Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 36 (2007) 127–135
computers & fluids
www.elsevier.com/locate/compfluid

Fluid–structure interaction modeling of complex parachute designs with the space–time finite element techniques

S. Sathe [a,*], R. Benney [b], R. Charles [b], E. Doucette [b], J. Miletti [b], M. Senga [a], K. Stein [c], T.E. Tezduyar [a]

[a] *Mechanical Engineering, Rice University, MS 321, 6100 Main Street, Houston, TX 77005, USA*
[b] *US Army Natick Soldier Center, Kansas Street, Natick, MA 10760, USA*
[c] *Department of Physics, Bethel University, 3900 Bethel Drive, St. Paul, MN 55112, USA*

Received 20 May 2005; accepted 20 July 2005
Available online 7 December 2005

Abstract

In recent years we introduced a number of enhancements to the space–time techniques we developed for computer modeling of Fluid–Structure Interaction (FSI) problems. These enhancements, which include more sophisticated fluid–structure coupling and improved mesh generation, are enabling us to address more of the computational challenges involved. Our objective here is to demonstrate the robustness of these techniques in FSI modeling of parachutes involving complex designs. As a numerical example, we have selected a conceptual parachute design with geometric complexities resembling those seen in some of the advanced parachute designs proposed and tested in recent times. We describe our FSI modeling techniques and how we compute the descent and glide performance of this conceptual parachute design.
© 2005 Elsevier Ltd. All rights reserved.

1. Introduction

Computer modeling of the Fluid–Structure Interactions (FSI) involved in parachute aerodynamics has always been challenging especially because a parachute is a light structure very sensitive to the unsteadiness of the aerodynamical forces. The FSI modeling techniques we developed to address those challenges are based on a stabilized space–time formulation. We have recently augmented these techniques with a number of enhancements to increase the robustness and scope of our FSI modeling. These enhancements include more sophisticated fluid–structure coupling techniques and improved mesh generation methods. With these enhancements, we are able to address more of the computational challenges involved in parachute FSI modeling. The challenges we can now address include the type of geometric complexities seen in some of the advanced parachute designs proposed and tested.

In our FSI modeling we prefer to use an interface-tracking technique. In this category of techniques, as the structure moves and the spatial domain occupied by the fluid changes its shape, the mesh moves to accommodate this shape change and to follow (i.e. "track") the fluid–structure interface. Moving the fluid mesh to track the interface enables us to control the mesh resolution near that interface and obtain more accurate solutions in such critical flow regions. One of the most well known examples of the interface-tracking techniques is the Arbitrary Lagrangian–Eulerian (ALE) finite element formulation [1]. The interface-tracking technique we use for discretizing the fluid dynamics equations is the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [2–4]. The stabilization is based on the Streamline-Upwind/Petrov–Galerkin (SUPG) [5,6] and

0045-7930/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2005.07.010

Pressure-Stabilizing/Petrov–Galerkin (PSPG) [2] formulations. The SUPG formulation prevents numerical instabilities that might be encountered when we have high Reynolds number and strong boundary layers. With the PSPG formulation, we can use, without numerical instabilities, equal-order interpolation functions for velocity and pressure. An earlier version of the pressure stabilization, for Stokes flows, was introduced in [7].

The DSD/SST formulation was originally developed as a general purpose interface-tracking technique for simulation of problems involving moving boundaries or interfaces, whether fluid–solid or fluid–fluid. Updating the mesh is based on moving it for as many time steps as we can and remeshing it only as frequently as we need to. The mesh moving algorithm is essentially based on the one introduced in [8], where the motion of the nodal points is governed by the equations of elasticity. The Jacobian of the transformation from the element domain to the physical domain is dropped in the finite element formulation of the elasticity equations. This is equivalent to dividing the elastic modulus by the element Jacobian and results in an increase in the stiffness of the smaller elements, which are typically placed near the fluid–structure interfaces. Mesh moving techniques with comparable features were later introduced in [9].

The DSD/SST formulation was first applied to FSI problems in [10] for 2D computation of vortex-induced vibrations of a cylinder, and in [11] for 3D computation of flow in a flexible, cantilevered pipe. In our FSI modeling of parachutes, the structural deformation is governed by the momentum equations for membranes and cables. They are solved using a semi-discrete, finite element formulation in a total Lagrangian framework (see [12]). We see no compelling reason to use a space–time formulation for the structural mechanics equations. The earliest applications of the DSD/SST formulation to FSI modeling of parachutes were reported in [13–20].

In modeling of FSI problems with the DSD/SST formulation (or any other interface-tracking technique), at each time step, we need to solve the fully-discretized, coupled fluid and structural mechanics and mesh-moving equations. What technique to use for the solution of these equations should, to some extent, depend on the nature of the application problem. With that in mind, we have developed block-iterative [11,13,16], quasi-direct [21–23], and direct coupling techniques [21–23]. The block-iterative technique gives us more flexibility in terms of algorithmic modularity and independence of the fluid and structural mechanics solvers and also better parallel efficiency. The quasi-direct and direct coupling techniques give us more robust algorithms for FSI computations where the structure is light. Various aspects of FSI modeling, including the coupling between the equations governing the fluid and structural mechanics and mesh motion, have also been addressed by many other researchers (see [24–31]) in recent years.

Our objective here is to demonstrate how we are addressing the computational challenges involved in FSI modeling of conceptual parachute designs with geometric complexities similar to those seen in some of the recently-proposed parachute designs. When the objective is to be able to quickly respond to the FSI modeling requirements of a newly-proposed parachute system, algorithmic robustness becomes the primary consideration instead of algorithmic modularity or parallel efficiency. In this paper, for FSI modeling of the conceptual parachute design we are targeting, we employ the quasi-direct coupling technique.

The governing equations are reviewed in Section 2. The finite element formulations are given in Section 3. In Section 4, we describe the modeling setup for the conceptual parachute design and present the computed results, including the descent and glide performance. We end with concluding remarks in Section 5.

2. Governing equations

2.1. Fluid mechanics

Let $\Omega_t \subset \mathbb{R}^{n_{\mathrm{sd}}}$ be the spatial domain with boundary Γ_t at time $t \in (0, T)$. The subscript t indicates the time-dependence of the domain. The Navier–Stokes equations of incompressible flows are written on Ω_t and $\forall t \in (0, T)$ as

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

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where ρ, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, respectively. The stress tensor $\boldsymbol{\sigma}$ is defined as $\boldsymbol{\sigma}(p,\mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u})$, with $\boldsymbol{\varepsilon}(\mathbf{u}) = ((\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}})/2$. Here p is the pressure, $\mathbf{I}$ is the identity tensor, $\mu = \rho\nu$ is the viscosity, ν is the kinematic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor. The essential and natural boundary conditions for Eq. (1) are represented as $\mathbf{u} = \mathbf{g}$ on $(\Gamma_t)_g$ and $\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h}$ on $(\Gamma_t)_h$, where $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field $\mathbf{u}_0(\mathbf{x})$ is specified as the initial condition.

2.2. Structural mechanics

Let $\Omega_t^{\mathrm{s}} \subset \mathbb{R}^{n_{\mathrm{xd}}}$ be the spatial domain with boundary Γ_t^{s}, where $n_{\mathrm{xd}} = 2$ for membranes and $n_{\mathrm{xd}} = 1$ for cables. The parts of Γ_t^{s} corresponding to the essential and natural boundary conditions are represented by $(\Gamma_t^{\mathrm{s}})_g$ and

$(\Gamma_t^{\mathrm{s}})_h$. The superscript "s" indicates the structure. The equations of motion are written as

$$\rho^{\mathrm{s}}\left(\frac{\mathrm{d}^2\mathbf{y}}{\mathrm{d}t^2}+\eta\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t}-\mathbf{f}^{\mathrm{s}}\right)-\nabla\cdot\boldsymbol{\sigma}^{\mathrm{s}}=\mathbf{0}, \tag{3}$$

where ρ^{s}, $\mathbf{y}$, $\mathbf{f}^{\mathrm{s}}$ and $\boldsymbol{\sigma}^{\mathrm{s}}$ are the material density, structural displacement, external force and the Cauchy stress tensor [32,33], respectively. Here η is the mass-proportional damping coefficient. The damping provides additional stability and can be used where time-accuracy is not required, such as in determining the deformed shape of the structure for specified fluid mechanics forces acting on it. The stresses are expressed in terms of the 2nd Piola–Kirchoff stress tensor $\mathbf{S}$, which is related to the Cauchy stress tensor through a kinematic transformation. Under the assumption of large displacements and rotations, small strains, and no material damping, the membranes and cables are treated as Hookean materials with linear elastic properties. For membranes, under the assumption of plane stress, $\mathbf{S}$ becomes (see [12]):

$$S^{ij}=\left(\bar{\lambda}_m G^{ij}G^{kl}+\mu_m[G^{il}G^{jk}+G^{ik}G^{jl}]\right)E_{kl}, \tag{4}$$

where for the case of isotropic plane stress $\bar{\lambda}_m=2\lambda_m\mu_m/(\lambda_m+2\mu_m)$. Here, E_{kl} are the components of the Cauchy–Green strain tensor, G^{ij} are the components of the contravariant metric tensor in the original configuration, and λ_m and μ_m are the Lamé constants. For cables, under the assumption of uniaxial tension, $\mathbf{S}$ becomes $S^{11}=E_{\mathrm{c}}G^{11}G^{11}E_{11}$, where E_{c} is the Young's modulus for the cable.

3. Finite element formulations

3.1. DSD/SST formulation of fluid mechanics

The DSD/SST formulation [2–4], which was originally based on a Galerkin/least-squares method, is written over a sequence of N space–time slabs Q_n, where Q_n is the slice of the space–time domain between the time levels t_n and t_{n+1}. At each time step, the integrations are performed over Q_n. The space–time finite element interpolation functions are continuous within a space–time slab, but discontinuous from one space–time slab to another. The notation $(\cdot)_n^-$ and $(\cdot)_n^+$ denotes the function values at t_n as approached from below and above. Each Q_n is decomposed into elements Q_n^e, where $e=1,2,\ldots,(n_{\mathrm{el}})_n$. The subscript n used with n_{el} is for the general case in which the number of space–time elements may change from one space–time slab to another. The essential and natural boundary conditions are imposed over $(P_n)_g$ and $(P_n)_h$, the complementary subsets of the lateral boundary of the space–time slab. The finite element trial function spaces $(\mathscr{S}_{\mathbf{u}}^h)_n$ for velocity and $(\mathscr{S}_p^h)_n$ for pressure, and the test function spaces $(\mathscr{V}_{\mathbf{u}}^h)_n$ and $(\mathscr{V}_p^h)_n$ $(=(\mathscr{S}_p^h)_n)$ are defined by using, over Q_n, first-order polynomials in space and time. The version of the DSD/SST formulation introduced in [34] is based on SUPG/PSPG stabilization and can be written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h\in(\mathscr{S}_{\mathbf{u}}^h)_n$ and $p^h\in(\mathscr{S}_p^h)_n$ such that $\forall\mathbf{w}^h\in(\mathscr{V}_{\mathbf{u}}^h)_n$ and $\forall q^h\in(\mathscr{V}_p^h)_n$:

$$\begin{aligned}
&\int_{Q_n}\mathbf{w}^h\cdot\rho\left(\frac{\partial\mathbf{u}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{u}^h-\mathbf{f}^h\right)\mathrm{d}Q\\
&+\int_{Q_n}\boldsymbol{\varepsilon}(\mathbf{w}^h):\boldsymbol{\sigma}(p^h,\mathbf{u}^h)\,\mathrm{d}Q-\int_{(P_n)_h}\mathbf{w}^h\cdot\mathbf{h}^h\,\mathrm{d}P\\
&+\int_{Q_n}q^h\nabla\cdot\mathbf{u}^h\,\mathrm{d}Q+\int_{\Omega_n}(\mathbf{w}^h)_n^+\cdot\rho\left((\mathbf{u}^h)_n^+-(\mathbf{u}^h)_n^-\right)\mathrm{d}\Omega\\
&+\sum_{e=1}^{(n_{\mathrm{el}})_n}\int_{Q_n^e}\frac{1}{\rho}\left[\tau_{\mathrm{SUPG}}\rho\left(\frac{\partial\mathbf{w}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{w}^h\right)+\tau_{\mathrm{PSPG}}\nabla q^h\right]\\
&\cdot[\text{Ł}(p^h,\mathbf{u}^h)-\rho\mathbf{f}^h]\,\mathrm{d}Q+\sum_{e=1}^{(n_{\mathrm{el}})_n}\int_{Q_n^e}\nu_{\mathrm{LSIC}}\nabla\cdot\mathbf{w}^h\rho\nabla\cdot\mathbf{u}^h\,\mathrm{d}Q=0,
\end{aligned} \tag{5}$$

where

$$\text{Ł}(q^h,\mathbf{w}^h)=\rho\left(\frac{\partial\mathbf{w}^h}{\partial t}+\mathbf{u}^h\cdot\nabla\mathbf{w}^h\right)-\nabla\cdot\boldsymbol{\sigma}(q^h,\mathbf{w}^h). \tag{6}$$

Here τ_{SUPG}, τ_{PSPG} and ν_{LSIC} are the SUPG, PSPG and LSIC (least-squares on incompressibility constraint) stabilization parameters. This formulation is applied to all space–time slabs $Q_0,Q_1,Q_2,\ldots,Q_{N-1}$, starting with $(\mathbf{u}^h)_0^-=\mathbf{u}_0$.

There are various ways of defining the stabilization parameters. The definitions given in [34] are used in the computations reported in this paper:

$$\tau_{\mathrm{SUPG}}=\left(\frac{1}{\tau_{\mathrm{SUGN12}}^2}+\frac{1}{\tau_{\mathrm{SUGN3}}^2}\right)^{-\frac{1}{2}}, \tag{7}$$

$$\tau_{\mathrm{SUGN12}}=\left(\sum_{a=1}^{n_{\mathrm{en}}}\left|\frac{\partial N_a}{\partial t}+\mathbf{u}^h\cdot\nabla N_a\right|\right)^{-1},$$

$$\tau_{\mathrm{SUGN3}}=\frac{h_{\mathrm{RGN}}^2}{4\nu}, \tag{8}$$

$$h_{\mathrm{RGN}}=2\left(\sum_{a=1}^{n_{\mathrm{en}}}|\mathbf{r}\cdot\nabla N_a|\right)^{-1},\quad \mathbf{r}=\frac{\nabla\|\mathbf{u}^h\|}{\|\nabla\|\mathbf{u}^h\|\|}, \tag{9}$$

$$\tau_{\mathrm{PSPG}}=\tau_{\mathrm{SUPG}}, \tag{10}$$

$$\nu_{\mathrm{LSIC}}=\tau_{\mathrm{SUPG}}\|\mathbf{u}^h\|^2. \tag{11}$$

For more ways of calculating τ_{SUPG}, τ_{PSPG} and ν_{LSIC}, see [35,34,36].

3.2. Semi-discrete formulation and time-integration of structural mechanics

With $\mathbf{y}^h$ and $\mathbf{w}^h$ coming from appropriately defined trial and test function spaces, respectively, the semi-discrete finite element formulation of the structural mechanics equations are written as

$$\int_{\Omega_0^s} \mathbf{w}^h \cdot \rho^s \frac{\mathrm{d}^2\mathbf{y}^h}{\mathrm{d}t^2}\,\mathrm{d}\Omega^s + \int_{\Omega_0^s} \mathbf{w}^h \cdot \eta\rho^s \frac{\mathrm{d}\mathbf{y}^h}{\mathrm{d}t}\,\mathrm{d}\Omega^s + \int_{\Omega_0^s} \delta\mathbf{E}^h : \mathbf{S}^h\,\mathrm{d}\Omega^s = \int_{\Omega_t^s} \mathbf{w}^h \cdot \left(\mathbf{t}^h + \rho^s\mathbf{f}^s\right)\mathrm{d}\Omega^s. \tag{12}$$

The fluid mechanics forces acting on the structure are represented by vector $\mathbf{t}^h$. This term is geometrically nonlinear and thus increases the overall nonlinearity of the formulation. The left-hand-side terms of Eq. (12) are referred to in the original configuration and the right-hand-side terms in the deformed configuration at time t. From this formulation, at each time step, we obtain a nonlinear system of equations. In solving that nonlinear system with an iterative method, we use the following incremental form:

$$\left[\frac{\mathbf{M}}{\beta\Delta t^2} + \frac{(1-\alpha)\gamma\mathbf{C}}{\beta\Delta t} + (1-\alpha)\mathbf{K}\right]\Delta\mathbf{d}^i = \mathbf{R}^i, \tag{13}$$

where $\mathbf{C} = \eta\mathbf{M}$. Here $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the consistent tangent stiffness matrix associated with the internal elastic forces, $\mathbf{C}$ is a damping matrix, $\mathbf{R}^i$ is the residual vector at the ith iteration, and $\Delta\mathbf{d}^i$ is the ith increment in the nodal displacements vector $\mathbf{d}$. The damping matrix $\mathbf{C}$ is used only in stand-alone structural mechanics computations with specified fluid mechanics forces, while establishing a starting shape for the FSI computations. In Eq. (13), all of the terms known from the previous iteration are collected into the residual vector $\mathbf{R}^i$. The parameters α, β, γ are part of the Hilber–Hughes–Taylor [37] time-integration scheme used here.

3.3. Quasi-direct coupling

The formulation for the quasi-direct coupling approach [21–23] is written as follows:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}_{1\mathrm{E}}^h \cdot \rho\left(\frac{\partial\mathbf{u}^h}{\partial t} + \mathbf{u}^h\cdot\nabla\mathbf{u}^h - \mathbf{f}^h\right)\mathrm{d}Q \\
&+ \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}_{1\mathrm{E}}^h) : \boldsymbol{\sigma}(p^h,\mathbf{u}^h)\,\mathrm{d}Q - \int_{(P_n)_\mathrm{h}} \mathbf{w}_{1\mathrm{E}}^h \cdot \mathbf{h}_{1\mathrm{E}}^h\,\mathrm{d}P \\
&+ \int_{Q_n} q_{1\mathrm{E}}^h \nabla\cdot\mathbf{u}^h\,\mathrm{d}Q + \int_{\Omega_n} (\mathbf{w}_{1\mathrm{E}}^h)_n^+ \cdot \rho\left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-\right)\mathrm{d}\Omega \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{1}{\rho}\left[\tau_\mathrm{SUPG}\rho\left(\frac{\partial\mathbf{w}_{1\mathrm{E}}^h}{\partial t} + \mathbf{u}^h\cdot\nabla\mathbf{w}_{1\mathrm{E}}^h\right) + \tau_\mathrm{PSPG}\nabla q_{1\mathrm{E}}^h\right] \\
&\cdot\left[Ł(p^h,\mathbf{u}^h) - \rho\mathbf{f}^h\right]\mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC}\nabla\cdot\mathbf{w}_{1\mathrm{E}}^h \rho\nabla\cdot\mathbf{u}^h\,\mathrm{d}Q = 0,
\end{aligned} \tag{14}$$

$$\begin{aligned}
&\int_{Q_n} q_{1\mathrm{I}}^h \nabla\cdot\mathbf{u}^h\,\mathrm{d}Q + \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{1}{\rho}\left[\tau_\mathrm{PSPG}\nabla q_{1\mathrm{I}}^h\right] \\
&\cdot\left[Ł(p^h,\mathbf{u}^h) - \rho\mathbf{f}^h\right]\mathrm{d}Q = 0,
\end{aligned} \tag{15}$$

$$\int_{\Gamma_{1\mathrm{I}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \left((\mathbf{u}_{1\mathrm{I}}^h)_{n+1}^- - \mathbf{u}_{2\mathrm{I}}^h\right)\mathrm{d}\Gamma = 0, \tag{16}$$

$$\begin{aligned}
&\int_{Q_n} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \rho\left(\frac{\partial\mathbf{u}^h}{\partial t} + \mathbf{u}^h\cdot\nabla\mathbf{u}^h - \mathbf{f}^h\right)\mathrm{d}Q \\
&+ \int_{Q_n} \boldsymbol{\varepsilon}\left((\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^-\right) : \boldsymbol{\sigma}(p^h,\mathbf{u}^h)\,\mathrm{d}Q - \int_{(P_n)_h} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \mathbf{h}_{1\mathrm{I}}^h\,\mathrm{d}P \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \frac{1}{\rho}\left[\tau_\mathrm{SUPG}\rho\left(\frac{\partial(\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^-}{\partial t} + \mathbf{u}^h\cdot\nabla(\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^-\right)\right] \\
&\cdot\left[Ł(p^h,\mathbf{u}^h) - \rho\mathbf{f}^h\right]\mathrm{d}Q \\
&+ \sum_{e=1}^{(n_\mathrm{el})_n} \int_{Q_n^e} \nu_\mathrm{LSIC}\nabla\cdot(\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \rho\nabla\cdot\mathbf{u}^h\,\mathrm{d}Q = 0,
\end{aligned} \tag{17}$$

$$\begin{aligned}
&\int_{(\Omega_2)_0} \mathbf{w}_2^h \cdot \rho_2 \frac{\mathrm{d}^2\mathbf{y}^h}{\mathrm{d}t^2}\,\mathrm{d}\Omega + \int_{(\Omega_2)_0} \mathbf{w}_2^h \cdot \eta\rho_2 \frac{d\mathbf{y}^h}{\mathrm{d}t}\,\mathrm{d}\Omega \\
&+ \int_{(\Omega_2)_0} \delta\mathbf{E}^h : \mathbf{S}^h\,\mathrm{d}\Omega = \int_{\Omega_2} \mathbf{w}_2^h \cdot \rho_2\mathbf{f}_2^h\,\mathrm{d}\Omega \\
&+ \int_{\Omega_{2\mathrm{E}}} \mathbf{w}_{2\mathrm{E}}^h \cdot \mathbf{h}_{2\mathrm{E}}^h\,\mathrm{d}\Omega - \int_{\Omega_{2\mathrm{I}}} \mathbf{w}_{2\mathrm{I}}^h \cdot \mathbf{h}_{1\mathrm{I}}^h\,\mathrm{d}\Omega.
\end{aligned} \tag{18}$$

The notation used here is slightly different from that used in Eqs. (5) and (12). Here, subscripts 1 and 2 are used for fluid and structure respectively, and I and E indicate the "interface" and "elsewhere in the domain", respectively. The traction force is denoted by $\mathbf{h}$. In this formulation, $(\mathbf{u}_{1\mathrm{I}}^h)_{n+1}^-$ and $\mathbf{h}_{1\mathrm{I}}^h$ (the fluid velocity and stress at the interface) are treated as separate unknowns, and Eqs. (16) and (17) can be seen as equations corresponding to these two unknowns, respectively. The structural displacement rate at the interface, $\mathbf{u}_{2\mathrm{I}}^h$, is derived from $\mathbf{y}^h$. This formulation allows for cases where the fluid and structure meshes at the interface are not identical. If they are identical, the same formulation can still be used, but one can also use its reduced version where Eq. (16) is no longer needed and $\mathbf{h}_{1\mathrm{I}}^h$ is no longer treated as a separate unknown.

Remark 1. Full discretizations of the quasi-direct formulation described by Eqs. (14)–(18) lead to coupled nonlinear equation systems that need to be solved at every time step. These coupled nonlinear equation systems can be written as follows:

$$\mathbf{N}_1(\mathbf{d}_1,\mathbf{d}_2) = \mathbf{F}_1, \tag{19}$$

$$\mathbf{N}_2(\mathbf{d}_1,\mathbf{d}_2) = \mathbf{F}_2, \tag{20}$$

where subscripts 1 and 2, again, denote the fluid and structure, respectively. Solution of these equations with the Newton–Raphson method would necessitate at every Newton–Raphson step solution of the following linear equation system:

$$\mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 = \mathbf{b}_1, \tag{21}$$

$$\mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 = \mathbf{b}_2, \tag{22}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the residuals of the nonlinear equations, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the correction increments for $\mathbf{d}_1$ and $\mathbf{d}_2$, and $\mathbf{A}_{\beta\gamma} = \partial\mathbf{N}_\beta/\partial\mathbf{d}_\gamma$. Careful inspection of Eq. (17) re-

veals that we could have convergence difficulties when the stabilization terms, which only contribute to the off-diagonal entries of $\mathbf{A}$, are dominant. In such cases, convergence of the iterative solution can be improved by activating the shortcut technique proposed in [36,38]. In this technique, to reduce "over-correcting" (i.e. "over-incrementing") the structural displacements, we increase the mass matrix contribution to $\mathbf{A}_{22}$. This is achieved without altering $\mathbf{b}_1$ or $\mathbf{b}_2$ (i.e. $\mathbf{F}_1 - \mathbf{N}_1(\mathbf{d}_1, \mathbf{d}_2)$ or $\mathbf{F}_2 - \mathbf{N}_2(\mathbf{d}_1, \mathbf{d}_2)$), and therefore when the Newton–Raphson iterations converge, they converge to the solution of the problem with the correct structural mass.

Remark 2. A slightly altered version of the formulation given by Eqs. (14)–(18) was introduced in [39] by suppressing the stabilization terms in Eq. (17):

$$\int_{Q_n} \left(\mathbf{w}^h_{1\mathrm{I}}\right)^-_{n+1} \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h\right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}\left(\left(\mathbf{w}^h_{1\mathrm{I}}\right)^-_{n+1}\right) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h)\,\mathrm{d}Q - \int_{(P_n)_h} \left(\mathbf{w}^h_{1\mathrm{I}}\right)^-_{n+1} \cdot \mathbf{h}^h_{1\mathrm{I}}\,\mathrm{d}P = 0. \tag{23}$$

Our recent experiences show that this leads to a more robust solution algorithm, and good convergence performance can be achieved without augmenting $\mathbf{A}_{22}$ as described in the earlier remark.

4. Test computations

4.1. Model setup

As the test case, we have selected a conceptual parachute design with geometric complexities similar to those seen in some of the advanced parachute designs proposed and tested in recent times. The shape and dimensions of the unstressed parachute are shown in Fig. 1. It is in the shape of a cross formed by two bands that are each 46.9 ft long and 21.1 ft wide. The adjacent arms of the "cross" are stitched together along the edges marked with red color in Fig. 1. The stitching forms four "vents" with edges marked with blue color. The edges of the skirt, marked with green color, are attached to 28 suspension lines, each 21.0 ft long. The suspension lines connect to 4 risers, each 4.0 ft long. The risers connect to the payload. The parachute is modeled with membranes and cables, and the payload is modeled with a point mass.

The membrane stiffness, density, and thickness are set to 2.0×10^5 lb/ft^2, 2.374 slugs/ft^3, and 1.0×10^{-4} ft, respectively. The cable stiffness, density, and cross-sectional area are set to 1.117×10^7 lb/ft^2, 2.374 slugs/ft^3, and 3.358×10^{-5} ft^2. For air, as the density and kinematic-viscosity values we use 2.378×10^{-3} slugs/ft^3 and 1.615×10^{-4} ft^2/s. The Reynolds number, based on a descent speed of 16 ft/s and a length scale of 21.1 ft, is 2.1×10^6.

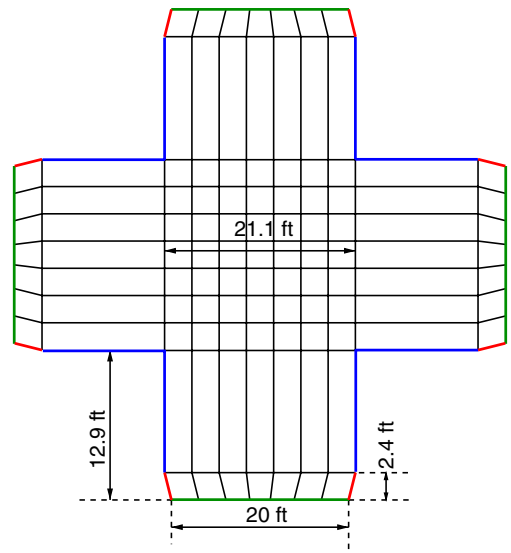

Fig. 1. Shape and dimensions of unstressed parachute. The adjacent arms of the "cross" are stitched together along the edges marked with red color (For interpretation of the references in colour in this figure legend, the reader is referred to the Web version of this article.)

The parachute canopy structure is discretized using a finite element mesh with 10,973 nodes and 21,504 elements. A constant air pressure is applied to inflate the parachute to its initial shape, which is determined with a stand-alone structural mechanics computation. The inflation-pressure is set to 1.3 times the stagnation

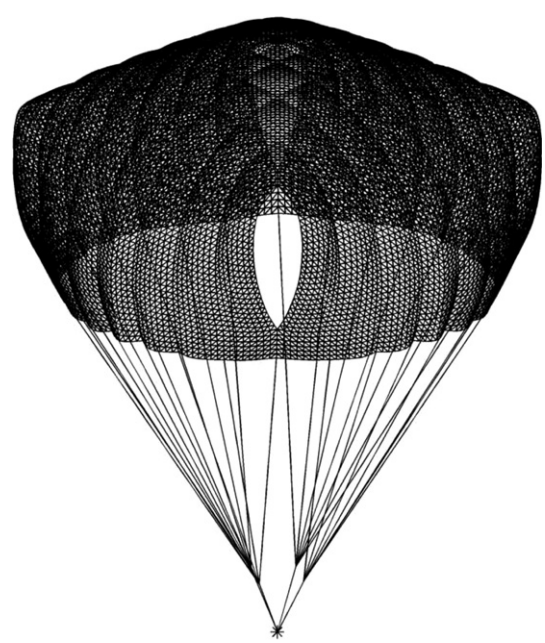

Fig. 2. Inflated parachute.

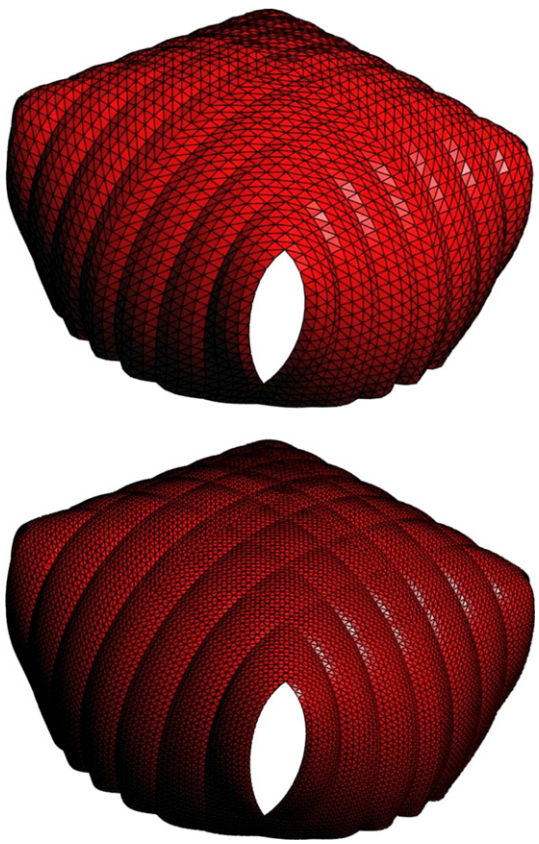

Fig. 3. Fluid (top) and structure (bottom) meshes at the interface.

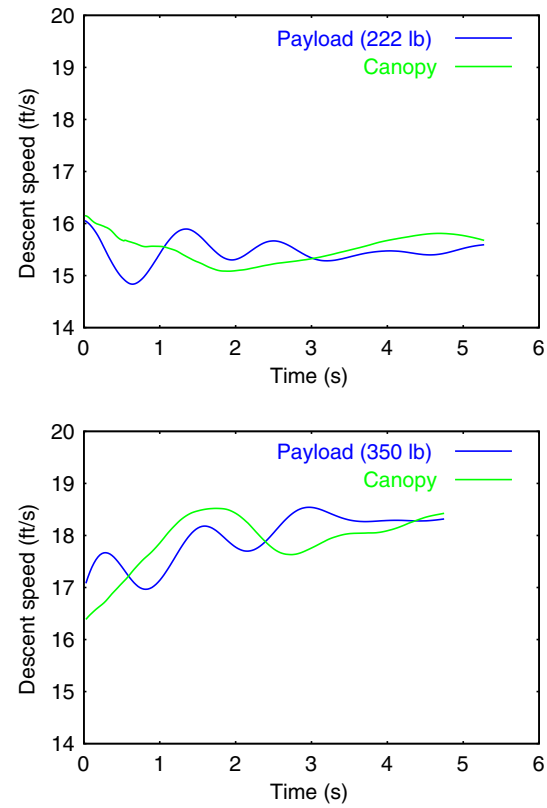

Fig. 4. Payload and canopy descent speeds with payload weighing 222 lb (top) and 350 lb (bottom).

pressure, calculated based on the descent speed of 16 ft/s. Fig. 2 shows the inflated parachute.

To limit the number of unknowns in the fluid mechanics part of the problem, we use a coarser fluid mechanics mesh at the fluid–structure interface. The interface mesh for the fluid has 2797 nodes and 5376 elements. Fig. 3 shows the fluid and structure meshes at the interface. We note that the quasi-direct coupling approach described in Section 3.3 allows for incompatible fluid and structure meshes at the interface. The volume meshes generated for the fluid typically have approximately 100,000 nodes and 600,000 elements.

4.2. Computation of the descent

In this set of computations we test two different payload weights: 222 lb and 350 lb. We use the quasi-direct coupling approach described by Eqs. (14)–(18). To further improve the convergence of the iterative solution, we activate the shortcut technique described in Section 3.3. In doing that, the mass matrix contribution to $\mathbf{A}_{22}$ increased by a factor 3.0. Both computations were carried out without any remeshing.

Fig. 4 shows the payload and canopy descent speeds with payload weighing 222 lb and 350 lb. As seen from the plots, the parachute system reaches a near steady descent speed of 15.7 ft/s with payload weighing 222 lb and 18.3 ft/s with payload weighing 350 lb.

4.3. Computation of the gliding

We compute the gliding performance of the parachute system with the payload weighing 222 lb. We attain gliding in the intended direction (x_1 direction) by pulling one of the risers by 2.0 ft and affecting a canopy shape change that generates a force in the x_1 direction. The parachute

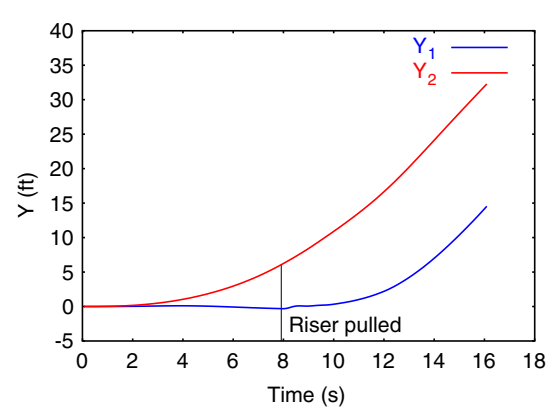

Fig. 5. Payload position as a function of time. The symbols Y_1 and Y_2 denote the payload position along the x_1 and x_2 directions. The intended glide direction is the x_1 direction.

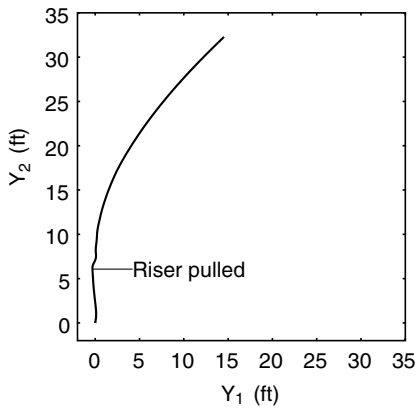

Fig. 6. Payload trajectory in the Y_1–Y_2 plane.

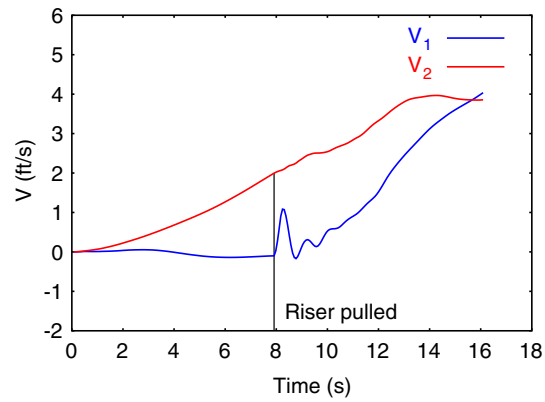

Fig. 7. Payload velocity as a function of time. The symbols V_1 and V_2 denote the payload velocity in the x_1 and x_2 directions. The intended glide direction is the x_1 direction.

is descending in the x_3 direction. The riser is pulled rather quickly, within a period less than 0.3 s. In this FSI computation, we use the quasi-direct coupling approach described by Eqs. (14)–(16), Eqs. (23) and (18). The $\frac{\partial N_a}{\partial t}$ term is dropped from the definition of τ_{SUGN12} given by Eq. (8). For fluid mechanics volume-mesh generation, we use an advanced automatic mesh generator [40] with improved mesh generation capabilities critical to parachute FSI computations. The computation was carried out without any remeshing.

Fig. 5 shows the payload position along the x_1 direction, denoted by the symbol Y_1, as a function of time. We also see, even before the riser is pulled, a glide in an unintended direction (x_2 direction). We believe that this unintended glide direction is selected somewhat randomly, very much like what typically happens in a bifurcation problem, from among the possible directions. The payload position along the x_2 direction, denoted by the symbol Y_2, is also shown in Fig. 5 as a function

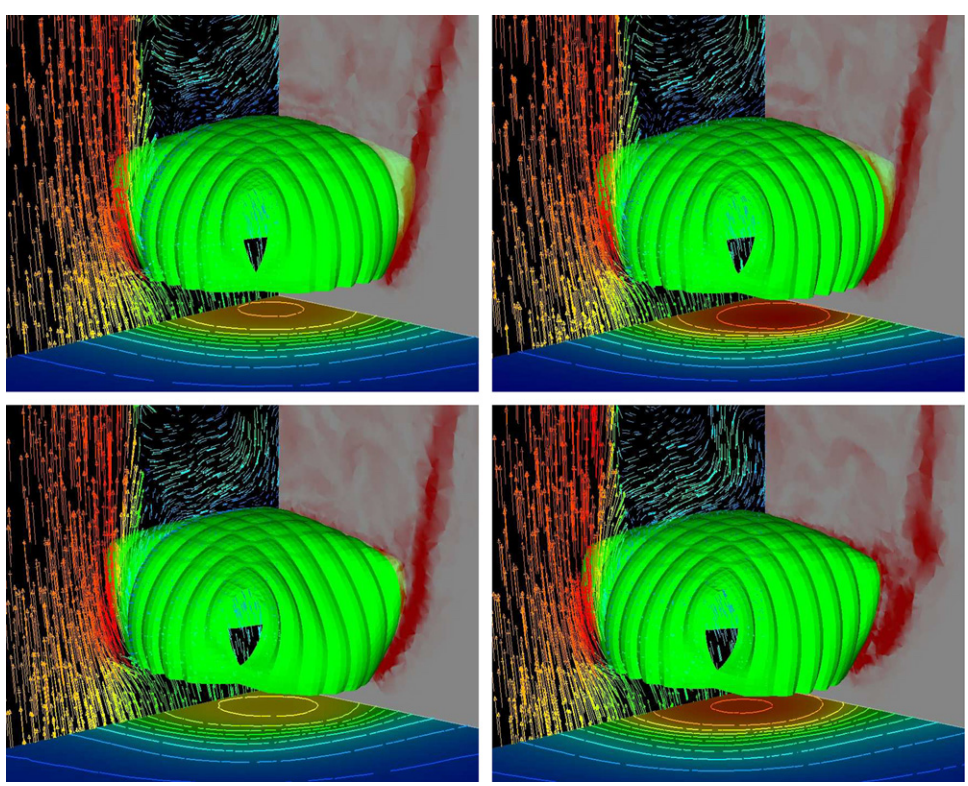

Fig. 8. Flow conditions just before (top-left) and just after (top-right) the riser is pulled, as well as at two instants (bottom-left and bottom-right) during the subsequent period. In each frame, the left, right, and bottom planes show, respectively, the velocity vectors colored with their magnitudes, vorticity, and pressure.

of time. Fig. 6 shows the payload trajectory in the Y_1–Y_2 plane. Fig. 7 shows the payload velocity, as a function time, in the x_1 and x_2 directions. We clearly see an increase in the glide velocity in the intended direction after the riser is pulled. Fig. 8 shows the flow conditions just before and just after the riser is pulled, as well as at two instants during the subsequent period.

5. Concluding remarks

We have described how we are addressing the computational challenges involved in FSI modeling of conceptual parachute designs with geometric complexities similar to those seen in some of the recently-proposed parachute designs. The core method in our FSI modeling is the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation, developed originally for flow problems with moving boundaries and interfaces. A mesh moving method that reduces the need for remeshing is also an important component of our overall computational strategy. In recent years we added a number of enhancements to this computational strategy, including more sophisticated fluid–structure coupling techniques and improved mesh generation. The quasi-direct coupling technique we used for the numerical examples of this paper gives us more algorithmic robustness. This is particularly important for better convergence of the iterative solution process when the structure is light and very sensitive to the unsteadiness of the aerodynamical forces. The improved mesh generation capability is playing an important role in addressing the challenges involved in representing the geometrically complex parachute canopies. As test cases with the conceptual parachute design we considered in this paper, we computed its decent for two different payload weights, as well as its glide, which was induced by pulling one of the risers. In addition to the glide in the intended direction, we also observed, even before the riser was pulled, a glide in an unintended direction. At this time, we believe that this unintended glide direction is selected somewhat randomly from among a set of possible directions. Better understanding of that unintended glide and its interaction with the intended glide will be among the topics we plan to study further in the future. We believe that such "physical explorations" will be more and more within our reach, because we are continuously increasing the robustness and scope of our FSI modeling and we are now able to address more of the computational challenges involved.

Acknowledgements

This work was supported by the US Army Natick Soldier Center (Contract No. DAAD16-03-C-0051), NSF (Grant No. EIA-0116289), and NASA Johnson Space Center (Grant No. NAG9-1435). We are grateful to Professor Genki Yagawa (Toyo University, Japan) and Mr. Masakazu Inaba for their help in using the advanced automatic mesh generator [40] mentioned in Section 4.3.

References

[1] Hughes TJR, Liu WK, Zimmermann TK. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. Comput Methods Appl Mech Eng 1981;29:329–49.
[2] Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. Adv Appl Mech 1992;28:1–44.
[3] Tezduyar TE, Behr M, Liou J. 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 numerical tests. Comput Methods Appl Mech Eng 1992;94:339–51.
[4] Tezduyar TE, Behr M, Mittal S, Liou J. 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 Eng 1992;94:353–71.
[5] Hughes TJR, Brooks AN. A multi-dimensional upwind scheme with no crosswind diffusion. In: Hughes TJR, editor. Finite element methods for convection dominated flows, AMD-vol. 34. New York: ASME; 1979. p. 19–35.
[6] Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. Comput Methods Appl Mech Eng 1982;32:199–259.
[7] Hughes TJR, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. Comput Methods Appl Mech Eng 1986;59:85–99.
[8] Tezduyar TE, Behr M, Mittal S, Johnson AA. Computation of unsteady incompressible flows with the finite element methods—space–time formulations, iterative strategies and massively parallel implementations. New methods in transient analysis, PVP-vol. 246/AMD-vol. 143. New York: ASME; 1992. p. 7–24.
[9] Masud A, Hughes TJR. A space–time Galerkin/least-squares finite element formulation of the Navier–Stokes equations for moving domain problems. Comput Methods Appl Mech Eng 1997;146:91–126.
[10] Mittal S, Tezduyar TE. A finite element study of incompressible flows past oscillating cylinders and aerofoils. Int J Numer Methods Fluids 1992;15:1073–118.
[11] Mittal S, Tezduyar TE. Parallel finite element simulation of 3D incompressible flows—fluid–structure interactions. Int J Numer Methods Fluids 1995;21:933–53.
[12] Accorsi ML, Leonard JW, Benney R, Stein K. Structural modeling of parachute dynamics. AIAA J 2000;38:139–46.
[13] Stein KR, Benney RJ, Kalro V, Johnson AA, Tezduyar TE. Parallel computation of parachute fluid–structure interactions. In: Proceedings of AIAA 14th aerodynamic decelerator systems technology conference, AIAA Paper 97-1505, San Francisco, California, 1997.
[14] Stein K, Benney R, Tezduyar T, Kalro V, Leonard J, Accorsi M. 3-D computation of parachute fluid–structure interactions: performance and control. In: Proceedings of CEAS/AIAA 15th aerodynamic decelerator systems technology conference, AIAA Paper 99-1714, Toulouse, France, 1999.

[15] Stein K, Benney R, Tezduyar T, Kalro V, Potvin J, Bretl T. 3-D computation of parachute fluid–structure interactions: performance and control. In: Proceedings of CEAS/AIAA 15th aerodynamic decelerator systems technology conference, AIAA Paper 99-1725, Toulouse, France, 1999.
[16] Stein K, Benney R, Kalro V, Tezduyar TE, Leonard J, Accorsi M. Parachute fluid–structure interactions: 3-D Computation. Comput Methods Appl Mech Eng 2000;190:373–86.
[17] Kalro V, Tezduyar TE. A parallel 3D computational method for fluid–structure interactions in parachute systems. Comput Methods Appl Mech Eng 2000;190:321–32.
[18] Stein K, Benney R, Tezduyar T, Potvin J. Fluid–structure interactions of a cross parachute: numerical simulation. Comput Methods Appl Mech Eng 2001;191:673–87.
[19] Tezduyar T, Osawa Y. The multi-domain method for computation of the aerodynamics of a parachute crossing the far wake of an aircraft. Comput Methods Appl Mech Eng 2001;191:705–16.
[20] Stein KR, Benney RJ, Tezduyar TE, Leonard JW, Accorsi ML. Fluid–structure interactions of a round parachute: modeling and simulation techniques. J Aircraft 2001;38:800–8.
[21] Tezduyar TE, Sathe S, Keedy R, Stein K. Space–time techniques for finite element computation of flows with moving boundaries and interfaces. In: Gallegos S, Herrera I, Botello S, Zarate F, Ayala G, editors. Proceedings of the III international congress on numerical methods in engineering and applied science, CD-ROM, 2004.
[22] Tezduyar TE, Sathe S, Keedy R, Stein K. Space–time finite element techniques for computation of fluid–structure interactions, Comput Methods Appl Mech Eng, in press.
[23] Sathe S. Enhanced-discretization and solution techniques in flow simulations and parachute fluid–structure interactions. Ph.D. thesis, Rice University, 2004.
[24] Ohayon R. Reduced symmetric models for modal analysis of internal structural–acoustic and hydroelastic–sloshing systems. Comput Methods Appl Mech Eng 2001;190:3009–19.
[25] Wall W. Fluid–structure interaction with stabilized finite elements. Ph.D. thesis, University of Stuttgart, 1999.
[26] Sameh A, Sarin V. Hybrid parallel linear solvers. Int J Comput Fluid Dyn 1999;12:213–23.
[27] Sameh A, Sarin V. Parallel algorithms for indefinite linear systems. Parallel Comput 2002;28:285–99.
[28] Heil M. An efficient solver for the fully coupled solution of large-displacement fluid–structure interaction problems. Comput Methods Appl Mech Eng 2004;193:1–23.
[29] Hubner B, Walhorn E, Dinkler D. A monolithic approach to fluid–structure interaction using space–time finite elements. Comput Methods Appl Mech Eng 2004;193:2087–104.
[30] Dettmer W. Finite element modeling of fluid flow with moving free surfaces and interfaces including fluid–solid interaction. Ph.D. thesis, University of Wales, Swansea, 2004.
[31] Dettmer W, Peric D. A computational framework for fluid–rigid body interaction: finite element formulation and applications, Comput Methods Appl Mech Eng, in press.
[32] Green AE, Adkins JE. Large elastic deformations. London, UK: Oxford Clarendon Press, Amen House; 1960.
[33] Green AE, Zerna W. Theoretical elasticity. London, UK: Oxford Clarendon Press, Ely House; 1968.
[34] Tezduyar TE. Computation of moving boundaries and interfaces and stabilization parameters. Int J Numer Methods Fluids 2003;43:555–75.
[35] Tezduyar TE, Osawa Y. Finite element stabilization parameters computed from element matrices and vectors. Comput Methods Appl Mech Eng 2000;190:411–30.
[36] Tezduyar TE. Finite element methods for fluid dynamics with moving boundaries and interfaces. In: Stein E, De Borst R, Hughes TJR, editors. Encyclopedia of computational mechanics. Fluids, vol. 3. John Wiley & Sons; 2004 [Chapter 17].
[37] Hilber HM, Hughes TJR, Taylor RL. Improved numerical dissipation for time integration algorithms in structural dynamics. Earthquake Eng Struct Dyn 1977;5:283–92.
[38] Tezduyar TE. Stabilized finite element methods for computation of flows with moving boundaries and interfaces. Lecture notes on finite element simulation of flow problems (basic–advanced course). Tokyo, Japan: Japan Society of Computational Engineering and Sciences; 2003.
[39] Tezduyar TE, Sathe S, Senga M, Stein K. Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations, Comput Methods Appl Mech Eng, in press.
[40] Fujisawa T, Inaba M, Yagawa G. Parallel computing of high-speed compressible flows using a node-based finite element method. Int J Numer Methods Fluids 2003;58:481–511.