



AIAA-2001-0605

Computational Study of 3-D Flapping Foil Flows

Ravi Ramamurti and William C. Sandberg

Laboratory for Computational Physics and Fluid Dynamics

Naval Research Laboratory

Washington, D.C. 20375

**39th AIAA Aerospace Sciences
Meeting & Exhibit**

8-11 January 2001 / Reno, NV

Computational Study of 3-D Flapping Foil Flows

Ravi Ramamurti* and William Sandberg†

Naval Research Laboratory, Washington, D.C. 20375

Abstract

A finite element flow solver based on unstructured grids is employed for studying the unsteady flow past oscillating wings. In order to understand the basis of lift and thrust generation mechanisms, we have performed computational studies on the flapping wing of the fruit fly, *Drosophila*. The computational model is based on the experimental setup of Dickinson *et al.* [1]. Computations are performed for various phase angles between the rotation and translation motions and the time history of the unsteady forces are compared with the experiments. Good agreement is obtained for the thrust and drag forces. Also, a grid refinement study is performed to validate the computational results. The unsteady flow is discussed in detail.

Nomenclature

AR	=	aspect ratio of the wing ($2R^2 / S$)
\bar{c}	=	mean chord length ($2R / AR$)
C_D	=	coefficient of drag ($D / q_\infty S$)
C_T	=	coefficient of thrust ($T / q_\infty S$)
D	=	drag force ($\sqrt{(F_y^2 + F_z^2)}$)
F	=	force
n	=	frequency of flapping motion in Hz
q_∞	=	reference dynamic pressure ($1/2 \rho_\infty \bar{r}_2^2 U_t^2$)
r	=	distance from center of rotation
\bar{r}_2^2	=	second moment of the wing area
S	=	surface area of the wing
t	=	time
T	=	thrust ($-F_x$)
U_t	=	wing tip velocity ($2 \phi R n$)
\mathbf{v}	=	flow velocity
\mathbf{w}	=	mesh velocity
x, y	=	Cartesian coordinates
ϕ	=	wingbeat amplitude
ρ_∞	=	free stream density
<i>Subscript</i>		
∞	=	free stream value

Introduction

Flapping foil propulsion has received considerable attention in the past few years as an alternative to the propeller. This mode of propulsion which involves no body undulation, has many applications, such as submersibles propulsion, maneuvering and flow control which are of interest to the hydrodynamic community, and unconventional aerodynamics of Micro Aerial

Vehicles (MAV) and the study of aircraft flutter for the aerodynamic community.

Flapping foil propulsion is also important in the area of bio-fluid dynamics, for the study of propulsion in insects, birds and certain aquatic animals. Flying animals generate the lift and thrust as a consequence of the interaction of the flapping motions of the wings with the surrounding air. These animals also perform rapid maneuvers involving rapid plunging and pitching motions. Conventional steady state theories do not predict sufficient forces required for flight, as shown by Ellington [2]. Therefore, we need to understand the unsteady aerodynamics of flapping wings undergoing highly three-dimensional motions with widely varying geometries.

Experimental works on 2-D flapping foils have been carried out by Anderson [3] and Freymuth [4]. Computational studies have been performed by Jones and Platzer [5] and Ramamurti and Sandberg [6]. While 2-D wing section investigations can yield useful insights on the coupled pitching and heaving dynamics, nothing can be learned concerning the influence of the spanwise flow. It is therefore essential to carry out computations for actual 3-D insect wings. The three dimensional wing strokes of the insects can be divided into two translational phases and two rotational phases. During the translational phases, upstroke and downstroke, the wings move through the air with high angles of attack and during the rotational phases, pronation and supination, the wings rotate rapidly and reverse direction. Dickinson *et al.* [1] has studied the effects of the translational and rotational mechanisms of the wing in *Drosophila*. They explained the aerodynamics of insect flight by direct measurement of the forces produced by flapping wings by interaction of

This paper is a work of the United States Government and is not subject to copyright protection in the United States.

* Aerospace Engineer, Code 6410, Laboratory for Computational Physics and Fluid Dynamics, Senior Member AIAA.

† Deputy Director, Code 6401, Laboratory for Computational Physics and Fluid Dynamics.

three unsteady flow mechanisms. The ‘delayed’ stall mechanism is a translational mechanism, which in 2-D produces high lift in the initial phases of translation until eventual flow separation, and in 3-D the spanwise flow effectively prevents stall. The rotational circulation and wake capture are rotational mechanisms that depend mainly on the pronation and supination of the wing during stroke reversal. Walker and Westneat [7] have studied the kinematics of the fin motion in a class of fishes, namely the bird wrasse, experimentally. Liu and Kawachi [8] have studied the flow over a hovering hawkmoth, numerically.

In this study, we extend the 2-D pitching and heaving airfoil computations to 3 dimensions. This study will address the role of the rotational motion in detail. Also, the role of the leading edge vortex, and the interaction of the axial flow with this leading edge vortex are investigated. Computations are performed for various phase angles between the rotation and translation motions and the time history of the unsteady forces is compared with the experimental results. The flow solver we employ is a finite element based incompressible flow solver based on simple, low-order elements is employed. The simple elements enable the flow solver to be as fast as possible reducing the overhead in building element matrices, residual vectors etc. The governing equations are written in Arbitrary Lagrangian Eulerian form, which enables simulation of flow with moving bodies. For high Reynolds number flow cases, the mesh requirement is met by employing arbitrary semi-structured grids close to wetted surfaces and wakes. The details of the flow solver, the rigid body motion and adaptive remeshing are given by Ramamurti *et al.* [9] and are summarized next.

The Incompressible Flow Solver

The governing equations employed are the incompressible Navier-Stokes equations in Arbitrary Lagrangian Eulerian (ALE) formulation. They are written as

$$\frac{D\mathbf{v}}{Dt} + \mathbf{v}_a \cdot \nabla \mathbf{v} + \nabla p = \nabla \cdot \boldsymbol{\sigma} \quad (1a)$$

$$\frac{D\mathbf{v}}{Dt} = \frac{\partial \mathbf{v}}{\partial t} + \mathbf{w} \cdot \nabla \mathbf{v} \quad (1b)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (2)$$

Here p denotes the pressure, $\mathbf{v}_a = \mathbf{v} - \mathbf{w}$, the advective velocity vector (flow velocity \mathbf{v} minus mesh velocity \mathbf{w}), and the material derivative is with respect to the mesh velocity \mathbf{w} . Both the pressure p and the stress tensor $\boldsymbol{\sigma}$ have been normalized by the (constant) density ρ , and are discretized in time using an implicit time stepping procedure. It is important for the flow solver to be able to capture the unsteadiness of a flow field. The present flow solver is built as time-accurate from the onset, allowing local timestepping as an

option. The resulting expressions are subsequently discretized in space using a Galerkin procedure with linear tetrahedral elements. In order to be as fast as possible, the overhead in building element matrices, residual vectors, etc. should be kept to a minimum. This requirement is met by employing simple, low-order elements that have all the variables (u, v, w, p) at the same location. The resulting matrix systems are solved iteratively using a preconditioned gradient algorithm (PCG), as described by Martin and Löhner [10]. The flow solver has been successfully evaluated for both 2-D and 3-D, laminar and turbulent flow problems by Ramamurti *et al.* [11,12].

Rigid Body Motion And Adaptive Remeshing

In order to fully couple the motion of rigid bodies with the hydrodynamic or aerodynamic forces exerted on them, consistent rigid body motion integrators must be developed. The governing equations of motion for rigid bodies are well known and are given by Meirovitch [13] and for multiple bodies in relative motion by Sandberg [14]. In the present work, the pressure distribution on the surface is integrated to compute forces and moments at each time step and the equations of motion are advanced in time to produce self-consistent trajectories. A more detailed description of the equations and the incorporation of the rigid body motion in the numerical scheme for solving the fluid flow are described in Ramamurti *et al.* [12].

In order to carry out computations of the flow about oscillating and deforming geometries one needs to describe grid motion on a moving surface, couple the moving surface grid to the volume grid. The volume grid in the proximity of the moving surface is then remeshed, to eliminate badly distorted elements. A representative application requiring these gridding capabilities is the computation of the flow about pitching and heaving airfoils and the computation of vorticity shedding from the edges of oscillating foils. It is also essential for computing the flow past objects which are both accelerating and deforming. In deformations, the surface motion may be severe, leading, in the absence of remeshing, to distorted elements which in turn lead to poor numerical results. If the bodies in the flow field undergo arbitrary movement, a fixed mesh structure will lead to badly distorted elements. This means that at least a partial regeneration of the computational domain is required. On the other hand, if the bodies move through the flow field, the positions of relevant flow features will change. Therefore, in most of the computational domain a new mesh distribution will be required.

One approach to solve these problems is to add several layers around the moving bodies which move rigidly with the body. As the elements (or edges) move, their geometric parameters (shape-function derivatives, jacobians, etc.) need to be recomputed at every timestep. If the whole mesh is assumed to be in motion, then these geometric parameters need to be

recomputed globally. In order to save CPU-time, only a small number of elements surrounding the bodies are actually moved. The remainder of the field is then treated in the usual Eulerian frame of reference, avoiding the need to recompute geometric parameters. This may be accomplished in a variety of ways, of which the two most common are 1) by identifying several layers of elements surrounding the surfaces that move, and 2) by moving all elements within a certain distance from the surfaces that move. Both approaches have their advantages and disadvantages, and are therefore treated in more detail.

1) Layers of Moving Elements: In this case the elements moved are obtained by starting from the moving surfaces, and performing n number of passes over nearest neighbours to construct the n layers of elements that move. This procedure is extremely fast and works only with integer variables. On the other hand, for situations where the element size varies rapidly, the moving mesh region can assume bizarre shapes. This, in turn, may force many remeshings at a later stage. This type of procedure is most commonly used for Euler calculations [15,16].

2) Elements Within a Distance: This second approach requires the knowledge of the distance of a point from the moving surfaces. All elements within a prescribed distance from the moving surfaces are considered as moving. Although this procedure required more CPU-time when being built, it offers the advantage of a very smooth boundary of the moving mesh region. Moreover, by specifying two distances, the region close to the moving surfaces may be moved in the same way the surfaces move, while further away the mesh velocity is smoothed as before. This allows the movement of Navier-Stokes type grids that are very elongated, and hence sensitive to any kind of distortion, as shown by Ramamurti *et al.* [11].

Mesh Movement Algorithms

An important question from the point of view of mesh distortion and remeshing requirements is the algorithm employed to move the mesh. Assume that the mesh velocity on the moving surfaces of the computational domain is prescribed as follows

$$\mathbf{w} \big|_{\Gamma_m} = \mathbf{w}_0 \quad (3)$$

At a certain distance from these moving surfaces, as well as all the remaining surfaces the mesh velocity vanishes

$$\mathbf{w} \big|_{\Gamma_0} = 0 \quad (4)$$

The question now is, how to obtain a mesh velocity field \mathbf{w} in such a way that element distortion is minimized? A number of algorithms have been

proposed. They may be grouped together into the following categories.

- a) prescribing the mesh velocity analytically,
- b) smoothing the coordinates, and
- c) smoothing the velocity field.

Prescription Via Analytic Functions

In this case the mesh velocity is prescribed to be an analytic function based on the distance from the surface. Using heap-lists, as well as other optimal data structures, the distance from the surface may be obtained in $O(N \log N)$ operations, where N is the number of grid points. Given this distance r and the point on the surface closest to it $\mathbf{x} \big|_{\Gamma}$, the mesh velocity is given by

$$\mathbf{w} = \mathbf{w}(\mathbf{x} \big|_{\Gamma}) f(r) \quad (5)$$

The function $f(r)$ assumes the value of unity for $r = 0$, and decays to zero as r increases. This makes the procedure somewhat restrictive for general use, particularly if several moving bodies are present in the flow field. On the other hand, the procedure is extremely fast if the initial distance r can be employed for all times.

Smoothing of the Coordinates

In this case, we start with the prescribed boundary velocities. This yields a new set of boundary coordinates at the new time step.

$$\mathbf{x}^{n+1} \big|_{\Gamma} = \mathbf{x}^n \big|_{\Gamma} + \Delta t \mathbf{w} \big|_{\Gamma} \quad (6)$$

The mesh is smoothed based on these new values for the coordinates of the boundary points. In most cases to date, a simple spring analogy smoother has been employed. The new values for the coordinates are obtained iteratively via a relaxation or conjugate gradient scheme. As before, a good initial guess may be extrapolated via

$$\mathbf{x}_0^{n+1} = 2\mathbf{x}^n - \mathbf{x}^{n-1} \quad (7)$$

The smoothed mesh velocity is then given by

$$\mathbf{w} = \frac{1}{\Delta t} (\mathbf{x}^{n+1} - \mathbf{x}^n) \quad (8)$$

Most of the potential problems that may occur for this type of mesh velocity smoothing are due to initial grids that have not been smoothed. For such cases, the velocity of the moving boundaries is superposed to a fictitious mesh smoothing velocity which may be quite large during the initial stages of a run. Moreover, if spring analogy smoothers are employed, there is no guarantee that negative elements won't appear.

Smoothing of the Velocity Field

In this case, the mesh velocity is smoothed, based on the exterior boundary conditions given by Eqs. (3) and (4). The aim, as stated before, is to obtain a mesh velocity field \mathbf{w} in such a way that element distortion is minimized.

In this study, the smoothing of the coordinates was employed for the mesh movement with a specified number of layers of elements that move rigidly with the wing. In 2-D studies [6] the grid showed that the elements at the edge of the rigid layers were quite distorted after one cycle of oscillation. This is due to a residual mesh velocity that is present due to the non-convergence of the mesh velocity field. This will appear whether a spring-analogy is used or a Laplacian based smoothing is used.

In order to reduce the distortion of the mesh, the coordinates at the new time were obtained as a weighted average of the original grid point location at time $t = 0$ and the location of the point as if it moved rigidly with the body.

$$\mathbf{x}^{n+1} = \mathbf{x}^0 f(r) + \mathbf{x}_{rigid}^{n+1} (1 - f(r)), \quad (9)$$

where the weighting function is a simple linear function based on the distance from the center of rotation r , and is given by

$$\begin{aligned} f(r) &= 0 & \text{for } r < r_{\min}, \\ &= 1 & \text{for } r > r_{\max} \text{ and } (10) \\ &= \frac{(r - r_{\min})}{(r_{\max} - r_{\min})} & \text{for } r_{\max} > r > r_{\min}. \end{aligned}$$

The mesh velocity is then obtained using Eq. (8).

Layers of gliding points

The experimental setup of Dickinson [1] is made up of two wings immersed in a tank. In order to reduce the computational effort, a symmetry plane was introduced in the present simulations. Due to the proximity of the wing at the beginning of the downstroke, the rotation of the wing during the pronation phase, the normal component mesh velocity of the points on the symmetry plane can become non-zero. This would result in the points being pulled away from the symmetry plane. In order to avoid this problem, the points on the symmetry plane are allowed to glide along this plane. Similar technique has been employed for the simulation of torpedo launch [6] where the gap between the launch way tube and the torpedo was small. In that case, the closeness of the body resulted in a large distortion of the elements, and the use of gliding points reduced the number of remeshings required substantially.

Discussion Of Results

The configuration used for the flow simulations over the flapping wing is shown in Fig. 1. This is based on the experimental setup of Dickinson [1]. The planform

of the wing is according to the *Drosophila* wing and is 25cm long and 3.2mm thick. The experimental apparatus consisted of two wings immersed in a tank of mineral oil. The viscosity of the oil, the length of the wing and the frequency of the flapping motion were chosen to match the Reynolds number (Re) of a typical *Drosophila*, approximately 136. The Re for the present calculations is defined based on the mean chord of the wing \bar{c} and the mean wing tip velocity U_t (ignoring the forward velocity), as follows

$$Re = \frac{\bar{c} U_t}{\nu} \quad (11)$$

where $\bar{c} = 2R / AR$, $U_t = 2\phi n R$, R is the wing length, AR is the aspect ratio of the wing, n is the frequency of flapping motion, and ϕ is the wingbeat amplitude (peak to peak, in radians).

The kinematics of the wing motion is obtained from the experiments. Figure 2 shows the translational velocity of the wing tip, the rotational (angular) velocity of the wing for three different phases between the translational and rotational motions. The ‘advanced’ wing motion corresponds to the case where the wing rotation precedes the stroke reversal by 8% of the wingbeat cycle, the ‘symmetric’ wing motion corresponds to the wing rotation occurring symmetrically with respect to the stroke reversal and the ‘delayed’ wing motion corresponds to the rotation being delayed by 8% with respect to the stroke reversal. The wingbeat amplitude is 160° , the flapping frequency is $0.145Hz$ and the angle of attack at midstroke is approximately 40° during both upstroke and downstroke.

Symmetric case

The flow solver described here is employed to compute the flow past the *Drosophila* wing undergoing translation and rotation. First, an inviscid solution was obtained using a grid consisting of 178,219 points and 965,877 tetrahedral elements. An initial steady state solution was obtained in 1500 time steps. The unsteady solution using the prescribed kinematics, Fig. 2, is then obtained. The surface pressure on the wing is integrated to obtain the forces on the wing. The thrust and the drag forces are then computed as, $T = -F_x$ and $D = \sqrt{(F_y^2 + F_z^2)}$, respectively. These forces are compared to that obtained from the experiments of Dickinson *et al.* [1].

The unsteady computation was carried out for 5 cycles of oscillation. Figure 3a and b shows the comparison of thrust and drag forces, respectively, during one cycle of the wingbeat. The comparison is good in that the present computations capture the peak forces well. The mean thrust force is approximately 0.318N and the mean thrust coefficient \bar{C}_T is 1.317. The mean drag force is 0.375N and the drag coefficient

\bar{C}_d is 1.55. The trend of the variation of these forces during the translational phase of the wing is also predicted correctly, but the magnitude of the thrust force during the downstroke is higher than that of the experiments. In order to understand the different mechanisms occurring during the cycle, we can break the cycle into two rotational and two translational phases. The rotational phase near the beginning of the downstroke, called the pronation, occurs between time t_0 and t_3 (Fig. 3a). During this interval, the thrust decreases during t_0 to t_1 and increases up to a time t_2 . This behaviour can be explained through a rotational mechanism. The wing continuously rotates in the counterclockwise direction producing a circulation pointing nearly along the $+y$ direction. During t_0 – t_1 , the wing is translating in the negative z direction, resulting in a force pointing in the negative x direction, thus producing a peak in thrust at t_0 . If a rotational mechanism were alone present, the thrust should continue to decrease until time t_3 . But, the thrust force increases during the interval t_1 – t_2 . This happens after the wing changes direction at the start of each half stroke. Dickinson *et al.* [1] attribute this increase in thrust to a wake capture mechanism, in which the wing passes through the shed vorticity of the previous stroke. In the present study, we observe a leading edge separation bubble attached to the leading edge during the interval t_0 – t_1 . This leading edge vortex is created at the end of the upstroke. A possible explanation for the increase of thrust in the interval t_1 – t_2 , is that the wing moving through this wake benefits from the shed vorticity. When the wing moves through this vortex during the downstroke, it results in a stagnation region on the bottom of the wing, thus resulting in an increase in thrust. As the wing continues to move down, the separation point of this bubble moves back along the wing chord during the interval t_1 – t_2 , as can be seen in Fig. 4a–c. During the interval t_2 – t_3 , we observe a trailing edge separation bubble forming, see Fig. 5a–c. A similar separation region forms at the wing tip, as can be seen in Fig. 6a–c. From Figs. 5a and 6a, a large recirculation region can be seen in the wake of the wing. This separated flow from the wing tip and trailing edge results in a higher pressure on the top surface of the wing, and hence a reduction in the thrust. Another observation is that during the interval t_1 – t_3 , the magnitude of the translational acceleration of the wing decreases while that of the angular acceleration increases, Fig. 7. In the interval t_2 – t_3 , the magnitude of the translational acceleration is large enough to overcome the rotational effect, and when the angular acceleration becomes large enough, the rotational mechanism takes over, resulting in a reduction of thrust up to a time t_3 . During the interval t_3 – t_4 , the translational effect should result in a constant thrust, as the translational acceleration is almost constant during this period. The rotational effect produces an increase in thrust from t_3 – t_4 , with a plateau in the middle, which

occurs when the trailing edge vortex is shed from the trailing edge. Similar trends are observed during the supination phase prior to the beginning of the upstroke, t_4 – t_5 , and in the beginning of the upstroke, t_5 – t_7 .

Grid refinement study

In order to validate our computational results, we carried out a grid refinement study. The resolution of the grid in the vicinity of the wing is doubled. The computations were carried out using a grid consisting of approximately 238K points and 1.3M tetrahedral elements. The time step was also halved for this computation. The computed thrust forces are shown in Fig. 8. It can be seen that the agreement is very good and even the coarse grid produces adequate resolution.

Viscous effect

Next, to the study the effects of viscosity, a viscous computation was carried out, for $Re \approx 120$. Figure 9 shows the time history of the thrust and drag forces for the inviscid and the viscous cases. It is clear that the viscous effect is minimal and the thrust and drag forces are dominated by the translational and rotational mechanisms.

Effect of Phasing

Next, the phasing between the translational and rotational motions was varied. Figure 10 shows the comparison of forces for the case when the rotational motion precedes the stroke reversal. Again, the comparison with the experimental results is good. In this case, the peak in the thrust force is achieved prior to the beginning of the downstroke at $t = 10.76\text{sec}$ and is approximately 0.56N compared to a value of 0.47N for the symmetric case. This can be explained as the effect of the rotational mechanism as before. The rotational effect diminishes prior to the beginning of the downstroke, producing a negative thrust of 0.2N. After this point, the thrust increases up to $t = 11.98\text{sec}$. During this interval, the wing moves through the wake created during the upstroke, similar to the symmetric case, resulting in a high pressure on the bottom of the wing. The velocity vectors near the leading edge are shown in Fig. 11a–c. Also, during this interval both the translational and rotational accelerations are in phase. The peak thrust is approximately 0.48N compared to a value of 0.28N for the symmetric case. Thereafter, the combined effect of rotational and translational motions produce a minimum in the thrust until a second peak is achieved due to the rotational motion at $t = 14.23\text{sec}$, prior to the beginning of the upstroke. The mean thrust force is approximately 0.312N and the mean thrust coefficient \bar{C}_T is 1.291. The mean drag force is 0.457N and the drag coefficient \bar{C}_d is 1.89.

As the rotational motion is delayed with respect to the stroke reversal, the rotational motion does not produce any thrust prior to the beginning of the downstroke, as can be seen from Fig. 12a. The mean

thrust force is approximately 0.206N and the mean thrust coefficient \overline{C}_T is 0.854. The mean drag force is 0.457N and the drag coefficient \overline{C}_D is 1.496. In the initial period following the stroke reversal, the rotational effect continues to produce a negative thrust. Figures 13a and b show the velocity vectors near the leading edge. It can be seen that the leading edge vortex from the upstroke is not present after $t = 12.05$ sec. In this case, the high pressure on the bottom of the wing together with the orientation of the wing produces a reduction in thrust. After this, the combined translational and rotational mechanisms result in an increase in thrust. At a time $t = 12.8$ sec, we observe a plateau in the thrust, Fig. 12a. During this interval, the presence of a trailing edge vortex on the top surface, as shown in Fig. 14a and b increases the pressure on the top surface of the wing, resulting in a temporary loss of thrust, and when this vortex lifts off the trailing edge, thrust continues to increase.

Figures 15a-c show the magnitude of velocity along a plane at the beginning of the downstroke for the three cases, in the wake created by the wing. It can be seen that the velocities are greater for the advanced case and is reduced for the symmetric case and further reduced for the delayed rotation. The wing moving through the higher velocity fluid therefore produces an additional thrust in the advanced rotation, and the wing for the delayed case intercepts the flow at an angle that produces negative thrust. Similar velocity fields are also observed in the particle image velocimetry data of Dickinson *et al.* [1].

Summary and Conclusions

A finite element flow solver was employed to compute unsteady flow past a 3-D *Drosophila* wing undergoing flapping motion. The computed thrust and drag forces have been compared to the experimental study of Dickinson *et al.* [1], and the agreement is very good. A grid refinement study was performed to validate the computational results, and a grid independent solution was achieved. The effect of phasing between the translational and rotational motions was studied by varying the rotational motion prior to the stroke reversal. It was observed that in the case where the wing rotation is advanced with respect to the stroke reversal, the peak in the thrust forces are higher compared to the case when the wing rotation is in phase with the stroke reversal, and the peak thrust is reduced further when the wing rotation is delayed. As suggested by Dickinson *et al.* [1], we observe that the rotational mechanism is important and the combined translational and rotational mechanisms are necessary to accurately describe the force time histories and unsteady aerodynamics of flapping wings.

Acknowledgements

This work was supported by the Office of Naval Research through the Tactical Electronic Warfare

Division Micro Air Vehicles Program of the Naval Research Laboratory. The authors would like to thank Prof. Rainald Löhner of the George Mason University for his support and helpful discussions throughout the course of this work. The computations carried out for this work were supported in part by a grant of HPC time from the DoD HPC centers, ARL MSRC SGI-O2K and NRL SGI-O2K.

References

1. Dickinson, M.H., Lehmann, F.-O. and Sane, S., "Wing Rotation and the Aerodynamic Basis of Insect Flight," *Science*, Vol. 284, pp. 1954-1960, 1999.
2. Ellington, C.P., "The Aerodynamics of hovering insect flight. IV. Aerodynamic Mechanisms," *Phil. Trans. R. Soc. Lond. B* 305, pp. 79-113, 1984.
3. Anderson, J.M., "Vorticity Control for Efficient Propulsion," Ph. D. Dissertation, Massachusetts Institute of Technology, February 1996.
4. Freymuth, P., "Thrust generation by an airfoil in hover modes," *Experiments in Fluids* 9, pp. 17-24, 1999.
5. Jones, K.D. and Platzer, M.F., "Numerical computation of Flapping-Wing propulsion and Power Extraction," *AIAA Paper* No. 97-0826, January 1997.
6. Ramamurti, R. and Sandberg, W.C., "Simulation of Flow about Flapping Airfoils using a Finite Element Incompressible Flow Solver," to appear in *AIAA J.*, Vol. 39, No. 2, pp. 253-260, 2001.
7. Walker, J.A. and Westneat, M.W., "Labriform propulsion in Fishes: Kinematics of Flapping Aquatic Flight in Bird Wrasse *Gomphosus Varius* (Labridae)," *J. exp. Biol.* 200, pp. 1549-1569, 1997.
8. Liu, H. and Kawachi, K., "A Numerical Study of Insect Flight," *J. comp. Phy.* 146, pp. 124-156, 1998.
9. Ramamurti, R., Sandberg, W.C. and Löhner, R., "Simulation of a Torpedo Launch Using a 3-D Incompressible Finite Element Flow Solver and Adaptive Remeshing," *AIAA Paper* No. 95-0086, January 1995.
10. Martin, D. and Löhner, R., "An Implicit Linelet-Based Solver for Incompressible Flows," *AIAA Paper* No. 92-0668, 1992.
11. Ramamurti, R. and Löhner, R., "Evaluation of an Incompressible Flow Solver Based on Simple Elements," *Advances in Finite Element Analysis in Fluid Dynamics*, FED Vol. 137, Editors: M.N Dhaubhadel et al., ASME Publication, New York, 33-42, 1992.
12. Ramamurti, R., Löhner, R. and Sandberg, W.C., "Evaluation of Scalable 3-D Incompressible Finite Element Solver," *AIAA Paper* No. 94-0756, 1994.
13. Meirovitch, L., 1970, *Methods of Analytical Dynamics*, McGraw-Hill, New York.

14. Sandberg, W.C., "The Estimation of Ship Motion Induced Loads," *SNAME*, 4th Ship Technology and Research, pp. 347-351, 1979.
15. Löhner, R., "An Adaptive Finite Element Solver Transient Problems with Moving Bodies," *Comp. Struct.*, Vol. 30, pp. 303-317, 1988.

16. Löhner, R. and Baum, J.D., "Three-Dimensional Store Using a Finite Element Solver and Adaptive Remeshing," *AIAA Paper* No. 91-0602, 1991.

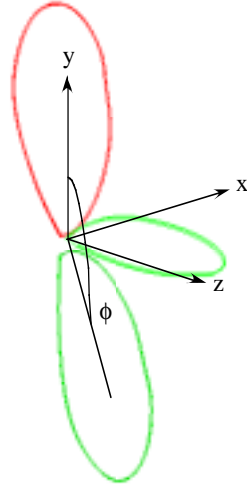


Fig. 1. Schematic of the Flapping *Drosophila* wing

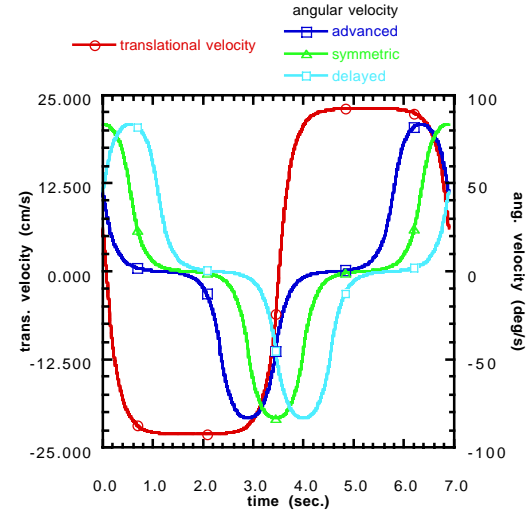
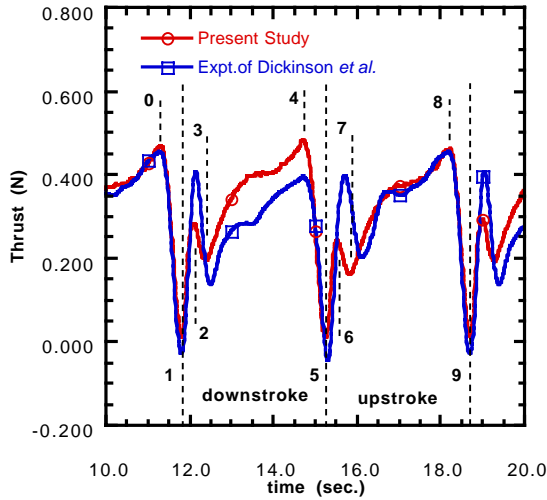
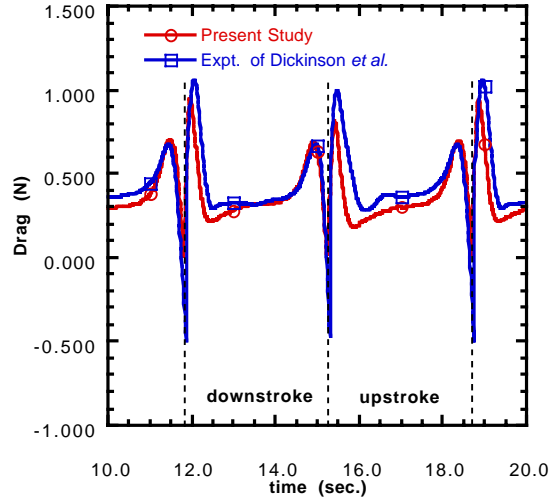


Fig. 2. Kinematics of the Flapping wing



a. thrust



b. drag

Fig. 3. Comparison of time history of thrust and drag forces

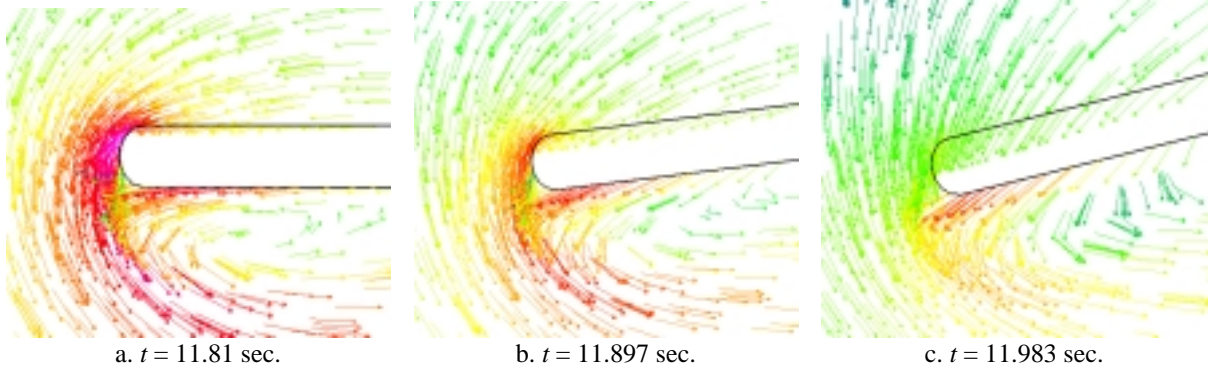


Fig. 4. Velocity vectors near the leading edge on a plane $y = 10\text{cm}$

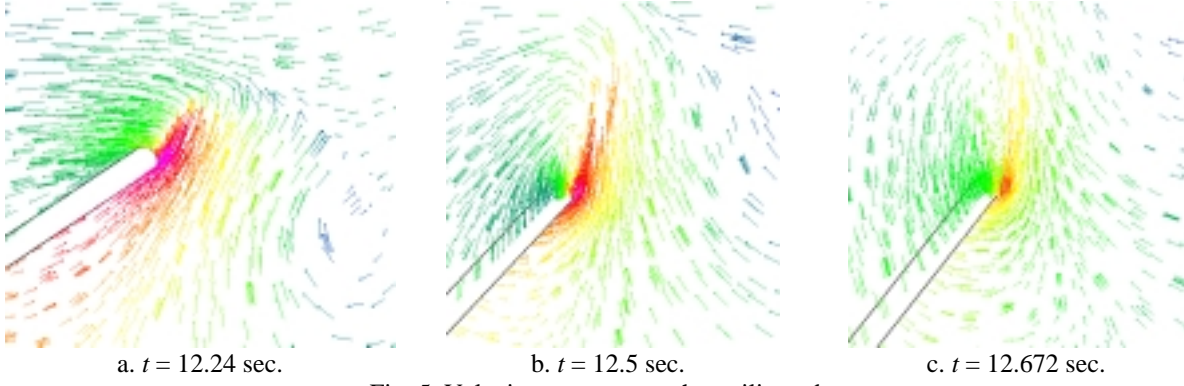


Fig. 5. Velocity vectors near the trailing edge

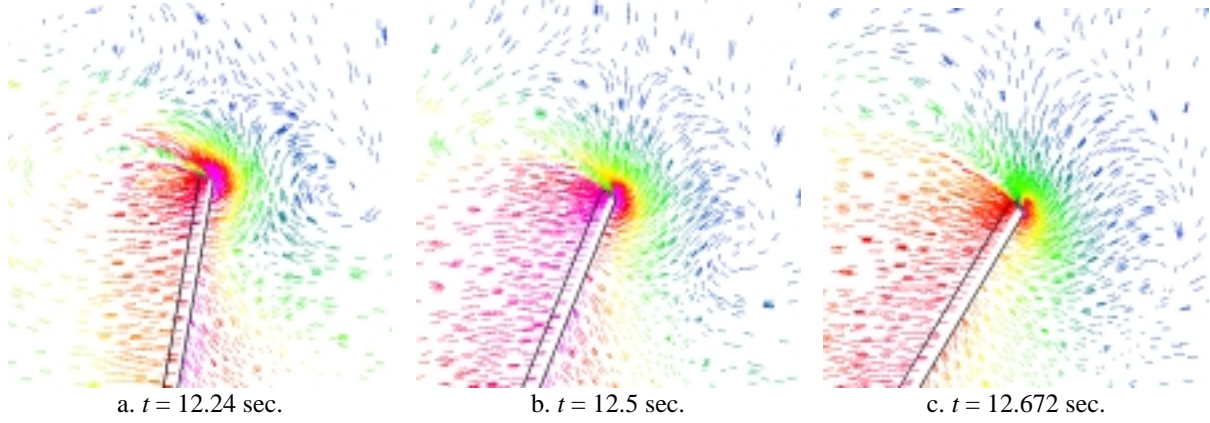


Fig. 6. Velocity vectors near the wing tip

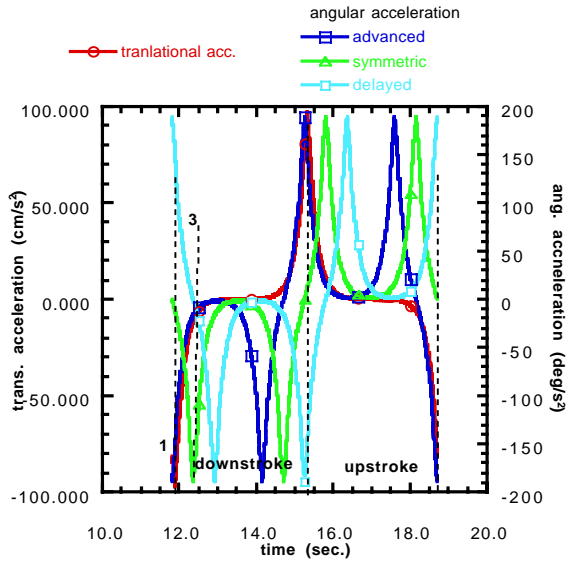


Fig. 7. Translational and angular accelerations of the wing

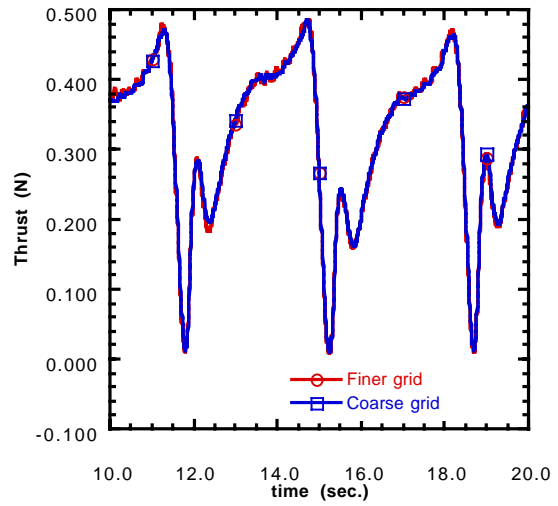


Fig. 8. Effect of grid refinement

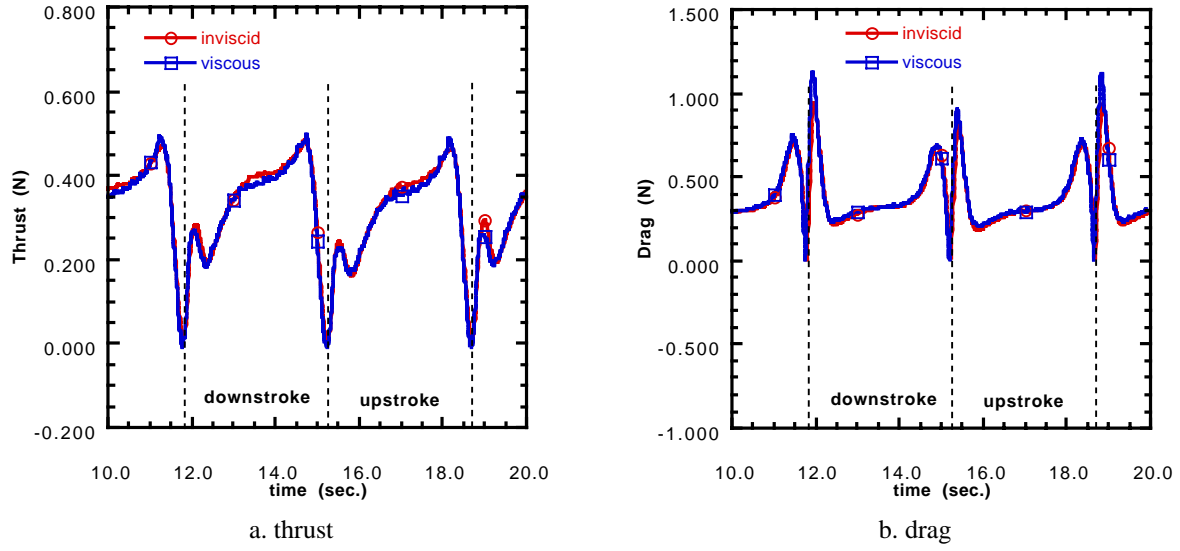


Fig. 9. Effect of viscosity

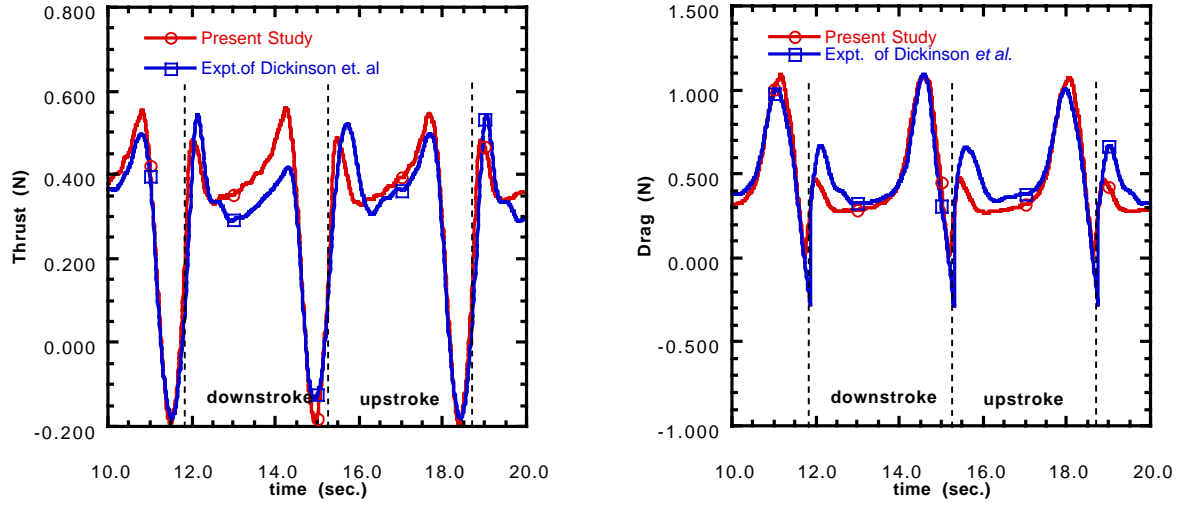


Fig. 10. Comparison of thrust and drag forces for the 'advanced' rotation

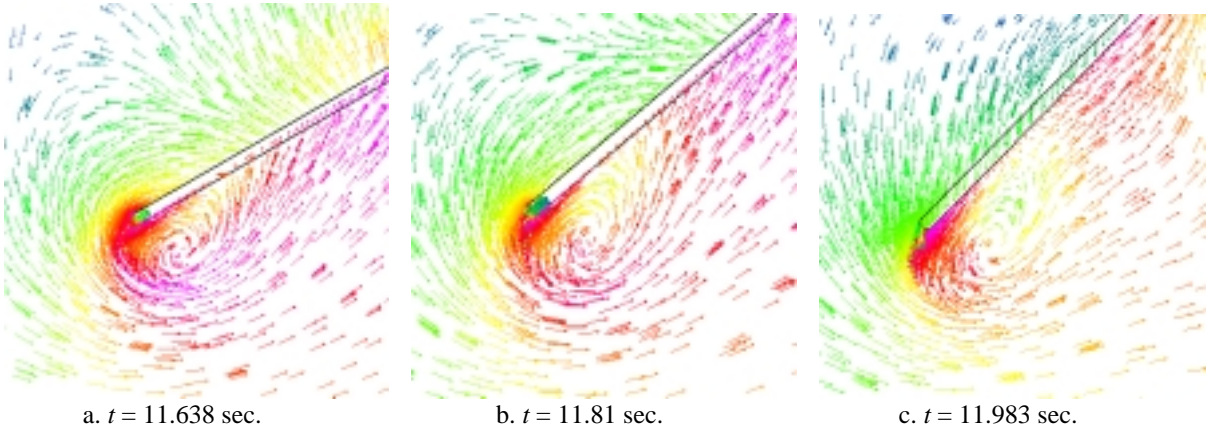


Fig. 11. Velocity vectors near the leading edge on a plane $y = 10\text{cm}$

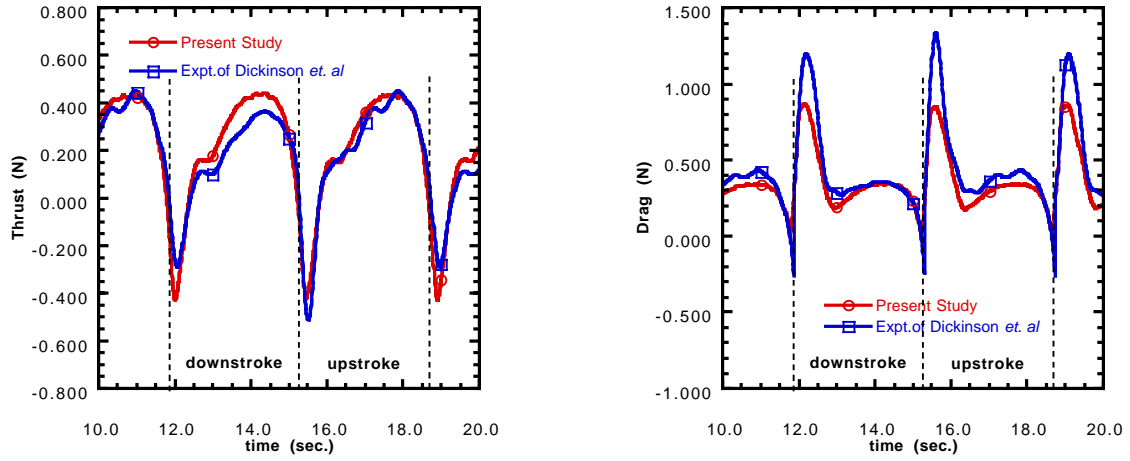


Fig. 12. Comparison of thrust and drag forces for the 'delayed' rotation

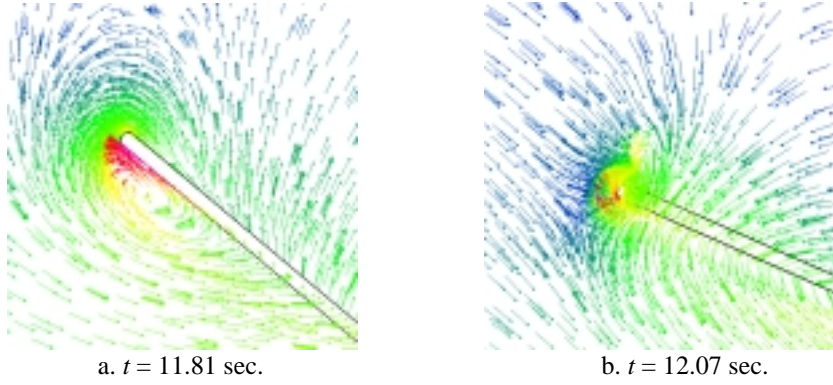


Fig. 13. Velocity vectors near the leading edge on a plane $y = 10\text{cm}$

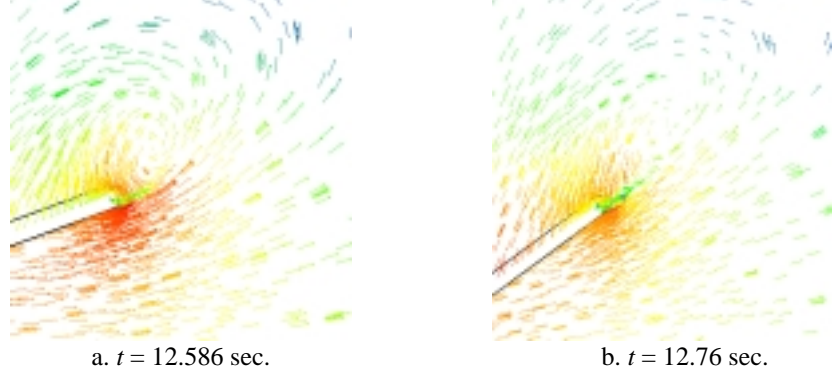


Fig. 14. Velocity vectors near the trailing edge on a plane $y = 10\text{cm}$

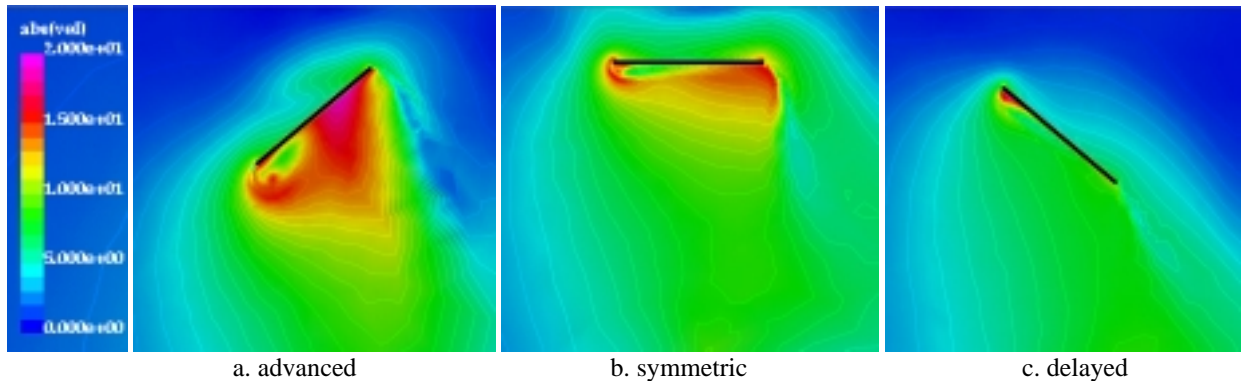


Fig. 15. Magnitude of velocity in the wake of the wing at the beginning of the downstroke