ELSEVIER

Comput. Methods Appl. Mech. Engrg. 190 (2001) 3189–3200

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Shear-slip mesh update in 3D computation of complex flow problems with rotating mechanical components

Marek Behr, Tayfun Tezduyar *

Mechanical Engineering and Materials Science, Rice University [1] *– MS 321, 6100 Main Street, Houston, TX 77005, USA*

Received 12 February 1999

Abstract

In this paper we present a 3D computational technique for simulation of complex, real-world flow problems with fast-rotating mechanical components. This technique is based on the Deformable-Spatial-Domain/Stabilized Space-Time (DSD/SST) formulation, Shear-Slip Mesh Update Method (SSMUM), and an efficient parallel implementation for distributed-memory parallel computing platforms. The DSD/SST formulation was developed earlier for flow problems with moving boundaries and interfaces, including flows with moving mechanical components. The DSD/SST formulation requires, as a companion method, an effective mesh update strategy, especially in complex flow problems. The SSMUM was developed to meet the mesh update requirements in simulation of flow problems with fast translations, and recently, with a new version of SSMUM, fast rotations. As an example of the class of challenging simulations that can be carried out by this technique, we present computation of flow around a helicopter with its rotor in motion.

1. Introduction

Flow problems with rotating mechanical components form a special subset of more general class of problems with moving boundaries and interfaces, such as free-surface and two-fluid flows, fluid–structure and fluid–object interactions, and flows with moving mechanical components. In recent past we developed a number computational methods to address this general class of problems. These methods include the Deformable-Spatial-Domain/Stabilized Space-Time (DSD/SST) formulation [1,2], which is an interface-tracking method; the Enhanced-Discretization Interface-Capturing Technique (EDICT) [3], which is particularly effective for free-surface and two-fluid flows; and advanced mesh update methods [4] designed in conjunction with the DSD/SST formulation. These methods were applied to a wide range of complex problems of engineering interest, such as free-surface flows in channels and past hydraulic structures such as dams, and liquid sloshing in moving containers, such as tanker trucks.

The mesh update methods developed in conjunction with the DSD/SST formulation and applied to many complex flow problems include special-purpose (problem-specific) and automatic (problem-independent) mesh moving techniques. In most cases, the automatic mesh moving needs to be combined with remeshing. Remeshing, defined as creating new inter-element connectivity and sometimes new set of nodes, could be a bottleneck, depending on factors such as how frequently and how locally remeshing is required and whether an automatic mesh generator needs to be called at each remesh. Limiting the frequency of

* Corresponding author. Tel.: +1-713-348-6051; fax: +1-713-348-5423.
E-mail address: tezduyar@rice.edu (T. Tezduyar).
[1] http://www.mems.rice.edu/TAFSM/

0045-7825/01/$ - see front matter
PII: S0045-7825(00)00388-1

remeshing would limit the applicability to problems with modest changes in the spatial domain. Generous use of remeshing would lift such limitations, but would result in excessive costs in terms of automatic mesh generation and projection of the solution from the old mesh to the new one. Furthermore, these projections introduce additional inaccuracies to the computation, which, depending on the frequency and scope of remeshing, could be excessive.

The Shear-Slip Mesh Update Method (SSMUM) [5,6] was designed to accommodate some specific classes of problems which could be handled with the DSD/SST formulation, without limitations on the frequency of remeshing. These problems are characterized by large, but linear or rotational, motion of some parts of the boundary with respect to other parts. This situation occurs when one object forming the computational boundary undergoes translation or rotation with respect to other boundaries. The underlying idea is to introduce a regular, typically "hand-made", layer of elements, restrict remeshing to that small layer, and control projection errors by limiting remeshing to changes in inter-element connectivity. With this approach, the cost of remeshing becomes negligible, and consequently, such remeshing can be carried out frequently, even at every time step.

We reported earlier examples of how SSMUM can function in 2D and 3D parallel computation of flow problems with rotating objects. These examples were 2D computation of flow past two counter-rotating squares [5,6] and 3D computation of flow past a rotating propeller [6]. Here, we continue our development of the SSMUM to the point of being applicable to a challenging problem: flow past a helicopter with its main rotor in motion. With this problem, we are able to demonstrate that, with proper enhancements, the SSMUM can be very effective in 3D computation of flow problems where two objects in close proximity (the fuselage and the rotor) are in relative motion.

A summary of the governing equations for the flow problem is given in Section 2, and the DSD/SST formulation is reviewed in Section 3. In Section 4, we provide a brief review of the SSMUM and a discussion of its implementation. We report the results from the helicopter simulation in Section 5, and present our concluding remarks in Section 6.

2. Governing equations

We will assume that the flow problem under consideration is governed by the time-dependent Navier–Stokes equations of incompressible flows. This is not necessarily true throughout all of the flight regime of the helicopter under consideration, but is nevertheless a suitable assumption for some portions of its flight envelope. Furthermore, the concepts described in this paper, including the DSD/SST formulation and SSMUM, are applicable also to the Navier–Stokes equations of compressible flows. Here we consider a viscous, incompressible fluid occupying at an instant $t \in (0,T)$ a bounded region $\Omega_t \subset \mathbb{R}^{n_{\mathrm{sd}}}$, with boundary Γ_t, where n_{sd} is the number of space dimensions. In the DSD/SST formulation, the spatial domain may change with respect to time, and the subscript t indicates such time-dependence. The symbols $\mathbf{u}(\mathbf{x},t)$ and $p(\mathbf{x},t)$ represent the velocity and pressure. The external forces (e.g., the gravity) are represented by $\mathbf{f}(\mathbf{x},t)$. The momentum and mass balance equations can be written as follows:

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{f}\right) - \nabla\cdot\boldsymbol{\sigma} = \mathbf{0} \quad \text{on } \Omega_t \quad \forall t \in (0,T), \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{on } \Omega_t \quad \forall t \in (0,T), \tag{2}$$

where the density ρ is assumed to be constant. The stress tensor $\boldsymbol{\sigma}$ can be decomposed into its isotropic and deviatoric parts

$$\boldsymbol{\sigma}(\mathbf{u},p) = -p\mathbf{I} + \mathbf{T}. \tag{3}$$

We consider only the Newtonian fluids, for which the deviatoric stress is related linearly to the strain rate tensor

$$\mathbf{T} = 2\mu\boldsymbol{\varepsilon}(\mathbf{u}), \quad \boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^{\mathrm{T}}\right), \tag{4}$$

where μ is the dynamic viscosity. The Dirichlet and Neumann-type boundary conditions are represented as:

$$\mathbf{u} = \boldsymbol{g} \quad \text{on } (\Gamma_t)_{\boldsymbol{g}}, \tag{5}$$

$$\mathbf{n}\cdot\boldsymbol{\sigma} = \boldsymbol{h} \quad \text{on } (\Gamma_t)_{\boldsymbol{h}}, \tag{6}$$

where $(\Gamma_t)_{\boldsymbol{g}}$ and $(\Gamma_t)_{\boldsymbol{h}}$ are complementary subsets of the boundary Γ_t.

The initial condition consists of a divergence-free velocity field specified over the entire domain

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0, \qquad \nabla\cdot\mathbf{u}_0 = 0 \quad \text{on } \Omega_0. \tag{7}$$

In most cases, the meshes used would not have sufficient resolution to resolve the flow features well enough to fully capture turbulence effects at high Reynolds numbers. This makes it necessary to use some type of turbulence model. While turbulence modeling is a wide area with many different kinds of approaches, here we use a simple Smagorinsky turbulence model. In this model, the kinematic viscosity $\nu = \mu/\rho$ is augmented by an eddy viscosity ν_t which is defined as

$$\nu_t = (Ch)^2(2\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}))^{1/2}, \tag{8}$$

where C is a constant and h is the element length. The element length is defined here as the maximum of the edge lengths for the element. See [7] for details and references on the turbulence model, and [8] for early definition.

3. Deformable-Spatial-Domain/Stabilized Space-Time (DSD/SST) formulation

In the DSD/SST approach the finite element formulation of the governing equations is written over the associated space-time domain of the problem, with the option for the spatial domain to change its shape in time. This approach is naturally suited for problems involving moving boundaries and interfaces, as the deformation of the spatial domain is taken into account automatically. To construct the finite element function spaces for the space-time method, we first partition the time interval $(0, T)$ into subintervals $I_n = (t_n, t_{n+1})$, where t_n and t_{n+1} belong to an ordered series of time levels $0 = t_0 < t_1 < \cdots < t_N = T$. Let $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$. We will define the space-time slab Q_n as the domain enclosed by the surfaces Ω_n, Ω_{n+1}, and P_n, where P_n is the surface described by the boundary Γ_t as t traverses I_n. As it is the case with Γ_t, surface P_n is decomposed into $(P_n)_{\boldsymbol{g}}$ and $(P_n)_{\boldsymbol{h}}$ with respect to the type of boundary condition (Dirichlet and Neumann) being applied.

For each space-time slab Q_n, we define the following finite-dimensional trial solution ($(\mathcal{S}^h_{\mathbf{u}})_n$ and $(\mathcal{S}^h_p)_n$) and test function ($(\mathcal{V}^h_{\mathbf{u}})_n$ and $(\mathcal{V}^h_p)_n$) spaces for the velocity and pressure:

$$(\mathcal{S}^h_{\mathbf{u}})_n = \left\{\mathbf{u}^h \mid \mathbf{u}^h \in \left[H^{1h}(Q_n)\right]^{n_{\mathrm{sd}}},\ \mathbf{u}^h \doteq \boldsymbol{g}^h \quad \text{on } (P_n)_{\boldsymbol{g}}\right\}, \tag{9}$$

$$(\mathcal{V}^h_{\mathbf{u}})_n = \left\{\mathbf{w}^h \mid \mathbf{w}^h \in \left[H^{1h}(Q_n)\right]^{n_{\mathrm{sd}}},\ \mathbf{w}^h \doteq \mathbf{0} \quad \text{on } (P_n)_{\boldsymbol{g}}\right\}, \tag{10}$$

$$(\mathcal{S}^h_p)_n = (\mathcal{V}^h_p)n = \left\{p^h \mid p^h \in H^{1h}(Q_n)\right\}, \tag{11}$$

where $H^{1h}(Q_n)$ represents the finite-dimensional function space constructed over the space-time slab Q_n by using first-order polynomials in space and in time. The interpolation function spaces are discontinuous in time and continuous in space.

The stabilized space-time formulation of Eqs. (1) and (2) can then be written as follows: given $(\mathbf{u}^h)^-_n$, find $\mathbf{u}^h \in (\mathcal{S}^h_{\mathbf{u}})_n$ and $p^h \in (\mathcal{S}^h_p)_n$ such that $\forall\mathbf{w}^h \in (\mathcal{V}^h_{\mathbf{u}})_n$ and $\forall q^h \in (\mathcal{V}^h_p)_n$:

$$\int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}\right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h)\, \mathrm{d}Q + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q$$
$$+ \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho\left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-\right) \mathrm{d}\Omega + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{MOM}} \frac{1}{\rho} \left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{w}^h, q^h)\right]$$
$$\cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}\right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h)\right] \mathrm{d}Q + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\mathrm{CONT}} \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q = \int_{(P_n)_h} \mathbf{w}^h \cdot \boldsymbol{h}^h \, \mathrm{d}P. \tag{12}$$

The following notation is being used in Eq. (12):

$$(\mathbf{u}^h)_n^{\pm} = \lim_{\varepsilon \to 0} \mathbf{u}(t_n \pm \varepsilon), \tag{13}$$

$$\int_{Q_n} \cdots \mathrm{d}Q = \int_{I_n} \int_{\Omega_t^h} \cdots \mathrm{d}\Omega \, \mathrm{d}t, \tag{14}$$

$$\int_{P_n} \cdots \mathrm{d}P = \int_{I_n} \int_{\Gamma_t^h} \cdots \mathrm{d}\Gamma \, \mathrm{d}t. \tag{15}$$

The definitions of the stabilization parameters τ_{MOM} and τ_{CONT} can be found in [9].

The solution to Eq. (12) is obtained for all of the space-time slabs $Q_0, Q_1, \ldots, Q_{N-1}$ sequentially, and the computations start with

$$(\mathbf{u}^h)_0^- = \mathbf{u}_0^h. \tag{16}$$

The nonlinear system given by Eq. (12) is solved with the Newton–Raphson method. The large linear system that arises at each step of the Newton–Raphson algorithm is solved in turn by using an iterative solution technique with the Generalized Minimum Residual (GMRES) [10] update method.

4. Shear-Slip Mesh Update Method (SSMUM) and its implementation

With the SSMUM approach, the DSD/SST formulation can be very effectively applied to modeling of geometries undergoing *large* but regular deformations, such as straight-line translations or rotations. This is accomplished by letting, at each time step, the elements in a thin zone of the mesh to undergo "shear" deformation, and remeshing this zone, with the required frequency, by regeneration of inter-element connectivity. The idea behind the special-purpose mesh design is the matching of the node positions in the old (deformed) and new (good-quality) meshes, so that the projection step can be skipped. Since only a small part of the overall connectivity is being regenerated, the computational cost is negligible. And because the meshes are designed so that node positions after the remeshing coincide with other node positions (before remeshing), the regeneration of the connectivity in the thin zone can be referred to simply as *re-connecting*. A typical sequence of events is shown in Fig. 1.

Applications of the SSMUM are based on special-purpose mesh designs which combine regions of rigid, non-deforming elements with layers of shear-absorbing, deforming elements. A translating object is embedded in a strip (in 2D) or tube (in 3D) of rigid elements that move "glued" to that object. Similarly, a rotating object is embedded in a disk of rigid elements which rotate glued to that object. These non-deforming regions are immersed in another set of non-deforming elements spanning the exterior boundaries, and connected with the previously described shear-slip layer. This concept, which was discussed in more detail in [6], is illustrated for the 2D case in Fig. 2.

Let us now limit our attention to cases of 3D rotation, such as the main rotor of a helicopter, to be covered in Section 5. The shear-slip layer in this case surrounds a 3D object (the rotor) which is rotating around its main axis. The most natural layer design is an axisymmetric disk with its main axis coinciding with the axis of rotation. The disk may or may not have openings at either of its points of intersection with

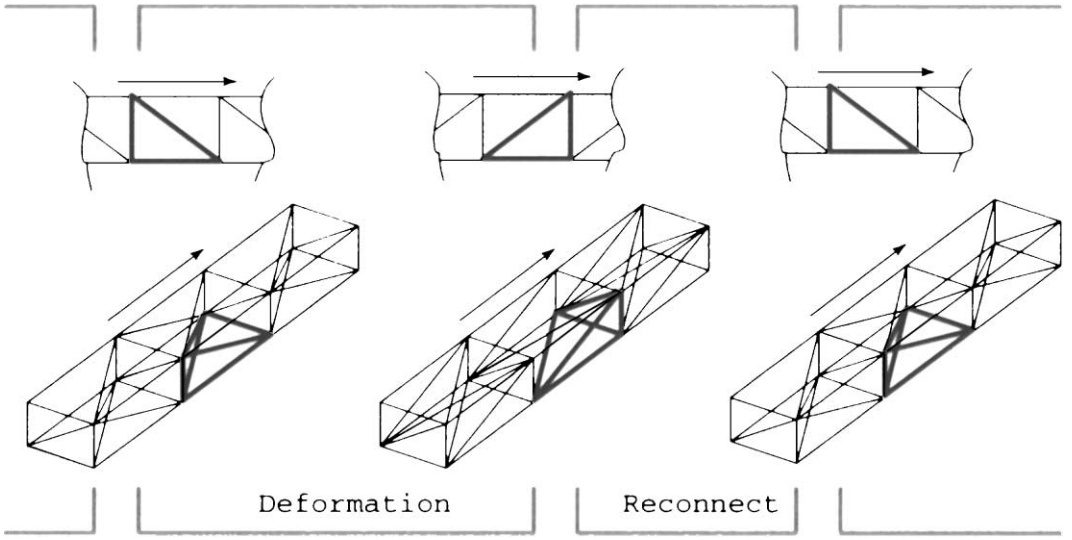

Fig. 1. Shear-slip layer concept in 2D (top) and 3D (bottom): the sequence of deformation followed by reconnecting.

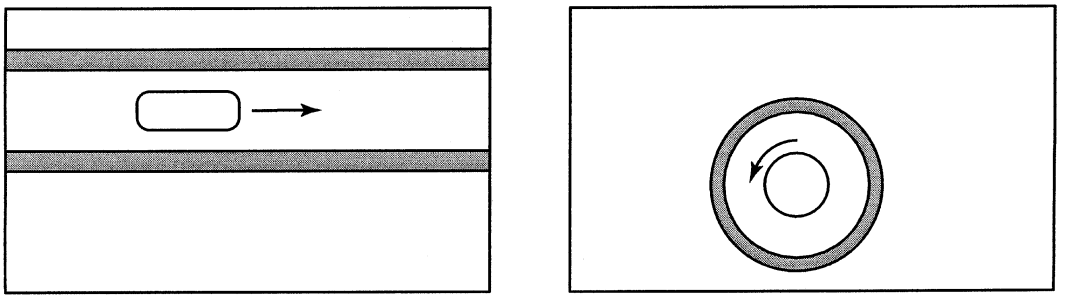

Fig. 2. Shear-slip layer concept: special-purpose meshes. Regions of deforming elements are shown in grey, and regions of rigid elements are shown in white.

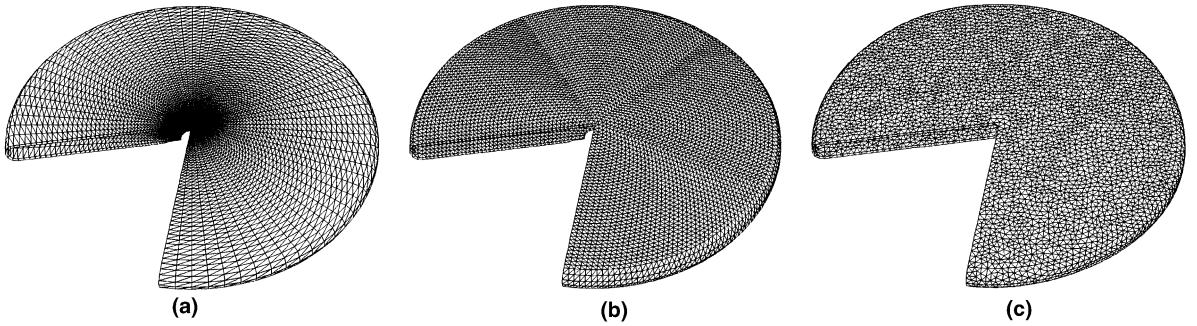

Fig. 3. Shear-slip layer disk: (a) regular, (b) semi-regular and (c) unstructured design.

the axis, depending on whether the rotating object is free or attached, with a spindle or axle, to other objects outside the layer. Some options for the disk layer mesh are illustrated in Fig. 3. The regular, semi-regular and unstructured designs shown can all be used with the SSMUM approach.

Of the examples shown in Fig. 3, the first two have been actually constructed for the purpose of the helicopter rotor simulation. They feature an opening at the bottom where the rotor hub must penetrate the shear-slip layer, and the thickness of the layer is reduced in the vicinity of that opening, so that small

clearance between the fixed (exterior) and rotating (interior) objects can be accommodated. The third example in Fig. 3 is for illustration purposes only. Both the regular and semi-regular designs have a one-to-one correspondence between surface elements on the interior and exterior surfaces. These pairs of triangular elements are then connected to create a prism "super-element", which is then subdivided into three tetrahedra. As the inner disk rotates, new surface nodes on the inner surface become aligned with the surface nodes on the outer surface. The prisms which contain these nodes can be then recreated (in a process we refer to as reconnecting), so that original element shapes are restored, as was shown previously in Fig. 1. In the case of the regular design, Fig. 3(a), all the elements in the layer are recreated at the same time, at intervals which are determined by the circumferential resolution of the surface mesh, time step, and the desired rotation velocity. As can be inferred from the Fig. 1, the reconnecting is a local process within the basic prisms into which the shear-slip layer is subdivided. This fact lends a degree of simplicity to the design, which is in fact the one chosen in the application example in Section 5.

The regular design of the layer disk has two drawbacks. First, the need to maintain the same number of elements in the circumferential direction throughout the disk leads to high aspect ratios close to the axis and at the outer rim. Second, all the "peels" of the layer meet at the axis (in our case on the top surfaces only), creating abnormally high edge-to-node ratio for several nodes. This can adversely affect load-balance and communication stages in distributed-memory implementations. The semi-regular design of the shear-slip layer, Fig. 3(b), does not exhibit these flaws. However, the process of reconnecting can be no longer synchronized for all the elements. The elements further removed from the axis, which undergo larger displacement, must reconnect more frequently, while the elements close to the axis will reconnect only rarely. The reconnecting is no longer localized within the prism super-elements, and certain number of inter-element transformations (edge-swapping) is necessary to maintain mesh validity.

Both regular and semi-regular designs have so far been created with special utilities. Another, slightly more automatic, approach to generating and regenerating shear-slip layers would involve creation, in the preprocessing phase, of a number of layer meshes. Each layer mesh would correspond to different angular displacement of the inner mesh with respect to the outer mesh. For example, if the period of rotation of the inner mesh is being divided into N time steps during the simulation, then the preprocessing phase would include the generation, with an automatic mesh generator, of N meshes filling the shear-slip layer, which would connect the outer surface of the layer mesh with its inner surface at all N possible stages of angular displacement. These meshes would then be progressively read in during the simulation, and used to connect the rigid portion of the mesh. Unlike the previous approaches, this unstructured design of the shear-slip layer, Fig. 3(c), does not require any particular structure from the surface meshes bounding the shear-slip layer, and therefore allows for their generation via an automatic surface mesh generator.

5. Numerical example

In this section, we apply the DSD/SST formulation and SSMUM to computation of flow past a helicopter with its main rotor in motion. Due to the relative motion between the rotor and the fuselage, the problem cannot be handled by attaching a moving reference frame to the rotor. In the current simulation, the smaller, tail rotor is omitted from the model, but in principle the SSMUM approach could be used to model it as well.

The computational domain, shown in Fig. 4, has dimensions $30.00\ \text{m} \times 20.00\ \text{m} \times 20.00\ \text{m}$. This domain is being viewed from the quadrant $(-x, -y, -z)$. The fuselage of the helicopter is modeled after Boeing Sikorsky Comanche prototype, although an exact reproduction of a particular type of helicopter was not among our top priorities. The length of the fuselage (without the rotor) is 13.22 m. The main rotor, with five blades, has a diameter of 11.90 m, and its axis coincides with the z-axis of the domain. The shear-slip layer is an axisymmetric shell with interior radii of 6.00 m and 0.25 m, and a mostly uniform thickness of 0.10 m. The shell is closed, except for the opening at the base of the rotor. To accommodate the close spacing between the top of the fuselage and the rotor blades, the thickness of the layer in that area is reduced to 0.04 m.The mesh consists of 361,434 space-time nodes and 1,096,225 tetrahedral elements, and is shown in Figs. 5 and 6.The regular shear-slip layer, which is one element thick, has 80 segments in the circumferential direction and 100 segments in the radial direction. This layer goes through shear

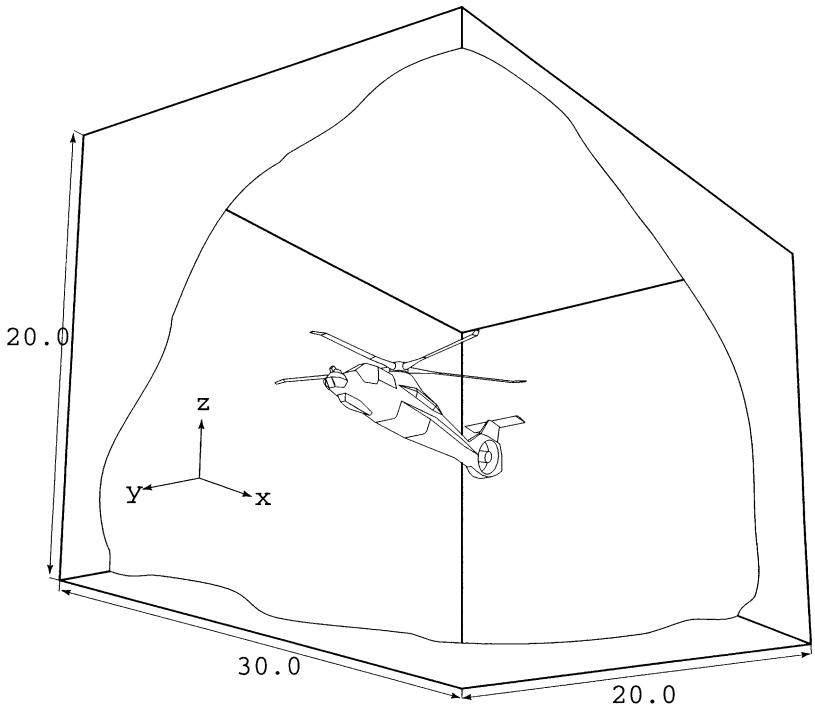

Fig. 4. Flow past a helicopter: boundaries of the computational domain.

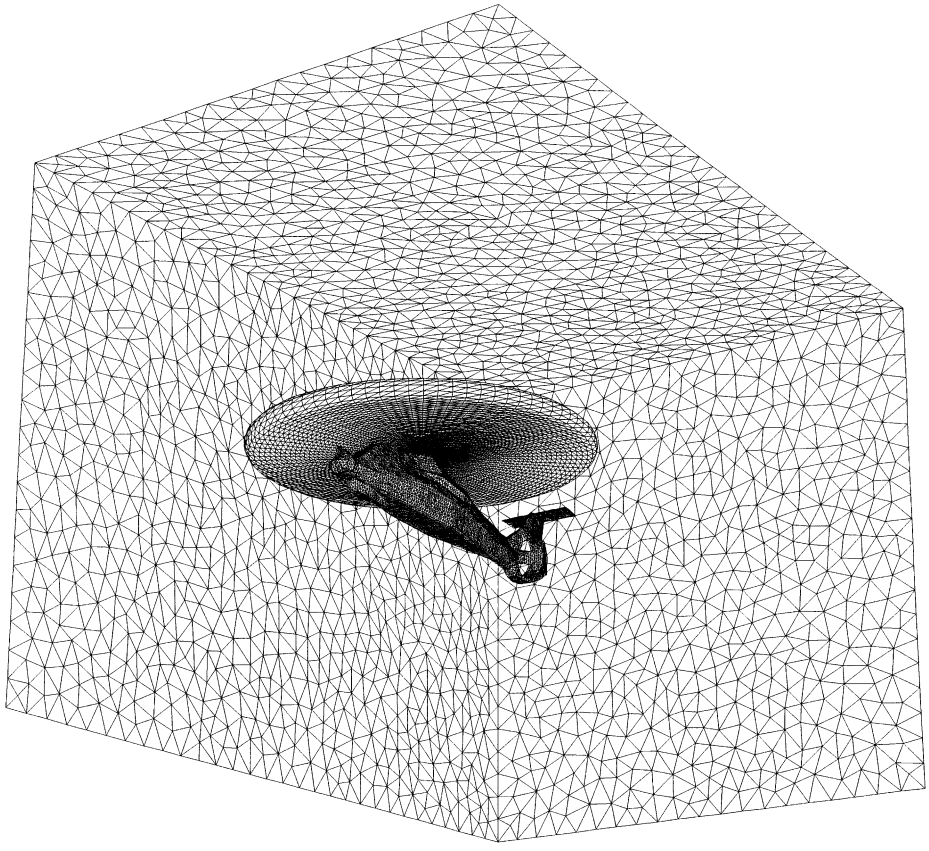

Fig. 5. Flow past a helicopter: selected boundaries of the initial mesh.

deformation during each time step, and at the end of the time step re-connects to the new nodes belonging to the rotating interior disk. The unstructured meshes in both the inner (rotating) and the outer (stationary) rigid regions of the domain were generated using an automatic mesh generator [11]. The structured mesh which fills the shear-slip region was generated manually. Two different cross-sections of the mesh are shown in Figs. 7 and 8. The helicopter is assumed to be in a forward horizontal flight, and a free-stream velocity of (10.0, 0.0, 0.0) m/s is imposed at the upstream boundary. The Reynolds number based on the upstream

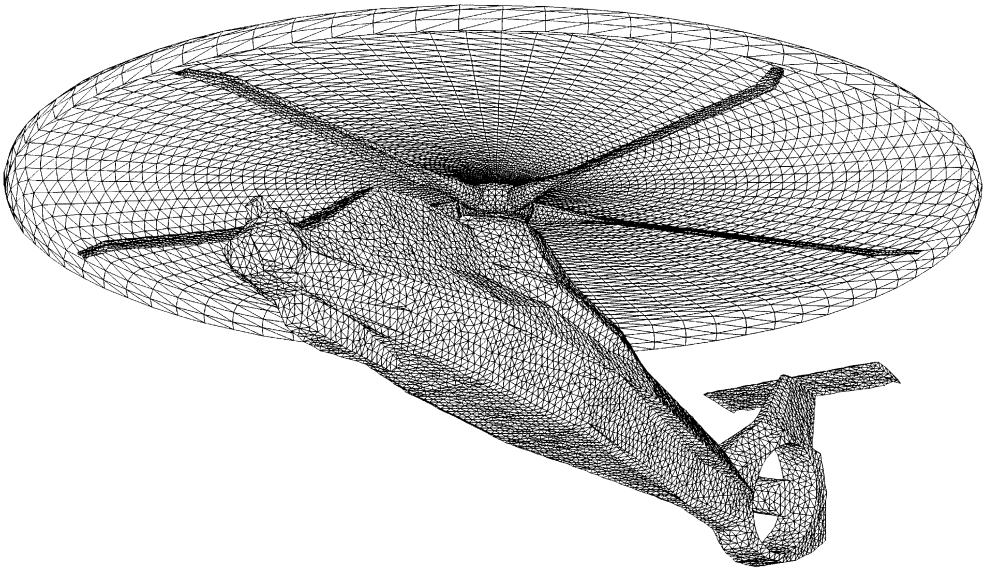

Fig. 6. Flow past a helicopter: close-up view of the surface meshes for the fuselage, rotor, and the top portion of the inner boundary of the shear-slip layer.

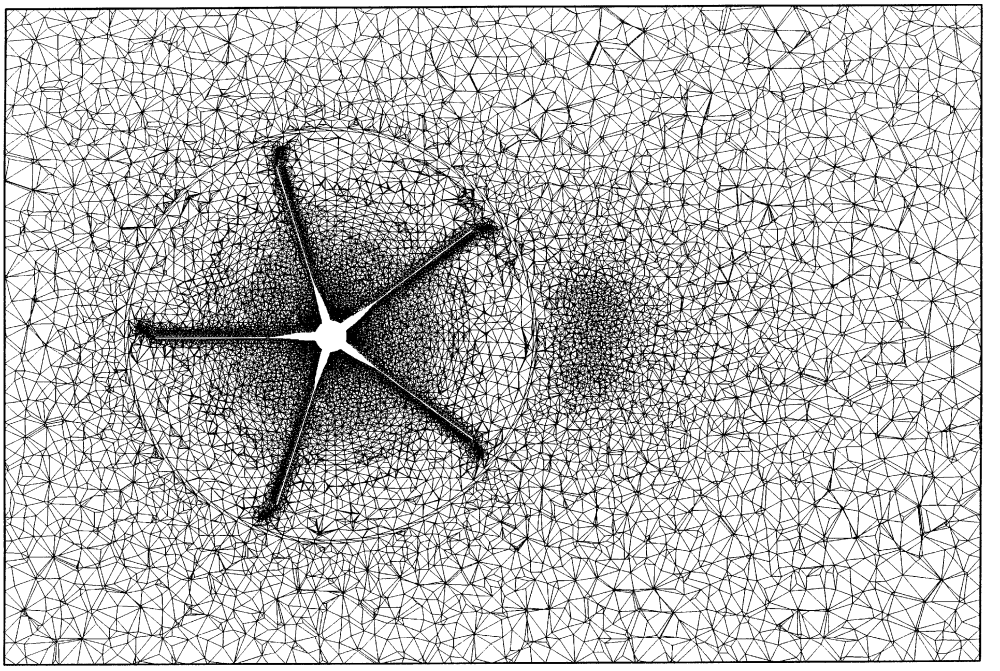

Fig. 7. Flow past a helicopter: cross-section of the mesh through the center of the rotor hub in the x–y plane.

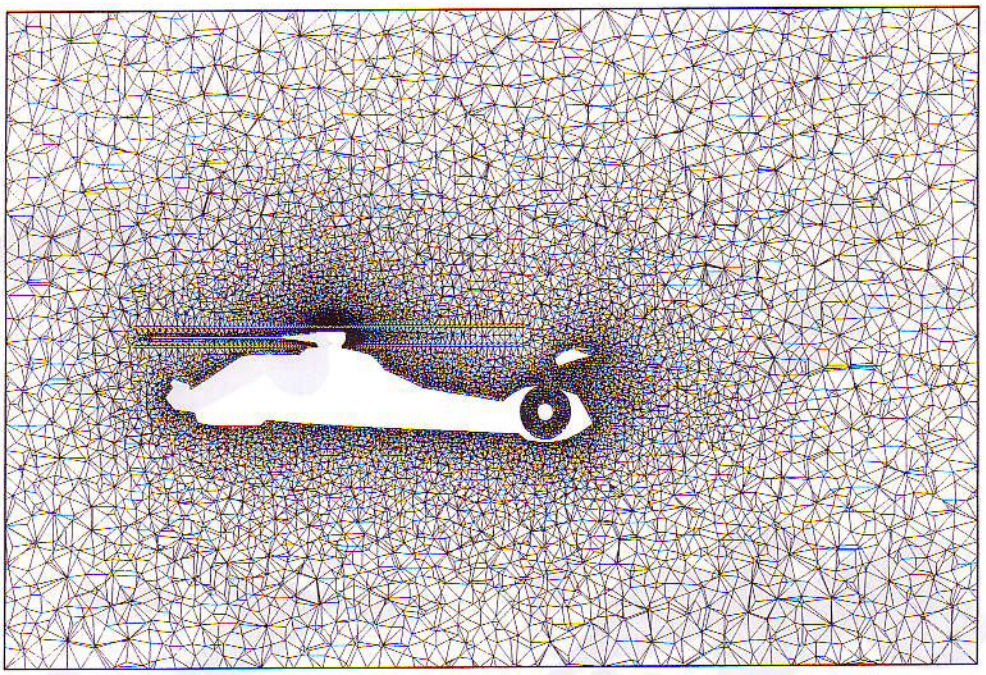

Fig. 8. Flow past a helicopter: cross-section of the mesh in the x–z plane.

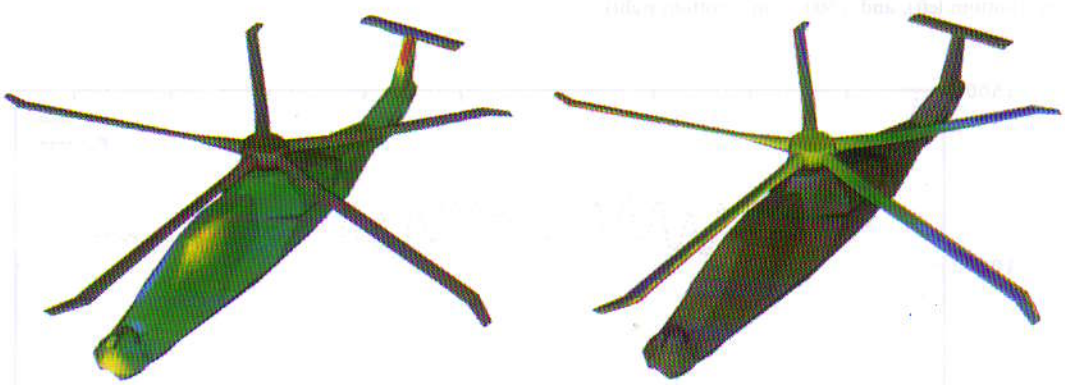

Fig. 9. Flow past a helicopter: air pressure at $t = 0.562$ s on the fuselage surface (left, limits $-250\ \mathrm{N/m^2} < p < +250\ \mathrm{N/m^2}$) and rotor surface (right, limits $-4000\ \mathrm{N/m^2} < p < +2000\ \mathrm{N/m^2}$).

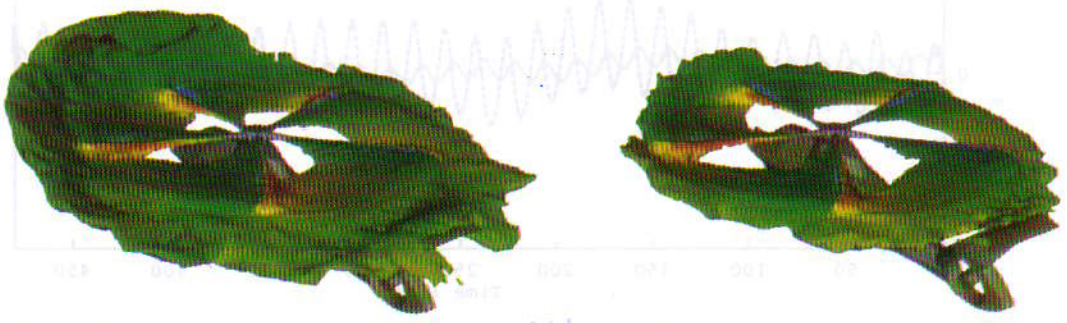

Fig. 10. Flow past a helicopter: iso-surfaces at $t = 0.562$ s corresponding to $|\mathbf{u}| = 40$ m/s (left) and 50 m/s (right).

velocity and the main rotor diameter is approximately 8×10^6. In the Smagorinsky turbulence model we use $C = 0.15$. Zero normal velocity and zero shear stress conditions are specified at all transverse boundaries, and a traction-free condition is imposed at the outflow. The steady-state solution at Reynolds

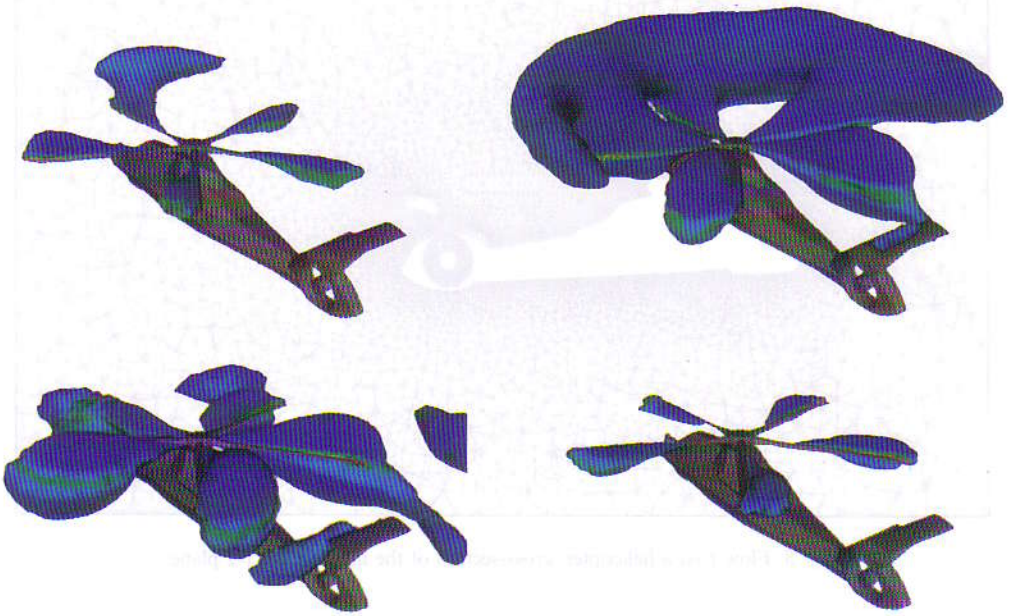

Fig. 11. Flow past a helicopter: iso-surfaces at $t = 0.562$ s corresponding to $p = -1000$ N/m^2 (top left), -300 N/m^2 (top right), $+100$ N/m^2 (bottom left), and $+500$ N/m^2 (bottom right).

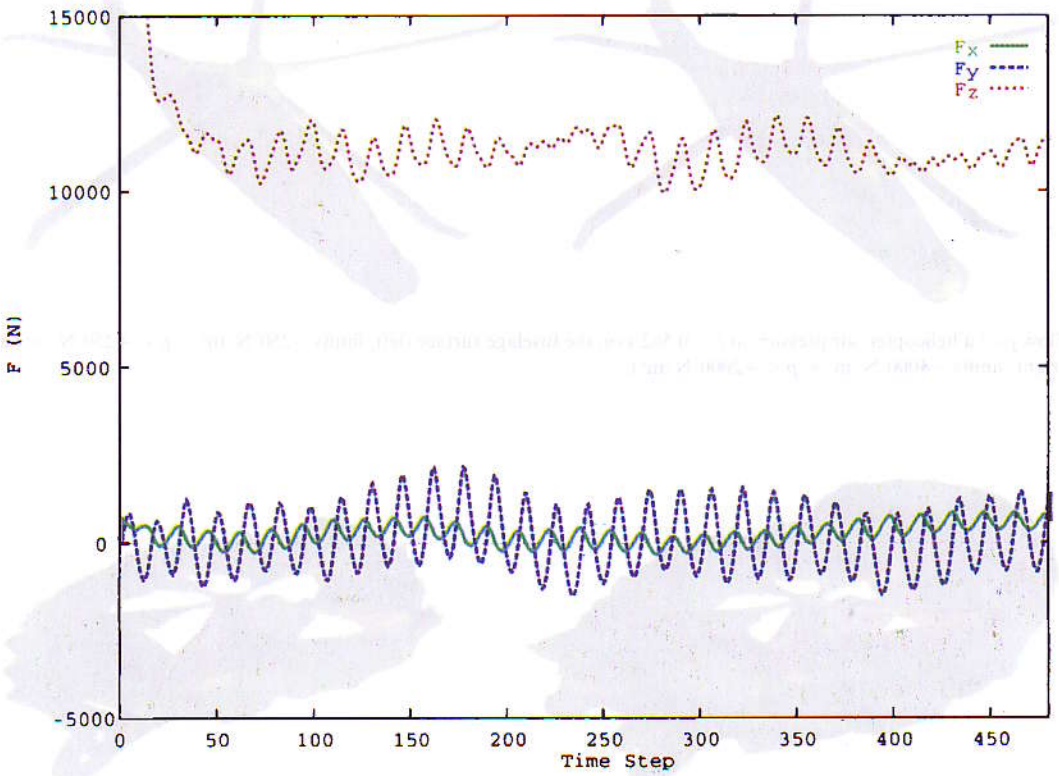

Fig. 12. Flow past a helicopter: time-histories of the forces acting on the fuselage and rotor.

number 8 serves as the initial condition. The unsteady flow is computed for 480 time steps with a time step size of 0.00234 s. The rotor is rotating in counter-clockwise direction when viewed from the top, with a tip velocity of 200 m/s.

Fig. 9 shows, at $t = 0.562$ s, air pressure on the surface of the fuselage and rotor. Fig. 10 shows, at the same instant, two iso-surfaces of the magnitude of the velocity, showing clearly the rotor wake and its roll-up. Fig. 11 illustrates the pressure field by using four selected iso-surfaces.Time-histories of the forces acting on the fuselage and rotor are shown in Fig. 12.

This computation has been carried out on a CRAY T3E-1200. At every time step the coupled, nonlinear equations are solved with four Newton–Raphson iterations. The coupled, linear equations that need to be solved at each Newton–Raphson step are solved also iteratively, with the GMRES update techniques with a Krylov space size of 40.

6. Concluding remarks

We presented a 3D computational technique for simulation of complex flow problems with fast-rotating mechanical components. We specifically targeted simulation of flow past a helicopter with its main rotor in motion. The technique we developed is based on the DSD/SST formulation, SSMUM, and an efficient parallel implementation for distributed-memory parallel computing platforms. The DSD/SST formulation can handle flow problems with moving boundaries and interfaces, including flows with moving mechanical components, but requires, as a companion method, an effective mesh update strategy. This requirement becomes especially critical in complex flow problems. The SSMUM, with specially-developed enhancements for flow past a helicopter with its rotor in motion, has been used to support the DSD/SST formulation in the simulation we reported in this paper. This simulation demonstrates that the computational technique we developed for this class of problems is very effective and has the potential of opening new doors in simulation of real-world problems.

Acknowledgements

This work was sponsored by the AHPCRC under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement number DAAH04-95-2-0003/contract number DAAH04-95-C-0008. The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

[1] T.E. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary tests, Comput. Methods Appl. Mech. Engrg. 94 (1992) 339–351.

[2] T.E. Tezduyar, M. Behr, S. Mittal, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders, Comput. Methods Appl. Mech. Engrg. 94 (1992) 353–371.

[3] T.E. Tezduyar, S. Aliabadi, M. Behr, Enhanced-discretization interface-capturing technique, in: Y. Matsumoto, A. Prosperetti (Eds.), Proceedings of the ISAC'97 High Performance Computing on Multiphase Flows, vols. 1–6, Japan Society of Mechanical Engineers, 1997.

[4] T.E. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, M. Litke, Flow simulation and high performance computing, Comput. Mech. 18 (1996) 397–412.

[5] M. Behr, T.E. Tezduyar, A note on shear-slip mesh update method, in: Lecture Notes of the Workshop on Parallel Computing in Applied Fluid Mechanics, Associazione Amici Scuola Normale Superiore, Pisa, Italy, 1997.

[6] M. Behr, T.E. Tezduyar, Shear-slip mesh update method, Comput. Methods Appl. Mech. Engrg. 174 (1999) 261–274.

[7] C. Kato, M. Ikegawa, Large eddy simulation of unsteady turbulent wake of a circular cylinder using the finite element method, in: I. Celik, T. Kobayashi, K.N. Ghia, J. Kurokawa (Eds.), Advances in Numerical Simulation of Turbulent Flows, FED-Vol. 117, ASME, New York, 1991, pp. 49–56.

[8] J. Smagorinsky, General circulation experiments with the primitive equations, Monthly Weather Rev. 91 (1963) 99–165.

[9] M. Behr, T.E. Tezduyar, Finite element solution strategies for large-scale flow simulations, Comput. Methods Appl. Mech. Engrg. 112 (1994) 3–24.
[10] Y. Saad, M. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Scientific Statist. Comput. 7 (1986) 856–869.
[11] A.A. Johnson, T.E. Tezduyar, Mesh generation and update strategies for parallel computation of 3D flow problems, in: Computational Mechanics'95, Proceedings of International Conference on Computational Engineering Science, Mauna Lani, Hawaii, 1995.