Proceedings of the 10th International Conference on Numerical Methods in Continuum Mechanics, Zilina, Slovakia, CD-ROM (2005).

Finite element modeling of fluid–structure interactions with space–time and advanced mesh update techniques

Tayfun E. Tezduyar*, Sunil Sathe, Masayoshi Senga, Luca Aureli
Mechanical Engineering, Rice University - MS 321
6100 Main Street, Houston, TX 77005, USA
e-mail: tezduyar@rice.edu

Keith Stein
Department of Physics, Bethel University
3900 Bethel Drive, St. Paul, MN 55112, USA

Barak Griffin
US Military Academy
West Point, NY 10996, USA

Abstract

We describe the space–time finite element formulations we developed for numerical modeling of fluid–structure interaction problems. The core method is the Deforming-Spatial-Domain/Stabilized Space–Time formulation, complemented with the mesh update methods, including the Solid-Extension Mesh Moving Technique and Move-Reconnect-Renode Mesh Update Method. Also complementing the core method are the block-iterative, quasi-direct and direct coupling methods for the solution of the fully-discretized, coupled fluid and structural mechanics equations. We present test computations for the mesh moving techniques and numerical examples with incompressible flows and membrane and cable structures.

1 Introduction

Numerical modeling of fluid–structure interactions (FSI) offers a variety of challenges in computational engineering. The spatial domain occupied by the fluid changes in time as the interface moves, and the mathematical model will need to handle that. Accurate representation of the flow field near the fluid–structure interface requires that the mesh be updated to track the interface, and this requires special attention in 3D problems with complex geometries. Solution of the coupled fluid and structural mechanics equations offers additional challenges, depending on the type of the problem solved. In this paper, we focus on the stabilized space–time finite element techniques, with emphasis on mesh update and light structures.

The Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation [1, 2, 3] was introduced in early 1990's as an interface-tracking technique for computation of flow problems with moving boundaries and interfaces. The DSD/SST formulation is the core method for our FSI algorithms; it is based on stabilized finite element formulations, which are written over the space–time domains of the fluid mechanics problems considered. The stabilized methods are the Streamline-Upwind/Petrov-Galerkin (SUPG) [4, 5, 6, 7] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [1, 8] formulations. An earlier version of the pressure stabilization, for Stokes flows, was reported in [9]. These stabilized formulations prevent numerical instabilities that might be encountered when we have high Reynolds or Mach numbers and shocks and strong boundary layers, as well as when using equal-order interpolation functions for velocity and pressure. The stabilized space–time formulations were used earlier by other researchers to solve problems with fixed spatial domains (see for example [10]). The space–time computations are carried out for one space–time "slab" at a time, where the "slab" is the slice of the space–time domain between the time levels n and $n+1$. This spares a 3D computational problem from

becoming a 4D problem including the time dimension. Some additional special features are exploited in the Special DSD/SST (S-DSD/SST) formulation [11, 12] to make the calculation of the element-level vectors and matrices more efficient.

Updating the mesh is based on moving it for as many time steps as we can and remeshing (generating fully or partially new set of nodes or elements) only as frequently as we need to. The mesh moving algorithm is essentially based on the one introduced in [13], where the motion of the nodes is governed by the equations of elasticity and the mesh deformation is dealt with selectively based on the sizes of the elements (see also [14]). We introduced a number of enhancements to this general mesh update technique. With the Solid-Extension Mesh Moving Technique (SEMMT) [15, 16], we address challenges involved in moving a mesh with very thin fluid elements near the solid surfaces. In the Move-Reconnect-Renode Mesh Update Method (MRRMUM) introduced in this paper, we define two remeshing options and propose to use each one when it is most effective to do so.

The fully-discretized equations of fluid and structural mechanics need to be solved in their coupled form, and we proposed a number of ways to accomplish that. They are: block-iterative coupling, which we have widely used in our computations (see [17, 18, 19]); quasi-direct coupling [11, 12]; and direct coupling [11, 12]. The direct coupling approach is based on the mixed analytical/numerical element-vector-based (AEVB/NEVB) computation technique introduced in [14, 20].

The core methods we developed for moving boundaries and interfaces have been applied to a wide range of FSI problems, some more than a decade ago (see, for example, [21, 22]), and some with very challenging complexities (see, for example, [17, 23]). The enhancements we describe in this paper, which include the advanced mesh update methods and the robust solution techniques for the coupled fluid and structure equations, are also applicable to a wide range of FSI problems. In this paper, however, we explain these enhancements in the context of FSI problems where the fluid mechanics is governed by the Navier–Stokes equations of incompressible flows and the structural mechanics, where the structures are light, is governed by the membrane and cable equations. These governing equations are reviewed in Section 2. The finite element formulations, including the DSD/SST formulation, are described in Section 3. In Section 4 we review the mesh update techniques, including the SEMMT and MRRMUM. In Section 5 we describe the block-iterative, quasi-direct, and direct coupling techniques. Numerical examples are presented in Section 6.

2 Governing equations

2.1 Fluid mechanics

Let $\Omega_t \subset I\!R^{n_{sd}}$ 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

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

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

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 $\boldsymbol{\varepsilon}(\mathbf{u}) = \left((\boldsymbol{\nabla}\mathbf{u}) + (\boldsymbol{\nabla}\mathbf{u})^T\right)/2$. 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 $\Omega_t^s \subset I\!R^{n_{xd}}$ be the spatial domain with boundary Γ_t^s, where $n_{xd} = 2$ for membranes and $n_{xd} = 1$ for cables. The parts of Γ_t^s corresponding to the essential and natural boundary conditions are represented by $(\Gamma_t^s)_g$ and $(\Gamma_t^s)_h$. The superscript "s" indicates the structure. The equations of motion are written as

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

where ρ^s, $\mathbf{y}$, $\mathbf{f}^s$ and $\boldsymbol{\sigma}^s$ are the material density, structural displacement, external force and the Cauchy stress tensor [24, 25], 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 [26]):

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

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 $S^{11} = E_c G^{11} G^{11} E_{11}$, where E_c is the Young's modulus for the cable. To account for stiffness-proportional material damping, the Hookean stress–strain relationships defined by Eq. (4) and its version for cables are modified, and E_{kl} is replaced by $\hat{E}_{kl}$, where $\hat{E}_{kl} = E_{kl} + \zeta \dot{E}_{kl}$. Here ζ is the stiffness-proportional damping coefficient and $\dot{E}_{kl}$ is the time derivative of E_{kl}.

3 Finite element formulations

3.1 DSD/SST formulation of fluid mechanics

In the DSD/SST method [1], the finite element formulation 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 $e = 1, 2, \ldots, (n_{el})_n$. 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 enforced over $(P_n)_{\mathrm{g}}$ and $(P_n)_{\mathrm{h}}$, the complementary subsets of the lateral boundary of the space–time slab. The finite element trial function spaces $(\mathcal{S}_{\mathbf{u}}^h)_n$ for velocity and $(\mathcal{S}_p^h)_n$ for pressure, and the test function spaces $(\mathcal{V}_{\mathbf{u}}^h)_n$ and $(\mathcal{V}_p^h)_n = (\mathcal{S}_p^h)_n$ are defined by using, over Q_n, first-order polynomials in space and time. The DSD/SST formulation is written as

follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in (\mathcal{S}_{\mathbf{u}}^h)_n$ and $p^h \in (\mathcal{S}_p^h)_n$ such that $\forall \mathbf{w}^h \in (\mathcal{V}_{\mathbf{u}}^h)_n$ and $\forall q^h \in (\mathcal{V}_p^h)_n$:

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

where

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

Here τ_{SUPG}, τ_{PSPG} and ν_{LSIC} are the SUPG, PSPG and LSIC (least-squares on incompressibility constraint) stabilization parameters. For ways of calculating τ_{SUPG}, τ_{PSPG} and ν_{LSIC}, see [27, 28, 20]. This formulation is applied to all space–time slabs $Q_0, Q_1, Q_2, \ldots, Q_{N-1}$, starting with $(\mathbf{u}^h)_0^- = \mathbf{u}_0$. For an earlier, detailed reference on the formulation see [1].

3.2 Semi-discrete formulation 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

$$\int_{\Omega_0^s} \mathbf{w}^h \cdot \rho^s \frac{d^2 \mathbf{y}^h}{dt^2} d\Omega^s + \int_{\Omega_0^s} \mathbf{w}^h \cdot \eta \rho^s \frac{d \mathbf{y}^h}{dt} d\Omega^s + \int_{\Omega_0^s} \delta \mathbf{E}^h : \mathbf{S}^h d\Omega^s = \int_{\Omega_t^s} \mathbf{w}^h \cdot \left(\mathbf{t}^h + \rho^s \mathbf{f}^s \right) d\Omega^s \ . \qquad (7)$$

The fluid mechanics forces acting on the structure are represented by vector $\mathbf{t}^h$. This force term is geometrically nonlinear and thus increases the overall nonlinearity of the formulation. The left-hand-side terms of Eq. (7) 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:

$$\left[\frac{\mathbf{M}}{\beta \Delta t^2} + \frac{(1-\alpha)\,\gamma \mathbf{C}}{\beta \Delta t} + (1-\alpha)\,\mathbf{K} \right] \Delta \mathbf{d}^i = \mathbf{R}^i \ , \qquad (8)$$

where $\mathbf{C} = \eta \mathbf{M} + \zeta \mathbf{K}$. Here $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ is the consistent tangent matrix associated with the internal elastic forces, $\mathbf{C}$ is a damping matrix, $\mathbf{R}^i$ is the residual vector at the i^{th} iteration, and $\Delta \mathbf{d}^i$ is the i^{th} 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. (8), all of the terms known from the previous iteration are lumped into the residual vector $\mathbf{R}^i$. The parameters α, β, γ are part of the Hilber–Hughes–Taylor [29] scheme, which is used here for time-integration.

4 Mesh update methods

How the mesh is updated depends on several factors, including the complexity of the fluid–structure interface and overall geometry, how unsteady the interface is, and how the starting

mesh was generated. In general, the mesh update has two components: moving the mesh for as long as it is possible, and full or partial remeshing (i.e., generating a new set of elements, and sometimes also a new set of nodes) when the element distortion becomes too high. In mesh moving, the only rule to follow is that at the interface the normal velocities of the mesh and the fluid have to match. With that constraint satisfied, the mesh can be moved in ways to reduce the frequency of remeshing. In most 3D applications, remeshing requires calling an automatic, unstructured-mesh generator. Reducing that cost becomes a major incentive for reducing the frequency of remeshing. Maintaining the parallel efficiency of the computations is another major incentive for reducing the frequency of remeshing, because parallel efficiency of most automatic mesh generators is substantially lower than that of most flow solvers. For example, reducing the frequency of remeshing to every ten time steps or less would sufficiently reduce the influence of remeshing in terms of its added cost and lack of parallel efficiency. In most of the complex flow problems we computed in the past, the frequency of remeshing was far less than every ten time steps. In our current parallel computations on a PC cluster, typically we perform remeshing on one of the nodes, which, with its 2 GigaBytes of memory, is powerful enough to generate large meshes. If remeshing does not consist of (full or partial) regeneration of just the element connectivities but also involves (full or partial) node regeneration, we need to project the solution from the old mesh to the new one. This involves a search process, which can be carried out in parallel. Still, the computational cost involved in this, and the projection errors introduced by remeshing, add more incentives for reducing the frequency of remeshing.

4.1 Automatic mesh moving technique

In the automatic mesh moving technique introduced in [13], the motion of the internal nodes is determined by solving the equations of elasticity. As the boundary condition, the motion of the nodes at the interfaces is specified to match the normal velocity of the fluid. Similar mesh moving techniques were used earlier by other researchers (see for example [30]). In [13] the mesh deformation is dealt with selectively based on the sizes of the elements. Mesh moving techniques with comparable features were later introduced in [31]. In the technique introduced in [13], selective treatment based on element sizes is attained by altering the way we account for the Jacobian of the transformation from the element domain to the physical domain. The objective is to stiffen the smaller elements, which are typically placed near solid surfaces, more than the larger ones. When this technique was first introduced in [13], it consisted of simply dropping the Jacobian from the finite element formulation of the mesh moving (elasticity) equations. This results in the smaller elements being stiffened more than the larger ones. The method described in [13] was augmented in [32] to a more extensive kind by introducing a stiffening power that determines the degree by which the smaller elements are rendered stiffer than the larger ones. This approach, when the stiffening power is set to 1.0, would be identical to the one first introduced in [13].

4.2 Solid-Extension Mesh Moving Technique (SEMMT)

In dealing with fluid–solid interfaces, sometimes we need to generate structured layers of elements around the solid objects to fully control the mesh resolution there and have more accurate representation of the boundary layers. In the mesh moving technique introduced in [13], such structured layers of elements move “glued” to the solid objects, undergoing a rigid-body motion. No equations are solved for the motion of the nodes in these layers, because these nodal motions are not governed by the equations of elasticity. This results in some cost reduction. But more importantly, the user has full control of the mesh resolution in these layers. For early examples of automatic mesh moving combined with structured layers of elements undergoing rigid-body motion with solid objects, see [33]. Earlier examples of element layers undergoing rigid-body

motion, in combination with deforming structured meshes, can be found in [1].

In computation of flows with fluid–solid interfaces where the solid is deforming, the motion of the fluid mesh near the interface cannot be represented by a rigid-body motion. Depending on the deformation mode of the solid, the automatic mesh moving technique described above may need to be used. In such cases, the thin fluid elements near the solid surface becomes a challenge for the automatic mesh moving technique. In the Solid-Extension Mesh Moving Technique (SEMMT) [15, 16], we proposed treating those thin fluid elements almost like an extension of the solid elements. In the SEMMT, in solving the equations of elasticity governing the motion of the fluid nodes, we assign higher rigidity to these thin elements compared to the other fluid elements. Two ways of accomplishing this were proposed in [15, 16]: solving the elasticity equations for the nodes connected to the thin elements separate from the elasticity equations for the other nodes, or together. If we solve them separately, for the thin elements, as boundary conditions at the interface with the other elements, we would use traction-free conditions. We refer to the separate solution option as "SEMMT – Multiple Domain (SEMMT–MD)" and the unified solution option as "SEMMT – Single Domain (SEMMT–SD)". In [34, 35], we presented test computations demonstrating how the SEMMT functions as part of our mesh update method. We employed both of the SEMMT options described above. The test computations included mesh deformation tests [34, 35] and a 2D FSI model problem [35]. In Subsection 4.4, we provide a brief description of the 2D FSI test computations.

4.3 General test conditions and mesh quality measures

The tests reported here are carried out with the standard technique (where stiffening power is set to 1.0 for all the elements, and all the nodes are moved together) and SEMMT–SD (with stiffening power set to 2.0 for the inner elements and 1.0 for the outer elements). The mesh over which the elasticity equations are solved is updated at each increment. This update is based on the displacements calculated over the current mesh that has been selectively stiffened. That way, the element Jacobians used in stiffening are updated every time the mesh deforms. As a result, the most current size of an element is used in determining how much it is stiffened. Also as a result, as an element approaches a tangled state, its Jacobian approaches zero, and its stiffening becomes very large. As a measure of mesh quality, we define, similar to the way done in [36], the Element Shape Change: $\mathsf{f}^e_{AR} = |\log(AR^e/AR^e_o)|$. Here subscript "$o$" refers to the undeformed mesh (i.e., the mesh obtained after the last remesh), and AR^e is the element aspect ratio, defined as $AR^e = (\ell^e{}_{max})^2/A^e$, where $\ell^e{}_{max}$ is the maximum edge length for element e.

4.4 2D FSI model problem with SEMMT

The model represents a parachute-like structure. The "canopy" is modeled with 50 membrane elements and the "suspension lines" with 22 cable elements. Two layers of elements extend outward from the upper and lower surfaces of the canopy. Detailed information on the flow conditions, structural parameters, and the solution steps can be found in [35]. In this test computation the $\frac{\partial \mathbf{w}^h}{\partial t}$ term in Eq. (5) has been dropped. Figure 1 shows the deformed meshes for the standard mesh moving technique and the SEMMT–SD. The orthogonality of the mesh lines at the canopy surface is much better preserved with the SEMMT–SD. For more on this test computation, see [35].

4.5 Move-Reconnect-Renode Mesh Update Method (MRRMUM)

We propose the MRRMUM as our most effective mesh update technique. In the MRRMUM, we define two remeshing options and propose to use each one when it is most effective to do so. In the "reconnect" option, we only change the way the nodes are connected and thus only

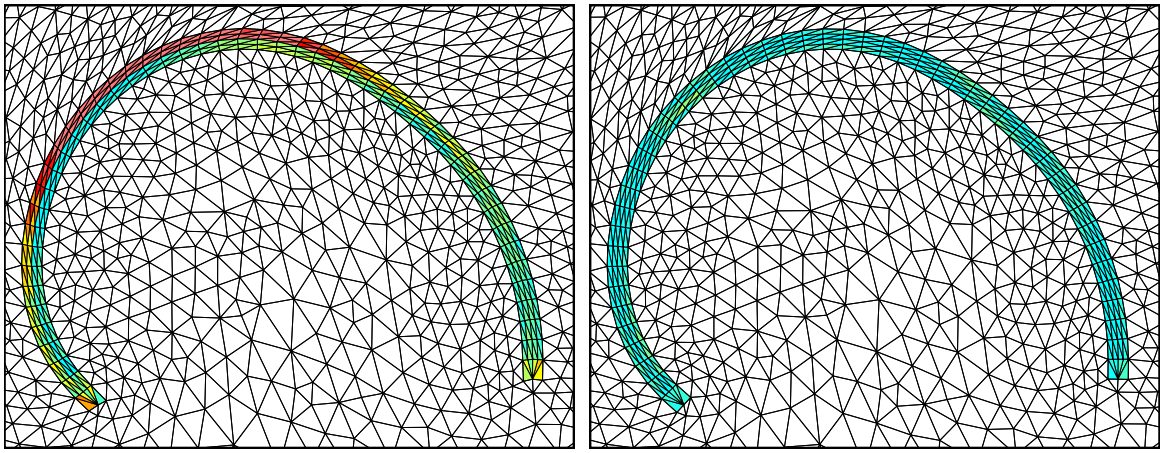

Figure 1: Mesh deformations of the FSI model problem for the standard mesh moving technique (left) and SEMMT–SD (right). Colors of the inner elements indicate element distortions as measured by the aspect ratio change. The color range from light blue to light red corresponds to the range $0 \leq \mathsf{f}^e_{AR} \leq 0.25$.

replace the existing elements (fully or partially) with a new set of elements. The mesh generator developed in [37] gives us the reconnect option. In the "renode" option, we replace the existing nodes (fully or partially) with a new set of nodes. This, of course, results in also replacing the existing elements with a new set of elements. Because the reconnect option is simpler and involves less projection errors, it is preferable to the renode option. In the MRRMUM, we move the mesh for as many time steps as we can, reconnect only as frequently as we need to, and renode only when doing so is the only remaining option.

5 Solution of fully-discretized equations

Full discretizations of the formulations described in Subsections 3.1 and 3.2 lead to coupled, nonlinear equation systems that need to be solved at every time step. In a form that is partitioned with respect to the models represented, these nonlinear equations can be written as follows:

$$\mathbf{N}_1\left(\mathbf{d}_1, \mathbf{d}_2\right) = \mathbf{F}_1 \,, \tag{9}$$

$$\mathbf{N}_2\left(\mathbf{d}_1, \mathbf{d}_2\right) = \mathbf{F}_2 \,, \tag{10}$$

where $\mathbf{d}_1$ and $\mathbf{d}_2$ are the vectors of nodal unknowns corresponding to unknown functions $\mathbf{u}_1$ and $\mathbf{u}_2$, respectively. In the context of a coupled FSI problem, $\mathbf{u}_1$ and $\mathbf{u}_2$ represent the fluid and structure unknowns, respectively. For the space–time formulation of the fluid mechanics problem, $\mathbf{d}_1$ represents unknowns associated with the finite element formulation written for the space–time slab between the time levels n to $n+1$ (see [1, 2, 3]). Solution of these equations with the Newton–Raphson method would necessitate at every Newton–Raphson step solution of the following linear equation system:

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

$$\mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 = \mathbf{b}_2 \,, \tag{12}$$

where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the residuals of the nonlinear equation system, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the correction increments for $\mathbf{d}_1$ and $\mathbf{d}_2$, and $\mathbf{A}_{\beta\gamma} = \partial\mathbf{N}_\beta/\partial\mathbf{d}_\gamma$. Keeping the coupling matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ in the picture would require taking into account the dependence of Eq. (9) on the mesh motion. In Subsections 5.1–5.3 we describe different ways of handling the coupling between Eqs. (9) and (10).

5.1 Block-iterative coupling

In the block-iterative coupling, we do not keep the coupling matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ in the picture. In an iteration step taking us from iterative solution i to $i+1$, we solve the following set of equations:

$$\left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_1}\right|_{(\mathbf{d}_1^i,\ \mathbf{d}_2^i)} \left(\Delta \mathbf{d}_1^i\right) = \mathbf{F}_1 - \mathbf{N}_1\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i\right) \ , \tag{13}$$

$$\left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_2}\right|_{(\mathbf{d}_1^{i+1},\ \mathbf{d}_2^i)} \left(\Delta \mathbf{d}_2^i\right) = \mathbf{F}_2 - \mathbf{N}_2\left(\mathbf{d}_1^{i+1},\ \mathbf{d}_2^i\right) \ . \tag{14}$$

The linear equations systems given by Eqs. (13)–(14) are also solved iteratively, using the GMRES search technique [38].

Because the matrix blocks representing the coupling between the fluid and structural mechanics systems are not in the picture, in computations where the structure is light, structural response becomes very sensitive to small changes in the fluid mechanics forces and convergence becomes difficult to achieve. In Subsections 5.2 and 5.3 we describe ways of keeping the coupling matrix blocks in the picture. In the absence of keeping the coupling matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$, we proposed in [20, 39] a short cut approach for improving the convergence of the block iterations. In this approach, to reduce "over-correcting" (i.e. "over-incrementing") the structural displacements during the block iterations, 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 block iterations converge, they converge to the solution of the problem with the correct structural mass.

5.2 Quasi-direct coupling

In the quasi-direct coupling approach, we keep the the coupling matrices $\mathbf{A}_{12}$ and $\mathbf{A}_{21}$ in the picture partially, without taking into account the dependence of $\mathbf{A}_{12}$ on the mesh motion. In describing this approach, we re-write the finite element formulations given by Eqs. (5) and (7), with a slight change of notation, and with a clarification of how the fluid–structure interface conditions are handled:

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}_{1\mathrm{E}}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}_{1\mathrm{E}}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ - \int_{(P_n)_{\mathrm{h}}} \mathbf{w}_{1\mathrm{E}}^h \cdot \mathbf{h}_{1\mathrm{E}}^h dP \\
+ &\int_{Q_n} q_{1\mathrm{E}}^h \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ + \int_{\Omega_n} (\mathbf{w}_{1\mathrm{E}}^h)_n^+ \cdot \rho \left((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^- \right) d\Omega \\
+ &\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_{\mathrm{SUPG}} \rho \left(\frac{\partial \mathbf{w}_{1\mathrm{E}}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{w}_{1\mathrm{E}}^h \right) + \tau_{\mathrm{PSPG}} \boldsymbol{\nabla} q_{1\mathrm{E}}^h \right] \cdot \left[\mathrm{L}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] dQ \\
+ &\sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot \mathbf{w}_{1\mathrm{E}}^h \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ = 0 \ ,
\end{aligned} \tag{15}$$

$$\int_{Q_n} q_{1\mathrm{I}}^h \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ + \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \frac{1}{\rho} \left[\tau_{\mathrm{PSPG}} \boldsymbol{\nabla} q_{1\mathrm{I}}^h \right] \cdot \left[\mathrm{L}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] dQ = 0 \ , \tag{16}$$

$$\int_{\Gamma_{1\mathrm{I}}} (\mathbf{w}_{1\mathrm{I}}^h)_{n+1}^- \cdot \left((\mathbf{u}_{1\mathrm{I}}^h)_{n+1}^- - \mathbf{u}_{2\mathrm{I}}^h \right) d\Gamma = 0 \ , \tag{17}$$

$$\int_{Q_n} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}((\mathbf{w}^h_{1\mathrm{I}})^-_{n+1}) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ$$
$$- \int_{(P_n)_\mathrm{h}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \mathbf{h}^h_{1\mathrm{I}} dP$$
$$+ \sum_{e=1}^{(n_{el})_n} \int_{Q^e_n} \frac{1}{\rho} \left[\tau_{\mathrm{SUPG}} \rho \left(\frac{\partial (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1}}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \right) \right] \cdot \left[\text{Ł}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h \right] dQ$$
$$+ \sum_{e=1}^{(n_{el})_n} \int_{Q^e_n} \nu_{\mathrm{LSIC}} \boldsymbol{\nabla} \cdot (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \rho \boldsymbol{\nabla} \cdot \mathbf{u}^h dQ = 0 \ , \tag{18}$$

$$\int_{(\Omega_2)_0} \mathbf{w}^h_2 \cdot \rho_2 \frac{d^2 \mathbf{y}^h}{dt^2} d\Omega + \int_{(\Omega_2)_0} \mathbf{w}^h_2 \cdot \eta \rho_2 \frac{d \mathbf{y}^h}{dt} d\Omega + \int_{(\Omega_2)_0} \delta \mathbf{E}^h : \mathbf{S}^h d\Omega$$
$$= \int_{\Omega_2} \mathbf{w}^h_2 \cdot \rho_2 \mathbf{f}^h_2 d\Omega + \int_{\Omega_{2\mathrm{E}}} \mathbf{w}^h_{2\mathrm{E}} \cdot \mathbf{h}^h_{2\mathrm{E}} d\Omega - \int_{\Omega_{2\mathrm{I}}} \mathbf{w}^h_{2\mathrm{I}} \cdot \mathbf{h}^h_{1\mathrm{I}} d\Omega \ . \tag{19}$$

While the subscript I refers to the fluid–structure interface, the subscript E refers to "elsewhere" in the fluid and structure domains or boundaries. In reconciling the slightly modified notation used here with the notation we used in Eqs. (5) and (7), we note that $\rho_2 = \rho^s$, $\mathbf{f}^h_2 = \mathbf{f}^s$, $(\Omega_2)_0 = \Omega^s_0$, $\Omega_2 = \Omega^s_t$, and $\Omega_{2\mathrm{I}}$ and $\Omega_{2\mathrm{E}}$ indicate the partitions of Ω_2 corresponding to the interface and "elsewhere". We also note that $\mathbf{h}^h_{1\mathrm{I}} = -\mathbf{t}^h$, and $\mathbf{h}^h_{2\mathrm{E}}$ denotes the prescribed external forces acting on the structure in $\Omega_{2\mathrm{E}}$, which is separate from $\mathbf{f}^h_2$. In this formulation, $(\mathbf{u}^h_{1\mathrm{I}})^-_{n+1}$ and $\mathbf{h}^h_{1\mathrm{I}}$ (the fluid velocity and stress at the interface) are treated as separate unknowns, and Eqs. (17) and (18) can be seen as equations corresponding to these two unknowns, respectively. The structural displacement rate at the interface, $\mathbf{u}^h_{2\mathrm{I}}$, is derived from $\mathbf{y}^h$.

The formulation above is based on allowing for cases when 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. (17) is no longer needed and $\mathbf{h}^h_{1\mathrm{I}}$ is no longer treated as a separate unknown. If the structure is represented by a 3D continuum model instead of a membrane model, the formulation above would still be applicable if the the domain integrations over $\Omega_{2\mathrm{E}}$ and $\Omega_{2\mathrm{I}}$ in the last two terms of Eq. (19) are converted to boundary integrations over $\Gamma_{2\mathrm{E}}$ and $\Gamma_{2\mathrm{I}}$. In such cases, $\mathbf{h}^h_{2\mathrm{E}}$ would represent the prescribed forces acting "elsewhere" on the surface of the structure.

In a slightly altered version of the formulation given by Eqs. (15) – (19), we suppress the SUPG and LSIC stabilizations in Eq. (18), and use the following equation:

$$\int_{Q_n} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \boldsymbol{\nabla} \mathbf{u}^h - \mathbf{f}^h \right) dQ + \int_{Q_n} \boldsymbol{\varepsilon}((\mathbf{w}^h_{1\mathrm{I}})^-_{n+1}) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) dQ$$
$$- \int_{(P_n)_\mathrm{h}} (\mathbf{w}^h_{1\mathrm{I}})^-_{n+1} \cdot \mathbf{h}^h_{1\mathrm{I}} dP = 0 \ . \tag{20}$$

Our more recent experiences indicate that Eq. (20) leads to a more robust solution algorithm. Although this may not be so surprising when Eq. (20) is considered in conjunction with Eq. (17), it is something that we plan to investigate further.

5.3 Direct coupling

The mixed analytical/numerical element-vector-based (AEVB/NEVB) computation technique introduced in [14, 20] can be employed to keep the coupling matrices in the picture fully by taking

into account their dependence on the mesh motion. In describing the mixed AEVB/NEVB technique, we first write the iterative solution of the equation system given by Eq. (11)–(12) as follows:

$$\begin{aligned} \mathbf{P}_{11}\Delta\mathbf{y}_1 + \mathbf{P}_{12}\Delta\mathbf{y}_2 &= \mathbf{b}_1 - (\mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2) \ , \quad (21) \\ \mathbf{P}_{21}\Delta\mathbf{y}_1 + \mathbf{P}_{22}\Delta\mathbf{y}_2 &= \mathbf{b}_2 - (\mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2) \ , \quad (22) \end{aligned}$$

where $\Delta\mathbf{y}_1$ and $\Delta\mathbf{y}_2$ represent the candidate corrections to $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\mathbf{P}_{\beta\gamma}$'s represent the blocks of the preconditioning matrix $\mathbf{P}$. Here we focus our attention on computation of the residual vectors on the right-hand side, and explore ways for evaluating the matrix–vector products.

Let us suppose that we are able to compute, without major difficulty, the element-level matrices $\mathbf{A}^e_{11}$ and $\mathbf{A}^e_{22}$ associated with the global matrices $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$, and that we prefer to evaluate $\mathbf{A}_{11}\mathbf{x}_1$ and $\mathbf{A}_{22}\mathbf{x}_2$ by using these element-level matrices. Let us also suppose that calculation of $\mathbf{A}^e_{12}$ and $\mathbf{A}^e_{21}$ is exceedingly difficult. Then the computations can be carried out by using a mixed element-matrix-based (EMB)/element-vector-based (EVB) technique [14, 20]:

$$\begin{aligned} (\mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2) &= \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\mathbf{A}^e_{11}\mathbf{x}_1) + \mathop{\mathsf{A}}_{e=1}^{n_{el}} \lim_{\epsilon_1\to 0} \left[\frac{\mathbf{N}^e_1(\mathbf{d}_1, \mathbf{d}_2 + \epsilon_1\mathbf{x}_2) - \mathbf{N}^e_1(\mathbf{d}_1, \mathbf{d}_2)}{\epsilon_1}\right], \quad (23) \\ (\mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2) &= \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\mathbf{A}^e_{22}\mathbf{x}_2) + \mathop{\mathsf{A}}_{e=1}^{n_{el}} \lim_{\epsilon_2\to 0} \left[\frac{\mathbf{N}^e_2(\mathbf{d}_1 + \epsilon_2\mathbf{x}_1, \mathbf{d}_2) - \mathbf{N}^e_2(\mathbf{d}_1, \mathbf{d}_2)}{\epsilon_2}\right], \quad (24) \end{aligned}$$

where ϵ_1 and ϵ_2 are small parameters used in numerical evaluation of the directional derivatives. Here, $\mathbf{A}_{11}\mathbf{x}_1$ and $\mathbf{A}_{22}\mathbf{x}_2$ are evaluated with an EMB technique and $\mathbf{A}_{12}\mathbf{x}_2$ and $\mathbf{A}_{21}\mathbf{x}_1$ with an EVB technique.

In extending the mixed EMB/EVB technique to a more general framework, evaluation of a matrix–vector product $\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma$ (for $\beta, \gamma = 1, 2, \ldots, N$ and no sum) appearing in a residual vector can be formulated as choice between the following EMB and EVB techniques:

$$\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma = \mathop{\mathsf{A}}_{e=1}^{n_{el}} (\mathbf{A}^e_{\beta\gamma}\mathbf{x}_\gamma), \quad (25)$$

$$\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \lim_{\epsilon_\beta\to 0} \left[\frac{\mathbf{N}^e_\beta(\ldots, \mathbf{d}_\gamma + \epsilon_\beta\mathbf{x}_\gamma, \ldots) - \mathbf{N}^e_\beta(\ldots, \mathbf{d}_\gamma, \ldots)}{\epsilon_\beta}\right]. \quad (26)$$

Sometimes computation of $\mathbf{A}^e_{\beta\gamma}$ might not be exceedingly difficult, but we might still prefer to evaluate $\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma$ with an EVB technique. In such cases, instead of an EVB technique requiring numerical evaluation of directional derivatives, we might want to use the EVB technique described below.

Let us suppose that the nonlinear vector function $\mathbf{N}_\beta$ corresponds to a finite element integral form $\mathbf{B}_\beta(\mathbf{W}_\beta, \mathbf{u}_1, \ldots, \mathbf{u}_N)$. Here $\mathbf{W}_\beta$ represents the vector of nodal values associated with the weighting function $\mathbf{w}_\beta$, which generates the nonlinear equation block β. Let us also suppose that we are able to, without major difficulty, derive the first-order terms in the expansion of $\mathbf{B}_\beta(\mathbf{W}_\beta, \mathbf{u}_1, \ldots, \mathbf{u}_N)$ in $\mathbf{u}_\gamma$. Let the finite element integral form $\mathbf{G}_{\beta\gamma}(\mathbf{W}_\beta, \mathbf{u}_1, \ldots, \mathbf{u}_N, \Delta\mathbf{u}_\gamma)$ represent those first-order terms in $\Delta\mathbf{u}_\gamma$. We note that this integral form will generate $\frac{\partial\mathbf{N}_\beta}{\partial\mathbf{d}_\gamma}$. Consequently, the product $\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma$ can be evaluated as [14, 20]

$$\mathbf{A}_{\beta\gamma}\mathbf{x}_\gamma = \frac{\partial\mathbf{N}_\beta}{\partial\mathbf{d}_\gamma}\mathbf{x}_\gamma = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \mathbf{G}_{\beta\gamma}(\mathbf{W}_\beta, \mathbf{u}_1, \ldots, \mathbf{u}_N, \mathbf{v}_\gamma) \ , \quad (27)$$

where, $\mathbf{v}_\gamma$ is a function interpolated from $\mathbf{x}_\gamma$ in the same way $\mathbf{u}_\gamma$ is interpolated from $\mathbf{d}_\gamma$. This EVB technique allows us to evaluate matrix–vector products without dealing with numerical

evaluation of directional derivatives. To differentiate between the EVB techniques defined by Eqs. (26) and (27), we call them, respectively, numerical EVB (NEVB) and analytical EVB (AEVB) techniques.

We propose two ways of using the mixed AEVB/NEVB computation technique to take into account the dependence of the coupling matrices on the mesh motion. In the first way, we propose to use the NEVB technique to compute $\mathbf{A}_{12}\mathbf{x}_2$ while including the dependence of $\mathbf{A}_{12}$ on the mesh motion. This would be done as seen in Eq. (23). In the second way we propose, we limit the use of the NEVB technique to evaluation of the matrix–vector products involving coupling matrices representing the dependence on the mesh motion. This would be done by considering a three-block version of the nonlinear equation system given by Eqs. (9)–(10), where $\mathbf{d}_3$ is the vector of nodal unknowns representing the mesh motion, and the third block of equations represents the mesh-moving equations. The three-block version of Eq. (11)–(12) can be solved iteratively with the three-block of version of Eqs. (21)–(22), which is written as follows:

$$\mathbf{P}_{11}\Delta\mathbf{y}_1 + \mathbf{P}_{12}\Delta\mathbf{y}_2 + \mathbf{P}_{13}\Delta\mathbf{y}_3 = \mathbf{b}_1 - (\mathbf{A}_{11}\mathbf{x}_1 + \mathbf{A}_{12}\mathbf{x}_2 + \mathbf{A}_{13}\mathbf{x}_3)\,, \tag{28}$$

$$\mathbf{P}_{21}\Delta\mathbf{y}_1 + \mathbf{P}_{22}\Delta\mathbf{y}_2 + \mathbf{P}_{23}\Delta\mathbf{y}_3 = \mathbf{b}_2 - (\mathbf{A}_{21}\mathbf{x}_1 + \mathbf{A}_{22}\mathbf{x}_2 + \mathbf{A}_{23}\mathbf{x}_3)\,, \tag{29}$$

$$\mathbf{P}_{31}\Delta\mathbf{y}_1 + \mathbf{P}_{32}\Delta\mathbf{y}_2 + \mathbf{P}_{33}\Delta\mathbf{y}_3 = \mathbf{b}_3 - (\mathbf{A}_{31}\mathbf{x}_1 + \mathbf{A}_{32}\mathbf{x}_2 + \mathbf{A}_{33}\mathbf{x}_3)\,. \tag{30}$$

The NEVB technique can be used for computing $\mathbf{A}_{13}\mathbf{x}_3$ as follows:

$$\mathbf{A}_{13}\mathbf{x}_3 = \mathop{\mathsf{A}}_{e=1}^{n_{el}} \lim_{\epsilon_1 \to 0} \left[\frac{\mathbf{N}_1^e(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3 + \epsilon_1\mathbf{x}_3) - \mathbf{N}_1^e(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3)}{\epsilon_1}\right]. \tag{31}$$

6 Numerical examples

6.1 Flow past a "flag"

In this test problem, we compute the FSI involved in the flapping of a "flag". Results from a very similar problem were presented in [40]. We use the quasi-direct coupling approach, as described by Eqs. (15)–(17) and Eqs. (19)–(20). The dimensions of the flag are 1.5 m in the flow direction and 1.0 m in the span-wise direction. The fluid velocity, density and kinematic viscosity are 2 m/s, 1 kg/m^3 and 0.008 m^2/s, respectively. The flag is modeled as a membrane with density 1000 kg/m^3, thickness 0.2 mm, and Young's modulus 40,000 N/m^2. The leading edge of the flag is held fixed and the lateral edges of the flag are constrained to move only in a normal plane. The FSI computations are carried out until a nearly cyclic pattern of flapping is reached. Figure 2 shows a sequence of snapshots of the flag and the horizontal velocity on a normal plane. Figure 3 shows the displacement and velocity for the midpoint of the free edge of the flag. We note that the results are very similar to those reported in [11, 12], using the quasi-direct coupling technique described by Eqs. (15)–(19).

6.2 Soft landing of a G–12 parachute

In this test problem, we compute the soft-landing of a G–12 parachute using a direct coupling technique based on Eqs. (15)–(19) and Eq. (31). This is a 64-ft diameter parachute with 64 suspension lines, each about 51 ft long, and 4 risers, where each riser is connected to 16 suspension lines. The risers meet at a single confluence point, which is connected to the payload with 4 pneumatic muscle actuators (PMAs), each about 15 ft long. The payload is 2,200 lb. The soft landing is accomplished by the retraction of these PMAs prior to landing. In this test case the PMAs are retracted 7.1 ft in 0.23 s while the parachute is descending at 28 ft/s. In the computation the $\frac{\partial \mathbf{w}^h}{\partial t}$ term in Eqs. (15) and (18) has been dropped. Figure 4 shows the dynamics of the parachute during and after the retraction. The payload descent speed becomes as low as 10 ft/s, and we see a large increase in drag. Figure 5 shows the flow field during and after the retraction. For more details on this computation, see [41].

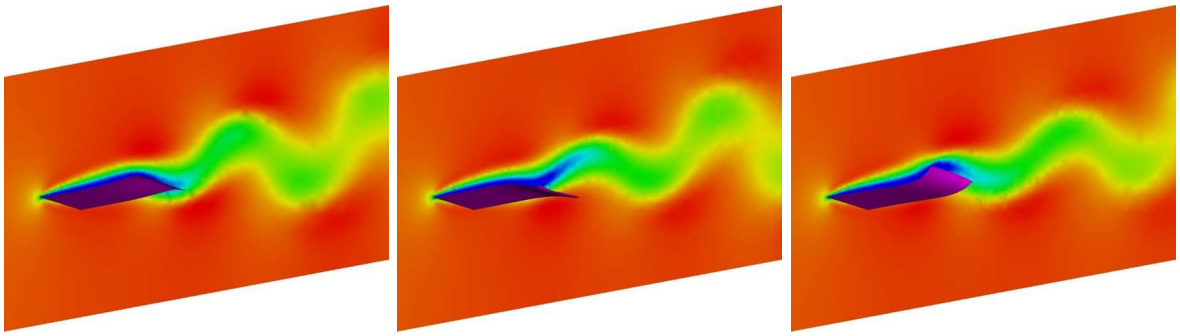

Figure 2: Time history (left to right) of the flag motion and the horizontal velocity on a normal plane.

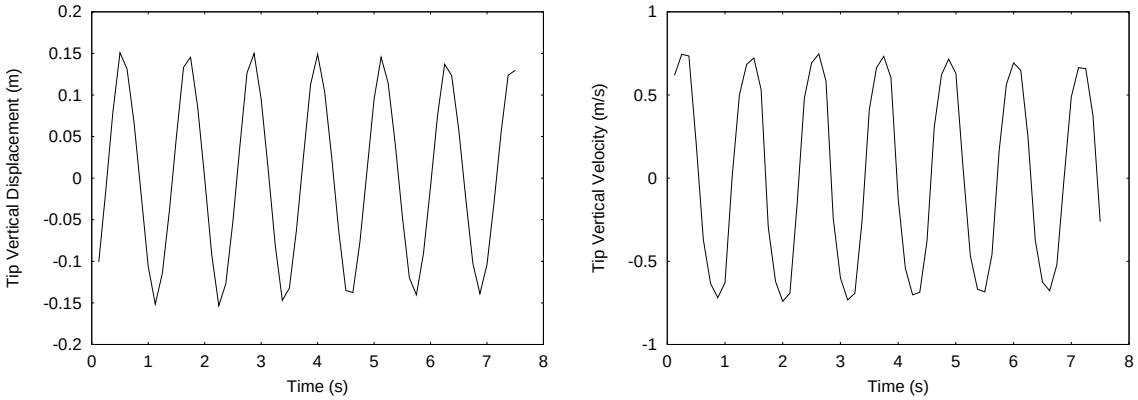

Figure 3: Vertical displacement (left) and velocity (right) for the midpoint of the free edge of the flag.

6.3 Disreefing of a G–12 parachute

A parachute can be initially deployed in a reefed configuration and with a lower drag area so that higher descent speeds are achieved during most of its descent. Prior to landing, it can be disreefed to increase the drag area to its full level and thus reduce the landing speed to its normal value. The challenge involved in mesh update increases significantly in this type of FSI problems, and we use this test problem to show how we address such challenges. The additional geometric complexities causing the increase in mesh-update difficulties are limited to certain periods of the FSI simulation. Therefore we propose to overcome those difficulties by selectively decreasing the time-step size or increasing the remeshing frequency during those periods. We start with a reefed parachute with a diameter 25% smaller than the diameter of a fully-inflated parachute. Figure 6 shows the fully-inflated and reefeed parachutes. Figure 7 shows the surface mesh for the reefed parachute. The FSI computation is carried out with the quasi-direct coupling approach, as described by Eqs. (15)–(17) and Eqs. (19)–(20). Figure 8 shows the parachute shape at various instants during the disreefing.

6.4 Remeshing performance for rigid-body rotation of a T–10 parachute

The objective here is to compare, for the prescribed rigid-body rotation of a parachute, the performances of the two remeshing options described in Subsection 4.5. We use a T–10 parachute, which has a diameter of 35 ft. Our reference frame is moving at 16 ft/s (in $-z$ direction), which is the standard descent speed for this parachute. We prescribe a rigid-body rotation of 0.0914 rad/s around the y axis. We then evaluate how remeshing with the "reconnect" and "renode"

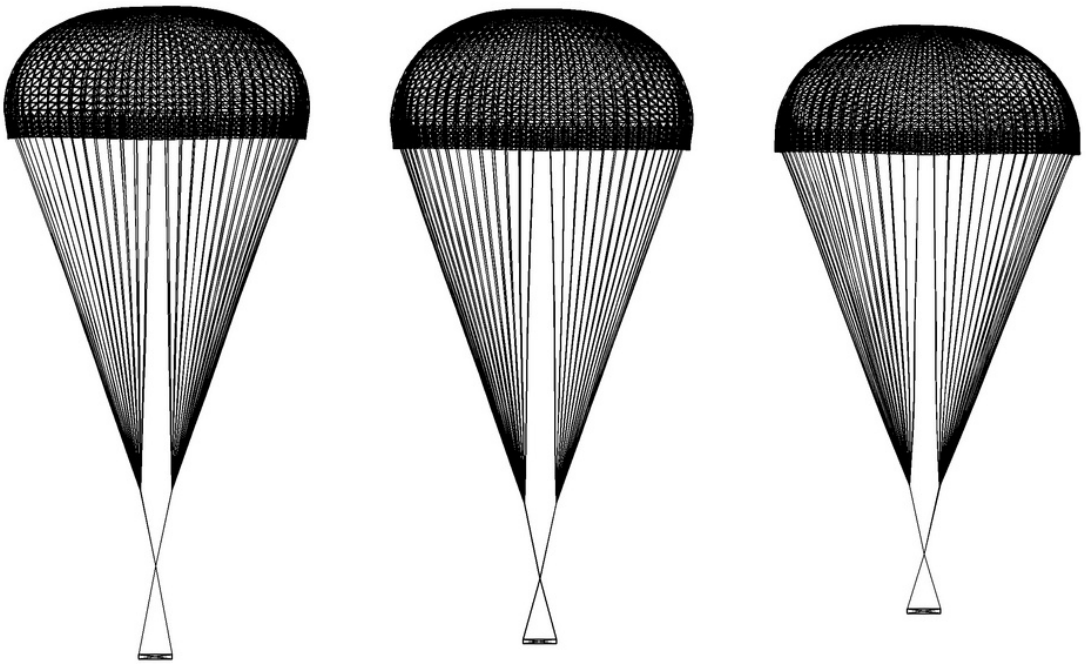

Figure 4: Soft landing of a G–12 parachute. Dynamics of the parachute, represented by images at three instants during and after the retraction. For more on this computation, see [41].

options perform. Figures 9, 10 and 11 show the aerodynamical forces acting on the parachute in the z, x and y directions, respectively, obtained with the "reconnect" and "renode" options. It can clearly be seen that the force oscillations seen immediately after the remeshing are reduced substantially with the "reconnect" option.

7 Concluding remarks

We provided an overview of the space–time finite element formulations we developed to address the computational challenges involved in numerical modeling of fluid–structure interactions. The core method is the Deforming-Spatial-Domain/Stabilized Space–Time (DSD/SST) formulation, which has been proven to be a powerful and comprehensive method for solving flow problems with moving boundaries and interfaces. The core method is enhanced by a number of additional methods we developed to address more specific challenges. The mesh update methods developed, including the Solid-Extension Mesh Moving Technique (SEMMT) and Move-Reconnect-Renode Mesh Update Method (MRRMUM), help us update the mesh effectively as the spatial domain occupied by the fluid changes in time. What technique to use for the solution of the fully-discretized, coupled fluid and structural mechanics equations should, to a certain extent, depend on the nature of the application problem. Keeping that in mind, we have developed block-iterative, quasi-direct, and direct coupling techniques. The block-iterative technique gives us more flexibility in terms of algorithmic modularity and independence of the fluid and structural mechanics solvers. The quasi-direct and direct coupling techniques give us more robust algorithms for FSI computations where the structure is light. Our overview included test computations for the mesh moving techniques we developed, as well as test computations for FSI applications with light structures. We believe that in this article we demonstrated that the core method and its enhancements are significantly increasing the scope and accuracy of FSI computations.

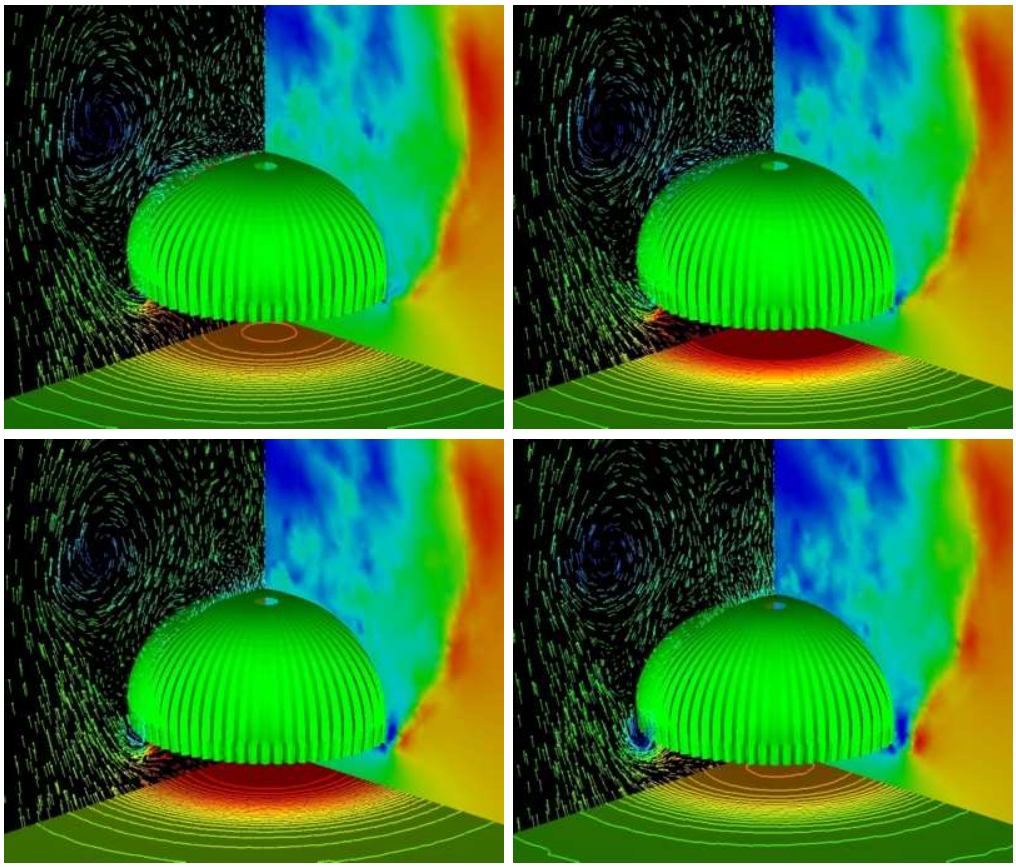

Figure 5: Soft landing of a G–12 parachute. Flow field at four instants during and after the retraction (left to right and top to bottom). Left plane: velocity vectors; right plane: magnitude of vertical velocity; bottom plane: pressure. For more on this computation, see [41].

Acknowledgments

This work was supported by the Natick Soldier Center, NSF and NASA.

References

[1] T.E. Tezduyar, "Stabilized finite element formulations for incompressible flow computations", *Advances in Applied Mechanics*, **28** (1992) 1–44.

[2] T.E. Tezduyar, M. Behr, and 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 numerical tests", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 339–351.

[3] T.E. Tezduyar, M. Behr, S. Mittal, and 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", *Computer Methods in Applied Mechanics and Engineering*, **94** (1992) 353–371.

[4] T.J.R. Hughes and A.N. Brooks, "A multi-dimensional upwind scheme with no crosswind diffusion", in T.J.R. Hughes, editor, *Finite Element Methods for Convection Dominated Flows*, AMD-Vol.34, 19–35, ASME, New York, 1979.

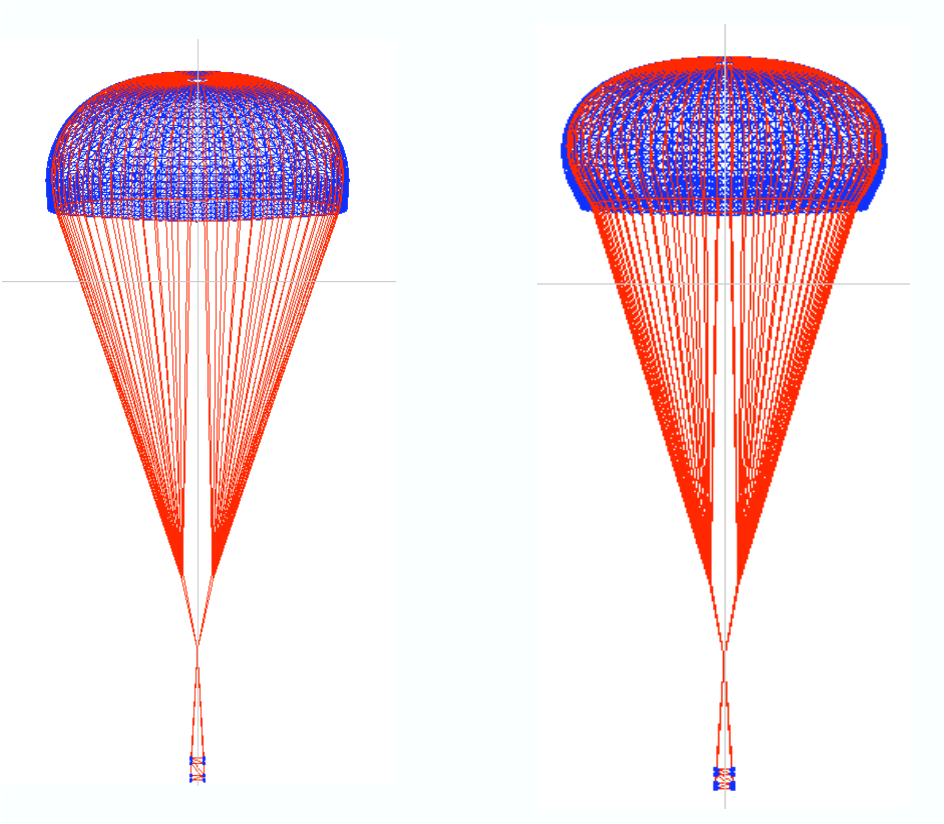

Figure 6: Disreefing of a G–12 parachute. Fully inflated (left) and reefed (right) parachutes. The reduction in diameter is 25%.

[5] A.N. Brooks and T.J.R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations", *Computer Methods in Applied Mechanics and Engineering*, **32** (1982) 199–259.

[6] T.E. Tezduyar and T.J.R. Hughes, "Finite element formulations for convection dominated flows with particular emphasis on the compressible Euler equations", in *Proceedings of AIAA 21st Aerospace Sciences Meeting*, AIAA Paper 83-0125, Reno, Nevada, (1983).

[7] T.J.R Hughes and T.E. Tezduyar, "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations", *Computer Methods in Applied Mechanics and Engineering*, **45** (1984) 217–284.

[8] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih, "Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements", *Computer Methods in Applied Mechanics and Engineering*, **95** (1992) 221–242.

[9] T.J.R. Hughes, L.P. Franca, and M. Balestra, "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", *Computer Methods in Applied Mechanics and Engineering*, **59** (1986) 85–99.

[10] T.J.R. Hughes and G.M. Hulbert, "Space–time finite element methods for elastodynamics: formulations and error estimates", *Computer Methods in Applied Mechanics and Engineering*, **66** (1988) 339–363.

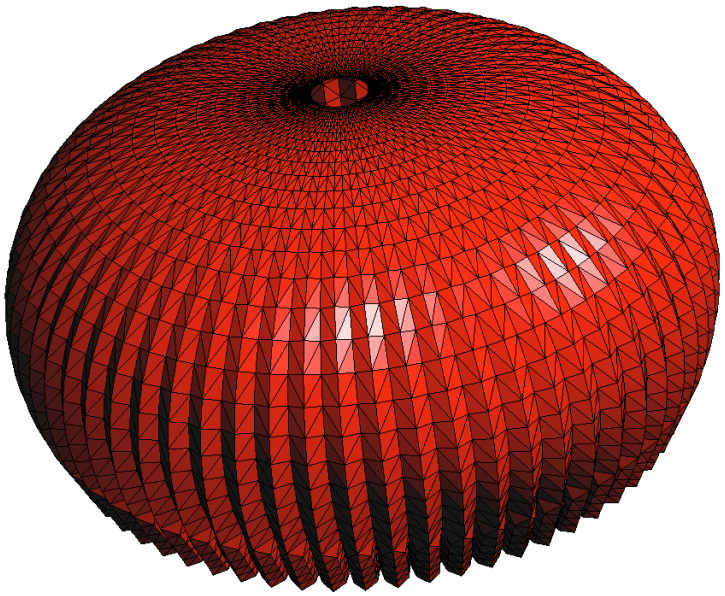

Figure 7: Disreefing of a G–12 parachute. Surface mesh for the reefed parachute.

[11] T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space–time techniques for finite element computation of flows with moving boundaries and interfaces", in S. Gallegos, I. Herrera, S. Botello, F. Zarate, and G. Ayala, editors, *Proceedings of the III International Congress on Numerical Methods in Engineering and Applied Science*, CD-ROM, 2004.

[12] T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein, "Space–time finite element techniques for computation of fluid–structure interactions", *Computer Methods in Applied Mechanics and Engineering*, **195** (2006) 2002–2027.

[13] T.E. Tezduyar, M. Behr, S. Mittal, and A.A. Johnson, "Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations", in *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, ASME, New York, (1992) 7–24.

[14] T.E. Tezduyar, "Finite element methods for flow problems with moving boundaries and interfaces", *Archives of Computational Methods in Engineering*, **8** (2001) 83–130.

[15] T. Tezduyar, "Finite element interface-tracking and interface-capturing techniques for flows with moving boundaries and interfaces", in *Proceedings of the ASME Symposium on Fluid-Physics and Heat Transfer for Macro- and Micro-Scale Gas-Liquid and Phase-Change Flows (CD-ROM)*, ASME Paper IMECE2001/HTD-24206, ASME, New York, New York, (2001).

[16] T.E. Tezduyar, "Stabilized finite element formulations and interface-tracking and interface-capturing techniques for incompressible flows", in M.M. Hafez, editor, *Numerical Simulations of Incompressible Flows*, World Scientific, New Jersey, (2003) 221–239.

[17] K. Stein, R. Benney, V. Kalro, T.E. Tezduyar, J. Leonard, and M. Accorsi, "Parachute fluid–structure interactions: 3-D Computation", *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 373–386.

[18] K. Stein, R. Benney, T. Tezduyar, and J. Potvin, "Fluid–structure interactions of a cross parachute: Numerical simulation", *Computer Methods in Applied Mechanics and Engineering*, **191** (2001) 673–687.

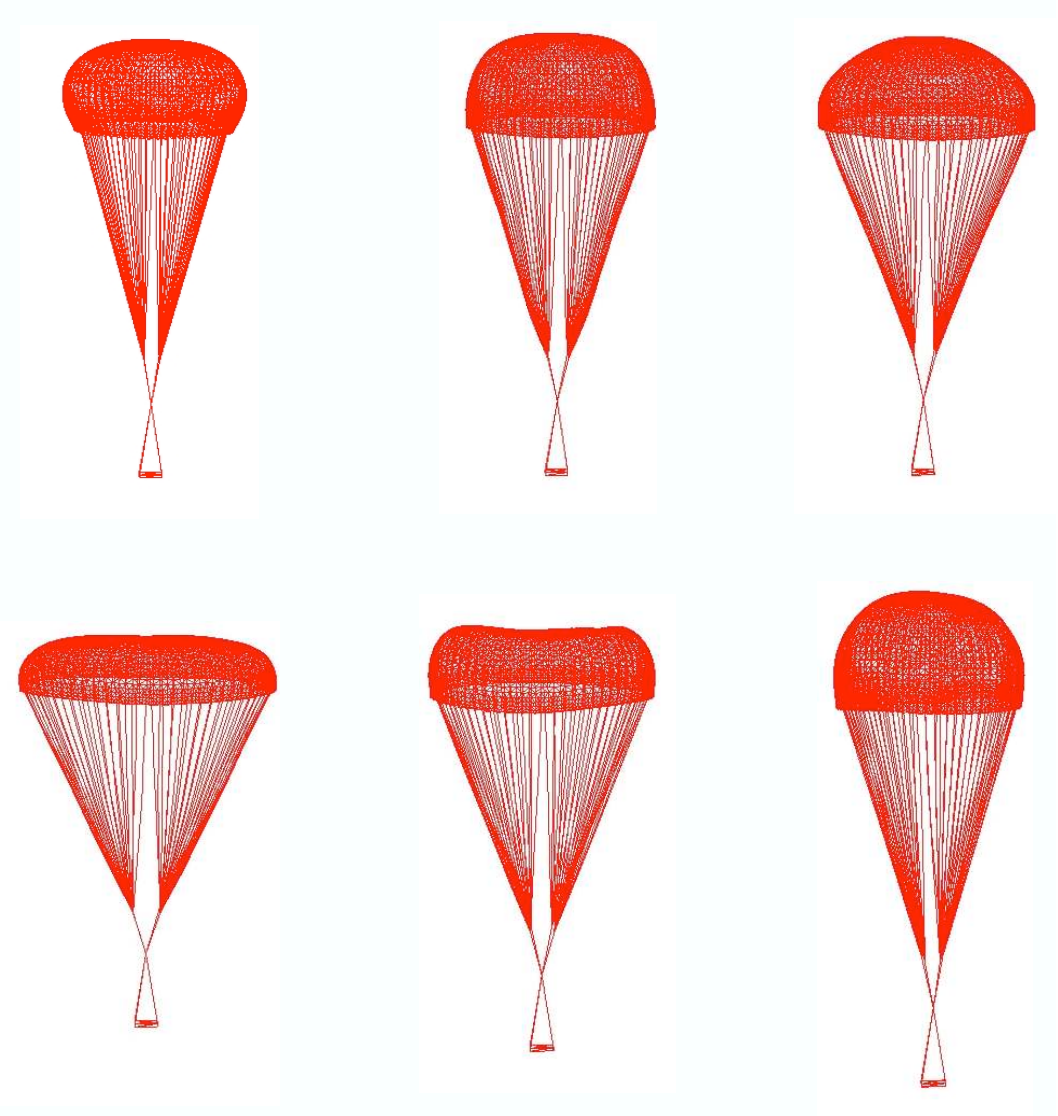

Figure 8: Disreefing of a G–12 parachute. Parachute shape at various instants during disreefing (left to right and top to bottom).

[19] K.R. Stein, R.J. Benney, T.E. Tezduyar, J.W. Leonard, and M.L. Accorsi, "Fluid–structure interactions of a round parachute: Modeling and simulation techniques", *Journal of Aircraft*, **38** (2001) 800–808.

[20] T.E. Tezduyar, "Finite element methods for fluid dynamics with moving boundaries and interfaces", in E. Stein, R. De Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, Chapter 17, John Wiley & Sons, 2004.

[21] S. Mittal and T.E. Tezduyar, "A finite element study of incompressible flows past oscillating cylinders and aerofoils", *International Journal for Numerical Methods in Fluids*, **15** (1992) 1073–1118.

[22] S. Mittal and T.E. Tezduyar, "Parallel finite element simulation of 3D incompressible flows – Fluid-structure interactions", *International Journal for Numerical Methods in Fluids*, **21** (1995) 933–953.

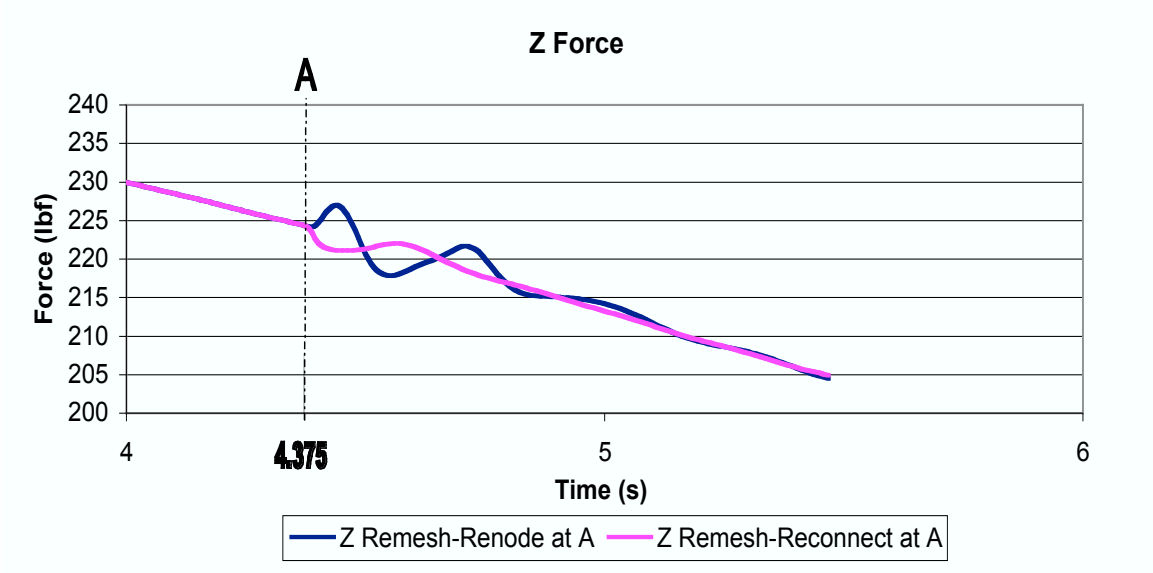

Figure 9: Remeshing performance for rigid-body rotation of T–10 parachute. The aerodynamical forces acting on the parachute in the z direction, obtained with the "reconnect" and "renode" options.

[23] T. Tezduyar and Y. Osawa, "Fluid–structure interactions of a parachute crossing the far wake of an aircraft", *Computer Methods in Applied Mechanics and Engineering*, **191** (2001) 717–726.

[24] A.E. Green and J.E. Adkins, *Large Elastic Deformations*. Oxford Clarendon Press, Amen House, London, U.K., 1960.

[25] A.E. Green and W. Zerna, *Theoretical Elasticity*. Oxford Clarendon Press, Ely House, London, U.K., 1968.

[26] M.L. Accorsi, J.W. Leonard, R. Benney, and K. Stein, "Structural modeling of parachute dynamics", *AIAA Journal*, **38** (2000) 139–146.

[27] T.E. Tezduyar and Y. Osawa, "Finite element stabilization parameters computed from element matrices and vectors", *Computer Methods in Applied Mechanics and Engineering*, **190** (2000) 411–430.

[28] T.E. Tezduyar, "Computation of moving boundaries and interfaces and stabilization parameters", *International Journal for Numerical Methods in Fluids*, **43** (2003) 555–575.

[29] H.M. Hilber, T.J.R. Hughes, and R.L. Taylor, "Improved numerical dissipation for time integration algorithms in structural dynamics", *Earthquake Engineering and Structural Dynamics*, **5** (1977) 283–292.

[30] D.R. Lynch, "Wakes in liquid-liquid systems", *Journal of Computational Physics*, **47** (1982) 387–411.

[31] A. Masud and T.J.R. Hughes, "A space–time Galerkin/least-squares finite element formulation of the Navier-Stokes equations for moving domain problems", *Computer Methods in Applied Mechanics and Engineering*, **146** (1997) 91–126.

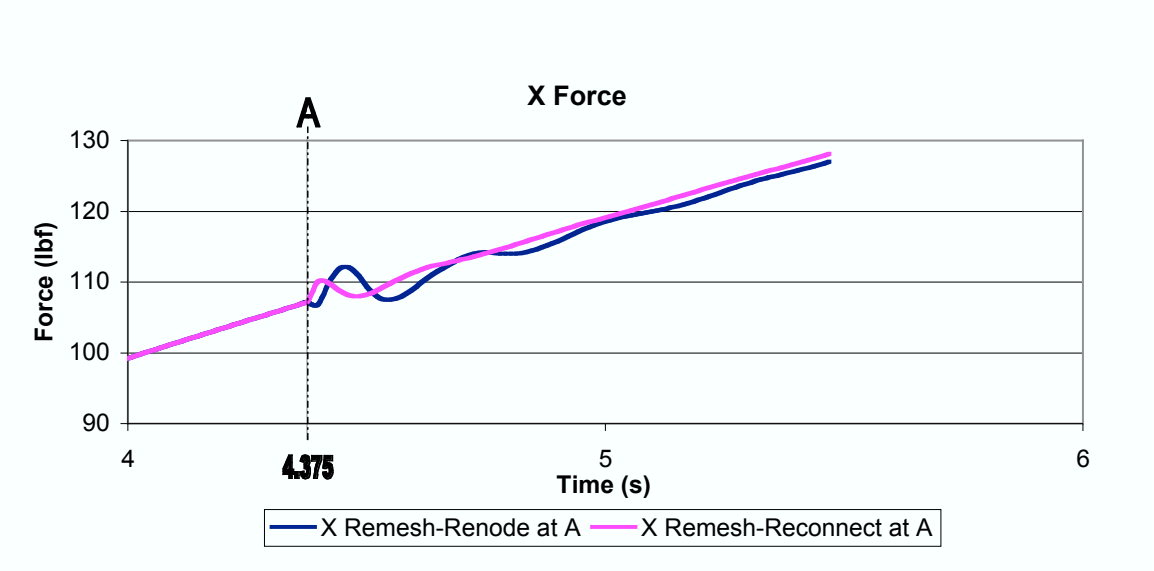

Figure 10: Remeshing performance for rigid-body rotation of T–10 parachute. The aerodynamical forces acting on the parachute in the x direction, obtained with the "reconnect" and "renode" options.

[32] K. Stein, T. Tezduyar, and R. Benney, "Mesh moving techniques for fluid–structure interactions with large displacements", *Journal of Applied Mechanics*, **70** (2003) 58–63.

[33] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal, "Parallel finite-element computation of 3D flows", *Computer*, **26** (1993) 27–36.

[34] K. Stein and T. Tezduyar, "Advanced mesh update techniques for problems involving large displacements", in *Proceedings of the Fifth World Congress on Computational Mechanics*, On-line publication: http://wccm.tuwien.ac.at/, Paper-ID: 81489, Vienna, Austria, (2002).

[35] K. Stein, T.E. Tezduyar, and R. Benney, "Automatic mesh update with the solid-extension mesh moving technique", *Computer Methods in Applied Mechanics and Engineering*, **193** (2004) 2019–2032.

[36] A.A. Johnson and T.E. Tezduyar, "Simulation of multiple spheres falling in a liquid-filled tube", *Computer Methods in Applied Mechanics and Engineering*, **134** (1996) 351–373.

[37] T. Fujisawa, M. Inaba, and G. Yagawa, "Parallel computing of high-speed compressible flows using a node-based finite element method", *International Journal for Numerical Methods in Fluids*, **58** (2003) 481–511.

[38] Y. Saad and M. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM Journal of Scientific and Statistical Computing*, **7** (1986) 856–869.

[39] T.E. Tezduyar, "Stabilized finite element methods for computation of flows with moving boundaries and interfaces", in *Lecture Notes on Finite Element Simulation of Flow Problems (Basic - Advanced Course)*, Japan Society of Computational Engineering and Sciences, Tokyo, Japan, (2003).

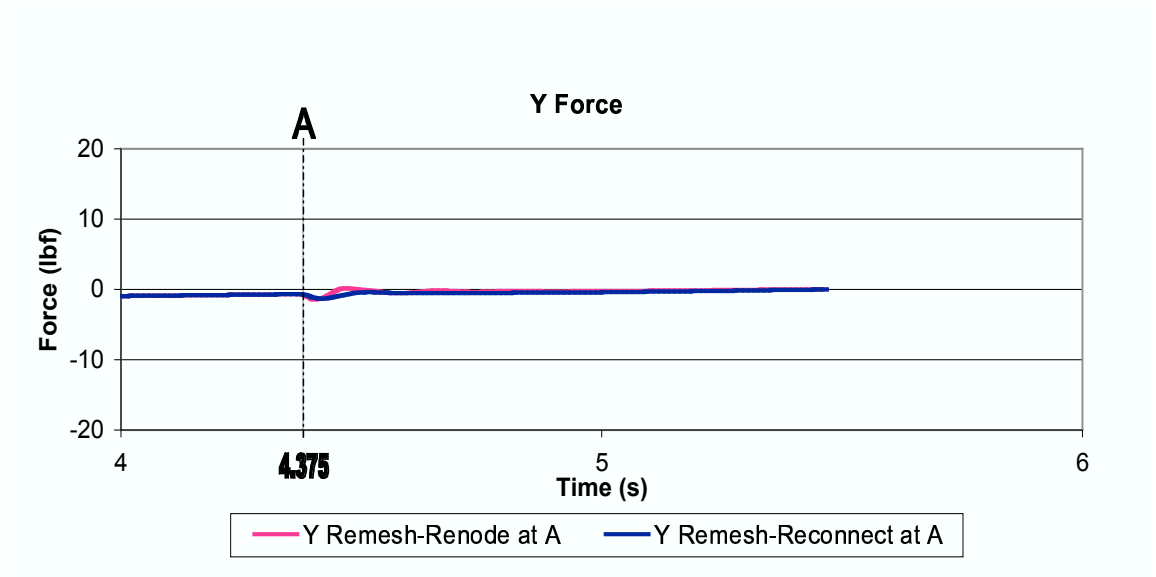

Figure 11: Remeshing performance for rigid-body rotation of T–10 parachute. The aerodynamical forces acting on the parachute in the y direction, obtained with the "reconnect" and "renode" options.

[40] T. Hisada, H. Watanabe, and S. Sugiura. "Fluid–structure interaction analysis of human heart by ALE finite element method", 2003, presentation at the Seventh US National Congress on Computational Mechanics, Albuquerque, New Mexico.

[41] S. Sathe, *Enhanced-Discretization and Solution Techniques in Flow Simulations and Parachute Fluid–Structure Interactions*, Ph.D. thesis, Rice University, 2004.