ELSEVIER

Comput. Methods Appl. Mech. Engrg. 190 (2000) 321–332

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A parallel 3D computational method for fluid–structure interactions in parachute systems

Vinay Kalro [1], Tayfun E. Tezduyar *

Team for Advanced Flow Simulation and Modeling, Mechanical Engineering and Materials Science, Rice University-MS 321, 6100 Main Street, Houston, TX 77005, USA

Received 26 July 1999; received in revised form 26 August 1999

Abstract

We present a parallel finite element computational method for 3D simulation of fluid–structure interactions (FSI) in parachute systems. The flow solver is based on a stabilized finite element formulation applicable to problems involving moving boundaries and governed by the Navier–Stokes equations of incompressible flows. The structural dynamics (SD) solver is based on the total Lagrangian description of motion, with cable and membrane elements. The nonlinear equation system is solved iteratively, with a segregated treatment of the fluid and SD equations. The large linear equation systems that need to be solved at every nonlinear iteration are also solved iteratively. The parallel implementation is accomplished using a message-passing programming environment. As a test case, the method is applied to computation of the equilibrium configuration of an anchored ram-air parachute placed in an air stream.

1. Introduction

The parallel 3D computational method presented in this paper has been developed for modeling fluid–structure interactions (FSI) encountered in airdrop systems. The airdrop systems we focus on here include conventional personnel round parachutes, cross parachutes which have limited glide capability, and large gliding ram-air parachutes (parafoils) which can carry payloads up to 21 tons. The design emphasis for these parachute systems is precision delivery under demanding deployment conditions such as strong wind gusts and large offsets. Since conventional design techniques are time-consuming, expensive, and semi-empirical at best, our objective is to build a reliable and cost effective design tool based on the advanced flow simulation and modeling methods we have been developing.

Parachute systems present very complex dynamics arising from interactions between the canopy, suspension lines, payload, and the surrounding air. Parachutes can experience significant canopy deformations and changes in orientation at any stage of their deployment and operation. To correctly represent the actual behavior, modeling of these parachute systems has to involve solution of the governing equations over computational domains which change their shapes in time.

In the earlier models we developed [1–3], the parachute canopy was represented as a structure with prescribed shape changes, and its dynamics was determined as part of the overall solution of the coupled flow and dynamics equations. In a complementary effort, Benney et al. [4] presented detailed structural analysis of parachute systems with assumed air pressure distributions. Stein et al. [5] presented an axi-

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

0045-7825/00/$ - see front matter
PII: S0045-7825(00)00204-8

symmetric model where the parachute structure was represented by cable and axisymmetric membrane elements, and its response was determined by solution of the equation system which took into account the coupling between this structural model and the fluid dynamics of the air surrounding it.

In the computational modeling presented in this paper, the fluid dynamics is governed by the Navier–Stokes equations of incompressible flows. The structural dynamics (SD) is governed by the membrane and cable equations with the Lagrangian description of motion.

In general, for flow problems involving moving boundaries and interfaces, including FSI, we employ the deformable-spatial-domain/stabilized space-time (DSD/SST) method introduced by Tezduyar et al. [6,7]. This method takes automatically into account the changes in the shape of the spatial domain. The stabilized finite element formulations prevent numerical oscillations and other instabilities in solving problems with high Reynolds numbers and strong boundary layers. They also allow us to use equal-order interpolation functions for velocity and pressure. Some of the most established stabilized formulations for incompressible flows are the streamline-upwind/Petrov–Galerkin (SUPG) formulation [8], Galerkin/least-squares (GLS) formulation [9]), and pressure-stabilizing/Petrov–Galerkin (PSPG) formulation [10]. These formulations stabilize the method without introducing excessive numerical dissipation. 2D FSI methods, based on the DSD/SST formulation and developed for flow problems with moving mechanical components, were reported by Wren et al. [11]. 3D FSI simulations of a round parachute canopy with a flow solver based on the DSD/SST method were reported by Stein et al. [12]. In this paper, we focus on computing the equilibrium configuration of the parachute. Since the temporal accuracy is not an important issue for this simulation, we carry out the computations with the arbitrary Lagrangian–Eulerian (ALE) version of our stabilized formulation.

The SD solver is also based on a finite element formulation, and uses cable and membrane elements. The coupled, nonlinear equations system arising from this FSI model is solved iteratively. This is accomplished with a segregated treatment of the fluid and SD equation systems. Each of these nonlinear equation systems is solved with the Newton–Raphson method. The linear equation systems that need to be solved at each Newton–Raphson step are also solved iteratively. The flow and SD solvers have both been implemented for parallel computation within a message-passing programming environment. The complete simulation tool includes an algebraic mesh mover developed for updating the fluid mesh based on the motion of the fluid–structure interface.

The flow solver is described in Section 2, and the SD solver in Section 3. The FSI coupling technique is described in Section 4. Section 5 is a brief summary of the parallel implementation. In Section 6, we report, as a test case, computation of the equilibrium configuration of an anchored ram-air parachute placed in an air stream. The concluding remarks are given in Section 7.

2. Flow solver

2.1. Governing equations

Let $\Omega_t^{\mathrm{f}} \subset \mathbb{R}^{n_{\mathrm{sd}}}$ be the spatial fluid domain of interest bounded by boundary Γ_t^{f} at any instant 't'. Here the superscript f stands for the fluid and n_{sd} is the number of spatial dimensions. The Navier–Stokes equations governing incompressible flows are

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

$$\nabla\cdot\mathbf{u} = 0 \quad \text{on } \Omega_t^{\mathrm{f}}. \tag{2}$$

Here ρ^{f}, $\mathbf{u}$, $\mathbf{f}^{\mathrm{f}}$ and $\boldsymbol{\sigma}^{\mathrm{f}}$ are the density, velocity, body force and the stress tensor, respectively. The stress tensor is written as the sum of its isotropic and deviatoric parts: $\boldsymbol{\sigma}^{\mathrm{f}}(p,\mathbf{u}) = -p\mathbf{I} + \mathbf{T}$; and the fluid is assumed to be Newtonian: $\mathbf{T} = 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, where μ is the viscosity and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor. The boundary Γ_t^{f} is composed of $(\Gamma_t^{\mathrm{f}})_g$ and $(\Gamma_t^{\mathrm{f}})_h$, corresponding to the Dirichlet- and Neumann-type boundary conditions respectively. A divergence-free velocity field is specified as initial condition. The turbulence model used here is a simple kind, namely a Baldwin–Lomax model.

2.2. Finite element formulation

To describe the ALE version of our stabilized formulation, we first assume that we have some suitably-defined finite-dimensional trial solution and test function spaces for velocity and pressure: $\mathscr{S}^h_{\mathbf{u}}$, $\mathscr{V}^h_{\mathbf{u}}$, $\mathscr{S}^h_p$ and $\mathscr{V}^h_p = \mathscr{S}^h_p$. The stabilized ALE finite element formulation can then be written as follows: find $\mathbf{u}^h \in \mathscr{S}^h_{\mathbf{u}}$ and $p^h \in \mathscr{S}^h_p$ such that $\forall \mathbf{w}^h \in \mathscr{V}^h_{\mathbf{u}}$ and $\forall q^h \in \mathscr{V}^h_p$

$$\begin{aligned}
&\int_\Omega \mathbf{w}^h \cdot \rho^{\mathrm{f}} \left(\left. \frac{\partial \mathbf{u}^h}{\partial t} \right|_{\xi} + (\mathbf{u}^h - \mathbf{v}^h) \cdot \nabla \mathbf{u}^h - \mathbf{f}^{\mathrm{f}} \right) \mathrm{d}\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}^{\mathrm{f}}(p^h, \mathbf{u}^h)\, \mathrm{d}\Omega + \int_\Omega q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}\Omega \\
&\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \frac{1}{\rho^{\mathrm{f}}} \left(\tau_{\mathrm{SUPG}} \rho^{\mathrm{f}} (\mathbf{u}^h - \mathbf{v}^h) \cdot \nabla \mathbf{w}^h + \tau_{\mathrm{PSPG}} \nabla q^h \right) \cdot \left[\text{Ł}(\mathbf{u}^{\mathrm{f}}, p^{\mathrm{f}}) - \mathbf{f}^{\mathrm{f}} \right] \mathrm{d}\Omega^e \\
&\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{LSIC}} \nabla \cdot \mathbf{w}^h \rho^{\mathrm{f}} \nabla \cdot \mathbf{u}^h \, \mathrm{d}\Omega^e = \int_{\Gamma_h} \mathbf{w}^h \cdot \mathbf{h}^h \, \mathrm{d}\Gamma,
\end{aligned} \tag{3}$$

where

$$\text{Ł}(\mathbf{w}^h, q^h) = \rho^{\mathrm{f}} \left(\left. \frac{\partial \mathbf{w}^h}{\partial t} \right|_{\xi} + (\mathbf{u}^h - \mathbf{v}^h) \cdot \nabla \mathbf{w}^h \right) - \nabla \cdot \boldsymbol{\sigma}^{\mathrm{f}}(q^h, \mathbf{w}^h). \tag{4}$$

In the above formulation $\mathbf{v}^h$ is the local mesh velocity, and τ_{SUPG}, τ_{PSPG}, and τ_{LSIC} are the stabilization parameters. The temporal derivatives are computed with respect to the moving mesh.

3. SD solver

3.1. Governing equations

Let $\Omega^{\mathrm{s}}_t \subset \mathbb{R}^{n_{\mathrm{sd}}-1}$ be the spatial structure domain of interest bounded by boundary Γ^{s}_t at any instant t. Here the superscript s stands for the structure. The governing equations for the structure can be written as follows:

$$\rho^{\mathrm{s}} \left(\frac{\mathrm{d}^2 \mathbf{y}}{\mathrm{d}t^2} - \mathbf{f}^{\mathrm{s}} \right) - \nabla \cdot \boldsymbol{\sigma}^{\mathrm{s}} = 0 \quad \text{on } \Omega^{\mathrm{s}}_t. \tag{5}$$

Here $\mathbf{y}$ is the structural displacement. The boundary Γ^{s}_t is composed of $(\Gamma^{\mathrm{s}}_t)_g$ and $(\Gamma^{\mathrm{s}}_t)_h$.

The structure undergoes large deformations leading to geometric nonlinearities. The resulting strains are assumed to be small, and therefore a materially linear elastic model is used. To account for the kinematic nonlinearities, the constitutive equations are written in the undeformed configuration, in terms of the 2nd Piola–Kirchoff stress tensor $\mathbf{S}$ and the Green–Lagrange strain tensor $\mathbf{E}$. The structure is composed of cable and membrane elements. The cable elements are assumed to be in a state of uniaxial stress:

$$S^{11} = E_{\mathrm{c}} G^{11} G^{11} E_{11}. \tag{6}$$

The membrane elements are assumed to be in a state of planar stress:

$$S^{ij} = (\overline{\lambda}_{\mathrm{m}} G^{ij} G^{kl} + \mu_{\mathrm{m}} [G^{il} G^{jk} + G^{ik} G^{jl}]) E_{kl}, \tag{7}$$

where

$$\overline{\lambda}_{\mathrm{m}} = \frac{2 \lambda_{\mathrm{m}} \mu_{\mathrm{m}}}{(\lambda_{\mathrm{m}} + 2 \mu_{\mathrm{m}})}. \tag{8}$$

Here G^{ij} are the components of the contravariant metric tensor, E_{c} is the modulus of elasticity for the cable, and λ_{m} and μ_{m} are the Lame parameters for the membrane. The indices i, j, k, l take on the range 1–2. For further details, the reader is referred to Bathe [13] and Po [14].

3.2. Finite element formulation

The finite element formulation is derived from the principle of virtual work:

$$\int_{\Omega_0^s} \rho^s \frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}t^2} \cdot \delta\mathbf{y}\,\mathrm{d}\Omega + \int_{\Omega_0^s} \mathbf{S} : \delta\mathbf{E}\,\mathrm{d}\Omega = \int_{\Gamma_t^s} \mathbf{t} \cdot \delta\mathbf{y}\,\mathrm{d}\Gamma + \int_{\Omega_t^s} \rho^s \mathbf{f}^s \cdot \delta\mathbf{y}\,\mathrm{d}\Omega. \tag{9}$$

Here $\mathbf{t}$ is the surface traction, which also adds to the nonlinearity since it results in a follower force field. The left-hand side terms in Eq. (9) are written in the original configuration. Upon discretization using appropriate function spaces, a nonlinear system of equations is obtained at each time-step. This equation system is solved with the Newton–Raphson method, where the increments of the structural displacements are calculated by solving the following linear equation system:

$$\begin{aligned} &\left(\frac{\mathbf{M}}{\beta\Delta t^2} + \frac{(1+\alpha)\gamma\mathbf{C}}{\beta\Delta t} + (1+\alpha)\mathbf{K}\right)\Delta\mathbf{d}^i = \mathbf{R}^i, \\ &\mathbf{C} = \eta\mathbf{M} + \zeta\mathbf{K}. \end{aligned} \tag{10}$$

Here $\mathbf{M}$ is the mass matrix, $\mathbf{C}$ is an artificial damping matrix, which stabilizes the structural system, $\mathbf{K}$ is the stiffness matrix, $\mathbf{R}^i$ is the residual vector at the ith iteration, and $\Delta\mathbf{d}^i$ is the increment in the nodal displacement vector $\mathbf{d}$. The parameters α, β, and γ come from the Hilber–Hughes–Taylor scheme [15], which is used for time-integration of the spatially discretized equations.

4. Fluid–structure coupling strategy

4.1. Interface treatment

Let Γ_t^{fs} be the fluid boundary interfacing with the structure domain. The coupling is enforced by transferring the velocity and displacements from the structure to the fluid, and in return the surface forces from the fluid to the structure. Therefore, for the flow solver, $\Gamma_t^{\mathrm{fs}} \subset (\Gamma_t^{\mathrm{f}})_g$; and for the SD solver, $\Gamma_t^{\mathrm{fs}} \subset \Omega_t^s$. In transferring the fluid forces at the interface to the structure, we neglect the viscous part of these forces, and transfer only the pressure part.

In terms of the relationship between the fluid and structure meshes at the interface, we could have either matching or nonmatching meshes (see Fig. 1). In the case of matching meshes, the transfer of data between the fluid and structure meshes would be very simple. However, having nonmatching meshes might be more practical or desirable, because the two meshes might have been generated independently, or we might prefer the fluid mesh to be more refined than the structure mesh. In such cases, the data between the fluid and structure meshes are transferred by least-squares projection. The fluid forces are projected to the structure nodes by using the following least-squares equation:

$$\int_{\Gamma_0^{\mathrm{fs}}} \delta p^s (p^s - p^{\mathrm{f}})\, d\Gamma = 0. \tag{11}$$

Similarly, the structure velocities and displacements are projected to the fluid nodes by using the following least-squares equations:

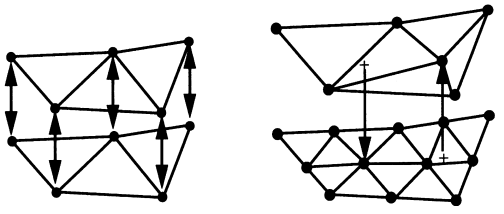

Fig. 1. Fluid–structure interface: matching (left) and nonmatching (right) meshes.

$$\int_{\Gamma_0^{fs}} \delta\mathbf{u}^f \cdot (\mathbf{u}^f - \mathbf{u}^s)\, d\Gamma = 0,$$
$$\int_{\Gamma_0^{fs}} \delta\mathbf{y}^f \cdot (\mathbf{y}^f - \mathbf{y}^s)\, d\Gamma = 0. \tag{12}$$

To evaluate the integrals in Eq. (11), the quadrature points of the surface elements in the structure mesh need to be searched for in the surface elements of the fluid mesh. The opposite is the case in evaluating the integrals in Eqs. (12). Since these projections are always carried out on the undeformed interface, these searches are performed only once. These equation systems are solved iteratively, where the original mass matrices are approximated by their lumped versions.

Other methods for exchanging information at the interface have been proposed. For example, for compatible meshes, the finite element methodology possesses a built-in approach to calculating the consistent forces at the interface, which are simply the unassembled right-hand side contributions to the interface nodes from the fluid elements. When dealing with incompatible meshes, energy conservation principles can be invoked to project these consistent forces onto the structure. These methods can also handle viscous forces with no special treatment. In our methodology (under the assumption of negligible viscous forces), the projections of the displacements (velocities) and pressure are simply the inverses of each other and can be viewed as energy preserving.

4.2. Time-integration algorithms

Depending on whether we are looking for a time-accurate solution or an equilibrium solution, we propose here two different time-integration algorithms: a time-accurate algorithm with matching fluid and structure time-steps, and, for the purpose of reaching an equilibrium solution, an algorithm with larger time-steps for the fluid compared to the structure.

The algorithm with matching time-steps is described below:

- increment time-step;
- calculate predictors for fluid and structure;
- fluid–structure iteration loop:
  - send fluid forces to structure,
  - Newton–Raphson iterations for structure,
  - send structure displacements and velocities to fluid,
  - fluid mesh update,
  - Newton–Raphson iteration(s) for fluid,
  - repeat until converged;
- update fluid and structure solutions;
- repeat until end of time-period is reached.

When we use time-marching just to reach an equilibrium solution, we can subcycle the structure by using time-steps smaller than the time-step used for the fluid. This is particularly desirable since normally the structure requires smaller time-steps. We also note that having multiple time-steps for the structure does not create a major additional cost, because the structure problem is essentially a 2D problem. In the algorithm described below, at each time-step of the fluid time-marching, the structure takes multiple time-steps to reach an equilibrium configuration based on the fluid forces corresponding to the fluid time-level. Because the cost for structure computations per time-step is much smaller compared to the fluid, at each fluid time-step we can afford to use a large number of structure time-step to insure that the structure reaches that equilibrium.

- increment fluid time-step;
- calculate predictors for fluid;
- send fluid forces to structure;
- subcyle structure:
  - increment structure time-step,
  - calculate predictors for structure,

  - Newton–Raphson iterations for structure,
  - update the structure solution,
  - repeat until structural equilibrium is reached;
- send structure displacements and velocities to fluid;
- fluid mesh update;
- Newton–Raphson iterations for fluid;
- update the fluid solution;
- repeat until final equilibrium is reached.

5. Parallel implementation

Both the flow and SD solvers have been implemented for parallel computation within a message-passing programming environment. In calculating the residual vectors for the flow solver, where the equation systems are a lot larger and therefore the memory requirements could be a lot higher, we have the option of using an element-vector-based (i.e., matrix-free) [16] or a sparse-matrix-based [17] computation method. In this paper, we use a special-purpose mesh moving method. However, in general, our FSI strategy includes using an automatic mesh moving method. The automatic mesh mover, which is based on solution of the equations of elasticity governing the motion of the fluid nodes, is very similar in functionality to the flow solver. Therefore, in calculating the residual vectors, we again have the options of using an element-vector-based or a sparse-matrix-based computation method. Since the SD solver involves a smaller number of equations, the residual vector calculations are sparse-matrix-based. The parallel implementation of the individual solvers are essentially identical. The meshes are partitioned into contiguous, even-sized clusters of elements, and each cluster is assigned to a processor. A two-step communication route is set up between the global and element-level data structures for the gather (localization) and scatter (globalization) operations (see Fig. 2).

The communication between the solvers are handled by a master processor. The interface data from both the fluid and structure meshes are extracted and stored on this master processor. Communication traces between these interface meshes and the associated fluid and structure meshes are also preprocessed. Data exchange at the interface is accomplished on the master processor and subsequently communicated to the other processors within the corresponding solver (see Fig. 3).

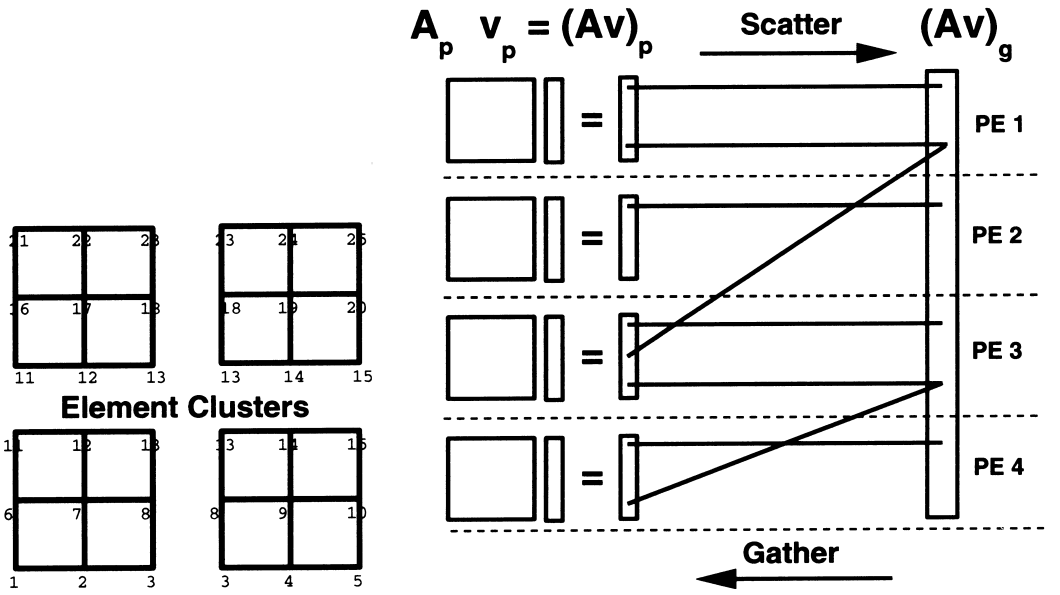

Fig. 2. Parallel implementation of the flow and SD solvers.

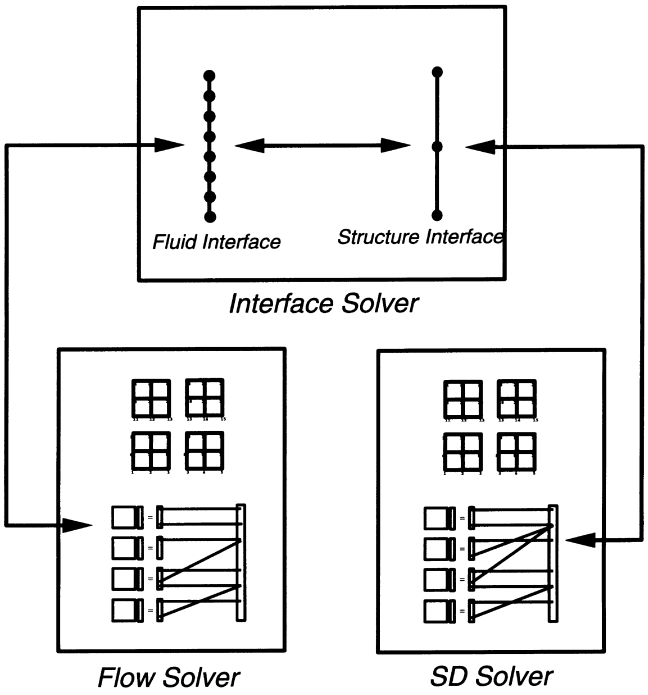

Fig. 3. Parallel implementation of the coupling between the flow and SD solvers.

6. Simulation

The methods presented have been applied to calculating the equilibrium shape and orientation of an anchored ram-air parachute. Once placed in the air stream, the moments due to line drag cause the parachute to pitch backwards to a configuration where these moments are balanced by the nose-down aerodynamic moment on the canopy.

The following conditions and assumptions have been used in the parachute model and mesh design:

- Ram-air parachutes have inlets at the leading edge which allow the air to flow in and inflate the parafoil. This inflation is essentially caused by the air pressure build up due to the stagnation of flow inside the parafoil cells. Our model has a closed leading edge, and we impose a uniform air pressure (equal to the dynamic pressure) inside the parafoil to keep it inflated.
- The initial shape of the parafoil is assumed to be a wing with smooth surfaces, with $R/b = 1$, and with an aspect ratio of 3. Here R is the radius of curvature of the wing, and b its span (see Fig. 4).
- The parachute canopy and ribs are modeled with quadrilateral membrane elements. The lines are modeled with 2-noded cables which cannot bear compressive loads. This is enforced by setting the cable modulus to 0 when the stresses are compressive. The mesh used in this simulation consists of 7474 nodes, with 4486 membrane elements, and 3740 cable elements (see Fig. 4), and with 80×48 elements on the interface.
- To account for the line drag, the lines are modeled as slender cylinders placed in the free stream. The normal force per unit length ($\mathbf{f}_l$) on the lines is computed by using the expression

$$\mathbf{f}_l = 0.5\ C_{\mathrm{D}} |\mathbf{U}_{\mathrm{n}}| \mathbf{U}_{\mathrm{n}} d. \tag{13}$$

  Here, C_{D} is the drag coefficient for a circular cylinder, d is the line diameter, and $\mathbf{U}_{\mathrm{n}}$ is the normal component of the free-stream velocity relative to the cable.
- Parachute fabrics are typically characterized by material properties along the so called warp and fill directions. In our simulation we model the fabric as isotropic, linear elastic material, using the average properties from the two directions, for a heavy weight nylon cloth.

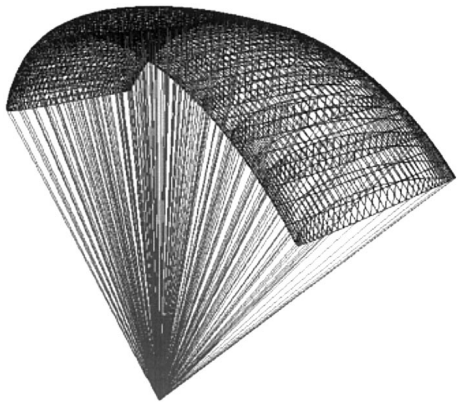

Fig. 4. FSI simulation: the structure mesh for the parachute with an assumed shape. The mesh consists of 7474 nodes, with 4486 quadrilateral membrane elements and 3740 cable elements, and with 80×48 elements on the interface.

- A significant amount of mass- and stiffness-proportional damping is necessary for stable time-integration of the structure equations. This damping causes the structure to 'ooze' into its final configuration. This is acceptable in this simulation since we are not looking for a time-accurate solution in reaching that final configuration. Therefore, of the two time-stepping algorithms we proposed earlier, we use the one with smaller time-steps for the structure and designed for the purpose of reaching an equilibrium solution. In this particular simulation, for each fluid time-step, we use 100 structure time-steps.
- The fluid mesh consists of 636,805 nodes and 616,320 elements, with 120×96 elements on the fluid–structure interface. The fluid meshes typically contain very stretched boundary layer elements. Consequently, using an automatic mesh moving scheme [18] in this case would be a major challenge in terms of potential mesh entanglement and the need for frequent remeshing. To bypass this, we designed a structured mesh for the fluid domain where the parafoil is surrounded by concentric layers of elements (see Fig. 5). Each node in these layers is associated with a node on the parachute surface (essentially by the radial lines emanating from the surface). Based on this special mesh design, we designed the special mesh moving scheme described below.
  - The nodes in the concentric layers are moved by a fraction of the motion of the associated surface node. This fraction is unity at the surface and decays to 0 at the outer edge of the region of concentric layers.
  - The nodes outside the region of concentric layers remain fixed.

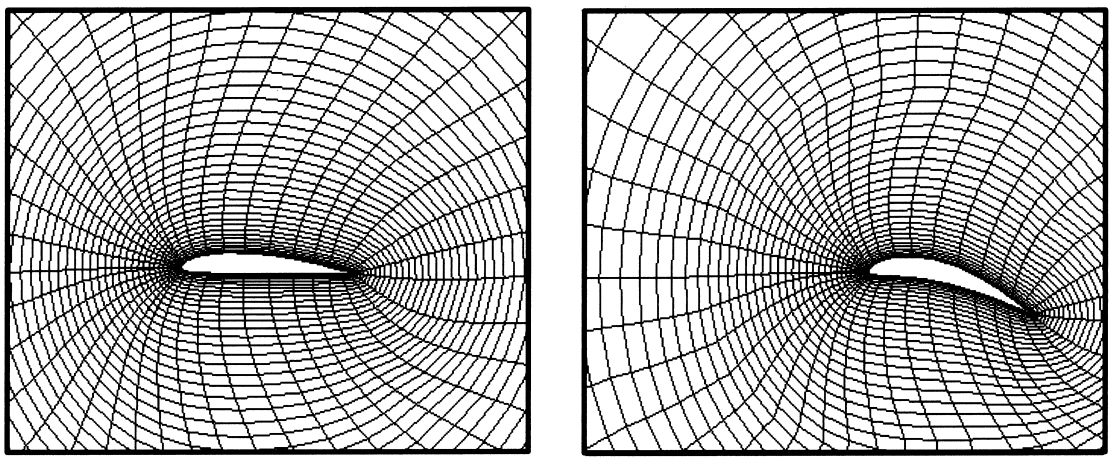

Fig. 5. FSI simulation: illustration of the algebraic mesh moving scheme with cross-sections of the fluid mesh at two instants.

- The cost involved in this mesh moving scheme is negligible. Moreover, the parallel implementation is quite efficient, as no remeshing takes place.

- Due to the nature of this special mesh moving scheme, the sides of the parachute are constrained to move in a vertical plane. Also, the side boundaries of the flow domain coincide with the sides of the parachute (see Fig. 6). For the current evaluation of our method this is not a major impediment.

In this simulation the free stream is inclined at 6^0 to the horizontal. The initial condition is a 'steady' flow field around a stress-free canopy with an assumed shape (left image in Fig. 7). The right image in Fig. 7 shows the final, deformed configuration of the parachute canopy. Fig. 8 shows the velocity vectors on two cross-sections in the deformed configuration. We observe that the largest shape changes occur at mid-cell,

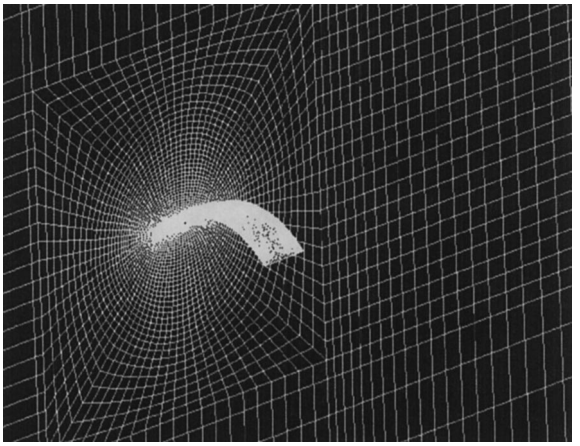

Fig. 6. FSI simulation: the fluid mesh on the parafoil surface and a side boundary. The mesh consists of 636,805 nodes and 616,320 elements, and with 120×96 elements on the fluid–structure interface.

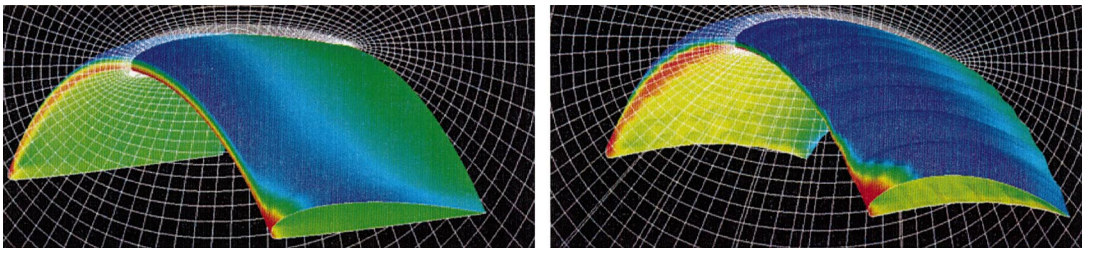

Fig. 7. FSI simulation: pressure distribution on the parafoil surface, for the initial assumed shape (left) and the deformed equilibrium shape (right).

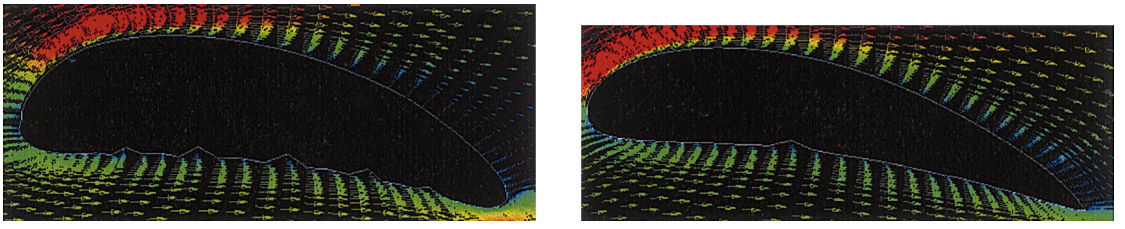

Fig. 8. FSI simulation: velocity vectors at cross-sections placed at a mid-cell (left) and at a rib (right).

Fig. 9. FSI simulation: maximum principal stress distribution on the canopy surface. Red indicates high stress, while blue indicates low stress.

and the smallest at the ribs. The lower surface of the canopy is in compression leading to the formation of 'kinks'. It is of course desirable to add a wrinkling model to compute these wrinkles more accurately. Fig. 9 shows two views of the deformed parachute structure. The canopy is colored with the maximum principal stress. The largest stresses occur at the mid-cells on the upper surface, coinciding with the maximum suction in the flow. The canopy material in our model approximately represents a heavy weight nylon cloth with a width of 0.02 in. and a breaking strength of 200 lbs/in. The average predicted principal stress is $\sim$110 lb/in.^{2}, with a maximum value of $\sim$629 lb/in^2.

In this application we also verify the capability to use fluid and structure meshes which do not match at the interface. Fig. 10 shows the fluid (left) and structure (right) surface meshes at the interface. The higher resolution fluid surface mesh accurately captures the shape from the lower resolution structure surface mesh. We verify that the aerodynamic forces and moments are balanced, and we show that in Fig. 11. The graphs show, for two test computations, time-evolution of the force and moment coefficients calculated over the fluid and structure surface meshes. For both tests, the structure mesh is the same as the one we used in the simulation described above (i.e., solid_M2 = solid_M1), with 80×48 elements on the interface. For the fluid, the mesh fluid_M1 used in the first test consists of 300,395 nodes and 288,720

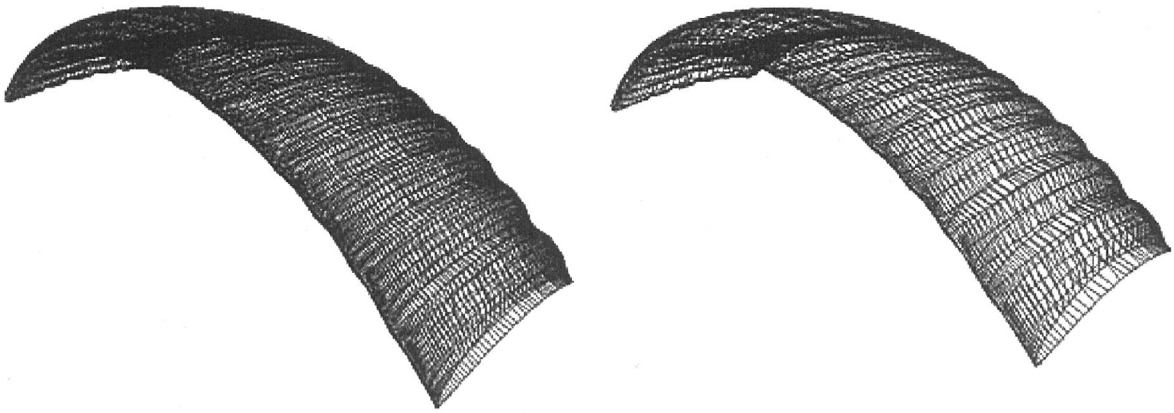

Fig. 10. Test computations with nonmatching interface meshes: the fluid surface mesh (left) has 120×96 elements, while the structure surface mesh (right) has 80×48 elements.

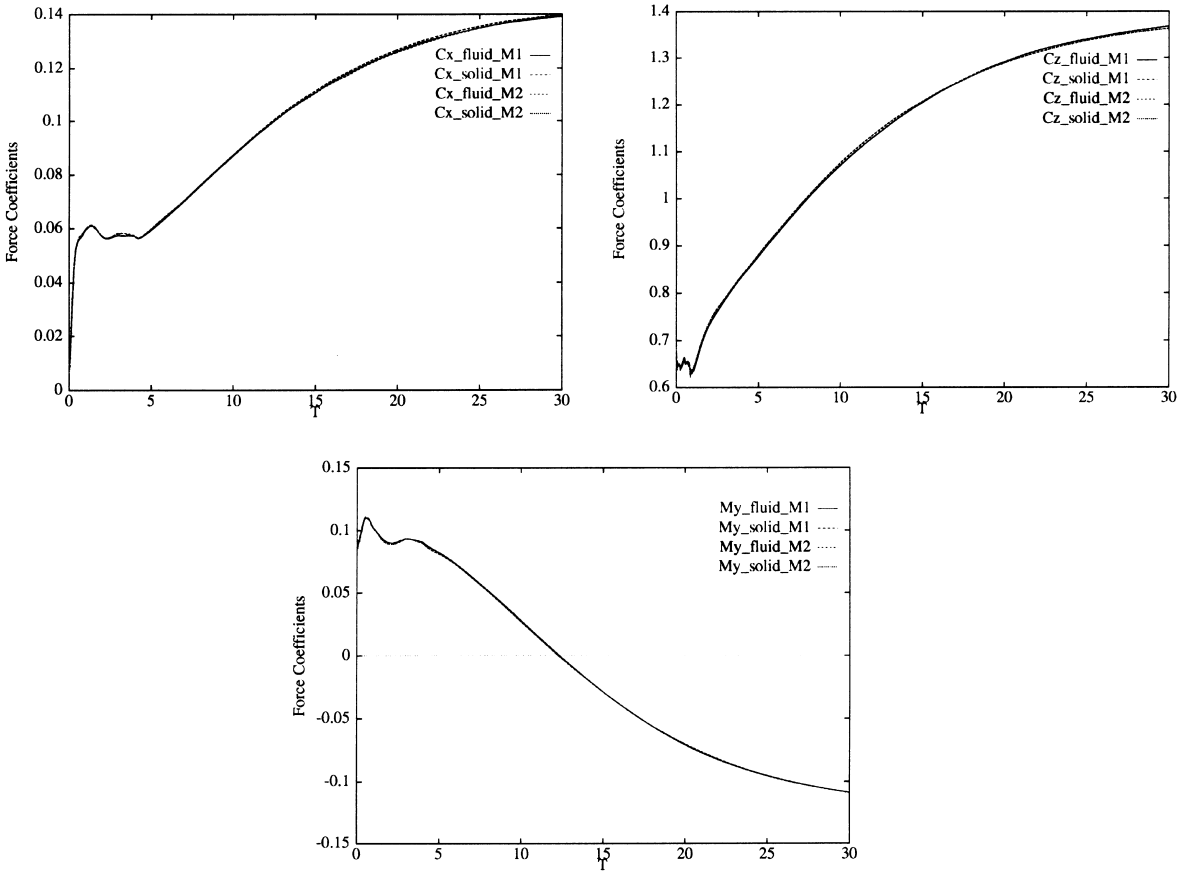

Fig. 11. Test computations with nonmatching interface meshes: the graphs show, for two test computations, time-evolution of the force and moment coefficients calculated over the fluid and structure surface meshes. For both tests, the structure mesh is the same (i.e., solid_$M2$ = solid_$M1$), with 80×48 interface elements. For the fluid, fluid_$M1$ and fluid_$M2$ have, 80×72 and 120×96 interface elements, respectively).

elements, with 80×72 interface elements. The mesh fluid_M2 used in the second test is the same as the one we used in the simulation described above (i.e., 636,805 nodes and 616,320 elements, with 120×96 interface elements).

7. Concluding remarks

We presented a finite element method for parallel 3D computation of FSI encountered in aerodynamics of parachute systems. The flow solver is based on a stabilized finite element formulation which can handle problems with moving boundaries. The structural solver is based on a finite element formulation with cable and membrane elements. Different coupling strategies, based on segregated iterative solution of the fluid and structure equations, were discussed for time-integration of the nonlinear equation systems obtained after the spatial discretization. The method presented has the flexibility of being applicable to both matching and nonmatching fluid and structure meshes at the interface. The parallel implementation was accomplished within a message-passing programming environment. The method was applied to calculate the equilibrium configuration of a ram-air parachute placed in an air stream. In our simulations, an artificial damping method was used for stable time-integration of the structure equations. Our future efforts will include investigating better damping models for time-accurate calculations and also more realistic material models for the parachute fabric.

Acknowledgements

This work was sponsored by NASA-JSC (grant NAG9-1059), AFOSR (contract F49620-98-1-0214), and by the Army High Performance Computing Research Center 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] V. Kalro, S. Aliabadi, W. Garrard, T. Tezduyar, S. Mittal, K. Stein, Parallel finite element simulation of large ram-air parachutes, Internat. J. Numer. Methods Fluids 24 (1997) 1353–1369.
[2] V. Kalro, W. Garrard, T.E. Tezduyar, Parallel finite element computation of the flare maneuver of a large ram-air parachute, AIAA-97-1427, in: 14th Aerodynamic Decelerator Systems Conference, San Francisco, 1997.
[3] T. Tezduyar, V. Kalro, W. Garrard, Parallel computational methods for 3D simulation of a parafoil with prescribed shape changes, Parallel Computing 23 (1997) 1349–1363.
[4] R.J. Benney, K.R. Stein, J.W. Leonard, M.L. Accorsi, Current 3-D structural dynamic finite element modeling capabilities, AIAA-97-1506, in: Proceedings of the 14th Aerodynamic Decelerator Technology Conference, San Francisco, 1997.
[5] K. Stein, R. Benney, V. Kalro, A. Johnson, T.E. Tezduyar Parallel computation of parachute fluid–structure interactions, AIAA-97-1505, in: 14th Aerodynamic Decelerator Systems Conference, San Francisco, 1997.
[6] 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.
[7] 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.
[8] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[9] T.J.R. Hughes, L.P. Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations, Comput. Methods Appl. Mech. Engrg. 73 (1989) 173–189. (1992)
[10] T.E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, Adv. in Appl. Mech. 28 (1991) 1–44.
[11] G. Wren, S. Ray, S. Aliabadi, T. Tezduyar, Simulation of flow problems with moving mechanical components, fluid–structure interactions and two-fluid interfaces, Internat. J. Numer. Methods Fluids 24 (1997) 1433–1448.
[12] K. Stein, R. Benney, V. Kalro, T.E. Tezduyar, J. Leonard, M. Accorsi, Parachute fluid–structure interactions: 3D computation, Comput. Methods Appl. Mech. Engrg. 190 (2000) 373–386.
[13] K.J. Bathe, Finite Element Procedures in Engineering Analysis, Prentice Hall Inc, New Jersey, 1982.
[14] A. Po, Nonlinear dynamic analysis of cable and membrane structures, Ph.D. Thesis, Oregon State University, 1982.
[15] H.M. Hilber, T.J.R. Hughes, R.L. Taylor, Improved numerical dissipation for time-integration algorithms in structural dynamics, Earthquake Engrg. Struct. Dyn. 5 (1977) 283–292.
[16] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, M. Litke, Flow simulation and high performance computing, Computational Mechanics 18 (1996) 397–412.
[17] V. Kalro, T. Tezduyar, Parallel iterative computational methods for 3D finite element flow simulations, Comput. Assisted Mech. Engrg. Sci. 5 (1998) 173–183.
[18] A.A. Johnson, T.E. Tezduyar, Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces, Comput. Methods Appl. Mech. Engrg. 119 (1994) 73–94.