ELSEVIER

Comput. Methods Appl. Mech. Engrg. 191 (2001) 705–716

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The multi-domain method for computation of the aerodynamics of a parachute crossing the far wake of an aircraft

T. Tezduyar *, Y. Osawa [1]

*Team for Advanced Flow Simulation and Modeling (T*AFSM), Mechanical Engineering and Materials Science, Rice University – MS 321, 6100 Main Street, Houston, TX 77005, USA*

Received 25 September 2000; received in revised form 19 January 2001

Abstract

We present the multi-domain method (MDM) for computation of unsteady flow past a cargo aircraft and around a parachute crossing the aircraft's far wake. The base computational methods used here are the stabilized semi-discrete and space-time finite element formulations developed earlier. In the MDM, the computational domain is divided into an ordered sequence of overlapping subdomains. The flow field computed over Subdomain-1, which contains the aircraft, supplies the inflow boundary conditions for Subdomain-2, which is used for computing the long-wake flow. Subdomain-3 contains the parachute, and moves across Subdomain-2. The boundary conditions for Subdomain-3 are extracted from the flow field computed over Subdomain-2, at locations corresponding to the positions of the boundaries of Subdomain-3 as it crosses Subdomain-2. The computation over Subdomain-1, which contains a complex but fixed object, is based on a general-purpose implementation of the semi-discrete formulation. The computation over Subdomain-2, which contains no objects, is based on a special-purpose implementation that exploits the simplicity of the mesh to increase the computational speed. The computation over Subdomain-3, which contains a complex and moving object, is based on a general-purpose implementation of the space-time formulation. With a numerical example, we show that different methods can be brought together in the context of the MDM to address the computational challenges involved in the aerodynamics of a parachute crossing the far wake of an aircraft.

1. Introduction

The multi-domain method (MDM) [1] was introduced for computation of a special class of problems, where the objective is to predict the long-wake flow generated by a primary object and, in some cases, also is to determine the influence of this wake flow on a secondary object placed far downstream. This class of problems poses a significant computational challenge because the length of the wake region is rather large compared to the length scales of the objects involved, and therefore unsteady flows need to be computed accurately over long-wake regions.

Applications of the MDM to a number of 3D problems were reported in [2–4], including flow around a small wing placed in the wake of a larger wing, and flow in the wake of a circular cylinder up to 300 diameters downstream. In the case of the cylinder problem, at Reynolds number 140, we were able to show that with the MDM we can extend our computations sufficiently downstream, and with sufficient accuracy, to successfully capture the second phase of the Karman vortex street, which has been observed in

* Corresponding author. Tel.: +1-713-348-6051; fax: +1-713-348-5423.
E-mail address: tezduyar@rice.edu http://www.mems.rice.edu/TAFSM/ (T. Tezduyar).
[1] Present address: Tire Research Department, Bridgestone Corporation, 3-1-1 Ogawahigashi-cho, Kodaira-shi, Tokyo 187, Japan.

0045-7825/01/$ - see front matter
PII: S0045-7825(01)00310-3

laboratory experiments, and which has double the spacing between the vortices compared to the first phase (see [1,3,4]).

In the MDM, the problem domain is subdivided into a sequence of overlapping subdomains. The primary object is placed in Subdomain-1. The subsequent subdomains are used for computing the long-wake flows and flow past secondary objects. The inflow boundary conditions for Subdomain-1 are the free-stream conditions. The inflow conditions for each of the remaining subdomains is extracted from the subdomain preceding it.

In this paper we present the MDM for computation of unsteady flow past a cargo aircraft, in the long-wake region behind the aircraft and around a parachute crossing the aircraft's far wake. Early results from this computation were first reported in [1]. Subdomain-1 contains the aircraft. The long-wake flow is computed over Subdomain-2. Subdomain-3 contains the parachute, and moves across Subdomain-2. In this particular case, the boundary conditions for Subdomain-3 are extracted from the flow field computed over Subdomain-2, at locations corresponding to the positions of the boundaries of Subdomain-3 as it crosses Subdomain-2. In our application here, we focus on the aerodynamics of a rigid parachute, and therefore do not take into account the fluid–structure interactions. However, the formulation we describe in this paper can be applied to cases where the parachute undergoes prescribed shape changes, and is free to undergo rigid-body motions that can be determined as part of the overall solution.

Depending on the nature of the flow problem in each subdomain, we use as base methods the stabilized semi-discrete and space-time finite element formulations developed earlier. The semi-discrete formulation is based on streamline-upwind/Petrov–Galerkin (SUPG) [5,6] and pressure-stabilizing/Petrov–Galerkin (PSPG) [7] stabilizations. The space-time formulation is the deforming-spatial-domain/stabilized space-time (DSD/SST) formulation [7–9]. These stabilized formulations remain stable in computation of flows with high Reynolds numbers and boundary layers, without introducing excessive numerical dissipation. They also allow us to use, without facing numerical stability problems, equal-order interpolation functions for velocity and pressure.

The class of problems we want to solve result in large, coupled nonlinear equation systems that need to be solved at every time step. We solve these equations with the Newton–Raphson method. At each step of the Newton–Raphson sequence, we are faced with solving a large, coupled linear equation system. We solve this linear system also iteratively, with the GMRES search technique [10]. We have implemented these solution techniques for distributed-memory parallel computing, and the results reported here were obtained by carrying out the computations on a CRAY T3E-1200.

In Section 2, we review the governing equations. The semi-discrete stabilized formulation and the DSD/SST formulation are described in Section 3. Further details and remarks on the MDM are given in Section 4. The numerical example is presented in Section 5, and the concluding remarks are provided in Section 6.

2. Governing equations

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

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

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

where ρ, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, respectively. The stress tensor $\boldsymbol{\sigma}$ is defined as

$$\boldsymbol{\sigma}(p,\mathbf{u}) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}). \tag{3}$$

Here p, $\mathbf{I}$ and μ are the pressure, identity tensor and the viscosity, respectively. The strain rate tensor $\boldsymbol{\varepsilon}(\mathbf{u})$ is defined as

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}\Big((\nabla\mathbf{u}) + (\nabla\mathbf{u})^{\mathrm{T}}\Big). \tag{4}$$

Both Dirichlet- and Neumann-type boundary conditions are accounted for:

$$\begin{aligned} &\mathbf{u} = \mathbf{g} \quad \text{on } (\Gamma_t)_g, \\ &\mathbf{n}\cdot\boldsymbol{\sigma} = \mathbf{h} \quad \text{on } (\Gamma_t)_h. \end{aligned} \tag{5}$$

Here $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t, $\mathbf{n}$ is the unit normal vector at the boundary, and $\mathbf{g}$ and $\mathbf{h}$ are given functions. A divergence-free velocity field is specified as the initial condition.

3. Finite element formulations for incompressible flows

3.1. Semi-discrete stabilized formulation

Let us consider a fixed spatial domain Ω and its boundary Γ, where subscript t is dropped from both Ω_t and Γ_t. The domain Ω is discretized into sub-domains Ω^e, $e = 1, 2, \ldots, n_{\mathrm{el}}$, where n_{el} is the number of elements. For this discretization, the finite element trial function spaces $\mathscr{S}_u^h$ for velocity and $\mathscr{S}_p^h$ for pressure, and the corresponding test function spaces $\mathscr{V}_u^h$ and $\mathscr{V}_p^h$ are defined as follows:

$$\mathscr{S}_u^h = \big\{\mathbf{u}^h \,|\, \mathbf{u}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{sd}}},\ \mathbf{u}^h \doteq \mathbf{g}^h \text{ on } \Gamma_g\big\}, \tag{6}$$

$$\mathscr{V}_u^h = \big\{\mathbf{w}^h \,|\, \mathbf{w}^h \in [H^{1h}(\Omega)]^{n_{\mathrm{sd}}},\ \mathbf{w}^h \doteq \mathbf{0} \text{ on } \Gamma_g\big\}, \tag{7}$$

$$\mathscr{S}_p^h = \mathscr{V}_p^h = \big\{q^h \,|\, q^h \in H^{1h}(\Omega)\big\}. \tag{8}$$

Here $H^{1h}(\Omega)$ is the finite-dimensional function space over Ω. The stabilized finite element formulation is written as follows: find $\mathbf{u}^h \in \mathscr{S}_{\mathbf{u}}^h$ and $p^h \in \mathscr{S}_p^h$ such that $\forall \mathbf{w}^h \in \mathscr{V}_{\mathbf{u}}^h$ and $q^h \in \mathscr{V}_p^h$:

$$\begin{aligned} &\int_\Omega \mathbf{w}^h \cdot \rho\left(\frac{\partial\mathbf{u}^h}{\partial t} + \mathbf{u}^h\cdot\nabla\mathbf{u}^h - \mathbf{f}^h\right)\mathrm{d}\Omega + \int_\Omega \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h,\mathbf{u}^h)\mathrm{d}\Omega - \int_{\Gamma_h} \mathbf{w}^h\cdot\mathbf{h}^h\mathrm{d}\Gamma + \int_\Omega q^h\nabla\cdot\mathbf{u}^h\mathrm{d}\Omega \\ &\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \frac{1}{\rho}\left[\tau_{\mathrm{SUPG}}\rho\mathbf{u}^h\cdot\nabla\mathbf{w}^h + \tau_{\mathrm{PSPG}}\nabla q^h\right]\cdot\left[\rho\left(\frac{\partial\mathbf{u}^h}{\partial t} + \mathbf{u}^h\cdot\nabla\mathbf{u}^h\right) - \nabla\cdot\boldsymbol{\sigma}(p^h,\mathbf{u}^h) - \rho\mathbf{f}^h\right]\mathrm{d}\Omega \\ &\quad + \sum_{e=1}^{n_{\mathrm{el}}} \int_{\Omega^e} \tau_{\mathrm{LSIC}}\nabla\cdot\mathbf{w}^h\rho\nabla\cdot\mathbf{u}^h\mathrm{d}\Omega = 0. \end{aligned} \tag{9}$$

In this formulation, τ_{SUPG}, τ_{PSPG} and τ_{LSIC} are the stabilization parameters (see [1]). For an earlier, detailed reference on this stabilized formulation see [7].

3.2. Deforming-spatial-domain/stabilized space-time formulation

In discretization of the space-time domain, the time interval $(0, T)$ is partitioned into subintervals $I_n = (t_n, t_{n+1})$, where t_n and t_{n+1} belong to an ordered series of time levels $0 = t_0 < t_1 \cdots < t_N = T$. Let $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$ to simplify the notation. The space-time slab Q_n is defined as the domain enclosed by the surfaces Ω_n, Ω_{n+1}, and P_n, where P_n is the lateral surface of Q_n described by the boundary Γ_n as t traverses I_n. The Dirichlet- and Neumann-type boundary conditions are specified over $(P_n)_g$ and $(P_n)_h$. For this discretization, the finite element trial function spaces $(\mathscr{S}_{\mathbf{u}}^h)_n$ for velocity and $(\mathscr{S}_p^h)_n$ for pressure, and the corresponding test function spaces $(\mathscr{V}_{\mathbf{u}}^h)_n$ and $(\mathscr{V}_p^h)_n$ are defined as follows:

$$(\mathscr{S}_{\mathbf{u}}^h)_n = \Big\{\mathbf{u}^h \,|\, \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{\mathrm{sd}}},\ \mathbf{u}^h \doteq \mathbf{g}^h \text{ on } (P_n)_g\Big\}, \tag{10}$$

$$(\mathscr{V}_{\mathbf{u}}^h)_n = \Big\{\mathbf{w}^h \,|\, \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{\mathrm{sd}}},\ \mathbf{w}^h \doteq \mathbf{0} \text{ on } (P_n)_g\Big\}, \tag{11}$$

$$(\mathscr{S}_p^h)_n = (\mathscr{V}_p^h)_n = \big\{q^h \,|\, q^h \in H^{1h}(Q_n)\big\}. \tag{12}$$

Here $H^{1h}(Q_n)$ is the finite-dimensional function space over the space-time slab Q_n. Over the element domain, this space is formed by using the first-order polynomials in both space and time. The interpolation functions are continuous in space but discontinuous in time.

The stabilized finite element formulation is written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in (\mathscr{S}_{\mathbf{u}}^h)_n$ and $p^h \in (\mathscr{S}_p^h)_n$ such that $\forall \mathbf{w}^h \in (\mathscr{V}_{\mathbf{u}}^h)_n$ and $q^h \in (\mathscr{V}_p^h)_n$:

$$\int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{f}^h\right) \mathrm{d}Q + \int_{Q_n} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(p^h, \mathbf{u}^h) \mathrm{d}Q - \int_{(P_n)_h} \mathbf{w}^h \cdot \mathbf{h}^h \, \mathrm{d}P + \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q$$
$$+ \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho\big((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-\big) \mathrm{d}\Omega + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \frac{\tau_{\mathrm{LSME}}}{\rho} \left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h\right) - \nabla \cdot \boldsymbol{\sigma}(q^h, \mathbf{w}^h)\right]$$
$$\cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h\right) - \nabla \cdot \boldsymbol{\sigma}(p^h, \mathbf{u}^h) - \rho \mathbf{f}^h\right] \mathrm{d}Q + \sum_{e=1}^{n_{\mathrm{el}}} \int_{Q_n^e} \tau_{\mathrm{LSIC}} \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q = 0. \tag{13}$$

This formulation is sequentially applied to all of the space-time slabs $Q_0, Q_1, Q_2, \ldots, Q_{N-1}$. The computation starts with

$$(\mathbf{u}^h)_0^- = \mathbf{u}_0, \quad \nabla \cdot \mathbf{u}_0 = 0 \text{ on } \Omega_0. \tag{14}$$

Here τ_{LSME} and τ_{LSIC} are the stabilization parameters (see [1,11]). For an earlier, detailed reference on this stabilized formulation see [7].

4. The multi-domain method

In the MDM, in addition to dividing the problem into computationally more manageable pieces, we can use different numerical methods for different subdomains. With this approach, the method used for a subdomain is the most effective one for the flow conditions of that subdomain. Furthermore, in the MDM, in addition to using parallel computation for each subdomain where clusters of elements in that subdomain are assigned to different processors, we find a second level of parallelism. This second parallelism is based on recognizing that the computation for each subdomain needs to lag only one time step behind the subdomain preceding it. In fact different subdomains can be assigned to different computers, or can even be assigned to computers at different geographical locations. This is because only 2D data needs to be transferred between the subdomains, and this does not result in as much of a communication burden as transferring 3D data.

Subdomain-1 contains a complex but fixed object, and consequently does not involve any changes in the spatial domain occupied by the fluid. Therefore, for Subdomain-1 we use a general-purpose implementation of the semi-discrete formulation. In the general-purpose implementation, we assume that the elements of the tetrahedral finite element mesh can be of any shape. Subdomain-2 does not involve any objects, and therefore is handled with a special-purpose implementation of the semi-discrete formulation, where the mesh is assumed to be made of box-shaped hexahedral elements. The special-purpose implementation exploits the simplicity of the mesh to increase the computational speed (see [2]). Furthermore, to capture the details of the wake behavior more accurately, the method used in Subdomain-2 is supplemented with the enhanced-discretization interface-capturing technique (EDICT).

The EDICT was first introduced in [12] for the purpose of increasing the accuracy in representing the interface in computation of a two-fluid flow problem with an interface-capturing method. To enhance the discretization near the interfaces, the finite element functions are defined as functions with multiple components, with each component coming from a different level of finite element mesh. Mesh-1 is the base mesh, and a Mesh-2 is generated by constructing a second-level mesh over a subset of the elements in Mesh-1. This subset of elements in Mesh-1 changes periodically during the computation, and as the interface moves, it is enveloped in regions covered by this subset of elements. For further details on the EDICT, see [13]. The EDICT was later extended in [14] to computation of compressible flows with shocks. This extension is based on re-defining the "interface" to mean the shock front, and at and near the shock fronts

enhanced discretization is used to increase the accuracy in representing those shocks. In the extension of the EDICT to computation of vortex flows, the definition of the interface is extended to mean regions where the vorticity magnitude is larger than a specified value. This version of the EDICT gives us the capability to compute the long-wake flows more accurately, without making the computations too costly.

Subdomain-3 contains a complex object that can move and change its shape. Therefore, the computation over Subdomain-3 is based on a general-purpose implementation of the DSD/SST formulation and a tetrahedral mesh.

For computations over Subdomain-1 and Subdomain-3, we use a Smagorinsky turbulence model [15] with its constant $C = 0.15$, and wall damping based on what has been proposed by Van Driest [16].

5. Numerical example

5.1. Subdomain-1: Unsteady flow past the cargo aircraft

The aircraft is assumed to be traveling at 200 ft/s, with an angle of attack of 10°. We assume symmetry with respect to the plane passing through the middle of the aircraft. The original aircraft model was introduced in [17]. Here the model has been improved by adding the wing flaps and winglets. The Reynolds number based on the aircraft length and flight speed is set to 2×10^9, which is about nine times larger than the typical value corresponding to the expected flight conditions. In nondimensional numbers, the aircraft length is taken as 8.8 units, and the flight speed as 1 unit. The upstream, downstream, lower crossflow, upper crossflow, and side boundaries are located, respectively, at 3.2, 6.8, 5.0, 3.0, and 2.7 unit lengths from the nearest aircraft surface. The boundary conditions are uniform inflow velocity, no-slip on the aircraft surface, zero normal velocity and zero shear stress at the crossflow and side boundaries, and traction-free condition at the outflow boundary. An automatic mesh generation software [18] has been used to produce the tetrahedral mesh that has 599,101 nodes and 3,489,881 elements (see Fig. 1). The vertical plane where the inflow conditions for Subdomain-2 are extracted from is placed at the end of the horizontal stabilizer. The computation is carried out with a time step size of 0.05. Fig. 2 shows the air pressure distribution on the aircraft surface, streamlines around the left wing, and isosurfaces corresponding to the 6.0 value of the vorticity magnitude.

5.2. Subdomain-2: Unsteady long-wake flow

We assume symmetry in the same way we did for Subdomain-1. This subdomain extends 33.7, 7.00 and 10.0 length units, respectively, in the streamwise, horizontal and vertical cross flow directions. The inflow conditions are extracted from the flow field computed over Subdomain-1, as described before. The other boundary conditions are zero normal velocity and zero shear stress at the crossflow and side boundaries, and traction-free condition at the outflow boundary. Mesh-1, shown in Fig. 3, consists of 602,112 nodes

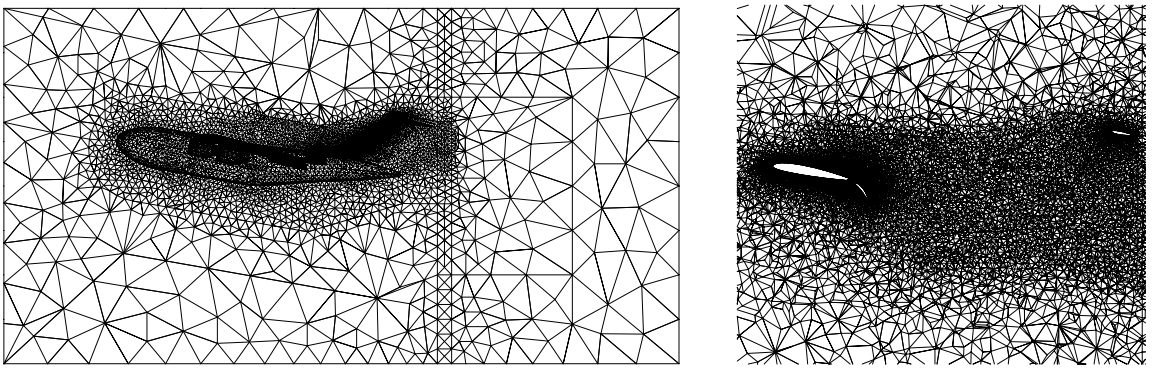

Fig. 1. Mesh for Subdomain-1.

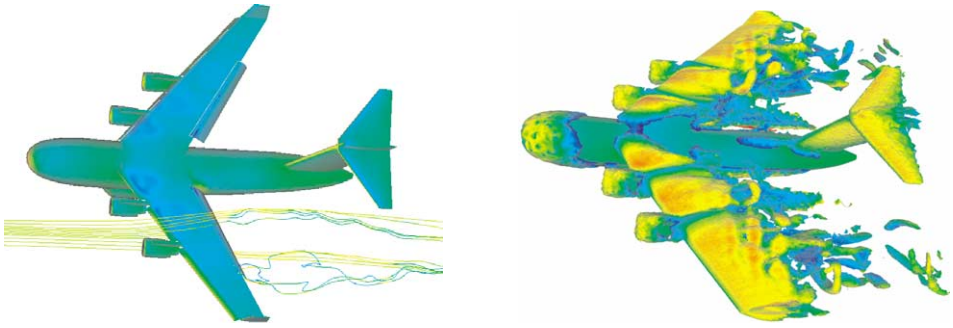

Fig. 2. Flow in Subdomain-1. Left: streamlines around the left wing. Right: isosurfaces corresponding to the 6.0 value of the vorticity magnitude. In both pictures, the colors on the aircraft surface show the pressure distribution.

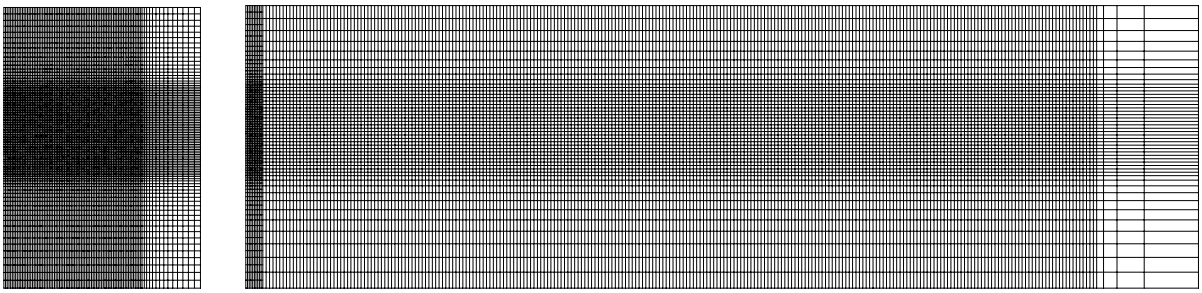

Fig. 3. Mesh-1 for Subdomain-2. Front and side views.

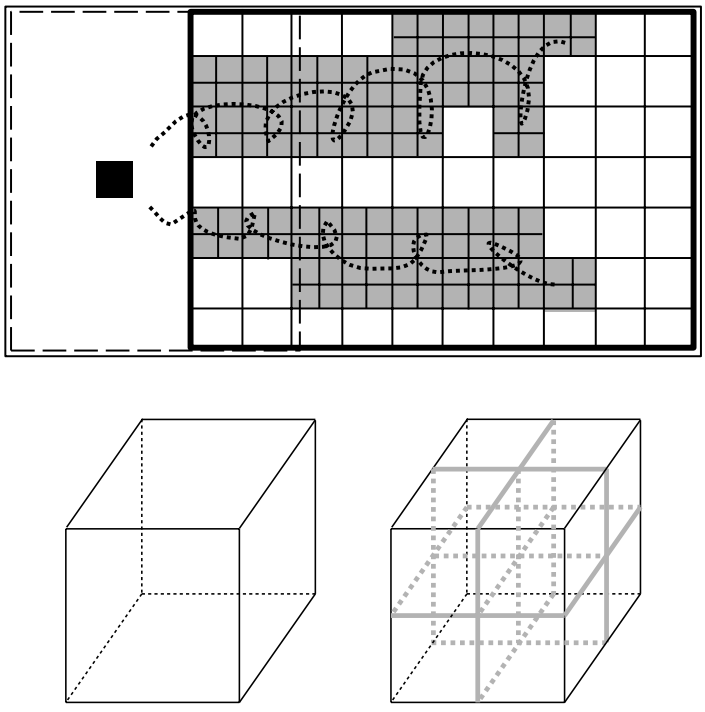

Fig. 4. EDICT applied to computation of wake flows. The top picture shows the EDICT concept for wake flows, and the lower pictures show an element from Mesh-1 (left) and a cluster of elements from Mesh-2 (right).

and 575,280 hexahedral elements. The top picture in Fig. 4 shows the EDICT concept for computation of wake flows. The lower pictures show an element from Mesh-1 and a cluster of elements from Mesh-2. Mesh-2 consists of two parts: a fixed part that does not change during the computation and a dynamic part that is updated as the computation proceeds. The fixed part extends from the upstream boundary to 0.6 length units downstream, and is made of 87,253 nodes and 90,240 hexahedral elements. Updating the dynamic part of the Mesh-2 is based on defining as interface the regions where the vorticity magnitude is

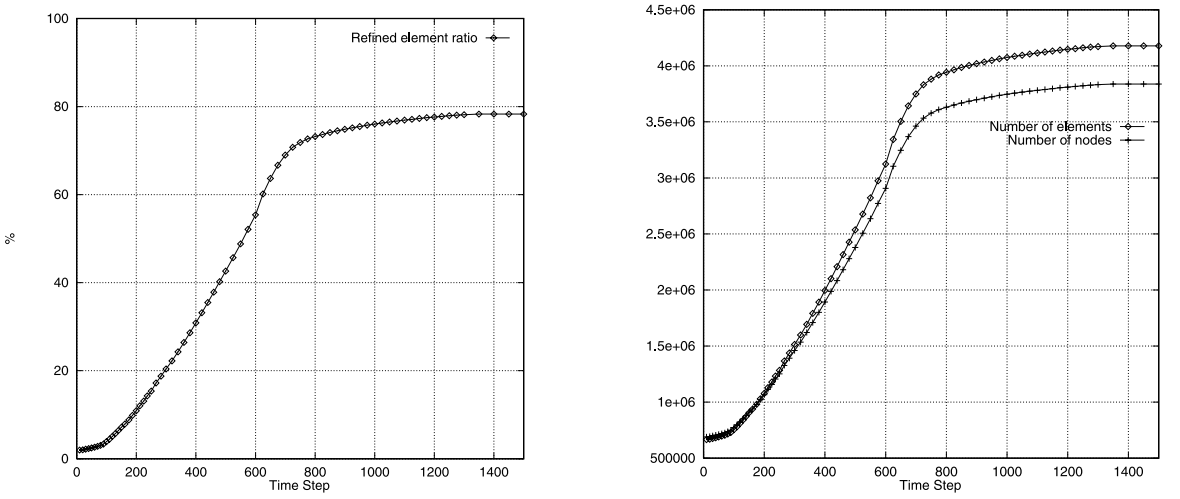

Fig. 5. EDICT for Subdomain-2. Time history of the enhanced discretization. Left: the ratio of the number of enhanced elements in Mesh-1 to the number of all elements in Mesh-1. Right: total number of elements in Mesh-1 plus Mesh-2.

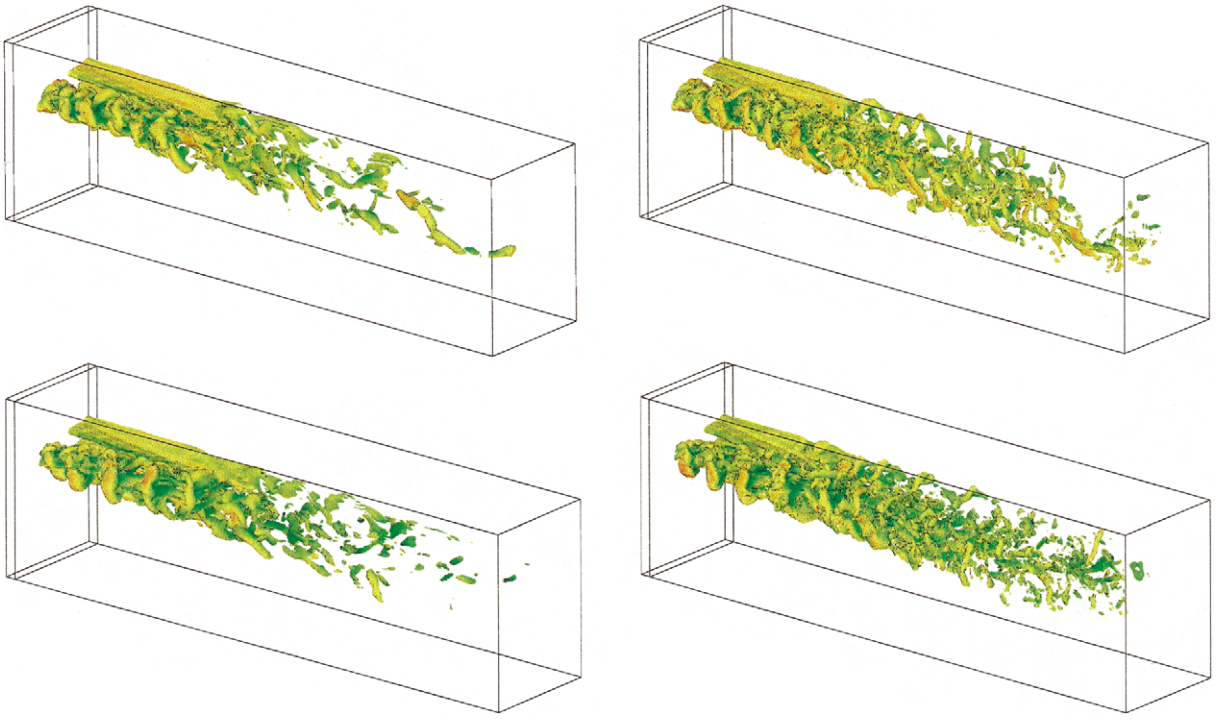

Fig. 6. Flow in Subdomain-2. Isosurfaces corresponding to the 1.0 value of the vorticity magnitude. Left: obtained with Mesh-1 plus the fixed part of Mesh-2. Right: obtained with Mesh-1 plus both the fixed and dynamic parts of Mesh-2. Shown at at time step 700 (upper) and 1500 (lower).

larger than 0.1. Although in this numerical example we use only one subdomain for computation of the wake flow, for more realistic modeling of the actual conditions, longer wake regions and more subdomains would be needed. Keeping this in mind, we designed Mesh-1 in such a way that the inflow conditions for a subsequent subdomain covering the wake flow would be extracted from a vertical plane positioned at 3.84 length units forward of the downstream boundary of Subdomain-2. The mesh between that vertical plane and the downstream boundary is made of stretched, coarser elements. For this computation, the time step size is 0.05. Fig. 5 shows the time history of the enhanced discretization in terms of the number of elements.

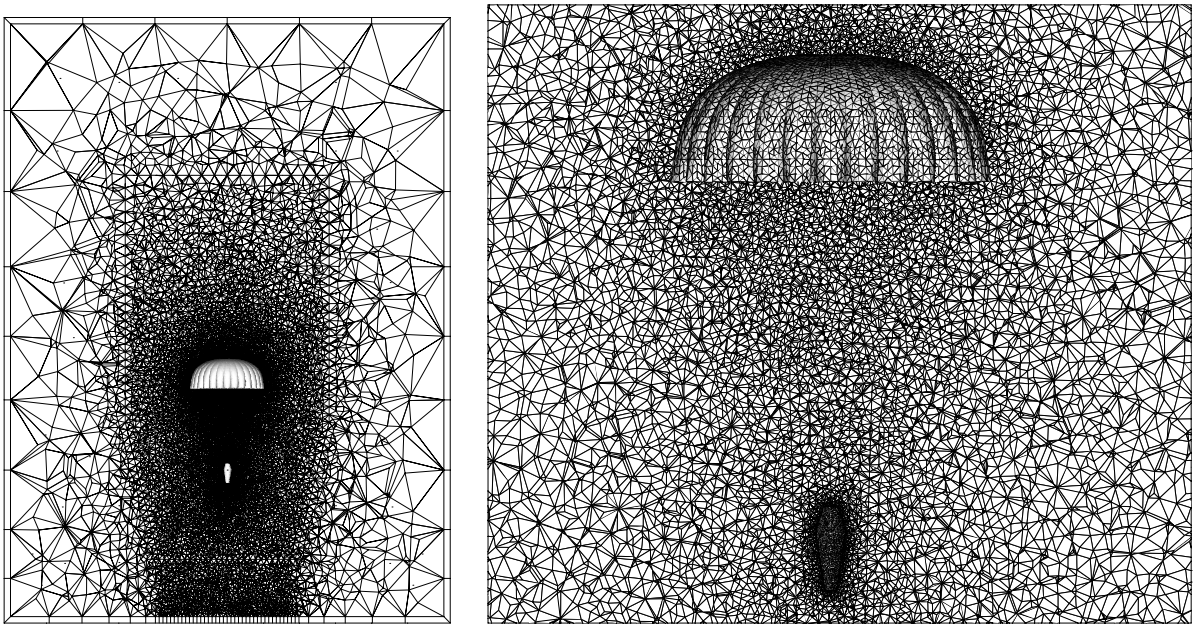

Fig. 7. Mesh for Subdomain-3.

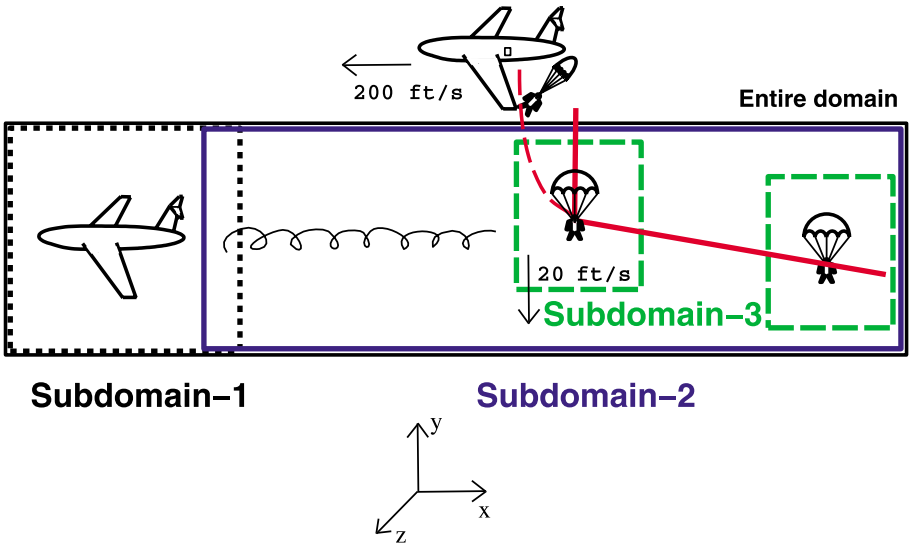

Fig. 8. Parachute trajectory in a coordinate frame attached to the aircraft. The broken line starting from the aircraft shows the assumed trajectory in the short, initial period prior to the parachute attaining zero horizontal velocity and 20 ft/s decent velocity. The oblique line following that shows the trajectory after this initial period. The short, vertical line shows the trajectory used in the computations corresponding to the short, initial period.

In Fig. 6, we compare, at time steps 700 and 1500, the solutions obtained with Mesh-1 plus the fixed part of Mesh-2 and Mesh-1 plus both the fixed and dynamic parts of Mesh-2. The comparison is made in terms of the isosurfaces of the vorticity magnitude. We note that the solution obtained with the inclusion of the dynamic part of Mesh-2 captures more details of the wake flow at the downstream regions.

5.3. Subdomain-3: Unsteady flow around the parachute

The parachute is modeled after a US Army C-9 personnel parachute, where the canopy diameter is approximately 1.2 (see Fig. 7). The Reynolds number based on the parachute diameter and the descent velocity of 20 ft/s is 2.33×10^6. We also include a barrel-shaped payload in our model. This payload is positioned relative to the parachute in such a way that the suspension lines join at the top of the payload.

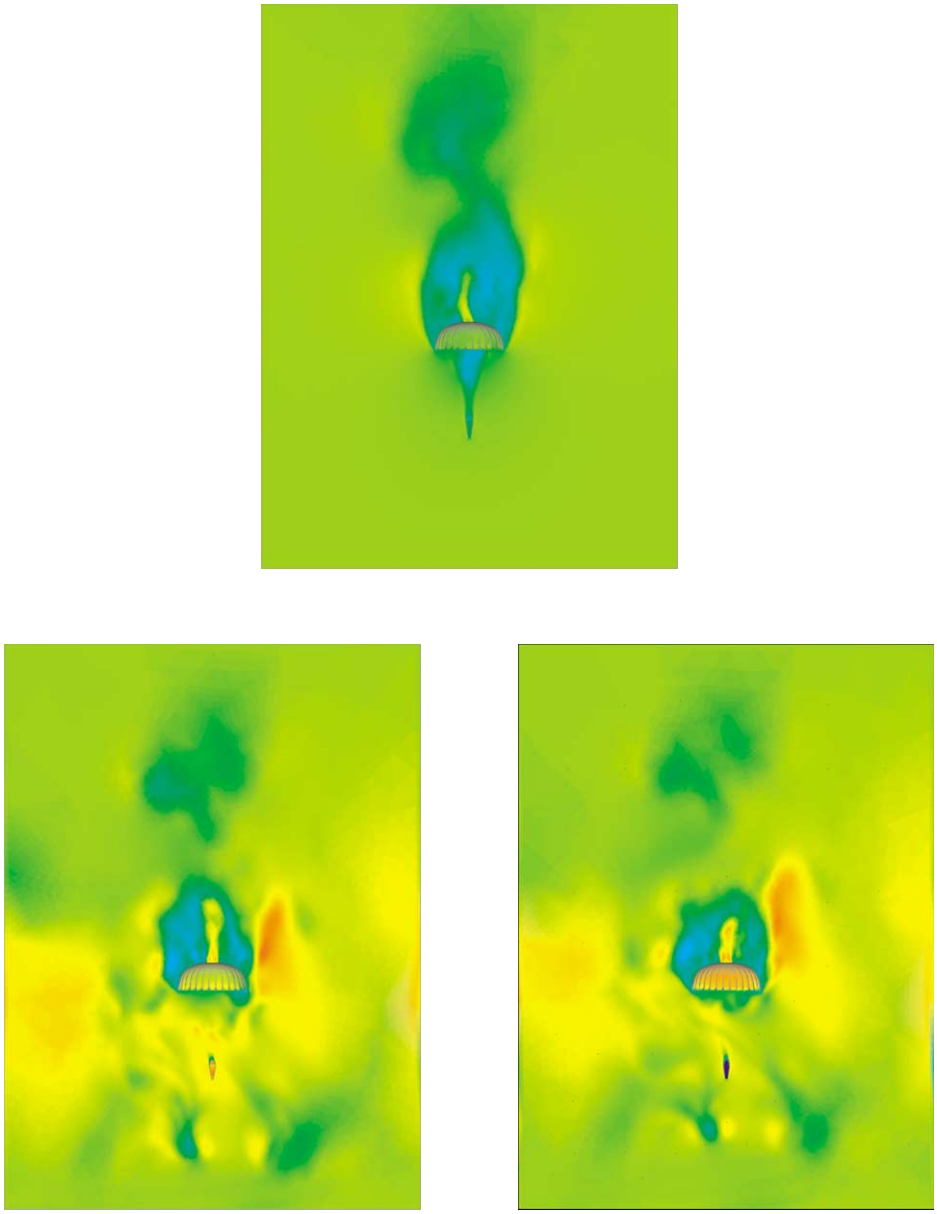

Fig. 9. Flow in Subdomain-3. Vertical component of the velocity in the centered vertical plane, and the pressure distribution on the parachute and payload surfaces. Parachute in uniform flow (upper), and in the wake at $t = 0.4$ (lower left) and $t = 0.8$ (lower right).

The dimensions of Subdomain-3 are 9.5 in height and 7×7 in the horizontal plane. The mesh is made of 212,177 nodes, 1,286,063 tetrahedral elements in the interior, and 10,089 prismatic elements in the one-element-thick layer surrounding the subdomain (see Fig. 7).

Subdomain-1 and Subdomain-2 computations were carried out in a coordinate frame attached to the aircraft, and we take this into account when we extract the boundary conditions for Subdomain-3. We assume the aircraft that deployed the parachute is flying at the same speed as the aircraft that generated the wake: 200 ft/s. We also assume that at the end of a short, initial period following the deployment of the parachute, its horizontal velocity becomes zero in a coordinate frame attached to the ground, and it attains a descent velocity of 20 ft/s.

Based on these conditions, we can approximate the trajectory of the parachute as a parabolic curve (corresponding to the short, initial period), and an oblique line with a tangent of 0.1 (corresponding to the period when the parachute has zero horizontal velocity and a descent velocity of 20 ft/s) (see Fig. 8). In this paper, we focus on the trajectory phase represented by the oblique line. Furthermore, we pick the case where the lateral position relative to the aircraft that the parachute crosses the wake is close to the midpoint between the wingtip and the outer engine.

The boundary conditions for Subdomain-3 consist of inflow conditions at the bottom and side boundaries, outflow conditions at the top boundary, and no-slip on the parachute and payload surfaces. The inflow conditions are extracted from the solution obtained in computations over Subdomain-2. The stress conditions at the outflow boundary are computed from the flow data extracted from Subdomain-2. Prior to using the velocity boundary conditions extracted from Subdomain-2, we modify the horizontal and vertical velocity components by amounts calculated from taking into account the parachute trajectory described above. In describing the results here, $t = 0$ marks the beginning of the trajectory phase represented by the oblique line in Fig. 8. In this computation the time step size is 0.005.

Fig. 9 shows the vertical component of the velocity for a parachute descending through a wake-free air (i.e., the parachute is in a uniform vertical flow) and the parachute crossing the wake. These results cor-

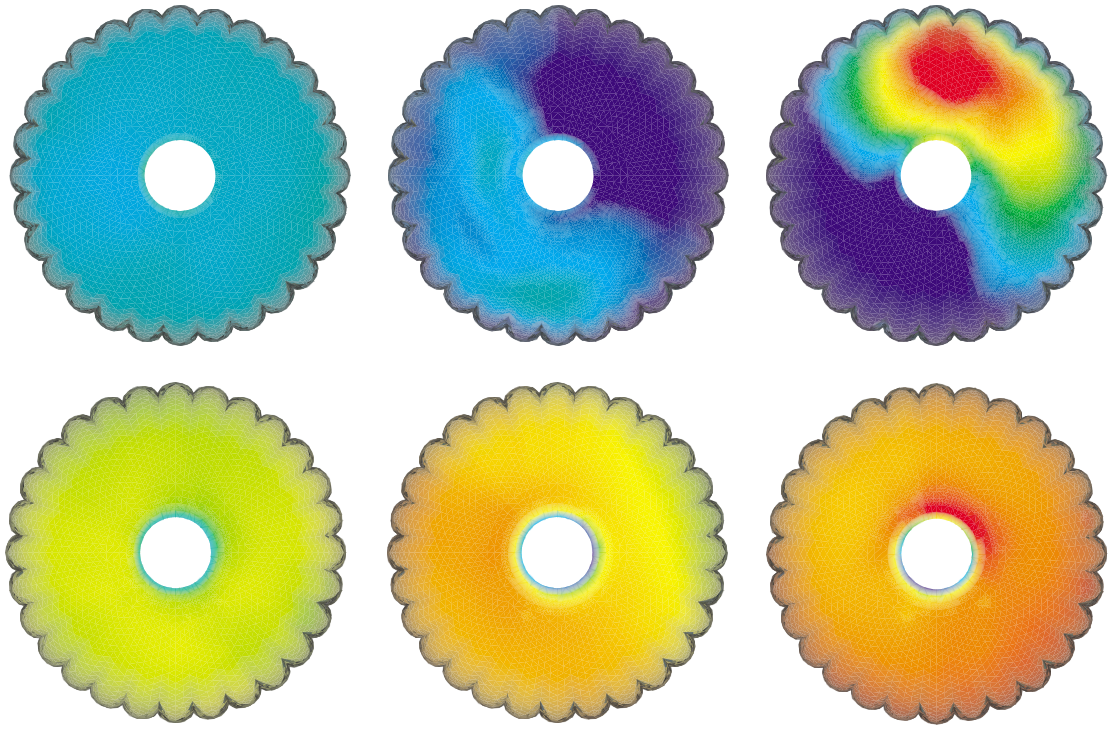

Fig. 10. Flow in Subdomain-3. Pressure distribution on the parachute surface. Parachute in uniform flow (left), and in the wake at $t = 0.4$ (middle) and $t = 0.8$ (right). The upper and lower parachute surfaces are shown, respectively, in the upper and lower rows of pictures.

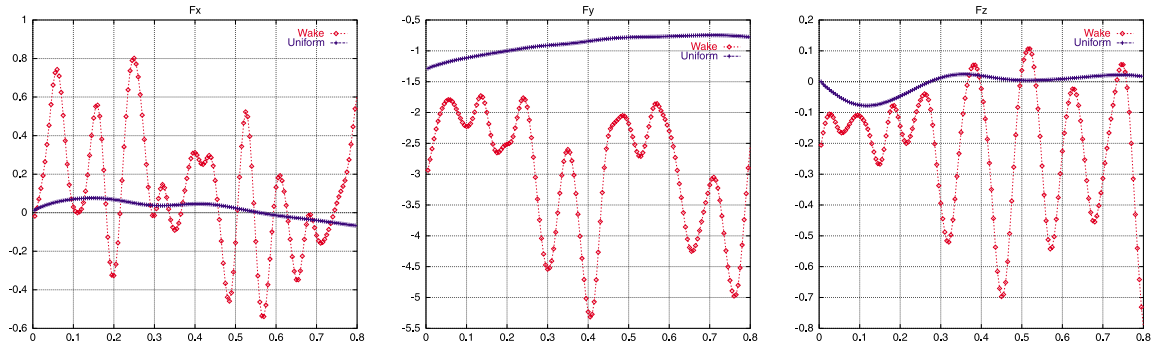

Fig. 11. Flow in Subdomain-3. Time history of the aerodynamical forces acting on the parachute for the one in uniform flow and the one in the wake. Graphs show the forces in the aircraft's flight direction (left), the vertical drag (middle), and the force in lateral direction (right).

respond to the centered vertical plane, and are displayed at two different instants for the parachute in the wake. Fig. 10 shows the pressure distribution on the upper and lower surfaces of the parachute for the one in uniform flow and the one crossing the wake. The results for the parachute crossing the wake are displayed at the same time steps as those in Fig. 9. We note that, at $t = 0.8$, for some parts of the parachute, the pressure on the upper surface is larger than the pressure on the lower surface. Fig. 11 shows the aerodynamical forces acting on the parachute for the one in uniform flow and the one in the wake.

6. Concluding remarks

We described the MDM for computation of the unsteady flow past an aircraft and in its long-wake and around a parachute crossing the far wake. The computational domain is divided into three overlapping subdomains. Subdomain-1 contains the aircraft. Subdomain-2 is used for computing the long-wake flows. Subdomain-3 contains the parachute and moves across Subdomain-2. The inflow boundary conditions for Subdomain-1 are the free-stream conditions, whereas the inflow conditions for Subdomain-2 are extracted from Subdomain-1. The boundary conditions for Subdomain-3 are extracted from Subdomain-2. Subdomain-1 contains a fixed object, and therefore does not need a method designed to handle changes in the spatial domain. With this in mind, for Subdomain-1, we use a semi-discrete stabilized formulation. This stabilized formulation is based on SUPG and PSPG stabilizations. However, because Subdomain-1 involves a complex object, we use the general-purpose implementation of the semi-discrete formulation, so that we can use unstructured meshes. On the other hand, Subdomain-2 does not involve any objects, and therefore we use a special-purpose implementation of the semi-discrete formulation. This special-purpose implementation takes into account the simplicity of the mesh, and results in faster computations. Furthermore, to capture the details of the wake behavior more accurately, the method used in Subdomain-2 is supplemented with the EDICT. Subdomain-3 contains a complex and moving object. For this subdomain, we use the general-purpose implementation of the DSD/SST formulation developed earlier. In our application here, we focused on the aerodynamics of a rigid parachute, and therefore did not take into account the fluid–structure interactions. Our presentation in this paper included a numerical example. With this example, we showed that, the MDM, which brings together different numerical methods most effectively for different subdomains, enables us to address the complexities involved in the aerodynamics of a parachute crossing the far wake of an aircraft.

Acknowledgements

This work was supported by NASA Johnson Space Center (grant no. NAG9-1059), AFOSR (contract no. F49620-98-1-0214), and by the US Army Natick Soldier Center (contract no. DAAD16-00-C-9222).

The content does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. The second author was supported by the Bridgestone Corp.

References

[1] T.E. Tezduyar, Y. Osawa, Methods for parallel computation of complex flow problems, Parallel Comput. 25 (1999) 2039–2066.
[2] Y. Osawa, V. Kalro, T.E. Tezduyar, Multi-domain parallel computation of wake flows, Comput. Methods Appl. Mech. Engrg. 174 (1999) 371–391.
[3] Y. Osawa, T.E. Tezduyar, A multi-domain method for 3D computation of wake flow behind a circular cylinder, Comput. Fluid Dyn. J. 8 (1999) 296–308.
[4] Y. Osawa, T.E. Tezduyar, 3D simulation and visualization of unsteady wake flow behind a cylinder, J. Visualization 2 (1999) 127–134.
[5] T.J.R. Hughes, A.N. Brooks, A multi-dimensional upwind scheme with no crosswind diffusion, in: T.J.R. Hughes (Ed.), Finite Element Methods for Convection Dominated Flows, AMD, vol. 34, ASME, New York, 1979, pp. 19–35.
[6] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[7] T.E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, Adv. Appl. Mech. 28 (1991) 1–44. (1992)
[8] T.E. Tezduyar, M. Behr, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: I. The concept and the preliminary tests, Comput. Methods Appl. Mech. Engrg. 94 (1992) 339–351.
[9] T.E. Tezduyar, M. Behr, S. Mittal, J. Liou, A new strategy for finite element computations involving moving boundaries and interfaces – the deforming-spatial-domain/space-time procedure: II. Computation of free-surface flows, two-liquid flows, and flows with drifting cylinders, Comput. Methods Appl. Mech. Engrg. 94 (1992) 353–371.
[10] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.
[11] M. Behr, T.E. Tezduyar, Finite element solution strategies for large-scale flow simulations, Comput. Methods Appl. Mech. Engrg. 112 (1994) 3–24.
[12] T.E. Tezduyar, S. Aliabadi, M. Behr, Enhanced-discretization interface-capturing technique (EDICT), in: Y. Matsumoto, A. Prosperetti (Eds.), Proceedings of the ISAC '97 High Performance Computing on Multiphase Flows, Japan Society of Mechanical Engineers, 1997, pp. 1–6.
[13] T.E. Tezduyar, S. Aliabadi, M. Behr, Enhanced-discretization interface-capturing technique (EDICT) for computation of unsteady flows with interfaces, Comput. Methods Appl. Mech. Engrg. 155 (1998) 235–248.
[14] S. Mittal, S. Aliabadi, T.E. Tezduyar, Parallel computation of unsteady compressible flows with the EDICT, Comput. Mech. 23 (1999) 151–157.
[15] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–165.
[16] E.R. Van Driest, On turbulent flow near a wall, J. Aerosp. Sci. 1 (1956) 1007–1011.
[17] T.E. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, V. Kalro, M. Litke, Flow simulation and high performance computing, Comput. Mech. 18 (1996) 397–412.
[18] A.A. Johnson, T.E. Tezduyar, Parallel computation of incompressible flows with complex geometries, Int. J. Numer. Meth. Fluids 24 (1997) 1321–1340.