JOURNAL OF AIRCRAFT
Vol. 46, No. 5, September–October 2009

Three-Dimensional Aerodynamic Simulations of Jumping Paratroopers and Falling Cargo Payloads

Victor Udoewa*
University of Cape Town, Rondebosch 7701, South Africa

DOI: 10.2514/1.42548

The aim is to develop computational techniques for studying aerodynamic interactions between multiple objects when an object exits and separates from an aircraft. The object could be a paratrooper jumping out of a transport aircraft or a package of emergency aid dropped from a cargo plane. In all these cases, the computational challenge is to predict the dynamic behavior and path of the object, so that the separation process is safe and effective. Here we have correctly modeled a paratrooper jumping from a transport aircraft and an emergency aid package dropped from a cargo plane. Because these phenomena have been realistically modeled, we can now make model design changes to the aircraft geometry that beneficially affect the falling trajectories, preventing problems such as paratrooper crossover as well as locating the fallen payload. Moreover, these multibody techniques can be extended to other problems, such as the hemological flow of human immunodeficiency virus virions.

I. Introduction

PARATROOPERS jump from a variety of different aircraft weighted with different supplies depending on the mission and experiencing an array of weather conditions. These meteorological conditions, the shape and size of the aircraft, and turbulence caused by nearby aircraft affect the free fall path of the paratrooper.

The U.S. Army has experienced a myriad of problems associated with this situation. Two examples are crossover and wake crossing. With a specific U.S. Air Force aircraft, the U.S. Army has had difficulties whenever paratroopers jump out of both the left and the right doors simultaneously: their paths tend to cross (crossover). This is, of course, potentially injurious. As paratroopers jump out of multiple U.S. Air Force aircraft, they must fall through the wake of the preceding aircraft (wake crossing). If the wake has not experienced sufficient dissipation, casualties could occur due to the difficulty in pulling parachute chords in such turbulent wake.

Obvious solutions include alternating left- and right-door paratrooper jumps and flying all aircraft very far apart. However, the U.S. Army has highly time-sensitive missions during which they need to exit all paratroopers in as short a time as possible using as little air space as possible. Therefore, they prefer to have paratroopers jumping simultaneously and want to know the safest closest distance these aircraft can be flown to avoid wake-crossing casualties.

Cargo drops also present a problem. Often, the U.S. Army may want to make a supply drop to an operative in another country. Alternatively, they want to drop off a shipment to a group of soldiers. In those cases, the U.S. Army wants to be able to pinpoint the location of a drop within a certain radius, so that it can be easily and quickly found. For that reason, cargo drops are also of interest. Some drops use an exit parachute to extract the payload from the cargo bay. Other cargo payloads slide out on frictionless tracks only with the help of gravity. The ability to predict the trajectory of a certain payload and its final target can significantly expedite many operations.

Wind-tunnel experiments can be even more inflexible when dealing with paratrooper jumps. How does one realistically model a falling paratrooper in a wind tunnel? Moreover, experimental data from actual jumps are also hard to obtain. The limitations of experimental research in this area are clear. First, the costs can be prohibitive. These include labor, soldier hours, pilot hours, fuel, aircraft rental, airfield rental, etc. Second, the time can be restrictive. One test jump can be planned for over an entire year. The amount of personnel needed creates inconvenience. Numerical modeling solves all of these problems. It is much cheaper, expends less time for the same experiment, and is more convenient, requiring less people (feasibly one).

Here, we explore simulation and modeling techniques for aerodynamic fluid–object interactions (FOI) between multiple objects. All objects will be treated as rigid bodies. The specific applications of jumping paratroopers and falling cargo payloads from cargo aircraft will be emphasized.

The computational tools developed here are based upon the simultaneous solution of the 3-D time-dependent Navier–Stokes equations governing the incompressible airflow around the aircraft and the separating object, as well as the equations governing the motion of that object. These computational methods include suitable mesh update techniques to be used in conjunction with the deforming spatial domain/stabilized space–time (DSD/SST) formulation [1].

Previously, techniques such as arbitrary Lagrangian–Eulerian formulation were used for moving problems with both finite difference modeling, finite volume modeling [2–5], and finite element modeling [6–9]. Because the relative positions of the aircraft and the separating object are changing in time, the DSD/SST formulation is written over the corresponding space–time domain of the problem and can therefore automatically handle the changes in the spatial domain. This method has been tested on many problems [10–12].

The computations presented use mesh stiffening tactics, remeshing techniques, and projection methods that work well for such a large-scale problem with arbitrary geometries [13]. However, many still have looked to developing alternatives to mesh moving. The fluid–object interaction subcomputation technique (FOIST) is such an alternative [1]. In this paper, we apply it to our problem to determine how well it can approximate the trajectories we seek. A 3-D slip approximation for arbitrary surfaces is applied here as well [1]. Again, the goal is the same.

For now, in this work, we commence with the governing equations for fluid flow. The fluid flow is governed by the Navier–Stokes equations for incompressible flow. Section II is a presentation of those equations with the constitutive relations and the boundary and initial conditions. Instead of directly resolving the turbulent flow features present at the Reynolds numbers of the problems presented in this research, turbulence affects are accounted for by using a zero-equation Smagorinsky turbulence model [14].

Received 8 December 2008; revision received 29 December 2008; accepted for publication 18 January 2009.

*Postdoctoral Fellow, Centre for Engineering Research and Computational and Applied Mathematics, Menzies Building, 5th Floor, Upper Campus; udwvic001@uct.ac.za.

The finite element formulations for the fluid dynamics (FD) governing equations are also explicated in Sec. II. In this section, for the fluid dynamics, the DSD/SST method is outlined. Related equations are highlighted.

Then, in Sec. III, we look at the jumping paratrooper simulation comparing FOIST, analytical results, experimental results, and a slip approximation. The same is done for the falling cargo payload in Sec. IV. Our goal is to accurately simulate the trajectory and separation phenomena. Unless otherwise stated, computations were carried out on a Cray T3E parallel supercomputer. All photographs were obtained by permission of the U.S. Army and U.S. Air Force either directly or through their web sites.

A finalizing overview of contributions and implications are made in the concluding section. This is followed by proposed areas for future research.

II. Materials and Methods

A. Navier–Stokes Equations of Incompressible Flows

The fluid is assumed to be incompressible and will be modeled with the Navier–Stokes equations of incompressible flow. Let $\Omega_t \subset R^{n_{sd}}$ be the spatial fluid mechanics domain and $(0, T)$ be the temporal domain, where n_{sd} is the number of space dimensions, letting Γ_t denote the boundary of the spatial domain Ω_t. The subscript t evinces the time dependence of the spatial domain and its boundary. The spatial and temporal coordinates are denoted by $\mathbf{x} = (x, y, z)$ and $t \in (0, T)$. The Navier–Stokes equations of incompressible flow can be written as follows:

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

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

where ρ is the constant fluid density, $\mathbf{f}$ is an external force term such as gravity, and σ is the stress tensor. Equation (1) ensures the conservation of momentum for the fluid system, Eq. (2) the conservation of mass. The flow variables that are being calculated are the velocity, $\mathbf{u}(\mathbf{x}, t)$, and pressure, $p(\mathbf{x}, t)$, which is embedded in the stress tensor defined in the next section.

1. Stress and Strain Definitions

Considering Newtonian fluids, one assumes a linear relationship between the fluid stress and rate-of-strain tensors. The stress tensor is written as the sum of its isotropic and deviatoric parts:

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

where $\propto$ is the dynamic viscosity coefficient, $\mathbf{I}$ is the identity tensor, p is pressure, and $\varepsilon(\mathbf{u})$ is the strain rate tensor:

$$\varepsilon(\mathbf{u}) = \tfrac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) \qquad (4)$$

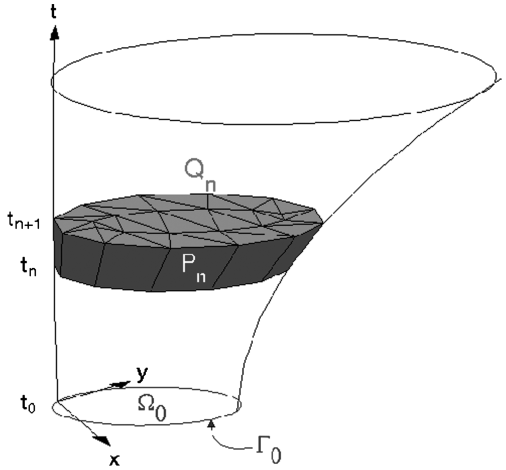

Fig. 1 Two-dimensional space–time slab concept.

2. Boundary and Initial Conditions

To appropriately represent the fluid dynamics, proper boundary conditions must be imposed on the outer boundaries of the fluid domain and on any inner surfaces, ensuring an accurate fluid environment. Both the (essential) Dirichlet- and (natural) Neumann-type boundary conditions are accounted for and are represented as

$$\mathbf{u} = \mathbf{g} \quad \text{on } (\Gamma_t)_g, \qquad \mathbf{n}\cdot\sigma = \mathbf{h} \quad \text{on } (\Gamma_t)_h \qquad (5)$$

where $(\Gamma_t)_g$ and $(\Gamma_t)_h$ are complementary subsets of the boundary Γ_t related to the Dirichlet- and Neumann-type boundary conditions, respectively.

In time-dependent problems, an initial condition is required. The initial condition on the velocity is specified as $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0$ on Ω_0, where $\mathbf{u_0}$ is divergence free.

B. Finite Element Formulations

To handle the time-variant spatial domains encountered in aerodynamic separation problems, the DSD/SST formulation is used.

To construct the finite element function spaces for the space–time method, the time interval $(0, T)$ must be 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 $0 = t_0 < t_1 < \cdots < t_n = T$. Letting $\Omega_n = \Omega_{t_n}$ and $\Gamma_n = \Gamma_{t_n}$, the space–time slab Q_n is defined as the domain enclosed by the

Fig. 2 Cargo aircraft in formation with paratroopers exiting.

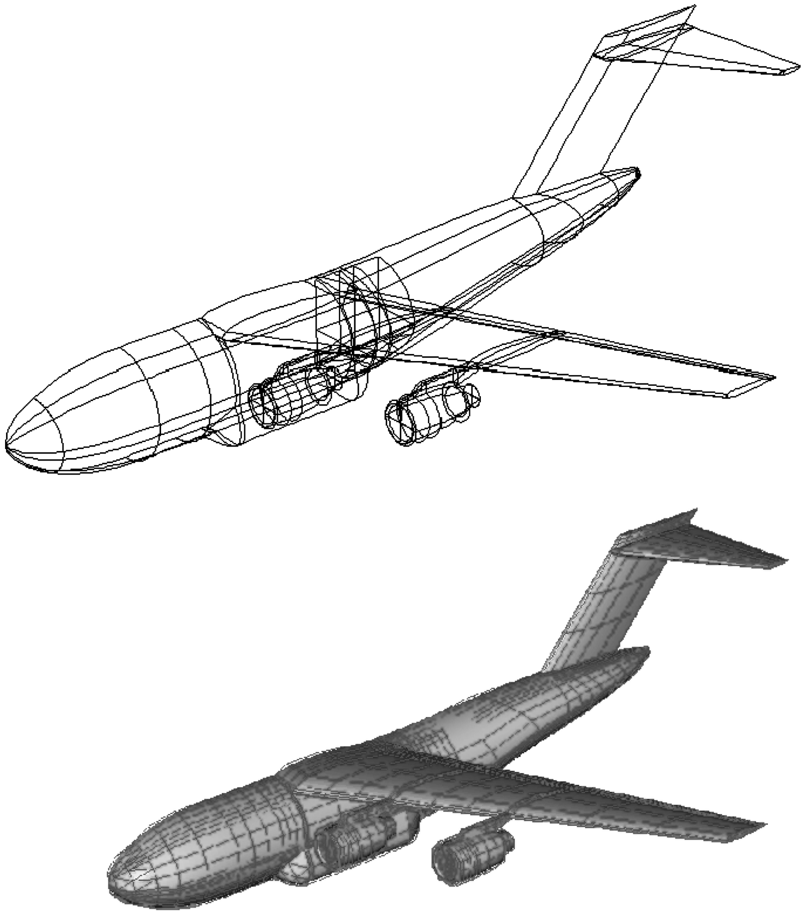

Fig. 3 Original cargo aircraft: a) wire frame model, and b) shaded wire frame model.

surfaces Ω_n, Ω_{n+1}, and P_n, where P_n is the surface inscribed by the boundary Γ_t as t traverses I_n. The space–time concept is depicted in Fig. 1 for the space–time slab Q_n.

The surface P_n is decomposed into $(P_n)_g$ and $(P_n)_h$ with respect to the type of boundary condition (Dirichlet or Neumann, respectively) being imposed. For each space–time slab, the corresponding finite element function spaces $((\mathcal{S}^h)_{\mathbf{u}})_n$, $((\mathcal{V}^h)_{\mathbf{u}})_n$, $((\mathcal{S}^h)_p)_n$, and $((\mathcal{V}^h)_p)_n$ are defined as follows:

$$((\mathsf{S}^h)_{\mathbf{u}})_n = \{\mathbf{u}^h | \mathbf{u}^h \in [H^{1h}(Q_n)]^{n_{\mathrm{sd}}}, \mathbf{u}^h \doteq \mathbf{g}^h \quad \text{on } (P_n)_g\} \tag{6}$$

$$((\mathsf{V}^h)_{\mathbf{u}})_n = \{\mathbf{w}^h | \mathbf{w}^h \in [H^{1h}(Q_n)]^{n_{\mathrm{sd}}}, \mathbf{w}^h \doteq \mathbf{0} \quad \text{on } (P_n)_g\} \tag{7}$$

$$((\mathsf{S}^h)_p)_n = ((\mathsf{V}^h)_p)_n = \{q^h | q^h \in H^{1h}(Q_n)\} \tag{8}$$

Here, $H^{1h}(Q_n)$ is the finite-dimensional function space over the space–time slab Q_n. First-order polynomials in both space and time are used to form the element domain. The interpolation functions, however, are continuous in space but discontinuous in time.

The stabilized space–time formulation for deforming domains is now written as follows: given $(\mathbf{u}^h)_n^-$, find $\mathbf{u}^h \in ((\mathcal{S}^h)_{\mathbf{u}})_n$ and $p^h \in ((\mathcal{S}^h)_p)_n$ such that $\forall \mathbf{w}^h \in ((\mathcal{V}^h)_{\mathbf{u}})_n$ and $q^h \in ((\mathcal{V}^h)_p)_n$,

$$\begin{aligned}
&\int_{Q_n} \mathbf{w}^h \cdot \rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h + \mathbf{f}^h\right) \mathrm{d}Q + \int_{Q_n} \varepsilon(\mathbf{w}^h) \colon \sigma(p^h, \mathbf{u}^h)\, \mathrm{d}Q \\
&+ \int_{Q_n} q^h \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q + \sum_{e=1}^{n_{\mathrm{el}}} \int_{Q_n^e} \frac{\tau}{\rho}\left[\rho\left(\frac{\partial \mathbf{w}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \mathbf{f}^h\right)\right. \\
&\left. - \nabla \cdot \sigma(q^h, \mathbf{w}^h)\right] \cdot \left[\rho\left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h + \mathbf{f}^h\right)\right. \\
&\left. - \nabla \cdot \sigma(p^h, \mathbf{u}^h)\right] \mathrm{d}Q + \sum_{e=1}^{n_{\mathrm{el}}} \int_{Q_n^e} \delta \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h \, \mathrm{d}Q \\
&+ \int_{\Omega_n} (\mathbf{w}^h)_n^+ \cdot \rho((\mathbf{u}^h)_n^+ - (\mathbf{u}^h)_n^-)\, \mathrm{d}\Omega \\
&= \int_{(P_n)_h} \mathbf{w}^h \cdot \mathbf{h}^h \, \mathrm{d}P
\end{aligned} \tag{9}$$

This process is applied sequentially to all the space–time slabs Q_0, Q_1, Q_2, …, Q_{N-1}. The explanation of certain notation in Eq. (9) follows:

$$(\mathbf{u}^h)_n^{\pm} = \lim_{\varepsilon \to 0} \mathbf{u}(t_n \pm \varepsilon) \tag{10}$$

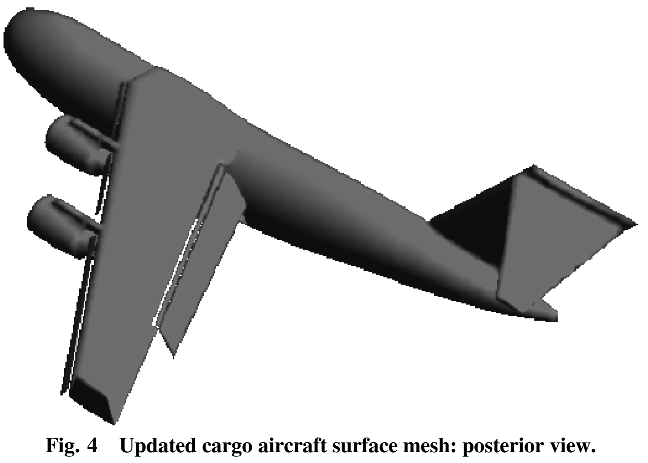

Fig. 4 Updated cargo aircraft surface mesh: posterior view.

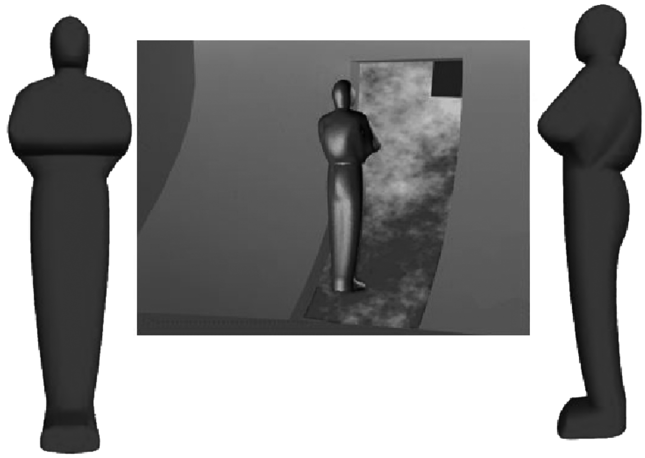

Fig. 5 Original paratrooper model.

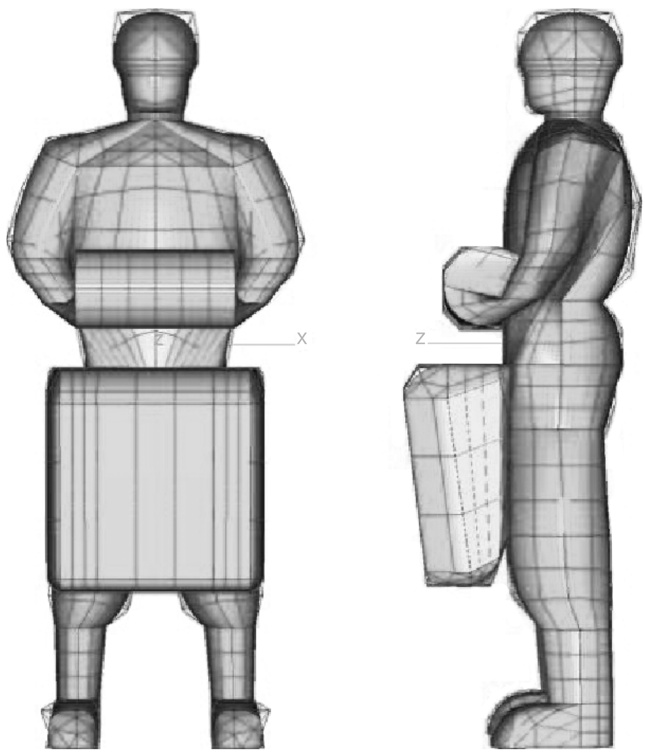

Fig. 6 Updated paratrooper model.

$$\int_{Q_n} (\cdots)\, \mathrm{d}Q = \int_{I_n} \int_{\Omega_t} (\cdots)\, \mathrm{d}\Omega\, \mathrm{d}tP \qquad (11)$$

$$\int_{P_n} (\cdots)\, \mathrm{d}P = \int_{I_n} \int_{\Gamma_t} (\cdots)\, \mathrm{d}\Gamma\, \mathrm{d}tP \qquad (12)$$

The computations start with

$$(\mathbf{u}^h)_0^- = \mathbf{u}_0 \qquad (13)$$

In the variational formulation given by Eq. (9), the first three terms, the sixth term, and the right-hand side constitute the Galerkin formulation of the problem, the momentum balance equation and the mass balance equation. The sixth term weakly enforces continuity of the velocity field across the space–time slabs, because the interpolation functions are discontinuous in time and because the equation is solved one space–time slab at a time. Because of numerical instabilities that occur in advection-dominated flows and from oscillations that occur when different combinations of interpolation functions for velocity and pressure are used, stabilizing terms are included. The first series of element-level integrals in Eq. (9) are least-squares terms based on the momentum equation. This term stabilizes the standard Galerkin form. The second series of element-level integrals are added to the formulation for numerical stability at high Reynolds numbers. These are least-squares terms based on the continuity equation. The stabilization coefficients τ and δ are defined at the element level. Both stabilization terms are weighted residuals and, therefore, maintain the consistency of the formulation. For an exact solution, they converge to zero. Further

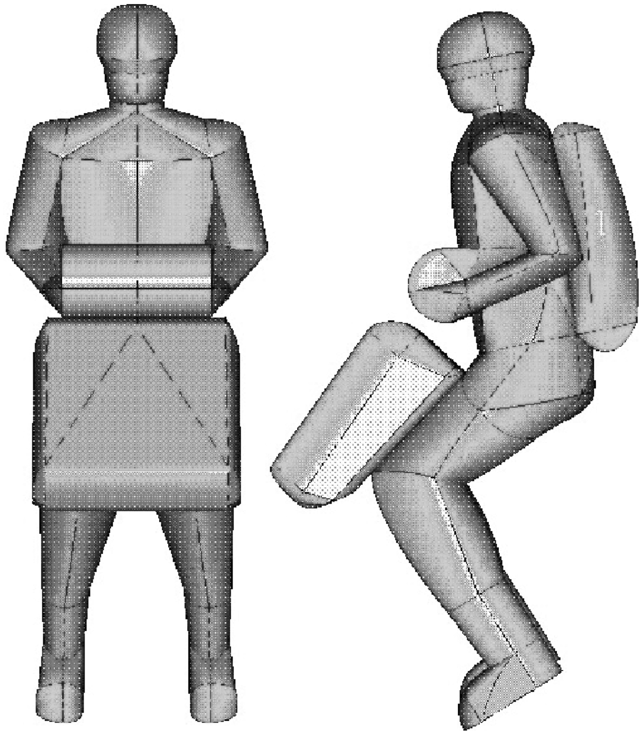

Fig. 7 Updated paratrooper model with parachute pack.

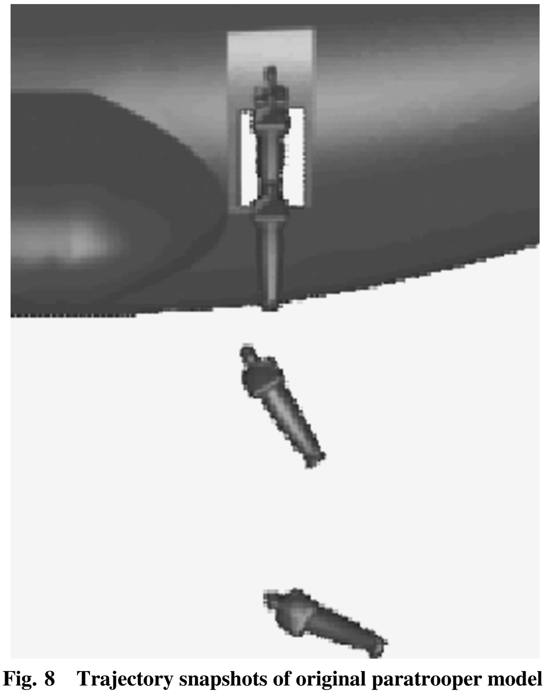

Fig. 8 Trajectory snapshots of original paratrooper model.

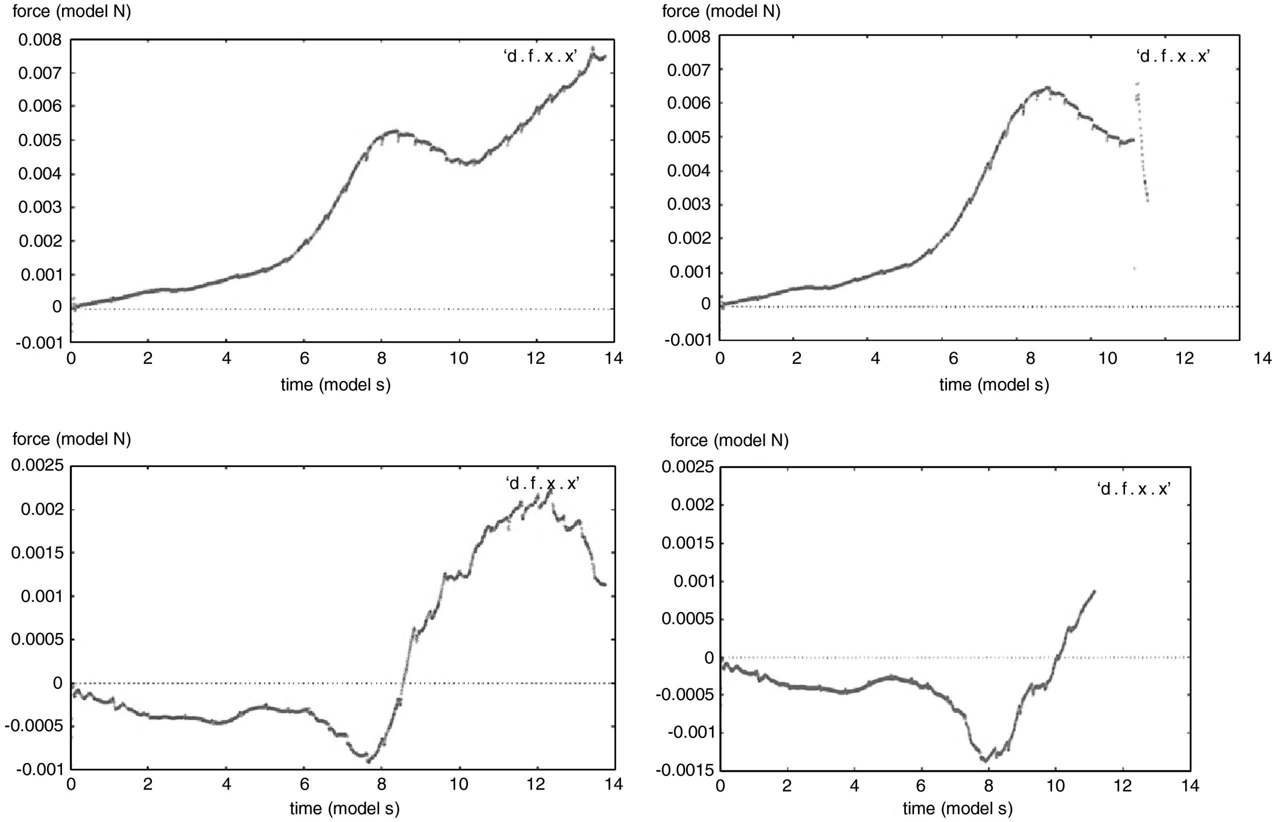

Fig. 9 Force on paratrooper: a) freestream component of force felt by paratrooper in which the inertia is treated as a vector, b) freestream component of force felt by paratrooper using a mass moment of inertia tensor, c) transversely (wingspan) directional component of force felt by paratrooper in which inertia is treated as a vector, and d) transversely (wingspan) directional component of force felt by paratrooper using a mass moment of inertia tensor.

discussions of these stabilization terms, including their derivations, can be found in the literature [15,16].

III. Results: Paratrooper Simulation

A. Fluid Model

In modeling the cargo aircraft, symmetry is assumed with respect to the 2-D plane passing through the middle of the aircraft from nose to tail. Therefore, only half of the aircraft is modeled for this simulation of a jumping paratrooper (Fig. 2). Seen in Fig. 3, the original aircraft model was introduced by Johnson in Tezduyar et al. [17]. In this work, wing flaps and wing tips were added to construct a more accurate model. This can be seen in Fig. 4.

The model update was performed to improve the results and accurately simulate the trajectory of the jumping paratroopers. By more accurately modeling the aircraft, a more accurate airflow can be computed about the aircraft. This air flow then helps determine the path of the paratrooper.

Additionally, the paratrooper model needed to be updated. Previously called Oscar due to its Academy Award resemblance, the older model by Johnson had a disproportionate volume relative to its height, as shown in Fig. 5. A real paratrooper is weighted by his equipment, increasing his volume. Even without such equipment, the average paratrooper's volume is increased by his bulky uniform. Without a proportionally correct volume, the paratrooper model has always suffered from a relatively high density. An increase in the accuracy of the simulation should result from an update of the model of the paratrooper.

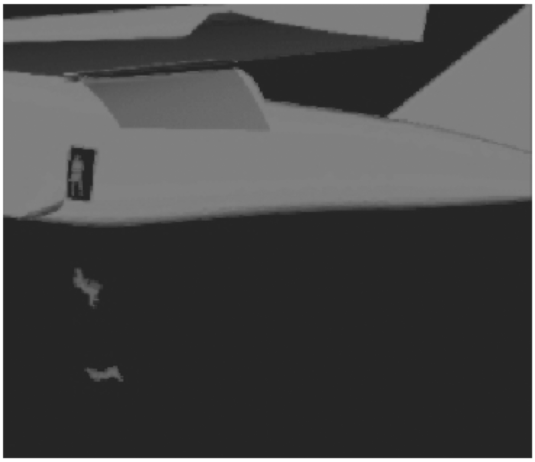

Fig. 10 Old trajectory of new paratrooper model.

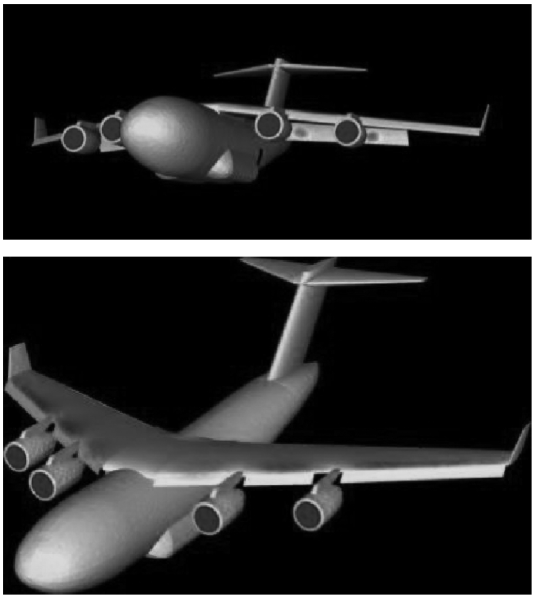

Fig. 11 Pressure distribution on cargo aircraft.

Front and side views of the paratrooper can be seen in Fig. 6. It should be noted that this paratrooper has no parachute pack on his back. This particular model is used when simulating later stages in the jump of the paratrooper, such as the time when the paratrooper falls through the wake of a preceding aircraft. At that point, the parachute pack has fully opened. This work focuses on the first 2–3 s of the paratrooper jump; the model used for this work is displayed in Fig. 7.

Three-dimensional triangular surface meshes were then created from the models, and then a 3-D tetrahedral volume mesh was generated for the fluid dynamics solution using these surface meshes. The mesh consists of 129,090 nodes and 728,902 tetrahedral elements. The modeling software, a 3-D surface mesh generator, and an automatic 3-D mesh generator were all developed by Johnson and Tezduyar [18].

B. Fluid–Object Interactions Setup

In this example, the U.S. Air Force cargo aircraft is traveling at 130 kt, or 66.88 m/s. This is a typical cruising speed for such drop missions as a paratrooper jumping or a cargo payload dropping. The aircraft flies with an angle of attack of 8 deg. The following boundary conditions are assigned for the FD model:

1) The *cargo aircraft and paratrooper surfaces* are prescribed a no-slip boundary condition for all components of velocity.

2) The *inflow boundary in front of the aircraft* is assigned a prescribed boundary condition, with the velocity components representative of the freestream airspeed.

3) The *crossflow and side boundaries* are assigned slip conditions, with no normal flow permitted through the boundaries.

4) The *outflow boundary behind the aircraft* is assigned a traction-free boundary condition.

In defining the parameters of the problem with nondimensional numbers, the length of the aircraft (nose to tail wing tip) is taken as 8.81 model units and the freestream velocity as 1 model unit per model time units. For the engines, the intake and exhaust flow velocity profiles are assumed to be uniform and at 1.0 and 3.0 model units per model time unit, respectively.

C. Fluid–Object Interactions Simulation

To correct the unreal, previously simulated trajectory with the Oscar model, shown in Fig. 8, the paratrooper model was updated and given a 3-D mass moment of inertia (previously treated as a sphere with an inertia vector as shown in Fig. 9). Figure 10 displays the old trajectory (using an inertia vector) of the new paratrooper model.

With the updates and corrections, a true, viable trajectory was finally simulated. The pressure distribution on the aircraft during the simulation can be viewed in Fig. 11. The jet engines are colored blue.

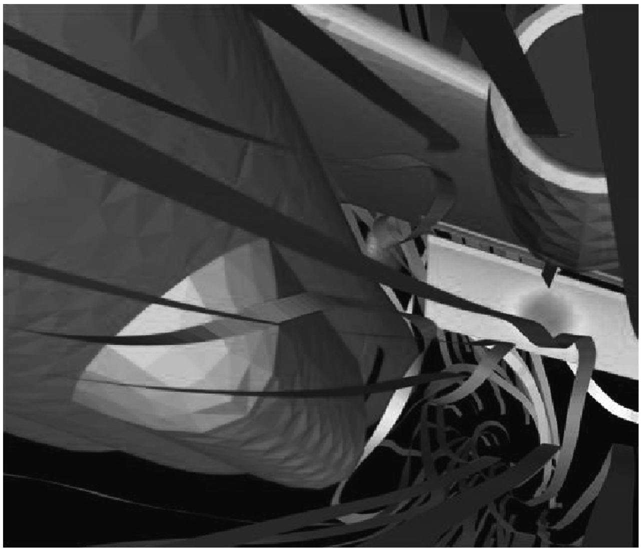

Fig. 12 Stream ribbons about cargo aircraft colored with pressure.

Figure 12 shows the turbulent flow the paratrooper must jump into. The pressure distribution on the surface of the paratrooper and surrounding streamlines can be seen in Fig. 13.

Produced after running the corrected simulation with the new, improved models, the new trajectory is shown in Fig. 14.

To test the accuracy of the simulation, the distance and slope information of real trajectories were extracted from videotaped footage of paratroopers jumping from a U.S. Air Force cargo aircraft. A comparison will be made between two of those real trajectories, the simulated trajectory, and an analytical trajectory.

From the many hundreds of videotaped jumps, two real trajectories were chosen as extremes: one paratrooper seemingly the most swept back or experiencing the most drag, and one paratrooper seemingly the least swept back, or experiencing the least amount of drag. The analytical trajectory was calculated using an estimated coefficient of drag of 2.3. Using Eq. (14), the drag force on the paratrooper is calculated:

$$F_D = C_D \rho V^2 A \tag{14}$$

where F_D and C_D are the drag force and coefficient of drag, respectively, ρ is the density, V is the speed of the flow, and A is the projected surface area exposed to the flow. The speed of the airflow is taken from the numerical simulation, which is 66.8 m/s. Summing with the gravitational force, a net force is calculated from which an

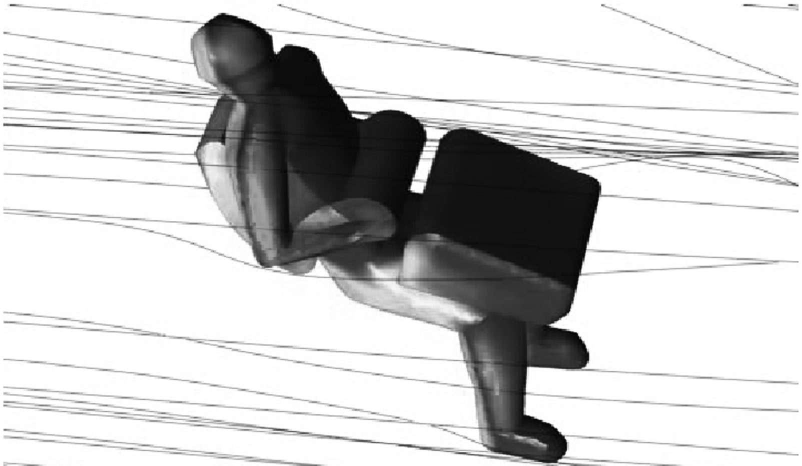

Fig. 13 Pressure distribution on paratrooper with streamlines.

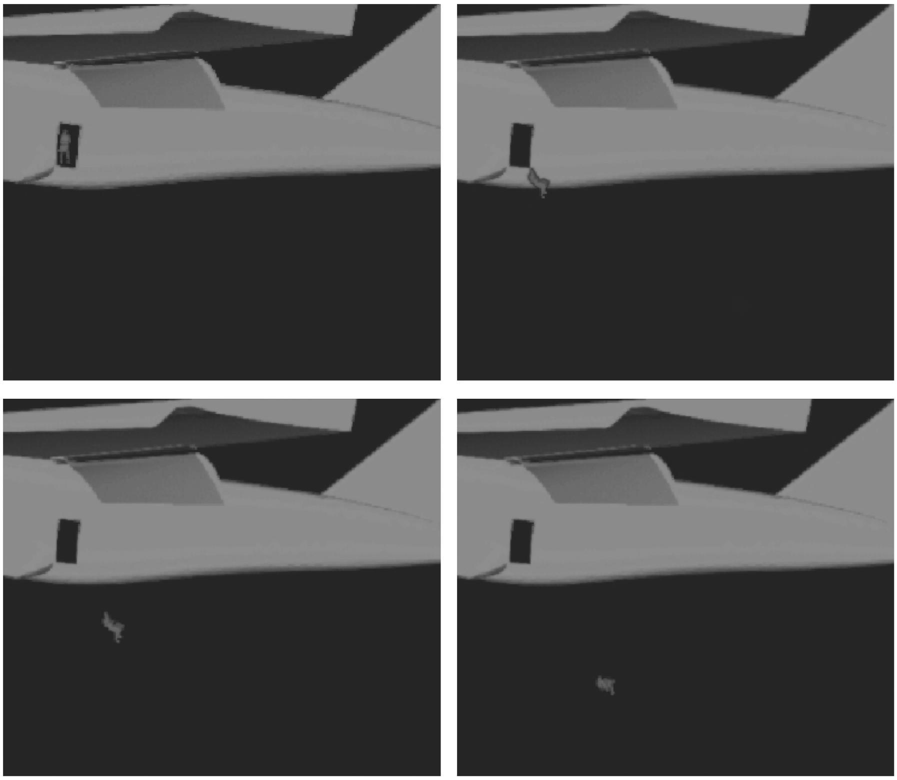

Fig. 14 New trajectory snapshots of new paratrooper model.

acceleration is derived. This is then integrated twice to produce the velocity and the new displacement.

Figure 15 compares the real trajectories with the analytical and numerical ones. The slopes of the analytical, numerical, and two real trajectories are −0.586, −1.687, −1.141, and −0.713, respectively. These slopes were measured only using the starting and ending points of each trajectory. The analytical trajectory was swept back the most. It can be observed that the trajectory produced from the numerical simulation is a possible, viable trajectory when compared with the real trajectory of the least swept back paratrooper. Because the paratroopers' jumps are so different and erratic, much freedom is allowed in the description of an "average" jump. In all videotaped jumps, there is no standard body configuration of the falling paratroopers and definitely no one standard jump trajectory. Therefore, because the trajectory from the numerical simulation is close to that of one of the real trajectories (the least swept back one), it is a possible trajectory.

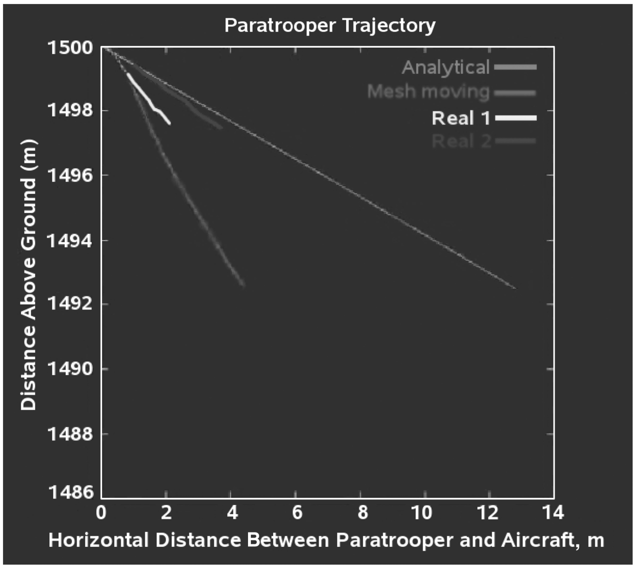

Fig. 15 Paratrooper trajectory comparison: real vs numerical vs analytical.

Still, it should be possible to simulate the real trajectory of the most swept back paratrooper. Believing this is due to a numerically thick boundary layer, especially because the analytical trajectory is more swept back than the numerical one, it is hoped that using a slip method will improve the solution.

1. *Slip Simulation*

Here, the same separation problem of the jumping paratrooper exiting from the cargo aircraft is simulated with slip conditions implemented implicitly [19]. The boundary conditions are the same as stated in Sec. III.A changing only the boundary condition on the aircraft. It is now given a slip condition allowing the fluid particles to flow along the surface of the aircraft, thereby eliminating the boundary layer and the low drag problem as before [19] (Fig. 16).

What is new here is a comparison of this trajectory to real videotaped trajectories, testing the accuracy of the slip simulation trajectory. Figure 17 clearly compares all trajectories and validates the numerical results. Using the slip approximation, clearly, this code has risen to prove new possible, viable, and true trajectories for the high-drag experiences of real paratroopers.

2. *Fluid–Object Interaction Subcomputation Technique Simulation*

Lastly, in this section, we compare the effects of the new FOIST method on such a complex 3-D problem as the aircraft–paratrooper separation problem [20]. FOIST is an alternative technique to mesh moving. The aim is to determine if the trajectory is approximated well enough that the cost savings are justified.

Accordingly, a subcomputational domain was modeled and constructed containing only the paratrooper. This is displayed in Fig. 18. Then, a main flow was simulated around the aircraft without the paratrooper assuming that the influence of the paratrooper on the main flowfield is negligible. Then, according to FOIST methodology, the subcomputation around the paratrooper was simulated using projected data from the main flowfield, as shown in Fig. 19. This produces a torque and force that is used to find the resulting

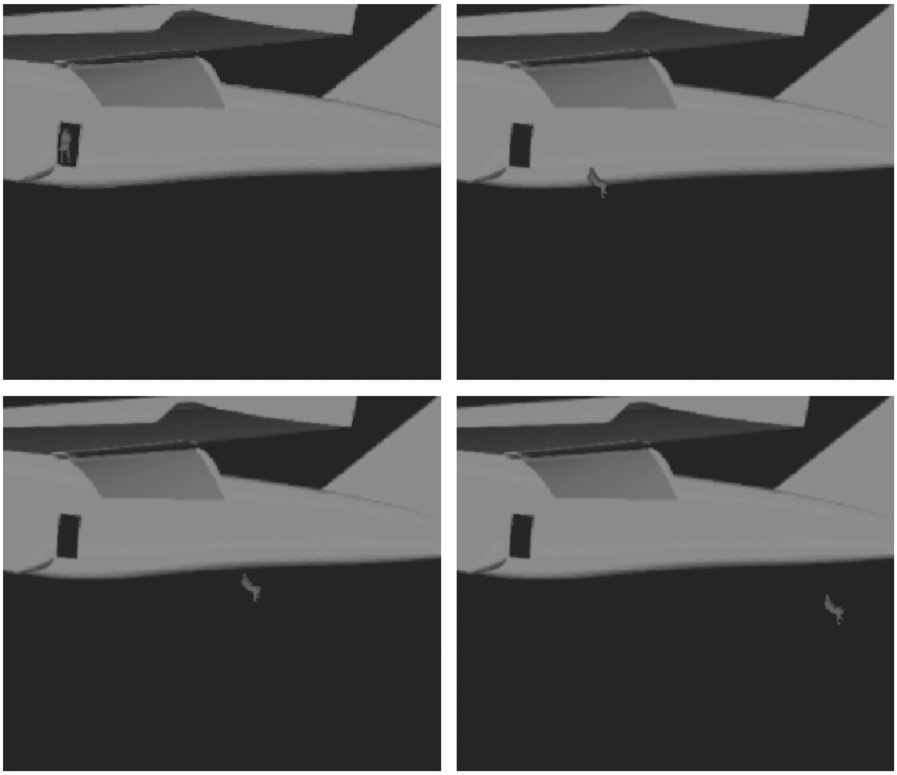

Fig. 16 Paratrooper trajectory with slip conditions on aircraft.

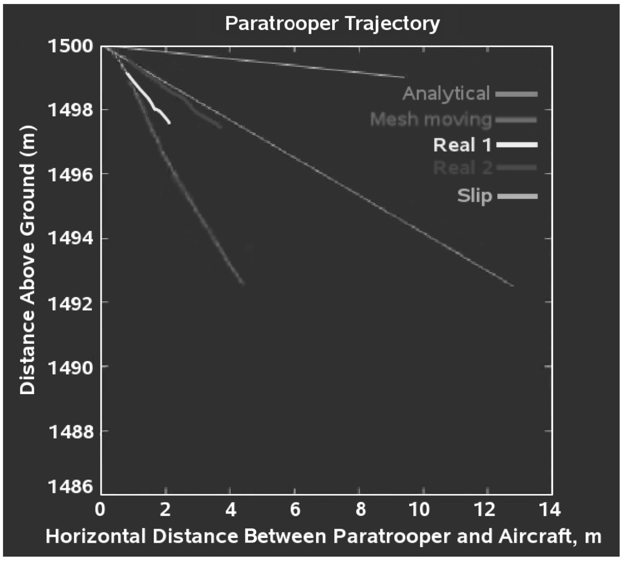

Fig. 17 Paratrooper trajectory comparison: real vs numerical vs analytical vs slip.

displacement and rotation of the paratrooper as if the paratrooper were in the main flowfield. The process is repeated, retrieving boundary conditions for the subcomputation from the new location in the main flowfield. By repeating this process, a trajectory is produced.

Ostensibly, the new FOIST approximation does not match perfectly with the mesh moving case. This is due to the experience of less drag due to approximations of boundary conditions and the closeness of the subcomputational walls to the object. These approximations are further explained by Udoewa in the first FOIST paper [20]. Here, the FOIST approximation is not too poor and is satisfactory in presenting a general idea of the shape and extent of the paratrooper trajectory. This can be a helpful tool in analytical and directive preprocessing. This is exactly the type of tool to be used in the prognostication of the trajectory tendencies of separating objects. Also, as the simplicity of the problem increases, so does the approximating power of FOIST.

IV. Results: Cargo Payload Simulation

Our next task is to simulate cargo payloads falling from cargo aircraft (Figs. 20 and 21). This is another separation problem, and the same techniques may be employed as before. Moreover, this is an entirely new application for these methods.

Such cargo drops could be used to deliver a package to an operative in another country. If the landing point of the package can be pinpointed within a 500 m radius, the operative can locate the package. The importance of such predictions to the operative is obvious.

Here again we focus on the first 2–3 s of the cargo drop after the payload exits the aircraft. The aim is to observe and simulate real trajectories in an effort to predictively determine the future paths of similar objects to be deployed from such cargo aircraft. This information can be adapted for various military missions.

A. Fluid Model

In modeling the cargo aircraft for use with payload drops, the location of the cargo door must be taken into account. It is located along the symmetry line of the plane in the center of the bottom of the aircraft fuselage. For this reason, to simulate payload drops directly from the center of the cargo door, the entire cargo aircraft must be modeled. Two views of the newly designed and created cargo aircraft are shown in Figs. 22 and 23.

Next, a cargo prototype was found (Fig. 24). Because no specific prototype was needed as long as it fit within the cargo aircraft, the first picture found was used. It is interesting to note the hole in the middle of the payload body. Perhaps interesting aerodynamics will be seen through that cavity.

Two different views of the payload model can be seen in Figs. 25 and 26. The cavity in the payload prototype was retained in the model. The model was created to convey the general shape of the prototype, but specifics, such as the crossbars in Fig. 24, were not necessary for analyzing the trajectory of general cargo payloads and

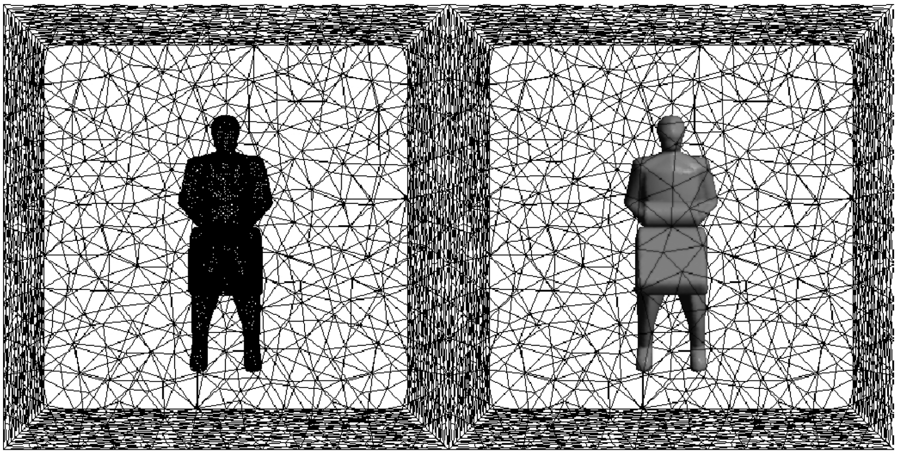

Fig. 18 Model of FOIST subcomputational domain containing paratrooper.

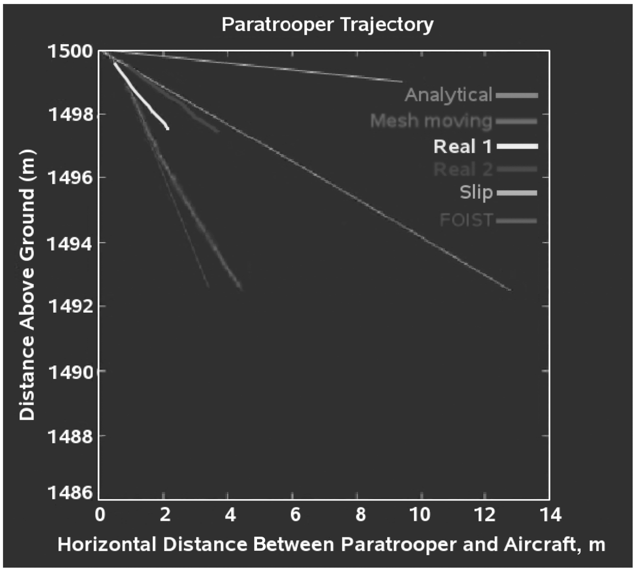

Fig. 19 Paratrooper trajectory comparison: real vs numerical vs analytical vs slip vs FOIST.

how much they are influenced by the aerodynamics of the specific cargo aircraft. Again, in this problem, the focus falls on only the first 2–3 s of the payload drop. The later parachute dynamics have been analyzed in other works [21,22].

Three-dimensional triangular surface meshes were created from the models. Then, the 3-D tetrahedral volume mesh was generated for the fluid dynamics solution using these surface meshes. This mesh consists of 332,498 nodes and 1,952,559 tetrahedral elements. It is important to note that the computational costs here are greater than those of the paratrooper simulation. This is due to the use of the full model of the aircraft, whereas the paratrooper simulation uses a half-model. The modeling software, 3-D surface mesh generator, and automatic 3-D mesh generator were the same used for the paratrooper simulation, all developed by Johnson and Tezduyar [18].

B. Fluid–Object Interactions Setup

In this example of a payload drop, the speed of the cargo aircraft, the angle of attack of the aircraft, the boundary conditions for the problem, and the length of the aircraft are the same as those in Sec. III. The same is true of the engine intake and exhaust flow velocities and the freestream velocity. The only difference in this simulation is the separating object, that is, cargo payload, and the model of the aircraft, that is, a full model with upper cargo bay doors opening upward and inward and lower cargo bay doors opening downward and outward (Fig. 22 and 23).

Fig. 20 Payloads with parachutes falling from bay of cargo aircraft.

Fig. 21 Payloads falling from C-130 aircraft.

C. Fluid–Object Interactions Simulation

The payload drop simulation was computed, and Fig. 27 displays the pressure distribution for this simulation. Figure 28 shows the stream ribbons that travel around the cargo room and the back of the aircraft. A side view is shown in Fig. 29.

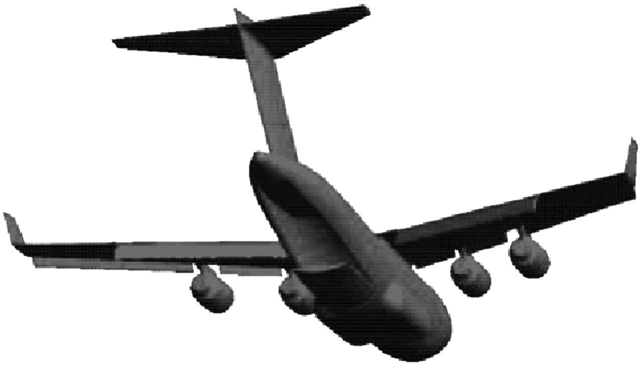

Fig. 22 New cargo aircraft model with cargo bay.

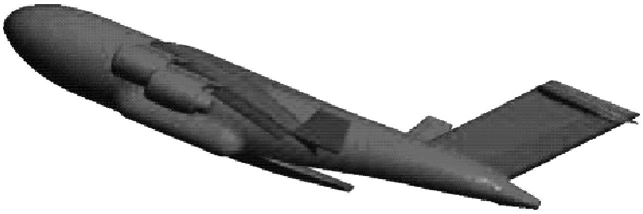

Fig. 23 Cargo aircraft with cargo bay, side view.

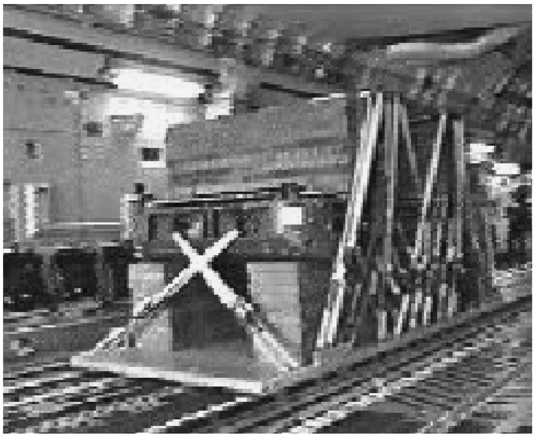

Fig. 24 Cargo payload prototype.

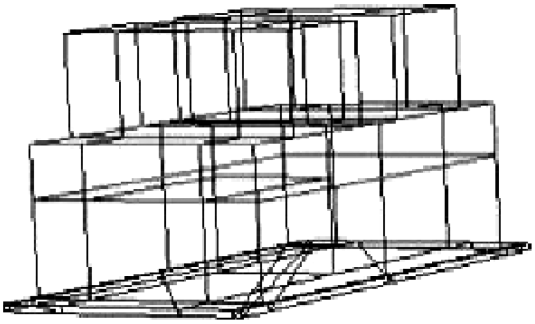

Fig. 25 Cargo payload model.

For this simulation, the payload slides along a frictionless cargo floor before tipping over the edge and falling down to the ground below. The code used for this research does not have contact capabilities, and so the sliding motion of the cargo payload had to be prescribed. To prescribe this noncontacting motion, some tiny bit of air between the bottom of the payload and the cargo floor was allowed. Figure 30 displays a surface mesh of the payload in the pretipping stage.

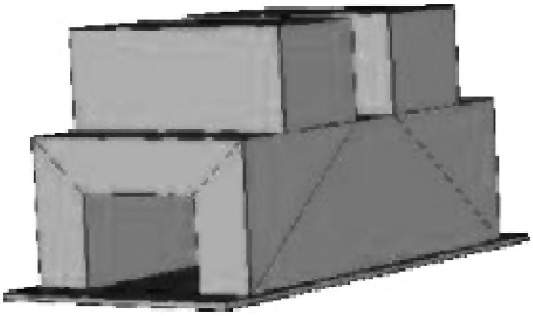

Fig. 26 Cargo payload model, shaded view.

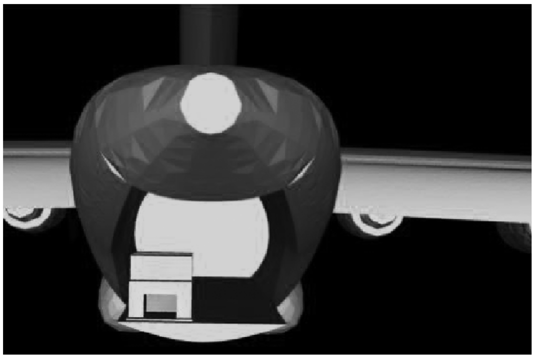

Fig. 27 Pressure distribution on cargo payload and aircraft.

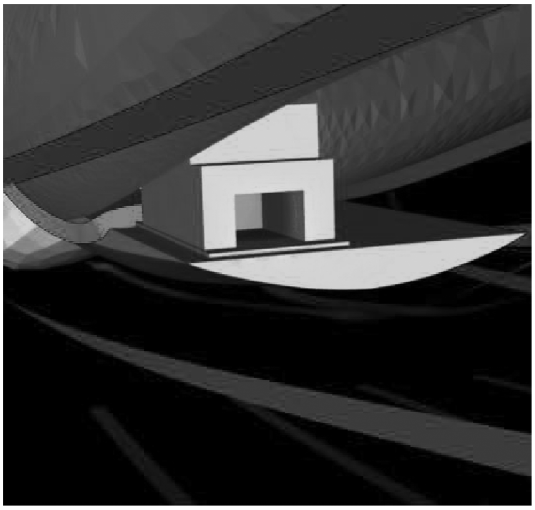

Fig. 28 Stream ribbons about cargo payload.

In the case of the payload drop, there were no videotapes given to allow real comparisons. With less information than for the paratrooper jumps but with a code validated by the jumping paratrooper experiments, the comparison is now made between the simulated mesh moving trajectory and an analytical trajectory.

The analytical trajectory was calculated using an estimated coefficient of drag of 1.5. Using Eq. (14) again and the same values for the variables in the equation, the drag force and trajectory were calculated analytically using the freestream value, as in Sec. III.C. A trajectory for the payload drop was simulated using the described models and specifications. This trajectory is shown in Fig. 31 (Fig. 32 is with slip conditions).

Figure 33 compares the analytical trajectory with the numerical one. It can be observed that the trajectory produced from the numerical simulation is a possible, viable trajectory when compared with the analytical trajectory. The slope of the analytical trajectory is

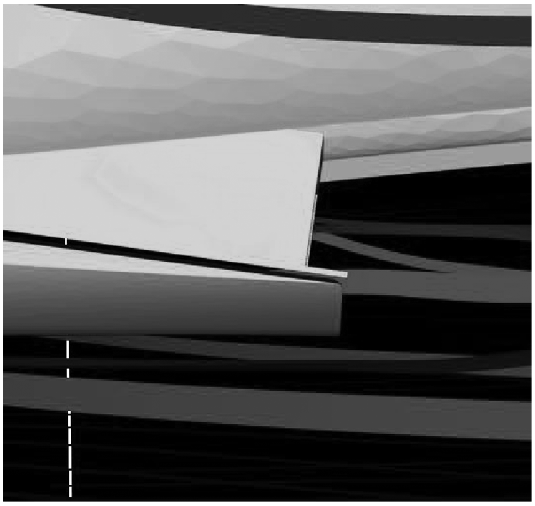

Fig. 29 Stream ribbons about cargo payload, side view.

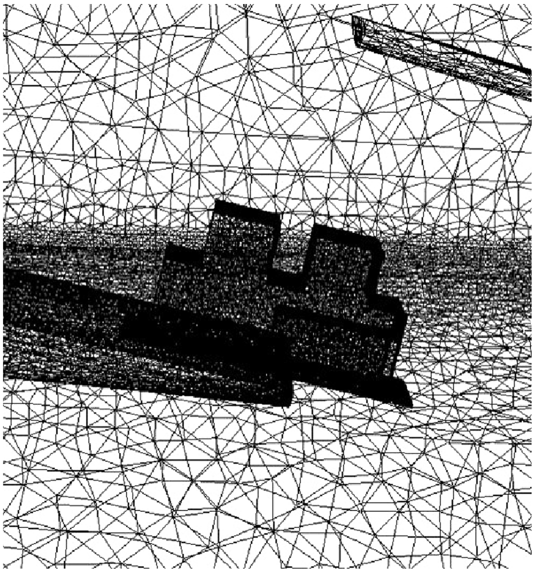

Fig. 30 Surface mesh of cargo in pretipping stage.

−2.347 and that of the numerical or computational one is −2.324, which is close. There seems to be no drag problem here when compared with the analytical solution because the numerical trajectory is swept back more.

Still, some may feel that if a real, physical trajectory were obtained, the cargo payload would be swept back even more. However, the cargo trajectory is gravity dominated. In other words, any effects of a thick numerical boundary layer will not be as prominent as they were in the paratrooper simulation. This is because the thick numerical boundary layer effects the calculation of drag. Because gravity dominates here, a slip formulation should have a smaller effect than it did in the paratrooper simulation. Still, for such speeds, we may apply a slip boundary condition on the aircraft [19] to increase the drag felt by the cargo and view the improvement on the trajectory.

1. *Slip Simulation*

The cargo payload is simulated falling from the cargo bay of the aircraft with newly applied slip conditions. The boundary conditions are the same as stated in Sec. III.B, except that the boundary condition on the aircraft surface has been changed to slip. It is now given a slip condition to allow the fluid particles to flow along the surface of the aircraft, thereby eliminating the boundary layer and the low-drag problem. Figure 32 shows this new trajectory.

Adding this new trajectory to our previous comparison of trajectories, the accuracy can be viewed. Figure 34 clearly compares all trajectories and validates the numerical slip results. Using the slip approximation has provided a trajectory that is even more swept back. In this case, there is definitely no boundary-layer-thickness problem. Again, the slip code formulation has proven a new possible, viable, and real trajectory for the high-drag experience of a cargo payload. As hypothesized, the improvement to the trajectory using a slip formulation was less pronounced here than it was in Section III for the paratrooper simulation, which is drag dominated.

2. *Fluid–Object Interaction Subcomputation Technique Simulation*

Lastly, in this section, we wish to examine the effects of the new FOIST method on this complex 3-D simulation of the falling cargo payload, a new application. First, a subcomputational domain containing only the payload was modeled. This is displayed in Fig. 35. Then, a main flow was simulated around the fully modeled cargo aircraft without the payload. Using the same specifications and conditions outlined in Sec. III.C.2, projecting from the main flowfield, the FOIST subcomputation around the payload was simulated (Fig. 36).

The results are not surprising and follow the approximations FOIST made for the jumping paratrooper simulation. For the same reasons, the drag experience of the cargo payload using the FOIST formulation is less, as expected, due to the aforementioned approximations with FOIST (Sec. III.C.2 [20]). In this case as well, the approximation is satisfactory in presenting a general estimate of the shape and size of the payload trajectory. Such an approximation can still be helpful for prognostic analyses for drop location, and it is clear from the divergence of paths that FOIST can predict the target location within a 500 m radius for drop heights up to thousands of meters.

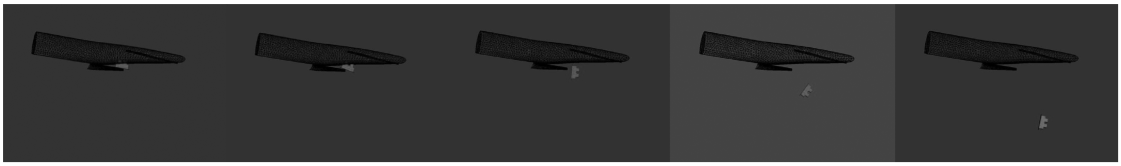

Fig. 31 Trajectory snapshots of payload drop simulation.

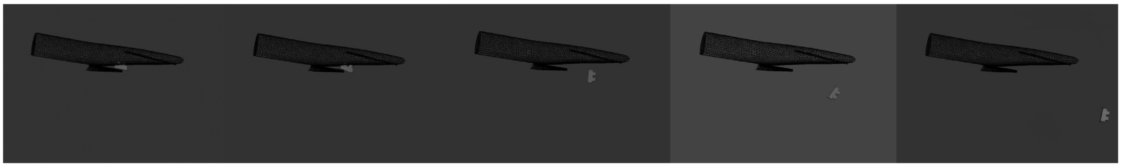

Fig. 32 Payload trajectory with slip conditions on aircraft.

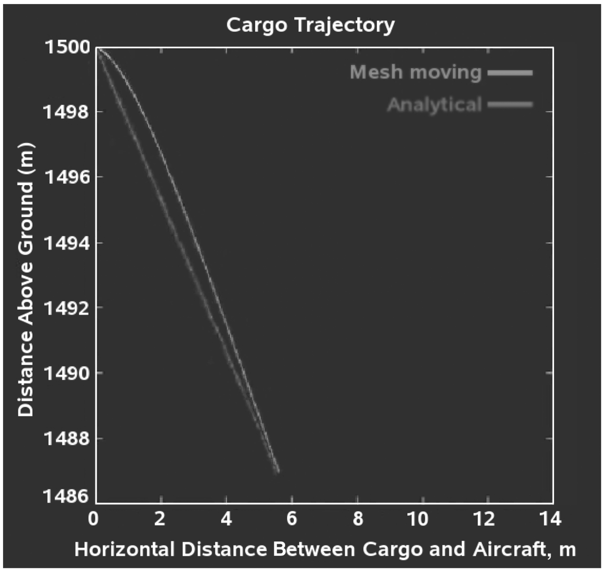

Fig. 33 Cargo payload trajectory comparison: numerical vs analytical.

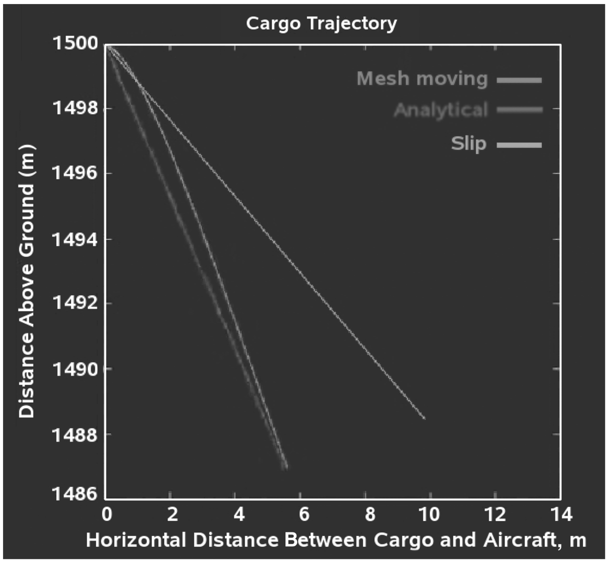

Fig. 34 Cargo payload trajectory comparison: numerical vs analytical vs slip.

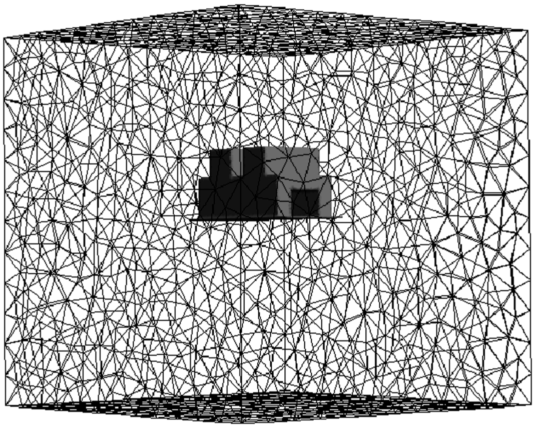

Fig. 35 Side view of colored payload in subcomputational domain for FOIST.

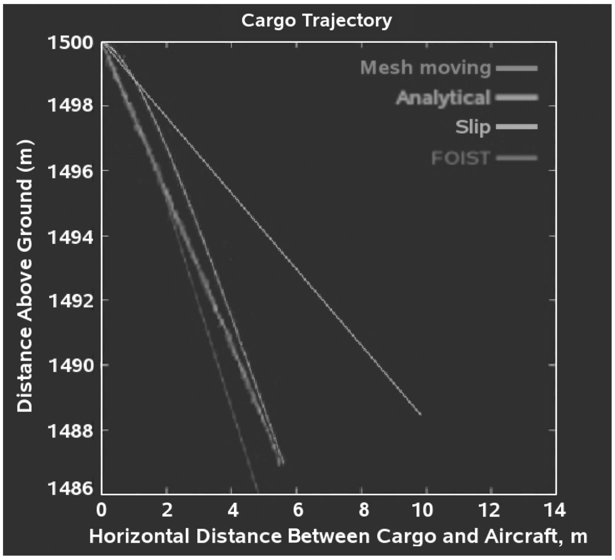

Fig. 36 Cargo payload trajectory comparison: numerical vs analytical vs slip vs FOIST.

V. Conclusions

The applications of a jumping paratrooper and dropped payload were examined. In both simulations, there was an excessively thick, unreal, numerical boundary layer. This layer reduced the drag felt by the falling object. However, the simulation of the falling cargo payload was found to be gravity dominated due to the large weight. Thus, the drag error did not affect the mesh moving simulation much, and the mesh moving simulation approximated analytical results very well. In fact, the mesh moving simulation was swept back more than the analytical solution. The slip formulation for the cargo simulation did not sweep the trajectory back as much as it did for the case of the jumping paratrooper, whose trajectory is drag dominated.

For the paratrooper, in contrast, the simulation was drag dominated. Therefore, the excessively thick, numerical boundary layer had a greater effect. Still, the mesh moving simulation produced a trajectory that could have been a possible trajectory due to the randomness of the trajectories of the actual paratrooper jumps. It produced a trajectory close to the real, physical trajectory of the least swept back paratrooper. However, due to the drag dominance, the slip formulation improved the trajectory greatly and allowed the modeled paratrooper to be swept back even more than the videotaped trajectory of the real paratrooper who was most swept back in the videotaped jumps. The greater difference between the mesh moving and the slip in the case of the paratrooper compared with cargo payload is due to the drag dominance of the paratrooper simulation.

It is important to note that the slip simulation of the paratrooper is swept back more than any real trajectory observed from the videotapes. We believe this is due to the absence of a spoiler door in the model. On an actual cargo aircraft, there is a door that swings open to a perpendicular position to the fuselage of the aircraft, such that a jumping paratrooper would see the door on his right while jumping out the left door of the aircraft (or on his left while jumping out the right door). This upstream door has holes in it that are deflected or grooved downward. Thus, air going through the door passes through the door with a downward-slanting velocity. The purpose of this is to reduce the force of the air that the paratrooper initially feels when jumping out. This door was missing from the model employed in this research. In a slip simulation with such a door, the paratrooper would not have been swept back as much by the air because of the significantly lower initial air pressure felt when jumping out of the aircraft.

FOIST simulations for both the paratrooper jumps and cargo drops approximated the general shape of the mesh moving trajectories. The moving objects in FOIST experienced much less drag force than the mesh moving simulations. More work must be done to remove some of the approximations FOIST introduces to reach a closer trajectory, though FOIST can be used as a general indicator. It definitely can be used to pinpoint the landing location of a falling payload within a 500 m radius for certain drop heights. Therefore, it is a viable alternative when computing resources for mesh moving simulations are low.

To further enhance the FOI simulations of separating objects and techniques that are presented in this research, a number of future research directions could be explored:

1) It is of extreme importance how meteorological conditions can affect the timing, quality, and safety of such jumps for falling paratroopers and payloads. Adding effects such as crosswind would be a needed and great addition as this situation arises during actual jumps and drops.

2) The addition of a spoiler door to the aircraft model will decrease the initial drag force felt by the falling payload and jumping paratrooper. We believe this is the main causal difference between the slip trajectory and the experimental trajectory. Without the spoiler door, the slip trajectory results are excellent.

Acknowledgments

The author would like to thank Richard Tapia at Rice University for his help. Thanks is also expressed to the U.S. Army Natick Soldier Center, NASA Johnson Space Center, the Alliances for Graduate Education and the Professoriate, the National Science Foundation, the NASA/Texas Space Grant Consortium, the Team for Advanced Flow Simulation and Modeling, the U.S. Air Force, the Roothbert Fund, and the Texas Association of Black Personnel in Higher Education for their support throughout the work.

References

[1] Udoewa, V., "Computational Techniques for Aerodynamic Interactions between Multiple Objects Emphasizing Paratrooper-Aircraft Separation," Ph.D. Dissertation, Mechanical Engineering and Materials Science Dept., Rice Univ., Houston, TX, 2005.

[2] Noh, W. F., "Cel: A Time-Dependent Two-Space Dimensional Coupled Eulerian-Lagrangian Code," *Methods in Computational Physics*, Vol. 3, 1964, pp. 117–179.

[3] Franck, R. M., and Lazarus, R. B., "Mixed Eulerian-Lagrangian Methods," *Methods in Computational Physics*, Vol. 3, Fundamental Methods in Hydrodynamics, Academic Press, New York, 1964, pp 47–67.

[4] Trulio, J. G., "Theory and Structure of the Afton Codes," Air Force Weapons Lab. Technical Rept. TR-66-19, Kirtland Air Force Base, NM, 1966.

[5] Hirt, C. W., Amsden, A. A., and Cook, J. L., "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics*, Vol. 14, 1974, pp. 227–253. doi:10.1016/0021-9991(74)90051-5

[6] Donea, J., Fasoli-Stella, P., and Giuliani, S., "Lagrangian and Eulerian Finite Element Techniques for Transient Fluid-Structure Interaction Problems," *Trans. 4th International Conference on Structural Mechanics in Reactor Technology*, San Francisco, CA, 1977, pp. B1/2.

[7] Belytschko, T., and Kennedy, J. M., "Computer Methods for Transient Fluid-Structure Analysis of Nuclear Reactors," *Nuclear Safety*, Vol. 26, No. 1, 1985, pp. 14–31.

[8] Belytschko, T., Kennedy, J. M., and Schoeberle, D. F., "Quasi-Eulerian Finite Element Formulation for Fluid-Structure Interaction," ASME, Paper 78-PVP-60, 1992.

[9] Hughes T. J. R., Liu, W. K., and Zimmermann, T. K., "Eulerian Finite Element Formulation for Incompressible Viscous Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 29, 1981, pp. 329–349. doi:10.1016/0045-7825(81)90049-9

[10] Aliabadi, S., and Tezduyar, T., "Space–Time Finite Element Computation of Compressible Flows Involving Moving Boundaries and Interfaces," *Computer Methods in Applied Mechanics and Engineering*, Vol. 107, No. 1–2, 1993, pp. 209–224. doi:10.1016/0045-7825(93)90176-X

[11] Behr, M., and Tezduyar, T., "Finite Element Solution Strategies for Large-Scale Flow Simulations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 112, 1994, pp. 3–24. doi:10.1016/0045-7825(94)90016-7

[12] Johnson, A. A., and Tezduyar, T., "Mesh Update Strategies in Parallel Finite Element Computations of Flow Problems with Moving Boundaries and Interfaces," *Computer Methods in Applied Mechanics and Engineering*, Vol. 119, 1994, pp. 73–94. doi:10.1016/0045-7825(94)00077-8

[13] Udoewa, V., "Mesh Generation and Update Techniques for 3D Aerodynamics Simulations," *International Journal for Numerical Methods in Fluids* (submitted for publication).

[14] Smagorinsky, J., "General Circulation Experiments with the Primitive Equations," *Monthly Weather Review*, Vol. 91, No. 3, 1963, pp. 99–165. doi:10.1175/1520-0493(1963)091&lt;0099:GCEWTP&gt;2.3.CO;2

[15] Mittal, S., "Stabilized Space–Time Finite Element Formulations for Unsteady Incompressible Flows Involving Fluid-Body Interactions," Ph.D. Dissertation, Aerospace Engineering and Mechanics Dept., Univ. of Minnesota, Minneapolis, MN, 1992.

[16] Behr, M., "Stabilized Finite Element Methods for Incompressible Flows with Emphasis on Moving Boundaries and Interfaces," Ph.D. Dissertation, Aerospace Engineering and Mechanics Dept., Univ. of Minnesota, Minneapolis, MN, 1992.

[17] Tezduyar, T. E., Aliabadi, S., Behr, M., Johnson, A., Kalro, V., and Litke, M., "Flow Simulation and High Performance Computing," *Computational Mechanics*, Vol. 18, 1996, pp. 397–412. doi:10.1007/BF00350249

[18] Johnson, A. A., and Tezduyar, T. E., "Parallel Computation of Incompressible Flows with Complex Geometries," *International Journal for Numerical Methods in Fluids*, Vol. 24, 1997, pp. 1321–1340. doi:10.1002/(SICI)1097-0363(199706)24:12&lt;1321::AID-FLD562&gt;3.0.CO;2-L

[19] Udoewa, V., "FOIST: Fluid–Object Interaction Subcomputation Technique," *Communications in Numerical Methods in Engineering* (to be published).

[20] Udoewa, V., "Slip Formulation for Jumping Paratrooper Simulations," *African Journal of Mathematics and Computer Science Research* (to be published).

[21] Stein, K., "Simulation and Modeling Techniques for Parachute Fluid-Structure Interactions," Ph.D. Dissertation, Aerospace Engineering and Mechanics Dept., Univ. of Minnesota, Minneapolis, MN, 2000.

[22] Kumar, V., "Advanced Computational Techniques for Incompressible/Compressible Fluid-Structure Interactions," Ph.D. Dissertation, Mechanical Engineering and Materials Science Dept., Rice Univ., Houston, TX, 2005.