DYNAMIC-MESH CFD AND ITS APPLICATION TO FLAPPING-WING MICRO-AIR VEHICLES

Andrew A. Johnson
Army High Performance Computing Research Center
Network Computing Services, Inc.
Minneapolis, Minnesota 55415

ABSTRACT

We are currently developing new numerical simulation methods and computational fluid dynamics (CFD) codes designed for advanced fluid-structure interaction (FSI) applications that have moving mechanical components and/or changing domain shapes. The method is called Dynamic-Mesh (DM) and is currently being implemented in parallel within our XFlow CFD simulation code. This method involves the tight coupling of automatic mesh generation (AMG) technology with more traditional parallel CFD methods designed for unstructured meshes. By coupling these two distinct technologies together, the mesh generation process never stops and continues throughout the entire simulation. By doing this, we can define a so-called "dynamic" mesh that has the ability to adjust, change, and modify its structure in response to any changes in geometry or other factors. DM-CFD technology of XFlow can be used to model the fluid flow around or within flapping-wing vehicles, rotorcraft, engines, turbines, pumps, airdrop systems, and has applicability to modeling free-surface flow, fluid-particle flow, energy/nuclear systems, and many bio-medical applications. Traditionally, these are some of the most difficult applications to simulate.

We are currently demonstrating and testing the DM technique and the capabilities of XFlow through a series of complex FSI applications. These applications include the simulation of airdrop systems involving the deployment (i.e. opening) of parachutes, bio-medical applications, and the simulation of micro air vehicles (MAV) and biological systems. Results of the modeling of a flapping-wing MAV will be highlighted here to demonstrate the capabilities and potential of the DM method in XFlow, as well as providing some illusrative results for an interesting application.

1. INTRODUCTION

The ability of numerical simulation methods to model 3D CFD problems with complex geometries and configurations with a relatively high degree of accuracy has improved dramatically over the last two decades. This is a result of exhaustive work on numerical methods, availability of parallel computing hardware and scalable software technology, advanced formulations including turbulence models and iterative equation solvers, and AMG techniques. One common factor that most of these numerical simulation methods have is their reliance on an underlying fixed and/or static computational mesh, either structured or unstructured. Unfortunately, a static mesh can restrict a researcher's ability to solve the large class of applications that have moving geometries and/or changing domain shapes, which includes FSI applications. Because of this, the use of numerical simulation techniques for these types of problems is less common, more difficult, and/or generates less accurate results.

We are proposing a Dynamic-Mesh method in which the underlying computational mesh is not static, but is dynamic and allowed to change its shape, structure, refinement levels, and nodal connectivity throughout the entire simulation process. We accomplish this by coupling AMG methods and technology (Johnson & Tezduyar 1997b, 1999), in parallel, with more traditional flow solver technology (Johnson & Tezduyar 1996, 1997b). By doing this, the AMG process never stops and is allowed to modify the mesh structure and shape at any time based on the changing geometries of the application or other conditions. The method places no restrictions on the type and/or extent of the motions of the boundaries involved, so it is truly a general-purpose and automatic method. We have implemented the DM technique within our new XFlow CFD simulation code and are currently testing it through a series of complex FSI applications on the AHPCRC's Cray X1E. We believe that our DM technology can enable us and others to now solve applications with moving geometries with similar facility and accuracy as is currently being accomplished on static geometry problems.

One of the main difficulties in implementing such a complex technique in parallel is the limitations imposed by the MPI programming model, and this has limited the applicability of our early work on automatic mesh re-generation methods discussed in Johnson & Tezduyar 1999. To overcome this difficulty in parallelization of our AMG methods, we have implemented XFlow using the newer Unified Parallel C (UPC) programming language which is based on the partitioned global address-space (PGAS) concepts which are just now emerging as a practical alternative to the more traditional message-

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 NOV 2006**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE
Dynamic-Mesh CFD and its Application to Flapping-Wing Micro-Air Vehicles

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Army High Performance Computing Research Center Network Computing Services, Inc. Minneapolis, Minnesota 55415

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002075., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**28**	

passing model based on MPI (El-Ghazawi et al. 2005, Johnson 2005). Using this PGAS model, all mesh data structures that are declared to be "shared" are visible and accessible by all processors at any time, even though the mesh is still partitioned and distributed amongst the processors in a traditional way. PGAS concepts are supported at the hardware level on the Cray X1E, and because of this, UPC is very efficient on that system.

As our Dynamic-Mesh methods and technology develops, we envision it as taking on more of a "library" and/or "framework" approach to form a high-level and unifying basis for performing many types of complex FSI applications on high performance computing systems. In that sense, it would be called a Dynamic-Mesh FSI method and not tied to any specific solver or underlying numerical simulation methodology. In that unifying framework, various types of flow solver codes and methods could be coupled with various computational structural mechanics (CSM) codes, all built around a common dynamic-mesh framework for managing the underlying mesh structure, parallelism, and other factors. It might also be possible to use the dynamic-mesh features of the method within CSM applications that have large deformations and/or displacements of the domain to maintain mesh quality. Our initial tests and applications use the DM method for CFD applications with moving geometries. Examples are shown later in this paper and demonstrate that this technological framework can be a significant step forward for simulating some of the most challenging FSI applications.

2. DYNAMIC-MESH CFD

The difficulty in simulating moving geometry and FSI applications using traditional techniques is mainly due to the limitations and restrictions imposed by the underlying computational mesh. We are mainly concerned with, and developing DM methods for, Arbitrary Lagrange-Euelerian (ALE) simulations where the moving mesh conforms to the geometry of the application at all times. If the geometry of the application is changing or the domain itself is changing shape in a CFD simulation, one can typically use a formulation based on linear-elasticity theory to move/deform the mesh (Johnson & Tezduyar 1994), but eventually, the mesh will tangle and go bad for large displacements as the mesh-elements stretch and deform to accommodate this motion. At that point, it might be possible to apply re-meshing techniques (Johnson & Tezduyar 1996, 1997a), but that is very complicated, time-consuming, difficult, and introduces unwanted, and in many cases, unacceptable projection errors into the solution. For most mesh-conforming ALE methodologies, these have typically been the only viable options for simulating complex mesh-moving CFD applications. There are other non-ALE techniques such as chimera and/or overset methods that can be applied to model certain classes of FSI applications (Meakin 1995), but our goal here is to develop a general-purpose framework and methodology that can be applied to all types of FSI applications with no limitations imposed on the extent or types of motions involved.

The Dynamic-Mesh method as incorporated in the XFlow CFD code is built from a combination of several more traditional numerical modeling techniques, and it is through this coupling that an advanced numerical simulation capability is created. At a high level, there are three methods that are being coupled to form the DM-CFD method. These include our "traditional" stabilized finite-element based CFD methods for solving time-accurate incompressible flow, our face-swapping and Delaunay-based AMG techniques and algorithms for generating and manipulating unstructured (tet-based) meshes, and our parallel implementation and inter-processor communication framework. In the DM-CFD method, we still perform a normal time-accurate CFD simulation using "traditional" formulations, solvers, and routines. This includes our ALE formulation so that the underlying computational mesh can be moving and/or deforming based on the results of or linear-elasticity formulation and solver which moves the mesh based on the displacement and/or movements of the boundaries. The parallelization of these CFD routines and solvers is traditional, based on mesh partitioning and distribution, with fast inter-processor communication routines tying the mesh together, but using UPC language constructs rather than MPI. As the simulation progresses and the mesh structure moves and is deformed by the linear-elasticity formulation in response to boundary displacements, the coupled AMG routines are called-upon to clean-up and reconfigure the underlying mesh structure at so-called "dynamic-mesh update" stages. This could happen at every time step, or at some fixed multiple of time steps set by the user. During these update stages, a small fraction of elements are rearranged, a few new nodal points could be added, and some existing nodal points could be taken away. This is all controlled by certain mesh quality criteria (both element shape and nodal refinement) which is maintained throughout the simulation. Since only a small fraction of the mesh changes at each dynamic-mesh update stage, the effect on the underlying fluid-flow solution is minimal. In locations where the mesh does change, we apply appropriate solution interpolation and/or projection techniques to accurately maintain the solution variables. After the dynamic-mesh update stage, the 3D mesh structure is again optimal with respect to our overall mesh quality criteria, and the simulation progresses on to the next time step (or steps). Periodically throughout the simulation, we apply a mesh re-partitioning stage to maintain optimal parallel distribution of the mesh in response to the underlying changes in the mesh structure.

3. MICRO AIR VEHICLES

Unmanned aerial vehicles (UAVs) are commonly used today for reconnaissance, surveillance, and even attack missions. UAVs currently exist in a wide variety of sizes to match specific mission goals and are typically sized anywhere from small model airplanes (Dragon Eye) to large jet aircraft (Global Hawk). It is anticipated that the size of UAVs will continue to become smaller as the technology develops.

A very small class of UAV sized similar to a small bird or insect with wingspans on the order of a few centimeters is classified as a micro air vehicle (MAV), and these systems could incorporate flapping wings to mimic biological systems. It is envisioned that such small MAV systems with hovering capabilities could be used someday by individual soldiers for reconnaissance missions. It is because of this future technology requirement/desire that we have targeted flapping-wing MAVs and biological systems (for example, our hovering hummingbird simulation discussed in Figure 1), as one of the main application areas for our DM method and XFlow CFD/FSI simulation technology. Although such MAV systems can't be physically built and flown today, they can be designed, tested, flown, and studied on the computer using numerical simulation techniques. By studying these systems computationally, the aerodynamic factors, performance, design requirements, control algorithms, and maneuvering can all be studied through simulation, thus allowing us to assess the feasibility of MAVs before they are built. But, it is not our goal in this work to actually design MAV systems, rather it is to develop numerical simulation techniques that could applied to such design. Our simulations here are for validation of the numerical techniques, demonstration of capabilities, and to validate the application space.

3.1 Robofly Experiments

In order to help validate our DM methods and XFlow CFD simulation code, specifically for small flapping-wing systems, we have initiated an effort to numerically reproduce the so-called "Robofly" experiments of Dr. M. Dickinson (Dickinson 1999, Birch & Dickinson 2003). In those experiments, a scaled-up pair of *Drosophila* (fruit-fly) wings with radius 0.25 meters are immersed in a tank of mineral oil. Rotations of the wings, both longitudinal and rotation (angle-of-attack) that models flapping cycles of the real insect are performed using a mechanical apparatus, and a force sensor attached to the base of one wing measures lift and drag forces. The maximum velocity of the wing-tip is on the order of 0.25 meters/second which corresponds to a fairly low Reynolds Number (RE) of around 136, which is similar to the RE of real fruit-flys. Dickinson and others (Sane 2003) report observed flow features such as rotational lift, wake re-capture, and attached leading-edge vorticies which all enhance the lift generated by insect wings, well beyond values predicted by conventional wing/airfoil theory.

Our geometric model that tries to reproduce these experiments is shown in Figure 2, as well as some of the initial lift forces and simulation results from our numerical experiments with XFlow. The 3D meshes in our experiments are on the order of 3 million tetrahedral elements, and we used 12 multi-streaming processors of the Cray X1E to perform each of our simulations. We tried to design a wing geometry, motions, and rotations to best match those values reported in Dickinson 1999, but that paper does not go into great detail about many of these values so there is some amount of uncertainty involved. Because of this, we actually performed several simulations and varied some of these estimated quantities, and we report two of those simulation results in Figure 2. As can be seen in this figure, our simulations provide respectable results in reproducing the behavior and flow features reported for the Robofly system (i.e. rotational lift, wake re-capture, etc.), but further work and refinements to our modeling of the system is required. We will be continuing to work on these, and other validation studies, and have began discussions with Dr. Dickinson on how to refine our Robofly set-up and simulations to best model his experiments and geometries.

3.2 Study of a Hypothetical MAV Design

In order to demonstrate the DM method as incorporated into XFlow for flapping-wing MAV applications, and to determine some of the factors and parameters that are involved in their performance, we have designed a simple (hypothetical) MAV shown and discussed in Figure 3, and are testing it through a series of simulations under cruise conditions. Its design, size, cruise performance, and flapping-wing motion is modeled after real insects as discussed in Ellington 1999 and Sane 2003. This MAV geometry has a long thin body with flat, flexible wings which could possibly be made using an electro-active polymer (EAP) material (Bar-Cohen 2004). EAP is a plastic-like material that bends and flexes based on the amount of electrical current fed to it. With the inclusion of sensors and a control circuit, it might be possible to perform complex wing motions and maneuvers, yet still be fairly simple in design and construction.

We tested two different cruise velocities for the MAV; one at 4.5 m/sec, which corresponds to a typical cruise velocity of an insect of similar size (Ellington 1999), and another at 2.25 m/sec. These cruise velocities correspond to RE of 1,552 and 776 respectively, based on an average wing chord length of 0.5 cm. We conducted a parametric study of this MAV design by performing a series of simulations where we vary the wing-beat frequency and amplitude of wing motion. Results of these simulations are shown and discussed in Figures 3 through 6. These results show that this could be a functional MAV design, and the airflow results and forces generated are similar to those observed in nature. However, the simulations also highlight some of the inefficiencies of this design, mainly that a higher wing-beat frequency (and thus, more power) is required for cruising flight which is most likely due to the high drag at the root of the wing where it attaches to the body.

4. OTHER APPLICATIONS

Along with the simulation and studies of flapping-wing micro air vehicles and biological systems discussed in Section 3, we are also testing and demonstrating the DM method and XFlow CFD simulation technology for other complex applications that have more immediate relevance to the U.S. Army. Two of these applications include airdrop systems and bio-medical devices as shown and discussed in Figures 7 and 8.

CONCLUSIONS

We have given a brief overview of the ability of our Dynamic-Mesh method to model different types of CFD problems with dynamically changing geometries. It is through this coupling of AMG technology with parallel CFD techniques that makes the simulation of this challenging class of applications much easier, efficient, accurate and robust. We also present simulation results of using the DM method as incorporated in the XFlow CFD code for a class of challenging applications of importance to the U.S. Army. These include flapping-wing micro air vehicles including biological systems, airdrop systems, and bio-medical applications. These initial results are encouraging and show the potential of the DM-CFD method and technology to solve a wide range of problems that traditionally have been difficult to address.

Some future development activities for the DM method and XFlow CFD simulation code will include; 1) enhancement and increased scalability of the UPC-based parallelization strategy, 2) increased robustness and flexibility of the dynamic-mesh update algorithms, 3) coupling with and/or incorporation of various structural models for fabrics, membranes, and other solid structures, 4) implementation of 6 degrees-of-freedom dynamics and basic control algorithms for MAV simulations.

ACKNOWLEDGEMENTS

This document was developed in connection with contract DAAD19-03-D-0001 with the U.S. Army Research Laboratory. The views and conclusions contained in this presentation are those of the authors and should not be interpreted as presenting the official policies or positions, either expressed or implied, of the U.S. Army Research Laboratory or the U.S. Government unless so designated by other authorized documents.

REFERENCES

Bar-Cohen, Y., 2004: *Electroactive Polymer Actuators as Artificial Muscles*, SPIE Press Monograph Vol. PM136.

Birch, J. and Dickinson, M., 2003: The Influence of Wing-Wake Interactions on the Production of Aerodynamic Forces in Flapping Flight, *J. Exp. Biology*, **206**, 2257-2272.

Dickinson, M., Lehmann, F., and Sane, S., 1999: Wing Rotation and the Aerodynamic Basis of Insect Flight, *Science*, **284**, 1954-1960.

El-Ghazawi, T., Carlson, W., Sterling, T., and Yelick, K., 2005: *UPC, Distributed Shared Memory Programming*, John Wiley & Sons Ltd.

Ellington, C., 1999: The Novel Aerodynamics of Insect Flight, Applications to Micro-Air Vehicles, *J. Exp. Biology*, **202**, 3439-3448.

Johnson, A. and Tezduyar, T., 1994: Mesh Update Strategies in Parallel Finite Element Computations of Flow Probblems with Moving Boundaries and Interfaces, *Comput. Methods Appl. Mech. Engrg.*, **119**, 73-94.

Johnson, A. and Tezduyar, T., 1996: Simulation of Multiple Spheres Falling in a Liquid-Filled Tube, *Comput. Methods Appl. Mech. Engrg.*, **134**, 351-373.

Johnson, A. and Tezduyar, T., 1997a: 3D Simulation of Fluid-Particle Interactions with the Number of Particles Reaching 100, *Comput. Methods Appl. Mech. Engrg.*, **145**, 301-321.

Johnson, A. and Tezduyar, T., 1997b: Parallel Computation of Incompressible Flows with Complex Geometries, *Internat. J. Numer. Methods Fluids*, **24**, 1321-1340.

Johnson, A. and Tezduyar, T., 1999: Advanced Mesh Generation and Update Methods for 3D Flow Simulations, *Comput. Mech.*, **23**, 130-143.

Johnson, A., 2005: Unified Parallel C within Computational Fluid Dynamics Applications on the Cray X1E, *Proceedings of the 2005 Cray Users Group Conference*, Albuquerque New Mexico.

Meakin, R., 1995: The Chimera Method of Simulation for Unsteady Three-Dimensional Viscous Flow, *Computational Fluid Dynamics Review 1995*, ed. Hafez, M. and Oshima, K., John Wiley & Sons Ltd., 70-86.

Sane, S. 2003: The Aerodynamics of Insect Flight, *J. Exp. Biology*, **206**, 4191-4208.

Warrick, D., Tobalske, B., and Powers, D., 2005: Aerodynamics of the Hovering Hummingbird, *Nature*, **435/23**, 1094-1096.

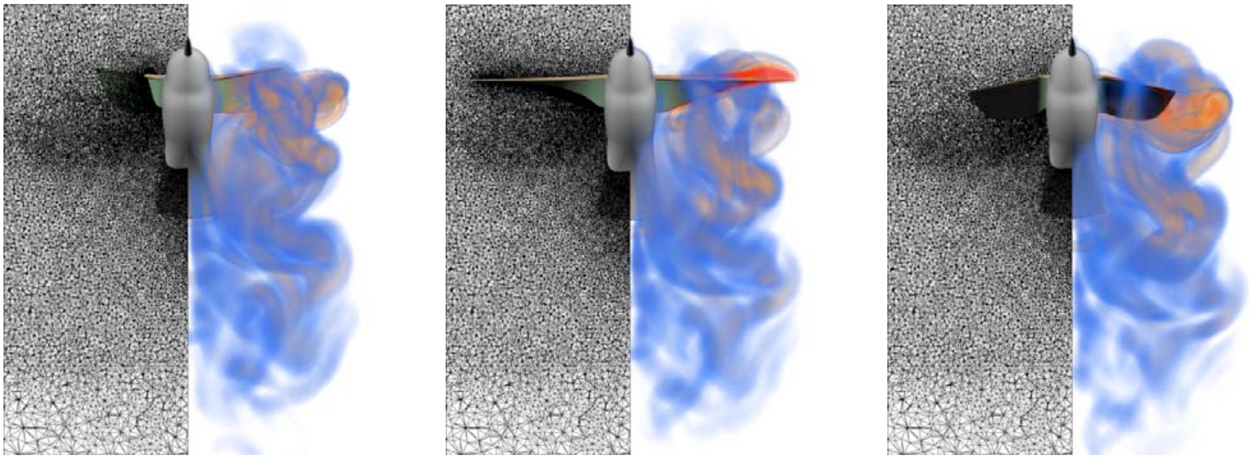


Figure 1. Numerical simulation of a hovering hummingbird. The wing motion is designed using algebraic and geometric equations designed to model wing flapping motions of real hummingbirds (Warrick et al. 2005) with a wing beat frequency of around 60 Hz. The system has a RE of roughly 2,000 based on the wing's average chord-length and average wing velocity. The mesh size increases from 5 to 7 million tetrahedral elements as the simulation progresses. We assumed symmetry and simulated half of the geometry. Shown here is the unstructured mesh at a vertical cross-section through the domain (left) and a volume-rendering of vorticity magnitude (right) at three time instances during the bird's forward stroke cycle. As can be seen here, a strong downward flow is generated under each wing that, as a result, produces lift.

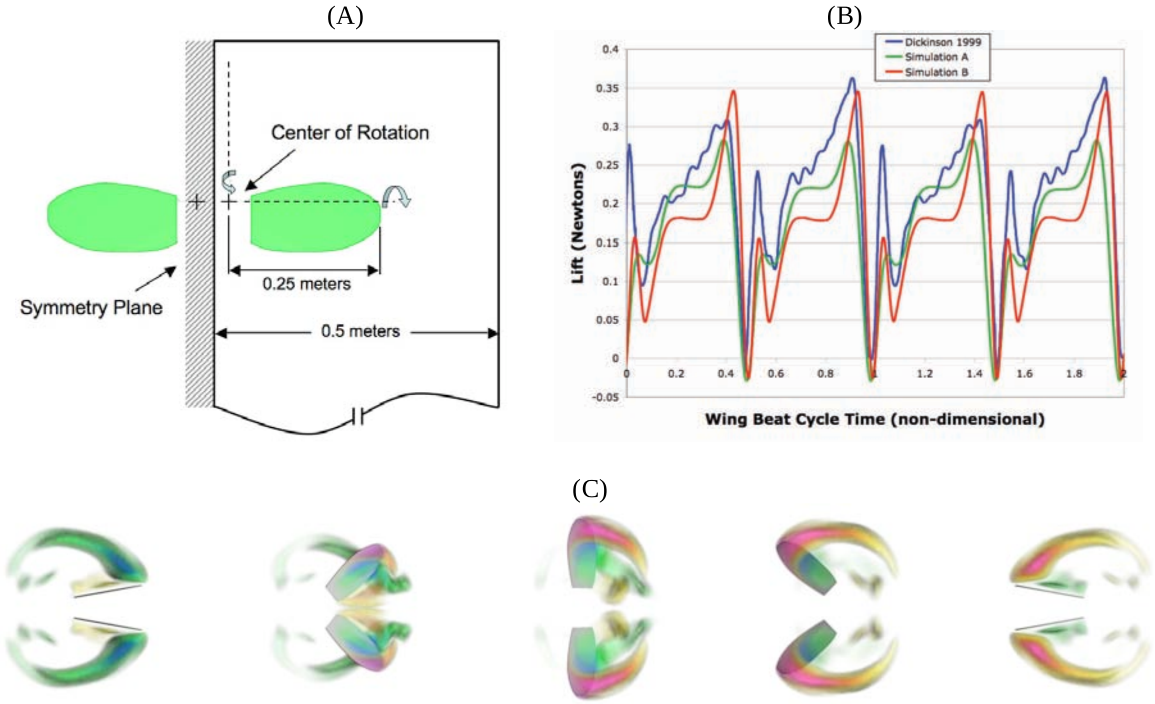


Figure 2. Validation studies based on scaled experiments of a robotic flapping *Drosophila* (fruit-fly) wing (Dickinson 1999). Our geometric model and set-up to reproduce these experiments is shown in (A). We designed a motion of the wings to try to reproduce as best we could the motion curves in Dickinson 1999 which were not based on simple sinusoidal rotations. There is a main translational-type rotation about the vertical axis through a 160 degree arc, and a secondary rotation about the wing's long-axis at the beginning and end of this cycle to obtain an angles-of-attack of 40 degrees at mid-stroke. The wing-beat frequency in the Robofly experiments is 145 mHz, which in the scaled experiment, closely matches those observed for real fruit-flys. Simulation results of the computed lift force over two wing-beat cycles for two of our simulations as compared to the Robofly experimental results (Dickinson 1999, Figure 3, symmetrical rotation) and are shown in (B). A volume rendering of stream-wise vorticity (helicity) for one half of a wing flapping cycle as seen from the top is shown in (C) where the red/yellow colors correspond to positive values and blue/green colors correspond to negative values.

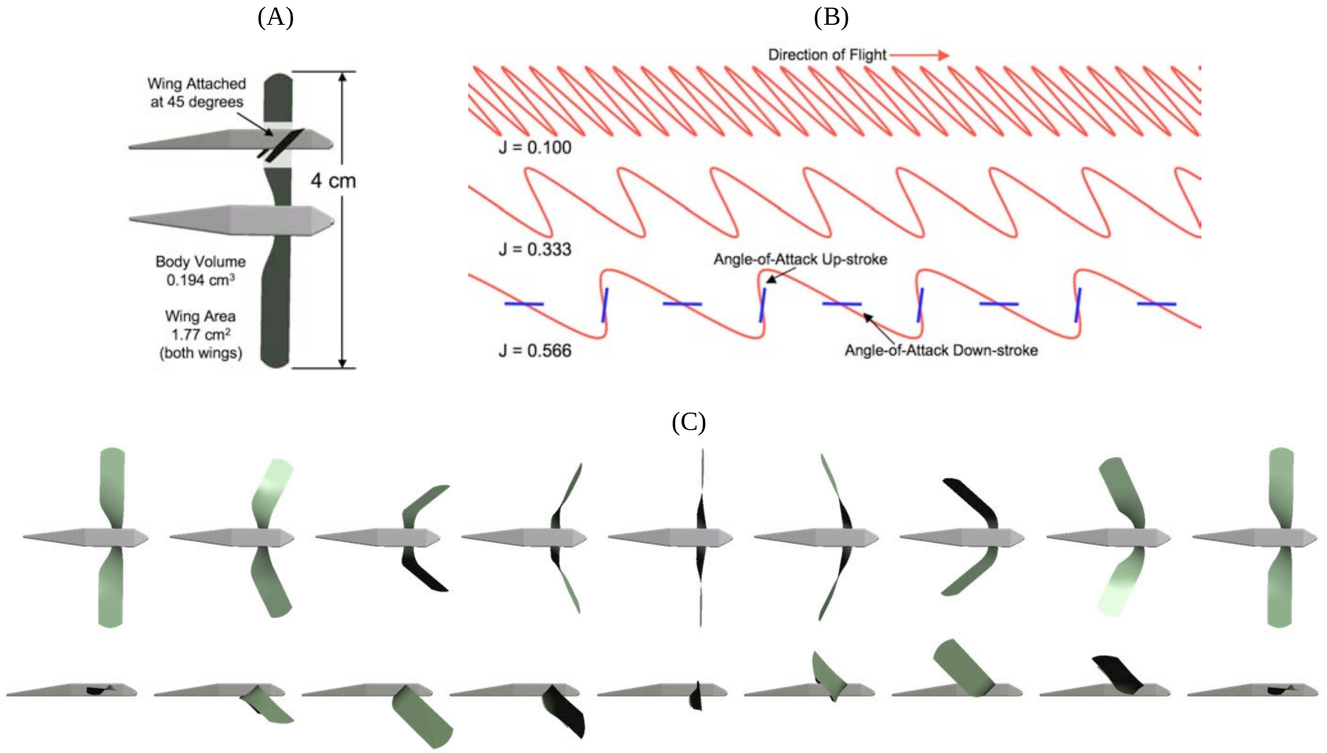

(A)
(B)
Wing Attached at 45 degrees
4 cm
Body Volume 0.194 cm³
Wing Area 1.77 cm² (both wings)
Direction of Flight
J = 0.100
J = 0.333
Angle-of-Attack Up-stroke
J = 0.566
Angle-of-Attack Down-stroke
(C)

Figure 3. The geometry and wing motion of our hypothetical MAV being simulated here. The size and shape of the MAV (A) along with the design of the wing motion is modeled after real insects as discussed in Ellington 1999. The wing oscillates through a 120 degree arc at a prescribed beat frequency along a stroke-plane which is inclined at 45 degrees to the MAV body with steady downward and upward (rotational) velocities at fixed angles-of-attack, with changes in direction and angle-of-attack at the maximum of the forward and backward stroke (C). The angle-of-attack of the wing relative to the wingtip path is set to 30 degrees for the downward stroke and 25 degrees for the upward stroke. The characterization of this motion can be seen in (B) where the path of the wing tip is plotted for several advance ratios [$J = V / (2 \Phi f R)$, V is the cruise speed, Φ is the wing beat amplitude in radians, f is the wing beat frequency, and R is the wing rotation radius]. Insects usually cruise at advanced ratio's of around 0.6 with an effective maximum of 1.0 for high-speed flight (Ellington 1999). For a fixed cruise speed, the advance ratio J directly corresponds to a specific wing-beat frequency [4.5 m/sec cruise speed, $J = 59.8\text{sec}^{-1} / f$; 2.25 m/sec cruise speed, $J = 29.9\text{sec}^{-1} / f$]. Although steady cruise conditions were simulated here, it is anticipated that this MAV could change its cruise velocity, and even hover, by simply varying the inclination of the stroke-plane by changing the orientation of its body, as it is done by real flying insects (Ellington 1999).

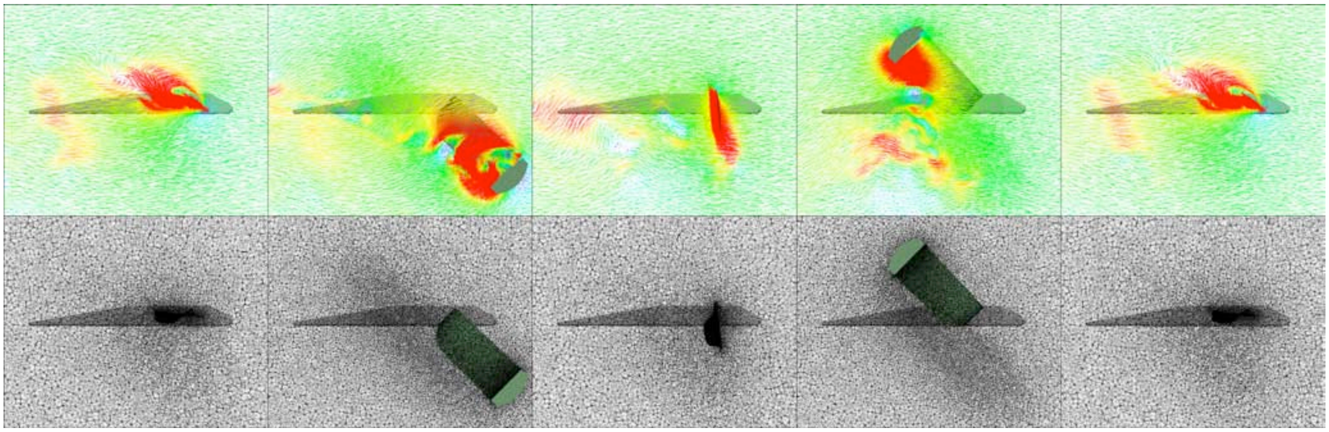


Figure 4. Velocity vectors (top) and mesh (bottom) at a vertical cross-section through the MAV wing at five time instances of one wing-beat cycle. The colors correspond to the speed of the flow. The attached leading-edge vortex can clearly be seen in the mid down-stroke images. The meshes used in these simulations contained between 6 and 8 million tetrahedral elements, and all simulations were conducted using 24 multi-streaming processors of the Cray X1E.

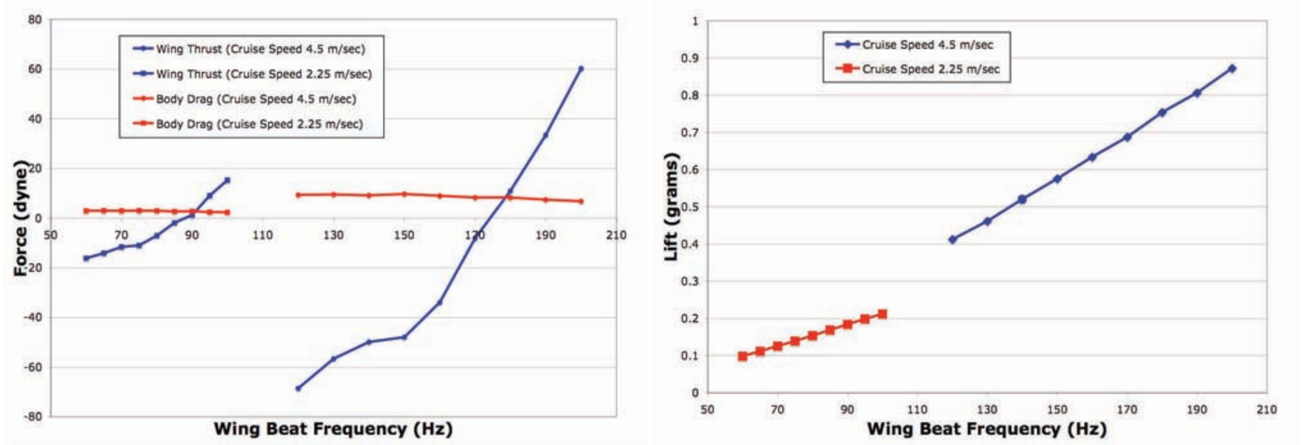


Figure 5. Results of parametric studies of changes in wing beat frequency for our hypothetical MAV system at two cruise velocities. Average drag (body) and thrust (wing) forces are plotted on the left as a function of the wing beat frequency. The MAV will cruise when the wing thrust compensates (matches) the body drag, which is at around 180 Hz for the faster cruise velocity and 90 Hz for the slower velocity. These frequencies are close to, but a little higher than those observed in nature for similar-sized insects. A bumblebee with a mass of 0.175 grams has a wing beat frequency of 149 Hz, and a hawkmoth with a mass of 1.6 to 2 grams has a wing beat frequency of around 25 Hz (Ellington 1999). On the right graph is plotted the average lift generated by the wings, measured in grams of weight supported, as a function of wing beat frequency, which shows a linear relationship. For the faster cruise velocity and "optimal" wing beat frequency of 180 Hz, the vehicle could support 0.75 grams, which is similar to an insect of this size. Figure 3 reports the MAV body volume as 0.194 cm^2. If its density was similar to aluminum (2.6 g/cc) it would have a mass of around 0.5 grams.

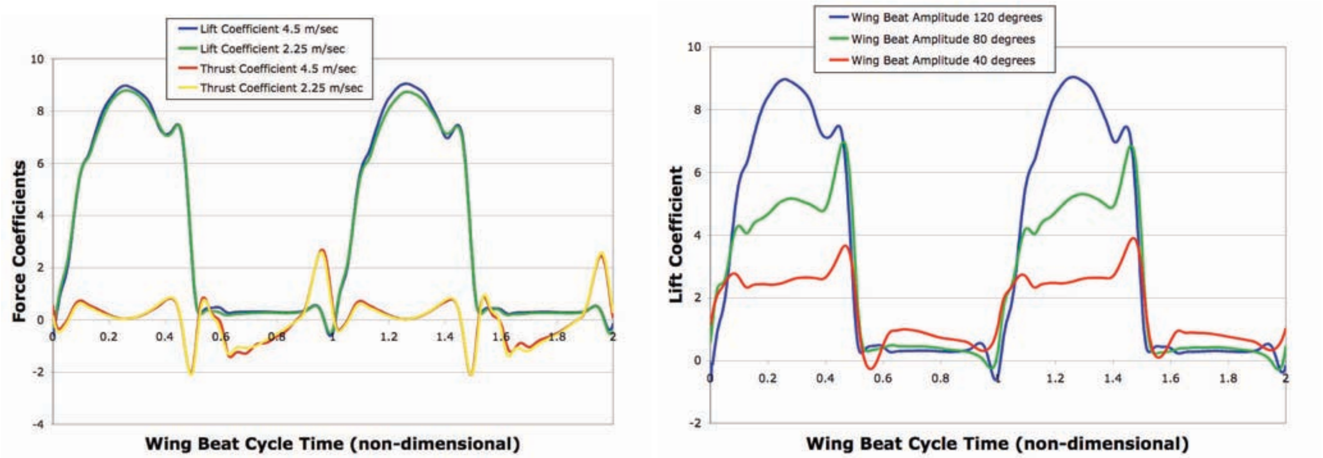


Figure 6. Shown are the lift and thrust force coefficients (based on the MAV cruise speed and wing's total area) generated by the wings over two wing beat cycles. The higher lift values are generated during the down-stroke. The wing's up-stroke produces little lift and high drag (negative thrust) in our design, but ideally, an effective MAV should be able to generate some thrust during the upstroke as it pushes back against the air. A wing beat amplitude of 120 degrees was used for most simulations, which is similar to amplitudes used by insects (Ellington 1999). The right graph shows the results of a series of simulations where the wing beat amplitude is varied. The average lift force generated by the wings varies linearly with the wing beat amplitude in our MAV system.

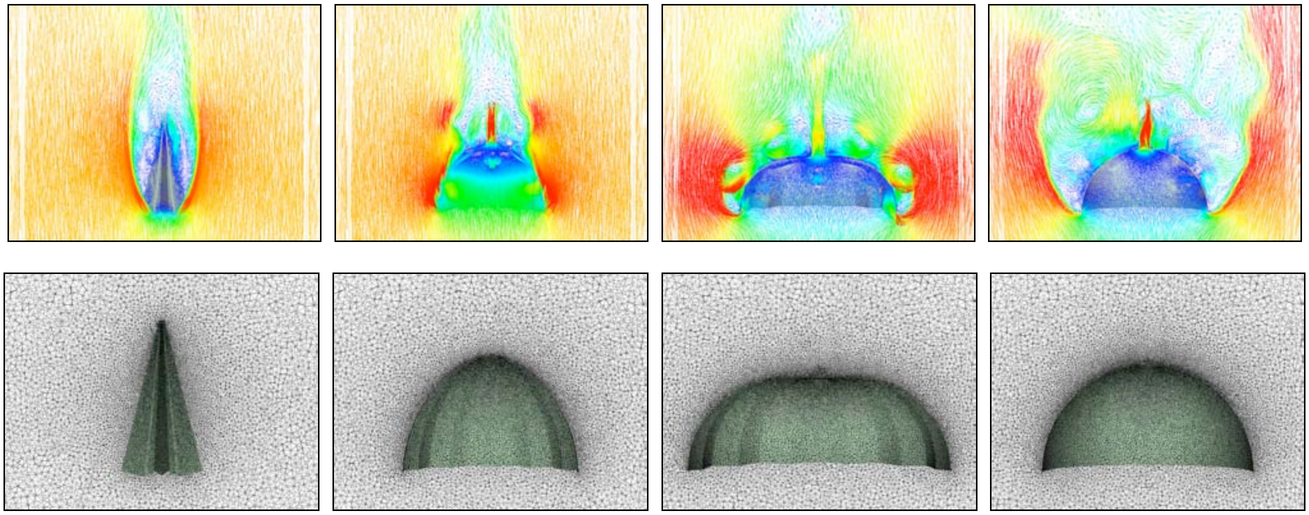


Figure 7. Numerical simulation of the opening (deployment) of a round parachute. This has been a very difficult problem to solve numerically using traditional methods due to the large changes in shape that are involved. In this simulation, the motion of the parachute surface is defined algebraically from an initially closed configuration where the sides are folded in (left image) to a fully opened and deployed half-sphere configuration on the right. We do not have a coupled fabric model within XFlow at this time, but we expect to accomplish that within the next year. Shown in these images are velocity vectors and the mesh at a vertical cross-section through the domain at four time instances during the opening sequence. As can be seen here, the geometry and resulting flow-field goes through very drastic changes. In this simulation, the vertical/descending velocity of the parachute is directly coupled to the instantaneous drag forces calculated on the parachute, and initially, the parachute with a 360 pound load (soldier and equipment) is at rest. As the system is released and accelerates in the closed configuration, a steady descent rate of 92 ft/sec is reached, and after opening, the descent rate drops to 20 ft/sec. During the 3 second parachute deployment, the deceleration of the system reaches a maximum G force of around 3. It is identified that DM-CFD methods and XFlow has great potential for other aspects of airdrop systems such as the separation of paratroopers and/or cargo from aircraft and the interaction of multiple canopies during decent.

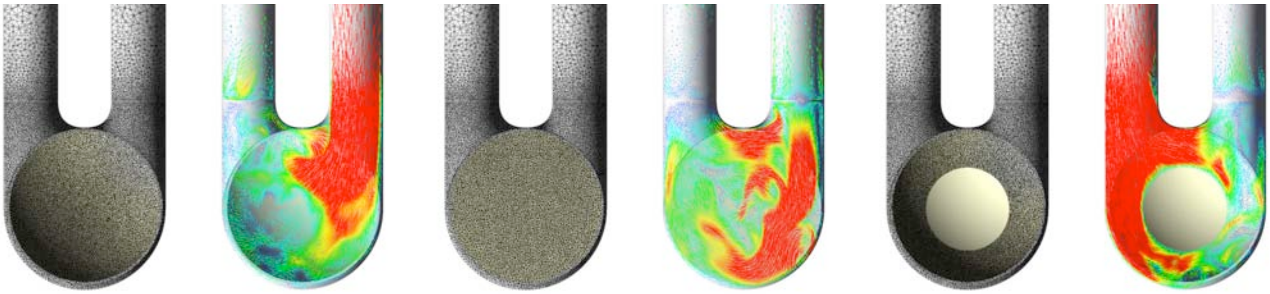
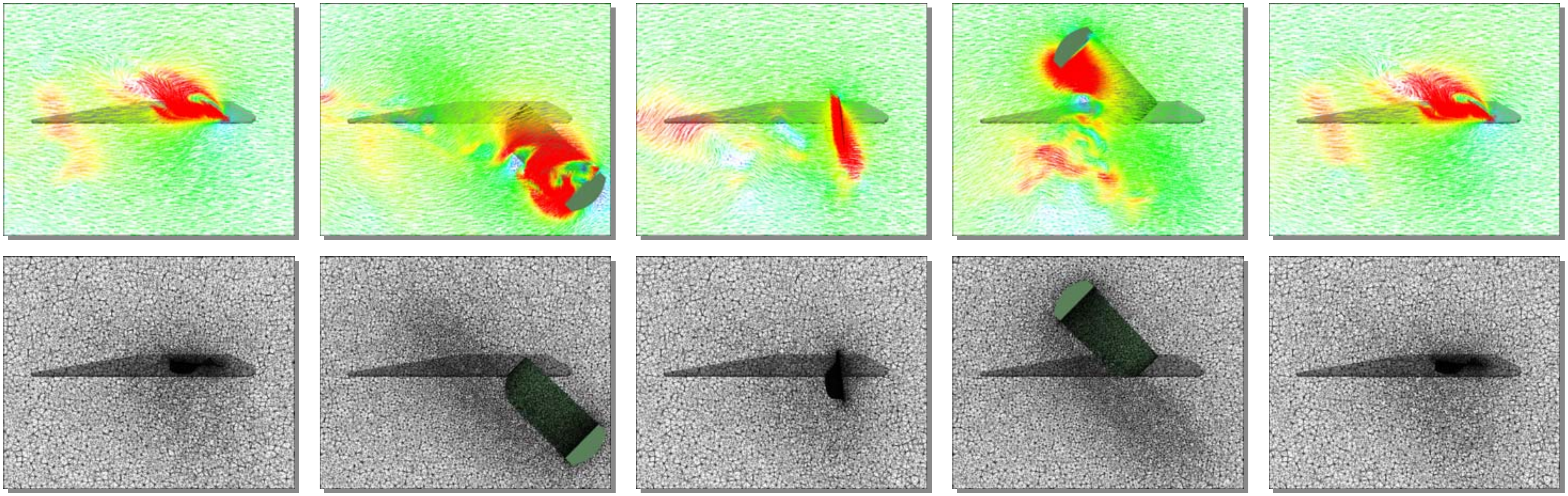


Figure 8. Numerical simulation of a self contained pumping device that could represent a total artificial heart. This simulation shows the potential of the DM-CFD method and XFlow to be applied to complex bio-medical applications. In this device, there is a flexible membrane (in the back of the geometry in these images) that moves in and out of the cylindrical chamber that drives the flow (it is envisioned that there are some mechanics and motors that are driving this motion behind the membrane). As the membrane moves backward (left two images), the volume is expanding which pulls the incompressible fluid in from the cylinder on the right, and as the membrane moves forward (right two images), the volume is contracting which forces the fluid out of the cylinder on the left. We do not model the valves in these cylinders physically, but we have developed and implemented a special internal boundary condition that only allows flow in one prescribed direction. These valves are opened-and-closed in sync with each other automatically based on instantaneous flow rates measured across the valves. The effect of these opening and closing valves can be seen here in the images that show velocity vectors and the mesh at a vertical cross-section through the domain at three time instances. The center two images are taken at an instance where one valve has just closed and the other has just opened. The fluid density and viscosity, volume flow rates, and pumping frequency of this simulation are modeled after actual values observed in the human circulatory system.

Dynamic-Mesh CFD and its Application to Flapping-Wing Micro-Air Vehicles

(25th Army Science Conference - November 2006)



Andrew A. Johnson, Ph.D.

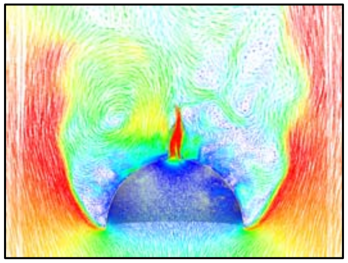
Army High Performance Computing Research Center

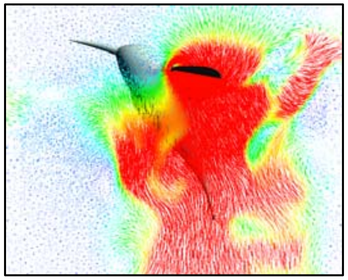
Network Computing Services, Inc.

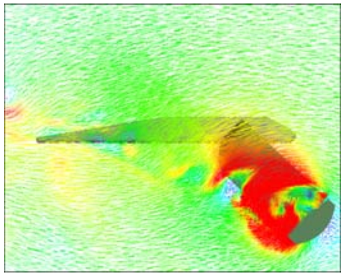
Minneapolis, Minnesota

Overview

- Developing advanced numerical simulation methods targeting complex fluid-dynamics applications
  - Moving geometries and changing domain shapes
    - Fluid-Structure Interactions (FSI)
    - Traditionally difficult to solve
  - General-purpose methods applied to any application
  - Fully parallel targeting large-scale applications
- Arbitrary Lagrange/Eulerian (ALE) methodology
  - Unstructured mesh conforms to the geometry at all times
  - Move the domain based on linear-elasticity theory
  - Older methods rely on re-meshing techniques





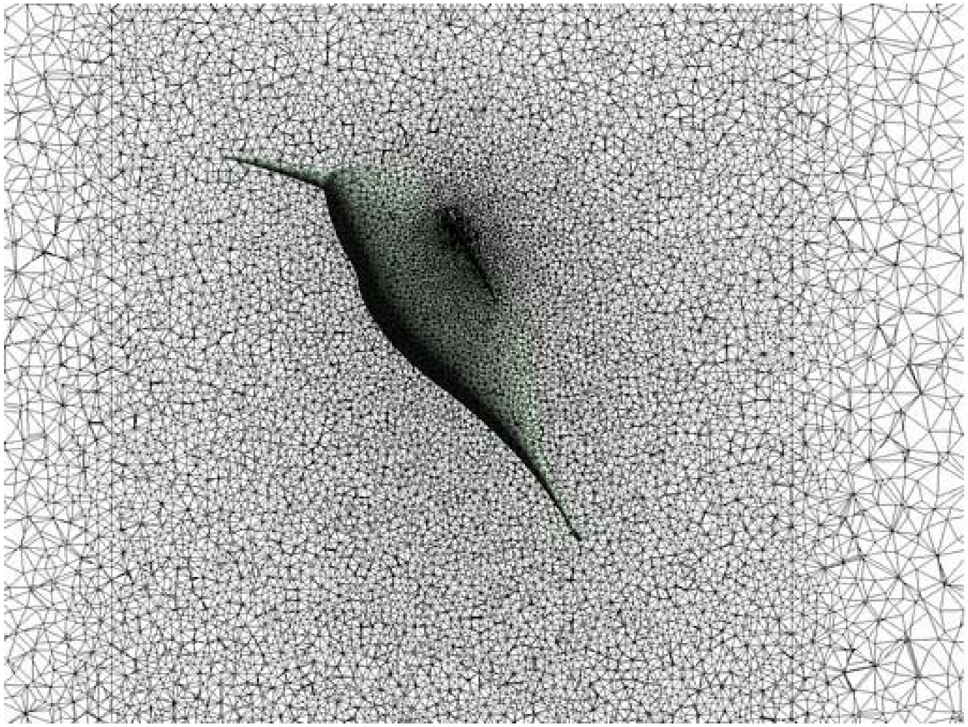


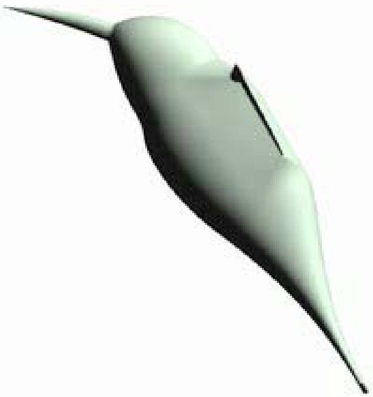
Overview (continued)

- Dynamic-Mesh Method (DM-CFD)
  - Evolution of previous work on automatic mesh regeneration methods
    - Could not implement in parallel using MPI in the ~1997 time frame
  - Using the newer Unified Parallel C (UPC) language for parallelism
  - Being implemented in our new **XFlow** simulation code
- All results from the AHPCRC Cray X1E

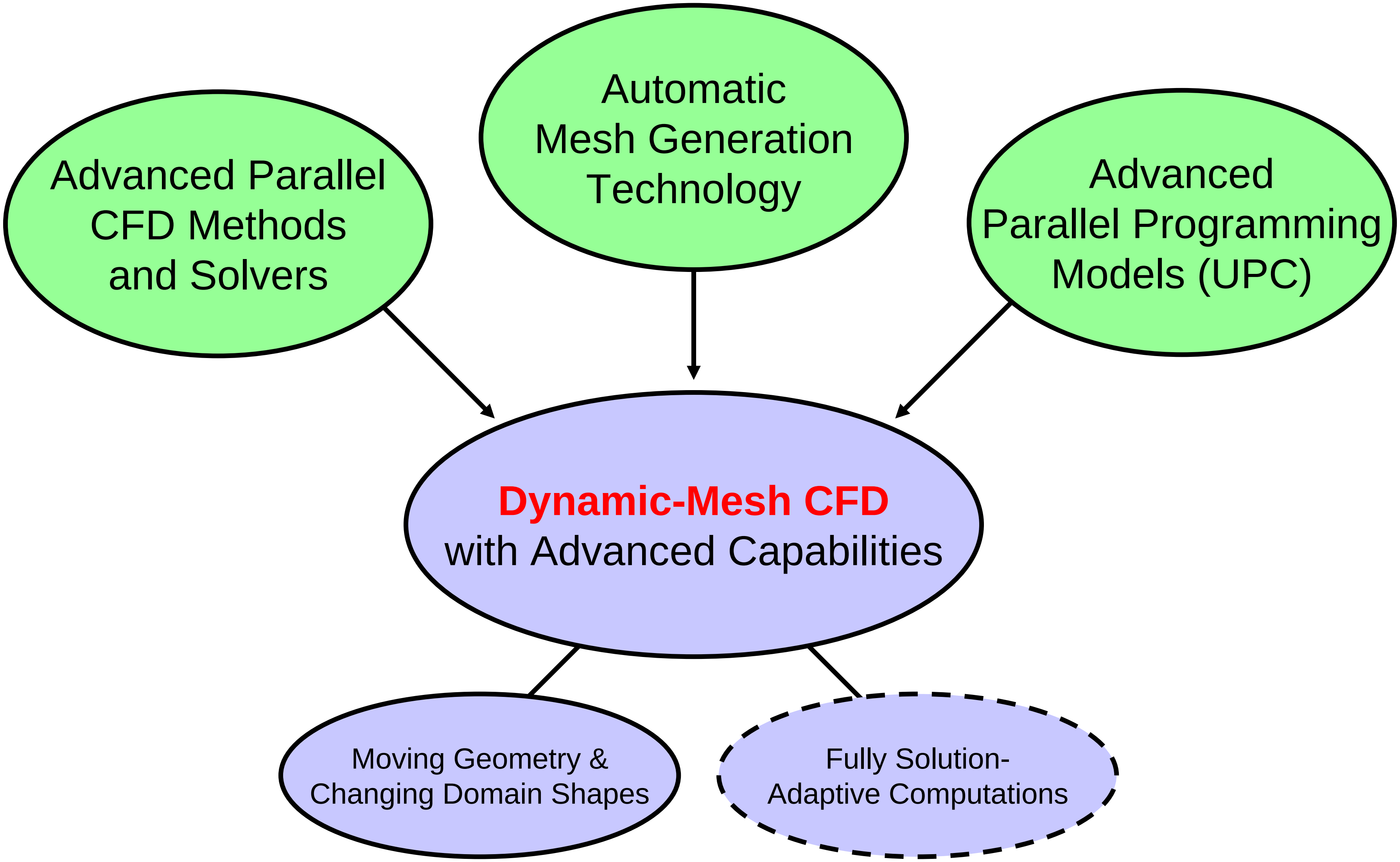
- CFD Methodology
  - Incompressible Flow
  - Stabilized Finite Element Methods
    - Unstructured meshes
  - Coupled Equation Systems
    - Pre-conditioned GMRES solvers
    - Matrix-Free implementation
  - Fully Parallel Implementation
    - Mesh partitioning, distribution, and fast inter-processor communication

Hummingbird Application





High-Level Overview



Advanced Parallel CFD Methods and Solvers

Automatic Mesh Generation Technology

Advanced Parallel Programming Models (UPC)

Dynamic-Mesh CFD with Advanced Capabilities

Moving Geometry & Changing Domain Shapes

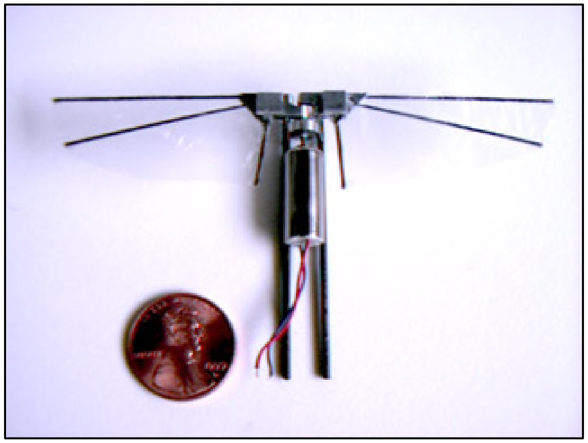
Fully Solution-Adaptive Computations

Dynamic-Mesh CFD

- Tightly couple automatic mesh generation technology within parallel flow solvers
  - Mesh generation never stops and runs in-conjunction with the flow solver
    - Mesh moves/deforms based on boundary displacements
      - Coupled linear-elasticity solver
    - Element connectivity changes as required
    - New nodes added as required
    - Existing nodes deleted when not needed
  - Mesh continuously changes due to changes in geometry or other conditions
- Complicated method
  - Parallelism (UPC), vectorization, dynamic data structures, solvers (mesh moving, fluid flow, others), general CFD accuracy, scalability, coupling, etc.

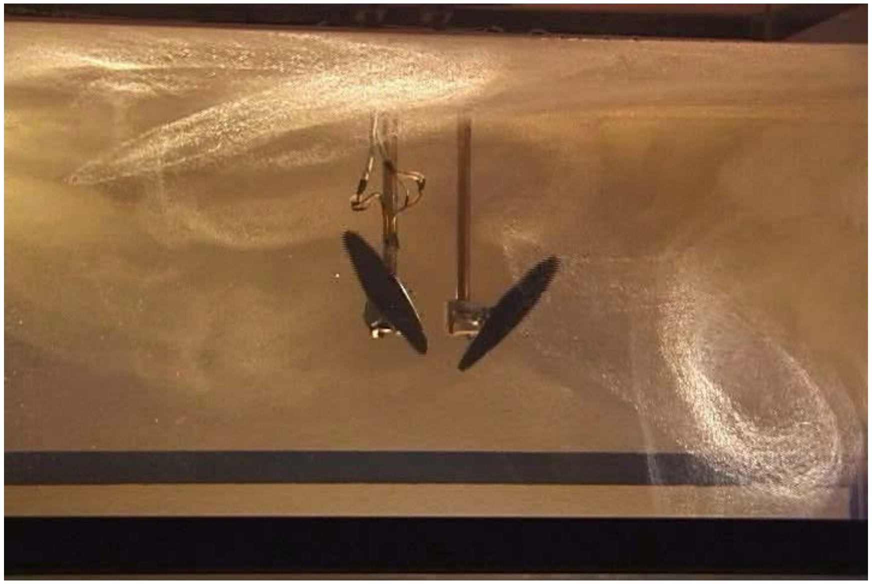
Micro-Air Vehicle (MAV)

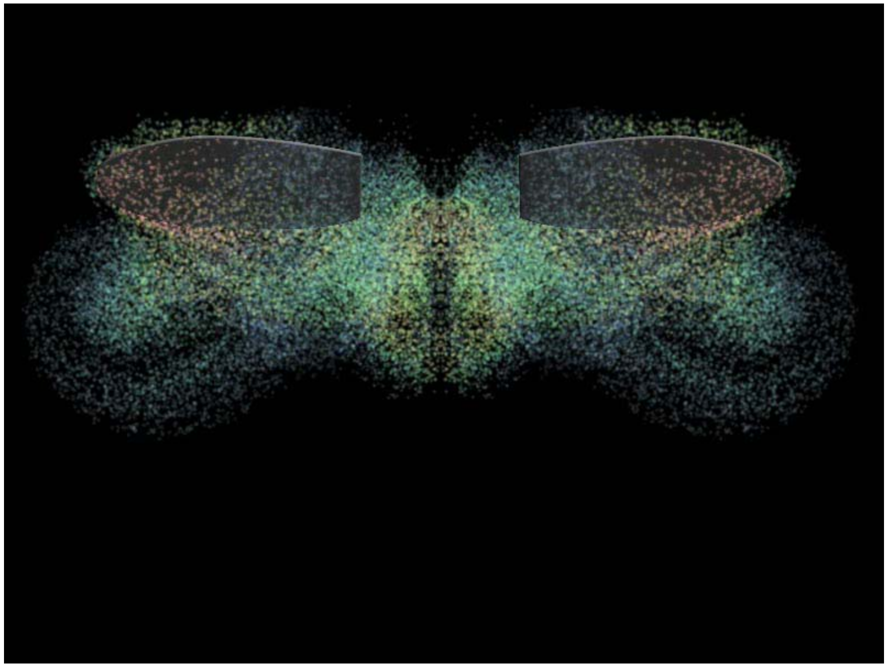
- Future soldier-portable unmanned aerial vehicle
  - Reconnaissance
    - Inside of buildings
    - Hovering capabilities
  - Detection of airborne hazards
  - Swarms of MAVs
- Targeted application area for DM-CFD
  - Validation of methodology, simulation code, and application space
  - Studies of hypothetical systems
    - Designs evaluated and requirements estimated in advance of actual vehicles



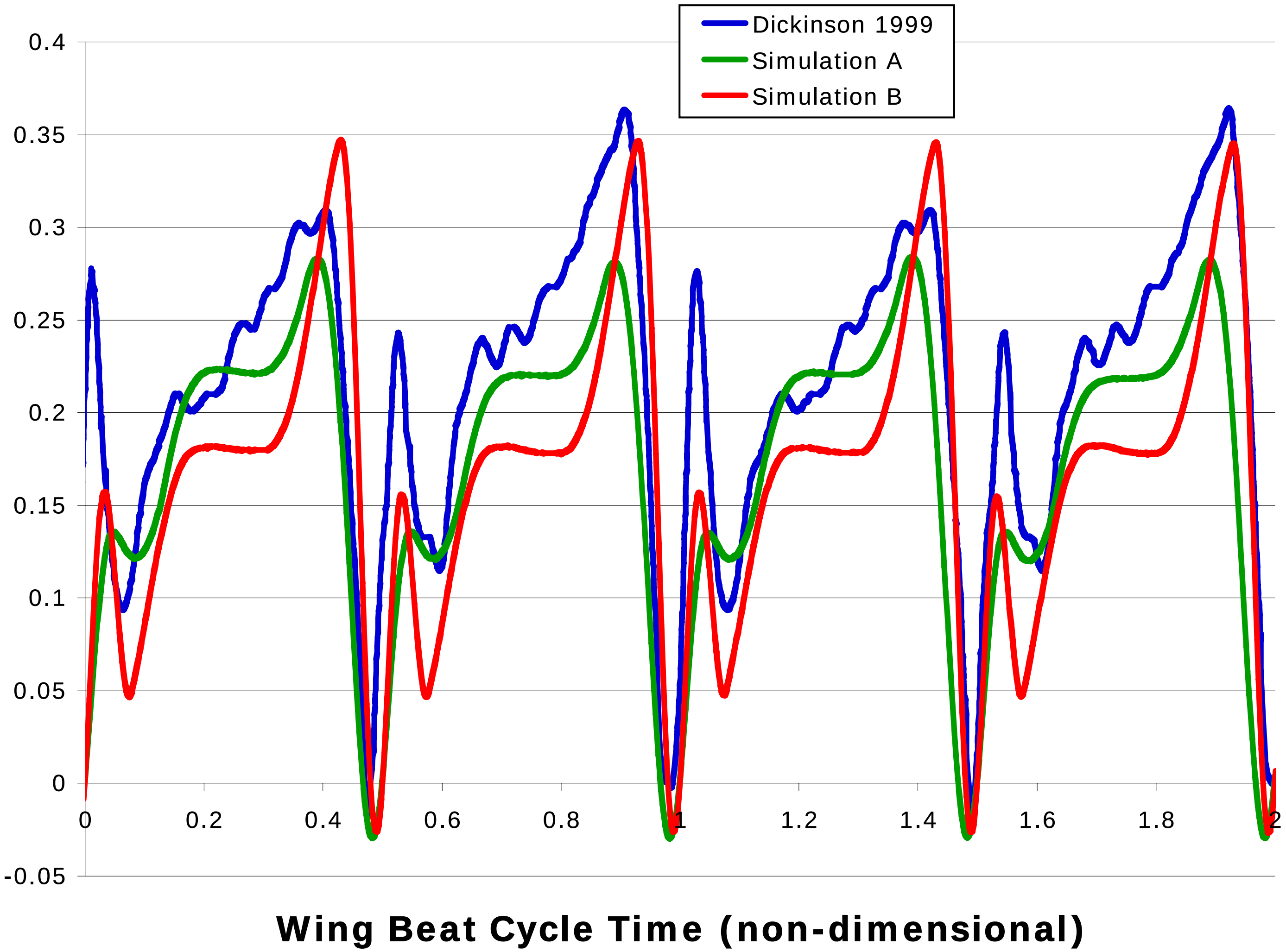
Validation Studies

- “Robofly” experiments of M. Dickinson (CalTech / ICB)
  - Experimental studies of flapping-wing aerodynamics
  - Estimated geometry and motion used in the Robofly experiments
- Studies of insect flight
  - Employ unique mechanisms to enhance lift



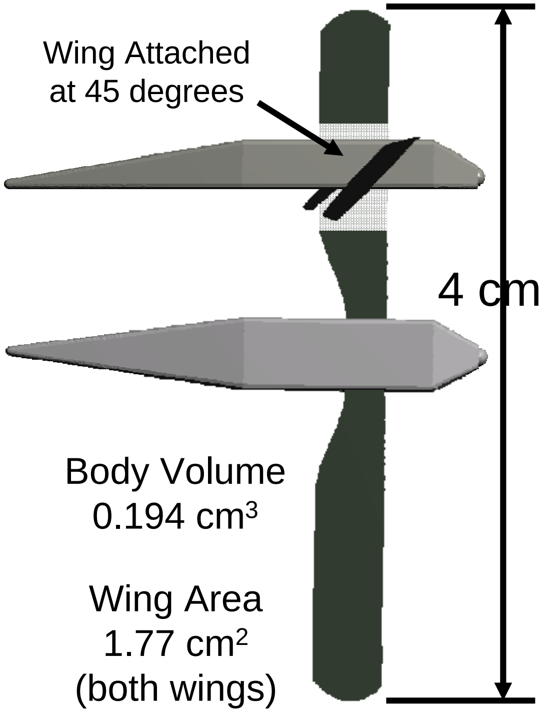


Validation Studies (continued)



Hypothetical Design

- Our own flapping-wing design modeled after insect flight and behavior
  - Initial simulations under “cruise” conditions
  - Wings modeled as thin electro-active polymers

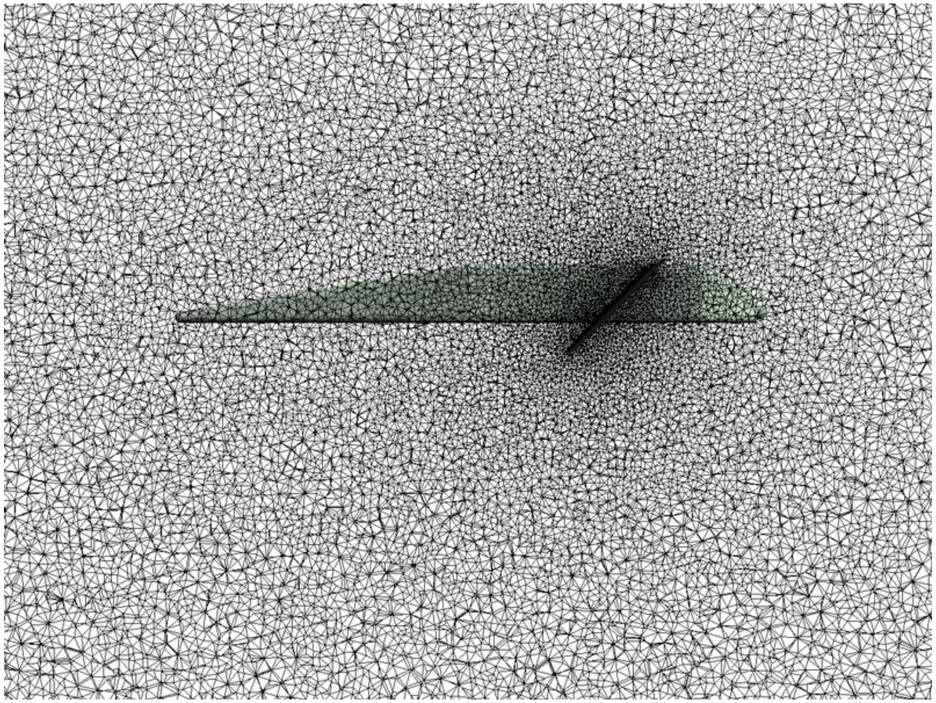


Body Volume
0.194 cm^3

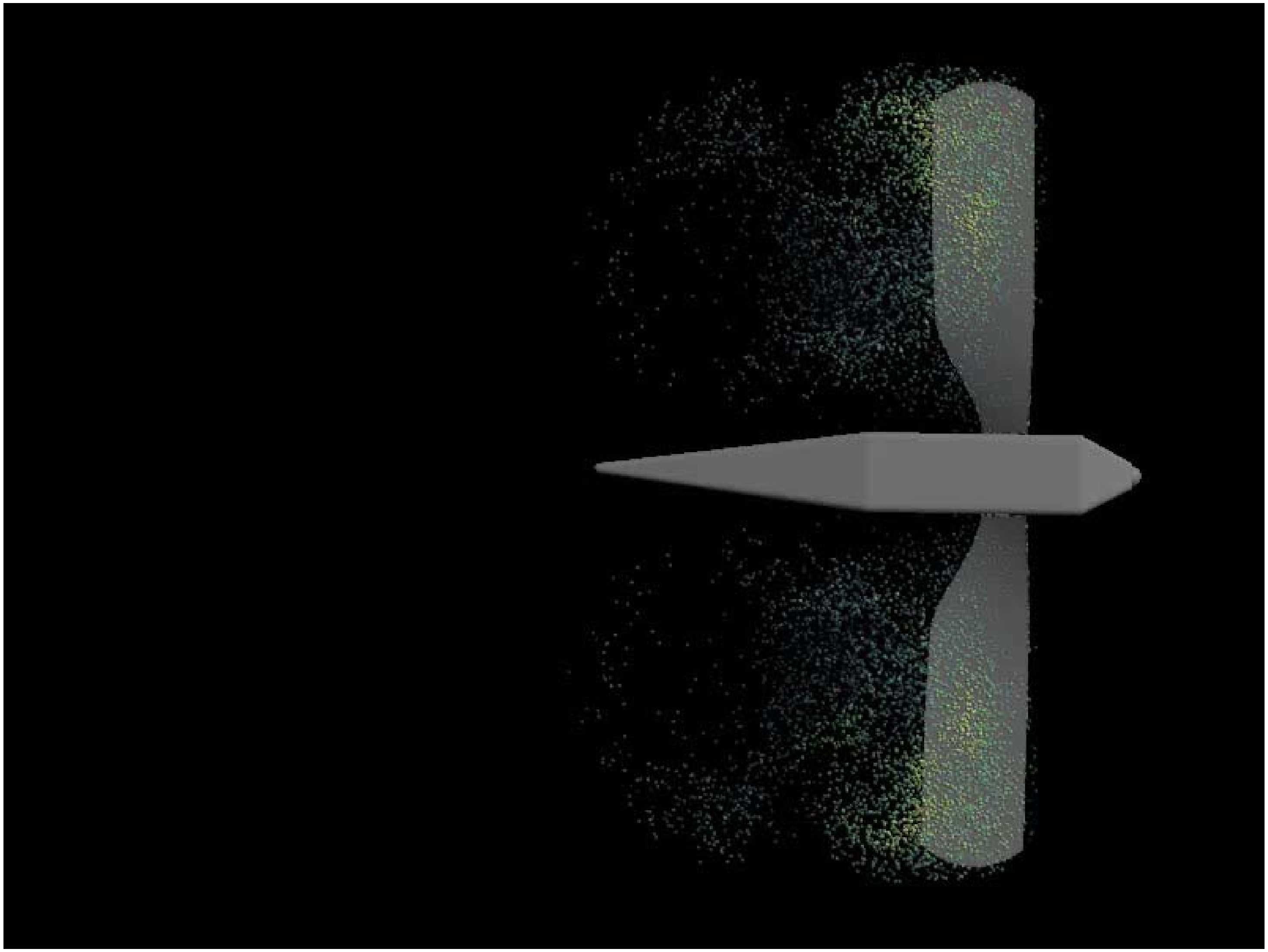
Wing Area
1.77 cm^2
(both wings)

- Wing flaps through a 120 degree arc
- Cruise 4.5 meters/second
  - Re = 1,552
- Cruise 2.25 meters/second
  - Re = 776
- Reynolds Numbers based on wing chord length of 0.5 cm

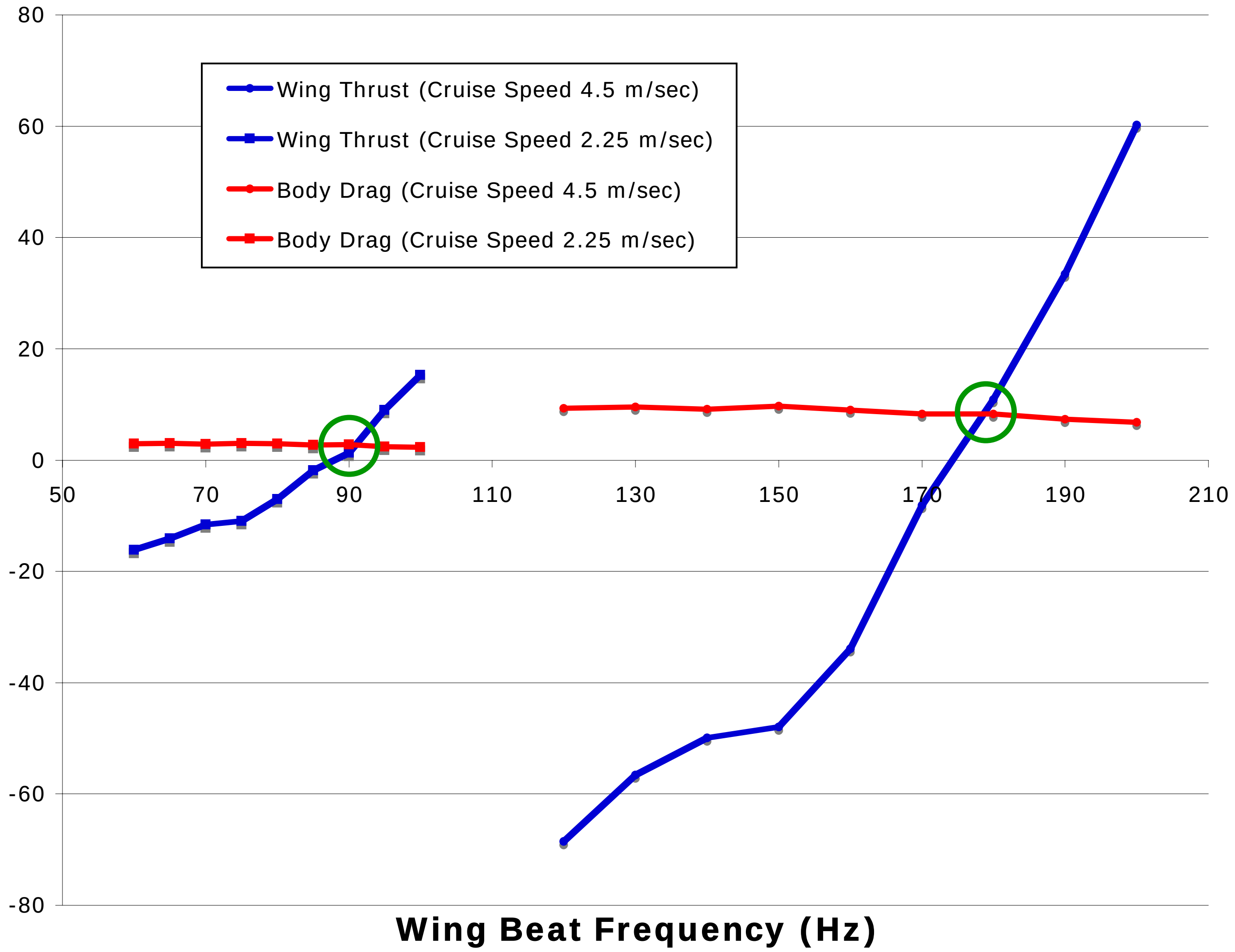
Micro-Air Vehicle (results)



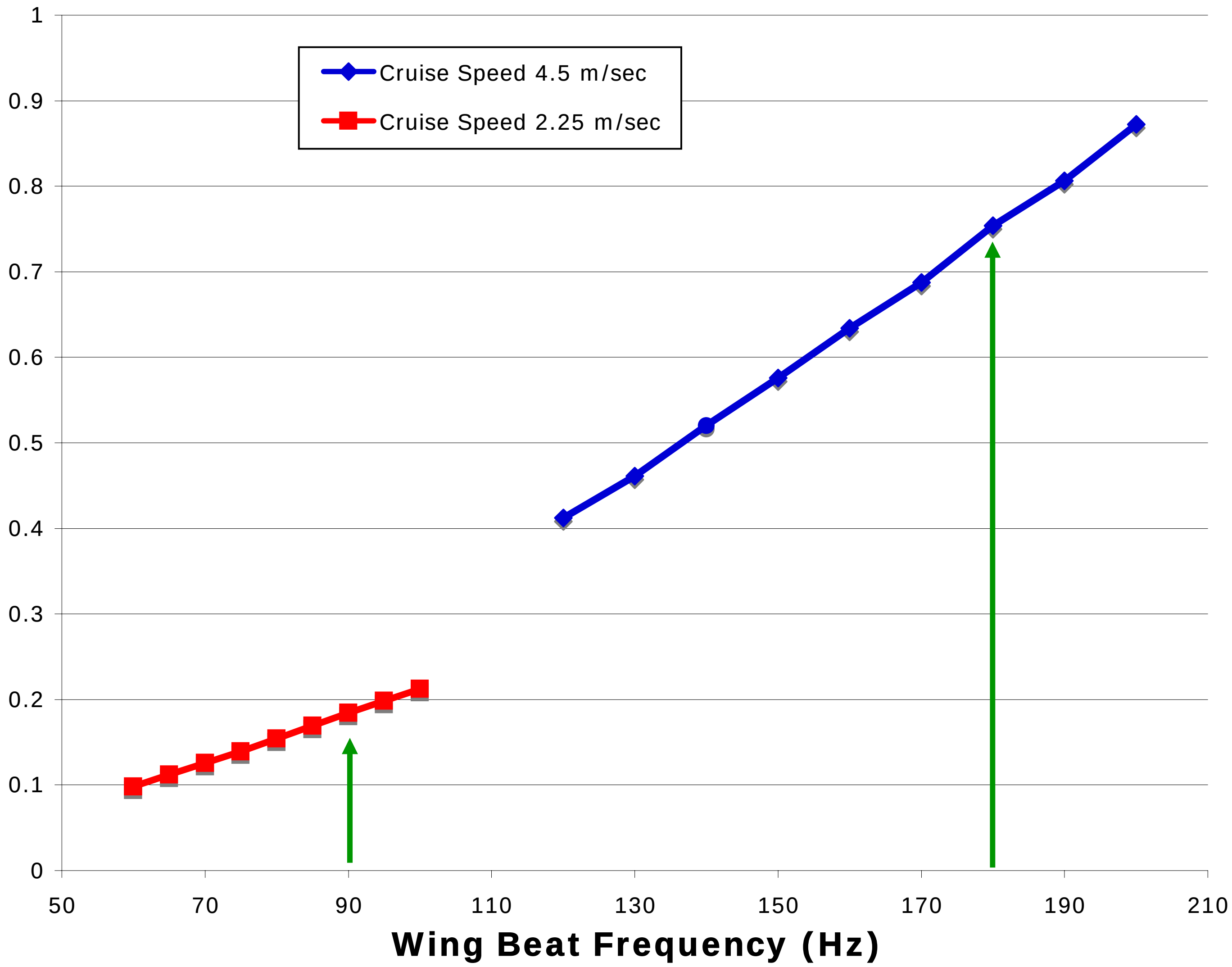
Micro-Air Vehicle (results)



Parametric Study of Wing-Beat Frequency



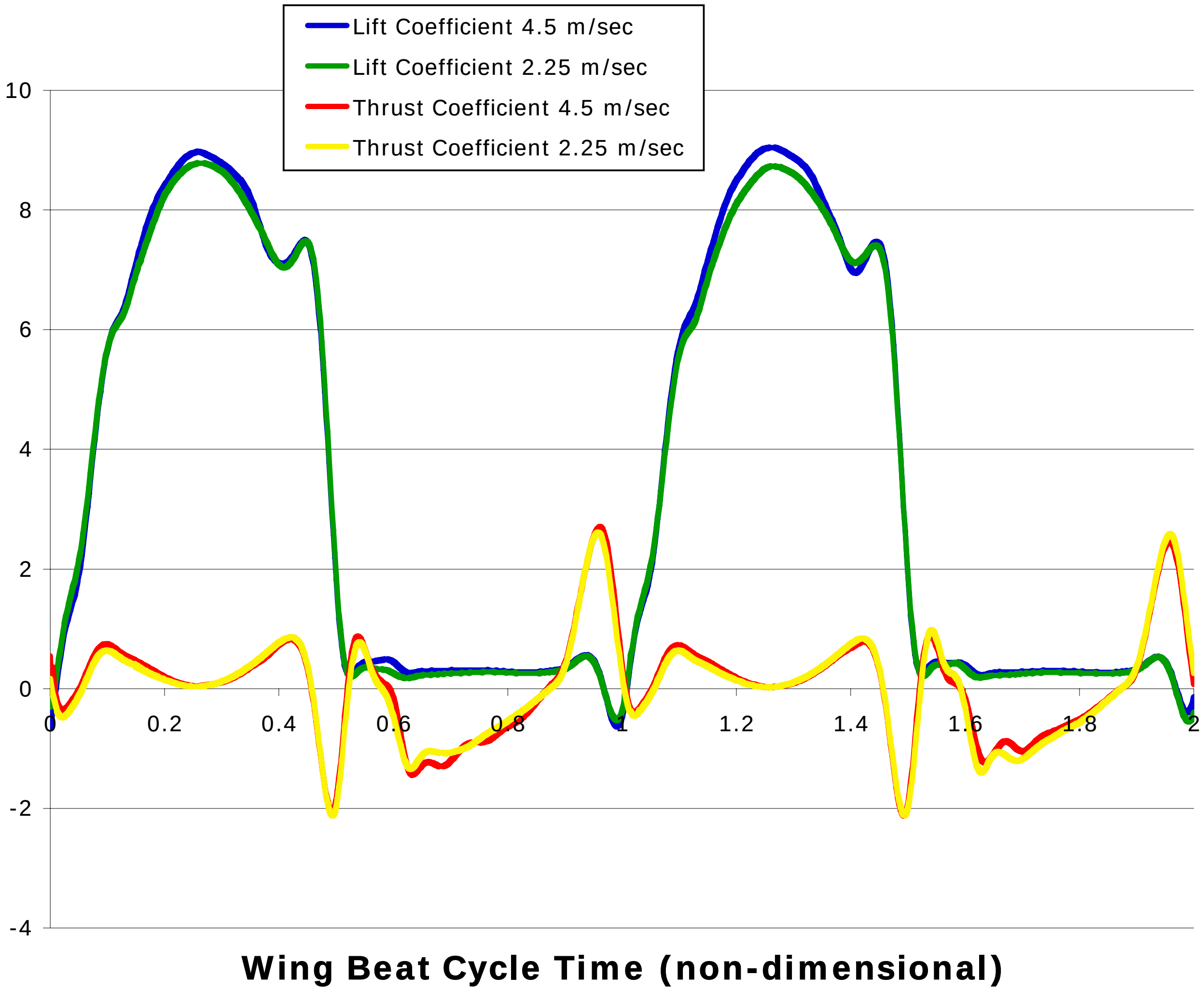
Parametric Study of Wing-Beat Frequency



AHPCRC

NETWORK COMPUTING SERVICES, INC.

Wing Lift and Thrust Time Histories

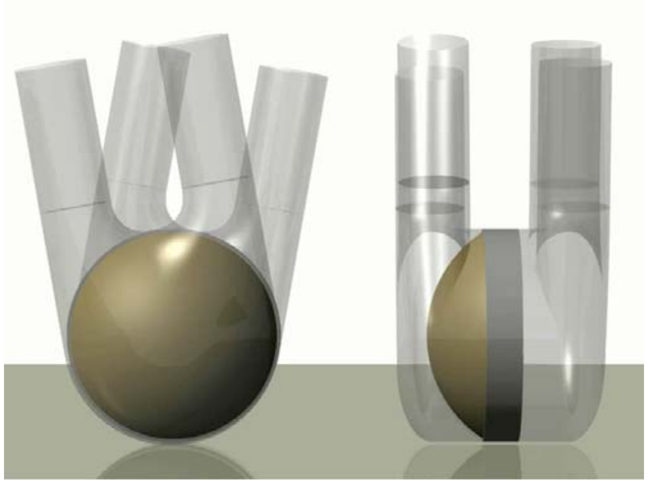


Force coefficients scaled with total wing area

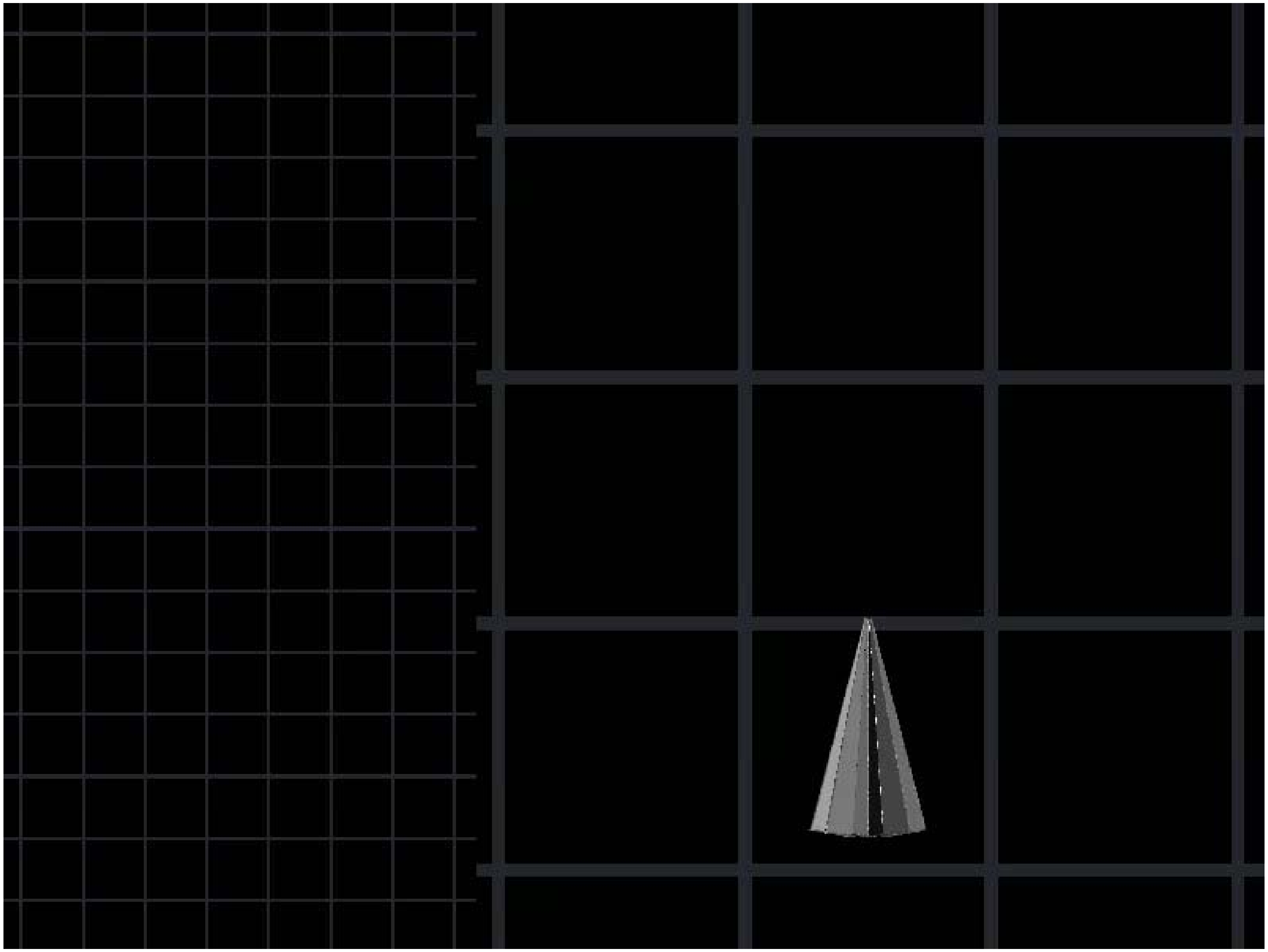
Other Targeted Applications

- Airdrop Systems
  - Broad, coupled application area with many aspects
  - Complicated requirements
    - Limited applicability of "traditional" numerical modeling based on a static mesh
  - Initial runs of the opening of a parachute

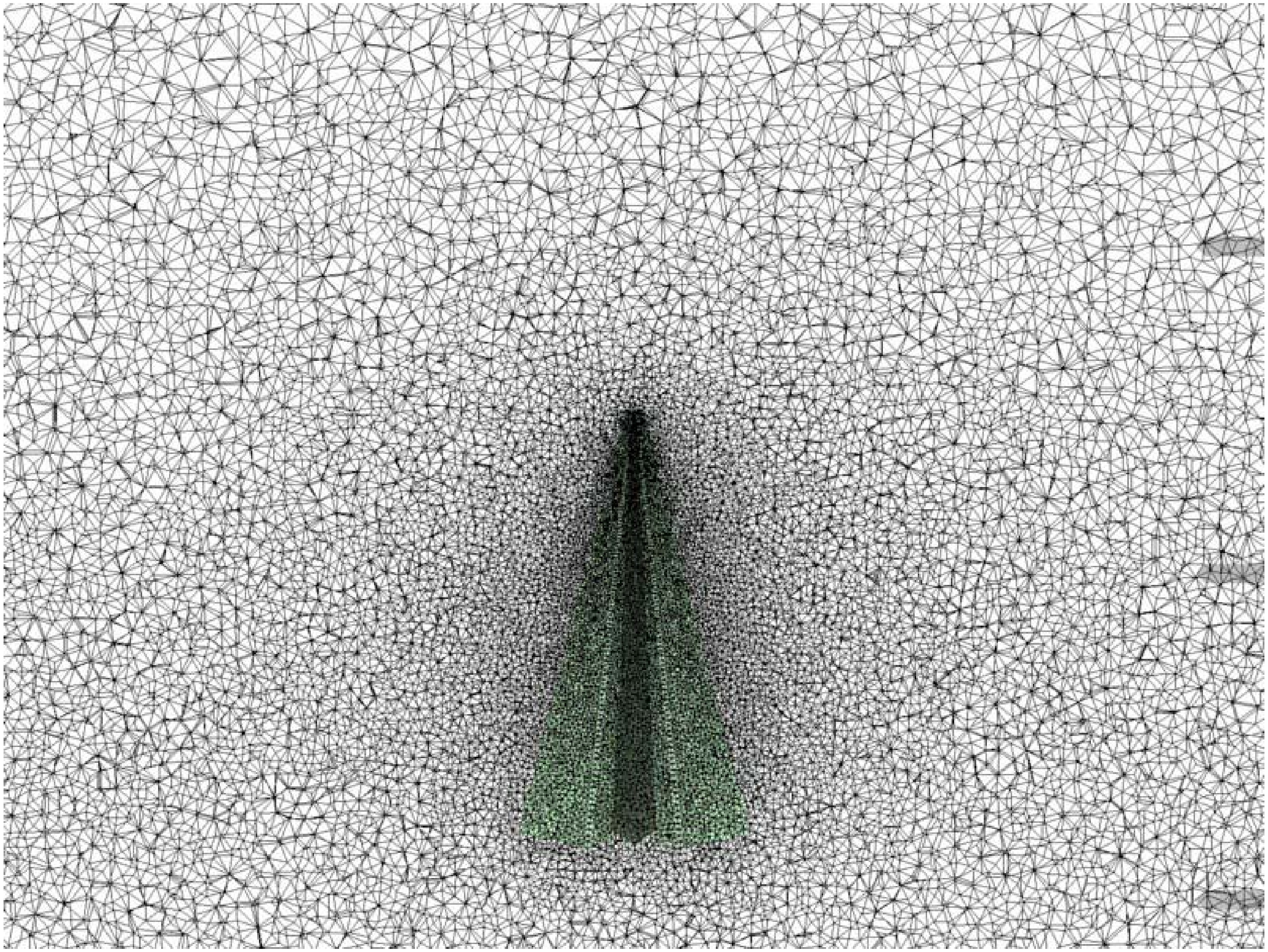
- Bio-Medical
  - Demonstrate DM-CFD for this new and challenging application area
  - Simulation of a hypothetical Total Artificial Heart



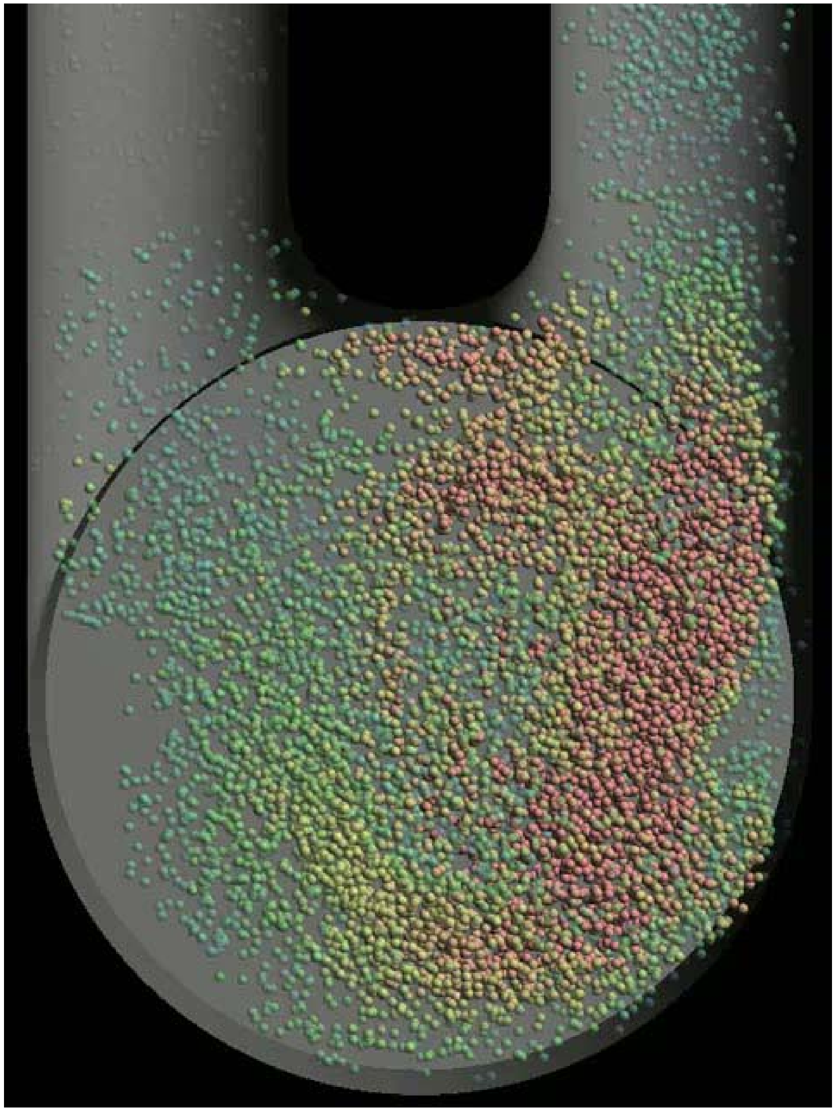
Airdrop Systems (results)



Airdrop Systems (results)



Total Artificial Heart (results)



Summary

- Dynamic-Mesh Methods allows for the simulation of complicated applications with moving components, including FSI applications
  - Possible extensions to large-deformation solid mechanics applications
- Technology is enabled by advanced parallel languages and hardware
- Demonstration of DM-CFD technology for several Army-relevant application areas
  - Validating of flapping-wing MAV application space
  - Great potential for airdrop systems applications

- Fluid-Structure Interaction coupling
  - Structural models for wings, fabrics, and other materials
  - Full 6 degrees-of-freedom dynamics (3 with symmetry)
- Enhanced scalability of XFlow's parallel methods and routines
- Increased robustness and capability of the Dynamic-Mesh algorithms
- More Applications