Philippine Engineering Journal (1993) XIV (1): 49-79

SOFTWARE DEVELOPMENT FOR FLUID-FLOW VISUALIZATION USING COMPUTER GRAPHICS AND ANIMATION TECHNIQUES

Leonardo Q. Liongson, Ph. D.
NWRC Professor of Water Resources Engineering
Department of Engineering Sciences
and
Research Fellow
National Hydraulic Research Center
University of the Philippines
Diliman, Quezon City

ABSTRACT

This paper presents the principal results of a two-year software development project for the visualization of fluid motion using computer graphics and animation techniques.

Twenty-five (25) visualization programs have been developed for potential flows, viscous flows, flows with heat and mass transfer, geostrophic flows, and open-channel flows. Single-page and double-page animation, and palette rotation (neon-light effect), featuring on the monitor screen the display of both text and graphics with color, animation, and sound, are the essential techniques utilized. Classical tools in engineering mathematics were numerically applied and implemented, such as complex variable theory, conformal mapping, both analytical and numerical solutions of ordinary and partial differential equations, and stochastic or random-walk simulation of the diffusion process.

The applied techniques (computer graphics, animation, analytical and numerical methods in engineering mathematics) exhibited varying degrees of performance in terms of a meaningful perception of fluid motion, depending on appropriateness of the application to certain types of flows and on the nature and amount of detail present in the images. With the added features of a simple and guided input menu, a facility for instant replay, and a real-time mode of computation and display, the programs serve as interactive teaching tools in fluid mechanics.

(A Professorial Chair Lecture delivered on 21 June 1993 at the National Hydraulic Research Center, MH 233-A, College of Engineering, University of the Philippines, Diliman, Quezon City)

INTRODUCTION

Fluid flow concepts and variables, until the advent of computer-aided visualization, have been traditionally defined and explained mostly with the aid of **static pictures**, and whenever possible, physically visualized by means of instructional **laboratory apparata** such as wind tunnels, flumes, pipe assemblies, and laminar flow apparata. While the traditional tools are still necessary and valid important components in fluid mechanics instruction, the lack of hydraulic and aerodynamic laboratory equipment in many schools has limited the local application of physical visualization for teaching fluid flow concepts. Aside from NHRC at U.P., very few other engineering schools have these complete physical facilities.

The availability of built-in **graphics and animation features** in microcomputer source languages offers a practical and convenient alternative as well as a timely opportunity to visualize fluid flow phenomena. Two-dimensional and three-dimensional computer graphics, with geometric figures displayed with color, animation, and sound on the monitor screen, provide an interactive medium to examine different flow behavior.

It is possible to visualize through the computer certain **abstract flow concepts otherwise not directly observable** in the laboratory. Perhaps the advantage of computer-aided visualization is in being able to visualize what is otherwise physically invisible, or too fast, or too slow, or too abstract. On the practical side, whereas physical flow visualization requires pumps, tanks, and pipes to supply the "wet" fluid, computer-aided visualization on the other hand is fluid flow visualization in the "dry", that is, with dry pixels on the phospor screen. Another term used in the 1970's is "numerical wind tunnel," (reminiscent of big mainframes and also big wind tunnels)

The visualization software can serve as an effective and practical tool for the **understanding and appreciation of important features of fluid motion.** The basic differential equations governing fluid motion (with unknowns such as velocity and pressure) are well known and understood, but there is no general analytical solution, only solutions for a number of particular cases, available in either analytical, semi-analytical, or numerical form. Analytical solutions, even in closed-form, are neither always simple nor fully understood, whereas numerical solutions, as tabulated numbers, are often not very descriptive without pictures. Visualization helps by extracting from the solutions the set of display variables such as velocity vector fields, pressure (also temperature and concentration) scalar fields, streamlines, pathlines, vortexlines, and vortex sheets. Using computer graphics and animation, these solution variables become visible and easier to describe, appreciate, and understand.

The software is principally both a simulation and visualization tool in computer-aided instruction (CAI). Beyond the classroom, with additional programming, they can be extended and linked as a microcomputer-based component of computer-aided-design (CAD) and computer-aided-engineering (CAE).

Compiled programs running in PC-80386 machines under MS.DOS can perform at speeds adequate for visualization of fluid flow in an instructional environment. In the absence of a physical visualization facility, a computer visualization set-up can be a suitable if not a better substitute. This is the rationale for software development for fluid flow visualization.

It can be mentioned that general-purpose **UNIX-based visualization platforms** are already available in the world market but these are less affordable, running on more expensive machines more powerful than the PC-80386, such as reduced-instruction-set machines. Their market includes scientific and medical research centers, the entertainment and advertising industries, and the military. Meanwhile, affordable PC-based instructional visualization software in **physics, mathematics and other natural sciences**, developed by the academe, has already appeared abroad.

OBJECTIVES

This paper is based on the final report for the two-year DOST-PCASTRD-supported Project **"Software Development for Visualization of Fluid Motion Using Computer Graphics and Animation Techniques."** The organizations which implemented the project, and to which the author belongs, are the **National Hydraulic Research Center** and the **Department of Engineering Sciences** of the University of the Philippines at Diliman, Quezon City.

The main objectives of the project were:

1. To develop computer-aided flow visualization programs for **two-dimensional (2-D) flows**, using analytical and numerical flow solutions, and applying two-dimensional graphics and animation techniques.

2. To develop computer-aided flow visualization programs for **three-dimensional (3-D) flows**, using analytical and numerical flow solutions, and applying three-dimensional graphics and animation techniques.

The types of fluid flows which have been investigated for visualization are:

1. Potential (irrotational or ideal) flows

2. Viscous flows (including boundary-layer flows, laminar and turbulent flows)

3. Flows with heat and mass transfer.

4. Geostrophic flows (flows affected by Corioli's forces)

5. Open-channel flows.

HARDWARE AND SOFTWARE REQUIREMENTS

The **minimum** computing requirements of the project are as follows:

1. Hardware

The minimum recommendation is a PC-80386/25-MHz compatible microcomputer, with math-coprocessor, 640 K RAM + 1 Mb extended memory, 8 Mb Hard Disk, and VGA monitor screen. A PC-80286 is accceptable but will be slower. A PC-80486, as expected, will be much faster, in fact too fast for certain types of animation where long eye contact (say 1 second at least) is needed to fully appreciate certain flow objects on display.

2. Software

A compiler language with adequate graphics features was required during development. The availability of and the familiarity with the superior graphics, string-processing, and run-time compilation capabilities of the Quick BASIC, or QBASIC, compiler language resulted in its adoption..

For direct users or nonprogrammers, pre-compiled MS-DOS executable files of the visualization programs are available in diskettes.

COMPUTER GRAPHICS AND ANIMATION TECHNIQUES

The computer graphics and animation techniques which are utilized are as follows:

1. Drawing a **line segment** with arbitrary end points and color. A series of such lines can form a polyline or a polygon. A line segment may be in **true length** as in 2-D flow visualization, or in **isometric and orthographic projections** as in 3-D flow visualization. Lines, polylines, and polygons are utilized to display solid boundaries, 2-D and 3-D flowlines (streamlines and pathlines), 2-D fluid interfaces, and 3-D vortexlines. **Color coding** can be assigned to these lines to suggest values of scalar properties such as pressure or to distinguish among them.

2. Turning on a **pixel** with arbitrary position and color. This is used to indicate current flow tracer positions. **Color coding** can also be used to distiguish among different tracers.

3. Drawing a **circle** of arbitrary center, radius and color. This is utilized to represent circular solid boundaries, and 2-D flow entities such as source, sink, vortex centers. A similar technique is drawing **small boxes** of arbitrary center and size, these being suitable as **color-coded tracers** with **temperature** or **concentration** values.

4. Playing a **sound** of specified pitch and duration. This can be used to suggest speed of flow by pitch and rhythm of the sound.

5. **Single-page animation** effected by the repeated and cumulative application of techniques 1, 2, 3, and 4. This is adequate for visualizing steady-state fluid flow (fixed streamlines), but can also be used for unsteady flow (pathlines) at the sacrifice of time-coherence of flowlines. This feature gives a cumulative appreciation (hence, the perception of animation) of the temporal and spatial variation of the flow, showing both the previous and current flow tracer

positions on a single visible screen page.

6. **Double-page animation** effected by alternately swapping two visible screen pages each with time-variable displays recurrently produced by techniques 1, 2, and 3. This is necessary if a time-coherent display of unsteady flow is desired. This is the fully animated feature wherein every flow element on the current visible page is occurring at the same time.

7. **Palette rotation (neon-light effect)** effected by first constructing a screen image display by single-page animation using one color palette, and then imparting a neon-light effect by assigning a temporally-alternating color palette to the original color codes. For example, a streamline consisting of a polyline colored blue-yellow-blue-etc. (in that order) is "palette rotated" to one with the same polyline colored yellow-blue-yellow-etc. This is applied recurrently to the same image: blue-yellow-blue-etc. becoming yellow-blue-yellow next time, becoming again blue-yellow-blue again, and so on. This technique does not require redrawing the lines and pixels in the image, only the rotation of the two color palettes. The technique can of course be applied to palettes of more than two colors or shades. This, obviously, achieves the neon-light effect, a technique identified and applied during the second year.

The visualization routines are performed in **real-time computation mode** in order to preserve the interactive features of the software:

1. **Fixed flow parameters** are interactively inputted by the user.

2. The **propagation of flowlines** in the form of moving tracer positions is achieved by time-integration of the computed velocity field, using the **Euler (first-order) or modified Euler (second-order) Method**. For visualization purposes, higher order Runge-Kutta method of time-integration is not essential.

3. After real-time computation, **repeatable fast instant replay** of the time-variable displays of computed and RAM-resident flow tracer positions can be done.

Large scale post-computation (non real time) visualization, meaning sequential display of several precomputed images stored in the hard disk, like a movie, removes the interactive features and thus is **not** implemented. At the other extreme, real-time visualization with the maximum interaction and speed of computer games is entirely possible. However, this aspect of software development is dependent on the availability of machines running the compiled QBASIC programs at more than 25 MHz speeds. This has been achieved in fact with the highly interactive sluice-gate and hydraulic-jump simulator program when run with a PC-80486 compatible machine with speed of much more than 25 MHz (the nameplate claims 118 MHz).

Numerical time-integration of the flow velocity components [u(x,y),v(x,y)] to obtain flow tracer positions (x,y), from time (t) to time ($t+\delta t$), illustrated here for two-dimensional flow, is achieved by either of two methods:

Euler Method (First Order):

$$x(t+\delta t) = x(t) + u(x(t),y(t))\ \delta t$$
$$y(t+\delta t) = y(t) + v(x(t),y(t))\ \delta t$$

Modified Euler Method (Second Order):

$$x(t+\delta t) = (x_o + 2\ x_1 + x_2)/4$$
$$y(t+\delta t) = (y_o + 2\ y_1 + y_2)/4$$

where

$$x_o = x(t)$$
$$y_o = y(t)$$
$$x_1 = x(t) + \tfrac{1}{2}\ u(x_o,y_o)\ \delta t$$
$$y_1 = y(t) + \tfrac{1}{2}\ v(x_o,y_o)\ \delta t$$
$$x_2 = x(t) + u(x_1,y_1)\ \delta t$$
$$y_2 = y(t) + v(x_1,y_1)\ \delta t$$

In certain flow cases, **analytical formulas** for the tracer positions $(x(t),y(t))$ are available. These are directly utilized by the programs, precluding the need to perform numerical integration of velocity functions.

FLOW VISUALIZATION CASES I: POTENTIAL FLOW

The subject area in fluid mechanics initially developed for visualization is the mathematically simpler **Potential Flow Theory**, as applied to different cases whose theories are discussed in the following subsections. In general, one requires functional relationships for the velocity field which can be time-integrated (analytically or numerically) to obtain the variable tracer locations.

Given a **velocity potential,** $\phi(x,y)$, in two-dimensional flow, the velocity components, u and v, are obtained theoretically from

$$u(x,y) = \delta\phi/\delta x$$

$$v(x,y) = \delta\phi/\delta y$$

Continuity or mass conservation for incompressible fluid requires:

$$\delta u/\delta x + \delta v/\delta y = 0$$

Hence, the velocity potential must satisfy **Laplace Equation:**

$$\delta^2\phi/\delta x^2 + \delta^2\phi/\delta y^2 = 0$$

From Bernoulli Equation on a plane,

$$P_o + \frac{1}{2}\, p\, U^2 = P + \frac{1}{2}\, p\, (u^2 + v^2)$$

the **pressure distribution** can be expressed in terms of the dimensionless pressure coefficient, C_p:

$$C_p = [P - P_o]/[\frac{1}{2}\, p\, U^2] = 1 - (u^2 + v^2)/U^2$$

where

P = pressure at any tracer position
(u,v) = velocity components in the same tracer position
P_o = free-stream pressure
U = free-stream velocity
p = fluid density

As a scalar field, pressure is visualized by color coding of tracer pathlines and positions.

Karman Vortex Wake behind a Circular Cylinder (Program KARMAN)

The particular objective in this case is to produce computer displays of the unsteady two-dimensional flow pattern due to a free stream past a circular cylinder with a **Karman Vortex Wake** - a developed drifting wake composed of two rows of either aligned or staggered vortices. These vortices separate from the cylinder with predicted strength and frequency, and mutually induce computed motion among themselves and on the surrounding fluid.

The theory utilized is essentially a kinematic approach based on the combined uniform free-stream flow and the induced potential vortex flows. The flow-tangency boundary condition on the cylindrical surface is maintained by the presence of conjugate vortices inside the cylinder, as defined by the **Circle Theorem** (Milne-Thomson, Theoretical Hydrodynamics). The computer displays correspond to time-exposure pictures of pathlines traced by selected tracer points.

Karman Vortex Wake behind a Flat Plate (Program KARPLAT)

The objective is to display the unsteady Karman Vortex Wake behind **a flat plate** inclined by an angle of attack relative to the free-stream velocity.

The mathematical technique used is first to derive the flow tracer positions **for a wake** behind a circular cylinder, as in program KARMAN, and then to apply **conformal mapping** from a flow domain behind a circular cylinder to that behind an inclined flat plate.

Potential Flow past a Circular Cylinder with Free Stream and Circulation (Program FLOWCYL)

The objective here is to display the steady two-dimensional flow pattern due to a free stream past a circular cylinder with or without circulation.

Potential Wake behind a Moving Cylinder with Circulation (Program FLODBLET)

The objective here is to display the unsteady-state potential wake created by a moving circular cylinder with circulation, i.e., the **relative motion** created if the cylinder is moving against static fluid, instead of the fluid moving against a static cylinder as in program FLOWCYL. The wake motion is created by the potential for a **moving doublet plus circulation**.

Potential Flow past a Flat Plate with Free Stream and Circulation (Program FLOWPLAT)

The objective here is to display the steady two-dimensional flow pattern due to a free stream past a flat plate with circulation, inclined by an angle of attack relative to the free-stream velocity.

The mathematical technique used is to first derive the flow tracers for potential flow past a circular cylinder, as in program FLOWCYL, and then to apply **conformal mapping** from a flow domain past a circular cylinder to that past an inclined flat plate, as also applied in program KARPLAT.

Potential Flow past an Thin Airfoil with Free Stream and Circulation (Program AIRFOIL)

The objective is to display the steady two-dimensional flow pattern due to a free stream past a thin airfoil with specified chord length and angle of attack. The velocity potential is given by a distribution of vortex strength along the thin airfoil.

Potential Flow past a Source-Sink Pair with Free Stream (Program FLOWSS)

The objective here is to display the steady two-dimensional flow pattern due to a free stream past a source-sink pair.

Potential Flow past a Vortex Pair with Free Stream (Program FLOWVORT)

The objective is to display the steady two-dimensional flow pattern due to a free stream past a vortex pair.

Potential Flow from a Channel Outfall
(Program FLOWCHAN)

The objective here is to display the potential flow which develops when an initially uniform flow confined in a channel expands and decelerates upon exit at the channel outfall. The mathematical technique used is to apply **conformal mapping** from uniform flow to outfall flow.

Conformal mapping leaves the potential function invariant but not the velocity. In order to depict the true velocity magnitudes in the transformed outfall domain, the uniform-flow tracer velocity is appropriately predistorted for a given straight streamline prior to transformation.

Small-amplitude Surface Waves (Eulerian Approximation)
(Program WAVEDYN)

The objective here is to display the unsteady two-dimensional flow pattern present in a small-amplitude gravity surface wave. The tracer positions are directly computed from an Eulerian approximation wherein variable position is a function of time and fixed arbitrary orbital centers. Tracer orbits are approximately **circular** for large values of depth-to-wavelength ratio (deep-water waves), while they are **elliptical** for smaller ratios (shallow-water waves).

Small-amplitude Surface Waves (Lagrangian Solution)
(Program WAVETRAN)

The objective is also to display the unsteady two-dimensional flow pattern present in a small-amplitude gravity surface wave. In contrast, the tracer positions are computed by means of an exact Lagrangian method of time-integration of the velocity components. The velocity components are functions of the varying tracer positions acting as instantaneous orbital centers. This flow model is capable of simulating and visualizing **incipient breaking of wave crests**.

Small-amplitude Surface Wave Groups
(Program WAVEGRP)

The objective here is to display the unsteady two-dimensional flow pattern present in a small-amplitude gravity surface wave group, which is produced by the **interference of two waves** with equal heights and almost equal wavelengths, for a given channel depth. The tracer positions are directly computed as in program WAVEDYN whereby variable position is obtained by linear superposition of two similar wave functions differing only in their wavelengths (Eulerian approximation).

3-D Vortex Wake by Vortextube Method (Program 3DVORTEX)

The particular objective in this case is to produce computer displays of the three-dimensional flow pattern or vortex wake, traced by generated bound, shed, and trailing vortextubes, and caused by the combined effects of ambient flow and the resulting mutual velocity induction due to these vortextubes. The shed and trailing vortextubes derive from the spatial and temporal variations of the bound circulations which develop on variable-lift thin airfoils or rotor blades. The generation of the shed and trailing vortextubes satisfies the principle of vortex continuity, whereas the velocity induction obeys the Biot-Savart Law.

The theory utilized is comprised of the principles of continuity and mutual velocity induction governing the kinematics of vorticity or its vortextube-integral called circulation. Three types of vortextube circulations are considered. These are as follows:

1. Bound Circulation associated with Lift developed on Airfoils or Rotor Blades
2. Shed Circulation generated by temporally-varying Bound Circulation
3. Trailing Circulation generated by spatially-varying Bound Circulation

The mutual velocity induction present among the vortextubes is governed by the Biot-Savart Law. The mutual advection of the vortextubes (of their end-nodes, in particular) is caused by the combined effects of an ambient flow and the summation of induced motions due to the other vortextubes.

A variety of three-dimensional flow examples may be visualized with the development of this software. Among these are:

1. Vortex-wake formation downstream of finite fixed-wing airfoils with given chord length and pitch schedule.
2. Helicoidal-vortex wake formation downstream of horizontal-axis rotors (HAWT or propeller type of rotors).
3. Cycloidal-vortex-wake formation downstream of vertical-axis rotors (VAWT or Darrieus type of rotors).
4. Circular-vortex wake formation behind shrouded or ducted turbine, or diffuser augmented wind turbines (DAWT).
5. Horseshoe-vortex formation due to interaction of vortexlines with stagnation points on cylindrical piers.

FLOW VISUALIZATION CASES II: VISCOUS FLOW

Viscous flow is in general governed by the Navier-Stokes Equations (NSE), together with the continuity equation for incompressible, Newtonian fluid. The NSE are Newton's equations of motions for three flow directions (forces are expressed per unit fluid mass):

acceleration = pressure force + gravity force
+ other body forces + viscous force

Navier-Stokes Equations:

$$\delta u/\delta t + [u\ \delta/\delta x + v\ \delta/\delta y + w\ \delta/\delta z]\ u =$$

$$- 1/p\ \delta(P + pgH)/\delta x + f_x + \tau\ [\delta^2/\delta x^2 + \delta^2/\delta y^2 + \delta^2/\delta z^2]\ u$$

$$\delta v/\delta t + [u\ \delta/\delta x + v\ \delta/\delta y + w\ \delta/\delta z]\ v =$$

$$- 1/p\ \delta(P + pgH)/\delta y + f_y + \tau\ [\delta^2/\delta x^2 + \delta^2/\delta y^2 + \delta^2/\delta z^2]\ v$$

$$\delta w/\delta t + [u\ \delta/\delta x + v\ \delta/\delta y + w\ \delta/\delta z]\ w =$$

$$- 1/p\ \delta(P + pgH)/\delta z + f_z + \tau\ [\delta^2/\delta x^2 + \delta^2/\delta y^2 + \delta^2/\delta z^2]\ w$$

Continuity equation:

$$\delta u/\delta x + \delta v/\delta y + \delta w/\delta z = 0$$

where

(x,y,z) = position coordinates
t = time
(u,v,w) = unknown velocity components
= function of (x,y,z,t)
P = unknown pressure
= function of (x,y,z,t)
p = fluid density
g = gravity acceleration
H = upward elevation opposite to gravity
τ = fluid kinematic viscosity
(f_x, f_y, f_z) = other body forces/unit mass

One boundary condition commonly prescribed on a stationary solid boundary, S, is given by the so-called **no-slip condition**:

BC1: $(u,v,w) = (0,0,0)$ for all (x,y,z) in S.

BC1 could give rise to boundary layers which are regions of flow adjacent to S, characterized by decelerated motion relative to the rest of faster moving fluid outside the layer. Boundary layers develop on the inner walls of closed-conduits (pipes) and open channels, and on the outer walls of immersed bodies.

For **boundary layer flows**, the NSE may be simplified for the following steady, two-dimensional case:

$$u\ \delta u/\delta x + v\ \delta u/\delta y = -\ 1/p\ \delta P/\delta x + \tau\ \delta^2 u/\delta y^2$$

$$0 = -\ 1/p\ \delta P/\delta y$$

$$\delta u/\delta x + \delta v/\delta y = 0$$

where the u- and x-directions are in the main flow direction along the boundary layer while the v- and y-directions are across it, such that $u >> v$ and $\delta u/\delta x << \delta u/\delta y$. Pressure, P, as shown above, does not vary across the boundary layer (does not vary with y).

Viscous flow is parameterized by the dimensionless **Reynolds Number**, Re, which expresses the ratio between inertial and viscous forces (friction):

$$Re = p\ L^2\ V^2/(p\tau\ V\ L) = L\ V/\tau$$

where

L = length scale for a particular system
V = velocity scale for a particular system

Laminar flows are characterized by low Reynolds Number while **turbulent flows** by high Reynolds Number. In turbulent flows, the velocity components are formulated as sums of mean components $(u,v,w)_{mean}$, and random fluctuations due to turbulent eddies, (u',v',w'):

$$(u,v,w) = (u,v,w)_{mean} + (u',v',w')$$

After substituting the above turbulent-flow formulation in the NSE and doing time-averaging of every term, the equations for the mean components of motion are derived, with additional so-called turbulent **Reynolds stress terms**, arising from the covariance or mean products of the random fluctuations such as [u'u'], [u'v'], [v'v'], etc. The mean motion is said to be affected by both viscous and turbulent stresses. Semi-analytical solutions of the turbulent-flow equations require a priori assumptions on the properties of the Reynolds stress terms, which are physically additional inertial forces due to momentum transport by turbulent eddies.

Another type of boundary condition may be applied on a **free-surface boundary** or **interface, S,** where pressure, P, is constant (BC2) and where movement of the interface elevation, H = H(x,y,t), should be consistent with fluid velocity (u,v,w), (BC3):

BC2: $P(x,y,z) = \text{constant}$
BC3: $w = \delta H/\delta t + u\ \delta H/\delta x + v\ \delta H/\delta y$ for (x,y,z) in S

BC2 and BC3 could give rise to surface waves typical of open-channel flows.

Open-channel flows are parameterized by the so-called **Froude Number**, Fr, which expresses the ratio between inertial and gravity forces:

$$Fr^2 = p L^2 V^2/(pg L^3) = V^2/gL$$

Low Froude Number flow ($Fr<1$) is identified with tranquil or subcritical flow (relatively deep and slow). High Froude Number flow ($Fr>1$) is called rapid or supercritical flow (relatively shallow and fast). Critical flow corresponds to $Fr=1$.

There is no available general solution of the NSE applicable to all problems. However, there are a number of **analytical solutions for particular cases**, as well as **semi-analytical and numerical solutions**, which can be utilized for visualization purposes. In most of the available solutions, mathematical simplifications have been achieved by temporal averaging, by spatial averaging, or else reduction of three dimensions (3-D) to two (2-D) or one (1-D) dimension, by assumption of steady flow (zero time derivatives), and by performing approximate integral analysis of the equations, such as in boundary-layer flow solutions, which converts partial differential equations to ordinary ones. Solutions of ordinary differential equations are generally more easily determined, whether analytically or numerically.

Boundary Layer Separation along a Straight Wall
(Program BLAYSEP)

The objective here is to display the **laminar boundary layer** flow along a straight wall, with possible **separation due to an adverse pressure gradient** created by an ambient velocity field above the boundary layer. Flow velocity, boundary-layer thickness, and separation point are predicted using the results of approximate integral momentum analysis. The velocity components are derived from self-similar profiles.

Laminar and Turbulent Flow in a Smooth Pipe
(Program TURBO)

The objective here is to display either the **laminar or turbulent flow** pattern inside a smooth circular pipe, depending on the value of the computed Reynolds Number. For laminar flow the velocity profile is parabolic, whereas for turbulent flow, the velocity profile is partly linear inside a viscous sublayer near the pipe wall, and logarithmic with an added random component in the rest of the turbulent zone. For purposes of visualization, the random velocity fluctuations for turbulent flow are formulated using a **stochastic simulation model**, based on an assumed rectangular grid of counter-rotating eddies with randomly varying intensities and a translational velocity equal to the mean velocity.

Laminar Jets from Narrow Slots and Orifices
(Program LAMJET)

The objective is to display the flow of a **laminar jet** issuing from either a narrow slot or an orifice. The jet thickens in the downstream direction due to entrainment of the surrounding fluid caused by viscous shear. Total mass flowrate and jet cross-sectional area slowly increase due to entrainment. Total jet momentum remains constant, however, although the centerline velocity decreases in the downstream direction due to viscous shear with the surrounding fluid.

Decaying Laminar Vortex
(Program LAMVORT)

The objective is to display the unsteady circumferential motion of a **decaying laminar vortex**. The vortex starts as a **free or irrotational vortex** with high center speeds. Due to strong viscous shear developed at the center, a **forced or rotational vortex** evolves there, growing in thickness but diminishing in strength. The entire motion eventually becomes a decaying rigid-body rotation.

The pressure head is also computed and displayed, in order to visualize the influence of varying centripetal accelerations on the radial pressure gradient.

3-D Boundary Layer below a Forced Vortex
(Program 3DBLAYER)

The objective is to display the **three-dimensional boundary-layer** motion induced near a solid floor by an overlying forced vortex.

Inside the boundary layer near the floor, where centrifugal forces are very weak, a certain amount of inward radial motion is induced by the negative center pressure of the forced vortex. This is accompanied by a growing circumferential motion near the floor, giving rise to a distinct **horizontal spiral motion** converging towards the center. After convergence, radial motion diminishes and ascension commences.

Further up, an upward axial velocity becomes dominant and approaches a constant value, so that together with strong circumferential motion of forced-vortex magnitude, a resultant **vertical helical motion** away from the floor develops. The flow pattern resembles a tornado with radial suction and spiral motion near the bottom, and strong helical or twisting motion further up.

FLOW VISUALIZATION CASES III: FLOW WITH HEAT AND MASS TRANSFER

Heat and mass transport phenomena can also be visualized in the same manner that fluid flow or momentum transport has been in all preceding cases. The type of heat transport to be considered is one consisting of the processes of **conduction (thermal diffusion), thermal advection due to fluid motion, and a possible linear decay process** due to other heat transfer mechanisms. In like manner, the mass transport processes of interest are **mass diffusion, mass advection due to fluid motion, and a similar linear decay process** due to physical or chemical reactions. These processes are represented as separate terms in the two-dimensional heat and solute mass conservation equations given below.

Heat Transport:

$$\delta T/\delta t + u\ \delta T/\delta x + v\ \delta T/\delta y = k\ (\delta^2 T/\delta x^2 + \delta^2 T/\delta y^2) - K_d\ T$$

Mass Transport:

$$\delta C/\delta t + u\ \delta C/\delta x + v\ \delta C/\delta y = D\ (\delta^2 T/\delta x^2 + \delta^2 T/\delta y^2) - K_d\ C$$

subject to certain **boundary conditions** on T or C, or their normal derivatives, in some prescribed curve or curves bounding the flow domain;

where

(x,y) = tracer position
t = time
(u,v) = velocity components
T = $T(x,y,t)$ = temperature
C = $C(x,y,t)$ = solute concentration
k = fluid thermal diffusivity
D = mass diffusivity or dispersion coeff.
K_d = decay coefficient

A limited number of **analytical solutions** to these equations are available for particular cases. On the other hand, **numerical solutions** have been obtained for a wider variety of problems, using general numerical methods such as finite-difference, finite-element, and boundary-element methods. However, there is another solution method which is mathematically simpler, computationally efficient, and conceptually more attractive in the sense that the computational steps simulate the actual random diffusion process together with advection and linear decay. This is the **stochastic or random-walk simulation method** described below.

Solution by Stochastic or Random-Walk Simulation:

$$T(x',y',t+dt) = \sum_{(x,y)} T(x,y,t)\ (1-K_d\ dt)/B$$

$$C(x',y',t+dt) = \sum_{(x,y)} C(x,y,t)\ (1-K_d\ dt)/B$$

where

(x,y) = tracer position at time t
(x',y') = tracer position at time t+dt, after advection, diffusion, and decay
dt = time step
B = branching order (2 to 8), to discretize the diffusion process
$(1-K_d\ dt)/B$ = partition factor, with linear decay

Accumulation (Σ) at (x',y') is indicated over all possible (x,y) which satisfies the kinematic relation:

$$x' = x + u\,dt + dL\,r\cos(\Theta)$$
$$y' = y + v\,dt + dL\,r\sin(\Theta)$$

where

$(u\ dt,\ v\ dt)$ = advection terms obtained by the modified Euler method
$dL\ r$ = random-walk step or diffusion term
dL = $\sqrt{(6\ k\ dt)}$ or $\sqrt{(6\ D\ dt)}$ = mixing length
r = uniformly-distributed random number within the interval $(-1,1)$, to be sampled B times for a given (x,y,t)
Θ = random-walk direction-angle within the interval $(0,2\pi)$, to be sampled B times for a given (x,y,t)

The above equations for new tracer position (x',y') are applied B times for a given (x,y,t), so that B random samples of (x',y') are generated. Physically, the tracer at (x,y) is made **to split or branch into B partitions** at B new positions (x',y'). At every new (x',y'), the new values of temperature or concentration are accumulated out of the partitions (with linear decay included) of the old values originating from several tracer positions (x,y). This is essentially the random process of diffusion, with advection and linear decay, simulated every time step.

Boundary conditions are handled under the random-walk method in the following manner. Open boundaries, such as downstream ends, are treated as absorbing barriers when they are crossed by tracers. Zero normal derivatives of temperature or concentration may be imposed on solid boundaries of the flow domain. These solid boundaries are treated as reflecting barriers, relative to the computed movement of the tracers. Source boundary points where temperature or concentration is prescribed are not overridden by tracer movement at all.

The visualization of temperature and solute concentration is achieved by color coding of tracers following a displayed **color scale** which indicates corresponding ranges of values of these variables. It is found effective to adopt a **logarithmic color scale** (similar to pH) instead of linear, so as to encompass values for different orders of magnitudes.

Boundary Layer Thermal Diffusion (Program BLAYDIS)

The objective here is to compute and display the **thermal boundary layer** developed in the boundary-layer flow above the straight wall described in program BLAYSEP, using stochastic or random-walk simulation as presented earlier. The **source of heat** is a hot point or segment on the straight wall. The particular boundary condition prescribed on the straight wall (y=0) is one of three options for temperature profile: distributed, point, or gaussian.

This program also illustrates the relative growth rate of the thicknesses of the momentum boundary layer and the thermal boundary layer. The parameter which best indicates this relation is also computed, namely the **Prandtl Number**:

$$Pr = \tau / k$$

where

Pr = Prandtl Number
= ratio of momentum diffusion to thermal diffusion
τ = fluid kinematic viscosity
k = fluid thermal diffusivity

At low Pr, the momentum boundary layer is thinner than the thermal boundary layer. At high Pr, the opposite is true.

Thermal Diffusion in Circular-Cylinder Vortex Wake (Program CYLDIS)

The objective here is to compute and display the **temperature field or thermal wake** developed in the vortex wake behind a circular cylinder described in program KARMAN, using stochastic or random-walk simulation as presented earlier. The **source of heat** is the entire circular boundary of the cylinder. The particular boundary condition prescribed on the circular boundary is a distributed uniform temperature in either continuous or batch mode.

Shear Flow with Stochastic Solute Dispersion (Program DISPER)

The objective here is to compute and display the **solute concentration field or plume** developed in a rectangular flow domain or channel with specified velocity field or shear flow, using stochastic or random-walk simulation as presented earlier. The combined mixing process due to enhancement of mass diffusion by shear-flow advection is termed as dispersion.

The options for the velocity field or shear flow are: uniform, linear, semi-parabolic, parabolic, or circulating field.

The **source of solute** is an upstream boundary point or line segment where a concentration distribution is prescribed. The particular boundary condition set on the source is one of three options for concentration profile (in either continuous or batch mode): distrubuted, point, or gaussian distribution.

FLOW VISUALIZATION CASES IV: GEOSTROPHIC FLOW

Geostrophic flow is fluid motion affected by **Corioli's forces** which arise from a rotating reference frame such as the earth's surface. The NSE are adjusted in this case by including Corioli's acceleration terms in the left-hand side of the equations. Corioli's acceleration points in the direction 90° to the left of the velocity. Equivalently this can be transposed to the right-hand side to represent the so-called Corioli's force, which therefore points in the direction 90° to the right of the velocity.

To illustrate the nature of geostrophic flows, the following formulation of the NSE for the two velocity components (u,v) is presented:

$$\delta u/\delta t + [u\ \delta/\delta x + v\ \delta/\delta y + w\ \delta/\delta z]\ u - f\ v =$$

$$- 1/p\ \delta(P + pgH)/\delta x + \tau\ [\delta^2/\delta x^2 + \delta^2/\delta y^2 + \delta^2/\delta z^2]\ u$$

$$\delta v/\delta t + [u\ \delta/\delta x + v\ \delta/\delta y + w\ \delta/\delta z]\ v + f\ u =$$

$$- 1/p\ \delta(P + pgH)/\delta y + \tau\ [\delta^2/\delta x^2 + \delta^2/\delta y^2 + \delta^2/\delta z^2]\ v$$

$$\delta u/\delta x + \delta v/\delta y + \delta w/\delta z = 0 \text{ (continuity)}$$

where the symbols are as originally defined for the NSE, except for the Corioli's terms (-fv, fu),

where

f = $2\ \Omega\ \sin(\phi)$ = Corioli's parameter
Ω = rotational velocity of the earth
ϕ = terrestial latitude

Two further special cases are derived below.

For steady flow where (u,v) are just functions of vertical coordinate z, the above equations can be reduced to:

$$- f\ v = -1/p\ \delta P/\delta x + \tau\ \delta^2 u/\delta z^2$$

$$f\ u = -1/p\ \delta P/\delta y + \tau\ \delta^2 v/\delta z^2$$

subject to the following boundary conditions:

At the bottom: $z = 0$: $u = v = 0$ (no-slip condition)
At large height: $z = \infty$: $u = V$, const., $v = 0$
all z-derivatives = 0
so that at all (x,y,z), $\delta P/\delta x = 0$, $\delta P/\delta y = - pfV$, const.

The equations are further reduced to:

$$- f\ v = \tau\ \delta^2 u/\delta z^2$$

$$f\ (u-V) = \tau\ \delta^2 v/\delta z^2$$

The solution to the above problem describes the motion inside the **boundary layer called Ekman layer**, discussed in the next section.

An unsteady-state, two-dimensional, frictionless, free-surface flow formulation, involving depth-averaged velocity components (u,v) and a variable free-surface elevation z, on top of a constant average depth H, can be derived.

Depth-averaging of the original continuity equation yields (for $|z| << H$):

$$(\delta u/\delta x + \delta v/\delta y) + (1/H)\ [w(H+z) - w(0)] = 0$$

but

$$\begin{aligned} w(H+z) &= \delta z/\delta t + u\ \delta z/\delta x + v\ \delta z/\delta y \\ &\quad \text{(boundary condition on free surface)} \\ &\approx \delta z/\delta t\ \text{(linearized)} \\ w(0) &= 0\ \text{(bottom boundary condition)} \end{aligned}$$

Hence, the linearized depth-averaged continuity equation becomes

$$\delta z/\delta t + H\ (\delta u/\delta x + \delta v/\delta y) = 0$$

In addition, depth-averaging of the NSE, replacing H by (H+z), and neglecting the advection, pressure gradient, and viscous shear terms, result in the following linearized equations of motion:

$$\delta u/\delta t + g\ \delta z/\delta x - f\ v = 0$$

$$\delta v/\delta t + g\ \delta z/\delta y + f\ u = 0$$

where

$u = u(x,y,t)$ = depth-averaged u-component
$v = v(x,y,t)$ = depth averaged v-component
$z = z(x,y,t)$ = free surface elevation

The above linearized equations of motion and continuity describe the movement of two kinds of gravity surface waves affected by Corioli's forces, namely the **Kelvin waves and Poincare waves**. These two wave functions constitute two linearly independent analytical solutions of the equations. These are also discussed in two succeeding sections.

To indicate the ratio of inertial forces to Corioli's forces, the **Rossby Number**, Ro, is computed:

$$Ro = [p\ V^2\ L^2]/[p\ L^3\ f\ V] = V/[fL]$$

where

Ro = Rossby Number
= ratio of inertial forces to Corioli's forces
f = Corioli's parameter
V = velocity scale
L = length scale
p = fluid density.

Low Ro represents dominance of Corioli's forces, while high Ro means relatively low Corioli's forces.

3-D Boundary Layer in an Ekman Layer
(Program 3DEKMAN)

The objective is to display the steady three-dimensional boundary-layer flow in an **Ekman layer** produced by the interaction between Corioli's and viscous shear forces near a solid bottom. The flow is characterized by a velocity (u,v) = (V,0) at large height z, with nonzero cross-flow velocity v present at intermediate z, down to zero velocity at the bottom (z=0). The flow is uniform in each horizontal plane (constant z), but the flow direction varies with height, in such a way that the ends of the velocity vectors form a **spiral** (Ekman spiral) when viewed from the top. The angle between the velocity at high z and that at low z approaches 45°.

Geostrophic Flow in a Kelvin Wave
(Program KELVIN)

The objective is to display the first type of gravity surface wave affected by Corioli's forces called **Kelvin wave**. The wave flow is characterized by a unidirectional oscillating flow velocity, and a free surface with a progressing **sinusoidal** shape in the x-direction, and an oscillating **hyperbolic** shape in the y-direction. There are stationary points in the Kelvin wave surface called **amphidromic points**, located at all $x=n\pi/k$, $y=0$, where z is zero at all time t.

Geostrophic Flow in a Poincare Wave
(Program POINCARE)

The objective is to display the second type of gravity surface wave affected by Corioli's forces called **Poincare wave**. The wave flow is characterized by a **bidirectional** oscillating velocity which traces **elliptical pathlines**, and a free surface with a progressing **sinusoidal** shape. There are **nodal lines** in the superposition of two Poincare waves, where z is zero at all time t.

FLOW VISUALIZATION CASES V: OPEN-CHANNEL FLOW

Interactive Sluice-Gate Hydraulic-Jump Simulator
(Program HJFLUME)

The objective here is to provide a highly interactive visualization platform for open-channel flow processes. The program is an **interactive sluice-gate hydraulic-jump simulator**. It is actually a **mathematical-numerical replica of an existing hydraulic laboratory apparatus** of the NHRC, and, as such, can provide a comparison or even prediction tool for laboratory observations. The program performs simulation with very small model-time step (less than 0.1 second), this being a requirement for numerical stability, so that the visualization becomes a "slow-motion" animation of flow processes which are in reality occurring too fast for long-eye-contact observation in the laboratory.

The interactive simulator incorporates the dynamic properties of the following hydraulic fixtures and installations in series:

1. A water-supply pipe installed with a head-valve, which can be manipulated to control the inflow of water coming from an elevated head tank.
2. An open mixing chamber which receives water from the supply pipe and where sufficient depth is maintained to provide head to a sluice gate located downstream.
3. A sluice gate with a variable opening at the bottom, being the primary control for the formation of hydraulic jump.
4. A long rectangular flume downstream of the sluice gate where a hydraulic jump may be formed.
5. A louvre-type weir at the exit end of the flume which can be manipulated to provide secondary control of the tailwater depth downstream of a hydraulic jump.

The simulator is so designed that at any time during simulation (computation and visualization), the user can interrupt it to vary head-valve, sluice-gate, or louvre weir opening. Upon resumption of simulation, the flow variables (depths and discharges) respond to the new settings, the changes becoming very visible and sometimes very spectacular.

Among the open-channel flow processes that can be visualized are formation of free or submerged hydraulic jump under either steady or unsteady condition of gate control, oscillations or else hysteresis behavior in free-jump to submerged-jump transitions and vice-versa, generation of moving jumps or surges due to sudden changes in sluice-gate or weir openings, critical or subcritical flows created by sudden loss of gate-control, and depth and discharge variations caused by changes in head-valve openings.

The mathematical features of the simulator are as follows:

1. The continuity equation for the mixing chamber.
2. The discharge-rating equations for the sluice gate, in either normal forward flow, or reverse flow, with or without gate control.
3. The test for either free or submerged hydraulic jump, through the use of the specific momentum function.

4. The flow in the flume itself, as governed by a set of continuity and momentum equations applicable to horizontal rectangular channels.

5. The discharge-rating equation for the louvre-weir derived from an adjusted rectangular-weir formula.

The simulation procedure performs a simultaneous solution of all continuity and momentum equations for unknown depths and discharges for every time step using an explicit finite-difference scheme. The discharge through the sluice gate poses as the upstream boundary condition on the flume, while the louvre-weir discharge holds as the downstream boundary condition.

RESULTS AND DISCUSSION

Summary of Developed Visualization Programs

Coding, debugging, test and demonstration runs have been accomplished for 25 fluid-flow visualization programs, grouped according to types of fluid motion.

(1) Potential (Irrotational, or Ideal) Flow

(a) *2-D Potential Flows Around Submerged Bodies*

KARMAN	2-D Karman Vortex Wake behind a Circular Cylinder
KARPLAT	2-D Karman Vortex Wake behind a Flat Plate
FLOWCYL	2-D Potential Flow past a Circular Cylinder
FLODBLET	2-D Potential Wake of a Moving Cylinder
FLOWPLAT	2-D Potential Flow past a Flat Plate
AIRFOIL	2-D Potential Flow past a Thin Airfoil

(b) *Additional 2-D Potential Flows*

FLOWSS	2-D Potential Flow past a Source-Sink Pair
FLOWVORT	2-D Potential Flow past a Vortex Pair
FLOWCHAN	2-D Potential Flow from a Channel Outfall

(c) *2-D Gravity Wave Motion*

WAVEDYN	2-D Small-amplitude Surface Waves (Eulerian)
WAVETRAN	2-D Small-amplitude Surface Waves (Lagrangian)
WAVEGRP	2-D Small-amplitude Surface Waves Group

(d) *3-D Potential Vortex Flows*

3DVORTEX	3-D Vortex Wake behind the ff: * Finite Airfoil * Rotors Propellers or HAWT Darrieus Rotors or VAWT Ducted Rotors or DAWT * Horseshoe Vortex

(2) Viscous Flow (including Boundary Layer Flows)

(a) *Laminar Flow*

BLAYSEP	2-D Boundary Layer Separation along a Straight Wall
LAMJET	2-D Submerged Laminar Jet from Narrow Slots and Orifices
LAMVORT	3-D Decaying Laminar Vortex ("after stirring the teacup")
3DBLAYER	3-D Boundary Layer below a Forced Vortex ("bottom of the teacup")

(b) *Turbulent Flow*

TURBO	2-D Laminar and Turbulent Flow in a Smooth Pipe

(3) Flow with Heat and Mass Transfer

BLAYDIS	2-D Thermal Diffusion in the Boundary Layer along a Straight Wall
CYLDIS	2-D Thermal Diffusion in the Vortex Wake of a Circular Cylinder
DISPER	2-D Shear Flow with Stochastic Solute Dispersion

(4) Geostrophic Flow (flow affected by Coriolis Force)

3DEKMAN	3-D Boundary Layer in an Ekman layer
KELVIN	2-D Geostrophic Flow in a Kelvin Wave
POINCARE	2-D Geostrophic Flow in a Poincare Wave

(5) Open-channel Flow

HJFLUME	Interactive Sluice-Gate and Hydraulic-Jump Simulator (a mathematical-numerical replica of a physical NHRC laboratory apparatus).

The various visualization programs, as implemented, utilize in different combinations the computer graphics and animation techniques discussed earlier. These are shown in Table 1, with a legend supplied below.

Table 1 - VISUALIZATION PROGRAMS, FLOW TYPES, AND TECHNIQUES

					TECHNIQUES						
	PROGRAMS	Yr	2-D	3-D	SPA	DPA	IR	PR	IOP	CM	SS
	POTENTIAL FLOWS										
1.	KARMAN	*(1)*	*2-D*		*SPA*		*IR*				
2.	KARPLAT	*(2)*	*2-D*		*SPA*		*IR*			*CM*	
3.	FLOWCYL	*(1)*	*2-D*		*SPA*		*IR*	*PR*			
4.	FLODBLET	*(2)*	*2-D*		*SPA*	*DPA*	*IR*				
5.	FLOWPLAT	*(2)*	*2-D*		*SPA*		*IR*			*CM*	
6.	AIRFOIL	*(1)*	*2-D*		*SPA*		*IR*	*PR*			
7.	FLOWSS	*(1)*	*2-D*		*SPA*		*IR*	*PR*			
8.	FLOWVORT	*(1)*	*2-D*		*SPA*		*IR*	*PR*			
9.	FLOWCHAN	*(2)*	*2-D*		*SPA*		*IR*	*PR*		*CM*	
10.	WAVEDYN	*(1)*	*2-D*		*SPA*	*DPA*	*IR*				
11.	WAVETRAN	*(1)*	*2-D*		*SPA*	*DPA*	*IR*				
12.	WAVEGRP	*(1)*	*2-D*		*SPA*	*DPA*	*IR*				
13.	3DVORTEX	*(1)*		*3-D*	*SPA*				*IOP*		
	VISCOUS FLOWS										
14.	BLAYSEP	*(2)*	*2-D*		*SPA*		*IR*	*PR*			
15.	TURBO	*(2)*	*2-D*		*SPA*		*IR*	*PR*			*SS*
16.	LAMJET	*(2)*	*2-D*		*SPA*		*IR*	*PR*			
17.	LAMVORT	*(2)*	*2-D*		*SPA*		*IR*	*PR*	*IOP*		
18.	3DBLAYER	*(2)*		*3-D*	*SPA*		*IR*	*PR*	*IOP*		
	FLOWS WITH HEAT AND MASS TRANSFER										
19.	BLAYDIS	*(2)*	*2-D*		*SPA*		*IR*				*SS*
20.	CYLDIS	*(2)*	*2-D*		*SPA*		*IR*				*SS*
21.	DISPER	*(2)*	*2-D*		*SPA*		*IR*				*SS*
	GEOSTROPHIC FLOWS										
22.	3DEKMAN	*(2)*		*3-D*	*SPA*		*IR*	*PR*	*IOP*		
23.	KELVIN	*(2)*	*2-D*		*SPA*	*DPA*	*IR*		*IOP*		
24.	POINCARE	*(2)*	*2-D*		*SPA*	*DPA*	*IR*		*IOP*		
	OPEN-CHANNEL FLOW										
25.	HJFLUME	*(2)*	*1-D*		*SPA*		*IR*				

Legend:

(1)	=	year-1 accomplishments
(2)	=	year-2 accomplishments

2-D = two-dimensional flow

3-D = three-dimensional flow

SPA = single-page animation (text, graphics, sound)

DPA = double-page animation (text & graphics)

IR = instant replay

PR = palette rotation (neon-light effect)

IOP = isometric and/or orthographic projections

CM = conformal mapping

SS = stochastic simulation

Discussion of Techniques

Out of the 25 programs, 19 programs for **steady 1-D, 2-D or 3-D flows** require only **single-page animation** (SPA) to obtain time-coherent visualization of fixed streamlines. The streamlines initially propagate at a slower pace during precomputation. Afterwards, **instant replay** (IR) gives a faster visualization. In addition, the use of sound and bright tracers during instant replay imparts a sense of rhythm to the flow. Repeatable instant replay allows one to look at the flow propagation more than once, and each time one could notice more features of motion. **Color coding** of tracers is used to indicate variation of scalars such as pressure, temperature, or concentration, or else to distinguish among individual tracers.

Palette rotation or neon-light effect (PR) is applied in 11 programs for 2-D or 3-D steady flows in which the appreciation of velocity variation along fixed streamlines is of interest. It imparts a very dynamic sense of linear motion along the streamlines, when in fact the plotted streamlines are not moving but only the color instead. While instant replay tends to emphasize and compare individual tracer (point) motion, palette rotation brings out the variation of vector linear motion within and between streamlines. Anyone who has gazed at a dynamic neon sign from a good distance will appreciate this technique.

Programs **KARMAN** and **KARPLAT**, which visualize **unsteady 2-D flow** in a Karman Vortex Wake, utilize **single-page animation**, and produces time-incoherence of pathlines. This is not too objectionable in the sense that the viewer gets a picture of "mixing" or a semblance of turbulence, as in reality. The use of sound and bright tracers during **instant replay** further emphasizes a strong sense of rhythm and complicated vortex motion.

The surface wave programs **(WAVEDYN, WAVETRAN, WAVEGRP, KELVIN, POINCARE)**, for visualizing unsteady 2-D periodic or quasi-periodic wave motion, initially uses **single-page animation** to show the time-incoherent circular, spiral, linear, or elliptical orbits of tracers as these are precomputed. After the precomputation, it applies a **double-page animation** (DPA) in order to display the distinct translational and oscillatory motion of the free surface, and the time-coherent orbital motion of tracers on and below the free surface. Color is utilized to differentiate the tracers, and to prove the point that the translational surface wave motion is not equivalent to material (tracer) orbital motion - instead, wave motion is the envelope of tracer orbital motion.

Similarly for the unsteady potential wake behind a moving cylinder (program **FLODBLET**), single-page and double-page animation with instant replay provide effective appreciation of the relative motion imparted by the cylinder to the initially static fluid.

It can be observed that **double-page animation**, as applied above, may slow down because of the time required to generate a new detailed image on the invisible screen page. This keeps the current visible page on wait status before swapping. For the sake of speed, a tolerable minimum of detail in an image has to be generated, such as a low but adequate number of tracers. In the case of program KARMAN, the large amount of detail that has to be present in the Karman Vortex Wake inhibits the use of double-page animation. In the case of the latter programs KELVIN and POINCARE, double-page animation of highly-detailed pictures are made, which turns out to be slow as expected in a PC-80386 but is fast enough in a PC-80486.

For the 3-D flow visualization (3DVORTEX), in particular for a progressively growing 3-D vortex wake, the number of vortextubes increases linearly with model time, but the computation time of their interaction grows geometrically with model time. Double-page animation would be too slow. Hence, only **single-page animation** is implemented. To fully visualize the three-dimensional nature of the flow, simultaneous **isometric and orthographic projections** (IOP) of vortextubes are displayed. Color is used to distinguish the different types of vortextubes.

After visualization of vortextubes, a single-page pathline tracing routine in isometric projection follows, wherein **rhythmic sound is added with pitch varying with flow speed**. High pitch is heard whenever a tracer encounters the neighborhood of a vortextube, while a flat low pitch is produced once the tracer moves through a slower uniform free stream. Since the pitch is taken from the familiar 8-note scale (octave), sometimes tracers moving through a complicated flow region can play almost musical numbers. Granting that the ears can have higher discrimination for pitch and rhythm, compared to the ability of the eyes to discriminate tracer speed and motion (at least the ears and eyes of the author), **computed sound** may prove to be a significant medium to accompany visualization.

Isometric and orthographic projections (IOP) are also applied to the other 3-D flows, or to 2-D flows more effectively shown in 3-D projections (we have to make that distinction!). These are programs **3DBLAYER** and **3DEKMAN** (3-D flows), and programs **LAMVORT**, **KELVIN**, and **POINCARE** (2-D flows in 3-D projections or IOP). The last three programs are displayed in 3-D projections in order to visualize a third dependent variable aside from two horizontal velocity components, namely, pressure head or free surface elevation. The true three-dimensional character of all fluid motion is emphasized here, regardless of whether the mathematical model is 2-D or 3-D.

Conformal mapping (CM), a classical textbook analytical method for solving differential equations, is here given a numerical calling card. In programs **KARPLAT** and **FLOWPLAT**, it is applied to a flow field around a circular cylinder previously computed numerically; conformal mapping (which is a fast analytical step) transforms the field into a numerical solution applicable to a flow around an inclined flat plate. In other words, a single numerical effort solves essentially two different flow problems.

While the **potential function is invariant** under conformal mapping and the streamlines in the new field are correct, the **velocity magnitudes plotted as line segments along them are distorted compared to their true transformed values**. This is the reason why palette rotation, which requires **undistorted linear segments per constant time-step along plotted streamlines**, is not used in KARPLAT and FLOWPLAT. The two programs suffer velocity distortion after conformal mapping, and velocity adjustment is made only to correct the **scalar pressure color-coding** during instant replay. Memory limitation prevents the **massive vector graphics correction** needed to adjust for velocity distortion in the final streamlines.

In a latter program, **FLOWCHAN**, where conformal mapping is also used, velocity distortion is preempted by actually **predistorting the velocity magnitude** of the uniform-flow field prior to conformal mapping to an outfall flow, in order to obtain true velocities for the latter, which can be validly displayed by palette rotation.

Stochastic simulation (SS) is used in program **TURBO**, with a simple model for random velocity fluctuations; and in programs **BLAYDIS**, **CYLDIS**, and **DISPER**, with a random-walk model to simulate thermal and mass diffusion. The technique allows the appreciation of turbulent flow and of the essentially random nature of diffusion and dispersion, against a background provided by the given mean motion or shear flow. In addition, random-walk simulation to solve trasport equations is proven to be a very feasible and effective numerical scheme of solving such kind of equations and visualizing the results. Visualization gives a picture of the diffusion process, which in the first place has provided the physical concept behind the random-walk solution method, which in turn makes its visualization possible.

Finally, highly-interactive simulation with graphics and animation, comparable to computer games, has been achieved with program **HJFLUME** which utilizes time-consuming explicit-scheme of solution. Although a 25 MHz machine will give a decent performance of this program, a much faster machine running is needed to obtain excellent performance.

CONCLUSION AND RECOMMENDATIONS

The following general conclusions and recommendations can be made:

1. Single-page animation (SPA) is adequate to visualize steady two-dimensional flow with perfect time-coherence of fixed streamlines. Speed is high provided

that the number of flow elements is not very large. It is a trade-off between detail and speed.

2. Single-page animation can be used to visualize unsteady two-dimensional flow, but at the sacrifice of time-coherence of pathlines. In certain cases, time-incoherence of pathlines may not be objectionable.

3. Double-page animation (DPA), applied to unsteady two-dimensional flow, preserves time-coherence of tracer motion. However, a large amount of detail in every image will significantly slow down the animation. This dictates keeping the amount of detail to the tolerable minimum.

4. Palette rotation or neon-light effect (PR) is effective in emphasizing variation of linear motion along and between streamlines. It requires undistorted velocity magnitudes plotted as line segments per time-step on the streamlines.

5. Color coding should be used unsparingly to represent scalar fields (such as pressure, temperature, concentration) and to distinguish among different flow elements (tracer, vortextube, flowline, free surface, solid boundary, etc.). Bright colors make impressive tracers.

6. Isometric and orthographic projections (IOP) are necessary techniques for visualization of three-dimensional flow and also for revealing the true three-dimensional character of some two-dimensional flow problems.

7. Computed sound proves to be an effective medium to accompany visualization in view of the ability of pitch and rhythm to strongly suggest fluid motion and speed.

8. Conformal mapping (CM) provides a technique for solving and visualizing two potential flow problems with a single numerical effort. The tradeoff is that velocity magnitude is distorted after conformal mapping; to avoid it in the final result, predistortion of velocity magnitudes prior to conformal mapping is required.

9. Stochastic simulation (SS) is a very feasible technique for visualization of turbulent flow and flows with heat and mass transfer. It is a solution method suggested by the physical nature of the phenomenon being visualized. For transport problems, it is a substitute to other time-consuming numerical methods of solutions, which may require matrix inversion.

10. The above conclusions apply to real-time mode of visualization, wherein input, computations, and display are done interactively. This should maintain the user-friendliness of the visualization program. Input menus should be made simple and understandable, containing suggested ranges of input parameter values. Highly-interactive visualization programs similar to computer games can also be made but in view of possible long model computations, machines running at faster than 25 MHz are needed for excellent results.

11. Flow visualization programs running in PC-80386 machines under MS.DOS can serve as effective, practical, and affordable tools for teaching and explaining flow concepts and variables.

12. Based on the positive results of this effort to visualize the 25 prototype cases of physical phenomena with momentum, heat, and mass transfer, a general recommendation can be made to apply computer graphics and animation techniques in order to visualize related phenomena such as:

    a. Multiphase systems (solid-liquid-gas), which include physical transformations such as boiling, condensation, crystallization, distillation, separation, sedimentation, fluidization, etc., provided that the mathematical models and solution methods are available.

    b. Chemical and biochemical reactions in homogeneous and heterogeneous systems or reactors. A large variety of chemical reactions can be studied, provided that the mathematical models and solution methods are also available.

    c. Other important problems involving simultaneous momentum, heat, and mass transfer, with chemical reactions, such as gasification and combustion, provided that the mathematical models and solution methods are available.

    d. Lahar flow, provided that a validated mathematical model has been developed (a non-Newtonian laminar flow model for thick lahar is a candidate model; thin lahar, being a highly turbulent sediment-transport phenomenon, is tougher to model mathematically).

ACKNOWLEDGEMENTS

The author wishes to acknowledge the financial support given to this two-year project by the Philippine Council for Advanced Science and Technology Research and Development (PCASTRD) of the Department of Science and Technology (DOST). Thanks and gratitude are due to Dr. R. Panlasigui, PCASTRD Director, for his trust and confidence in this project. Also thanks to Dr. S. Chua, Ms. H. Mercado, Mr. D. Santos and other staff members of PCASTRD for their strong concern, constant assistance, and unwavering patience in the administration and monitoring of this project.

The author also acknowledges the cooperation and institutional and staff support provided by the National Hydraulic Research Center (NHRC), U. P. Diliman, through its Director, Prof. Angel A. Alejandrino. Both for the sustained conduct and positive results of the research, and the practical application of the software for visualization in the course ES 15 (Mechanics of Fluids), the conducive research-and-teaching environment at NHRC has much to claim credit.

Also expressed is the gratitude towards the Department of Engineering Sciences and the College of Engineering, U. P. Diliman for the teaching load credit accorded to the author's time spent to do this research. In particular, thanks to Prof. E. P. Quiwa and Dr. R. B. Vea, past and present chairmen, and Dean F. L. Viray of said department and college, respectively.

REFERENCES

1. Ashurst, Wm. T. and D. I. Meiron (1987). *Numerical Study of Vortex Reconnection*, **Physical Review Letters**, 58(16), 20 April.

2. Beale, J. Thomas (1986). *A Convergent 3-D Vortex Method with Grid-free Stretching*, **Mathematics of Computation**, 46(174):401-424, April.

3. Chorin, Alexandre Joel (1980). *Vortex Models and Boundary Layer Instability*, **SIAM J. Sci. Stat. Comput.**, 1(1), March.

4. Chua, K., A. Leonard, F. Pepin, and G. Winckelmans (1988). *Robust Vortex Methods for Three-dimensional Incompressible Flows*, in ASME (1988). **Recent Developments in Computational Fluid Dynamics.**, AMD-95: 33-44. T. E. Tezduyar and T. J. R. Hughes (Eds.).

5. Colonna, Jean-Francois (1987). *Focus - Visualization*, **Computer Graphics World**, December.

6. Hasimoto, Hidenori (1985). *Several Aspects of the Motion of Vortex Filaments*, in **Studies on Turbulent Phenomena**, T. Tatsumi, H. Maruo, and H. Takami (eds.). Association for Science Documents Information.

7. Kawamura, Tetuya (1985). *Direct Simulation of Flows at High Reynolds number Using a Third-Order Upwind Scheme*, in **Studies on Turbulent Phenomena**, T. Tatsumi, H. Maruo, and H. Takami (eds.). Association for Science Documents Information.

8. Kida, S. and M. Takaoka (1987). *Bridging in Vortex Reconnection*, **Phys. Fluids**, 30(10), October.

9. Kutler, P. (1988). *Progress and Future Directions in Computational Fluid Dynamics*, in ASME (1988). **Large-Scale Interaction Problems**. AMD 94: 55-71. A. K. Noor and H. S. Levine (eds.).

10. Lam, K. (1987). *Potential Flow Calculation by Surface Vorticity Method and Computer Graphics*, **Comput. & Graphics**, 11(1):35-47.

11. Leonard, A. (1985). *Computing Three-dimensional Incompressible Flows with Vortex Elements*, in **Annual Review of Fluid Mechanics**, 17: 523-559.

12. Liongson, Leonardo Q. (1993). Software Development for Visualization of Fluid

Motion Using Computer Graphics and Animation Techniques. Second-Year Report, PCASTRD-UP NHRC.

13. Meyer, Ricahrd E. (1971). **Introduction to Mathematical Fluid Dynamics**. Dover Publications, Inc. New York.

14. Milne-Thomson, L. M. (1960). **Theoretical Hydrodynamics**. Macmillan.

15. Neal, Margaret (1988). *Displays on Display - When did Scientific Visualization Really Begin?*, **IEEE Computer Graphics and Applications**. November.

16. Olfe, Daniel B. (1987). **Fluid Mechanics Programs for the IBM PC**, Chapters 3 (Flow Visualization, Kinematics, Fluxes, and Forces) and Chapter 4 (Potential Flow). Mc-Graw Hill Book Co.

17. Pluir, Alain and Robert M. Kerr (1987). *Numerical Simulation of Interacting Vortex Tubes*, **Physical Review Letters**, 58(16), 20 April.

18. Schlichting, Hermann (1979). **Boundary-Layer Theory**,. seventh edition. Mc-Graw Hill Book Co.

19. Strickland, J. H. (1986). *A Review of Aerodynamic Analysis Methods for Vertical-Axis Wind Turbine*, in ASME (1986). **Fifth ASME Wind Energy Symposium**, SED-2:7-17. A. H. P. Swift (ed.).

20. Taneda, Sadatoshi (1985). *Flow Visualization*, in **Recent Studies on Turbulent Phenomena**, T. Tatsumi, H. Maruo, and H. Takami (eds.). Association for Science Documents Information.

21. Upson, C., T. Faulhaber, D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, and A. van Dam (1989). *Scientific Visualization - The Application Visualization System: A Computational Environment for Scientific Visualization*, **IEEE Computer Graphics and Applications**. July.

22. Weston, Robert P. (1987). *Focus - Mastering Fluid Dynamics*. **Computer Graphics World**. August.