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

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

Grétar Tryggvason
Spring 2011

Computational Fluid Dynamics
Objectives

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

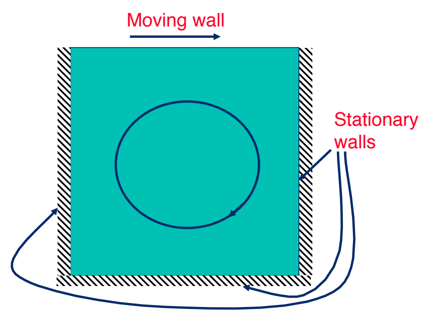
Write a simple code to solve the "driven cavity" problem using the Navier-Stokes equations in vorticity form

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

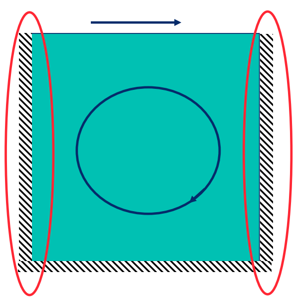
$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} + v\frac{\partial f}{\partial y} = D\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)$$

Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

At the right and the left boundary:

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

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

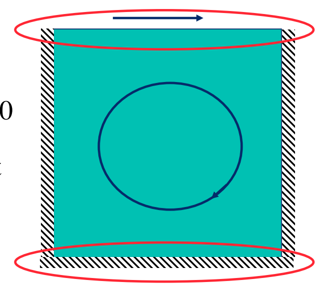


Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

At the top and the bottom boundary:

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

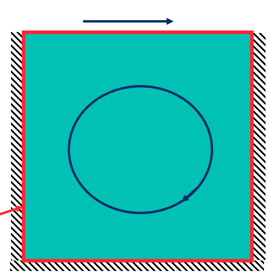
$$\Rightarrow \psi = \text{Constant}$$



Computational Fluid Dynamics
Boundary Conditions for the Streamfunction

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

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



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.

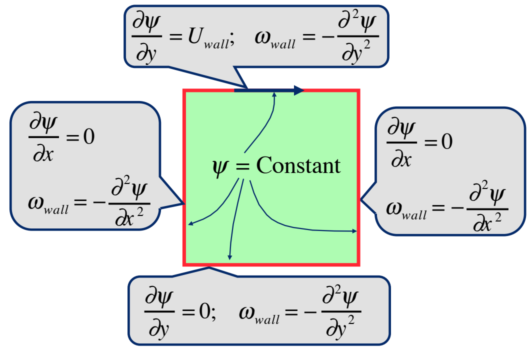
At the right and the left boundary:

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

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

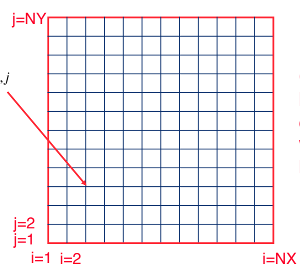


Computational Fluid Dynamics
Discretizing the Domain

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

$\psi_{i,j}$ and $\omega_{i,j}$

stored at each grid point



Grid boundaries coincide with domain boundaries

Computational Fluid Dynamics
Finite Difference Approximations

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

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

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

Computational Fluid Dynamics
Finite Difference Approximations

Finite difference approximations

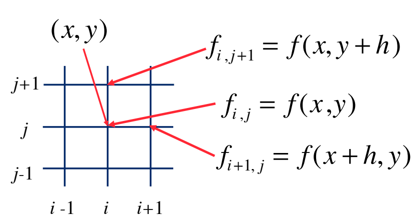
$$\frac{\partial f(x)}{\partial x}=\frac{f(x+h)-f(x-h)}{2h}-\frac{\partial^3 f(x)}{\partial x^3}\frac{h^2}{12}+\cdots$$

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

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

Computational Fluid Dynamics
Finite Difference Approximations

Use the notation developed earlier:



Computational Fluid Dynamics
Finite Difference Approximations

Laplacian

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

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

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

Computational Fluid Dynamics
Finite Difference Approximations

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

The advection equation is:

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

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

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

Computational Fluid Dynamics
Finite Difference Approximations

The vorticity at the new time is given by:

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

Computational Fluid Dynamics
Finite Difference Approximations

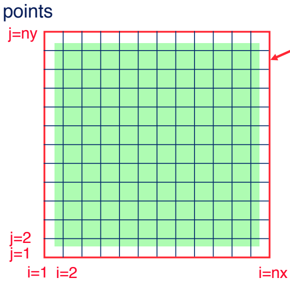
The elliptic equation is:

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

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

Computational Fluid Dynamics
Finite Difference Approximations

These equations allow us to obtain the solution at interior points



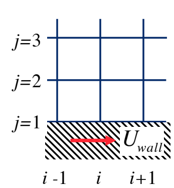
$\psi_{i,j}=0$ on the boundary

Need vorticity on the boundary!

Computational Fluid Dynamics
Discrete Boundary Condition

Consider the bottom wall (j=1):

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

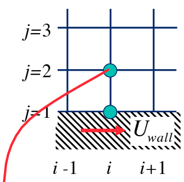


given:

$\psi=\mathrm{Constant}$

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

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

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)
```

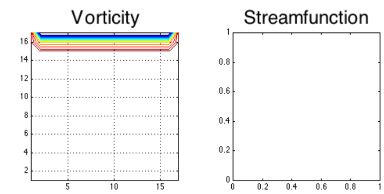
t=t+Δt

Computational Fluid Dynamics
The Code

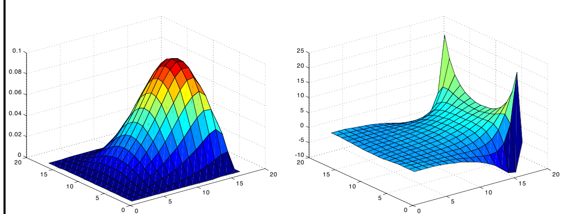
```
clf;nx=9; ny=9; MaxStep=60; Visc=0.1; dt=0.02;         % resolution & governing parameters
MaxIt=100; Beta=1.5; MaxErr=0.001;                     % parameters for SOR iteration
sf=zeros(nx,ny); vt=zeros(nx,ny); w=zeros(nx,ny); h=1.0/(nx-1); t=0.0;
for istep=1:MaxStep,                                   % 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:

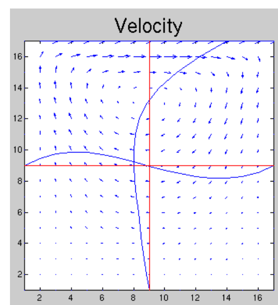


Computational Fluid Dynamics
Results:

17 by 17
Dt=0.01
D=0.1

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

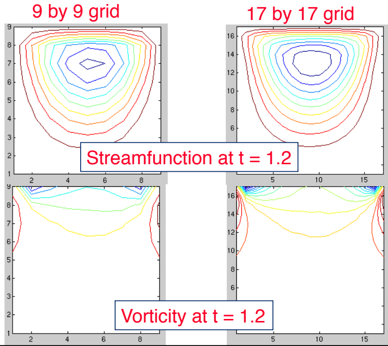
$$v_{i,j} = -\frac{\partial \psi}{\partial x} \approx -\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2h}$$



Computational Fluid Dynamics
Results:

9 by 9 grid

17 by 17 grid



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: $\Psi = (0,0,\psi) \quad \omega = (0,0,\omega)$

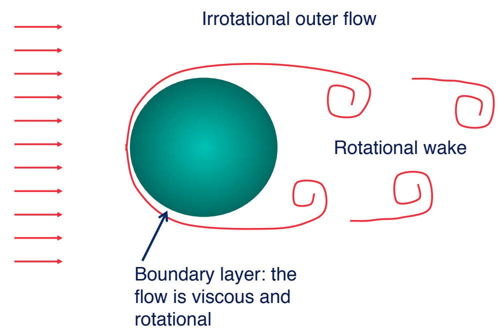
$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

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

$f=0$ at $x=0$, $\quad U \longrightarrow \quad$ $f=1$ at $x=L$

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

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

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

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

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

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

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

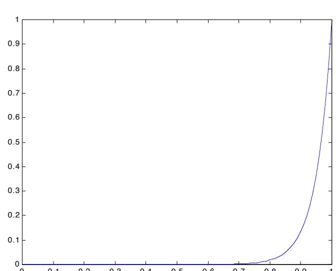
$$\underbrace{C_1=-\exp(C_2) \qquad \exp(C_2)=\frac{1}{\exp(UL/D)-1}}$$

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

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


```
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

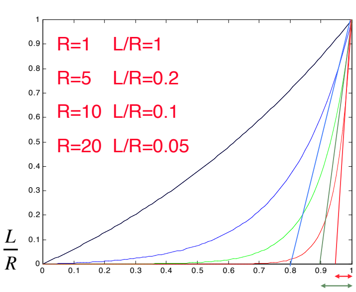


Scaling:

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

Estimate the thickness of the "boundary Layer"

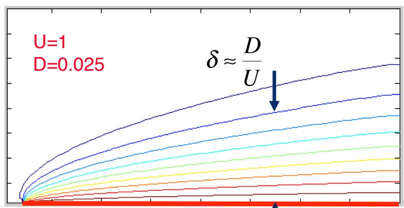
$$\frac{1}{\delta}=\frac{D}{U}\frac{1}{\delta^2}\Rightarrow\delta\approx\frac{DL}{UL}=\frac{L}{R}$$



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