2.29 Numerical Fluid Mechanics
Spring 2015 – **Lecture 14**

REVIEW Lecture 13:

- **Stability:** Von Neumann Ex.: 1st order linear convection/wave eqn., F-B scheme
- **Hyperbolic PDEs and Stability** $C = \frac{c\,\Delta t}{\Delta x} < 1$
  - 2^{nd} order wave equation and waves on a string
    - Characteristic finite-difference solution
    - Stability of C – C (CDS in time/space, explicit): $C = \frac{c\,\Delta t}{\Delta x} < 1$
    - Example: Effective numerical wave numbers and dispersion
  - CFL condition:
    - "Numerical domain of dependence" must include "Mathematical domain of dependence"
    - Examples: 1st order linear convection/wave eqn., 2nd order wave eqn.
    - Other FD schemes (C 2^{nd} – C 4^{th})
  - Von Neumann: 1^{st} order linear convection/wave eqn., F- C: unstable
  - Stability summary: Tables of schemes for 1^{st} order linear convection/wave eqn.
- **Elliptic PDEs**
  - FD schemes for 2D problems (Laplace, Poisson and Helmholtz eqns.)
    - Direct 2^{nd} order and Iterative (Jacobi, Gauss-Seidel)
  - Boundary conditions

TODAY (Lecture 14): FINITE DIFFERENCES, Cont'd

- **Elliptic PDEs, Continued**
  - Examples, Higher order finite differences
  - Irregular boundaries: Dirichlet and Von Neumann BCs
  - Internal boundaries
- **Parabolic PDEs and Stability**
  - Explicit schemes (1D-space)
    - Von Neumann
  - Implicit schemes (1D-space): simple and Crank-Nicholson
    - Von Neumann
  - Examples
  - Extensions to 2D and 3D
    - Explicit and Implicit schemes
    - Alternating-Direction Implicit (ADI) schemes

TODAY (Lecture 14, Cont'd): FINITE VOLUME METHODS

- Integral forms of the conservation laws
- Introduction to FV Methods
- Approximations needed and basic elements of a FV scheme
  - FV grids
  - Approximation of surface integrals (leading to symbolic formulas)
  - Approximation of volume integrals (leading to symbolic formulas)
- Summary: Steps to step-up FV scheme
- Examples: One Dimensional examples
  - Generic equations
  - Linear Convection (Sommerfeld eqn.): convective fluxes
    - 2^{nd} order in space, 4^{th} order in space, links to CDS
  - Unsteady Diffusion equation: diffusive fluxes
    - Two approaches for 2^{nd} order in space, links to CDS

References and Reading Assignments

- Lapidus and Pinder, 1982: Numerical solutions of PDEs in Science and Engineering. Section 4.5 on "Stability".
- Chapter 3 on "Finite Difference Methods" of "J. H. Ferziger and M. Peric, Computational Methods for Fluid Dynamics. Springer, NY, 3rd edition, 2002"
- Chapter 3 on "Finite Difference Approximations" of "H. Lomax, T. H. Pulliam, D.W. Zingg, *Fundamentals of Computational Fluid Dynamics (Scientific Computation).* Springer, 2003"
- Chapter 29 and 30 on "Finite Difference: Elliptic and Parabolic equations" of "Chapra and Canale, Numerical Methods for Engineers, 2014/2010/2006."

Elliptic PDEs
Iterative Schemes: Laplace equation $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

Finite Difference Scheme

$$u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^{k+1} = 0$$

Liebman Iterative Scheme (Jacobi/Gauss-Seidel)

$$u_{i,j}^{k+1} = u_{i,j}^k + r_{i,j}^k$$

$$r_{i,j}^k = \frac{u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k}{4}$$

equivalent

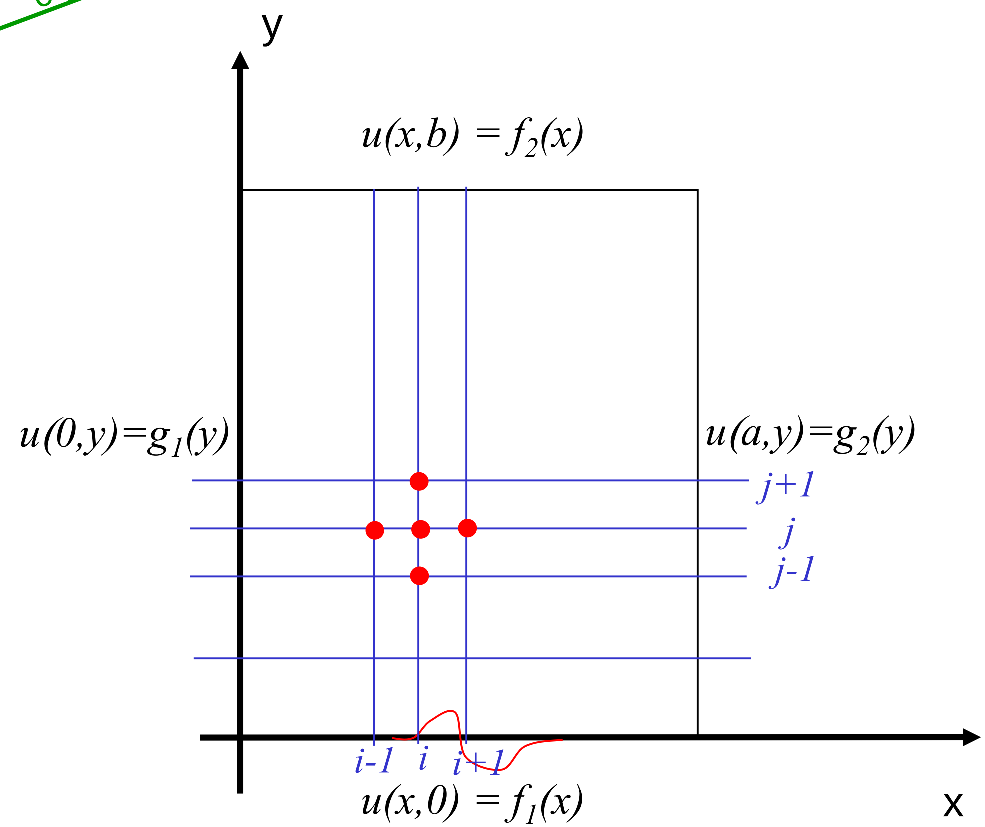
$$u_{i,j}^{k+1} = \frac{u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k}{4}$$

SOR Iterative Scheme, Jacobi

$$\begin{aligned} u_{i,j}^{k+1} &= u_{i,j}^k + \omega r_{i,j}^k \\ &= u_{i,j}^k + \omega \frac{u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k - 4u_{i,j}^k}{4} \\ &= (1-\omega)\, u_{i,j}^k + \omega \frac{u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k}{4} \end{aligned}$$

Optimal SOR (Equidistant Sampling *h*)

$$\omega = \frac{4}{2 + \sqrt{4 - \left[\cos\left(\frac{\pi}{n-1}\right) + \cos\left(\frac{\pi}{m-1}\right)\right]^2}}$$



Elliptic PDE: Poisson Equation

$$\nabla^2 u = g(x, y)$$

$$g_{i,j} = g(x_i, y_j)$$

SOR Iterative Scheme, with Jacobi

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \omega r_{i,j}^{k}$$

$$= u_{i,j}^{k} + \omega \frac{u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - 4u_{i,j}^{k} - h^2 g_{i,j}}{4}$$

$$= (1-\omega)\, u_{i,j}^{k} + \omega \frac{u_{i+1,j}^{k} + u_{i-1,j}^{k} + u_{i,j+1}^{k} + u_{i,j-1}^{k} - h^2 g_{i,j}}{4}$$

Elliptic PDE: Poisson Equation

$$\nabla^2 u = g(x, y)$$

$$g_{i,j} = g(x_i, y_j)$$

SOR Iterative Scheme, with Gauss-Seidel

$$u_{i,j}^{k+1} = u_{i,j}^{k} + \omega r_{i,j}^{k}$$

$$= u_{i,j}^{k} + \omega \frac{u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + u_{i,j+1}^{k} + u_{i,j-1}^{k+1} - 4u_{i,j}^{k} - h^2 g_{i,j}}{4}$$

$$= (1-\omega)\, u_{i,j}^{k} + \omega \frac{u_{i+1,j}^{k} + u_{i-1,j}^{k+1} + u_{i,j+1}^{k} + u_{i,j-1}^{k+1} - h^2 g_{i,j}}{4}$$

Laplace Equation
Steady Heat diffusion (with source: Poisson eqn)

duct.m

```
Lx=1;
Ly=1;
N=10;
h=Lx/N;
M=floor(Ly/Lx*N);
niter=20;
eps=1e-6;

x=[0:h:Lx]';
y=[0:h:Ly];
f1x='4*x-4*x.^2';
%f1x='0'
f2x='0';
g1x='0';
g2x='0';
gxy='0';
f1=inline(f1x,'x');
f2=inline(f2x,'x');
g1=inline(g1x,'y');
g2=inline(g2x,'y');
gf=inline(gxy,'x','y');

n=length(x);
m=length(y);
u=zeros(n,m);
u(2:n-1,1)=f1(x(2:n-1));
u(2:n-1,m)=f2(x(2:n-1));
u(1,1:m)=g1(y);
u(n,1:m)=g2(y);
for i=1:n
    for j=1:m
        g(i,j) = gf(x(i),y(j));
    end
end
```

```
u_0=mean(u(1,:))+mean(u(n,:))+mean(u(:,1))+mean(u(:,m));
u(2:n-1,2:m-1)=u_0*ones(n-2,m-2);
omega=4/(2+sqrt(4-(cos(pi/(n-1))+cos(pi/(m-1)))^2))
for k=1:niter
    u_old=u;
    for i=2:n-1
        for j=2:m-1
            u(i,j)=(1-omega)*u(i,j)
 +omega*(u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)-h^2*g(i,j))/4;
        end
    end
    r=abs(u-u_old)/max(max(abs(u)));
    k,r
    if (max(max(r))<eps)
        break;
    end
end
figure(3)
surf(y,x,u);
shading interp;
a=ylabel('x');
set(a,'Fontsize',14);
a=xlabel('y');
set(a,'Fontsize',14);
a=title(['Poisson Equation - g = ' gxy]);
set(a,'Fontsize',16);
```

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x,y)$$

BCs: $u(x,0,t) = f(x) = 4x - 4x^2$

Three other BCs are null

Helmholtz Equation

$$\nabla^2 u + f(x,y)u = g(x,y)$$

$$f_{i,j} = f(x_i, y_j)$$

$$g_{i,j} = g(x_i, y_j)$$

SOR Iterative Scheme, with Gauss-Seidel

$$u_{i,j}^{k+1} = u_{i,j}^k + \omega r_{i,j}^k$$

$$= u_{i,j}^k + \omega \frac{u_{i+1,j}^k + u_{i-1,j}^{k+1} + u_{i,j+1}^k + u_{i,j-1}^{k+1} - (4 - \underline{h^2 f_{i,j}}) u_{i,j}^k - h^2 g_{i,j}}{(4 - \underline{h^2 f_{i,j}})}$$

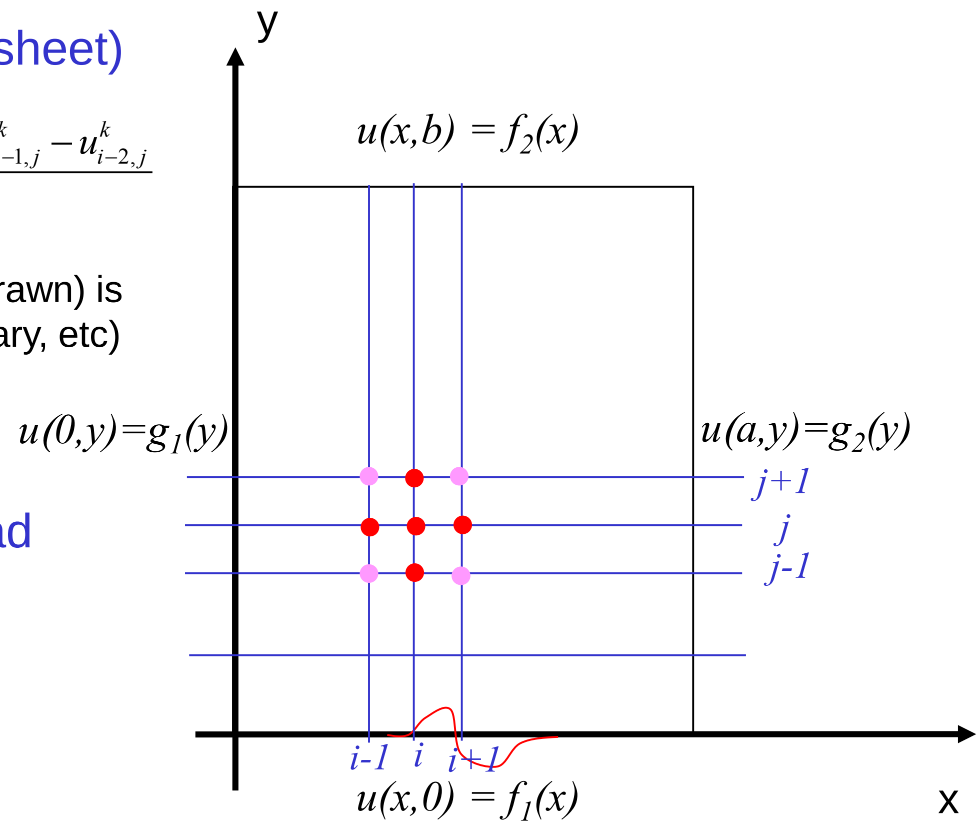
$$\boxed{= (1-\omega)\, u_{i,j}^k + \omega \frac{u_{i+1,j}^k + u_{i-1,j}^{k+1} + u_{i,j+1}^k + u_{i,j-1}^{k+1} - h^2 g_{i,j}}{(4 - h^2 f_{i,j})}}$$

Elliptic PDE's
Higher Order Finite Differences

CD, 4th order (see tables in eqs. sheet)

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{\text{CD, 4}^{\text{th}}\text{ order}} = \frac{-u_{i+2,j}^k + 16\,u_{i+1,j}^k + 30\,u_{i,j}^k + 16\,u_{i-1,j}^k - u_{i-2,j}^k}{12h^2}$$

The resulting 9 point "cross" stencil (not drawn) is more challenging computationally (boundary, etc) than CD 2nd order.



Use more compact scheme instead

Square stencil (see figure):

- Use Taylor series, then cancel the terms so as to get a 4th order scheme

- Leads to:

$$\nabla^2 u_{i,j} = \frac{1}{6h^2}[u_{i+1,j-1} + u_{i-1,j-1} + u_{i+1,j+1} + u_{i-1,j-1}$$
$$+4u_{i+1,j} + 4u_{i-1,j} + 4u_{i,j+1} + 4u_{i,j-1} - 20u_{i,j}] + O(h^4)$$

Elliptic PDEs: Irregular Boundaries

- Many elliptic problems don't have simple boundaries/geometries
- One way to handle them is through "irregular" discrete boundary cells (e.g. shaved cells)

1) Boundary Stencils (with Dirichlet BCs)

$$\left(\frac{\partial u}{\partial x}\right)_{i-1,i} \simeq \frac{u_{i,j} - u_{i-1,j}}{\alpha_1 \Delta x}$$

$$\left(\frac{\partial u}{\partial x}\right)_{i,i+1} \simeq \frac{u_{i+1,j} - u_{i,j}}{\alpha_2 \Delta x}$$

1st derivatives evaluated at center of edges, hence dx is sum of half edge lengths on each side

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \frac{\left(\frac{\partial u}{\partial x}\right)_{i,i+1} - \left(\frac{\partial u}{\partial x}\right)_{i-1,i}}{\frac{\alpha_1 \Delta x + \alpha_2 \Delta x}{2}}$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{2}{\Delta x^2}\left[\frac{u_{i-1,j} - u_{i,j}}{\alpha_1(\alpha_1 + \alpha_2)} + \frac{u_{i+1,j} - u_{i,j}}{\alpha_2(\alpha_1 + \alpha_2)}\right]$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{2}{\Delta y^2}\left[\frac{u_{i,j-1} - u_{i,j}}{\beta_1(\beta_1 + \beta_2)} + \frac{u_{i,j+1} - u_{i,j}}{\beta_2(\beta_1 + \beta_2)}\right]$$

Can be used directly with Dirichlet BCs

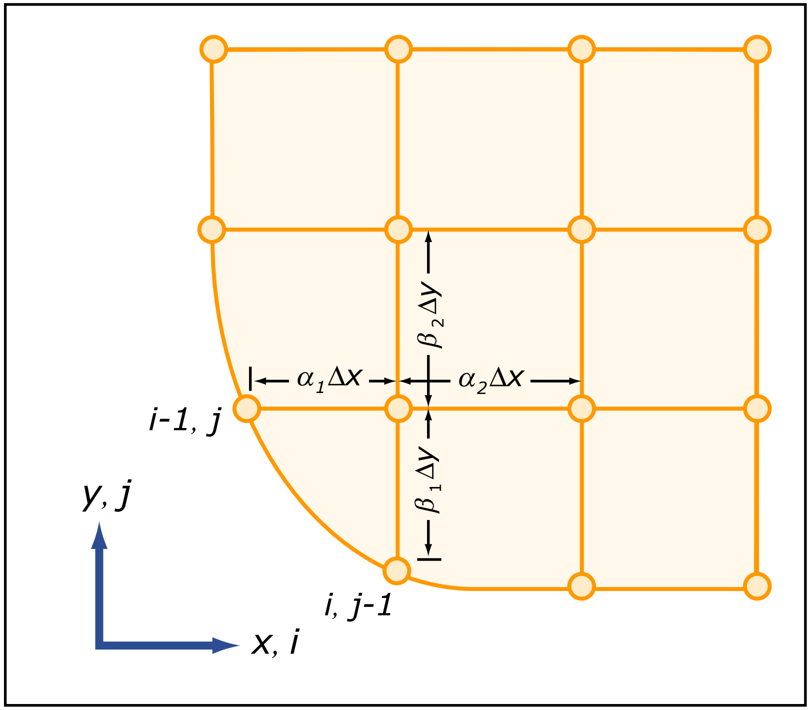


Image of a grid for a heated plate that has an irregularly shaped boundary. Image by MIT OpenCourseWare.

- Leads to direct and iterative elliptic solvers as before, but with updated coefficients for the boundary stencils
- Other options possible: curved boundary elements

Elliptic PDEs: Irregular Boundaries

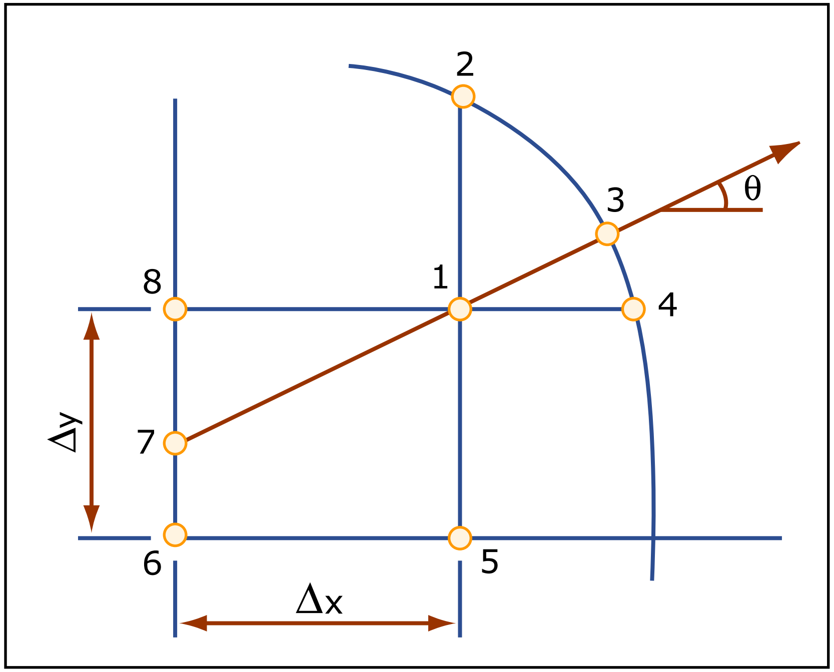
2) Neumann Boundary Conditions
(e.g. normal derivative given)

$$\frac{\partial u}{\partial \eta} = \frac{u_1 - u_7}{L_{17}}$$

$$L_{78} = \Delta x \tan\theta$$
$$L_{17} = \Delta x / \cos\theta$$

Linear interpolation at 7 $\quad u_7 = u_8 + (u_6 - u_8)\dfrac{\Delta x \tan\theta}{\Delta y}$

$$u_1 = \left(\frac{\Delta x}{\cos\theta}\right)\frac{\partial u}{\partial \eta} + u_6\frac{\Delta x \tan\theta}{\Delta y} + u_8\left(1 - \frac{\Delta x \tan\theta}{\Delta y}\right)$$



Curved boundary in which the normal gradient is specified.
Image by MIT OpenCourseWare.

- This is an approach given in Chapra & Canale
- One may instead estimate u_3 from neighbor nodes, then take the derivative along 1-3

Elliptic PDEs
Internal (Fixed) Boundaries

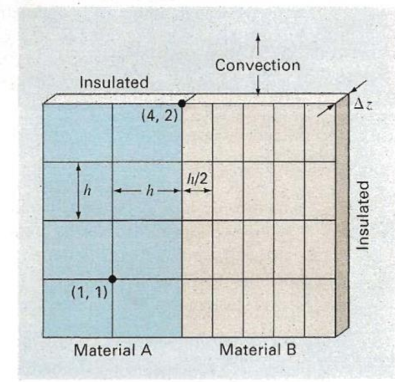


FIGURE 29.13
A heated plate with unequal grid spacing, two materials, and mixed boundary conditions.

Source: Chapra, S. and R. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 2005.

Velocity and Stress Continuity (heat flux or viscous stress)

$$u^+ = u^-$$

$$\mu^+ \frac{\partial u^+}{\partial y} = \mu^- \frac{\partial u^-}{\partial y}$$

Derivative Finite Differences (1^{st} order)

$$\mu^+ \frac{\partial u^+}{\partial y} = \mu^+ \frac{u_{i,j+1} - u_{i,j}}{h}$$

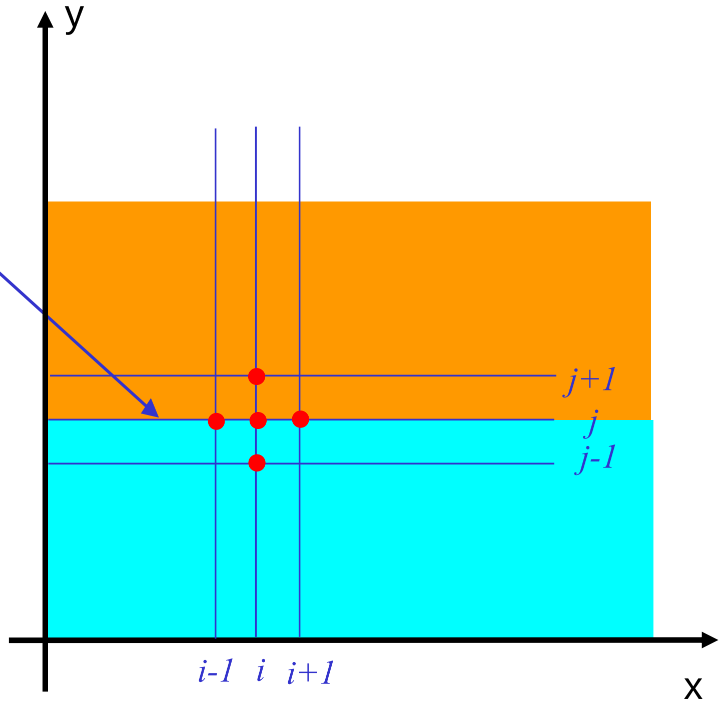
$$\mu^- \frac{\partial u^-}{\partial y} = \mu^- \frac{u_{i,j} - u_{i,j-1}}{h}$$

Finite Difference Equation at bnd.

$$(\mu^- + \mu^+)u_{i,j} = \mu^+ u_{i,j+1} + \mu^- u_{i,j-1}$$

SOR Finite Difference Scheme at bnd.

$$u_{i,j}^{k+1} = (1-\omega)u_{i,j}^k + \omega \frac{\mu^+ u_{i,j+1}^k + \mu^- u_{i.j-1}^k}{\mu^- + \mu^+}$$



Elliptic PDEs
Internal (Fixed) Boundaries – Higher Order

$$\mu^{\pm}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)=f(x,y), \quad \frac{f(x,y)}{\mu^{\pm}}=g^{\pm}(x,y)$$

Velocity and Stress Continuity

$$u^+ = u^-$$

$$\mu^+\frac{\partial u^+}{\partial y} = \mu^-\frac{\partial u^-}{\partial y}$$

Taylor Series, inserting the PDE

$$u_{i,j-1} \simeq u_{i,j} - h u_y(x_i,y_j) + \frac{h^2}{2}u_{yy}(x_i,y_j)$$

$$= u_{i,j} - h u_y(x_i,y_j) + \frac{h^2}{2}\left(g^-_{i,j} - u_{xx}(x_i,y_j)\right)$$

$$u_{i,j+1} \simeq u_{i,j} + h u_y(x_i,y_j) + \frac{h^2}{2}u_{yy}(x_i,y_j)$$

$$= u_{i,j} + h u_y(x_i,y_j) + \frac{h^2}{2}\left(g^+_{i,j} - u_{xx}(x_i,y_j)\right)$$

Derivative Finite Differences (2nd order)

$$\mu^+\frac{\partial u^+}{\partial y} = \mu^+\left[\frac{u_{i,j+1}-u_{i,j}}{h} + \frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{2h} - \frac{h}{2}g^+_{i,j}\right]$$

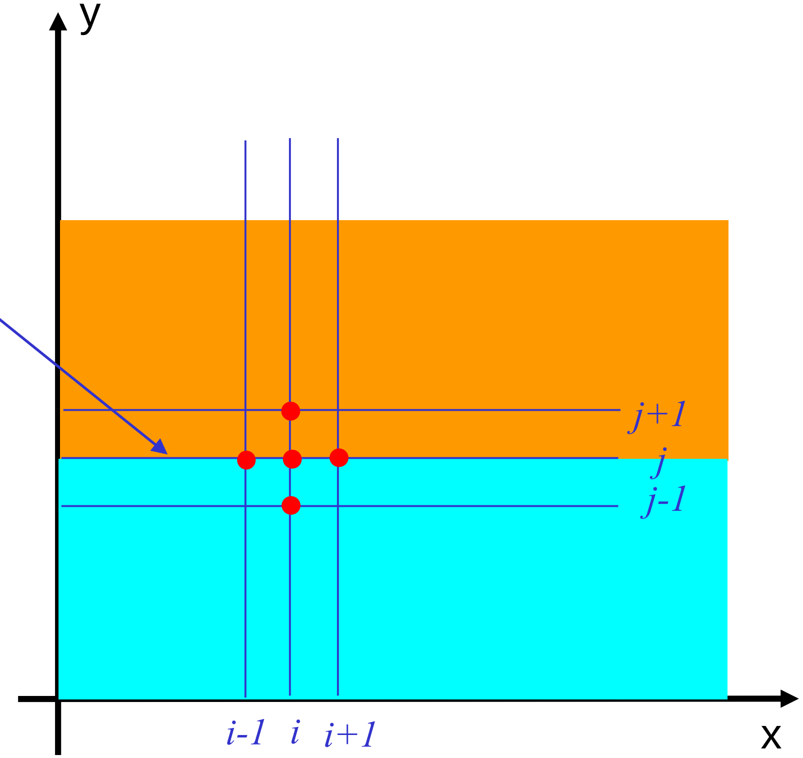
$$\mu^-\frac{\partial u^-}{\partial y} = \mu^-\left[\frac{u_{i,j}-u_{i,j-1}}{h} - \frac{u_{i-1,j}-2u_{i,j}+u_{i+1,j}}{2h} + \frac{h}{2}g^-_{i,j}\right]$$

Finite Difference Equation at bnd.

$$\left[\frac{2(\mu^+u_{i,j+1}+\mu^-u_{i,j-1})/(\mu^-+\mu^+) + u_{i+1,j} + u_{i-1,j} - 4u_{i,j} - h^2\bar{g}_{i,j}}{4}\right] = 0$$

SOR Finite Difference Scheme at bnd.

$$u_i^{k+1} = (1-\omega)u_i^k + \omega\left[\frac{2(\mu^+u^k_{i,j+1}+\mu^-u^k_{i,j-1})/(\mu^-+\mu^+) + u^k_{i+1,j} + u^k_{i-1,j} - h^2\bar{g}_{i,j}}{4}\right]$$



Source: Chapra, S. and R. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 2005.

$$\bar{g}_{i,j} = \frac{\mu^- g^-_{i,j} + \mu^+ g^+_{i,j}}{2} = \bar{f}_{i,j}$$

Partial Differential Equations
Parabolic PDE: $B^2 - 4AC = 0$

Examples

$$\frac{\partial T}{\partial t} = \frac{\kappa}{\rho c} \nabla^2 T + f \,, \quad (\alpha = \frac{\kappa}{\rho c})$$

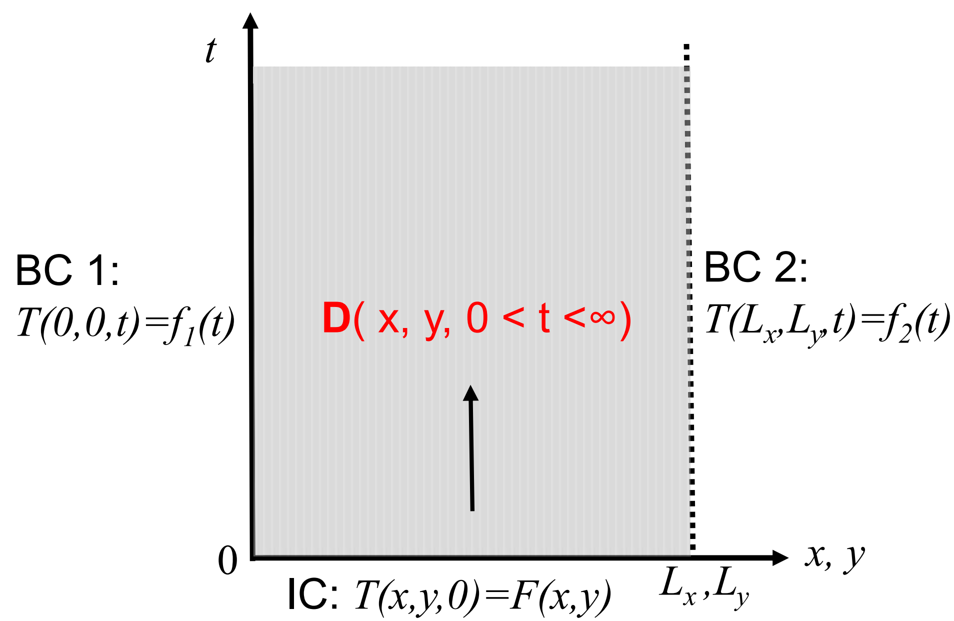
Heat conduction equation, forced or not (dominant in 1D)

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \, \nabla^2 \mathbf{u} + \mathbf{g}$$

Unsteady, diffusive, small amplitude flows or perturbations (e.g. Stokes Flow)

- Usually smooth solutions ("diffusion effect" present)
- "Propagation" problems
- Domain of dependence of solution is domain **D** (x, y, and 0 < t < ∞):

- Finite Differences/Volumes, Finite Elements



Partial Differential Equations
Parabolic PDE: 1D Heat Conduction example

Heat Conduction Equation

$$T_t(x,t) = \alpha T_{xx}(x,t), 0 < x < L, 0 < t < \infty$$

$$\alpha = \frac{\kappa}{\rho c}$$

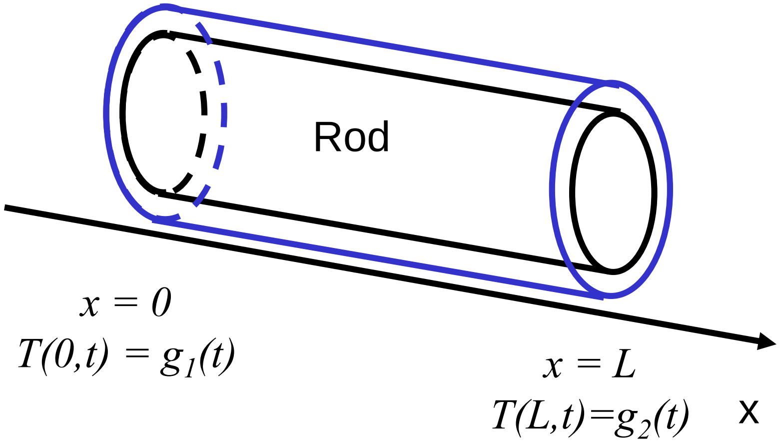
Initial Condition

$$T(x,0) = f(x), 0 \le x \le L$$

Boundary Conditions

$$T(0,t) = g_1(t), 0 < t < \infty$$

$$T(L,t) = g_2(t), 0 < t < \infty$$



Parabolic PDE

1D Heat Conduction: Forward in time, centered in space, explicit

Equidistant Sampling

$$h = L/n$$
$$k = T/m$$

Discretization

$$x_i = (i-1)h, \; i = 2, \ldots, n-1$$
$$t_j = (j-1)k, \; j = 1, \ldots, m$$

Forward (Euler) Finite Difference in time

$$T_t(x,t) = \frac{T(x_i, t_{j+1}) - T(x_i, t_j)}{k} + O(k)$$

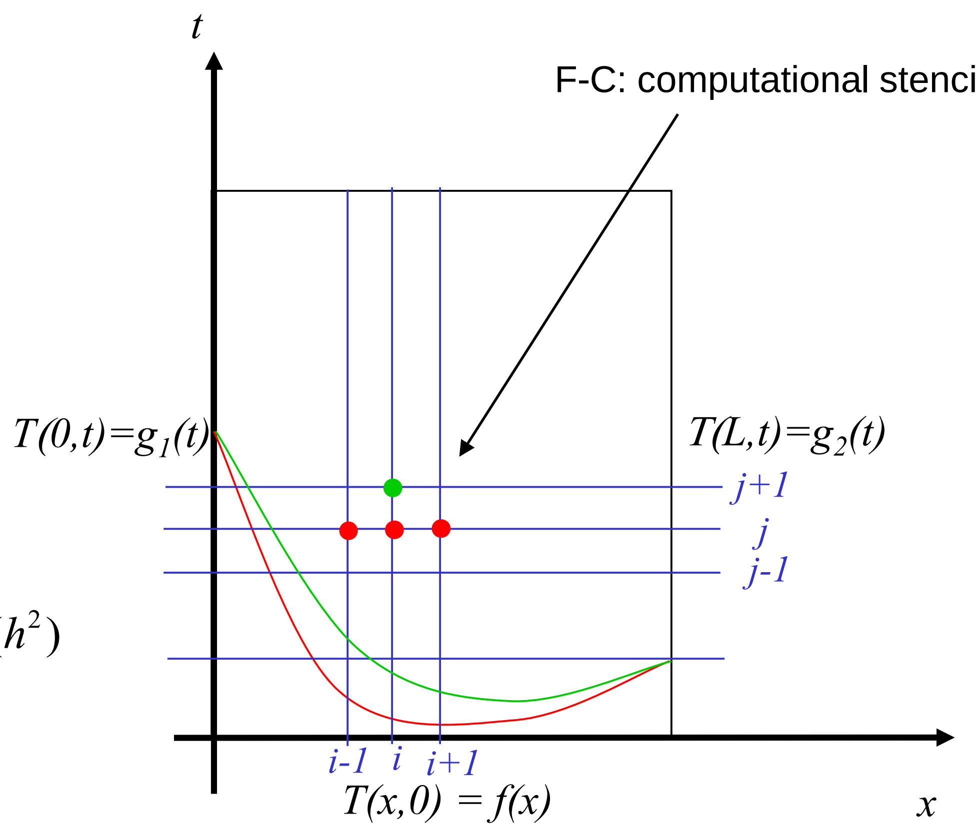
Centered Finite Difference in space

$$T_{xx}(x,t) = \frac{T(x_{i-1}, t_j) - 2T(x_i, t_j) + T(x_{i+1}, t_j)}{h^2} + O(h^2)$$

$$T_{i,j} = T(x_i, t_j)$$

Finite Difference Equation

$$\frac{T_{i,j+1} - T_{i,j}}{k} = \alpha \frac{T_{i-1,j} - 2T_{i,j} + T_{i+1,j}}{h^2}$$



Parabolic PDE

1D Heat Conduction: Forward in time, centered in space, explicit

Dimensionless diffusion coefficient

$$r = \frac{\alpha k}{h^2}$$

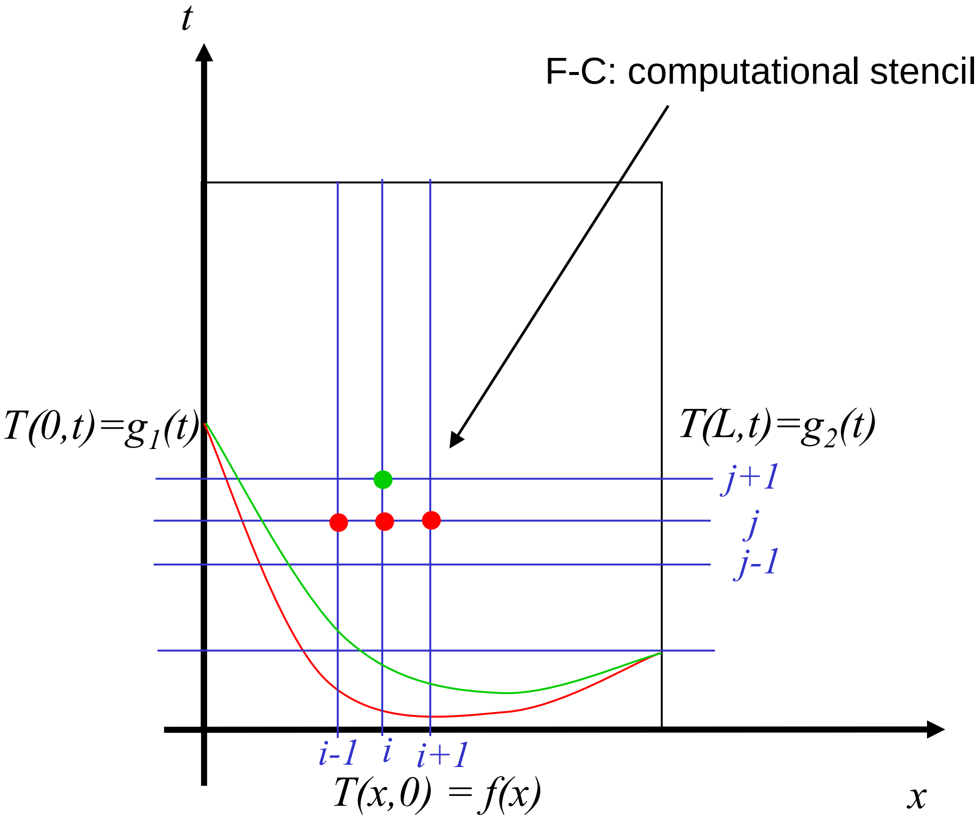
Explicit Finite Difference Scheme

$$T_{i,j+1} = (1-2r)T_{i,j} + r(T_{i-1,j} + T_{i+1,j})$$

Stability Requirement

$$r <= 0.5$$

Conditionally stable
(von Neumann)
Shown in class on blackboard



Heat Conduction Equation
Explicit Finite Differences

(1D-in-space, unsteady case;
similar to steady elliptic problem seen previously)

T denoted by u, i.e. $u_{i,j} \equiv T_{i,j}$

α denoted by c^2, i.e. $c^2 \equiv \alpha$

heat_fw.m

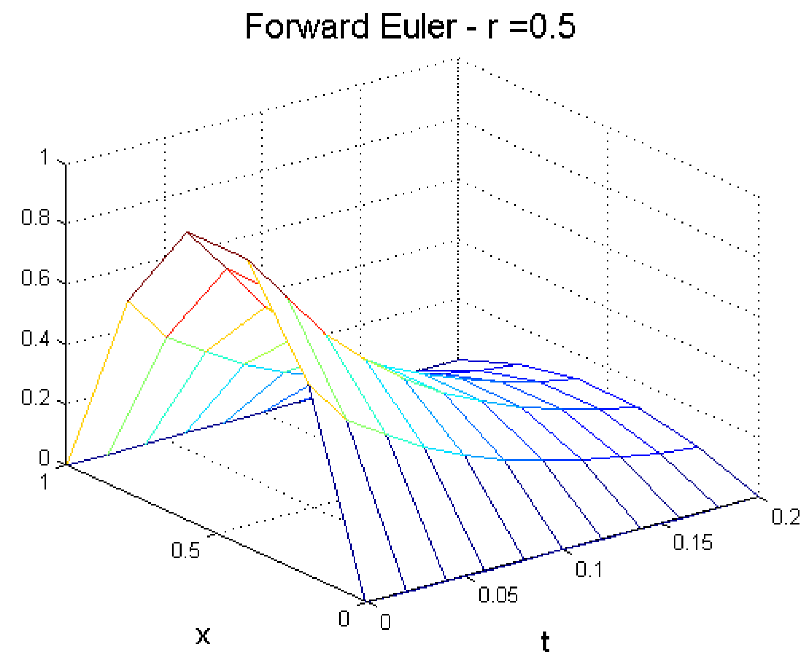
```
L=1; T=0.2; c=1;
N=5; h=L/N;
M=10; k=T/M;
r=c^2*k/h^2

x=[0:h:L]';
t=[0:k:T];
fx='4*x-4*x.^2';
g1x='0';
g2x='0';
f=inline(fx,'x');
g1=inline(g1x,'t');
g2=inline(g2x,'t');
n=length(x);
m=length(t);
u=zeros(n,m);
u(2:n-1,1)=f(x(2:n-1));
u(1,1:m)=g1(t);
u(n,1:m)=g2(t);
for j=1:m-1
    for i=2:n-1
u(i,j+1)=(1-2*r)*u(i,j) + r*(u(i+1,j)+u(i-1,j));
    end
end
 .
figure(4)
mesh(t,x,u);
a=ylabel('x');
set(a,'Fontsize',14);
a=xlabel('t');
set(a,'Fontsize',14);
a=title(['Forward Euler - r =' num2str(r)]);
set(a,'Fontsize',16);
```

$$u_t(x,t) = u_{xx}(x,t)\ , \quad 0 < x < 1\ , \quad 0 < t < 0.2$$

ICs: $u(x,0) = f(x) = 4x - 4x^2$

BCs:
$$u(0,t) = g_1(t) \equiv 0$$
$$u(1,t) = g_2(t) \equiv 0$$



Heat Conduction Equation
Explicit Finite Differences

heat_fw_2.m

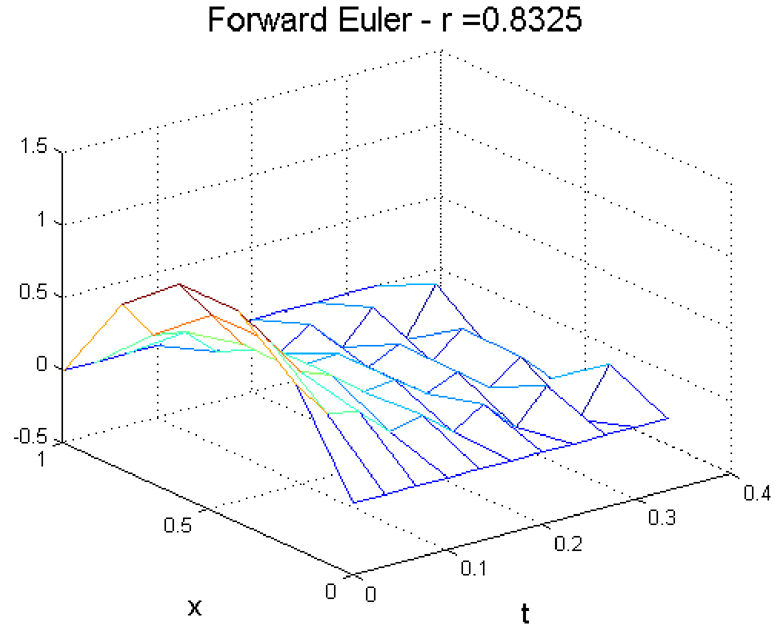
```
L=1; T=0.333; c=1;
N=5; h=L/N;
M=10; k=T/M;
r=c^2*k/h^2

x=[0:h:L]';
t=[0:k:T];
fx='4*x-4*x.^2';
g1x='0';
g2x='0';
f=inline(fx,'x');
g1=inline(g1x,'t');
g2=inline(g2x,'t');
n=length(x);
m=length(t);
u=zeros(n,m);
u(2:n-1,1)=f(x(2:n-1));
u(1,1:m)=g1(t);
u(n,1:m)=g2(t);
for j=1:m-1
    for i=2:n-1
u(i,j+1)=(1-2*r)*u(i,j) + r*(u(i+1,j)+u(i-1,j));
    end
end
 .
figure(4)
mesh(t,x,u);
a=ylabel('x');
set(a,'Fontsize',14);
a=xlabel('t');
set(a,'Fontsize',14);
a=title(['Forward Euler - r =' num2str(r)]);
set(a,'Fontsize',16);
```

$$u_t(x,t) = u_{xx}(x,t)\ ,\quad 0 < x < 1\ ,\quad 0 < t < 0.33$$

ICs: $u(x,0) = f(x) = 4x - 4x^2$

BCs:
$$\begin{aligned} u(0,t) &= g_1(t) \equiv 0 \\ u(1,t) &= g_2(t) \equiv 0 \end{aligned}$$



Parabolic PDE: Implicit Schemes

Leads to a system of equations to be solved at each time-step

B-C (Backward-Centered):
1^{st} order accurate in time,
2^{nd} order in space

Unconditionally stable

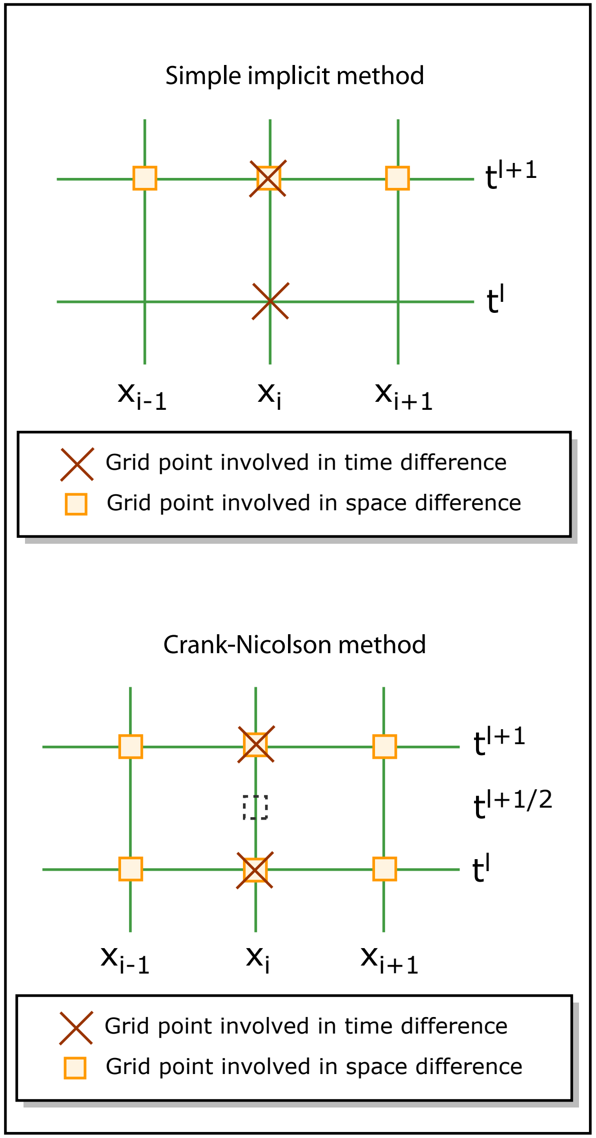


Image by MIT OpenCourseWare. After Chapra, S., and R. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 2005.

B-C:
- Backward in time
- Centered in space
- Evaluates RHS at time $t+1$ instead of time t (for the explicit scheme)

Crank-Nicolson:
2^{nd} order accurate in time,
2^{nd} order in space

Unconditionally stable

Time: centered FD, but evaluated at mid-point

2^{nd} derivative in space determined at mid-point by averaging at t and $t+1$

Parabolic PDE: Implicit Schemes
Crank-Nicolson Scheme

Equidistant Sampling

$$h = L/n$$
$$k = T/m$$

Discretization

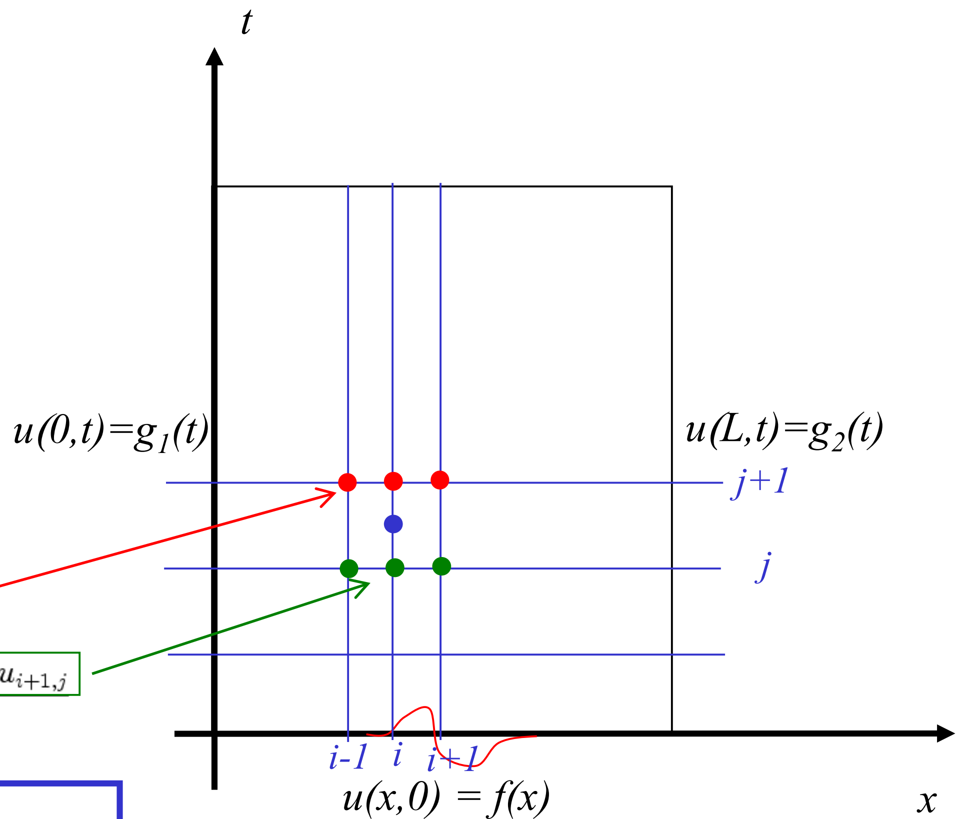
$$x_i = (i-1)h, \; i = 2, \ldots, n-1$$
$$t_j = (j-1)k, \; j = 1, \ldots, m$$

Mid-point Finite Difference in time

$$u_t\left(x, t + \frac{k}{2}\right) = \frac{u(x,t+k) - u(x,t)}{k} + O(k^2)$$

Finite Difference Equation

$$\frac{u_{i,j+1} - u_{i,j}}{k} = c^2 \frac{u_{i-1,j+1} - 2u_{i,j+1} + u_{i+1,j+1} + u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{2h^2}$$



Crank-Nicholson Implicit Scheme

$$r = \frac{c^2 k}{h^2}$$

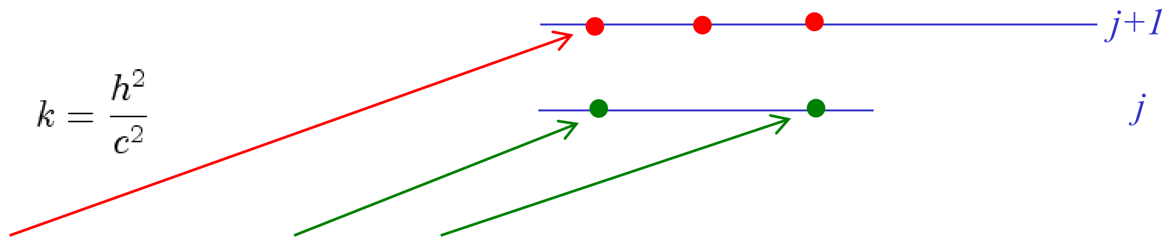
$$-r u_{i-1,j+1} + (2+2r)u_{i,j+1} - r u_{i+1,j+1} = (2-2r)u_{i,j} + r(u_{i-1,j} + u_{i+1,j})$$

Unconditionally stable
(by Von Neumann)

Parabolic PDEs: Implicit Schemes
Crank-Nicolson – special case of r = 1

$$r = \frac{c^2 k}{h^2} = 1$$

$$k = \frac{h^2}{c^2}$$



$$-u_{i-1,j+1} + 4u_{i,j+1} - u_{i+1,j+1} = u_{i-1,j} + u_{i+1,j}$$

$$\begin{bmatrix} 4 & -1 & & & & & \\ -1 & 4 & -1 & & & 0 & \\ & & \cdot & \cdot & & & \\ & & -1 & 4 & -1 & & \\ & & & \cdot & \cdot & & \\ & 0 & & -1 & 4 & -1 & \\ & & & & & -1 & 4 \end{bmatrix} \begin{Bmatrix} u_{2,j+1} \\ u_{3,j+1} \\ \cdot \\ u_{i,j+1} \\ \cdot \\ u_{n-2,j+1} \\ u_{n-1,j+1} \end{Bmatrix} = \begin{Bmatrix} g_{1,j} + u_{3,j} + g_{1,j+1} \\ u_{2,j} + u_{4,j} \\ \cdot \\ u_{i-1,j} + u_{i+1,j} \\ \cdot \\ u_{n-3,j} + u_{n-1,j} \\ u_{n-2,j} + g_{n,j} + g_{n,j+1} \end{Bmatrix}$$

Heat Flow Equation
Implicit Crank-Nicolson Scheme

heat_cn.m

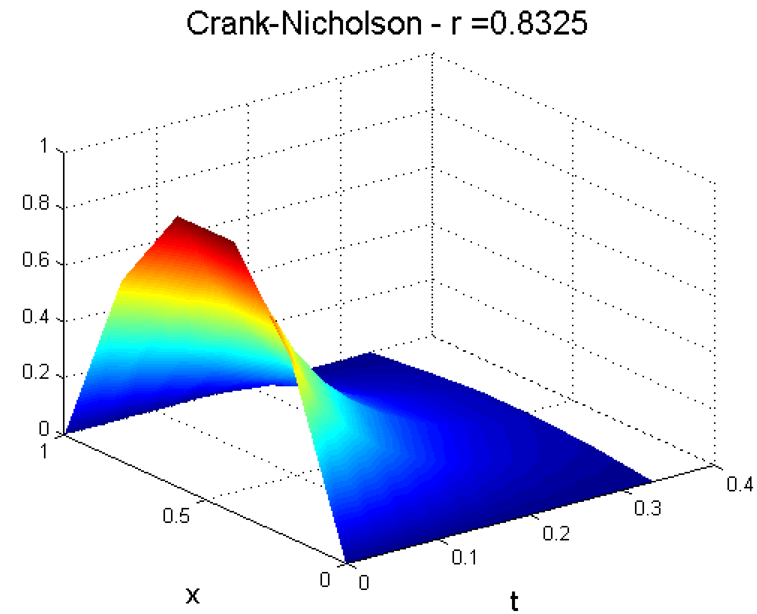
```
L=1; T=0.333; c=1;
N=5; h=L/N;
M=10;
k=T/M;
r=c^2*k/h^2

x=[0:h:L]';
t=[0:k:T];
fx='4*x-4*x.^2';
g1x='0';
g2x='0';
f=inline(fx,'x');
g1=inline(g1x,'t');
g2=inline(g2x,'t');
n=length(x); m=length(t); u=zeros(n,m);
u(2:n-1,1)=f(x(2:n-1));
u(1,1:m)=g1(t); u(n,1:m)=g2(t);
% set up Crank-Nicholson coef matrix
d=(2+2*r)*ones(n-2,1);
b=-r*ones(n-2,1);
c=b;
% LU factorization
[alf,bet]=lu_tri(d,b,c);
for j=1:m-1
    rhs=r*(u(1:n-2,j)+u(3:n,j)) +(2-2*r)*u(2:n-1,j);
    rhs(1) = rhs(1)+r*u(1,j+1);
    rhs(n-2)=rhs(n-2)+r*u(n,j+1);
% Forward substitution
    z=forw_tri(rhs,bet);
% Back substitution
    y_b=back_tri(z,alf,c);
    for i=2:n-1
      u(i,j+1)=y_b(i-1);
    end
end
```

$$u_t(x,t) = u_{xx}(x,t)\ ,\quad 0 < x < 1\ ,\quad 0 < t < 0.0$$

ICs: $u(x,0) = f(x) = 4x - 4x^2$

BCs:
$$\begin{aligned} u(0,t) &= g_1(t) \equiv 0 \\ u(1,t) &= g_2(t) \equiv 0 \end{aligned}$$



Heat Flow Equation
Implicit Crank-Nicolson Scheme

heat_cn_sin.m

```
L=1; T=0.1; c=1;
N=10; h=L/N;
M=10;
k=T/M;
r=c^2*k/h^2

x=[0:h:L]';
t=[0:k:T];
fx='sin(pi*x)+sin(3*pi*x)';
g1x='0';
g2x='0';
f=inline(fx,'x');
g1=inline(g1x,'t');
g2=inline(g2x,'t');
n=length(x); m=length(t); u=zeros(n,m);
u(2:n-1,1)=f(x(2:n-1));
u(1,1:m)=g1(t); u(n,1:m)=g2(t);
% set up Crank-Nicholson coef matrix
d=(2+2*r)*ones(n-2,1);
b=-r*ones(n-2,1);
c=b;
% LU factorization
[alf,bet]=lu_tri(d,b,c);
for j=1:m-1
    rhs=r*(u(1:n-2,j)+u(3:n,j)) +(2-2*r)*u(2:n-1,j);
    rhs(1) = rhs(1)+r*u(1,j+1);
    rhs(n-2)=rhs(n-2)+r*u(n,j+1);
% Forward substitution
    z=forw_tri(rhs,bet);
% Back substitution
    y_b=back_tri(z,alf,c);
    for i=2:n-1
      u(i,j+1)=y_b(i-1);
    end
end
```

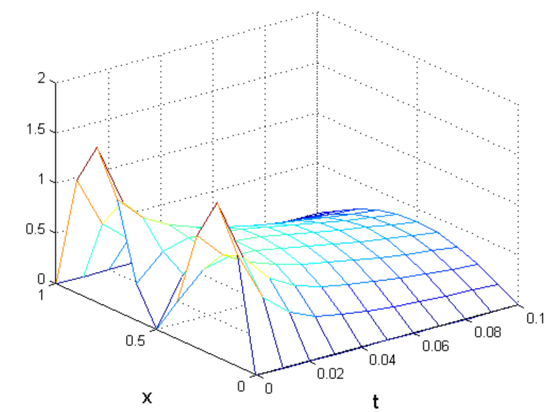
Initial Condition

$$f(x) = \sin \pi x + \sin 3\pi x$$

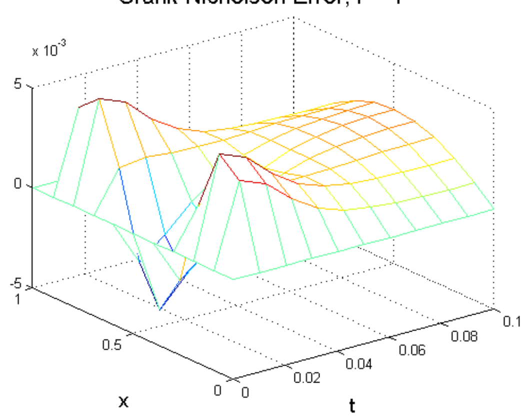
Analytical Solution

$$u(x,t) = e^{-\pi^2 t} \sin \pi x + e^{-9\pi^2 t} \sin 3\pi x$$

Crank-Nicholson - r =1



Crank-Nicholson Error, r = 1



Parabolic PDEs: Two spatial dimensions

- Example: Heat conduction equation/unsteady diffusive (e.g. negligible flow, no convection)

$$\frac{\partial T}{\partial t} = c^2 \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right), \qquad (0 \le t < \infty, 0 < x < L_x, 0 < y < L_y)$$

- Standard explicit and implicit schemes ($t = n\Delta t,\ x = i\Delta x,\ y = j\Delta y$)
  - Explicit: $$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = c^2 \frac{T_{i-1,j}^{n} - 2T_{i,j}^{n} + T_{i+1,j}^{n}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n} - 2T_{i,j}^{n} + T_{i,j+1}^{n}}{\Delta y^2} \qquad \left(O(\Delta t), O(\Delta x^2 + \Delta y^2)\right)$$
    - Stringent stability criterion:

$$\Delta t \le \frac{1}{8} \frac{\Delta x^2 + \Delta y^2}{c^2}$$

For uniform grid: $r = \dfrac{\Delta t\, c^2}{\Delta x^2} \le \dfrac{1}{4}$

  - Implicit: $$\frac{T_{i,j}^{n+1} - T_{i,j}^{n}}{\Delta t} = c^2 \frac{T_{i-1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i+1,j}^{n+1}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j+1}^{n+1}}{\Delta y^2} \qquad \left(O(\Delta t), O(\Delta x^2 + \Delta y^2)\right)$$
  - Crank-Nicolson Implicit (for $\Delta x = \Delta y$) $\qquad \left(O(\Delta t^2), O(\Delta x^2)\right)$

$$(1+2r)T_{i,j}^{n+1} - (1-2r)T_{i,j}^{n} = \frac{r}{2}\left(T_{i-1,j}^{n+1} + T_{i+1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1}\right) + \frac{r}{2}\left(T_{i-1,j}^{n} + T_{i+1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n}\right)$$

- Centered in time over dt = Sum explicit and implicit RHSs (given above), divided by two

Parabolic PDEs: Two spatial dimensions

- Crank-Nicolson Implicit (for $\Delta x = \Delta y$):

$$(1+2r)T_{i,j}^{n+1} - (1-2r)T_{i,j}^{n} = \frac{r}{2}\left(T_{i-1,j}^{n+1} + T_{i+1,j}^{n+1} + T_{i,j+1}^{n+1} + T_{i,j-1}^{n+1}\right) + \frac{r}{2}\left(T_{i-1,j}^{n} + T_{i+1,j}^{n} + T_{i,j+1}^{n} + T_{i,j-1}^{n}\right)$$

  - Five unknowns at the (n+1) time => penta-diagonal
  - Either elimination procedure or iterative scheme (Jacobi/Gauss-Seidel/SOR)
  - but not always efficient

- Alternating-Direction Implicit (ADI) schemes
  - Provides a mean for solving parabolic PDEs with tri-diagonal matrices
  - In 2D: each time increment is executed in two half steps: each step is conditionally stable, but "combination of two half-steps" is unconditionally stable (similar to Crank-Nicolson behavior)
  - It is one but a group of schemes called "splitting methods"
  - Extended to 3D (time increment divided in 3): varied stability properties

Parabolic PDEs: Two spatial dimensions
ADI scheme (Two Half steps in time)

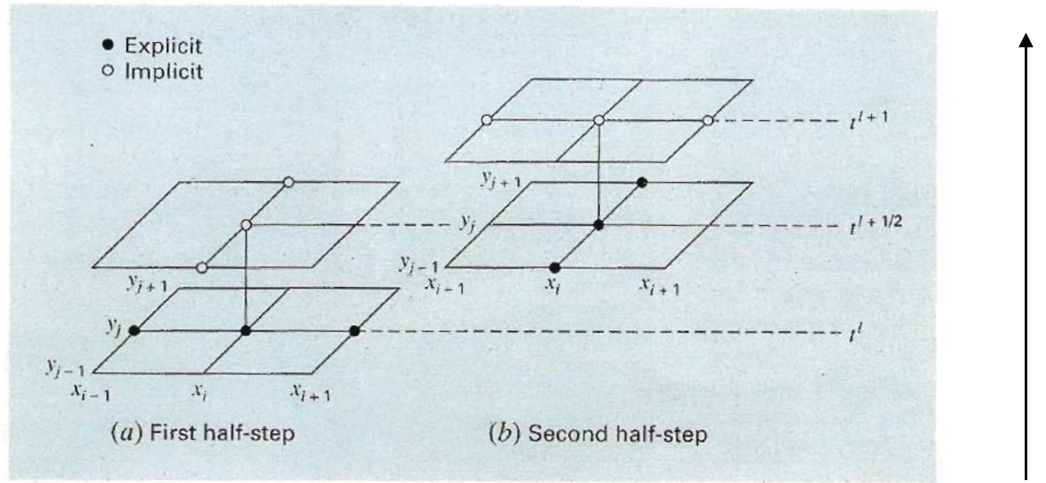


FIGURE 30.10
The two half-steps used in implementing the alternating-direction implicit scheme for solving parabolic equations in two spatial dimensions.

Source: Chapra, S., and R. Canale. *Numerical Methods for Engineers*. McGraw-Hill, 2005.

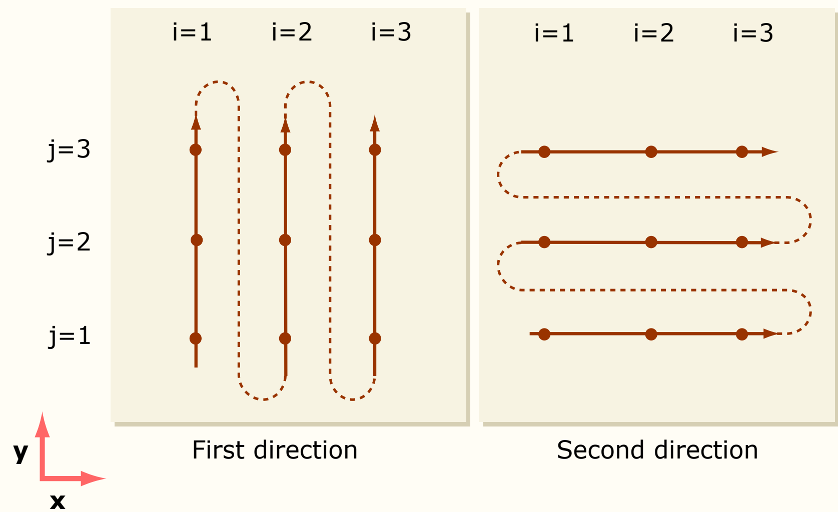
1) From time n to $n+1/2$: Approximation of 2nd order x derivative is explicit, while the y derivative is implicit. Hence, tri-diagonal matrix to be solved:

$$\frac{T_{i,j}^{n+1/2} - T_{i,j}^{n}}{\Delta t/2} = c^2 \frac{T_{i-1,j}^{n} - 2T_{i,j}^{n} + T_{i+1,j}^{n}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i,j+1}^{n+1/2}}{\Delta y^2} \qquad \left(O(\Delta x^2 + \Delta y^2)\right)$$

2) From time $n+1/2$ to $n+1$: Approximation of 2nd order x derivative is implicit, while the y derivative is explicit. Another tri-diagonal matrix to be solved:

$$\frac{T_{i,j}^{n+1} - T_{i,j}^{n+1/2}}{\Delta t/2} = c^2 \frac{T_{i-1,j}^{n+1} - 2T_{i,j}^{n+1} + T_{i+1,j}^{n+1}}{\Delta x^2} + c^2 \frac{T_{i,j-1}^{n+1/2} - 2T_{i,j}^{n+1/2} + T_{i,j+1}^{n+1/2}}{\Delta y^2} \qquad \left(O(\Delta x^2 + \Delta y^2)\right)$$

Parabolic PDEs: Two spatial dimensions
ADI scheme (Two Half steps in time)



The ADI method applied along the y direction and x direction. This method only yields tridiagonal equations if applied along the implicit dimension.

For $\Delta x = \Delta y$:

1) From time n to $n+1/2$:

$$-rT_{i,j-1}^{n+1/2} + 2(1+r)T_{i,j}^{n+1/2} - rT_{i,j+1}^{n+1/2} = rT_{i-1,j}^{n} + 2(1-r)T_{i,j}^{n} + rT_{i+1,j}^{n}$$

2) From time $n+1/2$ to $n+1$:

$$-rT_{i-1,j}^{n+1} + 2(1+r)T_{i,j}^{n+1} - rT_{i+1,j}^{n+1} = rT_{i,j-1}^{n+1/2} + 2(1-r)T_{i,j}^{n+1/2} + rT_{i,j+1}^{n+1/2}$$