in Tab. 8. The FETI-DP/CG solver is used on the

symmetric stiffness matrix corresponding to hover. The FETI-DP/GMRES solver is used on the non-symmetric stiffness matrix corresponding to a single time step of implicit Newmark for transient forward flight. The forward flight cases converge faster because the mass matrix improves the condition number of the dynamic stiffness matrix leading to lesser number of iterations. The iteration count can be reduced further by using a greater value of restart parameter m. The consequent increase in communication, however, does not appear to incur a penalty as the solver time follows the same trend as iteration count. The $m = 30$ value is considered baseline in this study.

Solver type	FE	Sub. LU	Coarse problem	FETI	Solver total	Iter.
CG	2.9	35.4	3.14	220	258	509
GM30	2.9	35.9	3.15	142	180	325
GM40	2.9	35.5	3.14	135	173	309
GM50	2.9	35.4	3.14	130	168	296

Table 8: **Solver times (s) for FETI-DP/CG and FETI-DP/GMRES ($m = 30, 40, 50$) prototypes; analysis of 0.48 M model on 128 processors, each substructure on each processor.**

The number of dual variables per edge corner (n_λ) has an important effect on solver time. Four variables per edge corner ($n_\lambda = 4$) is considered baseline in this study. The variation from a minimum of 3 to a maximum of 6 is shown in Tab. 9. In general, increase in number of dual variables leads to faster convergence but at a greater communication cost. From Tab. 9 however communication cost is not a concern — iteration count and solver times both show the same trends. It is clear that more than 3 is desired and 4 is close to optimal – hence chosen as baseline. 5 is not preferred as one out of the two cross directions (see Fig. 6) must be picked arbitrarily.

Dual variable per edge corner	GM30	GM40	GM50
3	252 (489)	225 (428)	220 (416)
4	180 (325)	173 (309)	167 (296)
5	183 (327)	159 (274)	164 (285)
6	202 (366)	183 (327)	179 (314)

Table 9: **Solver times (s) and iteration count (in brackets) vs. number of dual variables per edge corner; FETI-DP/GMRES with $m = 30, 40, 50$, analysis of 0.48 M model on 128 processors, each substructure on each processor.**

Finally, the solver times for a range of problem sizes are shown in Tab. 10. The cross sectional discretization is kept constant for these problems, only the spanwise discretization is increased from 32 to 128 progressively. Each problem is partitioned into 32, 48, 64 and 128 substructures and analyzed on the same number of processors. The DOFs per substructure remains fixed at 3750. For optimal numerical scalability, all of these problems should demonstrate the same solution time. However they do not – as shown in Tab. 10. From 32 to 48 substructures there is a loss in numerical scalability, between 48 and 64 scalability is maintained, and from 64 to 128 there is further deterioration. Thus even though the solver demonstrates linear parallel speed-up the numerical scalability of the underlying FETI algorithm has deteriorated. This is a recognized artifact of 3-D brick elements the remedy for which is an edge based augmentation to the coarse problem (Ref. [8, 18]). Note that this augmentation places the coarse problem effectively in-between the baseline and minimal selections. This augmentation has not been implemented yet. It is expected to not only restore numerical scalability but also reduce solver times further.

Note that the difference between the solver times of the large scale problem and small scale problems is caused by the increase in DOFs per processor. Thus, a thickness wise partitioning that will provide less DOFs per processor, and the edge based augmentation to the coarse problem, together, are expected to bring down the large scale solution time drastically to the same levels as the small scale problems.

Processors n_p	DOFs	GM30	GM40	GM50
32	120,000	112	104	96
48	180,000	135	117	114
64	240,000	134	123	117
128	480,000	180	173	167

Table 10: **Solver times (s) vs. problem size with fixed size per processor (3750); FETI-DP/GMRES with $m = 30, n_\lambda = 4$; analysis of 0.48 M model on 128 processors, each substructure on each processor.**

CONCLUSIONS AND FUTURE WORK

The main objective of this paper was to demonstrate a parallel and scalable solution for a 3-D FEM based dynamic analysis of helicopter rotor blades. The prototype analysis was formulated using second order, isoparametric, hexahedral brick elements to discretize a rotor blade structure. A dual-primal iterative substructuring based implicit Krylov solver was developed for a fully parallel solution. The method was built upon the FETI-DP domain decomposition algorithm, and equipped with both CG and GMRES updates to account for the non-symmetric nature of the inertial terms. A detailed scalability study was carried out for both hover and transient forward flight. Based on this study, the following key conclusions are drawn.

1. A 3-D FEM based rotor dynamic analysis can be carried out in a fully parallel and scalable manner. Given a fixed problem size, there is always an optimal number of substructures into which it can be decomposed that requires the minimum solution time.

2. The analysis presented in this paper exhibits parallel scalability up to substructure optimality. That is, p-processors, with a separate substructure in each processor, can solve a given problem p-times faster compared to a single processor. Beyond substructure optimality, there is no reason to use more processors – unless a larger problem is attacked – in which case, linear speed-up is restored again up to the new optimality.

3. The drop in scalability beyond substructure optimality is due to two factors: the increasing substructure to substructure communication cost, and, the global coarse problem communication cost. The first penalty is minor, and can be reduced by minimizing the number of synchronization points of the Krylov update. This is more relevant to the GMRES updates. The second penalty is major and arises out of the coarse problem size.

4. The size of the coarse problem is the key driver for both scalability as well as solution time. The global communication required by the coarse problem determines scalability. The size of the coarse problem determines solution time. In order to ensure scalability while minimizing solution time, a minimal coarse problem should be selected.

5. A minimal selection consists only of corner nodes that lie on substructure vertices. The remaining corner nodes that lie on substructure edges must then be treated as interface nodes equipped with multiple dual variables.

6. For a minimal coarse problem, the edge corners are associated with a minimum of three to a maximum of six dual variables. In the later case, three are redundant. It is observed that four variables, with one redundant, is the most efficient option.

In summary, the rotor FEM analysis developed in this study solves one Newton iteration of a $34,560$ DOFs problem in 4.52 seconds on 32 processors and 2.04 seconds on 64 processors. A large-scale problem of size $480,000$ DOFs required 258 seconds in hover and around 180 seconds in forward flight for a single Newton iteration. However it was carried out only on 128 processors due to the limitations of the current partitioner. Improved partitioning using thickness-wise decomposition, and further refinement to the solver using edge based augmentation of the coarse problem is expected to provide solution times of the large-scale problem that are comparable to the smaller problem. It is clear however that HPC is both the key driver as well as the key enabler for a 3-D FEM based rotor dynamic analysis.

In way of conclusion, a brief summary of future research directions is provided below, that are considered important for the essential capabilities of a next-generation, rotary wing dynamic analysis. They are:

1. *Fundamental research*: Domain decomposition in combined space and time for scalable solution of periodic dynamics. Multibody dynamics within 3-D FEM analysis.

2. *Applied research*: Advanced finite elements, e.g., locking-free, hierchical, composite brick elements. Isogeometric elements. 3-D multidisciplinary interfaces for fluids, non-rotating structures, thermal stresses, and electro-mechanical actuators. Reduced order structural/FEM hybrid models.

3. *Application development*: 3-D geometry, grids, and partitioning. Geometry parameterization for optimization. Smart substructuring.

Acknowledgments

This research is conducted at the U. S. Army Aeroflightdynamics Directorate as part of the HPC Institute of Advanced Rotorcraft Modeling and Simulation (HI-ARMS) program supported by the U. S. DoD HPC Modernization Program Office.

REFERENCES

[1] Johnson, W. and Datta, A., "Requirements for Next Generation Comprehensive Analysis of Rotorcraft," American Helicopter Society Specialist's Conference on Aeromechanics, San Francisco, CA. Jan. 23-25, 2008.

[2] Datta, A. and Johnson, W., "Three-dimensional Finite Element Formulation and Scalable Domain Decomposition for High Fidelity Rotor Dynamic Analysis," American Helicopter Society Annual Forum 65, Grapevine, Texas, May 27–29, 2009.

[3] Berdichevsky, V. L., "Variational-Asymptotic Method of Constructing a Theory of Shells," *Journal of Applied Mathematics and Mechanics*, translated from *Prikladnaya Matematika i Mekhanika*, Vol. 43, (4), 1979, pp. 664–687.

[4] Hodges, D. H., *Nonlinear Composite Beam Theory*, AIAA, Reston, VA, 2006.

[5] Datta, A., Nixon, M. and Chopra, I., "Review of Rotor Loads Prediction with the Emergence of Rotorcraft CFD," *Journal of the American Helicopter Society*, Vol. 52, (4), October 2007, pp. 287–217.

[6] Yamauchi, G. K. and Young, L. A., Editors, "A Status of NASA Rotorcraft Research," NASA/TP-2009-215369, 2009.

[7] Farhat, C., Lesoinne, M., and Pierson, K., "A Scalable Dual-Primal Domain Decomposition Method," *Numerical Linear Algebra with Applications*, Vol. 7, (7–8), 2000, pp. 687–714.

[8] Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., and Rixen, D., "FETI-DP: A Dual-Primal Unified FETI Method - Part I: A Faster Alternative to the Two-level FETI Method," *International Journal of Numerical Methods in Engineering*, Vol. 50, pp. 1523–1544, 2001.

[9] Bathe, K., *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Inc., NJ, 1982.

[10] Zienkiewicz, O. C., and Taylor, R. L., *The Finite Element Method for Solid and Structural Mechanics*, Elsevier Butterworth-Heinemann, Oxford, UK, Sixth edition, 2006.

[11] Zienkiewicz, O. C., *The Finite Element Method in Structural and Continuum Mechanics*, McGraw-Hill, London, U.K., 1967.

[12] Rixen, D. J. and Farhat, C. "A Simple and Efficient Extension of a Class of Substructure Based Preconditioners to Heterogenous Structural Mechanics Problems," *International Journal for Numerical Methods in Engineering*, Vol. 44, (4), pp. 489–516, January 1999.

[13] Klawonn, A. and Widlund, O., "FETI and Neumann-Neumann Iterative Substructuring Methods: Connections and New Results," *Communications on Pure and Applied Mathematics*, Vol. 54, (1), November 2000, pp. 57–90.

[14] Meurant, G., "Multitasking the Conjugate Gradient method on the CRAY X-MP/48", *Parallel Computing*, Vol. 5, (3), July 1987, pp. 267–280.

[15] Saad, Y., "Krylov Subspace Methods on Supercomputers," *SIAM Journal on Scientific and Statistical Computing*, Vol. 10, (6), November 1989, pp. 1200–1232.

[16] Daniel, J. W, Gragg, W. B., Kaufman, L. and Stewart, G. W., "Reorthogonalization and Stable Algorithms for Updating the Gram-Schmidt QR Factorization," *Mathematics of Computation*, Vol. 30, (136), October 1976, pp. 772-795.

[17] Giraud, L., Langou, J. and Rozioznik, M., "The Loss of Orthogonality in the Gram-Schmidt Orthogonalization Process," *Computers and Mathematics with Applications*, Vol. 50, (7), October 2005, pp. 1069–1075.

[18] Klawonn, A. and Rheinback, O., "Robust FETI-DP Methods for Heterogeneous Three Dimensional Elasticity Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, (8), pp. 1400–1414, January 2007.