Computational Fluid Dynamics

Lecture 3
January 25, 2017

Grétar Tryggvason

Analysis of a Numerical Scheme An Example

Use the leap-frog method (centered differences) to integrate the diffusion equation

$$\frac{\partial f}{\partial t} = D\frac{\partial^2 f}{\partial x^2} \qquad D > 0$$

in time. Use the standard centered difference approximation for the second order spatial derivative.
(a) Write down the finite difference equation.
(b) Write down the modified equation
(c) Find the accuracy of the scheme
(d) Use the von Neuman's method to derive an equation for the amplification factor *g*. Hint: assume that the amplification is the same for each step: $g = \varepsilon^{n+1}/\varepsilon^n = \varepsilon^n/\varepsilon^{n-1}$

(a) Write down the finite difference equation

$$\frac{\partial f}{\partial t} = D\frac{\partial^2 f}{\partial x^2}$$

Approximate both terms by centered differences

$$\frac{f_j^{n+1} - f_j^{n-1}}{2\Delta t} = D\frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{h^2}$$

(b) Write down the modified equation

$$(1) \quad f_j^{n+1} = f_j^n + \frac{\partial f_j^n}{\partial t}\Delta t + \frac{\partial^2 f_j^n}{\partial t^2}\frac{\Delta t^2}{2} + \frac{\partial^3 f_j^n}{\partial t^3}\frac{\Delta t^3}{6} + \cdots$$

$$(2) \quad f_j^{n-1} = f_j^n - \frac{\partial f_j^n}{\partial t}\Delta t + \frac{\partial^2 f_j^n}{\partial t^2}\frac{\Delta t^2}{2} - \frac{\partial^3 f_j^n}{\partial t^3}\frac{\Delta t^3}{6} + \cdots$$

$$(3) \quad f_{j+1}^n = f_j^n + \frac{\partial f_j^n}{\partial x}h + \frac{\partial^2 f_j^n}{\partial x^2}\frac{h^2}{2} + \frac{\partial^3 f_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$$

$$(4) \quad f_{j-1}^n = f_j^n - \frac{\partial f_j^n}{\partial x}h + \frac{\partial^2 f_j^n}{\partial x^2}\frac{h^2}{2} - \frac{\partial^3 f_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$$

Substitute into

$$\frac{f_j^{n+1} - f_j^{n-1}}{2\Delta t} = D\frac{f_{j+1}^n - 2f_j^n + f_{j-1}^n}{h^2}$$

Substitute

$$\frac{(1)-(2)}{2\Delta t} = D\frac{(3) - 2f_j^n + (4)}{h^2}$$

yielding

$$\frac{\partial f}{\partial t} + \frac{\partial^3 f}{\partial t^3}\frac{\Delta t^2}{6} = D\frac{\partial^2 f}{\partial x^2} + D\frac{\partial^4 f}{\partial x^4}\frac{h^2}{12} + \ldots$$

Rearrange

$$\frac{\partial f}{\partial t} - D\frac{\partial^2 f}{\partial x^2} = D\frac{\partial^4 f}{\partial x^4}\frac{h^2}{12} - \frac{\partial^3 f}{\partial t^3}\frac{\Delta t^2}{6} + \ldots$$

Modified Equation

(c) Find the accuracy of the scheme

$$O\left(\Delta t^2, h^2\right)$$

Computational Fluid Dynamics
Numerical Analysis—Example

(d) Use the von Neuman's method to derive an equation for the amplification factor *g*.

Substitute

$$\varepsilon_j^n = \varepsilon^n e^{ikx_j}$$

into

$$\frac{\varepsilon_j^{n+1} - \varepsilon_j^{n-1}}{2\Delta t} = D\frac{\varepsilon_{j+1}^n - 2\varepsilon_j^n + \varepsilon_{j-1}^n}{h^2}$$

giving

$$\frac{\varepsilon^{n+1}e^{ikx_j} - \varepsilon^{n-1}e^{ikx_j}}{2\Delta t} = D\frac{\varepsilon^n}{h^2}(e^{ikh}e^{ikx_j} - 2e^{ikx_j} + e^{-ikh}e^{ikx_j})$$

Computational Fluid Dynamics
Numerical Analysis—Example

$$\frac{\varepsilon^{n+1}e^{ikx_j} - \varepsilon^{n-1}e^{ikx_j}}{2\Delta t} = D\frac{\varepsilon^n}{h^2}(e^{ikh}e^{ikx_j} - 2e^{ikx_j} + e^{-ikh}e^{ikx_j})$$

Cancel the common factor

$$\frac{\varepsilon^{n+1} - \varepsilon^{n-1}}{2\Delta t} = D\frac{\varepsilon^n}{h^2}(e^{ikh} - 2 + e^{-ikh})$$

Rearrange

$$\frac{\varepsilon^{n+1}}{\varepsilon^n} - \frac{\varepsilon^{n-1}}{\varepsilon^n} = \frac{2\Delta t D}{h^2}(e^{ikh} - 2 + e^{-ikh})$$

Using that $g = \varepsilon^{n+1}/\varepsilon^n = \varepsilon^n/\varepsilon^{n-1}$

Gives: $g - \frac{1}{g} = -\frac{2\Delta t D}{h^2} 4\sin^2\frac{kh}{2}$

$e^{ikh} + e^{-ikh} = 2\cos kh$

$2\sin^2\theta = 1 - \cos 2\theta$

Computational Fluid Dynamics
Numerical Analysis—Example

$$g - \frac{1}{g} = -\frac{2\Delta t D}{h^2} 4\sin^2\frac{kh}{2}$$

$x^2 + bg + c = 0$

$x = -\frac{b}{2} \pm \sqrt{\left(\frac{b}{2}\right)^2 - c}$

Putting

$$B = 8\frac{\Delta t D}{h^2}\sin^2\frac{kh}{2}$$

gives

$$g^2 - 1 = -Bg \longrightarrow g^2 + Bg - 1 = 0$$

Solution

$$g = -\frac{B}{2} \pm \sqrt{\left(\frac{B}{2}\right)^2 + 1}$$

Which is always > 1 $\longrightarrow$ Unconditionally unstable

Computational Fluid Dynamics

Analysis of a Numerical Scheme Another Example

Computational Fluid Dynamics

The following finite difference approximation is given

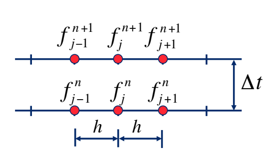
$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -\frac{1}{2}\left(\frac{U}{2h}\left(f_{j+1}^n - f_{j-1}^n\right) + \frac{U}{2h}\left(f_{j+1}^{n+1} - f_{j-1}^{n+1}\right)\right)$$

(a) Write down the modified equation
(b) What equation is being approximated?
(c) Determine the accuracy of the scheme
(d) Use the von Neuman's method to derive an equation for the stability conditions

Computational Fluid Dynamics

(a) Write down the modified equation

$$\frac{f_j^{n+1} - f_j^n}{\Delta t} = -\frac{1}{2}\left(\frac{U}{2h}\left(f_{j+1}^n - f_{j-1}^n\right) + \frac{U}{2h}\left(f_{j+1}^{n+1} - f_{j-1}^{n+1}\right)\right)$$



$$(1)\quad f_j^{n+1} = f_j^n + \frac{\partial f_j^n}{\partial t}\Delta t + \frac{\partial^2 f_j^n}{\partial t^2}\frac{\Delta t^2}{2} + \frac{\partial^3 f_j^n}{\partial t^3}\frac{\Delta t^3}{6} + \cdots$$

$$(2)\quad f_{j+1}^n = f_j^n + \frac{\partial f_j^n}{\partial x}h + \frac{\partial^2 f_j^n}{\partial x^2}\frac{h^2}{2} + \frac{\partial^3 f_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$$

$$(3)\quad f_{j-1}^n = f_j^n - \frac{\partial f_j^n}{\partial x}h + \frac{\partial^2 f_j^n}{\partial x^2}\frac{h^2}{2} - \frac{\partial^3 f_j^n}{\partial x^3}\frac{h^3}{6} + \cdots$$

$$(4)\quad f_{j+1}^{n+1} = f_j^{n+1} + \frac{\partial f_j^{n+1}}{\partial x}h + \frac{\partial^2 f_j^{n+1}}{\partial x^2}\frac{h^2}{2} + \frac{\partial^3 f_j^{n+1}}{\partial x^3}\frac{h^3}{6} + \cdots$$

$$(5)\quad f_{j-1}^{n+1} = f_j^{n+1} - \frac{\partial f_j^{n+1}}{\partial x}h + \frac{\partial^2 f_j^{n+1}}{\partial x^2}\frac{h^2}{2} - \frac{\partial^3 f_j^{n+1}}{\partial x^3}\frac{h^3}{6} + \cdots$$

Substitute

$$\frac{(1) - f_j^n}{\Delta t} = -\frac{1}{2}\left(\frac{U}{2h}\big((2)-(3)\big) + \frac{U}{2h}\big((4)-(5)\big)\right)$$

yielding

$$\left(\frac{\partial f}{\partial t} + \frac{\partial^2 f}{\partial t^2}\frac{\Delta t}{2} + \cdots\right)_j^n = -\frac{U}{2}\left(\left(\frac{\partial f}{\partial x} + \frac{\partial^3 f}{\partial x^3}\frac{h^2}{6} + \cdots\right)_j^n + \left(\frac{\partial f}{\partial x} + \frac{\partial^3 f}{\partial x^3}\frac{h^2}{6} + \cdots\right)_j^{n+1}\right)$$

Using (1) again we can write

$$\frac{\partial f_j^{n+1}}{\partial x} = \frac{\partial f_j^n}{\partial x} + \frac{\partial^2 f_j^n}{\partial t \partial x}\Delta t + \frac{\partial^3 f_j^n}{\partial t^2 \partial x}\frac{\Delta t^2}{2} + \frac{\partial^4 f_j^n}{\partial t^3 \partial x}\frac{\Delta t^3}{6} + \cdots$$

$$\frac{\partial^3 f_j^{n+1}}{\partial x^3} = \frac{\partial^3 f_j^n}{\partial x^3} + \frac{\partial^4 f_j^n}{\partial t \partial x^3}\Delta t + \frac{\partial^5 f_j^n}{\partial t^2 \partial x^3}\frac{\Delta t^2}{2} + \frac{\partial^6 f_j^n}{\partial t^3 \partial x^3}\frac{\Delta t^3}{6} + \cdots$$

Substitute again:

$$\left(\frac{\partial f}{\partial t} + \frac{\partial^2 f}{\partial t^2}\frac{\Delta t}{2} + \cdots\right)_j^n = -\frac{U}{2}\left(\left(\frac{\partial f}{\partial x} + \frac{\partial^3 f}{\partial x^3}\frac{h^2}{6} + \cdots\right)_j^n + \right.$$

$$\left.\left(\frac{\partial f_j^n}{\partial x} + \frac{\partial^2 f_j^n}{\partial t \partial x}\Delta t + \frac{\partial^3 f_j^n}{\partial t^2 \partial x}\frac{\Delta t^2}{2} + \cdots + \left(\frac{\partial^3 f_j^n}{\partial x^3} + \cdots\right)\frac{h^2}{6} + \cdots\right)_j^{n+1}\right)$$

Collecting the terms

$$\frac{\partial f}{\partial t} + \frac{\partial^2 f}{\partial t^2}\frac{\Delta t}{2} + \cdots = -\frac{U}{2}\left(\frac{\partial f}{\partial x} + \underline{\frac{\partial^3 f}{\partial x^3}}\frac{h^2}{6} + \frac{\partial f}{\partial x} + \frac{\partial^2 f}{\partial t \partial x}\Delta t + \underline{\frac{\partial^3 f}{\partial t^2 \partial x}}\frac{\Delta t^2}{2} + \underline{\frac{\partial^3 f_j^n}{\partial x^3}}\frac{h^2}{6} + \cdots\right)$$

$$\frac{\partial f}{\partial t} = -U\frac{\partial f}{\partial x} \quad \Rightarrow \quad -\frac{U}{2}\frac{\partial^2 f}{\partial t \partial x} = -\frac{U}{2}\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial t}\right) = \frac{1}{2}\frac{\partial^2 f}{\partial t^2}$$

Rearrange

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = -U\frac{\partial^3 f}{\partial x^3}\frac{h^2}{6} - U\frac{\partial^3 f}{\partial t^2 \partial x}\frac{\Delta t^2}{4} + \cdots$$

(b) What equation is being approximated

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = 0$$

(c) Find the accuracy of the scheme

$$O\left(\Delta t^2, h^2\right)$$

(d) Use the von Neuman's method to derive an equation for the amplification factor *g*.

Substitute

$$\varepsilon_j^n = \varepsilon^n e^{ikx_j}$$

into

$$\frac{\varepsilon_j^{n+1} - \varepsilon_j^n}{\Delta t} = -\frac{1}{2}\left(\frac{U}{2h}\left(\varepsilon_{j+1}^n - \varepsilon_{j-1}^n\right) + \frac{U}{2h}\left(\varepsilon_{j+1}^{n+1} - \varepsilon_{j-1}^{n+1}\right)\right)$$

giving

$$\frac{\varepsilon^{n+1}e^{ikx_j} - \varepsilon^n e^{ikx_j}}{\Delta t} = \frac{U}{4h}\left(\varepsilon^n\left(e^{ikh}e^{ikx_j} - e^{-ikh}e^{ikx_j}\right) + \varepsilon^{n+1}\left(e^{ikh}e^{ikx_j} - e^{-ikh}e^{ikx_j}\right)\right)$$

$$\frac{\varepsilon^{n+1}e^{ikx_j} - \varepsilon^n e^{ikx_j}}{\Delta t} = \frac{U}{4h}\left(\varepsilon^n\left(e^{ikh}e^{ikx_j} - e^{-ikh}e^{ikx_j}\right) + \varepsilon^{n+1}\left(e^{ikh}e^{ikx_j} - e^{-ikh}e^{ikx_j}\right)\right)$$

Cancel the common factor

$$\frac{\varepsilon^{n+1} - \varepsilon^n}{\Delta t} = \frac{U}{4h}\left(\varepsilon^n\left(e^{ikh} - e^{-ikh}\right) + \varepsilon^{n+1}\left(e^{ikh} - e^{-ikh}\right)\right)$$

Rearrange

$$\frac{\varepsilon^{n+1}}{\varepsilon^n} - 1 = \frac{\Delta t U}{4h}\left(\left(e^{ikh} - e^{-ikh}\right) + \frac{\varepsilon^{n+1}}{\varepsilon^n}\left(e^{ikh} - e^{-ikh}\right)\right)$$

writing $g = \varepsilon^{n+1}/\varepsilon^n$

Gives: $g - 1 = (1+g)\frac{\Delta t U}{4h}(2i\sin kh)$

$e^{ikh} - e^{-ikh} = 2i\sin kh$

Computational Fluid Dynamics

$$g-1=(1+g)\frac{\Delta t U}{4h}(2i\sin kh)$$
$$g\big(1-A(2i\sin kh)\big)=1+A(2i\sin kh)$$
$$g=\frac{1+A(2i\sin kh)}{1-A(2i\sin kh)}$$
$$g=\frac{1-4A^2\sin^2 kh+iA2\sin kh}{1+4A^2\sin^2 kh}$$
$$g=\frac{1-B^2+iB}{1+B^2}$$
$$|g|^2=\frac{(1-B^2)^2+B^2}{(1+B^2)^2}$$
$$1-B^2+B^4<1+2B^2+B^4$$

$$z=x+iy$$
$$|z|^2=(x+iy)(x-iy)=x^2+y^2$$

Unconditionally Stable

Computational Fluid Dynamics

Notes:
You will do a few other schemes as homework problems.

Generally we assume that results for the linear equations hold for the nonlinear one as well.

The algebraic equation for the amplification factor can often be fairly complicated. It is, however, routinely solved.

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
Objectives

Develop an understanding of the steps involved in solving the Navier-Stokes equations using a numerical method

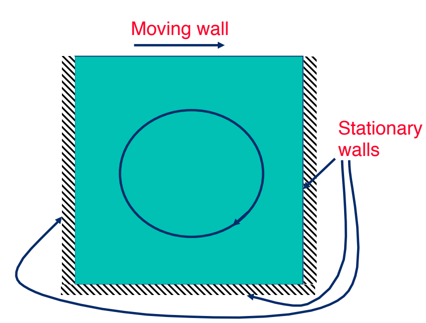
Write a simple code to solve the “driven cavity” problem using the Navier-Stokes equations in vorticity form

Short discussion about why looking at the vorticity is sometimes helpful

Computational Fluid Dynamics
Outline

- The Driven Cavity Problem
- The Navier-Stokes Equations in Vorticity/Streamfunction form
- Boundary Conditions
- The Grid
- Finite Difference Approximation of the Vorticity/Streamfunction equations
- Finite Difference Approximation of the Boundary Conditions
- Iterative Solution of the Elliptic Equation
- The Code
- Results
- Convergence Under Grid Refinement

Computational Fluid Dynamics
The Driven Cavity Problem



Computational Fluid Dynamics
The vorticity/streamfunction equations:

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

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

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

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

Computational Fluid Dynamics
The vorticity/streamfunction equations:

Solve the incompressibility conditions

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

by introducing the stream function

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

Substituting:

$$\cancel{\frac{\partial}{\partial x}\frac{\partial \psi}{\partial y}}-\cancel{\frac{\partial}{\partial y}\frac{\partial \psi}{\partial x}}=0$$

Computational Fluid Dynamics
The vorticity/streamfunction equations:

Substituting

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

into the definition of the vorticity

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

yields

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

Computational Fluid Dynamics
The vorticity/streamfunction equations:

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

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

Recall the advection-diffusion equation

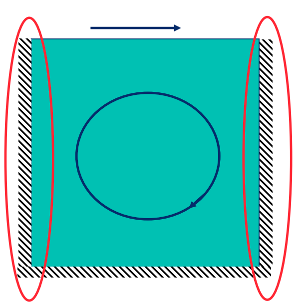
$$\frac{\partial f}{\partial t}+u\frac{\partial f}{\partial x}+v\frac{\partial f}{\partial y}=D\left(\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}\right)$$

Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

At the right and the left boundary:

$$u=0\Rightarrow\frac{\partial \psi}{\partial y}=0$$

$$\Rightarrow \psi=\text{Constant}$$

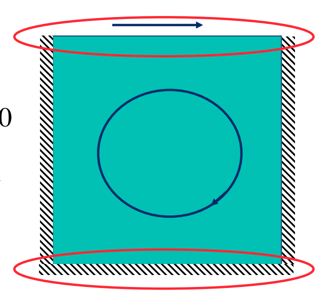


Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

At the top and the bottom boundary:

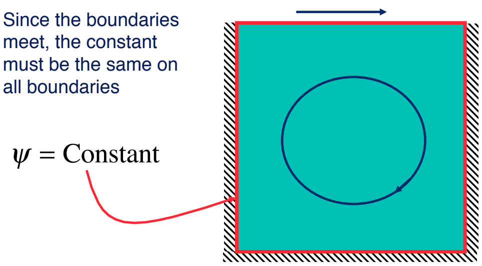
$$v=0\Rightarrow-\frac{\partial \psi}{\partial x}=0$$

$$\Rightarrow \psi=\text{Constant}$$



Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

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



Computational Fluid Dynamics
Boundary Conditions for the Vorticity

The normal velocity is zero since the streamfunction is a constant on the wall, but the zero tangential velocity must be enforced:

At the right and left boundary:

$$v = 0 \Rightarrow -\frac{\partial \psi}{\partial x} = 0$$

At the bottom boundary:

$$u = 0 \Rightarrow \frac{\partial \psi}{\partial y} = 0$$

At the top boundary:

$$u = U_{wall} \Rightarrow \frac{\partial \psi}{\partial y} = U_{wall}$$

Computational Fluid Dynamics
Boundary Conditions for the Vorticity

The wall vorticity must be found from the streamfunction. The stream function is constant on the walls.

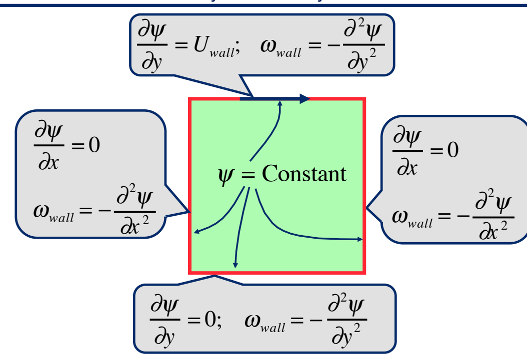
At the right and the left boundary:

$$\frac{\partial^2 \psi}{\partial x^2} + \cancel{\frac{\partial^2 \psi}{\partial y^2}} = -\omega \quad \Rightarrow \quad \omega_{wall} = -\frac{\partial^2 \psi}{\partial x^2}$$

Similarly, at the top and the bottom boundary:

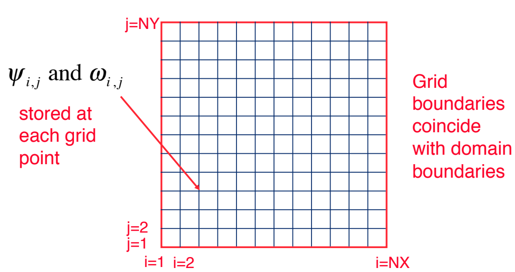
$$\cancel{\frac{\partial^2 \psi}{\partial x^2}} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad \Rightarrow \quad \omega_{wall} = -\frac{\partial^2 \psi}{\partial y^2}$$

Computational Fluid Dynamics
Summary of Boundary Conditions



Computational Fluid Dynamics
Discretizing the Domain

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



Computational Fluid Dynamics
Finite Difference Approximations

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

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

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

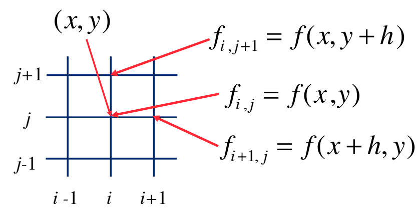
Finite difference approximations

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

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

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

Use the notation developed earlier:



Laplacian

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

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

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

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

The advection equation is:

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

$$-\left(\frac{\psi^n_{i,j+1} - \psi^n_{i,j-1}}{2h}\right)\left(\frac{\omega^n_{i+1,j} - \omega^n_{i-1,j}}{2h}\right) + \left(\frac{\psi^n_{i+1,j} - \psi^n_{i-1,j}}{2h}\right)\left(\frac{\omega^n_{i,j+1} - \omega^n_{i,j-1}}{2h}\right)$$

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

The vorticity at the new time is given by:

$$\omega^{n+1}_{i,j} = \omega^n_{i,j} + \Delta t\left[-\left(\frac{\psi^n_{i,j+1} - \psi^n_{i,j-1}}{2h}\right)\left(\frac{\omega^n_{i+1,j} - \omega^n_{i-1,j}}{2h}\right) + \left(\frac{\psi^n_{i+1,j} - \psi^n_{i-1,j}}{2h}\right)\left(\frac{\omega^n_{i,j+1} - \omega^n_{i,j-1}}{2h}\right) + \frac{1}{\mathrm{Re}}\left(\frac{\omega^n_{i+1,j} + \omega^n_{i-1,j} + \omega^n_{i,j+1} + \omega^n_{i,j-1} - 4\omega^n_{i,j}}{h^2}\right)\right]$$

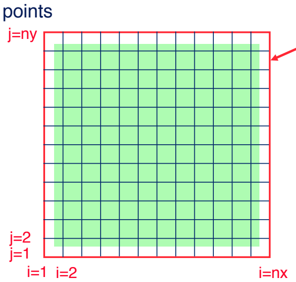
The elliptic equation is:

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

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

Computational Fluid Dynamics
Finite Difference Approximations

These equations allow us to obtain the solution at interior points



$\psi_{i,j} = 0$

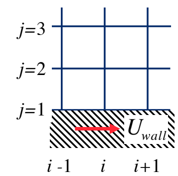
on the boundary

Need vorticity on the boundary!

Computational Fluid Dynamics
Discrete Boundary Condition

Consider the bottom wall ($j=1$):

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

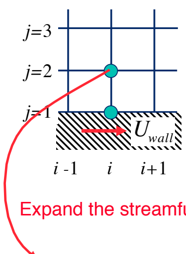


given:

$$\psi = \text{Constant}$$

$$\frac{\partial \psi}{\partial y} = U_{wall}; \quad \omega_{wall} = -\frac{\partial^2 \psi}{\partial y^2}$$

Computational Fluid Dynamics
Discrete Boundary Condition



given:

$$\psi = \text{Constant}$$

$$\frac{\partial \psi}{\partial y} = U_{wall}; \quad \omega_{wall} = -\frac{\partial^2 \psi}{\partial y^2}$$

Expand the streamfunction

$$\psi_{i,j=2} = \psi_{i,j=1} + \frac{\partial \psi_{i,j=1}}{\partial y} h + \frac{\partial^2 \psi_{i,j=1}}{\partial y^2} \frac{h^2}{2} + O(h^3)$$

Computational Fluid Dynamics
Discrete Boundary Condition

$$\psi_{i,j=2} = \psi_{i,j=1} + \frac{\partial \psi_{i,j=1}}{\partial y} h + \frac{\partial^2 \psi_{i,j=1}}{\partial y^2} \frac{h^2}{2} + O(h^3)$$

Using: $\omega_{wall} = -\frac{\partial^2 \psi_{i,j=1}}{\partial y^2}; \quad U_{wall} = \frac{\partial \psi_{i,j=1}}{\partial y}$

this becomes:

$$\psi_{i,j=2} = \psi_{i,j=1} + U_{wall} h - \omega_{wall} \frac{h^2}{2} + O(h^3)$$

Solving for the wall vorticity:

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

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
Time Step

Limitations on the time step

$$\frac{\nu \Delta t}{h^2} \le \frac{1}{4} \qquad \frac{(|u| + |v|)\Delta t}{\nu} \le 2$$

Computational Fluid Dynamics
Solution Strategy

Initial vorticity given

Solve for the stream function

Find vorticity on boundary

Find RHS of vorticity equation

Update vorticity in interior

t=t+Δt

Computational Fluid Dynamics
Solution Strategy

Initial vorticity given

Solve for the stream function

```
For I=1:MaxIterations
   for i=2:nx-1; for j=2:ny-1
      s(i,j)=SOR for the stream function
   end; end
end
```

Find vorticity on boundary

```
v(i,j)=…
```

Find RHS of vorticity equation

```
for i=2:nx-1; for j=2:ny-1
      rhs(i,j)=Advection+diffusion
end; end
```

Update vorticity in interior

```
v(i,j)=v(i,j)+dt*rhs(i,j)
```

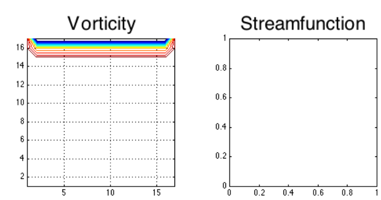
t=t+Δt

Computational Fluid Dynamics
The Code

```
% Driven Cavity by Vorticity-Stream Function Method
Nx=17; Ny=17; MaxStep=200; Visc=0.1; dt=0.005; time=0.0; h=1.0/(Nx-1);
MaxIt=100; Beta=1.5; MaxErr=0.001;  % parameters for SOR
sf=zeros(Nx,Ny); vt=zeros(Nx,Ny); vto=zeros(Nx,Ny); x=zeros(Nx,Ny); y=zeros(Nx,Ny);
for i=1:Nx,for j=1:Ny,x(i,j)=h*(i-1);y(i,j)=h*(j-1);end,end;
for istep=1:MaxStep,
  for iter=1:MaxIt,  % Time loop
    vto=sf;
    for i=2:Nx-1; for j=2:Ny-1   % solve for the stream function by SOR iteration
      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(vto(i,j)-sf(i,j)); end; end; % check error
    if Err <= MaxErr, break, end       % stop if converged
  end;
  vt(2:Nx-1,1)=-2.0*sf(2:Nx-1,2)/(h*h);              % vorticity on bdrys
  vt(2:Nx-1,Ny)=-2.0*sf(2:Nx-1,Ny-1)/(h*h)-2.0/h;  % top wall
  vt(1,2:Ny-1)=-2.0*sf(2,2:Ny-1)/(h*h);              % right wall
  vt(Nx,2:Ny-1)=-2.0*sf(Nx-1,2:Ny-1)/(h*h);          % left wall
  vto=vt;
  for i=2:Nx-1; for j=2:Ny-1
    vt(i,j)=vt(i,j)+dt*(-0.25*((sf(i,j+1)-sf(i,j-1))*...
      (vto(i+1,j)-vto(i-1,j))-(sf(i+1,j)-sf(i-1,j))*(vto(i,j+1)-vto(i,j-1)))/(h*h)+...
      Visc*(vto(i+1,j)+vto(i-1,j)+vto(i,j+1)+vto(i,j-1)-4.0*vto(i,j))/(h^2) );
  end; end;
  time=time+dt
  subplot(121), contour(x,y,vt,40), axis('square');
  subplot(122), contour(x,y,sf), axis('square'); pause(0.01)
end;
```

Computational Fluid Dynamics
Results:

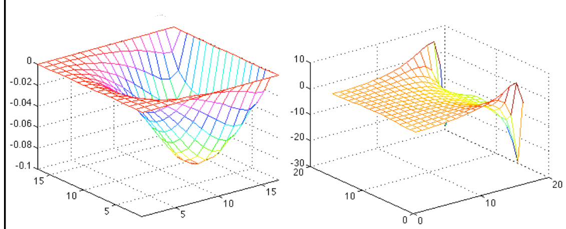
17 by 17
Dt=0.01
D=0.1



Computational Fluid Dynamics
Results:

Streamfunction

Vorticity

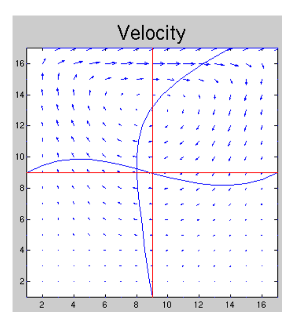


Computational Fluid Dynamics
Results:

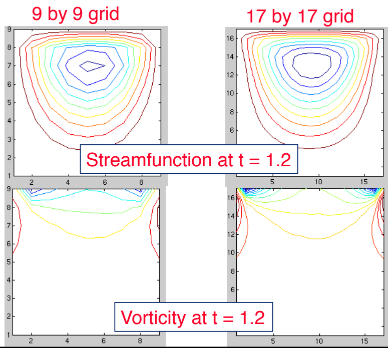
17 by 17
Dt=0.01
D=0.1

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

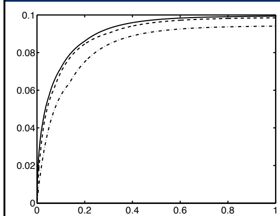
$$v_{i,j} = -\frac{\partial \psi}{\partial x} \approx -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}$$



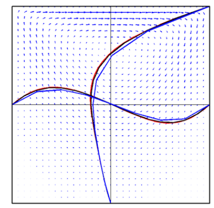
Computational Fluid Dynamics
Results:



Computational Fluid Dynamics



The total volume flow to the right in the driven cavity versus time, computed as the difference between the value of the stream function on walls and its minimum value, for three resolutions.



The velocity in the driven cavity, found by differentiating the stream function, along with the horizontal velocity on a vertical axis through the center of the domain and the vertical velocity on a horizontal axis through the center of the domain, for three resolutions.

Computational Fluid Dynamics

Periodic Domain in the x-direction

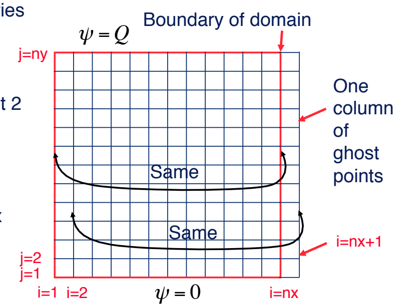
Computational Fluid Dynamics
Finite Difference Approximations

Periodic Boundaries

Set
point nx+1 = point 2

Update i=2;nx

Set
point 1 = Point nx



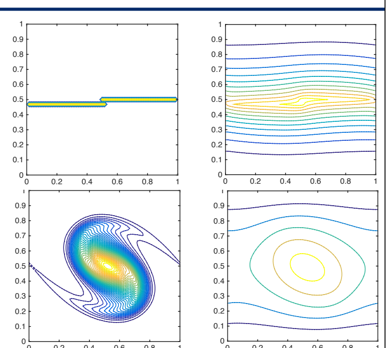
Computational Fluid Dynamics

Simulations of the rollup of a periodic vortex sheet

nx = 65; ny = 65;
MaxStep = 2000;
Visc = 0.001;
dt = 0.001;

sf(i,1)=sf(i,ny)=0;

% Initial vorticity distribution
vt(2:34,(ny-1)/2-1)=1.0/h;
vt(33:nx-1,(ny-1)/2+1)=1.0/h;



Computational Fluid Dynamics

Why is vorticity important?

Computational Fluid Dynamics
Vorticity

Helmholtz decomposition:
Any vector field can be written as a sum of

$$\boldsymbol{u} = \nabla \phi + \nabla \times \Psi$$

Take divergence

$$\nabla \cdot \boldsymbol{u} = \nabla \cdot \nabla \phi = \nabla^2 \phi = 0$$

Take the curl

$$\nabla \times \boldsymbol{u} = \nabla \times (\nabla \times \Psi) = \omega$$

By a Gauge transform this can be written as

$$\nabla^2 \Psi = -\omega$$

Computational Fluid Dynamics
Vorticity

For incompressible flow with constant density and viscosity, taking the curl of the momentum equation yields:

$$\frac{\partial \omega}{\partial t} + \boldsymbol{u} \nabla \cdot \omega = (\omega \cdot \nabla)\boldsymbol{u} + \nu \nabla^2 \omega$$

or:

$$\frac{D\omega}{Dt} = (\omega \cdot \nabla)\boldsymbol{u} + \nu \nabla^2 \omega$$

Helmholtz's theorem:
Inviscid Irrotational flow remains irrotational

Computational Fluid Dynamics
Vorticity

In two-dimensions: $\Psi = (0,0,\psi)$ $\omega = (0,0,\omega)$

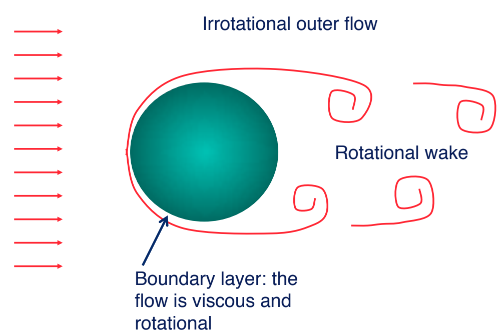
$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

or:

$$\frac{D\omega}{Dt} = \nu \nabla^2 \omega \qquad \nabla^2 \psi = -\omega$$

Zero viscosity: $\frac{D\omega}{Dt} = 0$ The vorticity of a fluid particle does not change!

Computational Fluid Dynamics
Flow over a body



Computational Fluid Dynamics

Advection and diffusion— Boundary layers

Computational Fluid Dynamics
Boundary Layers

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = D\frac{\partial^2 f}{\partial x^2}$$

Consider the steady state balance of advection and diffusion

$$f=0 \quad x=0 \qquad U \longrightarrow \qquad f=1 \quad x=L$$

Governed by: $U\frac{\partial f}{\partial x}=D\frac{\partial^2 f}{\partial x^2}$

Solve this equation analytically

$$\frac{df}{dx}=\frac{D}{U}\frac{d^2 f}{dx^2} \longrightarrow \frac{d}{dx}\left(f-\frac{D}{U}\frac{df}{dx}\right)=0$$

Integrate: $f-\frac{D}{U}\frac{df}{dx}=C_1$

Rearrange

$$f-C_1=\frac{D}{U}\frac{df}{dx} \longrightarrow \frac{1}{(f-C_1)}\frac{df}{dx}=\frac{U}{D}$$

or

$$\frac{df}{(f-C_1)}=\frac{U}{D}dx$$

Integrate

$$\int\frac{df}{(f-C_1)}=\int\frac{U}{D}dx \longrightarrow \ln(f-C_1)=\frac{U}{D}x+C_2$$

$$\longrightarrow f=\exp(Ux/D)\times\exp(C_2)+C_1$$

$$f=\exp(Ux/D)\times\exp(C_2)+C_1$$

Boundary conditions

At $x=0$: $\quad f=0 \longrightarrow 0=\exp(C_2)+C_1 \Rightarrow C_1=-\exp(C_2)$

At $x=L$: $\quad f=1 \longrightarrow 1=\exp(UL/D)\times\exp(C_2)+C_1$

$$\Rightarrow 1=\exp(UL/D)\times\exp(C_2)-\exp(C_2)$$

$$\Rightarrow 1=\exp(C_2)[\exp(U/D)-1]$$

$$\Rightarrow \exp(C_2)=\frac{1}{\exp(UL/D)-1}$$

$$f=\exp(Ux/D)\times\exp(C_2)+C_1$$

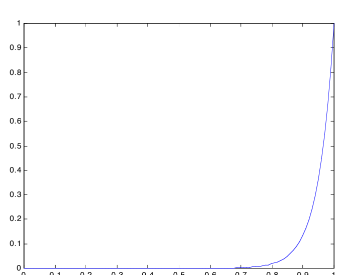
$$\underbrace{C_1=-\exp(C_2) \qquad \exp(C_2)=\frac{1}{\exp(UL/D)-1}}$$

$$f=\frac{\exp(Ux/D)-1}{\exp(UL/D)-1}$$

$$\longrightarrow f=\frac{\exp(Rx/L)-1}{\exp(R)-1} \qquad R=\frac{UL}{D}$$


```
r=20;for i=1:100,x(i)=(i-1)/99;end;
for i=1:100,f(i)=(exp(r*x(i))-1)/(exp(r)-1);end;
plot(x,f)
```

R=1
R=5
R=10
R=20

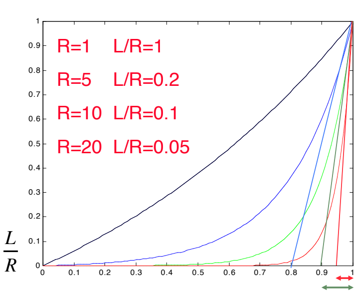


Scaling:

$$\frac{df}{dx}=\frac{D}{U}\frac{d^2 f}{dx^2}$$

Estimate the thickness of the "boundary Layer"

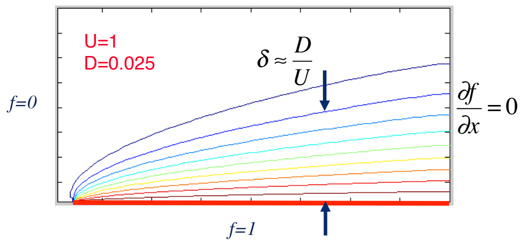
$$\frac{1}{\delta}=\frac{D}{U}\frac{1}{\delta^2} \Rightarrow \delta\approx\frac{DL}{UL}=\frac{L}{R}$$



Computational Fluid Dynamics
Boundary Layers

2D

Solution of: $U\frac{\partial f}{\partial x} = D\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)$



Computational Fluid Dynamics
Objectives

Develop an understanding of the steps involved in solving the Navier-Stokes equations using a numerical method

Write a simple code to solve the "driven cavity" problem using the Navier-Stokes equations in vorticity form

Short discussion about why looking at the vorticity is sometimes helpful

Computational Fluid Dynamics

Project I

Computational Fluid Dynamics

Time evolution of a one-dimensional equation

Write a program to compute the unsteady behavior of the following equation

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

in a periodic domain of length 1 with ν = 0.01. This is usually called the nonlinear advection-diffusion equation or the Burgers equation.

Take the initial conditions to be $f(x,t=0) = \sin(2\pi x) + 1.0$ and approximate the equation using a forward in time approximation for the time derivative, and centered approximations for the first and second derivative. Follow the evolution up to time 1.0 using at least three different grid resolutions, the finest of which should use at least 200 grid points. You can start from the program used in class for the linear advection- diffusion equation.