AIAA-2003-3831

A General Simulation Method for Multiple Bodies in Proximate Flight

Robert L. Meakin
U.S. Army Aeroflightdynamics Directorate
NASA Ames Research Center
Moffett Field, CA 94035-1000

16th AIAA Computational Fluid Dynamics Conference

June 23 - 26, 2003 / Orlando, FL

A General Simulation Method for Multiple Bodies in Proximate Flight

Robert L. Meakin[†]
NASA Ames Research Center
Moffett Field, CA 94035-1000

Abstract

Methods of unsteady aerodynamic simulation for an arbitrary number of independent bodies flying in close proximity are considered. A novel method to efficiently detect collision contact points is described. A method to compute body trajectories in response to aerodynamic loads, applied loads, and inter-body collisions is also given. The physical correctness of the methods are verified by comparison to a set of analytic solutions. The methods, combined with a Navier-Stokes solver, are used to demonstrate the possibility of predicting the unsteady aerodynamics and flight trajectories of moving bodies that involve rigid-body collisions.

1. INTRODUCTION

Multiple body aerodynamics is a broad field that includes complex steady and unsteady physical processes, complex geometry, and a wide variety of applications. A multiple body application can be anything from a single body that subsequently dispenses a number of components to multiple bodies flying in close proximity, or any combination of these extremes. A multiple body classification may also be notionally relative as in the case of a rotary powered aircraft where components of a single body move relative to other components. The present paper is focused on a subset of this general class of problems -- multiple bodies in proximate flight. Methods to accurately predict the response of systems within this subset are multidisciplinary in nature and expand the spectrum of problems solvable by computational means.

A general simulation method for multiple bodies in proximate flight must allow for the possibility of bodies moving independently with six degrees of freedom. Accordingly, the possibility of inter-body collisions must also be accommodated by the approach. The Chimera[1] overset grid approach is the discretization paradigm employed here to facilitate solution of the aerodynamic fields about multiple bodies. The approach is well suited to problems that involve relative motion between bodies, or component parts of a body[2]. The present paper describes a novel method to efficiently detect contact points (if any) between a number of arbitrarily shaped bodies. The paper reviews a method of computing body trajectory subject to aerodynamic loads, applied loads, and collisions. A method of computing reaction impulses due to rigid-body collisions is also described. All of the methods are described in the context of components to a general unsteady Navier-Stokes solver.[3] The physical correctness of the methods are verified by comparing computed results and corresponding analytic solutions. A description of simulation results from the integrated flow solver and collision methods is given for a set of three-dimensional application of practical interest.

2. COMPUTATIONAL METHODS

2.1 Discretization Paradigm

The "near-body / off-body" domain partitioning method[4] is used here as the basis of problem discretization. In the approach, the near-body portion of a domain is defined to include the surface geometry of all bodies being considered and the volume of space extending a short distance away from the respective surfaces. The construction of near-body grids and associated intergrid connectivity is a classical Chimera-style decomposition of the near-body domain. It is assumed that near-body grids provide grid point distributions of sufficient density to accurately resolve the flow physics of interest (i.e., boundary-layers, vortices, etc.) without the need for refinement. This is a reasonable constraint since near-body grids are only required to extend a short distance away from body surfaces. The off-body portion of the domain is defined to encompass the near-body domain and extend out to the far-field boundaries of the problem. The off-body domain is filled with overlapping uniform Cartesian grids of variable levels of refinement. The approach facilitates grid adaptation in response to proximity of body components and/or to estimates of solution error.

2.2 Solution Method

The multiple body proximate flight simulations presented in this paper are products of the OVERFLOW-D[3] code. OVERFLOW-D is based on version 1.6au of the well known NASA OVERFLOW[5] code, but has been significantly enhanced to accommodate mov-

[†] Senior Research Scientist, U.S. Army Aeroflightdynamics Directorate (AMCOM)

ing body applications. These attributes make an ideal framework in which to introduce the rigid-body collision methods presented in this paper. The OVERFLOW-D enhancements represent in-core subroutine actuated operations and include the following capabilities.

i. On-the-fly generation of off-body grid systems.
ii. MPI[6] enabled scalable parallel computing.
iii. Automatic load balancing.
iv. Aerodynamic force and moment computations.
***v.* General 6-degrees-of-freedom model.**
***vi.* Contact detection.**
***vii.* Impact mechanics for rigid-body collisions.**
ix. Domain connectivity.
x. Solution error estimation.
xi. Grid adaptation.

The only aspects of OVERFLOW-D that are discussed in this paper are those that directly relate to enabling general simulation capability for multiple bodies in proximate flight -- 6 degrees of freedom (6-DOF),[7-9] contact detection, and impact mechanics. Pseudo-code 1 outlines the general procedure used in OVERFLOW-D to carry out such simulations. Of course, the flow equations are solved at every time-step during a simulation. The body dynamics and domain connectivity operations are also addressed at each time-step. In the pseudo-code, "body dynamics" refers to impact mechanics and 6-DOF based trajectory prediction. Prior to any contact incidents, the simulations proceed by successive execution of flow solver, 6-DOF trajectory, and domain connectivity operations. At every time step, domain connectivity is re-established to accommodate relative motion between bodies. At this time, checks for inter-body contacts are carried out. If any contacts are detected, impact reaction impulses are computed following the next flow-solver step and are incorporated into the 6-DOF trajectory prediction operation.

Discussion of the proximate flight enabling simulation components of OVERFLOW-D are discussed in the order in which they are required in the solution process. Since body motion is required before any contact or impact reactions can occur, a review of the 6-DOF trajectory prediction method is given first. This is followed by descriptions of the present contact detection and impact mechanics methods.

2.3 Trajectory Prediction Method

The general motion of a rigid body is a combination of translation and rotation. The Newtonian linear momentum principle leads directly to relations which describe the translatory motion of the body mass center as it is acted upon by aero, applied, and body forces. Rotational motion is governed by Euler's equations. The form of these equations and useful auxiliary relations are given in this section. In this and all subsequent sections of the paper, non-dimensional quantities are used exclusively. Forces are non-dimensionalized by $\rho_\infty c^2 L^2$, masses by $\rho_\infty L^3$, accelerations by L/c^2, torques by $\rho_\infty c^2 L^3$, and moments of inertia by $\rho_\infty L^5$, where ρ_∞ is the free-stream mass-density of air, c is the sonic speed, and L is a reference length. Further, with respect to nomenclature, parenthetical subscripts are always used to differentiate between inertial and body aligned frames of reference, viz. *(i)* and *(b)*. Parenthetical superscripts are used to denote coordinate component, or other descriptive information, rather than exponents.

```
For N time-steps
    do every step
        ▪ Solve flow equations
        ▪ For Moving Body Problems
            - Body dynamics
              compute reaction impulses
              compute body trajectories
            - Domain connectivity
              detect inter-body contacts
              establish domain connectivity

    do every m^th step
        ▪ Adaptive Refinement
            - Error estimation
            - Off-body re-partitioning
            - Solution transfer
            - Domain connectivity
```

Pseudo-Code 1. Solution procedure for unsteady multiple body proximate flight simulations.

Consider first the translatory motion of a rigid body. In the present context, aero forces are obtained by surface integration of the corresponding Navier-Stokes solution. Applied forces are user inputs and correspond to things like thrusters, separation mechanisms, etc. and are generally operable for only a fraction of the temporal duration of the simulation. Body forces generally refer to gravitational force, but can also include magnetic forces, etc. if such are applicable. Equation 1 is an expression of Newton's law for the position of a body mass center.

$$\hat{F}_{(i)} = \hat{F}_{(i)}^{(aero)} + \hat{F}_{(i)}^{(applied)} + \hat{F}_{(i)}^{(body)} = M\,\ddot{\hat{x}}_{(i)} \quad (1)$$

where M is the Mass and $\ddot{\vec{x}}^{(n)}$ is the acceleration of the body.

If forces are assumed to be constant over one time-step of the solver, Equation 1 can be twice integrated to yield an expression for the body mass center as a function of time.

$$\vec{x}_{(i)}^{(n+1)} = \frac{1}{2M}\vec{F}_{(i)}\,\Delta t^{2} + \vec{v}_{(i)}^{(n)}\Delta t + \vec{x}_{(i)}^{(n)} \tag{2}$$

where v is the velocity of the mass center.

The rotational motion of a rigid body is described by Euler's equations. If a body-fixed coordinate system is chosen and it is aligned with the body principal axes of inertia, Euler's equations can be expressed as follows, where the subscript *(b)* is used to denote the body-fixed frame of reference and superscripts denote components of the body principal axes.

$$\begin{aligned} T_{(b)}^{(1)} &= I_{(b)}^{(1)}\,\dot{\omega}_{(b)}^{(1)} - \left(I_{(b)}^{(2)} - I_{(b)}^{(3)}\right)\omega_{(b)}^{(2)}\omega_{(b)}^{(3)} \\ T_{(b)}^{(2)} &= I_{(b)}^{(2)}\,\dot{\omega}_{(b)}^{(2)} - \left(I_{(b)}^{(3)} - I_{(b)}^{(1)}\right)\omega_{(b)}^{(3)}\omega_{(b)}^{(1)} \\ T_{(b)}^{(3)} &= I_{(b)}^{(3)}\,\dot{\omega}_{(b)}^{(3)} - \left(I_{(b)}^{(1)} - I_{(b)}^{(2)}\right)\omega_{(b)}^{(1)}\omega_{(b)}^{(2)} \end{aligned} \tag{3}$$

where $T^{(1)}$, $T^{(2)}$, $T^{(3)}$ are the torques acting on the body, $I^{(1)}$, $I^{(2)}$, $I^{(3)}$ are the principal moments of inertia, and $\omega^{(1)}$, $\omega^{(2)}$, $\omega^{(3)}$, are the angular velocities about the respective principal axes. Equation 3 is solved numerically using a 4th-order Runge-Kutta scheme for the angular velocities at time-level *n+1*.

It is frequently necessary to convert from the body-fixed to inertial frames of reference, and vice-versa. For example, inertial frame angular velocities are maintained by the flow solver, but the body-fixed components are required for solution of Equation 3. In the present work, transformation between reference frames is accomplished using Euler parameters which are quaternions defined as follows.

$$\vec{p} = [\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4]^{\mathrm{T}}$$

$$\begin{aligned} \varepsilon_1 &= \lambda_1\sin(\theta/2) & \qquad \varepsilon_3 &= \lambda_3\sin(\theta/2) \\ \varepsilon_2 &= \lambda_2\sin(\theta/2) & \qquad \varepsilon_4 &= \cos(\theta/2) \end{aligned}$$

where $\vec{\lambda}$ is a unit vector that defines an axis of rotation and θ is the angle of rotation about this axis that will transform a point between the inertial and body-fixed frames of reference. The transformation matrix is expressed in terms of the Euler parameters as follows.

$$R(t) = 2\begin{bmatrix} \varepsilon_1\varepsilon_1 + \varepsilon_4\varepsilon_4 - \frac{1}{2}, & \varepsilon_1\varepsilon_2 - \varepsilon_3\varepsilon_4, & \varepsilon_1\varepsilon_3 + \varepsilon_2\varepsilon_4 \\ \varepsilon_2\varepsilon_1 + \varepsilon_3\varepsilon_4, & \varepsilon_2\varepsilon_2 + \varepsilon_4\varepsilon_4 - \frac{1}{2}, & \varepsilon_2\varepsilon_3 - \varepsilon_1\varepsilon_4 \\ \varepsilon_3\varepsilon_1 - \varepsilon_2\varepsilon_4, & \varepsilon_3\varepsilon_2 + \varepsilon_1\varepsilon_4, & \varepsilon_3\varepsilon_3 + \varepsilon_4\varepsilon_4 - \frac{1}{2} \end{bmatrix} \tag{4}$$

The temporal nature of R demands that the Euler parameters be updated dynamically in response to body motion. This is accomplished by solving Equation 5 for the Euler parameters at time-level *n+1*. In the present case, this is done using a 4th-order Runge-Kutta scheme subject to the constraint relation $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_4 \equiv 1$.

$$\dot{\vec{p}} = \frac{1}{2}L^{\mathrm{T}}\omega_{(b)} \tag{5}$$

The transpose of the L matrix is defined as follows.

$$L^{\mathrm{T}} = \begin{bmatrix} +\varepsilon_4 & +\varepsilon_3 & -\varepsilon_2 & +\varepsilon_1 \\ -\varepsilon_3 & +\varepsilon_4 & +\varepsilon_1 & +\varepsilon_2 \\ +\varepsilon_2 & -\varepsilon_1 & +\varepsilon_4 & +\varepsilon_3 \\ -\varepsilon_1 & -\varepsilon_2 & -\varepsilon_3 & +\varepsilon_4 \end{bmatrix}$$

As an illustration of the utilitiy of the transformation matrix, consider the wedge-shaped body given in Figure 1. Suppose that the "black vectors" *u-v-w* are always aligned with the inertial frame coordinate axes and that the "red vectors" *p-q-r* are always aligned with the principal axes of the body. Further, suppose that the body is in unconstrained rigid-body motion such that the mass center $\vec{x}_{(i)}$ and the rotational orientation of the red vectors are functions of time. The particular orientation shown in the figure corresponds to an instant in time.

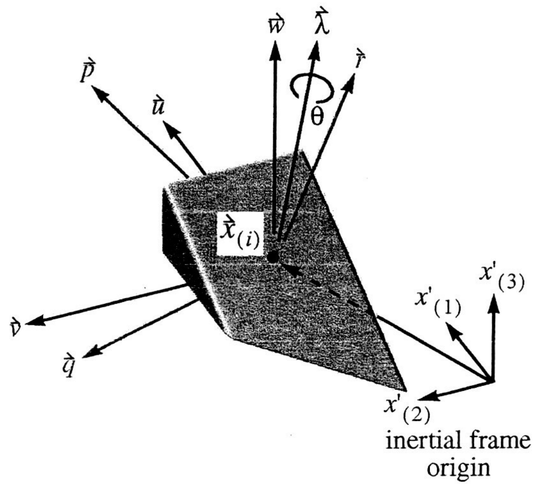

Figure 1. An irregular wedge-shaped body. Vectors aligned with the inertial frame and body principal axes.

The transformation matrix given by Equation 4 can be used to transform back-and-forth between the vectors aligned with the inertial frame and those aligned with the body's principal axes as follows.

$$\vec{p} = [R]\vec{u} \qquad \vec{q} = [R]\vec{v} \qquad \vec{r} = [R]\vec{w}$$
$$\vec{u} = [R]^{T}\vec{p} \qquad \vec{v} = [R]^{T}\vec{q} \qquad \vec{w} = [R]^{T}\vec{r}$$

The algorithm to dynamically position grid points associated with a moving body is outlined in Pseudo-Code 2. Note that required values of $\vec{x}_{(i)}(t_{n+1})$ and $R(t_{n+1})$ are obtained from solutions of Equations 2 and 5, respectively.

```
For all grid points associated with body N
    • Start with point definition at $t_0$
      $\vec{p}_{(i)}(t_0)$
    • Subtract center of mass coordinates at $t_0$
      $\vec{s} = \vec{p}_{(i)}(t_0) - \vec{x}_{(i)}(t_0)$
    • Use $R(t_{n+1})$ to update rotational
      orientation of grid-point
      $\vec{s}' = [R(t_{n+1})]\vec{s}$
    • Add center of mass coordinates at $t_{n+1}$
      $\vec{p}_{(i)}(t_{n+1}) = \vec{s}' + \vec{x}_{(i)}(t_{n+1})$
```

Pseudo-Code 2. Transform grid from initial position to dynamic position.

2.4 Contact Detection Method

Independent bodies in proximate flight have the potential of interbody collisions. The focus of the present work is on cases where the assumptions of rigid-body collisions are appropriate. The separation of a store from an aircraft, interaction between rigid pieces of debris and a launch vehicle, and simultaneous dispensing of multiple projectiles from an airborne delivery system represent examples that may be of this problem type. Such situations require two algorithmic capabilities not generally available in current Navier-Stokes and Euler solvers. Specifically, these are methods to detect contact between bodies and to compute the corresponding reaction impulses. The present section describes a novel method to efficiently detect contact points (if any) between a number of arbitrarily shaped bodies.

2.4.1 Object X-Rays

The contact detection method described here requires a list of discrete points and unit normals of the trimmed closed outer surface of each body being considered. This information is easily extracted from the near-body grid system and boundary conditions used by the flow solver. The present detection method exploits the Chimera hole-cutting technique used in OVERFLOW-D. Creating a Chimera hole involves a test to determine if a point is inside a closed "cutting" surface, or not. A contact detection test is essentially the same thing. If points on a given body are within a specified tolerance of the surface of another body, contact is indicated. If points on a given body are inside the surface of another body, penetration has occurred. Accordingly, the present method employs an efficient hole-cutting technique to detect contact. The reaction impulse computations described in the next section prevent penetration and provide post-impact angular and translational velocities based on rigid-body collision theory.

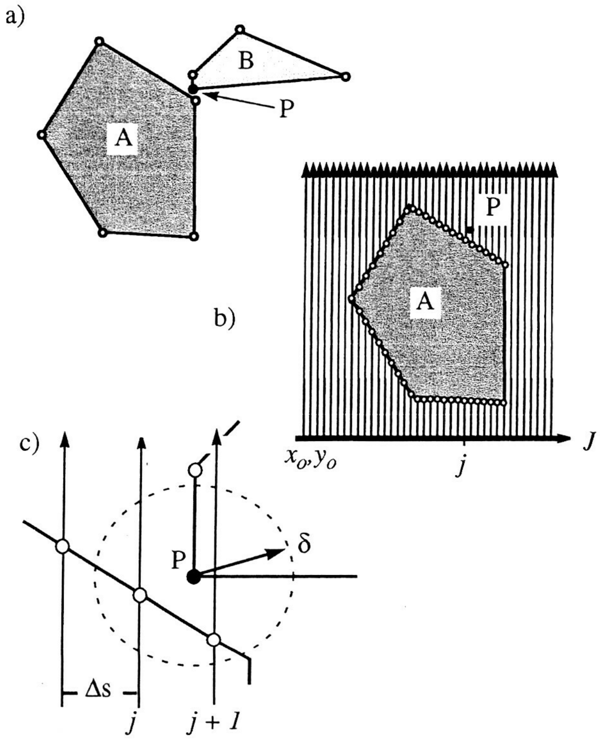

Figure 2. Object x-rays to detect contact. a) Pair of bodies. b) X-ray of body A. c) Position of vertex P of body B relative to x-ray of body A.

Consider the simple bodies shown in Figure 2a. Suppose that body A is stationary and that body B is traveling toward A. Figure 2b is an illustration of a method to detect contacts between body pairs. The set of vertical rays and corresponding points of intersection with body A are referred to as the "x-ray" of A. An x-ray is simply a compact structured list of the body/ray pierce-points. Given proper structure, such pierce-point information can be accessed extremely fast to provide a robust and efficient means of determining the proximate location of a given point to a body. A detailed description of x-ray creation, data structure, and utility is given in Reference 10 and is not repeated here. Creation of an object x-ray is a one-time expense, even for cases that involve relative motion between components.

As an example of utility, consider vertex P of body B and the x-ray of body A shown in the close-up view of Figure 2c. The uniform Cartesian structure of x-ray image-planes allow direct computation of the image-plane element (viz., j to $j+1$) that bounds vertex P. The proximate location of P is determined by comparison with the pierce-point elevations (if any) along the rays of the bounding image-plane element. X-rays are created such that rays only intersect the corresponding body zero or an even number of times. Hence, if the elevation of P is between any consecutive pair of odd/even pierce-points along a ray, it is inside the body. Since the goal of a contact detection method is to allow prevention of inter-body penetration, it is useful to establish a tolerance. Specifically, if P is within δ of body A, bodies A and B shall be said to be in contact at point P. Object x-ray spacing, Δs, controls the precision with which contact point coordinates can be determined. Object x-rays typically are created with spacing that is 5 to 10 times more fine than that used in the distribution of points on CFD surface grids.

Now consider the alternative position of body B relative to body A shown in Figure 3a. In this case, none of the vertices of body B are in contact with body A, yet body A is in obvious contact with body B. In the present method of contact detection, mutual inspection is carried out. Specifically, the vertices of body A are tested against the x-ray of body B, and the vertices of body B are tested against the x-ray of body A. Accordingly, vertex P of body A will be identified as a contact point prior to penetration via the x-ray of body B (see Figure 3b). Contact detection is guaranteed prior to penetration provided that the relative speed of approach multiplied by the flow-solver time-step is less than Δs (where Δs is the x-ray spacing).

Pseudo-code 3 outlines the mutual inspection algorithm employed in the present contact detection method. A double loop is established over the number of bodies in the domain. All surface points of a given body are tested for contact using the x-rays of all other bodies. Generally, this is a very simple test -- points not visible in the image-plane of a given x-ray don't even need to be checked. Points visible to a given x-ray image plane can be tested extremely efficiently. For every point within δ of an x-ray defined body surface, the identity of the body pairs in contact plus the coordinates and surface normal at the point of contact are entered into a list.

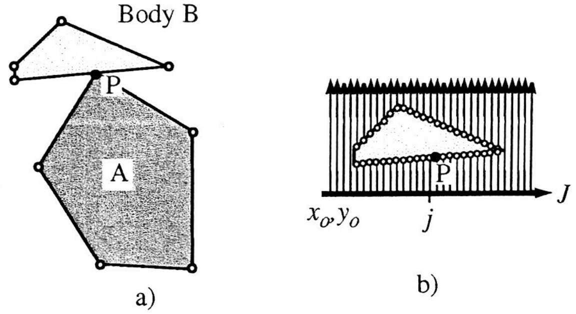

Figure 3. Object x-rays to detect contact. a) Pair of bodies. b) X-ray of body B.

```
For all bodies, N
    For all bodies, M
        IF $M \neq N$
            For all points on body M
            ▪ Test for contact w/r x-ray of
              body N
            ▪ If point is in contact, then
              add to list...
                ID of body M
                ID of body N
                coordinates of contact point
                unit surface normal
```

Pseudo-Code 3. Mutual inspection of object x-rays to identify points in contact.

2.4.2 Contact Clouds

The mutual inspection method discussed above and the use of practical values of δ in determining interbody contact, make it probable that multiple points are reported for a single contact incident. For example, mutual inspection alone produces a list of 3 points in contact between the bodies A and B shown in Figure 2. If the value of δ is increased only slightly, the body B vertex just above point P is also included in the list for a total of 4 points for one contact incident. This issue is illustrated again in Figure 4 for a pair of darts. In the case of the darts, there are two contact incidents -- one forward and one aft. As can be seen in Figures 4b and 4c, a number of points are reported as being in contact at both incident locations.

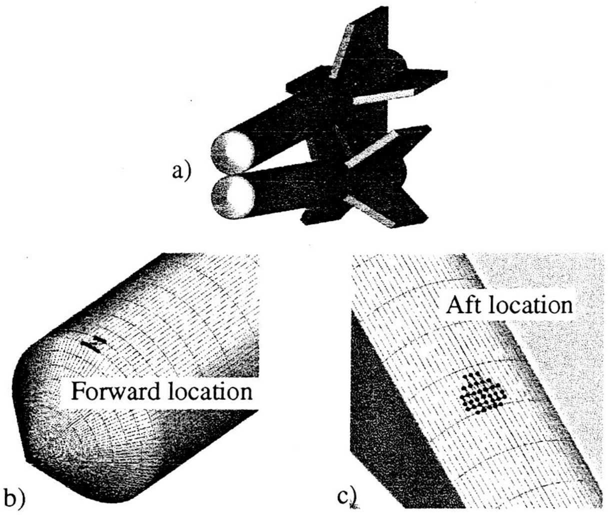

Figure 4. Pair of darts with contact points fore and aft. a) Relative position of darts. b) Cloud of points reported for forward incident. c) Cloud of points reported for aft incident.

One way to deal with multiple contacts for a single incident is to consider the set of contacts to be like simultaneous independent collisions. Methods to solve simultaneous collision problems are described in the literature.[11,12] The present approach is not to adopt such a view, largely because the surface grid resolution in CFD models can be so fine as to render a simultaneous collision reaction method complicated and inefficient. The task then remains to determine unique conditions for each contact incident. The present approach is to organize reported contact points into clouds. By definition, each cloud corresponds to a contact incident between a pair of bodies. The clouds are then "condensed" to produce the needed unique contact conditions for each incident. Pseudo-code 4 outlines the present contact cloud formation/condensation algorithm.

```
For all bodies, N
  For all bodies, M
    IF $M \neq N$
      ▪ Extract list of all NP contact points
        reported between bodies N & M
      ▪ Form Contact Clouds
        - Set cloud identity to NULL value
          for NP contact points
        - Set No. of clouds (NC) to zero
        - Begin Cloud Formation
          a) NC = NC + 1
          b) Seed NC w/ 1 unassigned point
          c) Recursively fill NC by adding un-
             assigned points that are within $\delta$
             of current members of cloud
          d) Go to step a) if there are any
             unassigned points left

      ▪ Condense Clouds
        - Loop over NC clounds
          a) count points in cloud from
             bodies N & M
          b) Designate A/B-identity of
             bodies N & M
             IF body M has more points
             in cloud than body N,
             designate
               Body M as A-identity
               Body N as B-identity
             ELSE
               make reverse designations
          c) Cloud average A-identity points for
             coordinates of contact point and
             contact-plane unit normal vector
```

Pseudo-Code 4. Cloud formation/condensation.

The algorithm begins with a dual loop over all bodies to sequentially identify pairs of bodies and all reported contact points associated with the pair. The list of reported contact points associated with the current pair of bodies are analyzed to form a discrete set of contact clouds. Each cloud is then condensed into a unique contact incident between the pair; providing the coordinates of the contact point, the A/B identity of the body pair, and the unit normal vector of the corresponding contact plane.

For a given pair of bodies during the dual loop, contact clouds are formed sequentially in four steps following cloud initialization. Cloud initialization is simply setting the number of clouds to zero and the cloud identity for each contact point in the list to a null value. The first step of cloud formation is to increment the number of clouds. The second step is to seed the current cloud with the first remaining unassigned contact point in the list. The third step is to recursively fill the current cloud with all unassigned contact points that are within a critical distance of the seed, or any points that have been added to the cloud during recursion. Finally, after formation of the current cloud, remaining unassigned contact points trigger formation of a new cloud by going to the first step of the process. When all contact points in the list are assigned a cloud identity, cloud formation is complete. The critical distance used in the third step of the cloud formation process is the maximum spacing that exists in the surface CFD grid components associated with the body pair.

Clouds are condensed sequentially with a simple loop over all contact clouds defined for the body pair. The list of contact points reported for the body pair is first evaluated. By convention, the body which contributes the most contact points to the cloud is given the A-identity. The other body, by default, is given the B-identity. This convention is exploited in the impact dynamics routines to determine whether the bodies are advancing toward each other, or are retreating. The unique contact point coordinates are taken as the cloud average of all A-identity points in the cloud. The thinking here is that the surface definition of the A-identity body is likely to be of higher resolution than that of the B-identity body. Hence, the A-identity body should provide a better description of the contact point and surface normal vector. The contact plane unit normal vector is derived from the average of all A-identity surface normals in the cloud and points, by convention, from body B toward body A. Figure 5 shows the result of cloud condensation applied to the clouds indicated in Figures 4b and 4c for the pair of darts.

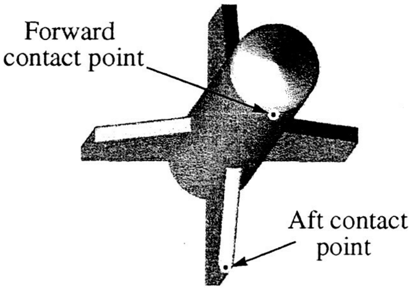

Figure 5. Cloud averaged contact points.

2.4.3 Types of Contact

The contact cloud formation and condensation algorithm described above yields a set of contact points for each incident. Point contacts, however, represent only one of three types of contact that are possible for rigid body collisions. Line and planar contacts are also possible, as illustrated in Figures 6 and 7. Cloud averaging of the line contact shown in Figure 6a would lead to the impact mechanics problem indicated in Figure 6b. If a true line contact occurs and the line segment of the contact is on the order of a few surface grid elements, cloud averaging can provide realistic (non-penetrating) results. However, if the contact line segment is relatively large and the contact plane unit normal vector is not aligned with the vector connecting the object mass centers (as in Figure 6a), reaction impulsive torques can initiate penetration.

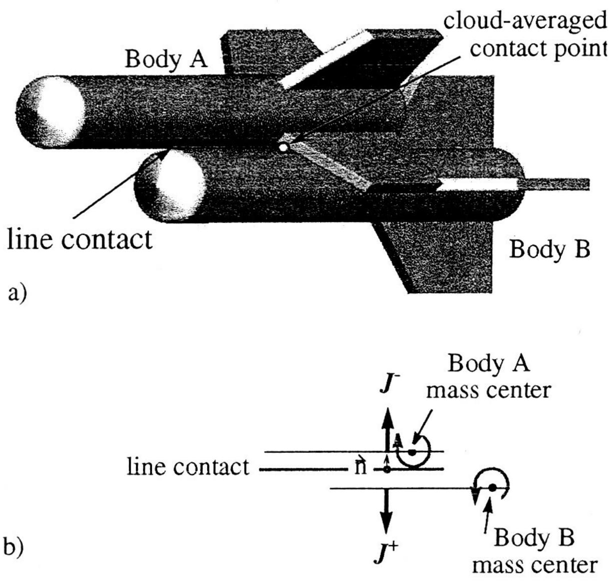

Figure 6. Line contact. a) Bodies A and B. b) Reaction impulses at cloud-averaged contact point.

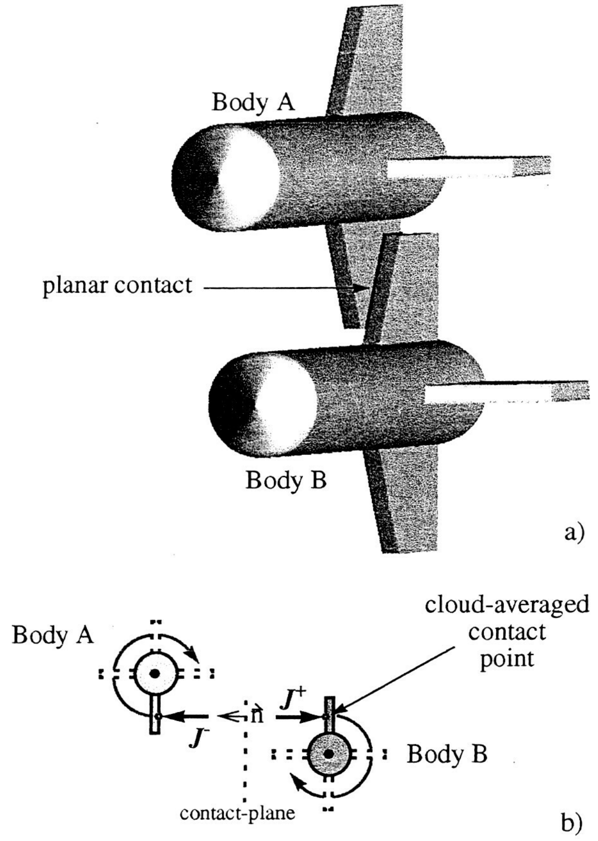

Figure 7. Planar contact. a) Bodies A and B. b) Reaction impulses at cloud-averaged contact point.

A similar situation exists for planar contacts. Although cloud averaging can yield realistic reaction impulses, as in the case shown in Figure 7, it is not sufficient in general. Plans to generalize the present contact cloud method are in progress such that for each contact incident, one point for point contacts; two points to define a contact line segment for line contacts; and three points to define a contact triangle for planar contacts are determined. Accordingly, cloud formation and condensation operations that yield multiple contact points for a given incident will initiate simultaneous impact mechanics methods to compute reaction impulses. Simultaneous impact mechanics methods for a small number of points can be carried out at minimal expense.

2.5 Impact Mechanics Method

The methods adopted in this work for dealing with collisions between bodies in proximate flight are based on rigid-body collision theory. This implies that the contact area for any incident is small and that the period of contact is negligible. These assumptions allow the impact mechanics to be described by algebraic relations between velocity changes and reaction impulses.[13]

The relative state between body pairs that give rise to contact detection is not unique. The bodies may be retreating, at rest, or advancing at the point of contact at the time impact (t_{imp}) is detected. The first two states are benign. Only the third condition initiates reaction impulses. Accordingly, it is necessary to evaluate the state of the body pair for all contact incidents. If

$$\vec{r} = \vec{p}_{(i)}(t_{imp}) - \vec{x}_{(i)}(t_{imp})$$

and defines the position of the contact point relative to the body mass center at the time of impact, then the velocity at the point of impact for the body in question is

$$\vec{u} = \vec{v}_{(i)} + \vec{\omega}_{(i)} \times \vec{r}$$

The relative velocity at the point of impact and normal to the contact plane between the body pair is simply

$$u_{rel} = (\vec{u}_a - \vec{u}_b) \cdot \vec{n}_{(i)}(t_{imp})$$

Recall from Section 2.4.2 that, by convention, the unit normal vector at the point of impact, $\vec{n}_{(i)}(t_{imp})$, points from body B to body A. Accordingly, if $u_{rel} \geq 0$, the bodies are either retreating or at rest. In either case, computation of reaction impulses is not necessary. However, if $u_{rel} < 0$, the bodies are advancing and contact is imminent, making the computation of reaction impulses a necessity.

An impulse $\vec{J}$ applied to a body results in changes in translational and angular velocities described by Equations 6 and 7, where M is the body mass, $\vec{r}$ is the position of the contact point relative to the body mass center at the time of impact, and $I^{-1}(t_{imp})$ the inverse of the body inertia tensor at impact.

$$\Delta\vec{v}_{(i)} = \frac{1}{M}\vec{J} \qquad (6)$$

$$\Delta\vec{\omega}_{(i)} = I^{-1}(t_{imp})\ \vec{r} \times \vec{J} \qquad (7)$$

If frictional effects are ignored during impact, the reaction impulse is in the direction normal to the contact plane, viz.

$$\vec{J} = j\ \vec{n}_{(i)}(t_{imp}) \qquad (8)$$

where, j is the impulse magnitude. By convention, body A is subject to an impulse of $+j\ \vec{n}_{(i)}(t_{imp})$ and B is subject to an equal but opposite impulse of $-j\ \vec{n}_{(i)}(t_{imp})$.

The reaction impulse magnitude is given as Equation 9 (see Ref. 14 for derivation).

$$j = \frac{-(1-\varepsilon)u_{rel}}{\left[\frac{1}{M_a} + \frac{1}{M_b} + T_a + T_b\right]} \qquad (9)$$

Here, ε is the coefficient of restitution and M_a and M_b are masses of the respective bodies. T_a and T_b represent the contribution of angular momentum to the reaction impulse magnitude and are defined as follows.

$$T_a = \vec{n}_{(i)}(t_{imp}) \cdot (I_a^{-1}(t_{imp})(\vec{r}_a \times \vec{n}_{(i)}(t_{imp}))) \times \vec{r}_a \qquad (10)$$

$$T_b = \vec{n}_{(i)}(t_{imp}) \cdot (I_b^{-1}(t_{imp})(\vec{r}_b \times \vec{n}_{(i)}(t_{imp}))) \times \vec{r}_b \qquad (11)$$

The coefficient of restitution (ε) is bounded between 0 (perfectly plastic) and 1 (perfectly elastic). It is useful to note that the temporal state of the inertia tensor for a given body can be expressed as $I(t) = R(t)\ I_{body}\ R(t)^T$, where $R(t)$ is the transformation matrix defined by Equation 4 and I_{body} is the inertia tensor specified in body coordinates (constant over the simulation).

In order to exploit the relations given by Equations 6 through 11, the state of the body pair at impact is required. As indicated in Pseudo-code 1, the pre-impact trajectories of bodies in proximate flight are determined using a conventional 6-DOF model as described in Section 2.3. Accordingly, "at impact" conditions are computed the time-step immediately prior to contact detection. Specifically, the mass center ($\vec{x}_{(i)}(t_{imp})$) and transformation matrix ($R(t_{imp})$) are available from the 6-DOF computation. The subsequent "domain connectivity" step yields the A/B identity for the incident body pair, the unit normal vector ($\vec{n}_{(i)}(t_{imp})$), and the point of contact coordinates ($\vec{p}_{(i)}(t_{imp})$). From this information, $\vec{r}_a$, $\vec{r}_b$, T_a, T_b, u_{rel}, and j can be computed directly. Then, Equations 6 and 7 can be used to compute the post-impact changes in translational and angular velocity for the respective bodies. Specifically,

Body A:

$$\Delta\vec{v}_{(i)} = +\frac{1}{M_a}\vec{J} \qquad \Delta\vec{\omega}_{(i)} = +I_a^{-1}(t_{imp})\,\vec{r}_a \times \vec{J}$$

Body B:

$$\Delta\vec{v}_{(i)} = -\frac{1}{M_b}\vec{J} \qquad \Delta\vec{\omega}_{(i)} = -I_b^{-1}(t_{imp})\,\vec{r}_b \times \vec{J}$$

3. RESULTS

3.1 Method Verification

The 6-DOF model and impact mechanics method presented in this paper are verified by simulating a set of simple problems for which analytic solutions are known. The first two problems focus exclusively on the 6-DOF model. The final problems in the set focus primarily on the impact mechanics, but also exercise the 6-DOF model.

3.1.1 Sphere Drop

A sphere at standard atmospheric conditions having a radius r and mass M at an initial altitude of z_0 is dropped at time t_0. The net forces acting on the sphere after release vary as a function of time. The forces include the weight F_w of the sphere, a buoyant force F_b, and a drag force F_d. If the sphere is very small (e.g., an aerosol particle, or a particle of sediment in water), then F_b is significant. However, for the present case, consider a sphere of sufficient size and mass as to make F_b negligible. Accordingly, the net forces acting on the sphere can be expressed as a function of time as follows.

$$F(t) = -Mg + \frac{1}{2}\rho\pi r^2 C_D V(t)^2$$

where C_D is the coefficient of aerodynamic drag and V is the velocity magnitude of the sphere. At t_0 the sphere is at rest and the drag force is zero. On release, the drag force increases asymptotically until it equals the weight of the sphere, limiting the rate-of-decent to a "terminal velocity."

$$V_{term} = \sqrt{(2Mg)/(\rho\pi r^2 C_D)}$$

An analytic expression for the sphere velocity as a function of elevation can be derived in terms of the terminal velocity.

$$V(z) = \sqrt{V_{term}^2\left(1 - e^{2gz/v_{term}^2}\right)}$$

Given the following conditions, the 6-DOF model described in Section 2.3 predicts the velocity versus elevation correlations shown in Figure 8.

$$r = 1/2 \qquad z(t_0) = 0 \qquad g = 2.67\times10^{-5}$$
$$M = 100 \qquad \rho = 1 \qquad C_D = 1/2$$

The analytic solution is shown in black. Simulation results using 10 and 100 time-steps to resolve 1,000 sphere diameter's distance of travel are shown in red and blue, respectively. Clearly, the 6-DOF solutions converge to the analytic solution, verifying the implementation of the methods to solve Equation 2 for translatory motion.

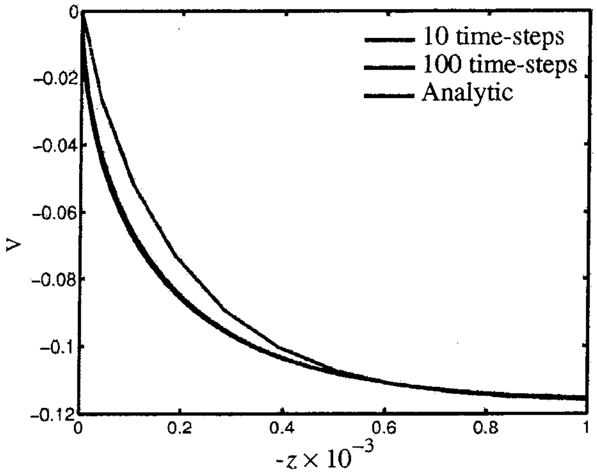

Figure 8. Sphere velocity vs elevation.

3.1.2 Tumbling Spinning Cylinder

Consider a rigid cylindrical body of mass M that has the following inertial properties.

$$I = \begin{bmatrix} I_{(b)}^{(1)} & 0 & 0 \\ 0 & I_{(b)}^{(2)} & 0 \\ 0 & 0 & I_{(b)}^{(3)} \end{bmatrix}, \text{ where}$$

$$I_{(b)}^{(1)} = I_{(b)}^{(2)} \text{ and } I_{(b)}^{(1)} - I_{(b)}^{(3)} = I_{(b)}^{(2)} - I_{(b)}^{(3)} = \alpha I_{(b)}^{(1)}$$

In the absence of applied forces and viscous effects, the body will fall due to the effects of gravity and tumble according to Euler's equations of motion. These equations de-couple when the rotation axes are aligned with any of the body principal axes. For the conditions noted above, an exact solution to Euler's equations exists[15], viz.

$$\omega_{(b)}^{(1)} = a\cos(\lambda t) \qquad \omega_{(b)}^{(2)} = b\sin(\lambda t) \qquad \omega_{(b)}^{(3)} = c$$

Suppose that $\lambda = \alpha c$, where $\alpha = c = \frac{1}{2}$, $a = 1$, $b = -1$, and that the principal moments of inertia for the cylinder, $I_{(b)}^{(n)}$, are 1, 1, and 1/4, for n = 1,2,3. At $t = 0$, the initial position of the body mass center is [0, 0, 0] and the initial angular components of velocity are [1, 0, 1/2]. The analytic solution for one period of the body's angular velocity is shown in Figure 9. The analytic solution for the translational motion of the body mass center is shown in Figure 10 over the same interval.

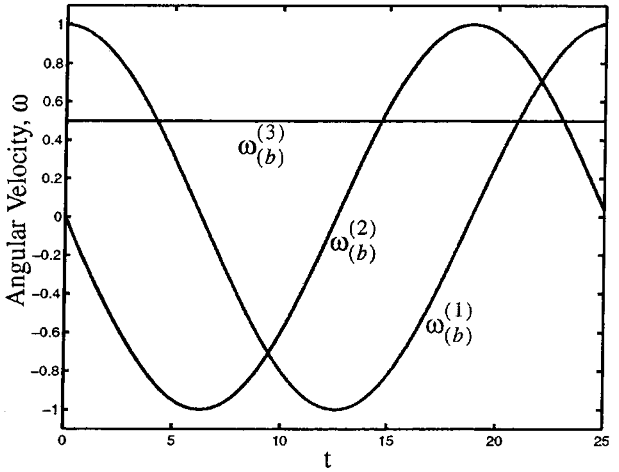

Figure 9. Analytic solution for the angular components of velocity.

As with the sphere-drop case, 6-DOF simulation results for the translational motion of this problem are in excellent agreement with the analytic solution. The 6-DOF simulation was carried out for a total of 36 periods of the rotational behavior indicated in Figure 9. Even after 36 periods of rotation, differences between the analytic solution and 6-DOF results using as few as 250 time-steps per period cannot be seen in a simple line graph. Figure 11 shows the error in simulation results at the end of 3.6 periods of oscillation versus the number of steps used per period. The error plotted in the figure is defined as

$$\omega_{err} = \left|\omega(t_{final})\big|_{simulation} - \omega(t_{final})\big|_{exact}\right|$$

Doubling of the number of time-steps per period reduces the error by a factor of about 16. This is consistent with a 4th-order time integration scheme employed.

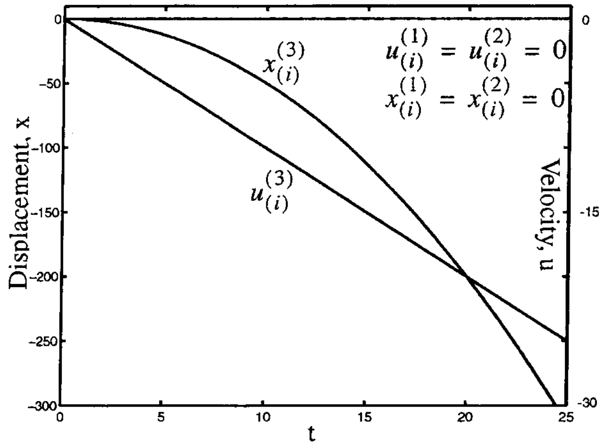

Figure 10. Analytic solution for the body center of mass trajectory.

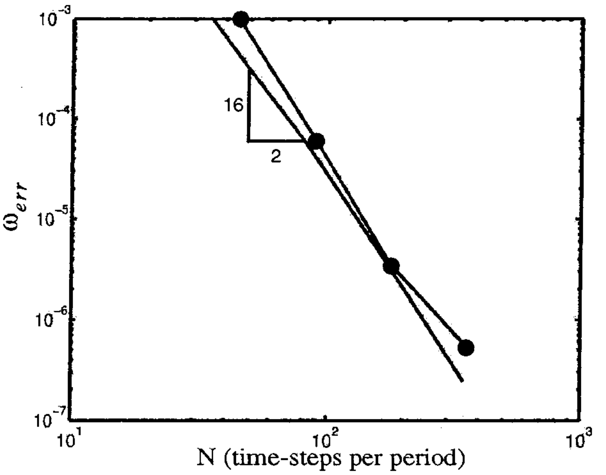

Figure 11. Error in 6-DOF solution at the end of 3.6 periods of rotation versus number ot time-steps per period.

3.1.3 Sphere-Sphere Impact

Consider a pair of spheres of radius r_A and r_B, respectively. The spheres are positioned as shown in Figure 12. Sphere A travels toward sphere B in a direction that is aligned with the positive $x_{(i)}^{(1)}$-axis, but with an eccentricity of e. The spheres are sufficiently smooth such that frictional forces between the bodies during collision can be ignored. If the initial velocity of sphere A is V_0, sphere B is initially at rest, the initial separation between the centers of mass is l, and the respective masses are M_A and M_B, the time of impact and post-impact trajectories and speeds can be determined analytically.

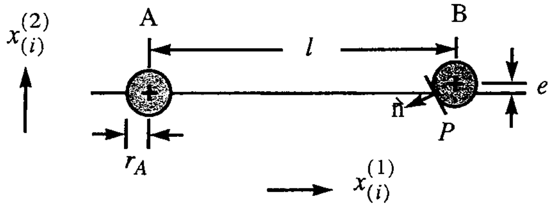

Figure 12. Sphere-sphere impact. Sphere A travels toward sphere B, which is static. Impact occurs at *P*.

The time to impact is simply the initial distance between the spheres divided by the initial velocity of sphere A, viz. $t_{imp} = d/V_0$, or

$$t_{imp} = \left|-l + \sqrt{(r_A + r_B)^2 - e^2}\right| / V_0$$

Now, taking time relative to t_{imp}, and assuming perfectly elastic collisions, the post-impact trajectories can be expressed as follows.

$$\vec{x}(t) = \vec{v}t$$

Sphere A velocity components:

$$U^{(1)}_{(i)} = V_0 - 2\frac{M_B}{(M_A + M_B)}\frac{((r_A + r_B)^2 + e^2)}{(r_A + r_B)^2}V_0$$

$$U^{(2)}_{(i)} = -2e\frac{M_B}{(M_A + M_B)}\frac{\sqrt{(r_A + r_B)^2 + e^2}}{(r_A + r_B)^2}V_0$$

Sphere B velocity components:

$$V^{(1)}_{(i)} = 2\frac{M_A}{(M_A + M_B)}\frac{(r_A + r_B)^2 + e^2}{(r_A + r_B)^2}V_0$$

$$V^{(2)}_{(i)} = 2e\frac{M_A}{(M_A + M_B)}\frac{\sqrt{(r_A + r_B)^2 + e^2}}{(r_A + r_B)^2}V_0$$

The contact detection rule for the present verification case is invoked when the distance between the sphere mass centers is less than $r_A + r_B + V_0\Delta t$. Accordingly, the error in the predicted value of t_{imp} is always less than Δt. In practice, the precision of the contact detection method ultimately controls the accuracy of trajectory predictions in proximate flight simulations that have actual inter-body collisions. This is so because the contact point location and surface normal for each incident are the product of the detection process. The post-impact trajectory prediction method itself is quite accurate. Although a range of time-step sizes have been evaluated, results presented here are based on a time-step size of 0.001, which is comparable to what is typically used in applications of OVERFLOW-D. This time-step size predicts the value of t_{imp} to within 5×10^{-4}.

Table 1 verifies the accuracy of the present impact mechanics method in predicting post-impact velocities for a variety of conditions. The sphere post-impact velocities are computed as functions of eccentricity, radius, and mass. Errors are computed as the absolute value of differences between the analytic solution given above and the present simulation results. Only the maximum error in the post-impact components of velocity are reported. Figure 13 shows the predicted sphere trajectories for one of the cases included in Table 1. Differences between the analytic trajectory and the plotted prediction are less than the line thickness used in the figure.

Variation in eccentricity

e	$U^{(1)}_{(i)}$	$U^{(2)}_{(i)}$	$V^{(1)}_{(i)}$	$V^{(2)}_{(i)}$	$\text{Error}_{\text{max}}$
0	0.000000	0.000000	1.000000	0.000000	0
1/4	0.015614	-0.123978	0.984385	0.123978	4×10^{-5}
1/2	0.062469	-0.242005	0.937530	0.242005	5×10^{-5}

constants: $r_A = r_B = 1 \quad M_A = M_B = 1$

Variation in r_B

r_B	$U^{(1)}_{(i)}$	$U^{(2)}_{(i)}$	$V^{(1)}_{(i)}$	$V^{(2)}_{(i)}$	$\text{Error}_{\text{max}}$
1.0	0.062469	-0.242005	0.937530	0.242005	5×10^{-5}
1.5	0.039984	-0.195921	0.960015	0.195921	3×10^{-5}
2.0	0.027760	-0.164285	0.972239	0.164285	5×10^{-5}

constants: $e = 1/2 \quad r_A = 1 \quad M_A = M_B = 1$

Variation in M_B

M_B	$U^{(1)}_{(i)}$	$U^{(2)}_{(i)}$	$V^{(1)}_{(i)}$	$V^{(2)}_{(i)}$	$\text{Error}_{\text{max}}$
1.0	0.062469	-0.242005	0.937530	0.242005	5×10^{-5}
1.5	-0.125036	-0.290407	0.750024	0.193604	6×10^{-5}
2.0	-0.250040	-0.322674	0.625020	0.161337	7×10^{-5}

constants: $e = 1/2 \quad r_A = r_B = 1 \quad M_A = 1$

Table 1. Sphere-Sphere verification results for post-impact trajectory prediction.

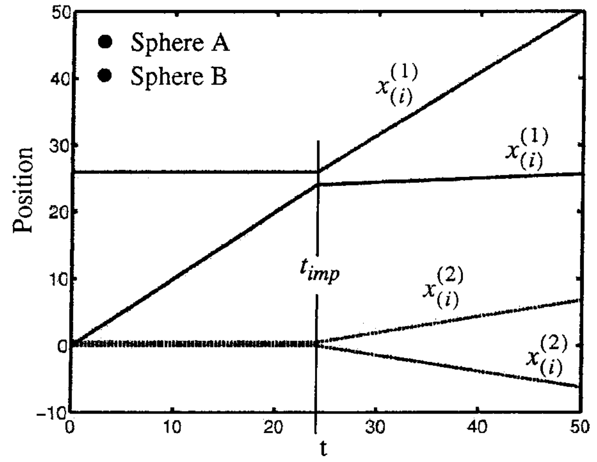

Figure 13. Coordinates of sphere centers versus time. $r_A = r_B = 1, M_A = M_B = 1$, and $e = 1/2$.

3.1.4 Sphere-Rod Impact

Consider a sphere of radius r_S and mass M_S that strikes the end of the rod as shown in Figure 14. The rod is initially at rest and is characterized by radius r_R, length l, mass M_R, and moment of inertia I_R. The coefficient of restitution between the sphere and rod is ε and the initial separation between the respective mass centers is $l/2$.

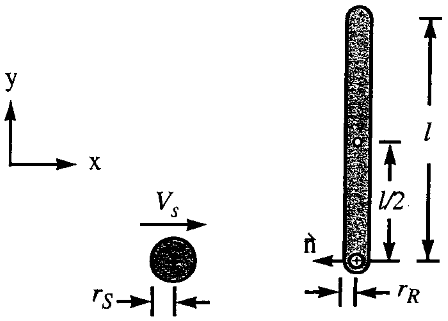

Figure 14. Sphere impacts the "positive" pole of rod.

If the rod is unconstrained and all aerodynamic forces are ignored, then the motion of the rod is governed entirely by the reaction impulse delivered by the sphere. Conversely, the motion of the sphere is governed by it's initial velocity, V_S, and the reaction impulse delivered by the rod. Analytic inertial frame expressions for the post impact translational velocity of the sphere and the translational and angular velocity of the rod are given as follows.

Sphere velocity components:

$$U^{(1)}_{(i)} = \left[1 - (1+\varepsilon)\Big/\left[1 + \frac{M_S}{M_R} + \frac{M_S\, l^2}{4I_R}\right]\right]V_S \qquad U^{(2)}_{(i)} = 0$$

Rod velocity components:

$$V^{(1)}_{(i)} = (1+\varepsilon)V_S\Big/\left[\frac{M_R}{M_S} + 1 + \frac{M_R\, l^2}{4I_R}\right] \qquad V^{(2)}_{(i)} = 0$$

Rod angular velocity:

$$V^{(3)}_{(i)} = l\,(1+\varepsilon)V_S\Big/\left[\frac{2I_R}{M_S} + \frac{2I_R}{M_R} + \frac{l^2}{2}\right]$$

Analytic and predicted values of these quantities are given in Table 2 for specified values of M_S, M_R, l, I_R, and V_S. Results are given for coefficient of restitution values of ε=1.0 (perfectly elastic) and ε=0.8. As indicated in the table, the errors are near machine zero and provide verification of the present implementation of the methods described in Section 2.5.

Sphere Velocity and Rod Angular Velocity

ε = 1.0

	Exact	Computed	Error
$U^{(1)}_{(i)}$	0	0.00000000000	0
$V^{(1)}_{(i)}$	5/4	1.25000000000	0
$\omega^{(3)}_R$	24/4	6.24999975165	3×10^{-7}

ε = 0.8

	Exact	Computed	Error
$U^{(1)}_{(i)}$	1/2	0.49999997019	1×10^{-7}
$V^{(1)}_{(i)}$	9/8	1.12500000745	1×10^{-7}
$\omega^{(3)}_R$	45/8	5.62499981374	2×10^{-7}

constants: $r_S = r_R = 1 \quad M_S = 2 \quad M_R = 8$
$l = 1.2 \quad \varepsilon = 0.8 \quad I_R = 0.96$

Table 2. Sphere-Rod verification results for post-impact trajectory prediction.

Trajectories of the sphere center of mass and rod positive pole are given in Figure 15. Figure 15a shows simulation trajectories based on a coefficient of restitution of ε=0.8. Figure 15b provides a comparison between the prediction of the rod's positive pole trajectory for values of ε of 0.8 (red) and 1.0 (black). The sphere impact on the rod propels the rod in a counter clock-wise motion. Undamped by applied loads or aerodynamic drag, the rod spins indefinitely at the post impact angular velocity and travels downstream in response to the momentum transferred from the sphere during the collision. The predicted time history shown in Figure 15 matches the analytic solution with no discernible phase error and a lag equal to the temporal error associated with impact detection. In the present example, the observed impact detection lag is 6×10^{-9}, but in general is only guaranteed to be less than Δt (0.001 in this case). Accordingly, there are no discernible differences between the computed and analytic time-histories shown.

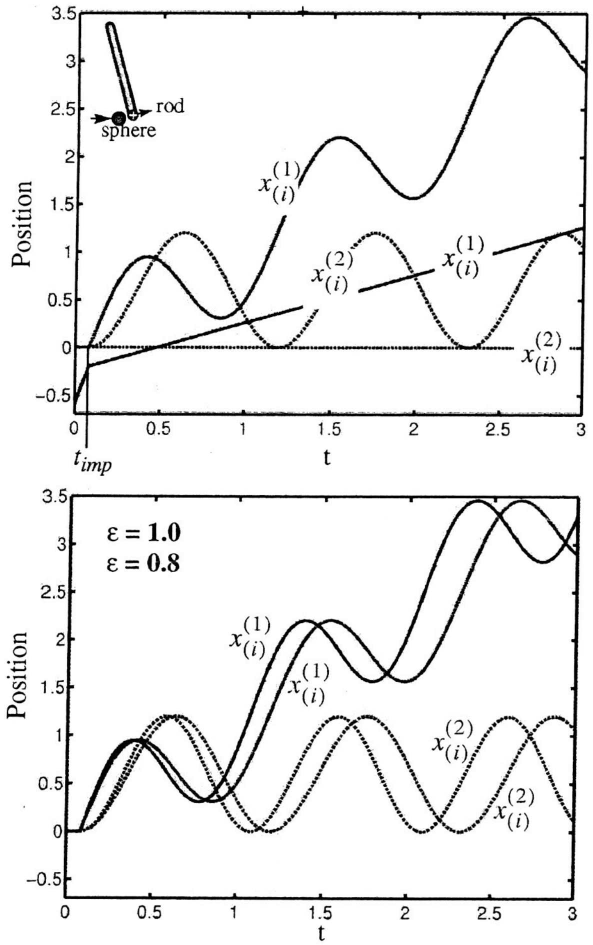

Figure 15. Time-histories. a) ε = 0.8 -- sphere mass center and rod positive pole positions versus time. b) Rod positive pole position versus time for ε = 0.8 (red) and 1.0 (black).

3.2 Applications

The 6-DOF model, contact detection method, and impact mechanics method have been integrated into the general Navier-Stokes solver noted in Section 2.2 and applied to a set of three-dimensional problems of practical interest. Results of some of these applications are presented here.

3.2.1 Darts in Proximate Flight

The simultaneous release of multiple bodies from a parent delivery system can easily involve body-body and/or parent-body collisions. Consider the idealization of this problem shown in Figure 16 that involves a pair of perfectly elastic (ε = 1.0) darts. Suppose that dart A is positioned beneath dart B at zero angle-of-attack and fixed in space for all time. If dart B is initially in the nose-down position indicated in the figure and is free to move subject to aerodynamic and gravitational forces, collision is imminent. In this discussion, dart A is referred to as the "stationary dart." Discussion of loads and movement relate exclusively to dart B.

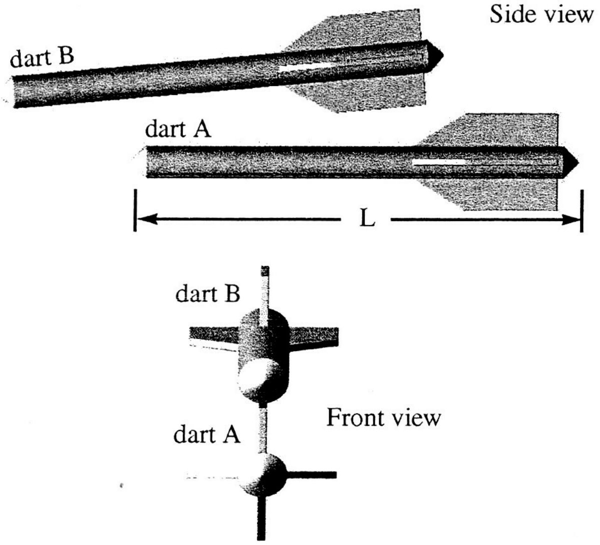

Figure 16. Initial orientation of darts in proximate flight.

The initial conditions for this problem are given in Table 3 and illustrated in Figure 17a. The figure shows the $x^{(2)}_{(i)} = 0$ plane colored by Mach number. The stationary dart is shown to be entirely within the canopy of the dart B bow-shock. Conversely, the bow-shock of the stationary dart pushes on dart B to produce a nose-up pitching moment. At t_0, dart B is released and moves quickly aft and down primarily in response to drag and negative lift. The entire temporal duration of the simulation is not large enough for gravitational effects, or even the initial nose-up pitching moment to significantly affect the dart trajectory.

Case Conditions

$$M_\infty = 1.86 \quad R_e = 5 \times 10^6$$

L = 0.5 ft	$I^{(1)}_{(b)} = 4.37 \times 10^{-7}$ Lb-ft/sec^2
c = 1098 fps	$I^{(2)}_{(b)} = 5.58 \times 10^{-5}$
M = 0.002683 Lb-sec^2/ft	$I^{(3)}_{(b)} = I^{(2)}_{(b)}$
h = 10,000 ft	

Table 3. Darts in proximate flight application.

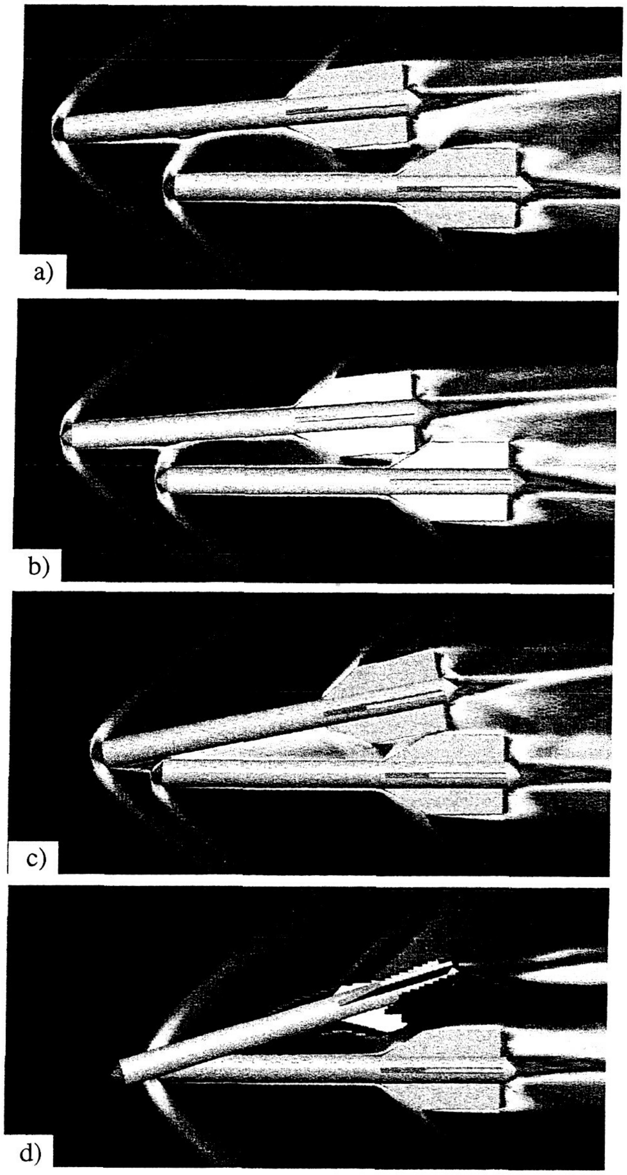

Figure 17. Darts in proximate flight. Mach contour plots at $x_{(i)}^{(2)} = 0$ plane. a) Initial position and flow conditions: t = 0. b) First contact: t = 8.75. c) Simultaneous fore and aft contacts: t = 22.5. d) t = 25.

At $t = 8.75$, the lower fin of dart B strikes the upper fin of the stationary dart (see Figure 17b). The impact of this collision and the negative lift combine to reverse the sense of the pitching moment acting on the dart. As a result of the continued downstream movement of the dart and now the post-impact nose-down angular velocity, dart B experiences nearly simultaneous collisions fore and aft at $t = 22.5$ (see Figure 17c). The result of these secondary and tertiary impacts prompt a more violent response than the initial impact. Here, relatively strong roll, pitch, and yaw moments produce significant out-of-plane movement and angular velocities (see Figure 17d).

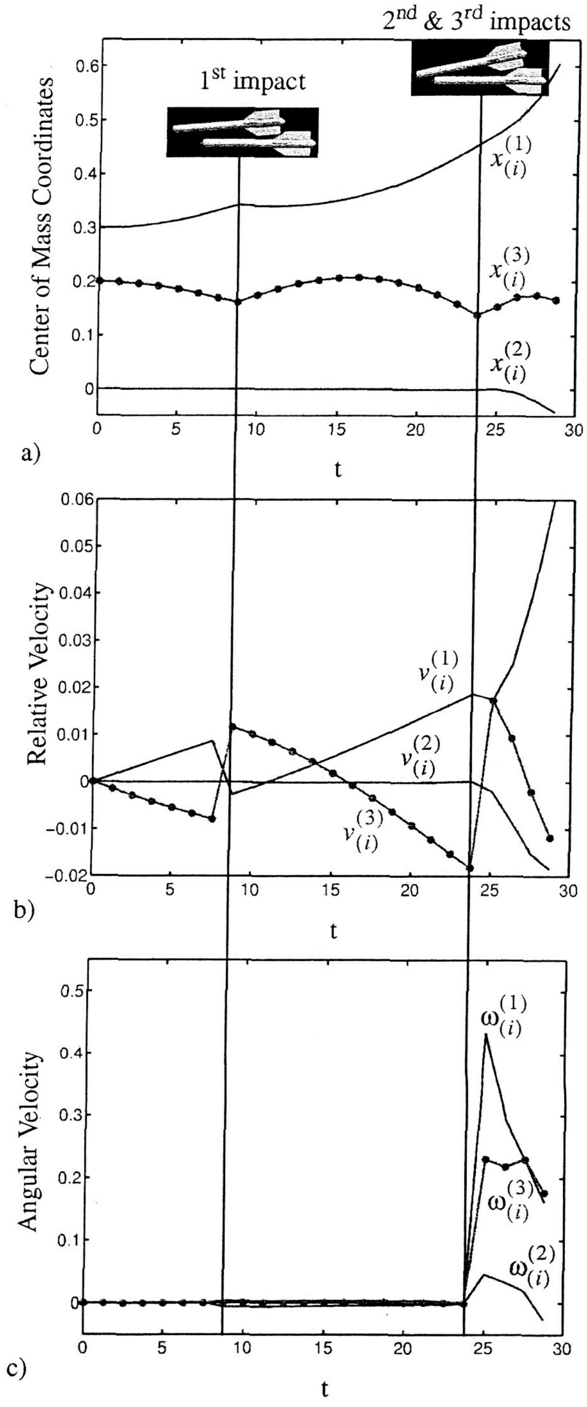

Figure 18. Motion of dart B. a) Position of mass center. b) Relative velocity. c) Angular velocity.

Time histories of the dart B center of mass position, relative velocity, and angular velocity versus time are given in Figure 18. Results of each impact are evident in this figure. Between impacts, the dart relative velocity varies linearly indicating nearly constant acceleration due to drag, lift, and side-force. This is true for the angular velocities as well.

3.2.2 Debris in a Supersonic Flow-Field

Launch vehicles of today can be characterized by complex geometry and most have many irregular protuberances. Given the extreme flow environments that such vehicles pass through during launch, concern regarding possible vehicle damage from debris is always a concern. Debris can be anything from airborne objects at altitude to pieces of the vehicle itself that break-way during ascent. The present rigid-body impact methods presented in this work are not appropriate for high-speed collisions. However, the trajectory of break-away debris can be significantly influenced by early low relative velocity collisions. The present methods are appropriate for such cases.

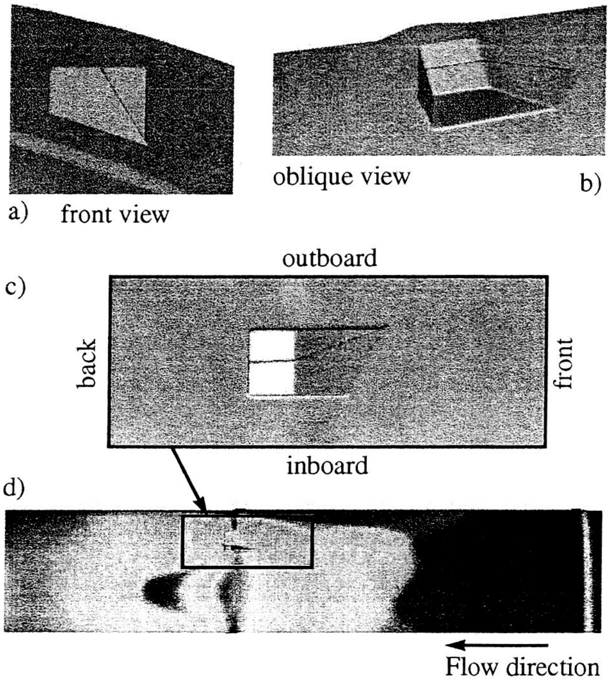

Figure 19. Debris and flow environment definition. a) Front view of geometry. b) Oblique view of geometry. c) Top view of geometry. d) Surface Cp distribution in the vicinity of debris initial position.

Consider the geometry and flow environment illustrated in Figure 19. A section of a launch vehicle is shown in the vicinity of a ramp-shaped protuberance (see Figures 20a-c). At the supersonic conditions indicated in Figure 20d via surface C_p distribution, a complex system of interfering shocks induce boundary layer separation just upstream of the ramp. The separation line corresponds closely with sharp changes in color from dark blue to cyan and green. Suppose that part, or all, of the ramp-shaped protuberance breaks-away during these ascent conditions. The methods presented in this paper provide a way to compute the trajectory of the debris subject to any initial applied loads (if any), aerodynamic loads, and collisions.

As a demonstration of the present methods, consider the three alternative pieces of debris shown in Figure 20. The pieces of debris correspond to the upper outboard aft corner of the ramp; a slice of the outboard side of the ramp; and the entire ramp itself. Results for each of these hypothetical pieces of debris are discussed as each are allowed to break-away from the rest of the launch vehicle entirely by aerodynamic forces (i.e., no separation loads are applied).

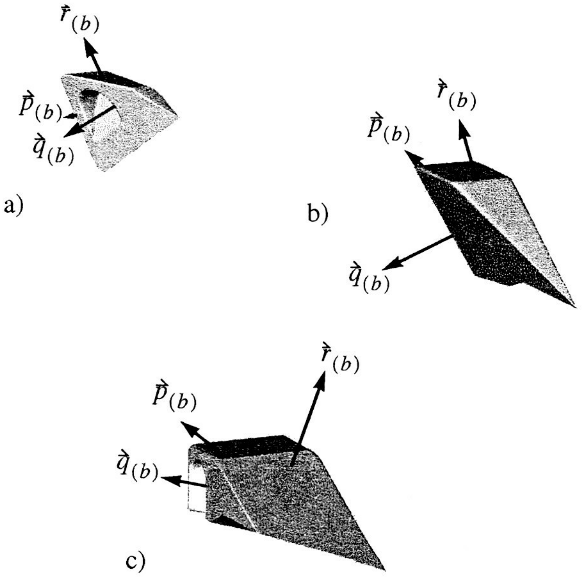

Figure 20. Alternative pieces of debris. Geometry definitions and principal axes of inertia. a) Upper corner of outboard back of ramp. b) Outboard slice of ramp. c) Entire ramp.

3.2.2.1 Ramp Corner. If the ramp corner is released into these ascent conditions, it is torn away from the ramp with a strong initial nose-out yawing moment and dragged violently downstream. The initial trajectory of this piece is shown in Figure 21 where 3 impact events are evident (impact velocities less than 0.15 Mach). At each incident, the present methods correctly detect contact and impose corresponding reaction impulses. After the third impact, the debris becomes airborne and rapidly accelerates in response to aerodynamic loads.

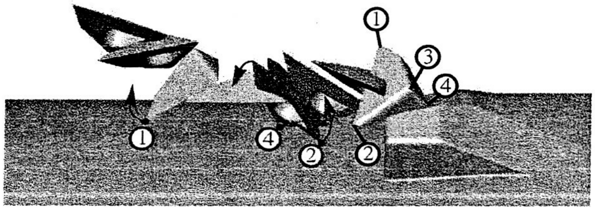

Figure 21. Break-away and initial trajectory of ramp corner (3 impact events indicated).

3.2.2.2 Ramp Slice. Results from the ramp slice break-away simulation reveals a limitation in the current contact point determination algorithm. As noted in Section 2.4.3, point, line, and planar contacts are possible. However, the present contact detection algorithm reduces all contact incidents to points. Sometimes linear and planar contacts can be treated this way without gross error. However, the present case represents a planar contact situation that must be treated as a planar contact for accurate post-impact trajectory prediction.

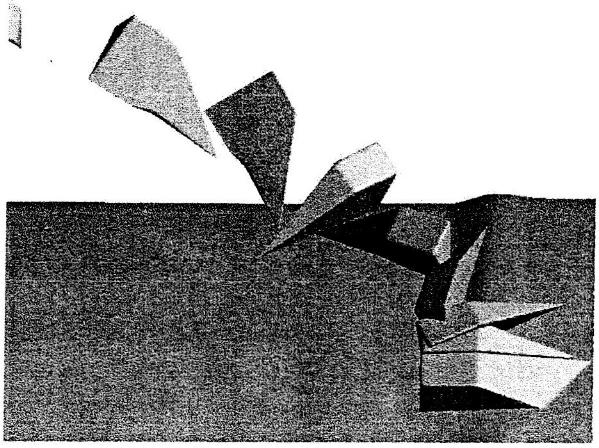

Figure 22. Break-away and initial trajectory of outboard slice of ramp.

Figure 22 illustrates the break-away trajectory of the ramp slice. This appears to be a nominal result and free of contact incidents. However, closer inspection of the results is productive. Figure 23 shows the initial and one post break-away position of the slice and remaining portion of the ramp as viewed looking upstream from the rear. The initial position of the slice is within the contact detection tolerance of the ramp. Accordingly, the present contact detection routine correctly identifies all surface grid points inside the gap between the slice and ramp to be in contact. The contact cloud routines categorize these points as being in a single cloud which is reduced to a single contact point located at the centroid of the gap cloud.

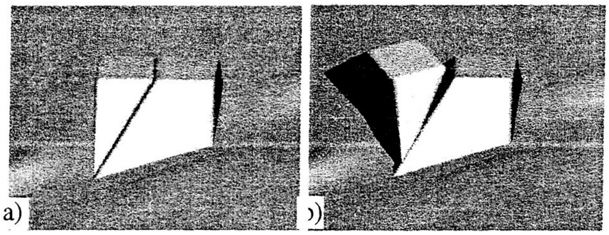

Figure 23. Penetration of outboard slice in remaining portion of ramp.

Similar to the ramp corner, if the slice is released into the ascent conditions of the launch vehicle, it is initially torn away from the ramp with a nose-out yawing moment. Even at release, contact is detected and considered by the impact mechanics routines. However, a nose out motion of the slice indicates movement away from the ramp relative to the cloud averaged contact point values. This is so even though the lower aft corner of the slice is moving in the opposite direction. As a result, penetration occurs briefly in this case as shown in Figure 23b. Accordingly, the cloud condensation algorithm described in Section 2.4.2 is to be extended to accommodate line and planar contacts.

3.2.2.3 Entire Ramp. If the entire ramp is released into the ascent conditions of the launch vehicle, it is torn away from the ramp with an inboard side down rolling moment, combined with a nose down pitching moment, and dragged downstream. The initial trajectory of this piece is shown in Figure 24. Four impact events occur during the very early stages of this break-away sequence (see Figure 25). At each incident, the present methods correctly detect contact and impose corresponding reaction impulses. After the fourth impact, the debris becomes airborne and rapidly accelerates in response to aerodynamic loads.

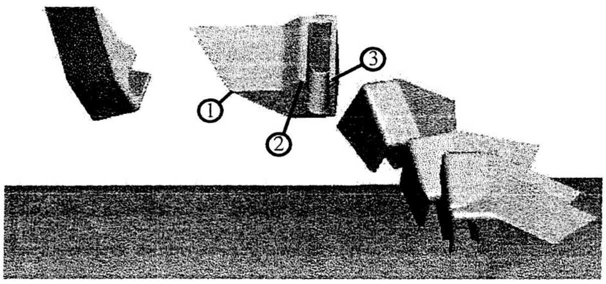

Figure 24. Break-away and initial trajectory of the entire ramp.

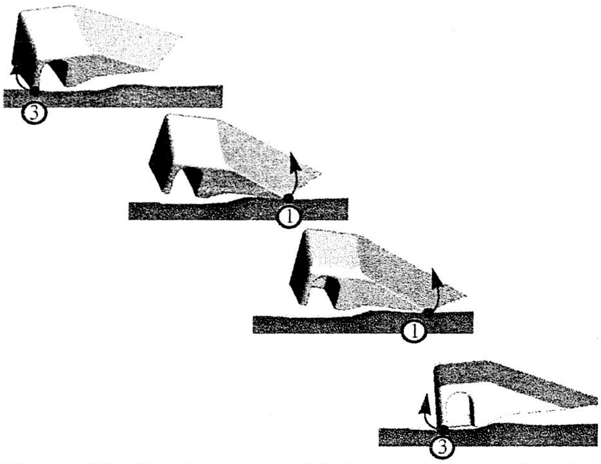

Figure 25. Break-away and initial trajectory of entire ramp (4 impact events indicated).

3. CONCLUSIONS

Methods to accurately predict the response of multiple body systems in proximate flight are available. This class of problems require predictive capability for body motion in response to aerodynamic loads, applied loads, and interbody collisions. The methods described in the present paper are shown to accurately predict analytic solutions to relevant test problems. In addition, the methods are successfully applied to 2 realistic unsteady, viscous, multiple body applications in three-dimensions. The "darts in proximate flight" case is a prototype of store separation and munition dispense problems. The "tumbling debris" case is a prototype for the new engineering discipline of computational aviation forensics.

The verification and demonstration computations indicate that the computational overhead associated with contact detection and impcat mechanics is negligible. One of the demonstration computations accentuates the need to extend the present contact-cloud methodolgy to accomodate line and planar contacts in addition to the currently supported and more common point contacts. Future work is to be focused on this topic and also on methods to further generalize and improve the computational efficiency of domain connectivity.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the contributions of several individuals. Dr. William Chan (NASA ARC) generated all of the grids for the pieces of debris considered in this paper. Mr. Reynaldo Gomez (NASA JSC) is the source of the launch vehicle grids employed in the debris simulations. Drs. Michael Aftosmis and Scott Murman (NASA JSC) provided clean triangulated surfaces of the debris and thereby contributed to this work. All of the above named individuals contributed valuable critical evaluation of the author's work during method development. Support for the contact detection and impact mechanics method development was provided by the U.S. Army Aeroflightdynamics Directorate (AMCOM) and the Naval Sea Systems Command, Indian Head.

REFERENCES

1. Steger, J., Dougherty, F. C., and Benek, J., "A Chimera Grid Scheme," Advances in Grid Generation, K. N. Ghia and U. Ghia, eds., ASME FED-Vol 5., June 1983.

2. Meakin, R., "Composite Overset Structured Grids," in Handbook of Grid Generation, J. Thompson, N. Weatherill, and B. Soni, eds., CRC Press, Inc., Boca Raton, FL, 1998.

3. Chan, W., Meakin, R., and Potsdam, M., "CHSSI Software for Geometrically Complex Unsteady Aerodynamic Applications," AIAA 2001-0539, AIAA 39th Aerosciences Meeting and Exhibit, January 2001, Reno, NV.

4. Meakin, R., "Automatic Off-Body Grid Generation for Domains of Arbitrary Size," AIAA 2001-2536, 15th AIAA Computational Fluid Dynamics Conference, June 2001, Anaheim, CA.

5. Buning, P. G., Jespersen, D. C., Pulliam, T. .H, Chan, W. M., Slotnick, J. P., Krist, S. E. and Renze, K. J., "OVERFLOW User's Manual," Version 1.8, NASA Langley Research Center, 1998.

6. MPI: A Message-Passing Interface Standard, Published by the University of Tennessee, Knoxville, TN, 1995, http://www.erc.MsState.Edu/mpi

7. Meakin, R., "Overset Grid Methods for Aerodynamic Simulation of Bodies in Relative Motion," Proceedings of the 8th Aircraft/Stores Compatability Symposium, October 1990.

8. Meakin, R., "Computations of the Unsteady Flow About a Generic Wing/Pylon/Finned-Store Configuration," AIAA 92-4568, AIAA Atmospheric Flight Mechanics Conference, August 1992, Hilton Head Island, SC.

9. Murman, S., Aftosmis, M., and Berger, M., "Simulation of 6-DOF Motion with a Cartesian Method," AIAA 2003-1246, 41st AIAA Aerospace Sciences Meeting, January 2003, Reno, NV.

10. Meakin, R., "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," AIAA 2001-2537, 15th AIAA Computational Fluid Dynamics Conference, June 2001, Anaheim, CA.

11. Baraff, D., "Analytical Methods for Dynamic Simulation of Non-Penetrating Rigid Bodies," Computer Graphics, Vol. 23, No. 3, July 1989.

12. Chatterjee, A. and Ruina, A., "A New Algebraic Rigid Body Collision Law Based on Impulse Space Considerations," J. Applied Mechanics, Vol 65, No. 4, pp. 939-951, Dec 1998.

13. Stronge, W. J., Impact Mechanics, Cambridge University Press, 2000.

14. Baraff, D., "An Introduction to Physically Based Modeling: Rigid Body Simulation II -- Nonpenetration Constraints," Online Siggraph '97 Course notes, http://www-2.cs.cmu.edu/~baraff/sigcourse/

15. E. Howard Smart., *Advanced Dynamics*, vol II. Dynamics of a Solid Body. MacMillen and Co., 1951.