

Fluid–structure interaction modeling of clusters of spacecraft parachutes with modified geometric porosity

Kenji Takizawa · Tayfun E. Tezduyar · Joseph Boben ·
Nikolay Kostov · Cody Boswell · Austin Buscher

Received: 24 April 2013 / Accepted: 17 May 2013 / Published online: 13 July 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract To increase aerodynamic performance, the geometric porosity of a ringsail spacecraft parachute canopy is sometimes increased, beyond the “rings” and “sails” with hundreds of “ring gaps” and “sail slits.” This creates extra computational challenges for fluid–structure interaction (FSI) modeling of clusters of such parachutes, beyond those created by the lightness of the canopy structure, geometric complexities of hundreds of gaps and slits, and the contact between the parachutes of the cluster. In FSI computation of parachutes with such “modified geometric porosity,” the flow through the “windows” created by the removal of the panels and the wider gaps created by the removal of the sails cannot be accurately modeled with the Homogenized Modeling of Geometric Porosity (HMGP), which was introduced to deal with the hundreds of gaps and slits. The flow needs to be actually resolved. All these computational challenges need to be addressed simultaneously in FSI modeling of clusters of spacecraft parachutes with modified geometric porosity. The core numerical technology is the Stabilized Space–Time FSI (SSTFSI) technique, and the contact between the parachutes is handled with the Surface–Edge–Node Contact Tracking (SENCT) technique. In the computations reported here, in addition to the SSTFSI and SENCT techniques and HMGP, we use the special techniques we have developed for removing the numerical spinning component of the parachute motion and for restoring the mesh integrity without a

remesh. We present results for 2- and 3-parachute clusters with two different payload models.

Keywords Fluid–structure interaction · Parachutes · Space–time techniques · Ringsail parachutes · Parachute clusters · Contact · Modified geometric porosity

1 Introduction

Fluid–structure interaction (FSI) modeling of ringsail spacecraft parachute clusters poses a number of computational challenges [1,2]. These include the lightness of the parachute canopy compared to the air masses involved in the parachute dynamics, the geometric porosity created by the construction of the canopy from “rings” and “sails” with hundreds of “ring gaps” and “sail slits,” and the contact between the parachutes of the cluster. The Team for Advanced Flow Simulation and Modeling (T★AFSM) has been addressing these computational challenges with the Stabilized Space–Time FSI (SSTFSI) technique [3], which was developed and improved over the years by the T★AFSM and serves as the core numerical technology, and a number of special techniques developed in conjunction with the SSTFSI technique.

The SSTFSI technique originates from the Deforming–Spatial–Domain/Stabilized ST (DSD/SST) method [4–7] and its new versions [3,8,9]. The DSD/SST formulation is a general-purpose moving-mesh (interface-tracking) method for flows with moving interfaces. Its stabilization parts are the Streamline–Upwind/Petrov–Galerkin (SUPG) [10] and Pressure–Stabilizing/Petrov–Galerkin (PSPG) [4,11] methods. The DSD/SST method is used with the advanced mesh update methods [3,12–15] developed by the T★AFSM. Mesh update includes moving the mesh for as long as possible and remeshing when needed. The ST approach, with higher-

K. Takizawa
Department of Modern Mechanical Engineering and Waseda
Institute for Advanced Study, Waseda University,
1-6-1 Nishi-Waseda, Shinjuku-ku, Tokyo 169-8050, Japan

T. E. Tezduyar (✉) · J. Boben · N. Kostov · C. Boswell · A. Buscher
Department of Mechanical Engineering, Rice University—MS 321,
6100 Main Street, Houston, TX 77005, USA
e-mail: tezduyar@rice.edu

order functions in time, gives us more effective ways of mesh moving and remeshing [16–20]. While the Arbitrary Lagrangian–Eulerian (ALE) finite element formulation [21] is the most commonly used moving-mesh approach in FSI computations [22–38], the DSD/SST formulation now also has a good record of being applied to some of the most challenging FSI computations (see [1–3, 9, 39–42] and references therein).

Parachute FSI computations of the T★AFSM with the DSD/SST method precede the development of the SSTFSI technique and the associated special techniques. These computations started as early as 1997 [43], with 3D computations going as far back as 2000 [44, 45], and with a good number of parachute FSI problems solved [46–52] before reaching the SSTFSI technique. However, it was the SSTFSI technique, and the special techniques developed in conjunction with it, that brought the parachute FSI computations to a new era in addressing some of the most formidable computational challenges and truly supporting actual parachute design and testing [1, 3, 19, 41, 53–60].

In an FSI computation with a moving-mesh method, the FSI coupling technique determines how the coupling between the equation blocks representing the fluid mechanics, structural mechanics, and mesh moving equations is handled. The coupling techniques used in the T★AFSM parachute computations evolved from block-iterative FSI coupling [61] (see [3, 51] for the terminology) used in the computations reported in [44–47] to a more robust version of block-iterative coupling [51, 52, 61, 62] and to quasi-direct coupling [51, 52] and direct coupling [51, 52] techniques. The quasi-direct and direct coupling techniques, which are applicable to cases with nonmatching fluid and structure meshes at the interface, yield more robust algorithms for FSI computations where the structure is light, such as parachute FSI computations. The SSTFSI technique is based on the new versions of the quasi-direct and direct coupling techniques with upgraded and additional interface projection methods [3, 53, 55, 56, 63], has a substantially increased robustness in FSI computations, and rendered the earlier ST FSI solvers obsolete. These new quasi-direct and direct coupling techniques automatically reduce to “monolithic” coupling when the interface has matching fluid and structure meshes. Allowing nonmatching meshes at the interface substantially increases the scope of the FSI solver, leading to success in FSI modeling of challenging problems, such as ringsail spacecraft parachutes [1, 19, 41, 53–60].

A good number of special FSI techniques were introduced in [3, 19, 53, 55–60, 63] in conjunction with the SSTFSI technique. These special techniques are mostly in the category of interface projection techniques. They include the FSI Geometric Smoothing Technique (FSI-GST) [3], Separated Stress Projection (SSP) [53, 56], Homogenized Modeling of Geometric Porosity (HMGP) [53], adaptive HMGP [55],

“symmetric FSI” method [55], accounting for fluid forces acting on structural components (e.g. parachute suspension lines) that are not expected to influence the flow [55], new versions of the HMGP that are called “HMGP-FG” [56] and “HMGP-FGR” [60], and other interface projection techniques [63]. The special FSI techniques in other categories include the Surface-Edge-Node Contact Tracking (SENCT) technique [3], which is a contact algorithm, multiscale sequentially-coupled FSI techniques [55, 57], rotational-periodicity techniques [56, 57], a new, conservative version of the SENCT technique [58], computed-data reduction techniques [58, 59], intra-canopy versions of the contact algorithm [60], and using higher-order temporal functions in mesh moving [19].

The ringsail spacecraft parachutes the T★AFSM has been focusing on are very large, made of a large number of gores. A gore is the slice of the canopy between two radial reinforcement cables running from the parachute vent to the skirt. The construction of the canopy from rings and sails happens at the gore level. With the HMGP, we bypass the intractable complexities of the geometric porosity by approximating it at the fluid interface with an “equivalent,” locally-varying “homogenized” porosity. This is obtained from an HMGP computation with an n -gore slice of the parachute canopy where the flow through the ring gaps and sail slits is actually resolved (see [53, 54, 56, 57, 60] for details). In the earlier HMGP computations with a 4-gore slice, slip conditions were applied on the boundaries intersecting the canopy. With the rotational-periodicity techniques, less constraining conditions can be imposed on those boundaries [56, 57].

Spacecraft parachutes are typically used in clusters of two or three parachutes. The computational challenge associated with the contact between the parachutes of a cluster is addressed with the conservative version of the SENCT technique [58], which is also more robust than the original SENCT technique [3]. During the FSI computation, there might also be a contact within a canopy. This could be a contact between the gores of a parachute canopy, or even a contact between the nodes of a gore. These computational challenges are addressed with the intra-canopy versions of the contact algorithm [60].

As an additional computational challenge, the ringsail parachute canopy might, by design, have some of its panels and sails removed. The purpose is to increase the aerodynamic performance of the parachute. In FSI computation of parachutes with such “modified geometric porosity,” the flow through the “windows” created by the removal of the panels and the wider gaps created by the removal of the sails cannot be accurately modeled with the HMGP and needs to be actually resolved. This and the other computational challenges described in the earlier paragraphs all need to be addressed simultaneously in FSI modeling of clusters of spacecraft parachutes with modified geometric porosity. This

is what we have succeeded in doing to a large extent for the computations reported in this paper.

In the parachute FSI computations reported here, we also use two special techniques that we have developed. With one of the special techniques we remove the numerical spinning component of the parachute motion. We believe that this nonphysical spinning has its roots in two of our modeling features: (a) the parachute risers are modeled with cable structures, which are currently unable to represent the torsional stiffness that the risers should have, (b) in the HMGP, the fluid interface does not see the “valleys” and “peaks” of the gores that the structure interface has, currently not even at some milder level, offering basically no aerodynamic resistance to spinning. Therefore, once the spinning is triggered, for example by an unsymmetric vortex, there is essentially no resistance to it. With the other one of the special techniques, we restore the mesh integrity lost during the mesh motion without resorting to remeshing. The loss of mesh integrity, though not frequent because of the advanced mesh moving methods we are using, should be expected in FSI computations with the level of complexity we have in clusters of ringsail parachutes with modified geometric porosity. When we face such a loss of mesh integrity, as an alternative to remeshing, we use a technique where the mesh is “relaxed” without altering the mesh at the fluid–structure interface and thus the mesh integrity is restored to some extent. This is of course a less costly and less disruptive alternative to remeshing.

These two special techniques are described in Sect. 2. The computational conditions and results for 2- and 3-parachute clusters with two different payload models are presented in Sect. 3, and the concluding remarks are given in Sect. 4.

2 Special techniques

2.1 Removing spinning

We first define, as described in Sect. 4.1.2 of [58], the individual parachute axes:

$$\mathbf{r}_k \equiv (\mathbf{x}_c)_k - \mathbf{x}_p, \tag{1}$$

where k indicates the k th parachute, and $(\mathbf{x}_c)_k$ and \mathbf{x}_p are the centroid of the parachute and the confluence of the cluster, respectively, and obtain

$$(\mathbf{g}_r)_k = \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|}. \tag{2}$$

We consider the angular momentum

$$\mathbf{L}_k = \int_{\Omega_k} \boldsymbol{\zeta} \times \rho \mathbf{u} d\Omega, \tag{3}$$

where ρ is the density, \mathbf{u} is the velocity, and $\boldsymbol{\zeta}$ is the vector from the origin. Based on that, we define an average angular velocity:

$$\bar{\boldsymbol{\omega}}_k = (\mathbf{I}_{\text{total}})_k^{-1} \mathbf{L}_k, \tag{4}$$

where $(\mathbf{I}_{\text{total}})_k$ is the inertia tensor. We want to remove the component of that in the $(\mathbf{g}_r)_k$ direction:

$$(\boldsymbol{\omega}_S)_k = (\bar{\boldsymbol{\omega}}_k \cdot (\mathbf{g}_r)_k) (\mathbf{g}_r)_k. \tag{5}$$

That would lead to the following new velocity for the points of the k th parachute:

$$\mathbf{u}_{RS} = \mathbf{u} - (\boldsymbol{\omega}_S)_k \times \boldsymbol{\zeta}. \tag{6}$$

Remark 1 Choosing the origin to be along the parachute axis, which passes through the parachute centroid, makes the velocity modification unique. Here we choose the origin to be the confluence of the cluster.

2.2 Mesh relaxation

In the mesh relaxation technique we propose here, the new mesh will have the same number of nodes and elements as before, but certain nodes are moved slightly to improve the quality of certain elements within the domain. For that, we use the large-deformation solid mechanics equations rather than the linear-elasticity equations with Jacobian-based stiffening [3, 12–15] that we use in mesh moving. One of the advantages of this is the ability to choose from many constitutive models and include geometric stiffness, which means that we can define the undeformed shapes in arbitrary orientations.

To add more flexibility to the equations we are using, we introduce an element-based undeformed shape Ω_0^e for the e th element. This is essentially a shape generated for each element. In general we do not need to relate it to the current domain, but in the mesh relaxation technique we propose here, we do. Given mesh position $\mathbf{X}_{\text{REF}} \in \Omega_{\text{REF}}$ for the current mesh, we solve for the displacement \mathbf{y} and obtain a better quality mesh with $\mathbf{X}_{\text{REF}} + \mathbf{y}$. In solving the solid mechanics equations, we integrate over each element domain Ω_0^e , and the full displacement for an element node a is obtained as follows:

$$\mathbf{x}_a^e = (\mathbf{X}_{\text{REF}}^e)_a - (\mathbf{X}_0^e)_a + \mathbf{y}_a^e. \tag{7}$$

By design, the undeformed shape is the shape we want to obtain from solving the solid mechanics equations. We now call those elements “target elements.” There are several options for constructing the target element shapes, and the current mesh could be thought of as the starting point.

The first option is setting a minimum desired element volume. If an element in the current mesh has a volume less than that, the volume of the element is increased incrementally

until it reaches the minimum desired volume. This option is useful for increasing the volume of specific elements in a mesh relative to other elements.

The second option is a global volume scaling. It is a factor by which all element volumes are multiplied, thus placing all elements in the mesh in either tension or compression. This is sometimes important for obtaining a stable solution from the solid mechanics equations.

The third option is to shape the element such that it is a regular tetrahedron, has its shape in the current mesh we are relaxing, or is a linear interpolation between the two. The linear interpolation needs to be done carefully. First we define a regular tetrahedron $\mathbf{X}_{\text{REG}}^e$ such that the volume is the same as the volume in the current mesh and the centroid is at the origin. With that, we calculate the following deformation gradient tensor at the element center:

$$\mathbf{F} = \frac{\partial \mathbf{X}_{\text{REF}}^e}{\partial \mathbf{X}_{\text{REG}}^e}, \quad (8)$$

and obtain the right Cauchy–Green tensor:

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}. \quad (9)$$

After the polar decomposition of \mathbf{C} , we obtain the rigid-body rotation tensor \mathbf{R} . With this, we exclude the rigid-body rotation for an element node a :

$$\left(\hat{\mathbf{X}}_{\text{REG}}^e \right)_a = \mathbf{R} \left(\mathbf{X}_{\text{REG}}^e \right)_a, \quad (10)$$

and obtain the target element shape as

$$\left(\mathbf{X}_0^e \right)_a = (1 - s) \left(\mathbf{X}_{\text{REF}}^e \right)_a + s \left(\hat{\mathbf{X}}_{\text{REG}}^e \right)_a. \quad (11)$$

Here $0 \leq s \leq 1$, $s = 1$ leads to the target element that is a regular tetrahedron with the same volume as the volume in the current mesh, and $s = 0$ leads to the element in the current mesh.

The fourth, and the last, option is to specify a minimum aspect ratio to fix: A_{min} . For any element with an aspect ratio less than or equal to that, we use $s = 0$ in the interpolation given by Eq. (11). For any element that has an aspect ratio above A_{min} , we use a globally specified s value to relax the shape.

These options, listed in order of precedence, are combined to generate a database of target element shapes. With these element-specific unstressed shapes, the solid mechanics solver attempts to reduce the stress in the new mesh. This makes the elements in the new mesh more closely resemble the target elements, thus increasing the quality of the elements in the mesh.

Relaxing the mesh has several advantages over remeshing. It is much less time consuming, is easier to automate, and reduces the amount of error introduced by projecting data from one mesh to a new mesh.

3 Parachute clusters

The objective of the parachute cluster computations presented in this paper is to determine how the newest generation of the NASA Orion Capsule parachute system, the modified-porosity (MP) parachute, performs when used in cluster configurations. The purpose of the MP design is to, compared to previous designs, reduce cluster flyout angles, increase stability, and reduce canopy interactions.

The MP parachute modeled in this paper has the same physical parameters as the pad abort (PA) parachute modeled in [57,58], except it has a suspension-line to nominal diameter ratio, L_s/D_o , of 1.44 instead of $L_s/D_o = 1.15$. In addition, every 5th gore on Sail 11 is removed as well as the top 25 % of Sail 6. Each parachute in the cluster has 80 gores, a nominal diameter of about 120 ft, 4 rings and 9 sails. The material properties are the same as described in [60].

Four different fully-open MP cluster configurations are modeled, each with a different number of parachutes and payload configuration. A single initial coning angle (θ_{INIT}) is used for each cluster. All 2-parachute MP clusters (2-MP) in this paper have $\theta_{\text{INIT}} = 15^\circ$. All 3-parachute MP clusters (3-MP) have $\theta_{\text{INIT}} = 25^\circ$.

The payload is modeled as either a point mass at the confluence of the risers (PAC) or a series of truss elements (PTE) [58]. In drop tests, the parachutes are connected to a rectangular pallet that is designed to represent the mass and inertial properties of a proposed crew capsule. The PTE configuration distributes the payload mass at nine different points to match the mass, center of gravity, and six components of the inertia tensor of the pallet used in drop tests. The points are connected by 5 cable elements and 26 truss elements below the confluence (Fig. 1). To facilitate comparison to PA clus-

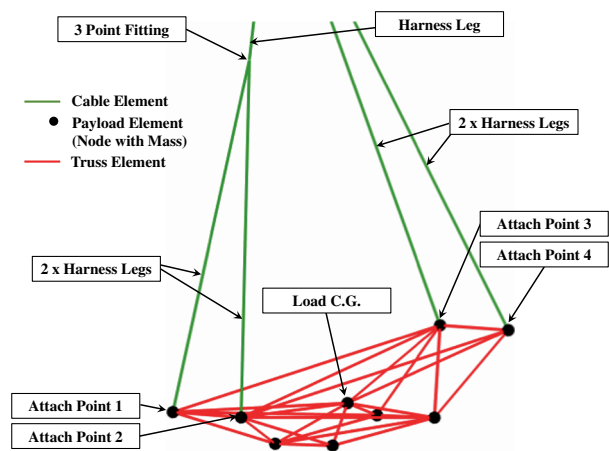


Fig. 1 Payload as a truss element (PTE) configuration showing the point masses and the cable and truss elements (figure from [64]). The cable elements are the four longer ones and the truss elements are the remaining, shorter ones

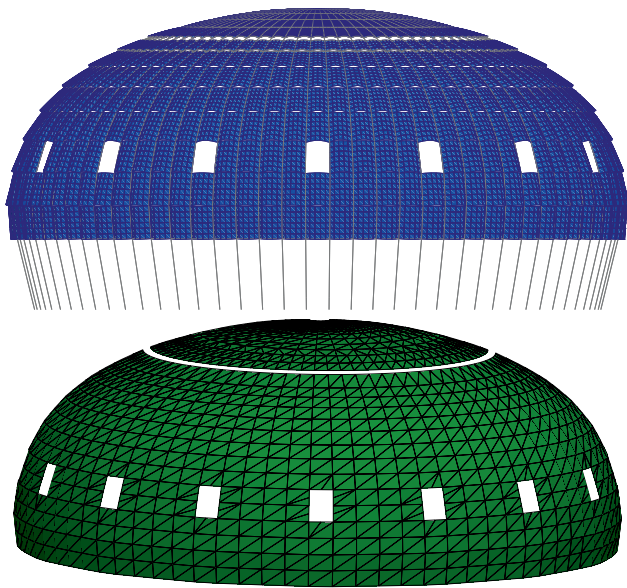


Fig. 2 Structural mechanics mesh (*top*) and fluid interface mesh (*bottom*) for a single MP parachute. For the number of nodes and elements at these interfaces, see Table 1

Table 1 Number of nodes (*m*) and elements (*ne*) for the parachute clusters

		2-MP	3-MP
Structure			
	<i>mn</i>	58,147	87,220
Membrane	<i>ne</i>	48,480	72,720
Cable	<i>ne</i>	24,642	36,963
Payload	<i>ne</i>	1	1
Interface	<i>mn</i>	55,104	82,656
	<i>ne</i>	48,480	72,720
Fluid			
Volume	<i>mn</i>	335,212	307,217
	<i>ne</i>	1,966,851	1,786,212
Interface	<i>mn</i>	4,472	6,708
	<i>ne</i>	8,104	12,156

The PTE configuration has 10 more structure nodes (8 of them point masses), 5 more cable elements, and 26 truss elements

ters reported in [58], PAC computations use a payload mass of 16,704 lbs, while PTE computations have total payload mass of 19,200 lbs.

Remark 2 In the computations reported in [58], the payload mass of the PAC configuration was incorrectly reported as 19,200 lbs. The correct weight for this specific configuration is 16,704 lbs, and that is what was used in the computations reported in [58].

All computations are carried out using air properties at standard sea-level conditions. The density is 2.38×10^{-3} slug/ft³. The kinematic viscosity is 1.57×10^{-4} ft²/s.

Table 2 HMGP coefficients k_F and k_G used in the PA parachute computations reported in [57–59] (note that Patch 1 and Patch 14 do not have a k_G value because those patches do not contain a ring gap or sail slit)

Patch	k_F (CFM)	k_G
1	125.5	
2	115.0	0.967
3	115.0	0.971
4	75.8	0.960
5	39.9	0.949
6	39.8	0.756
7	39.7	0.769
8	39.7	0.824
9	39.6	0.831
10	68.8	0.816
11	98.3	0.819
12	97.7	0.820
13	97.0	0.867
14	97.1	

We use these values also in the MP parachute computations reported in this paper, with some special treatment where we created windows and wide gaps. For the patch where we created a wide gap, we use only the corresponding k_F value. For the two gore patches that contain a window, we use only the two corresponding k_F values. This is what was done also in the single-MP parachute computations reported in [60]

3.1 Starting conditions

We first build a starting condition for a single parachute. We begin with a parachute shape obtained with the symmetric FSI computation reported in [60]. We do another single-parachute symmetric FSI computation with a horizontal inflow velocity of $U_{ref} \sin(\theta_{INIT})$, where U_{ref} is the reference value of the payload descent speed. It is 30 ft/s for 2-MP, and 25.7 ft/s for 3-MP. We compute for three breathing cycles. We use the parachute shape and position corresponding to the time when the parachute skirt diameter is at its average value to assemble the cluster structural mechanics mesh. A cluster of parachutes is generated by duplicating and rotating the structure and interface meshes from the single parachute such that each parachute is at the specified θ_{INIT} and joined together at a confluence.

After that, we generate a fluid mechanics mesh and do a fluid mechanics computation, holding the structural parachute shapes and positions fixed. The inflow velocity is U_{ref} . Next, we do a fluid mechanics computation with a prescribed, time-dependent shape for all parachutes of the cluster. The time-dependent shape comes from the single-parachute symmetric FSI computation carried out earlier with a horizontal inflow velocity of $U_{ref} \sin(\theta_{INIT})$. We use the solution from the fluid mechanics computation with prescribed parachute motion as the starting condition for the FSI computation.

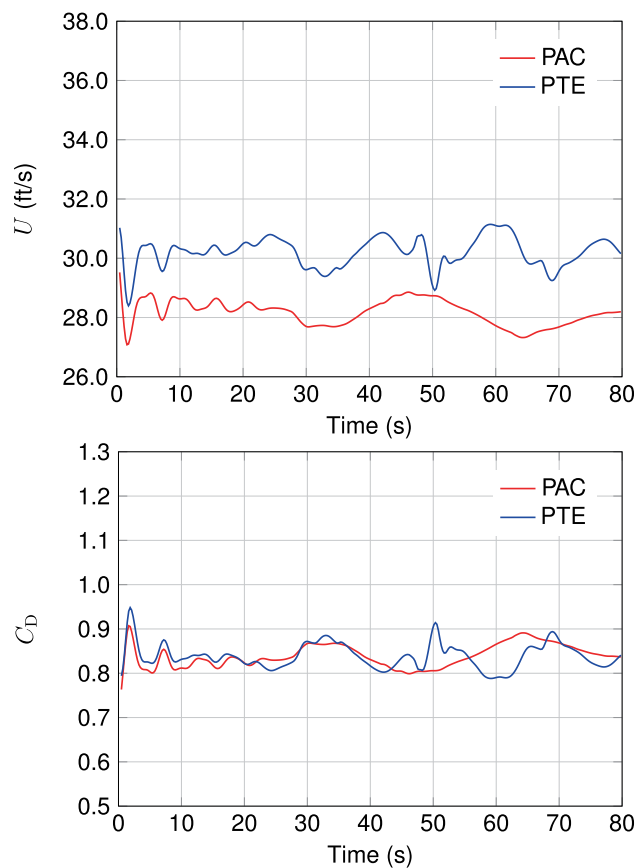


Fig. 3 Payload descent speed and drag coefficient for 2-MP with PAC and PTE

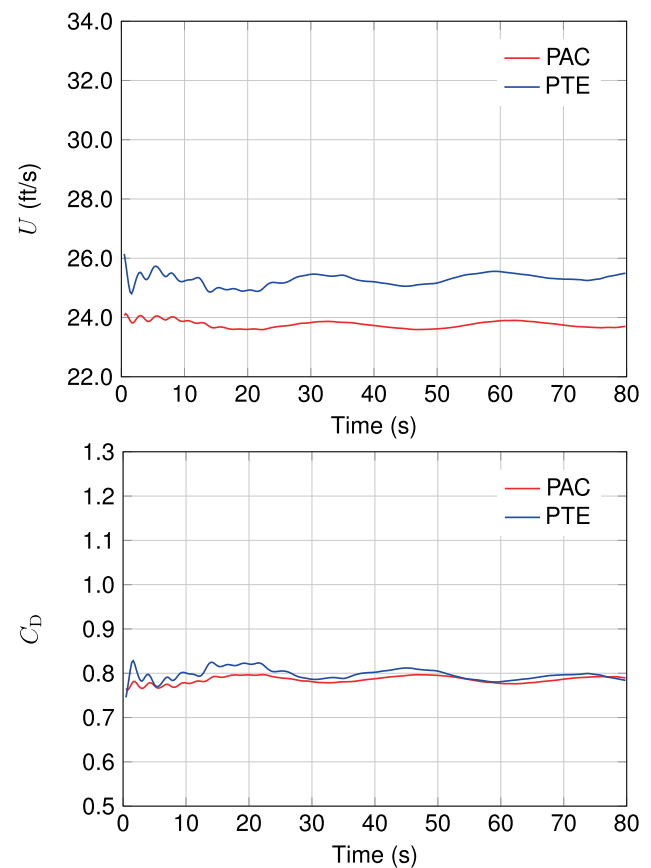


Fig. 4 Payload descent speed and drag coefficient for 3-MP with PAC and PTE

3.2 Computational methods and parameters

The computational domain is cylindrical with a diameter of 1,740 ft and a height of 1,566 ft. Figure 2 shows, for a single parachute, the canopy structure mesh and the fluid mechanics interface mesh. The fluid mechanics volume mesh consists of four-node tetrahedral elements, and the membrane elements used in the parachute structure are quadrilateral. The number of nodes and elements for the fluid and structure are given in Table 1. We move the reference frame with a vertical velocity of U_{ref} , and translate the mesh horizontally and vertically with the average displacement rate of the structure beyond the reference velocity U_{ref} . We use the velocity form of the free-stream conditions at the lateral boundaries as well since the mesh translates horizontally.

All computations are carried out in a parallel computing environment. The meshes are partitioned to enhance the parallel efficiency of the computations. Mesh partitioning is based on the METIS [65] algorithm. In solving the linear equation systems encountered at every nonlinear iteration, the GMRES search technique [66] is used with a diagonal preconditioner.



Fig. 5 2-MP with PTE at $t = 44.08$ s and $t = 46.40$ s

In the symmetric FSI computation with a single MP for three breathing cycles, we use the SSTFSI-TIP1 technique (see Remarks 5 and 10 in [3]), with the SUPG test function option WTSA (see Remark 2 in [3]). The stabilization parameters used are those given in [3] by Eqs. (9)–(12), (14)–(15) and (17), with the τ_{SUGN2} term dropped from Eq. (14). The



Fig. 6 2-MP with PTE at $t = 47.79$ s and $t = 51.04$ s



Fig. 7 3-MP with PTE at $t = 60.55$ s and $t = 67.28$ s



Fig. 8 3-MP with PTE at $t = 74.47$ s and $t = 79.11$ s

porosity model is HMGP-FGR [60]. The values of the HMGP “fabric porosity (k_F)” and “geometric porosity (k_G)” for the 14 “patches” we have (see [53,56] for the terminology) come essentially from the values used in the PA parachute computations reported in [57–59]. Those values are given in Table 2. The interface projection methods used include the SSP tech-

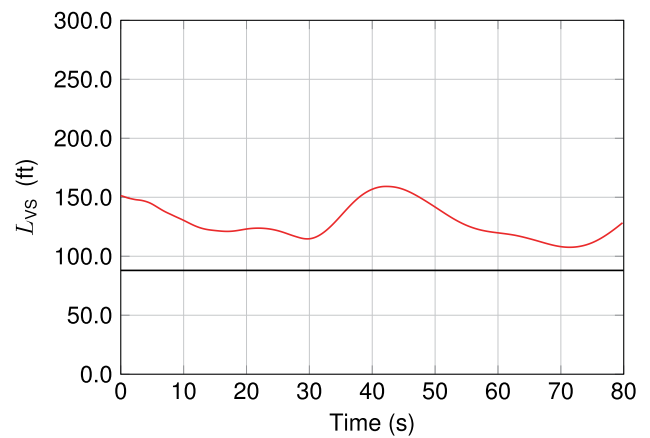


Fig. 9 Vent-separation distance for 2-MP with PAC

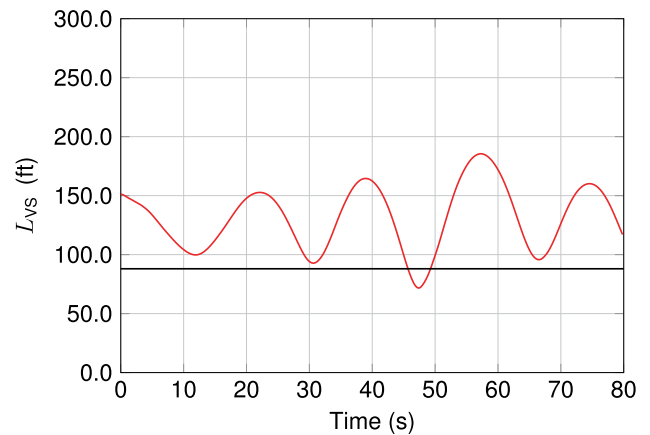


Fig. 10 Vent-separation distance for 2-MP with PTE

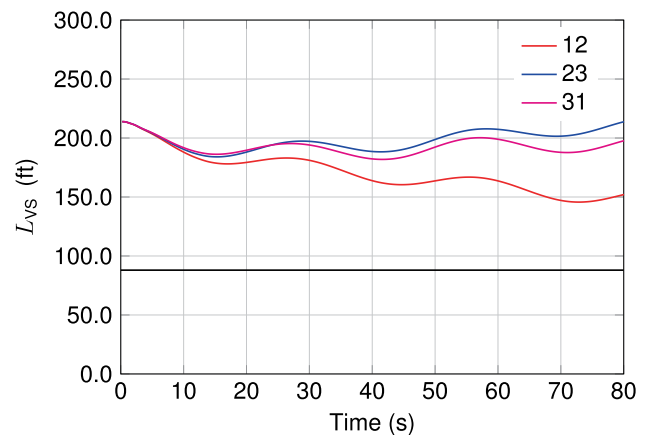


Fig. 11 Vent-separation distances for 3-MP with PAC

nique [53]. The fully-discretized, coupled fluid and structural mechanics and mesh-moving equations are solved with the quasi-direct coupling technique (see Sect. 5.2 in [3]). We use selective scaling [3], with the scale for the structure part set to 10 and for the other parts set to 1. The time-step size is 0.0232 s, with 6 nonlinear iterations per time step. The num-

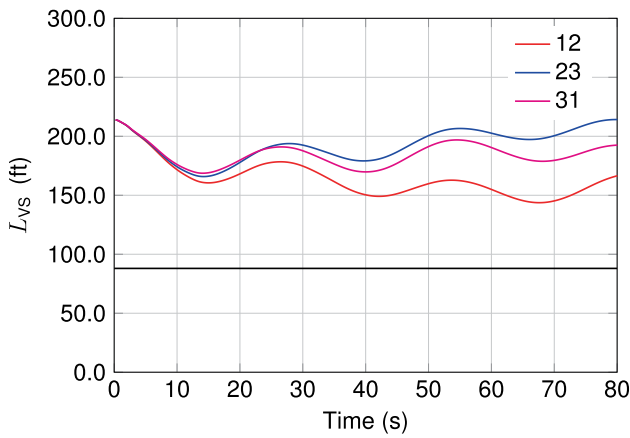


Fig. 12 Vent-separation distances for 3-MP with PTE

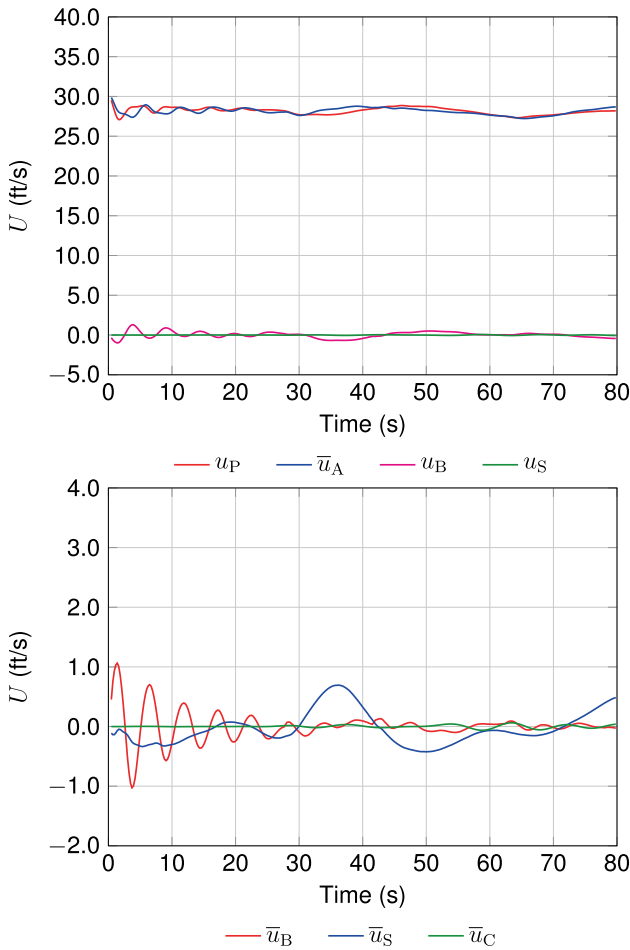


Fig. 13 Decomposition of the descent speed for 2-MP with PAC

ber of GMRES iterations per nonlinear iteration is 90 for the fluid + structure block, and 30 for the mesh-moving block.

The cluster fluid mechanics computations with fixed shapes and positions are done in two parts. The first part uses the semi-discrete formulation given in [7]. We compute 800 time steps with a time-step size of 0.232 s and 6

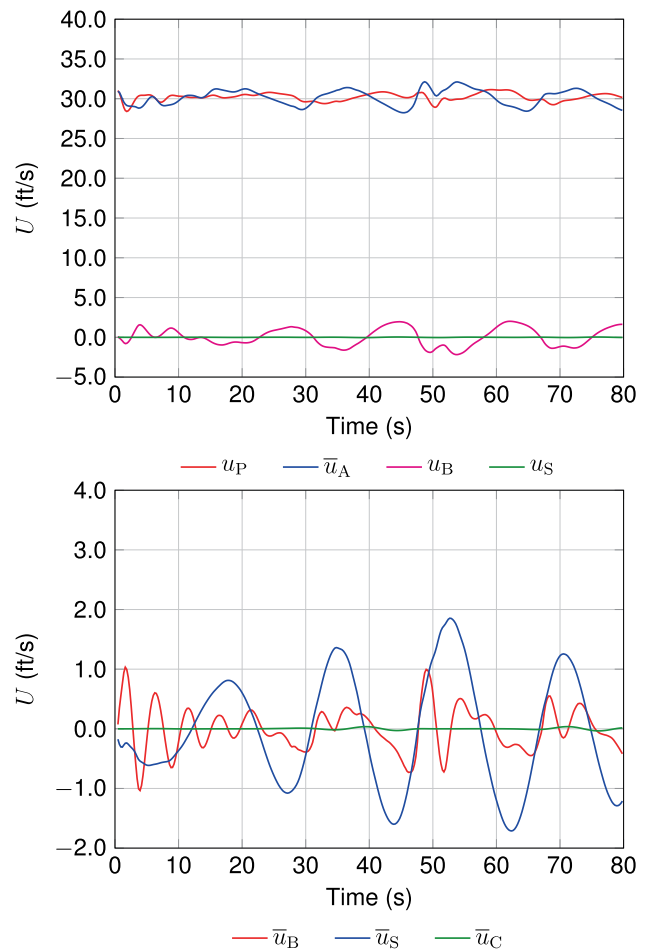


Fig. 14 Decomposition of the descent speed for 2-MP with PTE

nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90. There is no porosity model in the first part. The second part uses the DSD/SST-TIP1 technique [3], with the same SUPG test function option and stabilization parameters as those described above. We compute 1,800 time steps with a time-step size of 0.0232 s, 6 nonlinear iterations per time step, and 90 GMRES iterations per nonlinear iteration. The porosity model is HMGP-FG [56].

For the cluster fluid mechanics computations with prescribed, time-dependent shapes, we use the DSD/SST-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described above. The porosity model is HMGP-FGR. We compute roughly 300 time steps with a time-step size of 0.0232 s, with 6 nonlinear iterations per time step. The number of GMRES iterations per nonlinear iteration is 90 for the fluid mechanics block, and 30 for the mesh-moving block.

In the cluster FSI computations we use the SSTFSI-TIP1 technique, with the same SUPG test function option and stabilization parameters as those described above. The poros-

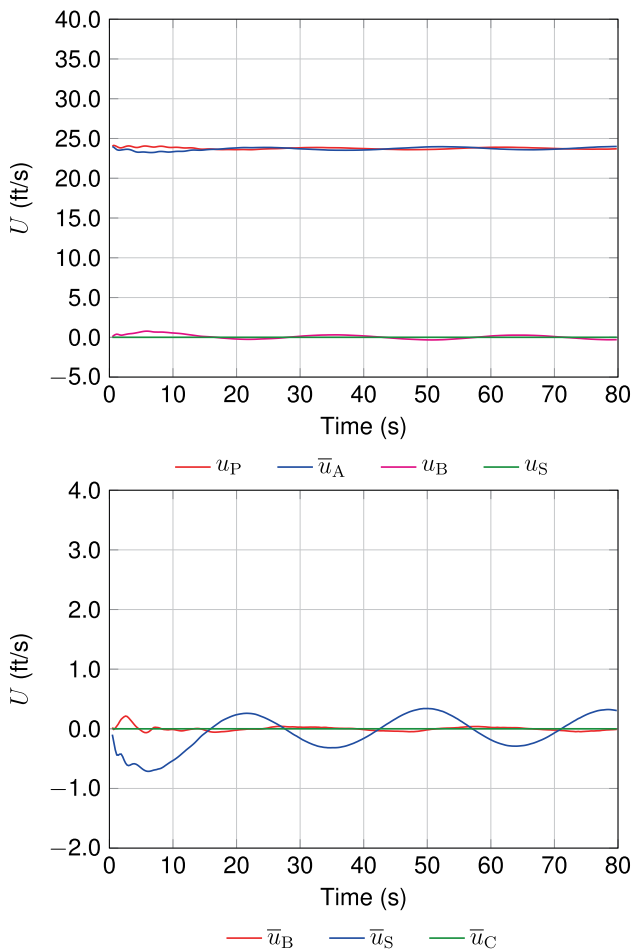


Fig. 15 Decomposition of the descent speed for 3-MP with PAC

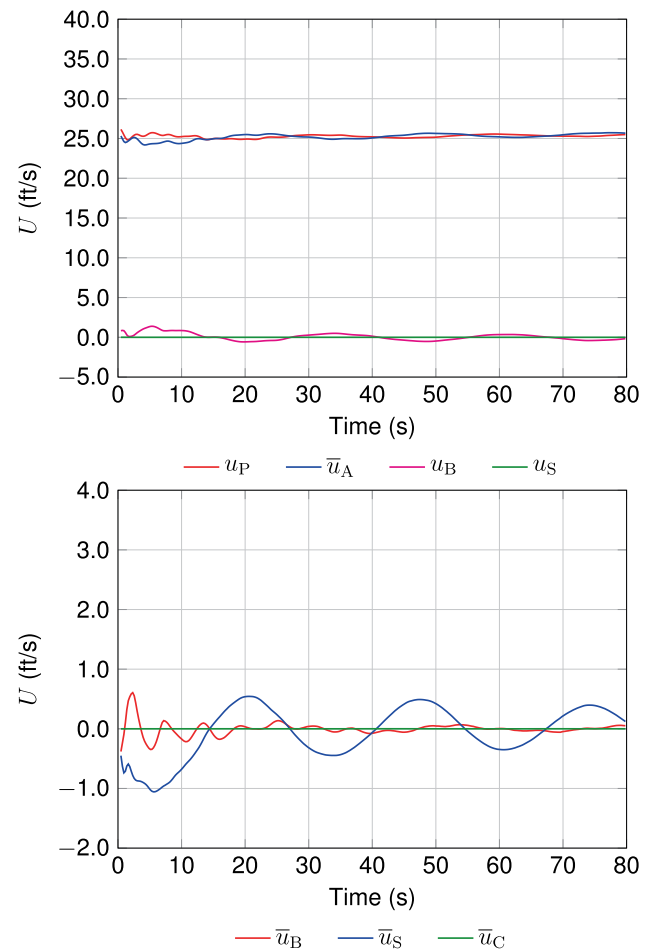


Fig. 16 Decomposition of the descent speed for 3-MP with PTE

ity model is HMGP-FGR. The interface projection methods used include the SSP technique. The SENCT-FC contact algorithm [58] is used with $\epsilon^S = \epsilon^C = 2.9$ ft. We use the quasi-direct coupling technique and selective scaling, with the scale for the structure part set to 1,000 and for the other parts set to 1. The time-step size is 0.0232 s, and the number of nonlinear iterations per time step is 6. The number of GMRES iterations per nonlinear iteration is 120 for the fluid + structure block, and 30 for the mesh-moving block.

We compute each parachute cluster for about 80 s. The numerical spinning component of the parachute motion is removed using the technique described in Sect. 2.1, approximately every 150 time steps. The mesh update for the fluid mechanics part includes, as needed, mesh relaxation, as described in Sect. 2.2, or remeshing. The frequency and the choice between mesh relaxation and remeshing vary for each computation and depend on how much the cluster rotates about the vertical axis and how much each parachute rotates about its own axis.

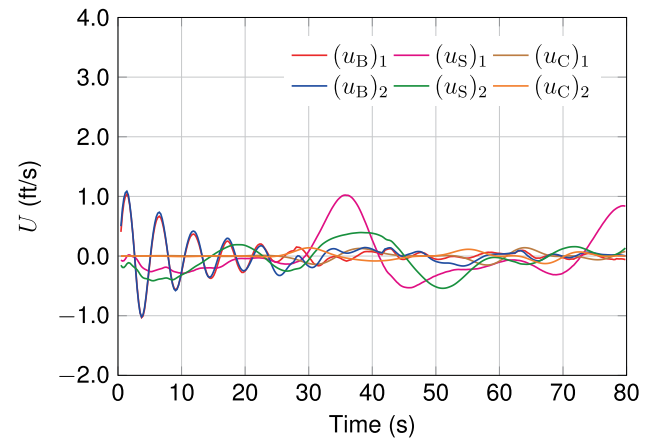


Fig. 17 Individual-parachute contributions to descent speed for 2-MP with PAC

3.3 Results

The critical measure of performance for the parachutes modeled in this paper is the payload descent speed, not only in terms of its average value, but also in terms of its fluctua-

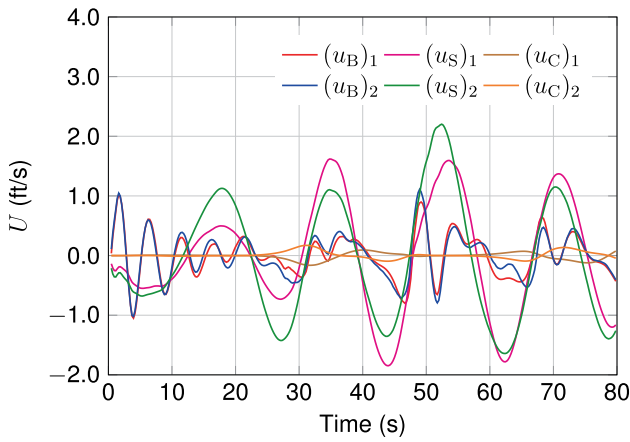


Fig. 18 Individual-parachute contributions to descent speed for 2-MP with PTE

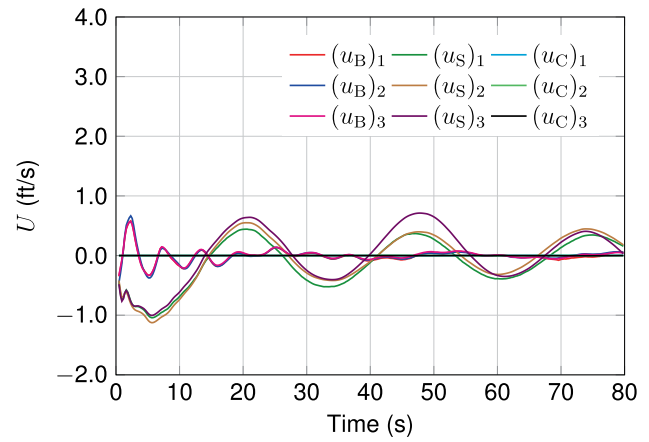


Fig. 20 Individual-parachute contributions to descent speed for 3-MP with PTE

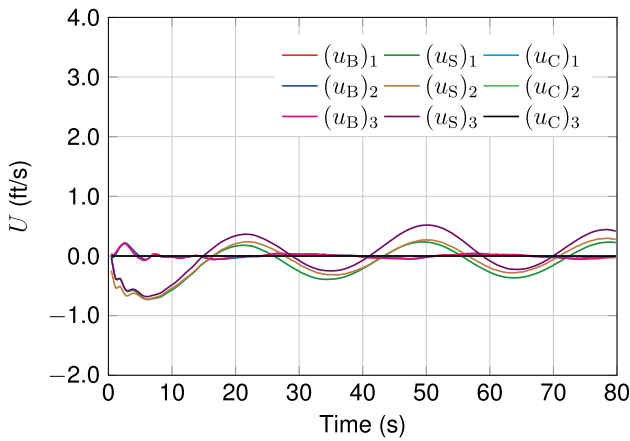


Fig. 19 Individual-parachute contributions to descent speed for 3-MP with PAC

tions. Drag coefficient is another way of expressing the performance, and is calculated as follows:

$$C_D = \frac{W}{\frac{1}{2}\rho U^2 S_o}, \tag{12}$$

where W is the payload weight, ρ is the density of the air, U is the payload descent speed, and S_o is the nominal area of the parachute. Figures 3 and 4 show the payload descent speed and drag coefficient for the MP parachute clusters. Figures 5, 6, 7 and 8 show the clusters at different instants. Figures 9, 10, 11 and 12 show the vent-separation distances (“ L_{VS} ”) for the clusters. The horizontal black line on each plot shows the approximate vent-separation distance when the parachutes are in contact.

To better understand the descent speed fluctuations, a method to decompose the payload velocity into components based on geometric contributing factors was intro-

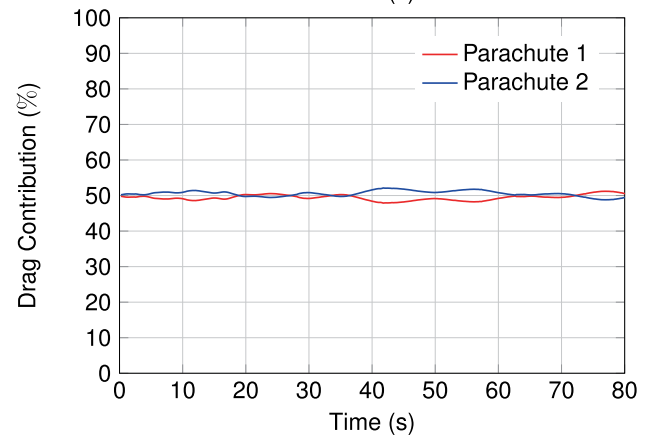
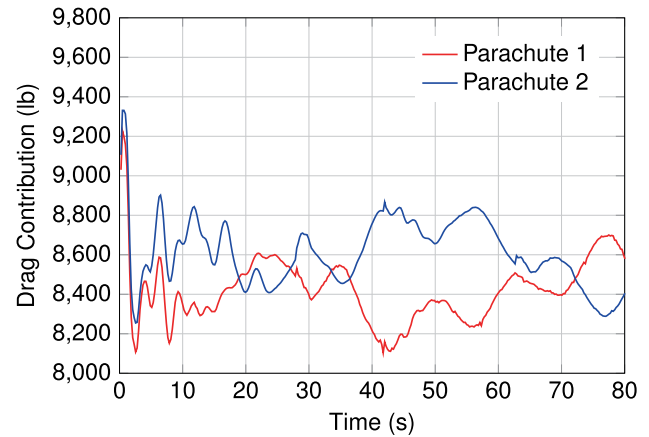


Fig. 21 Individual-parachute contributions to drag for 2-MP with PAC

duced in [58]. We use that method here to decompose the payload velocity of the MP parachute clusters. Figures 13, 14, 15 and 16 show the decomposition results. In those figures, as in [58], \bar{u}_B , \bar{u}_S , and \bar{u}_C refer to the aver-

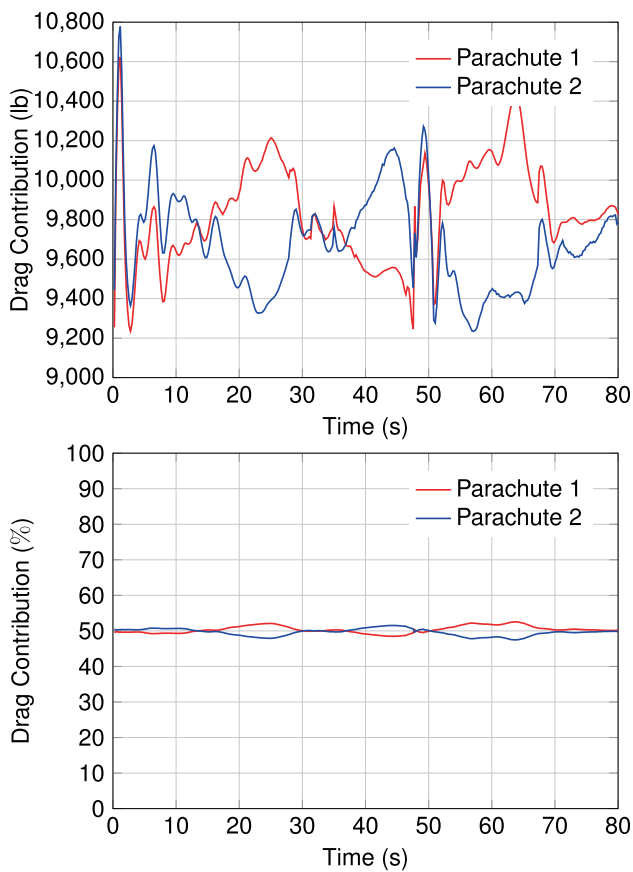


Fig. 22 Individual-parachute contributions to drag for 2-MP with PTE

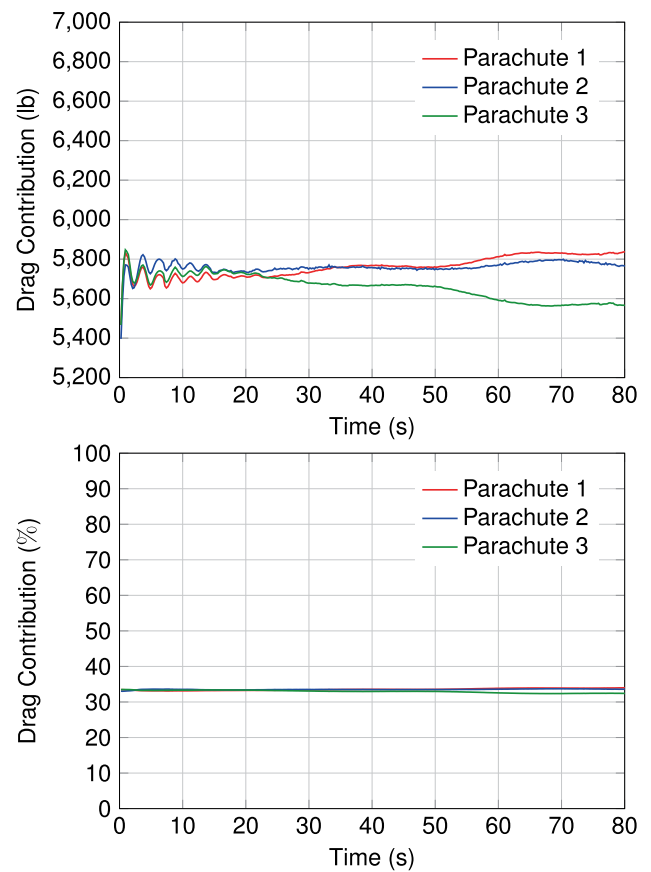


Fig. 23 Individual-parachute contributions to drag for 3-MP with PAC

age, over the number of parachutes, the parachute breathing, swinging, and coning, respectively. The symbols u_B and u_S represent the breathing and swinging for the cluster, u_P is the payload velocity, and \bar{u}_A is the aerodynamic contributor to that. Figures 17, 18, 19 and 20 show the individual-parachute contributions to the payload descent speed.

To further analyze the results, Figs. 21, 22, 23 and 24 show the individual-parachute contributions to the drag. Figures 25, 26, 27 and 28 show the payload and canopy-centroid descent speeds.

4 Concluding remarks

In this paper we focused on FSI computation of clusters of ringsail spacecraft parachutes with modified geometric porosity. The modification, intended to increase the aerodynamic performance, consists of increasing the geometric porosity beyond the rings and sails with hundreds of ring

gaps and sail slits by creating windows with removal of panels and wider gaps with removal of sails. This creates computational challenges beyond those created by the lightness of the canopy structure compared to the air masses involved in parachute dynamics, geometric complexities of hundreds of gaps and slits, and the contact between the parachutes of the cluster or intra-canopy contact between the structural surfaces of a parachute. This is because the flow through the windows and wider gaps cannot be accurately modeled with the HMGP and needs to be actually resolved. All these computational challenges needed to be addressed simultaneously in FSI modeling reported in the paper. This is accomplished with the SSTFSI technique, which serves as the core numerical technology, and a number of special FSI techniques. The special techniques include the newest version of the HMGP, which deals with the hundreds of gaps and slits, conservative version of the SENCT technique, which serves as a contact algorithm, and intra-canopy contact algorithms. In addition, we use two special techniques developed recently, one to remove the nonphysical spinning component of the parachute motion, and the other one to restore the mesh integrity lost

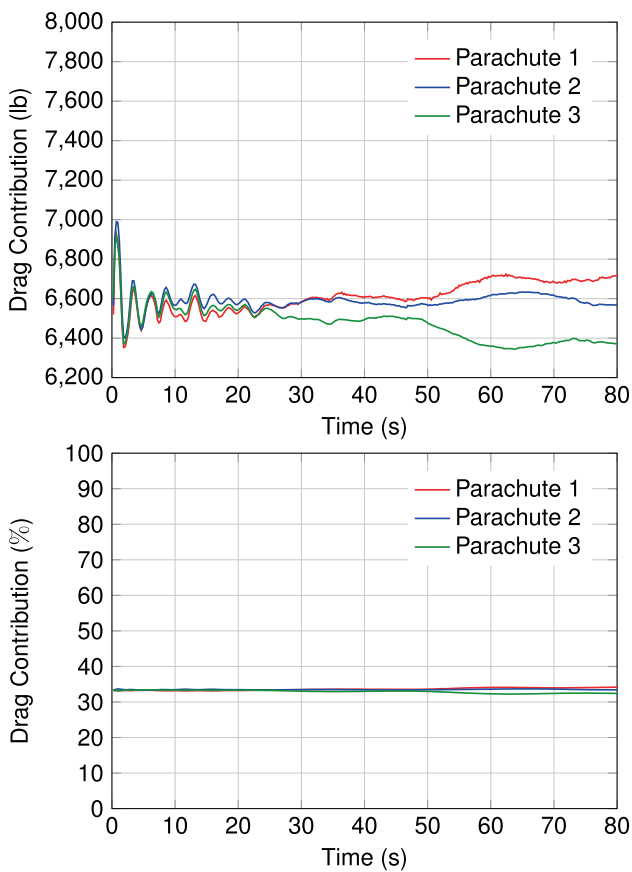


Fig. 24 Individual-parachute contributions to drag for 3-MP with PTE

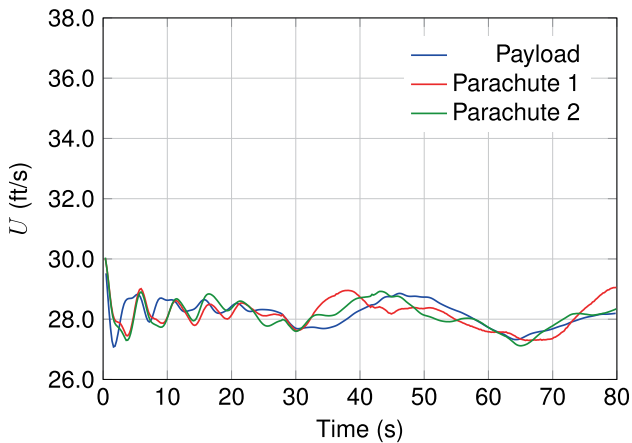


Fig. 25 Payload and canopy-centroid descent speeds for 2-MP with PAC

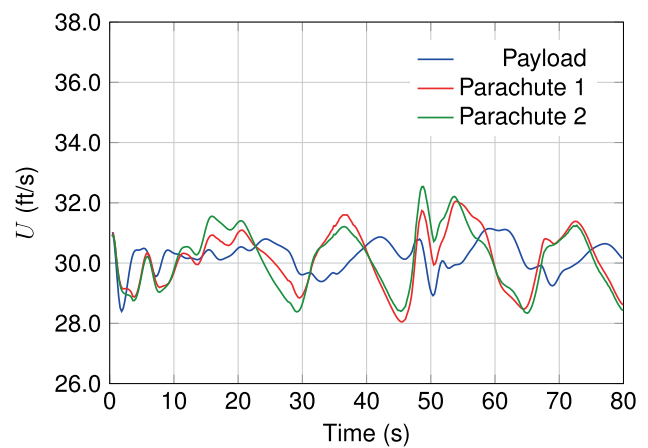


Fig. 26 Payload and canopy-centroid descent speeds for 2-MP with PTE

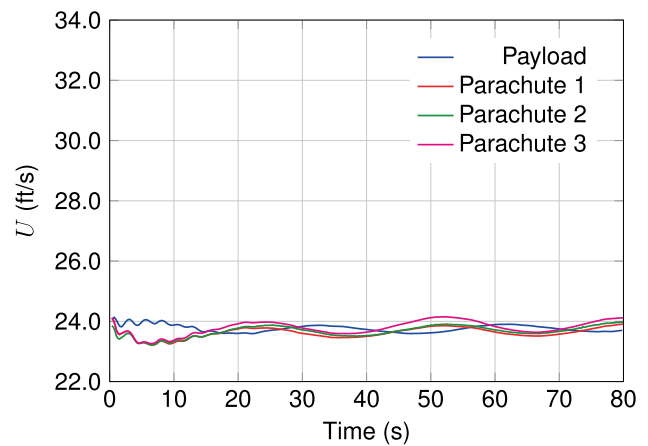


Fig. 27 Payload and canopy-centroid descent speeds for 3-MP with PAC

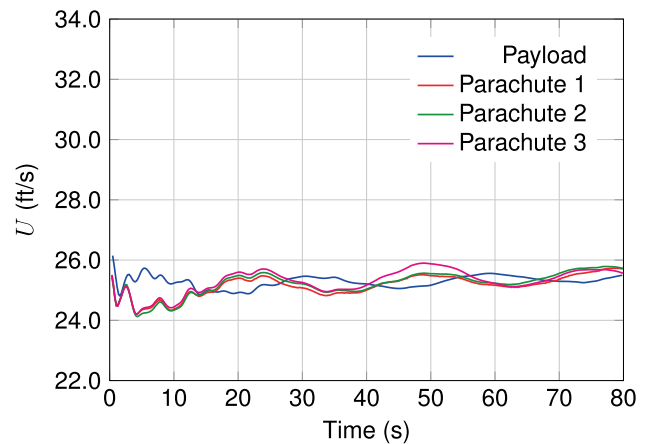


Fig. 28 Payload and canopy-centroid descent speeds for 3-MP with PTE

during the mesh motion, but without resorting to remeshing. We presented results for 2- and 3-parachute clusters with two different payload models.

Acknowledgments This work was supported in part by NASA Johnson Space Center grant NNX13AD87G. It was also supported in part by the Rice–Waseda research agreement (first author).

References

1. Takizawa K, Tezduyar TE (2012) Computational methods for parachute fluid–structure interactions. *Arch Comput Methods Eng* 19:125–169. doi:[10.1007/s11831-012-9070-4](https://doi.org/10.1007/s11831-012-9070-4)
2. Bazilevs Y, Takizawa K, Tezduyar TE (2013) *Computational fluid–structure interaction: methods and applications*. Wiley, New York
3. Tezduyar TE, Sathe S (2007) Modeling of fluid–structure interactions with the space–time finite elements: solution techniques. *Int J Numer Methods Fluids* 54:855–900. doi:[10.1002/fld.1430](https://doi.org/10.1002/fld.1430)
4. Tezduyar TE (1992) Stabilized finite element formulations for incompressible flow computations. *Adv Appl Mech* 28:1–44. doi:[10.1016/S0065-2156\(08\)70153-4](https://doi.org/10.1016/S0065-2156(08)70153-4)
5. Tezduyar TE, Behr M, Liou J (1992) 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* 94:339–351. doi:[10.1016/0045-7825\(92\)90059-S](https://doi.org/10.1016/0045-7825(92)90059-S)
6. Tezduyar TE, Behr M, Mittal S, Liou J (1992) 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* 94:353–371. doi:[10.1016/0045-7825\(92\)90060-W](https://doi.org/10.1016/0045-7825(92)90060-W)
7. Tezduyar TE (2003) Computation of moving boundaries and interfaces and stabilization parameters. *Int J Numer Methods Fluids* 43:555–575. doi:[10.1002/fld.505](https://doi.org/10.1002/fld.505)
8. Takizawa K, Tezduyar TE (2011) Multiscale space–time fluid–structure interaction techniques. *Comput Mech* 48:247–267. doi:[10.1007/s00466-011-0571-z](https://doi.org/10.1007/s00466-011-0571-z)
9. Takizawa K, Tezduyar TE (2012) Space–time fluid–structure interaction methods. *Math Model Methods Appl Sci* 22:1230001. doi:[10.1142/S0218202512300013](https://doi.org/10.1142/S0218202512300013)
10. Brooks AN, Hughes TJR (1982) Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Comput Methods Appl Mech Eng* 32:199–259
11. Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity–pressure elements. *Comput Methods Appl Mech Eng* 95:221–242. doi:[10.1016/0045-7825\(92\)90141-6](https://doi.org/10.1016/0045-7825(92)90141-6)
12. Tezduyar TE, Behr M, Mittal S, Johnson AA (1992) 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, pp 7–24
13. Tezduyar T, Aliabadi S, Behr M, Johnson A, Mittal S (1993) Parallel finite-element computation of 3D flows. *Computer* 26:27–36. doi:[10.1109/2.237441](https://doi.org/10.1109/2.237441)
14. Johnson AA, Tezduyar TE (1994) Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Comput Methods Appl Mech Eng* 119:73–94. doi:[10.1016/0045-7825\(94\)00077-8](https://doi.org/10.1016/0045-7825(94)00077-8)
15. Tezduyar TE (2001) Finite element methods for flow problems with moving boundaries and interfaces. *Arch Comput Methods Eng* 8:83–130. doi:[10.1007/BF02897870](https://doi.org/10.1007/BF02897870)
16. Takizawa K, Henicke B, Puntel A, Spielman T, Tezduyar TE (2012) Space–time computational techniques for the aerodynamics of flapping wings. *J Appl Mech* 79:010903. doi:[10.1115/1.4005073](https://doi.org/10.1115/1.4005073)
17. Takizawa K, Henicke B, Puntel A, Kostov N, Tezduyar TE (2012) Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust. *Comput Mech* 50:743–760. doi:[10.1007/s00466-012-0759-x](https://doi.org/10.1007/s00466-012-0759-x)
18. Takizawa K, Kostov N, Puntel A, Henicke B, Tezduyar TE (2012) Space–time computational analysis of bio-inspired flapping-wing aerodynamics of a micro aerial vehicle. *Comput Mech* 50:761–778. doi:[10.1007/s00466-012-0758-y](https://doi.org/10.1007/s00466-012-0758-y)
19. Takizawa K, Montes D, Fritze M, McIntyre S, Boben J, Tezduyar TE (2013) Methods for FSI modeling of spacecraft parachute dynamics and cover separation. *Math Model Methods Appl Sci* 23:307–338. doi:[10.1142/S0218202513400058](https://doi.org/10.1142/S0218202513400058)
20. Takizawa K, Henicke B, Puntel A, Kostov N, Tezduyar TE (2012) Computer modeling techniques for flapping-wing aerodynamics of a locust. *Comput Fluids*, published online, November 2012. doi:[10.1016/j.compfluid.2012.11.008](https://doi.org/10.1016/j.compfluid.2012.11.008)
21. Hughes TJR, Liu WK, Zimmermann TK (1981) Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Comput Methods Appl Mech Eng* 29:329–349
22. Ohayon R (2001) Reduced symmetric models for modal analysis of internal structural-acoustic and hydroelastic-sloshing systems. *Comput Methods Appl Mech Eng* 190:3009–3019
23. van Brummelen EH, de Borst R (2005) On the nonnormality of subiteration for a fluid–structure interaction problem. *SIAM J Sci Comput* 27:599–621
24. Bazilevs Y, Calo VM, Zhang Y, Hughes TJR (2006) Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Comput Mech* 38:310–322
25. Lohner R, Cebra JR, Yang C, Baum JD, Mestreau EL, Soto O (2006) Extending the range of applicability of the loose coupling approach for FSI simulations. In: Bungartz H-J, Schafer M (eds) *Fluid–structure interaction*, vol 53 of *Lecture notes in computational science and engineering*. Springer, New York, pp 82–100
26. Bletzinger K-U, Wuchner R, Kupzok A (2006) Algorithmic treatment of shells and free form-membranes in FSI. In: Bungartz H-J, Schafer M (eds) *Fluid–structure interaction*, vol 53 of *Lecture notes in computational science and engineering*. Springer, New York, pp 336–355
27. Bazilevs Y, Calo VM, Hughes TJR, Zhang Y (2008) *Isogeometric fluid–structure interaction: theory, algorithms, and computations*. *Comput Mech* 43:3–37
28. Dettmer WG, Peric D (2008) On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Comput Mech* 43:81–90
29. Bazilevs Y, Gohean JR, Hughes TJR, Moser RD, Zhang Y (2009) Patient-specific isogeometric fluid–structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device. *Comput Methods Appl Mech Eng* 198:3534–3550
30. Bazilevs Y, Hsu M-C, Benson D, Sankaran S, Marsden A (2009) *Computational fluid–structure interaction: methods and application to a total cavopulmonary connection*. *Comput Mech* 45:77–89
31. Calderer R, Masud A (2010) A multiscale stabilized ALE formulation for incompressible flows with moving boundaries. *Comput Mech* 46:185–197
32. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Liang X, Kvamsdal T, Brekken R, Isaksen J (2010) A fully-coupled fluid–structure interaction simulation of cerebral aneurysms. *Comput Mech* 46:3–16
33. Bazilevs Y, Hsu M-C, Zhang Y, Wang W, Kvamsdal T, Hentschel S, Isaksen J (2010) *Computational fluid–structure interaction: methods and application to cerebral aneurysms*. *Biomech Model Mechanobiol* 9:481–498
34. Bazilevs Y, Hsu M-C, Akkerman I, Wright S, Takizawa K, Henicke B, Spielman T, Tezduyar TE (2011) 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *Int J Numer Methods Fluids* 65:207–235. doi:[10.1002/fld.2400](https://doi.org/10.1002/fld.2400)

35. Bazilevs Y, Hsu M-C, Kiendl J, Wüchner R, Bletzinger K-U (2011) 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *Int J Numer Methods Fluids* 65:236–253
36. Hsu M-C, Bazilevs Y (2011) Blood vessel tissue prestress modeling for vascular fluid–structure interaction simulations. *Finite Elements Anal Des* 47:593–599
37. Nagaoka S, Nakabayashi Y, Yagawa G, Kim YJ (2011) Accurate fluid–structure interaction computations using elements without mid-side nodes. *Comput Mech* 48:269–276. doi:[10.1007/s00466-011-0620-7](https://doi.org/10.1007/s00466-011-0620-7)
38. Bazilevs Y, Hsu M-C, Takizawa K, Tezduyar TE (2012) ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. *Math Model Methods Appl Sci* 22:1230002. doi:[10.1142/S0218202512300025](https://doi.org/10.1142/S0218202512300025)
39. Tezduyar TE, Takizawa K, Brummer T, Chen PR (2011) Space–time fluid–structure interaction modeling of patient-specific cerebral aneurysms. *Int J Numer Methods Biomed Eng* 27:1665–1710. doi:[10.1002/cnm.1433](https://doi.org/10.1002/cnm.1433)
40. Takizawa K, Bazilevs Y, Tezduyar TE (2012) Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. *Arch Comput Methods Eng* 19:171–225. doi:[10.1007/s11831-012-9071-3](https://doi.org/10.1007/s11831-012-9071-3)
41. Takizawa K, Tezduyar TE (2012) Bringing them down safely. *Mech Eng* 134:34–37
42. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Challenges and directions in computational fluid–structure interaction. *Math Model Methods Appl Sci* 23:215–221. doi:[10.1142/S0218202513400010](https://doi.org/10.1142/S0218202513400010)
43. Stein KR, Benney RJ, Kalro V, Johnson AA, Tezduyar TE (1997) Parallel computation of parachute fluid–structure interactions. Proceedings of AIAA 14th aerodynamic decelerator systems technology conference, AIAA Paper 97–1505, San Francisco, CA
44. Kalro V, Tezduyar TE (2000) A parallel 3D computational method for fluid–structure interactions in parachute systems. *Comput Methods Appl Mech Eng* 190:321–332. doi:[10.1016/S0045-7825\(00\)00204-8](https://doi.org/10.1016/S0045-7825(00)00204-8)
45. Stein K, Benney R, Kalro V, Tezduyar TE, Leonard J, Accorsi M (2000) Parachute fluid–structure interactions: 3-D computation. *Comput Methods Appl Mech Eng* 190:373–386. doi:[10.1016/S0045-7825\(00\)00208-5](https://doi.org/10.1016/S0045-7825(00)00208-5)
46. Tezduyar T, Osawa Y (2001) Fluid–structure interactions of a parachute crossing the far wake of an aircraft. *Comput Methods Appl Mech Eng* 191:717–726. doi:[10.1016/S0045-7825\(01\)00311-5](https://doi.org/10.1016/S0045-7825(01)00311-5)
47. Stein K, Benney R, Tezduyar T, Potvin J (2001) Fluid–structure interactions of a cross parachute: numerical simulation. *Comput Methods Appl Mech Eng* 191:673–687. doi:[10.1016/S0045-7825\(01\)00312-7](https://doi.org/10.1016/S0045-7825(01)00312-7)
48. Stein KR, Benney RJ, Tezduyar TE, Leonard JW, Accorsi ML (2001) Fluid–structure interactions of a round parachute: modeling and simulation techniques. *J Aircr* 38:800–808. doi:[10.2514/2.2864](https://doi.org/10.2514/2.2864)
49. Stein K, Tezduyar T, Kumar V, Sathe S, Benney R, Thornburg E, Kyle C, Nonoshita T (2003) Aerodynamic interactions between parachute canopies. *J Appl Mech* 70:50–57. doi:[10.1115/1.1530634](https://doi.org/10.1115/1.1530634)
50. Stein K, Tezduyar T, Benney R (2003) Computational methods for modeling parachute systems. *Comput Sci Eng* 5:39–46. doi:[10.1109/MCISE.2003.1166551](https://doi.org/10.1109/MCISE.2003.1166551)
51. Tezduyar TE, Sathe S, Keedy R, Stein K (2006) Space–time finite element techniques for computation of fluid–structure interactions. *Comput Methods Appl Mech Eng* 195:2002–2027. doi:[10.1016/j.cma.2004.09.014](https://doi.org/10.1016/j.cma.2004.09.014)
52. Tezduyar TE, Sathe S, Stein K (2006) Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations. *Comput Methods Appl Mech Eng* 195:5743–5753. doi:[10.1016/j.cma.2005.08.023](https://doi.org/10.1016/j.cma.2005.08.023)
53. Tezduyar TE, Sathe S, Pausewang J, Schwaab M, Christopher J, Crabtree J (2008) Interface projection techniques for fluid–structure interaction modeling with moving-mesh methods. *Comput Mech* 43:39–49. doi:[10.1007/s00466-008-0261-7](https://doi.org/10.1007/s00466-008-0261-7)
54. Tezduyar TE, Sathe S, Schwaab M, Pausewang J, Christopher J, Crabtree J (2008) Fluid–structure interaction modeling of ringsail parachutes. *Comput Mech* 43:133–142. doi:[10.1007/s00466-008-0260-8](https://doi.org/10.1007/s00466-008-0260-8)
55. Tezduyar TE, Takizawa K, Moorman C, Wright S, Christopher J (2010) Space–time finite element computation of complex fluid–structure interactions. *Int J Numer Methods Fluids* 64:1201–1218. doi:[10.1002/fld.2221](https://doi.org/10.1002/fld.2221)
56. Takizawa K, Moorman C, Wright S, Spielman T, Tezduyar TE (2011) Fluid–structure interaction modeling and performance analysis of the Orion spacecraft parachutes. *Int J Numer Methods Fluids* 65:271–285. doi:[10.1002/fld.2348](https://doi.org/10.1002/fld.2348)
57. Takizawa K, Wright S, Moorman C, Tezduyar TE (2011) Fluid–structure interaction modeling of parachute clusters. *Int J Numer Methods Fluids* 65:286–307. doi:[10.1002/fld.2359](https://doi.org/10.1002/fld.2359)
58. Takizawa K, Spielman T, Tezduyar TE (2011) Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters. *Comput Mech* 48:345–364. doi:[10.1007/s00466-011-0590-9](https://doi.org/10.1007/s00466-011-0590-9)
59. Takizawa K, Spielman T, Moorman C, Tezduyar TE (2012) Fluid–structure interaction modeling of spacecraft parachutes for simulation-based design. *J Appl Mech* 79:010907. doi:[10.1115/1.4005070](https://doi.org/10.1115/1.4005070)
60. Takizawa K, Fritze M, Montes D, Spielman T, Tezduyar TE (2012) Fluid–structure interaction modeling of ringsail parachutes with disreefing and modified geometric porosity. *Comput Mech* 50:835–854. doi:[10.1007/s00466-012-0761-3](https://doi.org/10.1007/s00466-012-0761-3)
61. Tezduyar TE (2004) Finite element methods for fluid dynamics with moving boundaries and interfaces. In: Stein E, Borst RD, Hughes TJJ (eds) *Encyclopedia of computational mechanics*, volume 3: fluids, chap. 17. Wiley, New York
62. Tezduyar TE (2007) Finite elements in fluids: special methods and enhanced solution techniques. *Comput Fluids* 36:207–223. doi:[10.1016/j.compfluid.2005.02.010](https://doi.org/10.1016/j.compfluid.2005.02.010)
63. Takizawa K, Moorman C, Wright S, Christopher J, Tezduyar TE (2010) Wall shear stress calculations in space–time finite element computation of arterial fluid–structure interactions. *Comput Mech* 46:31–41. doi:[10.1007/s00466-009-0425-0](https://doi.org/10.1007/s00466-009-0425-0)
64. Moorman CJ (2010) Fluid–structure interaction modeling of the Orion spacecraft parachutes. Master’s thesis, Rice University
65. Karypis G, Kumar V (1998) A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J Sci Comput* 20:359–392
66. Saad Y, Schultz M (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput* 7:856–869