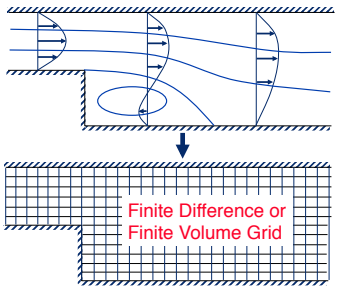
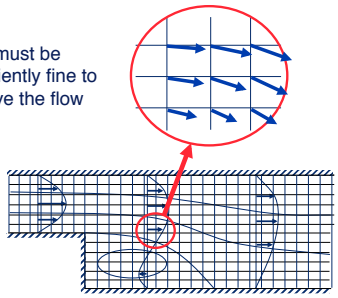
What is Computational Fluid Dynamics (CFD)?



Grid must be sufficiently fine to resolve the flow



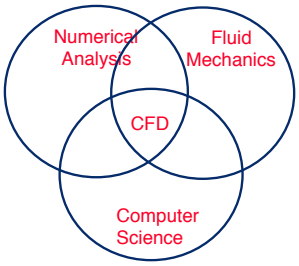
Using CFD to solve a problem:

Preparing the data (preprocessing):
Setting up the problem, determining flow parameters and material data and generating a grid.

Solving the problem

Analyzing the results (postprocessing):
Visualizing the data and estimating accuracy. Computing forces and other quantities of interest.

CFD is an interdisciplinary topic



Many website contain information about fluid dynamics and computational fluid dynamics specifically. Those include

NASA site with CFD images
http://www.nas.nasa.gov/SC08/images.html

CFD Online: An extensive collection of information but not always very informative
http://www.cfd-online.com/

eFluids.com is a monitored site with a large number of fluid mechanics material
http://www.e-fluids.com/

Beginning of CFD

CFD at Los Alamos

The MANIAC at Los Alamos had already stimulated considerable interest in numerical solutions at the Laboratory. However, CFD in the modern sense started with the formation of the T3 group. Early development in the Group focused on:

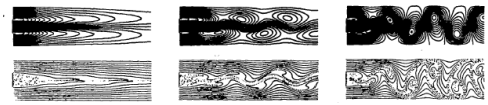
Compressible flows: Particles in Cells (1955): Eularian –Lagrangian method where particles move through a fixed grid

Low-speed (incompressible) flows
- Vorticity streamfunction for homogeneous (mostly 2D) flows (1963)
- Marker and Cell for free surface and Multiphase flows in primitive variables (1965)

All-speed flows—ICE (1968, 1971)

Arbitrary–Lagrangian–Eulerian (ALE) methods

Incompressible flows—Vorticity-Streamfunction Method



Computations of the development of a von Karman vortex street behind a blunt body by the method developed by J. Fromm. Time goes from left to right showing the wake becoming unstable.

From: J. E. Fromm and F. H. Harlow, Numerical Solution of the Problem of Vortex Street Development: ***Phys. Fluids 6 (1963), 975.***

Incompressible flows—the MAC Method

Primitive variables (velocity and pressure) on a staggered grid

The velocity is updated using splitting where we first ignore pressure and then solve a pressure equation with the divergence of the predicted velocity as a source term

Marker particles used to track the different fluids

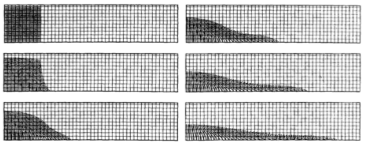
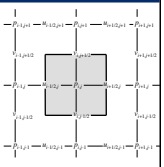


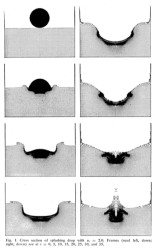
FIG. 5. Configuration of marker particles for the broken dam at times $t = 0, 0.5, 1.0, 1.5, 2.0, 2.5$. There were four computational cells for each square of the grid shown here.



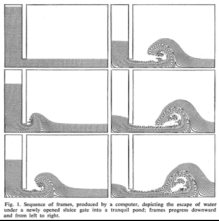
The dam breaking problem simulated by the MAC method, assuming a free surface. From F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. Phys. Fluid, 8: 2182–2189, 1965.

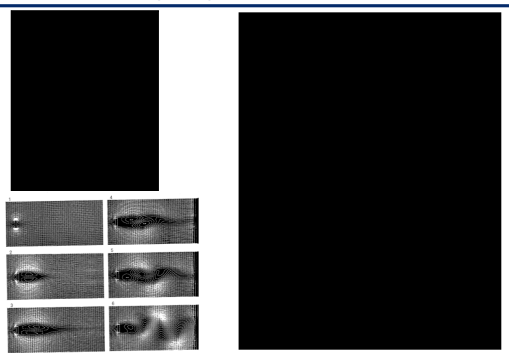
Early Publicity: Science Magazine

F.H. Harlow, J.P. Shannon, Distortion of a splashing liquid drop, Science 157 (August) (1967) 547–550.



F.H. Harlow, J.P. Shannon, J.E. Welch, Liquid waves by computer, Science 149 (September) (1965) 1092–1093.





Computational Fluid Dynamics
Beginning of CFD

"While most of the scientific and technological world maintained a disdainful distaste (or at best an amused curiosity) for computing, the power of the stored-program computers came rapidly into its own at Los Alamos during the decade after the War."

From: Computing & Computers: Weapons Simulation Leads to the Computer Era, *by Francis H. Harlow and N. Metropolis*

"Another type of opposition occurred in our interactions with editors of professional journals, and with scientists and engineers at various universities and industrial laboratories. One of the things we discovered in the 1950s and early 1960s was that there was a lot of suspicion about numerical techniques. Computers and the solutions you could calculate were said to be the playthings of rich laboratories. You couldn't learn very much unless you did studies analytically."

From: Journal of Computational Physics 195 (2004) 414–433
Review: Fluid dynamics in Group T-3 Los Alamos National Laboratory (LA-UR-03-3852), by Francis H. Harlow

Computational Fluid Dynamics
Beginning of CFD

CFD goes mainstream

Early studies using the MAC method:

R. K.-C. Chan and R. L. Street. A computer study of finite-amplitude water waves. J. Comput. Phys., 6:68–94, 1970.

G. B. Foote. A numerical method for studying liquid drop behavior: simple oscillations. J. Comput. Phys., 11:507–530, 1973.

G. B. Foote. The water drop rebound problem: dynamics of collision. J. Atmos. Sci., 32:390–402, 1975.

R. B. Chapman and M. S. Plesset. Nonlinear effects in the collapse of a nearly spherical cavity in a liquid. Trans. ASME, J. Basic Eng., 94:142, 1972.

T. M. Mitchell and F. H. Hammitt. Asymmetric cavitation bubble collapse. Trans ASME, J. Fluids Eng., 95:29, 1973.

Beginning of commercial CFD—Imperial College (Spalding)

Computations in the Aerospace Industry (Jameson and others)

And others!

Computational Fluid Dynamics
Beginning of CFD

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 195 (2004) 414–433
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Review

Fluid dynamics in Group T-3
Los Alamos National Laboratory ☆
(LA-UR-03-3852)

Francis H. Harlow *

Los Alamos National Laboratory, MS-B216, Los Alamos, NM 87544, USA

Received 12 June 2003; received in revised form 25 August 2003; accepted 3 September 2003

The development of computer fluid dynamics has been closely associated with the evolution of large high-speed computers. At first the principal incentive was to produce numerical techniques for solving problems related to national defense. Soon, however, it was recognized that numerous additional scientific and engineering applications could be accomplished by means of modified techniques that extended considerably the capabilities of the early procedures. This paper describes some of this work at The Los Alamos National Laboratory, where many types of problems were solved for the first time with the newly emerging sequence of numerical capabilities. The discussions focus principally on those with which the author has been directly involved.

Computational Fluid Dynamics
Beginning of CFD

Although the methods developed at Los Alamos were put to use in solving practical problems and picked up by researchers outside the Laboratory, considerable development took place that does appear to be only indirectly motivated by it. Those development include:

- Panel methods for aerodynamic computations (Hess and Smith, 1966)
- Specialized techniques for free surface potential and stokes flows (1976) and vortex methods
- Spectral methods for DNS of turbulent flows (late 70's, 80's)
- Monotonic advection schemes for compressible flows (late 70's, 80's)
- Steady state solutions (SIMPLE, aeronautical applications, etc)
- Commercial CFD

However, with outside interest in Multifluid simulations (improved VOF, level sets, front tracking) around 1990, the legacy became obvious

Computational Fluid Dynamics

Commercial Codes

Computational Fluid Dynamics
Commercial Codes

CHAM (Concentration Heat And Momentum) founded in 1974 by Prof. Brian Spalding was the first provider of general-purpose CFD software. The original PHOENICS appeared in 1981.

The first version of the FLUENT code was launched in October 1983 by Creare Inc. Fluent Inc. was established in 1988.

STAR-CD's roots go back to the foundation of Computational Dynamics in 1987 by Prof. David Gosman,

The original codes were relatively primitive, hard to use, and not very accurate.

Computational Fluid Dynamics
Commercial Codes

What to expect and when to use commercial package:

The current generation of CFD packages generally is capable of producing accurate solutions of simple flows. The codes are, however, designed to be able to handle very complex geometries and complex industrial problems. When used with care by a knowledgeable user CFD codes are an enormously valuable design tool.

Commercial CFD codes are rarely useful for state-of-the-art research due to accuracy limitations, the limited access that the user has to the solution methodology, and the limited opportunities to change the code if needed

Computational Fluid Dynamics
Commercial Codes

Major current players include

Ansys (Fluent and other codes)
http://www.fluent.com/
http://www.ansys.com/

adapco: (starCD)
http://www.cd-adapco.com/

Others
CHAM: http://www.cham.co.uk/
CFD2000: http://www.adaptive-research.com/

Computational Fluid Dynamics
http://www.nd.edu/~gtryggva/CFD-Course/

A Finite Difference Code for the Navier-Stokes Equations in Vorticity/ Streamfunction Form

Computational Fluid Dynamics
The vorticity/streamfunction equations:

$$-\frac{\partial}{\partial y}\left[\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)\right]$$

$$\frac{\partial}{\partial x}\left[\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)\right]$$

$$\frac{\partial \omega}{\partial t}+u\frac{\partial \omega}{\partial x}+v\frac{\partial \omega}{\partial y}=\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \omega}{\partial x^2}+\frac{\partial^2 \omega}{\partial y^2}\right)$$

where $\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}$

Computational Fluid Dynamics
Objectives

Defining the streamfunction by

$$u=\frac{\partial \psi}{\partial y};\quad v=-\frac{\partial \psi}{\partial x}$$

And substituting into the definition of the vorticity

$$\omega=\frac{\partial v}{\partial x}-\frac{\partial u}{\partial y}$$

gives

$$\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}=-\omega$$

Computational Fluid Dynamics
The vorticity/streamfunction equations:

The system to be solved is the Navier-Stokes equations in vorticity-stream function form are:

Advection/diffusion equation

$$\frac{\partial \omega}{\partial t}=-\frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x}+\frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}+\frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \omega}{\partial x^2}+\frac{\partial^2 \omega}{\partial y^2}\right)$$

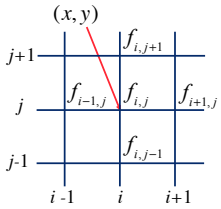
Elliptic equation

$$\frac{\partial^2 \psi}{\partial x^2}+\frac{\partial^2 \psi}{\partial y^2}=-\omega$$

where

$$u=\frac{\partial \psi}{\partial y};\quad v=-\frac{\partial \psi}{\partial x}$$

Notation: the location of variables on a structured grid



$$f_{i,j} = f(x,y)$$
$$f_{i+1,j} = f(x+h,y)$$
$$f_{i-1,j} = f(x-h,y)$$
$$f_{i,j+1} = f(x,y+h)$$
$$f_{i,j-1} = f(x,y-h)$$

Then we replace the equations at each grid point by a finite difference approximation

$$\left.\frac{\partial\omega}{\partial t}\right)_{i,j}^{n} = -\left.\frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x}\right)_{i,j}^{n} + \left.\frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y}\right)_{i,j}^{n} + \frac{1}{\mathrm{Re}}\left(\left.\frac{\partial^2\omega}{\partial x^2}\right)_{i,j}^{n} + \left.\frac{\partial^2\omega}{\partial y^2}\right)_{i,j}^{n}\right.$$

$$\left.\frac{\partial^2\psi}{\partial x^2}\right)_{i,j}^{n} + \left.\frac{\partial^2\psi}{\partial y^2}\right)_{i,j}^{n} = -\omega_{i,j}^{n}$$

Finite difference approximations

$$\frac{\partial f(x)}{\partial x} = \frac{f(x+h)-f(x-h)}{2h} - \frac{\partial^3 f(x)}{\partial x^3}\frac{h^2}{12} + \cdots$$

$$\frac{\partial^2 f(x)}{\partial x^2} = \frac{f(x+h)-2f(x)+f(x-h)}{h^2} - \frac{\partial^4 f(x)}{\partial x^4}\frac{h^2}{xx} + \cdots$$

$$\frac{\partial f(t)}{\partial t} = \frac{f(t+\Delta t)-f(t)}{\Delta t} - \frac{\partial^2 f(t)}{\partial t^2}\frac{\Delta t}{2} + \cdots$$

Laplacian

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} =$$

$$\frac{f_{i+1,j}^n - 2f_{i,j}^n + f_{i-1,j}^n}{h^2} + \frac{f_{i,j+1}^n - 2f_{i,j}^n + f_{i,j-1}^n}{h^2} =$$

$$\frac{f_{i+1,j}^n + f_{i-1,j}^n + f_{i,j+1}^n + f_{i,j-1}^n - 4f_{i,j}^n}{h^2}$$

$$\frac{\partial\omega}{\partial t} = -\frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} + \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2}\right)$$

The advection equation is:

$$\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^n}{\Delta t} =$$

$$-\left(\frac{\psi_{i,j+1}^n - \psi_{i,j-1}^n}{2h}\right)\left(\frac{\omega_{i+1,j}^n - \omega_{i-1,j}^n}{2h}\right) + \left(\frac{\psi_{i+1,j}^n - \psi_{i-1,j}^n}{2h}\right)\left(\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2h}\right)$$

$$+\frac{1}{\mathrm{Re}}\left(\frac{\omega_{i+1,j}^n + \omega_{i-1,j}^n + \omega_{i,j+1}^n + \omega_{i,j-1}^n - 4\omega_{i,j}^n}{h^2}\right)$$

The vorticity at the new time is given by:

$$\omega_{i,j}^{n+1} = \omega_{i,j}^n + \Delta t\left[-\left(\frac{\psi_{i,j+1}^n - \psi_{i,j-1}^n}{2h}\right)\left(\frac{\omega_{i+1,j}^n - \omega_{i-1,j}^n}{2h}\right)\right.$$

$$+\left(\frac{\psi_{i+1,j}^n - \psi_{i-1,j}^n}{2h}\right)\left(\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2h}\right)$$

$$\left.+\frac{1}{\mathrm{Re}}\left(\frac{\omega_{i+1,j}^n + \omega_{i-1,j}^n + \omega_{i,j+1}^n + \omega_{i,j-1}^n - 4\omega_{i,j}^n}{h^2}\right)\right]$$

Computational Fluid Dynamics
Finite Difference Approximations

The elliptic equation is:

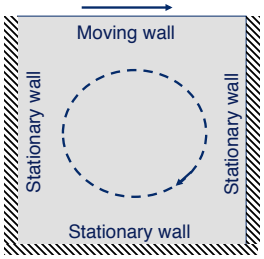
$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega$$

$$\frac{\psi^n_{i+1,j} + \psi^n_{i-1,j} + \psi^n_{i,j+1} + \psi^n_{i,j-1} - 4\psi^n_{i,j}}{h^2} = -\omega^n_{i,j}$$

Computational Fluid Dynamics
The Driven Cavity Problem

The Driven Cavity Problem

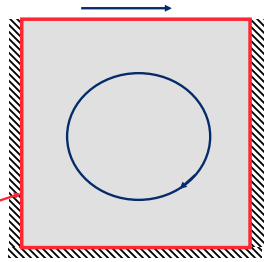
We will now use this approach to solve for the flow in a driven cavity. The driven cavity is a square domain with a moving wall at the top and stationary side and bottom walls. The simple geometry and the absence of in and outflow makes this a particularly simple and popular test problem.



Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

Since the boundaries meet, the constant must be the same on all boundaries

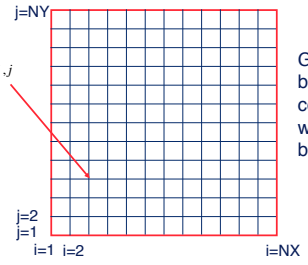
$\psi = \text{Constant}$



Computational Fluid Dynamics
Discretizing the Domain

To compute an approximate solution numerically, we start by laying down a discrete grid:

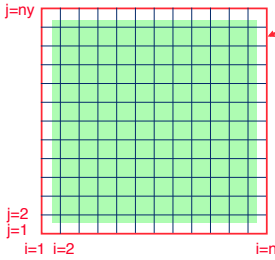
$\psi_{i,j}$ and $\omega_{i,j}$ stored at each grid point



Grid boundaries coincide with domain boundaries

Computational Fluid Dynamics
Finite Difference Approximations

These equations allow us to obtain the solution at interior points



$\psi_{i,j} = 0$ on the boundary

Need vorticity on the boundary!

Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

To update the vorticity in the interior of the domain, we need the vorticity at the wall. Once the streamfunction (and thus the velocity) everywhere has been found, this can be done.

At the bottom wall:

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = -\frac{\partial u}{\partial y}$$

Since the normal velocity is zero

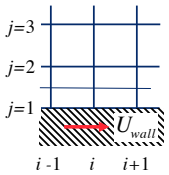
Where the velocity can be found from the streamfunction

$$u = \frac{\partial \psi}{\partial y}$$

Computational Fluid Dynamics
Discrete Boundary Condition

Consider the bottom wall (j=1):

Need to find $\omega_{wall} = \omega_{i,j=1}$



The velocity at j=3/2 can be found by centered differences:

$$u(i,\tfrac{3}{2}) = \frac{\partial \psi}{\partial y} \approx \frac{\psi(i,2)-\psi(i,1)}{h}$$

The vorticity is then found by one-sided differences

$$\omega(i,1) = -\frac{\partial u}{\partial y} \approx \frac{u(i,\tfrac{3}{2})-u(i,1)}{h/2}$$

$$= \frac{-2}{h}\left(\frac{\psi(i,2)-\psi(i,1)}{h} - U_{wall}\right)$$

Similar expressions can be found for the other walls

Computational Fluid Dynamics
Solving the elliptic equation

The elliptic equation:

$$\frac{\psi^n_{i+1,j} + \psi^n_{i-1,j} + \psi^n_{i,j+1} + \psi^n_{i,j-1} - 4\psi^n_{i,j}}{h^2} = -\omega^n_{i,j}$$

Rewrite as

$$\psi^{n+1}_{i,j} = 0.25\,(\psi^n_{i+1,j} + \psi^n_{i-1,j} + \psi^n_{i,j+1} + \psi^n_{i,j-1} + h^2\omega^n_{i,j})$$

Solve by SOR

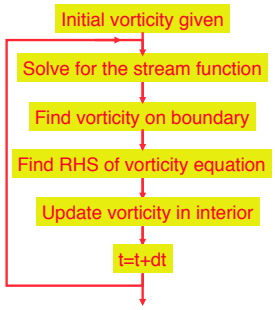
$$\psi^{\alpha+1}_{i,j} = \beta 0.25\,(\psi^{\alpha}_{i+1,j} + \psi^{\alpha+1}_{i-1,j} + \psi^{\alpha}_{i,j+1} + \psi^{\alpha+1}_{i,j-1} + h^2\omega^n_{i,j}) + (1-\beta)\psi^{\alpha}_{i,j}$$

Computational Fluid Dynamics
Solution Strategy

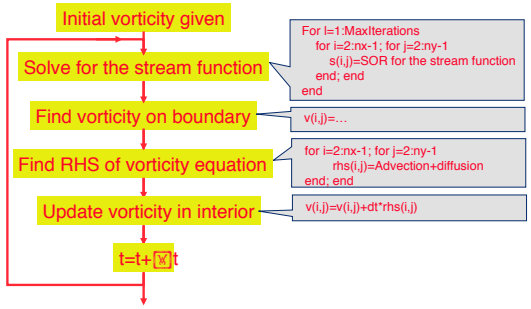
Limitations on the time step

$$\frac{\nu\Delta t}{h^2} \le \frac{1}{4}$$

$$\frac{(|u|+|v|)\Delta t}{\nu} \le 2$$



Computational Fluid Dynamics
Solution Strategy

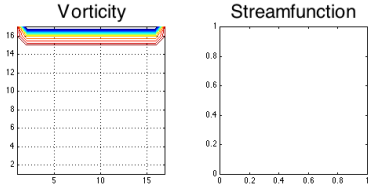


Computational Fluid Dynamics
The Code

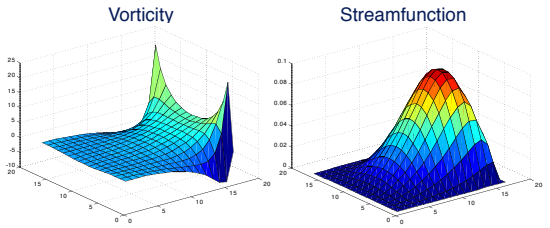
```
clf;nx=9; ny=9; MaxStep=60; Visc=0.1; dt=0.02;      % resolution & governing parameters
MaxIt=100; Beta=1.5; MaxErr=0.001;                  % parameters for SOR iteration
sf=zeros(nx,ny); vt=zeros(nx,ny); w=zeros(nx,ny); h=1.0/(nx-1); t=0.0;
for istep=1:MaxStep,                                % start the time integration
 for iter=1:MaxIt,                                  % solve for the streamfunction
   w=sf;                                            % by SOR iteration
   for i=2:nx-1; for j=2:ny-1
       sf(i,j)=0.25*Beta*(sf(i+1,j)+sf(i-1,j)...
         +sf(i,j+1)+sf(i,j-1)+h*h*vt(i,j))+(1.0-Beta)*sf(i,j);
   end; end;
   Err=0.0; for i=1:nx; for j=1:ny, Err=Err+abs(w(i,j)-sf(i,j)); end; end;
   if Err <= MaxErr, break, end                     % stop if iteration has converged
 end;
 vt(2:nx-1,1)=-2.0*sf(2:nx-1,2)/(h*h);              % vorticity on bottom wall
 vt(2:nx-1,ny)=-2.0*sf(2:nx-1,ny-1)/(h*h)-2.0/h;    % vorticity on top wall
 vt(1,2:ny-1)=-2.0*sf(2,2:ny-1)/(h*h);              % vorticity on right wall
 vt(nx,2:ny-1)=-2.0*sf(nx-1,2:ny-1)/(h*h);          % vorticity on left wall
 for i=2:nx-1; for j=2:ny-1                         % compute
     w(i,j)=-0.25*((sf(i,j+1)-sf(i,j-1))*(vt(i+1,j)-vt(i-1,j))...   % the RHS
       -(sf(i+1,j)-sf(i-1,j))*(vt(i,j+1)-vt(i,j-1)))/(h*h)...       % of the
       +Visc*(vt(i+1,j)+vt(i-1,j)+vt(i,j+1)+vt(i,j-1)-4.0*vt(i,j))/(h*h); % vorticity
 end; end;                                          % equation
 vt(2:nx-1,2:ny-1)=vt(2:nx-1,2:ny-1)+dt*w(2:nx-1,2:ny-1);   % update the vorticity
 t=t+dt                                             % print out t
 subplot(121), contour(rot90(fliplr(vt))), axis('square');  % plot vorticity
 subplot(122), contour(rot90(fliplr(sf))), axis('square');pause(0.01) % streamfunction
end;
```

Computational Fluid Dynamics
Results:

17 by 17
Dt=0.01
D=0.1



Computational Fluid Dynamics
Results:

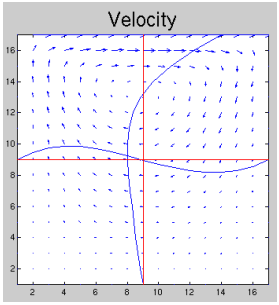


Computational Fluid Dynamics
Results:

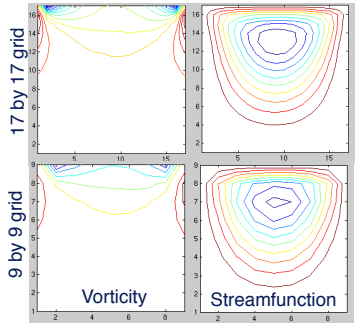
17 by 17
Dt=0.01
D=0.1

$$u_{i,j} = \frac{\partial \psi}{\partial y} \approx \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2h}$$

$$v_{i,j} = -\frac{\partial \psi}{\partial x} \approx -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}$$



Computational Fluid Dynamics
Results:



Computational Fluid Dynamics
IN/OUT FLOW

The driven cavity problem is particularly simple since there is no in or out flow. In the streamfunction-vorticity form of the Navier-Stokes equations it is, however, relatively easy to include specified in and outflow.

An inflow through the right horizontal wall, for example, can be imposed by increasing the streamfunction between each grid point by: $\Delta\psi = u\Delta y$

For incompressible flow the inflow must matched by outflow and if we specify the outflow velocity the the streamfunction must be decreased by the same amount elsewhere on the boundary