



**AIAA 96-0055**  
**NUMERICAL WAKE VISUALIZATION**  
**FOR AIRFOILS UNDERGOING FORCED**  
**AND AEROELASTIC MOTIONS**

K.D.Jones  
Naval Postgraduate School  
Monterey, CA

K.B.Center  
University of Colorado  
Boulder, CO

**34th Aerospace Sciences**  
**Meeting & Exhibit**  
January 15-18, 1996 / Reno, NV

# NUMERICAL WAKE VISUALIZATION FOR AIRFOILS UNDERGOING FORCED AND AEROELASTIC MOTIONS

K. D. Jones<sup>†</sup>

Naval Postgraduate School  
Monterey, California

K. B. Center<sup>‡</sup>

University of Colorado  
Boulder, Colorado

## Abstract

A *virtual wind tunnel* is developed by combining a fast, time-stepping flow solver with an interactive animation interface. Inviscid, incompressible flow solutions are provided by an unsteady, potential-flow code with arbitrary airfoils undergoing forced or aeroelastic motions. Aeroelastic response is predicted by a two-degree-of-freedom spring/mass system modeling the structural dynamics of a flexible wing. A boundary layer code may be used to predict viscous flow effects such as transition and the onset of flow separation. The aeroelastic algorithm is coupled with an interactive graphical animation front end, enabling visualization and measurement of the unsteady wake and surrounding flow field. To further enhance the *virtual wind tunnel* look and feel, an assortment of tools is provided to mimic smoke/dye-injection, wake-vorticity tagging and flow anemometry. The complete package is used both as a research and an instructional tool, providing insight into the complex, unsteady vortical structures behind moving airfoils. Solutions compare well with experimental data while the animation allows a more thorough investigation of the development and evolution of the wake structures in time.

## Nomenclature

$A$  = plunge amplitude in terms of  $c$   
 $c$  = chord length  
 $C_d$  = drag coefficient per unit span,  $D/(q_\infty c)$   
 $C_l$  = lift coefficient per unit span,  $L/(q_\infty c)$   
 $C_m$  = pitching moment coef. per unit span,  $M/(q_\infty c^2)$

$D$  = drag per unit span  
 $h$  = bending displacement (positive downward)  
 $I_\alpha$  = moment of inertia about the elastic axis  
 $k$  = reduced frequency,  $\omega c/V_\infty$   
 $k_\alpha$  = reduced natural pitching frequency  
 $k_h$  = reduced natural plunging frequency  
 $K_h$  = spring constant for plunging  
 $K_\alpha$  = spring constant for pitching  
 $L$  = lift per unit span  
 $m$  = mass of the wing per unit span  
 $M$  = pitching moment per unit span  
 $q_\infty$  = freestream dynamic pressure  $\frac{1}{2}\rho_\infty V_\infty^2$   
 $q_{jk}$  = source strength on panel  $j$  at  $t_k$   
 $S_\alpha$  = static moment,  $x_\alpha m$   
 $t$  = time  
 $V_p$  = maximum plunge velocity,  $Ak$   
 $V_\infty$  = freestream velocity magnitude  
 $x_p$  = leading edge to elastic axis distance  
 $x_\alpha$  = elastic axis to center of mass distance  
 $\alpha$  = angle of attack  
 $\Delta_k$  = length of wake panel at  $t_k$   
 $\gamma_k$  = circulation on single panel at  $t_k$   
 $\gamma_{wk}$  = circulation on wake panel at  $t_k$   
 $\Gamma_k$  = total circulation about foil at  $t_k$   
 $\theta_k$  = wake panel deflection angle at  $t_k$   
 $\omega$  = circular frequency  
 $\omega_h$  = uncoupled natural bending freq.,  $\sqrt{K_h/m}$   
 $\omega_\alpha$  = uncoupled natural torsional freq.,  $\sqrt{K_\alpha/I_\alpha}$   
 $\rho_\infty$  = freestream density  
 $\tau$  = nondimensional time,  $tV_\infty/c$   
 $(\cdot)$  = differentiation with respect to  $t$   
 $(\cdot)'$  = differentiation with respect to  $\tau$

## Introduction

Numerical solution techniques have been used to solve complicated fluid dynamics problems for many decades, but the field of Computational Fluid Dynamics (CFD) didn't come of age until the rather recent development of the digital computer. With the ad-

---

<sup>†</sup> NRC Research Associate, Member, AIAA

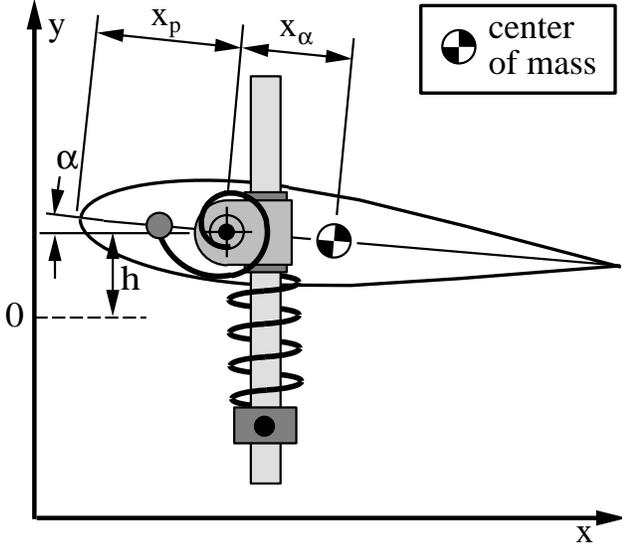
<sup>‡</sup> Postdoctoral Research Associate, Member, AIAA

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



Note, implementation of this approach requires an iterative scheme, since the velocity direction and magnitude used to define the wake panel are not initially known. Details of the code are given in Ref. 1, and the code is evaluated through comparisons with theoretical studies and Navier-Stokes simulations in Refs. 2-5,8 and 9.

**Airfoil Motion** In the present implementation the motion of the airfoil may either be prescribed as a forced pitching or plunging motion, or it may be determined by the instantaneous aerodynamic forces and moments acting on it using a two-degree-of-freedom spring/mass system (Fig. 2) to model the torsional and bending stiffness of a wing.



**Fig. 2.** Schematic of the spring/mass system.

The equations governing this aeroelastic motion are

$$m\ddot{h} + S_\alpha\ddot{\alpha} + m\omega_h^2 h = -L \quad (2)$$

and

$$S_\alpha\ddot{h} + I_\alpha\ddot{\alpha} + I_\alpha\omega_\alpha^2 \alpha = M, \quad (3)$$

where the dots denote differentiation with respect to time.

Nondimensionalizing the system using reference values of length= $c$ , velocity= $V_\infty$ , time= $c/V_\infty$  and mass= $c^2\pi\rho_\infty/4$ , and rewriting the system in matrix notation, one obtains

$$[\mathbf{M}]\{X\}'' + [\mathbf{k}]\{X\} = \{F\} \quad (4)$$

where

$$[\mathbf{M}] = \begin{bmatrix} m & S_\alpha \\ S_\alpha & I_\alpha \end{bmatrix}, \quad [\mathbf{k}] = \begin{bmatrix} m\omega_h^2 & 0 \\ 0 & I_\alpha\omega_\alpha^2 \end{bmatrix},$$

$$\{X\} = \begin{Bmatrix} h \\ \alpha \end{Bmatrix} \quad \text{and} \quad \{F\} = \frac{2}{\pi} \begin{Bmatrix} -C_l \\ C_m \end{Bmatrix},$$

and where the primes denote differentiation with respect to nondimensional time,  $\tau$ .

Equation (4) is a system of two, coupled, second-order, nonlinear, differential equations, with the nonlinearity arising from the influence of the deflected wake on the foil. Single-degree-of-freedom simulations are performed by setting  $S_\alpha = 0$  and either  $m = \infty$  and  $\omega_h = 0$  or  $I_\alpha = \infty$  and  $\omega_\alpha = 0$  for pitching-only or plunging-only motions, respectively.

Equation (4) is advanced in time by inverting the system, yielding

$$\{X\}'' = [\mathbf{M}]^{-1}\{F\} - [\mathbf{M}]^{-1}[\mathbf{k}]\{X\}, \quad (5)$$

then rewriting the result as a system of two coupled, first-order equations

$$\begin{aligned} \{X\}' &= \{Y\} \\ \{Y\}' &= [\mathbf{M}]^{-1}\{F\} - [\mathbf{M}]^{-1}[\mathbf{k}]\{X\}, \end{aligned} \quad (6)$$

and, finally, integration is performed using either a 2nd-order modified Euler scheme or a 4th-order Runge-Kutta scheme.

### Graphical Interface

Prior to the development of the graphical interface, efforts were made to output data from the panel code in a format that could later be viewed with widely accepted visualization software packages (e.g., Plot3D, FAST). Several disadvantages were inherent to this approach. Not only was the presentation of data restricted by the available options in the software, but it was impractical to write the data in the format of packages that are geared toward Eulerian-mesh-based data. The unsteady, Lagrangian-mesh-based data generated by the panel code required special treatment, particularly because available software proved grossly inefficient for analysis.

The solution was to write a graphical interface geared specifically toward the data being computed, namely the strength and sign of the vortex elements, and their unsteady positions in the fixed two-dimensional coordinate system. Other data of interest, such as the airfoil surface pressure and aerodynamic coefficients, were readily available within the computational code, and it seemed natural to directly extract that information and present it in a way most useful for data analysis.

As the project evolved it became clear that much more information could be extracted from the flow

with minimal computational overhead. An assortment of *tools* for the visualization and measurement of selected flow features was developed, in effect, creating a *virtual wind tunnel*.

The resulting interface runs on the Silicon Graphics platform, utilizing the IRIS Graphics Library. The program provides an instantaneous display of results during flow computation with direct interaction by the user. However, for large computations or when running on slower machines the computational frame-rate may be something less than real-time. To visualize the motion at a more fluent frame-rate a replay utility with limited features is provided to reanimate previously computed solutions.

**General Layout** At program startup a fullscreen window is opened with viewports distributed on the workspace as depicted in Fig. 3.

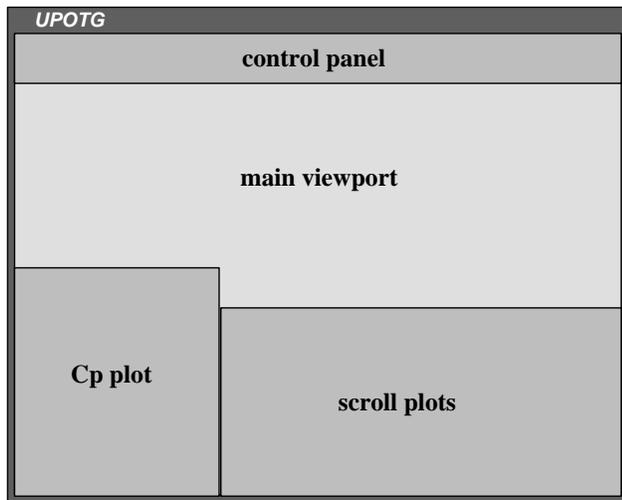


Fig. 3. Workspace layout.

The airfoil motion and the wake dynamics are shown in the main viewport. The image may be scaled and positioned as desired using mouse or keyboard inputs. Many variations in the portrayal of the wake structure are possible, as will be discussed in a later section, but a typical snapshot is shown in Fig. 4.

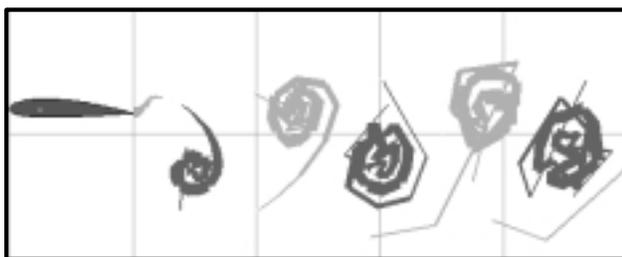


Fig. 4. Typical wake image from the main viewport.

A viewport may be brought up in the lower left corner of the workspace which displays a time-accurate representation of surface pressure coefficient on the airfoil upper and lower surfaces superimposed on a detailed schematic of the airfoil dynamics (Fig. 5). In situations where the airfoil undergoes pitching dynamics, the pivot point (elastic axis) is indicated on the moving airfoil, and an indexed angular grid is drawn. Similarly, for two-degree-of-freedom aeroelastic simulations, the center of mass is also indicated on the moving foil. The limits of the  $C_p$  plot may be interactively adjusted. When the mouse is placed within the  $C_p$  viewport, mouse activated adjustment buttons appear at the graph end points.

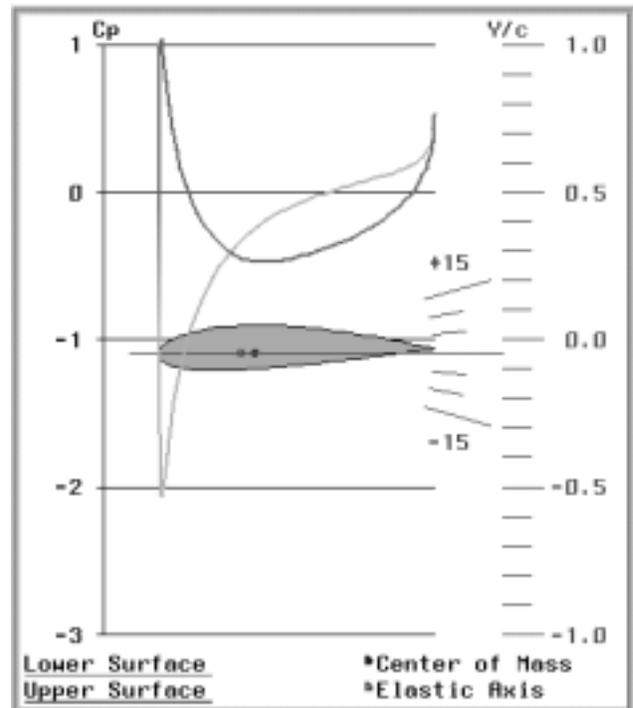


Fig. 5. Typical image of the  $C_p$  viewport.

Scrolling plots depicting time histories of various performance parameters may be displayed in the lower right corner of the workspace, offering a view into the time variance and phase relationships of the integrated lift, drag and moment coefficients and the geometric parameters,  $h$  and  $\alpha$  (Fig. 6). As with the  $C_p$  plot, the limits of each of the scroll graphs may be adjusted individually, with the limit buttons appearing only when the mouse is within the graph to be adjusted.

All data presented in the  $C_p$  viewport and the scroll plots may be written to files for external processing. Data archiving procedures and available formats are discussed in a later section.

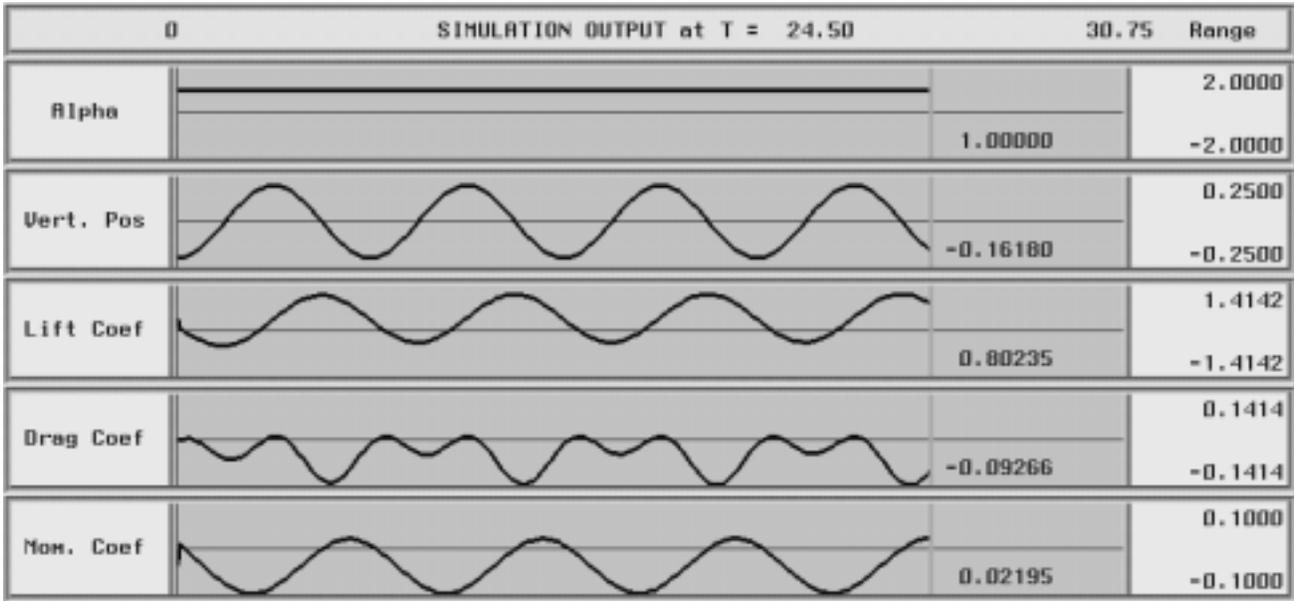


Fig. 6. Typical image of the scroll plots.

**Program Control** The primary objective of such an interface is to make the graphical manipulations and computations as transparent and user friendly as possible, such that the user can fully concentrate on the real problem; analyzing the flow. Control of the interface is accomplished through mouse-driven graphical controls which appear at the top of the workspace when the cursor is within that region (keyboard commands are available, but seldom as convenient). The control panel is divided into three primary sections for program control (Fig. 7), program execution (Fig. 8a) or movie playback while in the replay mode (Fig. 8b) and display control (Fig. 9).

The program control section includes exit, edit and I/O controls. The edit key invokes the graphical editor for modifying the control parameters, and the I/O key brings up a submenu for reading and writing selected files. Details of these options are discussed in later sections.

While in the computation mode the solution may be paused at a given frame or reinitialized to the first frame. While in the replay mode VCR-like controls for play/pause, single-step forward/backward and go to the first/last frame are available. In the pause mode the screen is continually refreshed, allowing the image to be modified with the display controls.

The display control section provides controls for manipulating the image and selecting the desired viewports. Controls include plot centering, zoom-in, zoom-out, translate-left, translate-right, translate-up, translate-down, toggle on/off a grid for wake structure mea-

surements, toggle on/off the  $C_p$  plot and scroll graphs and toggle on/off boundary layer calculations. Also included in the display control section (but not shown in the figure) is a *tool* button that brings up a submenu with available flow visualization and measurement tools. These will be discussed in later sections.



Fig. 7. Control panel; program control section.

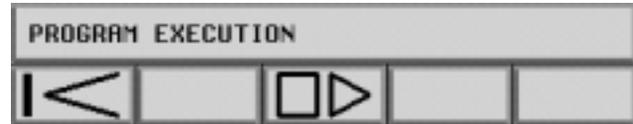


Fig. 8a. Control panel; execution section (UPOTG).



Fig. 8b. Control panel; playback section (REPLAY).

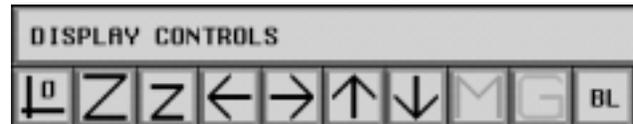


Fig. 9. Control panel; display control section.

**Graphical Editor** An input file is used to operate the program, with parameters defining the airfoil topology, motion controls and flowfield characteristics. To ease program operation the input file may be edited using an interactive, mouse/keyboard-operated graphical editor. The graphical editor is invoked by clicking on the edit button in the program control panel. A fullscreen display is created with sections for airfoil definition, time-stepping parameters, accuracy limits, boundary layer information, defined-motion parameters and aeroelastic-motion parameters. Each section consists of logical buttons and data-entry lines. A sample section is shown in Fig. 10. Parameter modifications can be written to the input file, or to a new file, and new input files may be read in.



Fig. 10. Graphical editor; time-stepping section.

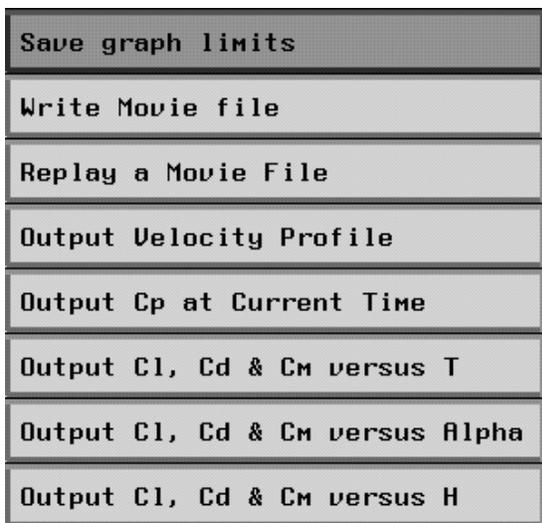


Fig. 11. I/O panel.

**I/O Panel** Data I/O is enabled by clicking on the I/O button, bringing up the I/O submenu (Fig. 11). The present graph limits may be saved (primarily for use in the *replay* utility), movie files may be written or replayed, and airfoil and flow data may be written to specified files in various formats.

**Tool Panel** A tool submenu (Fig. 12) is brought up by clicking on the tool button, and the **Wake menu**, the **Tracer menu** and the **Profiler menu** are invoked by clicking on the respective button. These applications appear as windows that can be moved around within the workspace and closed.



Fig. 12. Tool panel.

**Wake Representation** The **Wake menu** (Fig. 13) provides control over the display of the wake. Control features include the wake symbol type and size. Recall that the wake is represented numerically as a finite chain of discrete vortices. Three symbol options are available for expressing the vortex location and magnitude. In all cases the wake color is set by the local rotational direction of the discrete vortices; magenta for clockwise rotation, and cyan for counter-clockwise rotation.

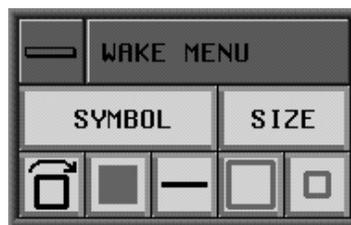


Fig. 13. Wake menu.

The first symbol option delineates each vortex as a rotating hollow square with the size and rotational speed proportional to the vorticity. This option (shown in Fig. 14a), while quite expressive during animation, is generally not well suited for viewing still frames. The second option delineates each vortex with a filled circle with diameter scaled by the vorticity. This option (Fig. 14b) is often best for viewing still images. The third option connects the vortices in the order that they are created with a line with thickness scaled by the local vorticity. This option (shown

previously in Fig. 4 and here in Fig. 14c) has the appearance of smoke or dye injection in an experimental facility, and generally provides excellent moving and still images, the exception being for high frequency high amplitude motions. For more energetic motions the connectivity of the discrete vortices may become confused as the individual vortices roll up into large eddies which, in turn, may interact with or be split by other large eddies.



Fig. 14a. Rotating hollow square wake symbols.

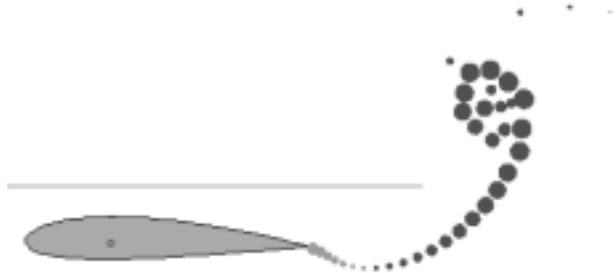


Fig. 14b. Filled circle wake symbols.

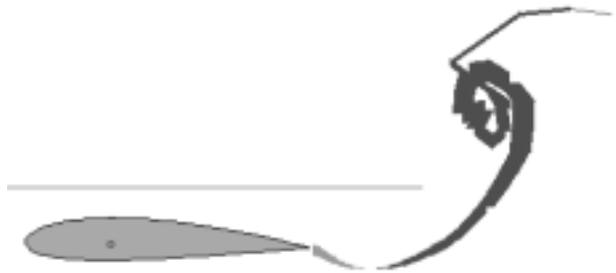


Fig. 14c. Connected line wake representation.

The symbol size is always scaled by the local vorticity magnitude, but the magnitude of the vorticity may vary by several orders of magnitude from one simulation to the next; therefore, the size scale-factor may be interactively adjusted by the user.

**Particle Tracer** Particle trace functions are available through the **Tracer menu**. When the Tracer is invoked the cursor is shown as a circle with cross-hairs when the mouse is within the main viewport. All particles are deleted when the menu is closed.



Fig. 15. Tracer menu.

Several symbol or trace options are available. The first option releases a single particle from the current mouse location each time the left mouse button is pressed, and the particle is portrayed as a vector with its length scaled by the local velocity. The second option is similar to the first, but the particles are displayed as dots. The last option releases a continuous stream of particles from the current position of the mouse, as shown in Fig. 16, mimicking smoke or dye-injection in an experimental facility.

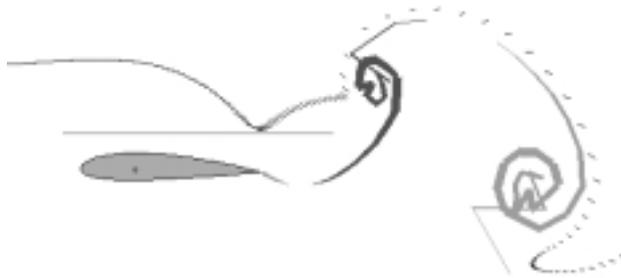


Fig. 16. Example of the particle tracer.

**Velocity Profiler** In the panel code, the lift, drag and moment of the airfoil are computed directly from the surface pressure distribution. Experimentally this is often not possible. To compute the net drag (or thrust) of moving airfoils in experimental facilities the velocity profile downstream of the airfoil may be measured using a hot-wire or Laser Doppler Velocimetry (LDV), where the resulting velocity deficit (or surplus) is due to the drag (or thrust) of the airfoil. This cross-sectional velocity measurement capability is modeled in the panel code by the velocity profiler. Velocity profile computations are controlled through the **Profiler menu** shown in Fig. 17.

Using the menu, the  $x$ -location of the velocity profile, the  $y$ -coordinate range of the profile, the number of data points in the profile and the time-window for averaging can be set. A hatched grid is drawn behind the wake image, and the velocity profile is plotted as shown in Fig. 18.

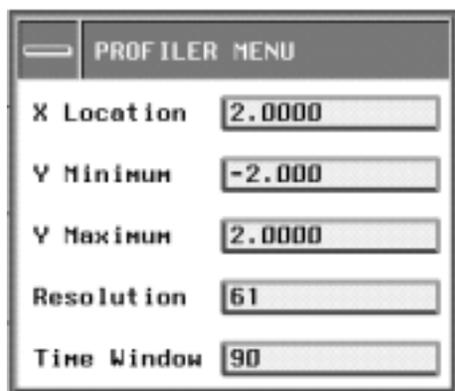


Fig. 17. Profiler menu.

Averaging with the specified time-window cannot be correctly performed until a sufficient number of time-steps have been computed. In this case averaging is performed with a smaller time-window, and the numeric value of the time-window displayed in the **Profiler menu** blinks to indicate this. The time-averaged velocity profile may be written to an external file for external processing and/or comparisons with experimental data.

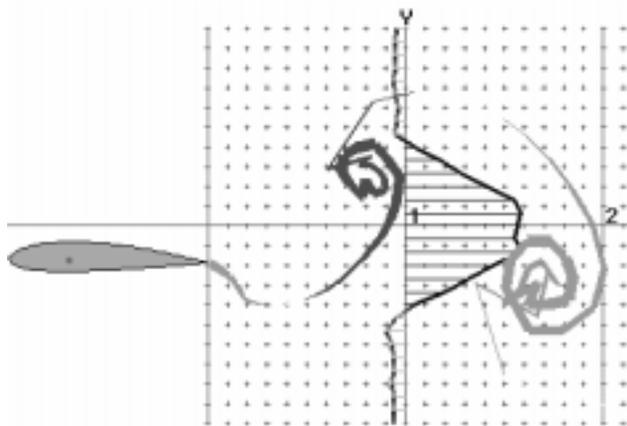


Fig. 18. Time-averaged velocity profile.

### Results

Unfortunately, the results from a graphics tool such as this cannot easily be portrayed as still images in a printed paper. While the images presented here are hardly representative of the true capabilities of the complete animation package, they may illustrate the ability of the available tools to present data in an understandable, expressive format.

A primary motivation for the development of the graphical animation interface was to enable direct comparison of the computed wake structures with experimental data. Details of an ongoing experimental study

as well as a more comprehensive presentation of comparative results are given in Ref. 5 with a few examples included here.

Comparisons of wake structures computed by the flow solver and photographed in the NPS water tunnel are shown in Figs. 19a and 19b, for a NACA 0010 airfoil plunging sinusoidally with  $k = 3.0$ ,  $A = 0.20$  and  $k = 10.1$ ,  $A = 0.20$ , respectively.

Two color dye-injection is used to visualize the wake in the water tunnel with dye-injected from two thin tubes placed slightly upstream of the airfoil. With the tubes properly placed, the dye is entrained in the accelerated flow near the foil surface and appears to emanate from the trailing edge as it does in the virtual wind tunnel.

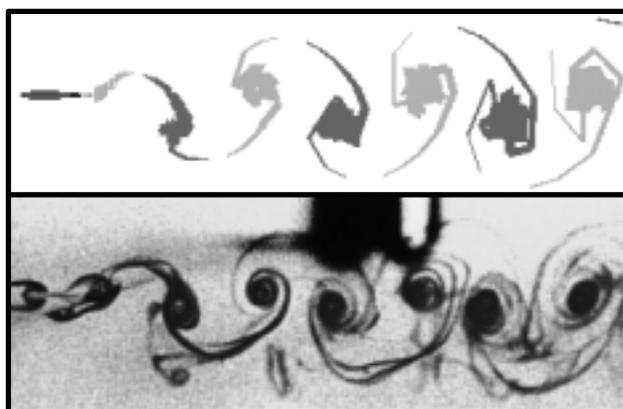


Fig. 19a. Wake comparison;  $k = 3.0$ ,  $A = 0.20$ .

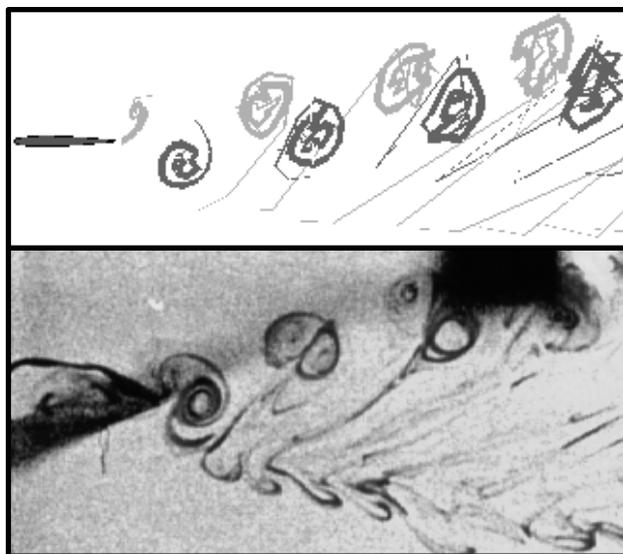


Fig. 19b. Wake comparison;  $k = 10.1$ ,  $A = 0.20$ .

The accuracy of the solutions is discussed in depth in Ref. 5 and is of little concern here. The ability of the graphics package to present numerical data in a suitable format, however, is of great interest, and it can be seen that a reasonable facsimile of the experimentally obtained photographs is produced in both cases. The case shown in Fig. 19b is of particular interest, as it is a highly nonlinear case where a symmetric motion generates a nonsymmetric wake. This loss of symmetry is discussed in detail in Ref. 5 and was previously recorded by other experimentalists.<sup>10</sup>

In the experimentally obtained photographs the structure of the wake is unclear. It appears as if a primary vortex street composed of jet-like vortex pairs is deflected upward as it convects downstream, while a secondary vortex street comprised of much smaller elements is deflected downward. It was originally theorized that the apparent dual-street structure of the wake might be a result of flow viscosity, however, the still image extracted from the numerical animation sequence demonstrates a very similar structure, indicating that it is an essentially inviscid process. Viewing the simulation in motion using the replay utility it is easily observed that the so-called secondary vortex street is merely remnants of vorticity stretched and detached from the large eddies.

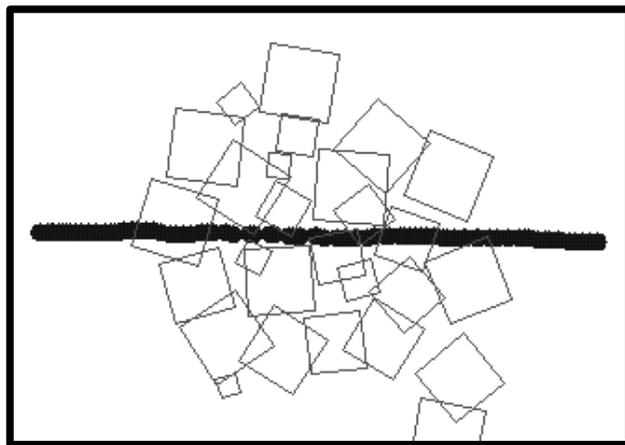
The visualizations shown in Figs. 19a and 19b are often referred to as vorticity tagging, i.e., releasing particles that highlight vorticity generation and convection. The panel code provides the additional capability of indicating the magnitude and orientation of the vorticity.

As shown in Fig. 16 particles can also be released from arbitrary positions in the flow simulating a smoke wand or dye injection in an experimental facility. Particles can also be released individually in a way not easily duplicated by experimentalists. The flow may be frozen at a given time, and any number of particles may be released at arbitrary points in the flowfield. The following example illustrates one use of this feature.

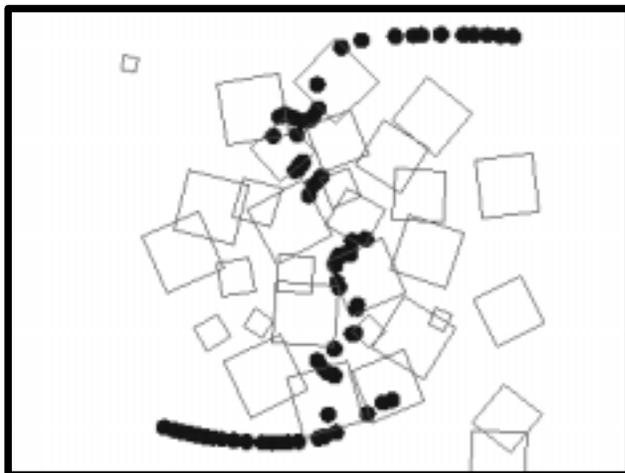
Recall that the numerical wake is comprised of a finite number of discrete ideal vortices. When the airfoil is moved energetically the stream of discrete vortices may roll up creating a street of larger eddies, as seen in Figs. 19a and 19b. While the individual ideal vortices have an ideal velocity distribution,  $u_\theta = 1/r$ , the large eddies should have a more realistic velocity distribution; i.e., the velocity in the core should resemble solid-body rotation, while the velocity at the perimeter should resemble an ideal vortex.

In Fig. 20a the flow is frozen and particles are released along a straight line through the center of an

eddy. The eddy is represented by the roughly circular conglomeration of discrete vortices shown as hollow squares. The particles are shown as black dots forming a line. The simulation is then restarted and frozen again at a later time, with the new particle positions shown in Fig. 20b. It can be seen that the particles in the core still form a straight line (with some diffusion) indicative of solid-body rotation, and the particles outside the eddy now form spiral arms indicative of an ideal vortex.

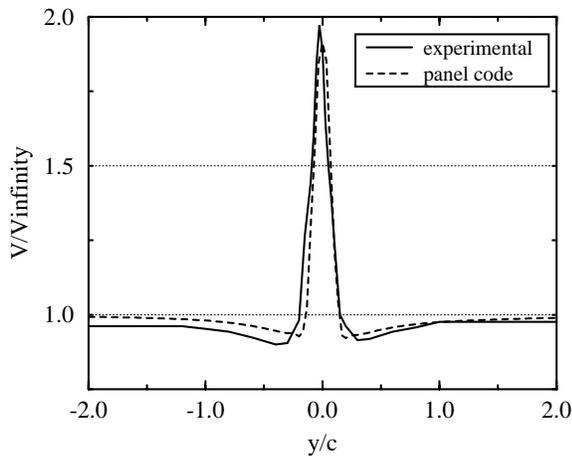


**Fig. 20a.** Particles released in a line through an eddy.



**Fig. 20b.** Particle positions after several time steps.

As previously mentioned, the net drag (or thrust) of an airfoil is often predicted experimentally by measuring the velocity profile downstream of an airfoil. A velocity profile measured by LDV in Ref. 5 is compared to the profile generated by the Profiler in the panel code in Fig. 21 for a NACA 0012 airfoil plunging sinusoidally with  $k = 15$  and  $A = 0.04$ .



**Fig. 20.** Downstream velocity profile comparison.

The comparison is quite good with the large velocity spike indicating substantial thrust production.

### Conclusions

An interactive graphics animation interface was developed as a front end for an unsteady potential flow code. The aeroelastic panel code simulates flows about airfoils undergoing forced or aerodynamically driven motions with one or two-degrees-of-freedom. The interactive, mouse-driven interface generates a graphical workspace for displaying the airfoil and wake dynamics as well as a time-accurate display of surface pressure and time histories of position variables and integrated forces and moments.

The graphics package may be used concurrently with the unsteady panel code, displaying the results frame-by-frame as they are computed, or a compact binary file containing the necessary data to regenerate the animation sequence may be created, such that the simulation may be reanimated at a later time and at a more realistic frame-rate. During replay, the sequence may be paused and advanced or rewind one frame at a time.

Several tools are available for probing the flow in ways that mimic experimental data acquisition techniques, giving the package the look and feel of a *virtual wind tunnel*. Several methods of flow visualization, force balance measurements, surface pressure measurements and flow anemometry are all available options.

Presented results demonstrate the ability of the code to accurately duplicate experimental flow visualization imagery. Additionally, it was shown that flow

visualization techniques not possible experimentally could easily be applied in the *virtual wind tunnel* to investigate complex flow features. The complete package is presently used both as a research tool and as an instructional tool for the investigation of unsteady dynamics and airfoil stability.

### References

- <sup>1</sup> Teng, N. H., "The Development of a Computer Code for the Numerical Solution of Unsteady, Inviscid and Incompressible Flow over an Airfoil," Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1987.
- <sup>2</sup> Platzer, M. F., Neace, K. S. and Pang, C. K., "Aerodynamic Analysis of Flapping Wing Propulsion," AIAA Paper No. 93-0484, Jan. 1993.
- <sup>3</sup> Turner, M., "A Computational Investigation of Wake-Induced Airfoil Flutter in Incompressible Flow and Active Flutter Control," Master's Thesis, Naval Postgraduate School, Monterey, CA, March, 1994.
- <sup>4</sup> Jones, K. D. and Platzer, M. F., "Time-Domain Aeroelastic Analysis of a two Airfoil System with Application to Unsteady Rotary Wing Flowfields," AIAA Paper No. 95-0337, Jan., 1995.
- <sup>5</sup> Jones, K. D., Dohring, C. M. and Platzer, M. F., "Wake Structures Behind Plunging Airfoils: A Comparison of Numerical and Experimental Results," AIAA Paper No. 96-0078, 1996.
- <sup>6</sup> Hess, J. L. and Smith, A. M. O., "Calculation of Potential Flow about Arbitrary Bodies," **Progress in Aeronautical Sciences**, Vol. 8, pp. 1-138, Pergamon Press, Oxford, 1966.
- <sup>7</sup> Basu, B. C. and Hancock, G. J., "The Unsteady Motion of a Two-Dimensional Aerofoil in Incompressible Inviscid Flow," **Journal of Fluid Mechanics**, Vol. 87, 1978, pp. 159-168.
- <sup>8</sup> Riester, P. J., "A Computational and Experimental Investigation of Incompressible Oscillatory Airfoil Flow and Flutter Problems," Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1993.
- <sup>9</sup> Tuncer, I. H., Platzer, M. F. and Ekaterinaris, J. A., "Computational Analysis of Flapping Airfoil Aerodynamics," ASME Fluids Engineering Division, Summer Meeting, June, 1994.
- <sup>10</sup> Bratt, J. B., "Flow Patterns in the Wake of an Oscillating Airfoil," Aeronautical Research Council, R&M 2773, 1953.