Numerical Marine Hydrodynamics

- Numerical Differentiation
  - Newton Interpolation
  - Finite Differences
- Ordinary Differential Equations
  - Initial Value Problems
    - Euler's Method
    - Taylor Series Methods
      - Error analysis
    - Predictor-Corrector Methods
    - Runge-Kutta Methods
    - Stiff Differential Equations
    - Multistep Methods
    - Error Analysis and Error Modifiers
  - Systems of differential equations
  - Boundary Value Problems
    - Shooting method
    - Direct Finite Difference methods

Initial Value Problems
Error Analysis

Initial Value Problem

$$\frac{dy}{dx} = f(x,y)$$

Integrate

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x,y)dx$$

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

Trapezoidal Rule

$$\int_{x_i}^{x_{i+1}} f(x,y)dx = \frac{f(x_i,y_i) + f(x_{i+1},y_{i+1})}{2}h + O(h^3)$$

Heun's Corrector

$$y_{i+1} = y_i + \frac{f(x_i,y_i) + f(x_{i+1},y_{i+1})}{2}h + O(h^3)$$

Trapezoidal Rule Error

$$E_c = -\frac{1}{12}h^3 f''(\xi_c)$$

Heun's non-Self-starter Predictor

$$y_{i+1} = y_{i-1} + \int_{x_{i-1}}^{x_{i+1}} f(x,y)dx$$

Mid-point Integration

$$\int_{x_{i-1}}^{x_{i+1}} f(x,y)dx = 2hf(x_i,y_i)$$

Heun's non Self-Starter Predictor

$$y_{i+1} = y_{i-1} + 2hf(x_i,y_i)$$

Mid-point Integration Error

$$E_p = \frac{1}{3}h^3 f''(\xi_p)$$

Initial Value Problems
Error Modifiers

Predictor

$$y(x_{i+1}) = y^0_{i+1} + \frac{1}{3}h^3 y^{(3)}(\xi_p)$$

Same Order

Corrector

$$y(x_{i+1}) = y^m_{i+1} - \frac{1}{12}h^3 y^{(3)}(\xi_c)$$

Subtract

$$0 = y^m_{i+1} - y^0_{i+1} - \frac{5}{12}h^3 y^{(3)}(\xi)$$

$$\frac{y^m_{i+1} - y^0_{i+1}}{5} = -\frac{1}{12}h^3 y^{(3)}(\xi)$$

Corrector Error

$$E_c = \frac{y^m_{i+1} - y^0_{i+1}}{5}$$

Corrector Modifier

$$y^m_{i+1} \leftarrow y^m_{i+1} - \frac{y^m_{i+1} - y^0_{i+1}}{5}$$

$$h^3 y^{(3)}(\xi) = -\frac{12}{5}(y^m_i - y^0_i)$$

Predictor Error

$$E_p = \frac{4}{5}(y^m_i - y^0_i)$$

Predictor Modifier

$$y^0_{i+1} \leftarrow y^0_{i+1} - \frac{4}{5}(y^m_i - y^0_i)$$

Replace by

Initial Value Problems
Higher Order Differential Equations

Differential Equation

$$
\begin{aligned}
y^{(n)}(t) &= f(t, y, y', \dots y^{(n-1)}) \\
y(t_0) &= y_0 \\
y'(t_0) &= y_1 \\
&\cdot \\
&\cdot \\
y^{(n-1)}(t_0) &= y_{n-1}
\end{aligned}
$$

The conditions from $y(t_0) = y_0$ to $y^{(n-1)}(t_0) = y_{n-1}$ are the Initial Conditions.

Convert to 1st Order System

$$
\left.
\begin{array}{l}
x_1 = y \\
x_2 = y' \\
x_3 = y'' \\
\cdot \\
\cdot \\
x_n = y^{(n-1)}
\end{array}
\right\} \Rightarrow
\left\{
\begin{array}{ll}
x_1' = x_2 & x_1(t_0) = y_0 \\
x_2' = x_3 & x_2(t_0) = y_1 \\
x_3' = x_4 & x_3(t_0) = y_2 \\
\cdot & \\
\cdot & \\
x_n' = f(t, x1, x2, \dots x_n) & x_n(t_0) = y_{n-1}
\end{array}
\right.
$$

Matrix form

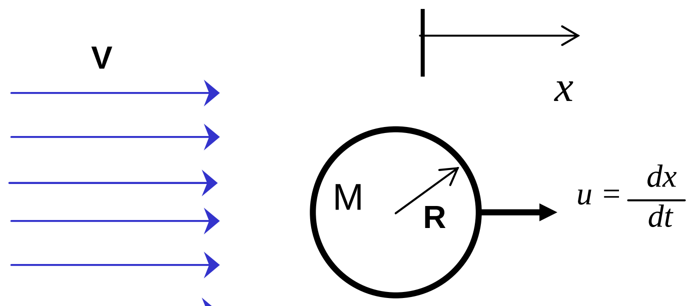
$$\overline{\mathbf{x}}' = \overline{\overline{\mathbf{A}}}\,\overline{\mathbf{x}} + \overline{\mathbf{g}}$$

$$
\overline{\mathbf{x}} = \left\{ \begin{array}{c} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{array} \right\}, \quad
\overline{\overline{\mathbf{A}}} = \left[\begin{array}{ccc} a_{11} & \cdots & a_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{n1} & \cdots & a_{nn} \end{array} \right]
$$

Solved using e.g. Runge-Kutta (ode45)

Sphere Motion in Fluid Flow

MATLAB Solutions



dudt.m

```
function [f] = dudt(t,u)
% u(1) = u
% u(2) = x
% f(2) = dx/dt = u
% f(1) = du/dt=rho*Cd*pi*r/(2m)*(v^2-2uv+u^2)
[rho,Cd,m,r,v] = sph_param();
fac=rho*Cd*pi*r^2/(2*m);

f(1)=fac*(v^2-2*u(1)+u(1)^2);
f(2)=u(1);
f=f';
```

sph_drag_2.m

$$u_{i+1} = u_i + \left(\frac{du}{dt}\right)_i \Delta t, \quad u(0) = 0$$

$$x_{i+1} = x_i + \left(\frac{dx}{dt}\right)_i \Delta t, \quad x(0) = 0$$

```
%step size
h=1.0;
% Euler's method, forward finite difference
t=[0:h:10];
N=length(t);
u_e=zeros(N,1);
x_e=zeros(N,1);
u_e(1)=0;
x_e(1)=0;
[rho,Cd,m,r,v] = sph_param();
fac=rho*Cd*pi*r^2/(2*m);
for n=2:N
    u_e(n)=u_e(n-1)+h*fac*(v^2-2*v*u_e(n-1)+u_e(n-1)^2);
    x_e(n)=x_e(n-1)+h*u_e(n-1);
end
% Runge Kutta
u0=[0 0]';
[tt,u]=ode45(@dudt,t,u0);

figure(1)
hold off
a=plot(t,u_e,'+b');
hold on
a=plot(tt,u(:,1),'.g');
a=plot(tt,abs(u(:,1)-u_e),'+r');
...
figure(2)
hold off
a=plot(t,x_e,'+b');
hold on
a=plot(tt,u(:,2),'.g');
a=plot(tt,abs(u(:,2)-x_e),'xr');
...
```

Boundary Value Problems
Shooting Method

Differential Equation

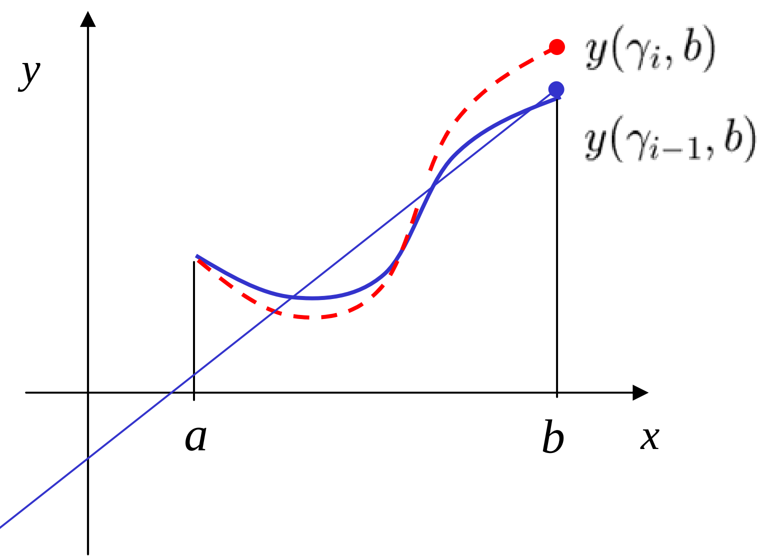
$$y'' = f(x, y, y')$$

$$\left.\begin{aligned} y(a) &= y_a \\ y(b) &= y_b \end{aligned}\right\}$$ Boundary Conditions

'Shooting' Method

$$\left.\begin{aligned} y'' &= f(x, y, y') \\ y(a) &= y_a \\ y'(a) &= \gamma_i \end{aligned}\right\} \rightarrow y(\gamma_i, b)$$

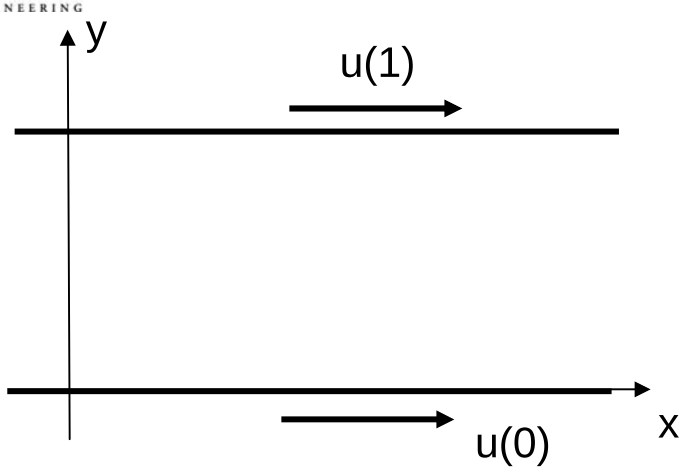
Initial value Problem
Solve by Runge-Kutta



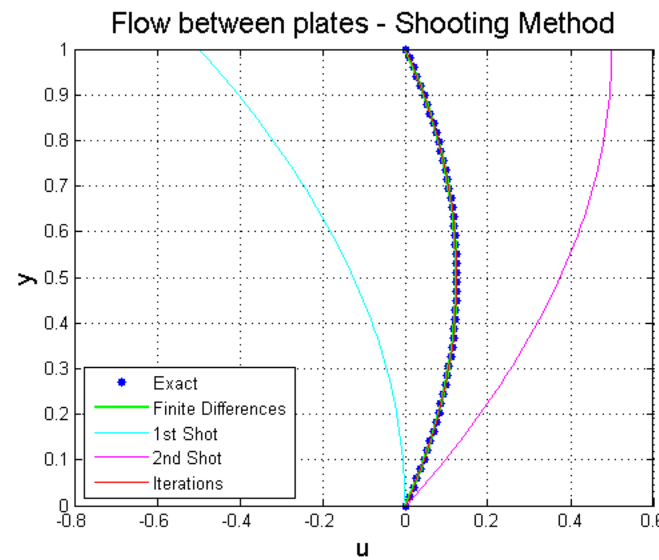
'Shooting' Iteration

$$\gamma_{i+1} = \gamma_{i-1} + (\gamma_i - \gamma_{i-1}) \frac{y_b - y(\gamma_{i-1}, b)}{y(\gamma_i, b) - y(\gamma_{i-1}, b)}$$

Potential Flow between Plates



u(0) = 0
u(1) = 0



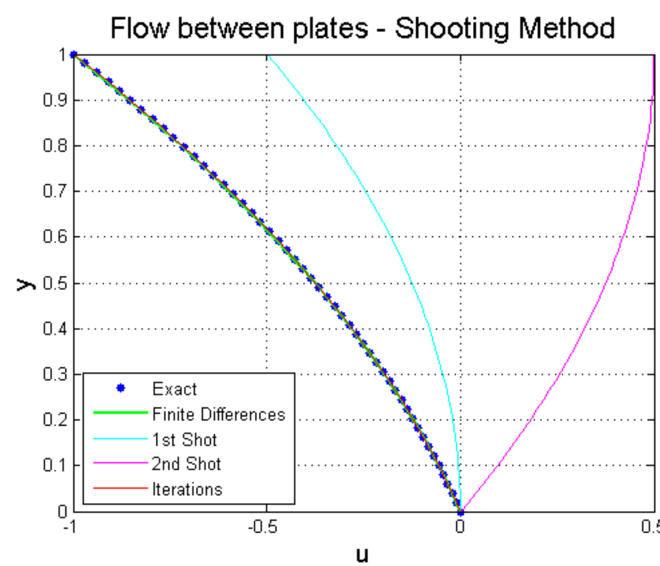
Boundary Value Problem

$$\frac{d^2u}{dy^2} = -v_0$$

$$u(0) = u_0$$

$$u(1) = u_1$$

u(0) = 0
u(1) = -1



Potential Flow between Plates
Shooting Method

$$\frac{d^2u}{dy^2} = -v_0$$
$$u(0) = u_0$$
$$u(1) = u_1$$

dudy.m

```
function [f] = dudy(y,u)
% First order decomp for plate-flow problem
% u(1) = u
% u(2) = q
% f(1) = du/dy = q
% f(2) = dq/dy = -1

f(1)=u(2);
f(2)=-1;
f=f';
```

plate_flow.m

```
% Shooting method
niter=4
u2=zeros(niter,1);
q=zeros(niter,1);
% first guess
q(1)=0;
u_0=[u0 q(1)]';
[x,u]=ode45(@dudy,x,u_0);
plot(u(:,1),x,'c');
u2(1)=u(n,1);
% second guess
q(2)=1;
u_0=[u0 q(2)]';
[x,u]=ode45(@dudy,x,u_0);
plot(u(:,1),x,'m');
u2(2)=u(n,1);
% iteration loop
for i=3:niter
 q(i)=q(i-2) +(q(i-1)-q(i-2))*(u1-u2(i-2))/(u2(i-1)-u2(i-2));
 u_0=[u0 q(i)]';
 [x,u]=ode45(@dudy,x,u_0);
 plot(u(:,1),x,'r');
 u2(i)=u(n,1);
end
grid on
h=legend('Exact','Finite Differences','1st Shot','2nd Shot','Iterations','Location','SouthWest');
set(h,'Fontsize',14);
h=ylabel('y');
set(h,'Fontsize',14);
h=xlabel('u');
set(h,'Fontsize',14);
h=title('Flow between plates - Shooting Method');
set(h,'Fontsize',16);
```

Boundary Value Problems
Direct Finite Difference Methods

Differential Equation

$$y'' = f(x, y, y')$$

$$y(a) = y_a$$
$$y(b) = y_b$$

Boundary Conditions

Discretization

$$h = \frac{b-a}{N}$$

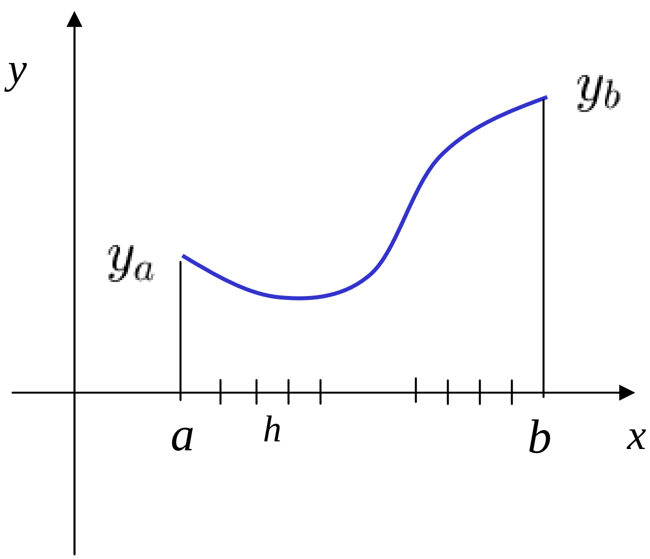
$$x_n = a + nh\ , \quad n = 0, 1 \ldots N$$

Finite Differences

$$y_n = y(x_n)$$

$$y'_n = y'(x_n) = \frac{y_{n+1} - y_{n-1}}{2h} + O(h^2)$$

$$y''_n = y''(x_n) = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + O(h^2)$$



Boundary Value Problems
Direct Finite Difference Methods

Ocean ENGINEERING

Boundary value Problem

$$\begin{aligned} y'' &= f(x, y, y') \\ y(a) &= y_a \\ y(b) &= y_b \end{aligned}$$

Finite Differences

$$\begin{aligned} y_n &= y(x_n) \\ y'_n &= y'(x_n) = \frac{y_{n+1} - y_{n-1}}{2h} + O(h^2) \\ y''_n &= y''(x_n) = \frac{y_{n+1} - 2y_n + y_{n-1}}{h^2} + O(h^2) \end{aligned}$$

Substitute Finite Differences

$$\begin{aligned} y_{n+1} - 2y_n + y_{n-1} &= h^2 f(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}) \ , \quad n = 1, 2, \ldots N\text{-}1 \\ y_0 &= y_a \\ y_n &= y_b \end{aligned}$$

Difference Equations

$$\begin{aligned} -2y_1 + y_2 &= h^2 f\left(x_1, y_1, \frac{y_2 - y_a}{2h}\right) - y_a \\ y_{n-1} - 2y_n + y_{n+1} &= h^2 f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right) \ , \quad n = 2, \ldots N - 2 \\ y_{N-2} - 2y_{N-1} &= h^2 f\left(x_{N-1}, y_{N-1}, \frac{y_b - y_{N-2}}{2h}\right) - y_b \end{aligned}$$

N-1 equations, N-1 unknowns

Matrix Equations

$$\begin{bmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_{N-1} \end{bmatrix} - h^2 \begin{bmatrix} f(x_1, y_1, \frac{y_2 - y_a}{2h}) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f(x_{N-1}, y_{N-1}, \frac{y_b - y_{N-2}}{2h}) \end{bmatrix} = \begin{bmatrix} -y_a \\ 0 \\ \cdot \\ \cdot \\ 0 \\ -y_b \end{bmatrix}$$

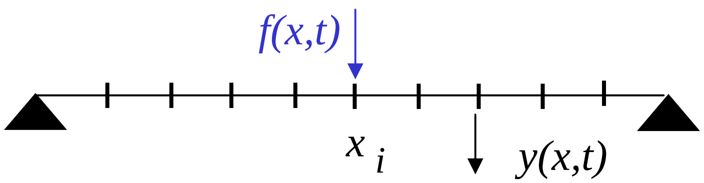
$$\overline{\overline{\mathbf{A}}}\overline{\mathbf{y}} - h^2 \overline{\mathbf{f}}(\overline{\mathbf{y}}) = \overline{\mathbf{r}}$$

$$\begin{aligned} \overline{\overline{\mathbf{A}}}\overline{\mathbf{y}} - h^2 \overline{\overline{\mathbf{G}}}(x)\overline{\mathbf{y}} &= \overline{\mathbf{r}} \\ \left[\overline{\overline{\mathbf{A}}} - h^2 \overline{\overline{\mathbf{G}}}(x)\right]\overline{\mathbf{y}} &= \overline{\mathbf{r}} \end{aligned}$$

Solve using standard linear system solver

Boundary Value Problems
Finite Difference Methods

Forced Vibration of a String



Harmonic excitation

$$f(x,t) = f(x)\cos(\omega t)$$

Differential Equation

$$\frac{d^2y}{dx^2} + k^2 y = f(x)$$

Boundary Conditions

$$y(0) = 0\,, \quad y(L) = 0$$

Finite Difference

$$\left.\frac{d^2y}{dx^2}\right|_{x_i} \simeq \frac{y_{i-1} - 2y_i + y_{i+1}}{h^2}$$

Discrete Difference Equations

$$y_{i-1} + \left((kh)^2 - 2\right) y_i - y_{i+1} = f(x_i)h^2$$

Matrix Form

$$\begin{bmatrix} (kh)^2-2 & 1 & \cdot & \cdot & \cdot\ \cdot & 0 \\ 1 & (kh)^2-2 & 1 & & & \cdot \\ \cdot & & \cdot & \cdot & & \cdot \\ \cdot & & 1 & (kh)^2-2 & 1 & \cdot \\ \cdot & & & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot\ 1 & (kh)^2-2 \end{bmatrix} \overline{\mathbf{x}} = \begin{Bmatrix} f(x_1)h^2 \\ \cdot \\ \cdot \\ f(x_i)h^2 \\ \cdot \\ \cdot \\ f(x_n)h^2 \end{Bmatrix}$$

Tridiagonal Matrix

$kh < 1$ Symmetric, positive definite: No pivoting needed

Linear Systems of Equations
Tri-diagonal Systems

General Tri-diagonal Systems

$$\begin{bmatrix} a_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ b_2 & a_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & b_i & a_i & c_i & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & b_n & a_n \end{bmatrix} \overline{\mathbf{x}} = \begin{Bmatrix} f_1 \\ \cdot \\ \cdot \\ f_i \\ \cdot \\ \cdot \\ f_n \end{Bmatrix}$$

LU Factorization

$$\overline{\overline{\mathbf{A}}} = \overline{\overline{\mathbf{L}}}\,\overline{\overline{\mathbf{U}}}$$

$$\begin{aligned} \overline{\overline{\mathbf{L}}}\overline{\mathbf{y}} &= \overline{\mathbf{f}} \\ \overline{\overline{\mathbf{U}}}\overline{\mathbf{x}} &= \overline{\mathbf{y}} \end{aligned}$$

$$\overline{\overline{\mathbf{L}}} = \begin{bmatrix} 1 & & \cdot & \cdot & \cdot & \cdot & 0 \\ \beta_2 & 1 & & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & \beta_i & 1 & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \beta_n & 1 \end{bmatrix}$$

$$\overline{\overline{\mathbf{U}}} = \begin{bmatrix} \alpha_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ & \alpha_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \alpha_i & c_i & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & & \alpha_n \end{bmatrix}$$

Linear Systems of Equations
Tri-diagonal Systems

LU Factorization

$$\overline{\overline{\mathbf{L}}} = \begin{bmatrix} 1 & & \cdot & \cdot & \cdot & \cdot & 0 \\ \beta_2 & 1 & & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & \beta_i & 1 & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & \beta_n & 1 \end{bmatrix}$$

$$\overline{\overline{\mathbf{U}}} = \begin{bmatrix} \alpha_1 & c_1 & \cdot & \cdot & \cdot & \cdot & 0 \\ & \alpha_2 & c_2 & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \alpha_i & c_i & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & & \alpha_n \end{bmatrix}$$

Reduction

$$\alpha_1 = a_1$$

$$\beta_k = \frac{b_k}{\alpha_{k-1}} \; , \quad \alpha_k = a_k - \beta_k c_{k-1} \; , \; k = 2, 3, \ldots n$$

Forward Substitution

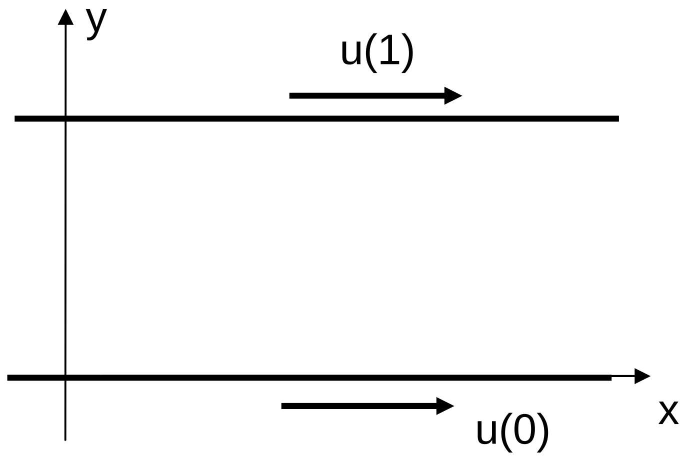
$$y_1 = f_1 \; , \quad y_i = f_i - \beta_i y_{i-1} \; , \; i = 2, 3, \ldots n$$

Back Substitution

$$x_n = \frac{y_n}{\alpha_n} \; , \quad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_1} \; , \; i = n-1, \ldots 1$$

LU Factorization:	2*(n-1) operations
Forward substitution:	n-1 operations
Back substitution:	n-1 operations
Total:	4(n-1) ~ O(n) operations

Potential Flow between Plates
Direct Finite Difference Method



$$\frac{d^2u}{dy^2} = -v_0$$

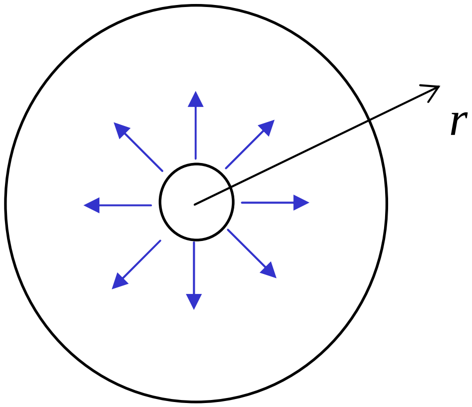
$$u(0) = u_0$$

$$u(1) = u_1$$

plate_flow.m

```
x=[0:h:L]';
a(1,1) =-2;
a(1,2)=1;
for i=2:n-1
    a(i,i)=a(1,1);
    a(i,i-1) = 1;
    a(i,i+1) = 1;
end
a(n,n) = a(1,1);
a(n,n-1)=1;
f=-ones(n,1)*h^2;
%Boundary condition
a(1,1)=1e20;
a(n,n)=1e20;
f(1)=u0*1e20;
f(n)=u1*1e20;
% Tri-diagonal form
d=diag(a);
b=ones(n,1);
c=b;
% LU factorization
[alf,bet]=lu_tri(d,b,c);
% Forward substitution
z=forw_tri(f,bet);
% Back substitution
y_b=back_tri(z,alf,c);
hold on
h=plot(y_b,x,'g');
set(h,'Linewidth',2);
```

Potential Flow
Distributed Sources and Drains



Solution Interval

$$r_0 \leq r \leq 4r_0$$

Potential ODE

$$\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr}\right)\phi = \omega_0 \frac{r}{r_0}$$

$$u_r = \frac{d\phi}{dr}$$

Dimensionless Form

$$R = \frac{r}{r_0}, \ \phi = \frac{\Phi}{r_0 u_0}, \ u = \frac{u_r}{u_0}$$

$$1 \leq R \leq 4$$

$$\left(\frac{d^2}{dR^2} + \frac{1}{R}\frac{d}{dR}\right)\Phi = \alpha R$$

$$U = \frac{d\Phi}{dR}, \ \alpha = \frac{\omega_0 r_0}{u_0}$$

Potential Flow
Distributed Sources and Drains

Example

$$\left(\frac{d^2}{dR^2} + \frac{1}{R}\frac{d}{dR}\right)\Phi = 9R$$

Analytical Solution

$$\Phi = R^3 - 24\log R$$

$$U = 3R^2 - 24/R$$

Boundary Conditions

$$\Phi = 1 \qquad \text{at} R = 1$$

$$\Phi = 64 - 24\log 4 \quad \text{at} R = 4$$

1st order form

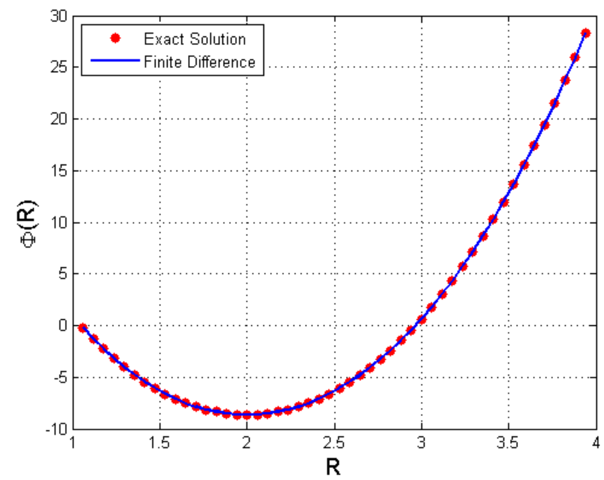
$$\frac{d\Phi}{dR} = U$$

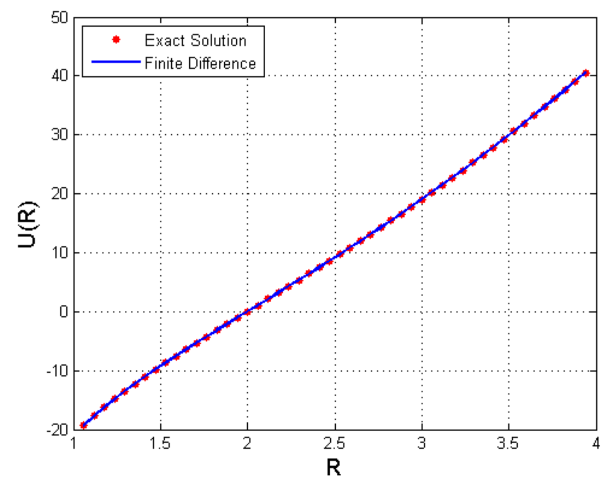
$$\frac{dU}{dR} = 9R - U/R$$

Potential Flow
Distributed Sources and Drains

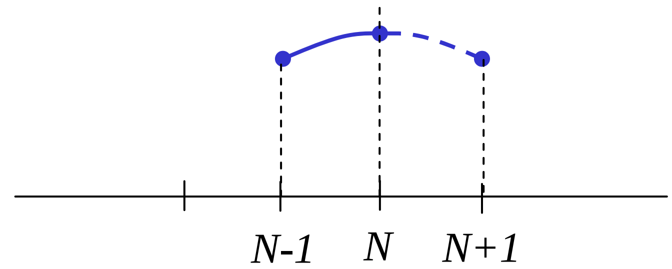
sou_flow.m

```
n=50; r1=1; r2=4;
phi1=1;
phi2=64-24*log(4);
h=(r2-r1)/(n+1);
r=[r1+h:h:r2-h]';
a=zeros(n,n);
f=9*r*h^2;
a(1,1) =- 2;
a(1,2)=1;
for i=2:n-1
    a(i,i)=a(1,1);
    a(i,i-1) = 1;
    a(i,i+1) = 1;
end
a(n,n) = a(1,1);
a(n,n-1)=1;
% off-diagonal terms
for i=1:n-1
    a(i,i+1)=a(i,i+1)+ 0.5*h/r(i);
    a(i+1,i)=a(i+1,i)- 0.5*h/r(i+1);
end
% right hand side
f(1)=f(1)-phi1+0.5*h*phi1/r(1);
f(n)=f(n)-phi2-0.5*h*phi2/r(n);
d=diag(a);
for i=1:n-1
    b(i+1)=a(i+1,i);
    c(i)=a(i,i+1);
end
% LU factorization
[alf,bet]=lu_tri(d,b,c);
% Forward substitution
z=forw_tri(f,bet);
% Back substitution
y_b=back_tri(z,alf,c);
```





Boundary Value Problems
Finite Difference Methods



Boundary Conditions with Derivatives

$$y'' - yx = g(x)$$
$$y(a) = 0$$

Central Difference

$$y'(b) = 0$$

Difference Equations

$$y_0 = 0$$
$$y_{n-1} - 2y_n + y_{n+1} - h^2 y_n x_n = h^2 g(x_n)\ , \quad n = 1,$$
$$y_N = ?$$

Backward Difference

$$y'(b) = 0 = \frac{y_N - y_{N-1}}{h} + O(h)$$

$$y_0 = 0$$
$$y_{n-1} - 2y_n + y_{n+1} - h^2 y_n x_n = h^2 g(x_n)\ , \quad n = 1, 2, \ldots N-1$$
$$y_N - y_{N-1} = 0$$

$O(h^4)$

$O(h^2)$

Central Difference

$$y'(b) = 0 = \frac{y_{N+1} - y_{N-1}}{2h} + O(h^2)$$

$$y_0 = 0$$
$$y_{n-1} - 2y_n + y_{n+1} - h^2 y_n x_n = h^2 g(x_n)\ , \quad n = 1, 2, \ldots N-1$$
$$2(y_{N-1} - y_N) - h^2 y_N x_N = 0$$

$O(h^3)$

General Boundary Conditions

$$p_0 y(b) + p_1 y'(b) = p_2$$

Finite Difference Representation

$$p_0 y_N + \frac{p_1 (y_{N+1} - y_{N-1})}{2h} = p_2$$

Add extra point - N equations, N unknowns