Computational Fluid Dynamics: Lecture 7 (ME EN 6720)

Prof. Rob Stoll
Department of Mechanical Engineering
University of Utah

Spring 2014

Parabolic Equations

Parabolic Equations:

- Equations of the form: $$\frac{\partial u}{\partial t} = \alpha \nabla^2 u$$
- This equation is representative of diffusion type problems
- Examples of problems dominated by diffusion:
  - -Heat transfer in a solid
  - -mass transport through porous media
  - -Couette flow
  - -Poiseuille flow
  - -laminar-fully developed flow phenomena
- Lets **examine the 1D version of this equation** and look at examples of techniques:

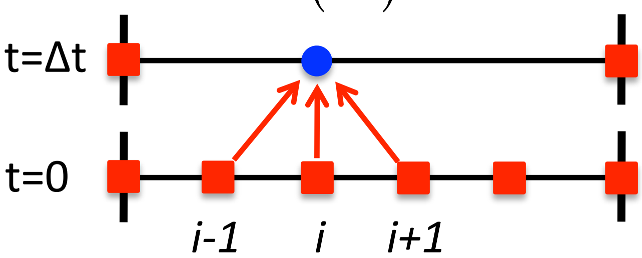
$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Explicit Schemes

- **Simple Explicit Method**:

  -using a forward in time and centered in space (FTCS) scheme we get the following algebraic equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{1}{(\Delta x)^2}\left[u_{i+1}^n - 2u_i^n + u_{i-1}^n\right] \quad \text{or} \quad u_i^{n+1} = u_i^n + \frac{\alpha \Delta t}{(\Delta x)^2}\left[u_{i+1}^n - 2u_i^n + u_{i-1}^n\right]$$



The scheme has:

- T.E. ~ ***O***[Δt,Δx²]
- stability: conditioned on $d = \frac{\alpha \Delta t}{(\Delta x)^2} \leq \frac{1}{2}$

diffusion number

  - in general we have N-2 equations with N-2 unknowns at each time step

  - for this explicit method the equations are independent. i.e. we can calculate point *i* at Δt without points *i-1* or *i-2*, etc.

- **How could we improve on this** (better T.E. and/or stability)?

-Use a centered in time discretization for $\partial u / \partial t$ ➔ $$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \alpha \frac{1}{(\Delta x)^2}\left[u_{i+1}^n - 2u_i^n + u_{i-1}^n\right]$$

-**T.E.** ~ ***O***[Δt²,Δx²]

-**stability:** we can show that this scheme is unconditionally unstable

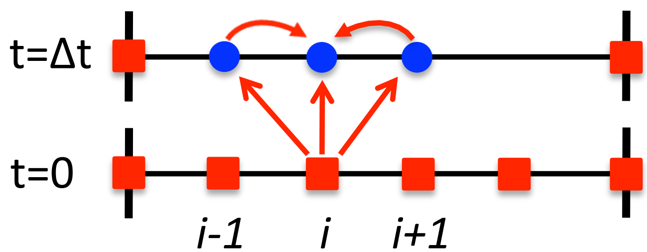
-**This scheme is basically useless**!

Implicit Schemes

- **Simple Implicit Method**:

-using forward in time and centered in space:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \alpha \frac{1}{(\Delta x)^2}\left[u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right]$$



The scheme has:

- T.E. ~ ***O***[Δt,Δx^2]
- stability: unconditionally stable

-here we end up with:

- N-2 unknowns
- N-2 equations

Equations are coupled

-we have to solve a systems of linear algebraic equations at each time step

Implicit Schemes

- **how can we improve our implicit scheme?**

  -average our two schemes (simple explicit and implicit)

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\alpha\left[\theta\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{(\Delta x)^2}+(1-\theta)\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{(\Delta x)^2}\right] \quad \text{with} \quad 0\le\theta\le1$$

  -for:

  - θ=0 ➔ simple explicit method T.E.~ ***O***[Δt,Δx^2] and stability: d≤½
  - θ=1 ➔ simple implicit method T.E.~ ***O***[Δt,Δx^2] and stability: unconditional
  - θ=½ ➔ **Crank-Nicolson** scheme.

  -Crank-Nicolson is one of the most popular ways to solve parabolic equations.

System of Equations

- **<u>In general for all the methods we have a system of Equations</u>**
  - What does this system look like?

-simple implicit: $$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\alpha\frac{1}{(\Delta x)^2}\left[u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}\right]$$

-separating known and unknown terms: $$-\frac{\alpha\Delta t}{(\Delta x)^2}u_{i-1}^{n+1}+\left(1+\frac{2\alpha\Delta t}{(\Delta x)^2}\right)u_i^{n+1}-\frac{\alpha\Delta t}{(\Delta x)^2}u_{i+1}^{n+1}=u_i^n$$

or $\boxed{b_i u_{i-1}^{n+1}+d_i u_i^{n+1}+a_i u_{i+1}^{n+1}=u_i^n}$ where $a_i=b_i=\dfrac{-\alpha\Delta t}{(\Delta x)^2};\ c_i=u_i^n;\ d_i=\left(1+\dfrac{2\alpha\Delta t}{(\Delta x)^2}\right)$

-we can write this in matrix form: $\mathbf{A}\vec{x}=\vec{b}$

$$\begin{array}{cc} d_1 & a_1 \;\cdots\cdots\cdots\cdots \\ b_2 & \underbrace{\begin{bmatrix} d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ b_3 & d_3 & a_3 & 0 & \cdots & 0 \\ 0 & b_4 & d_4 & a_4 & 0 & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & b_{N-2} & d_{N-2} & a_{N-2} \\ 0 & \cdots & \cdots & \cdots & b_{N-1} & d_{N-1} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} c_2^* \\ c_3^* \\ \\ \vdots \\ c_{N-2}^* \\ c_{N-1}^* \end{bmatrix}}_{\vec{b}} \end{array}$$

- we get c_2^* and c_{N-1}^* from our BCs:

$$c_2^*=c_2-b_2u_1^{n+1}$$
$$c_{N-1}^*=c_{N-1}-a_{N-1}u_N^{n+1}$$
$$c_i^*=c_i$$

Implicit 1-D Heat Equation

Implicit Solutions to the 1-D Heat Equation:

- we need to solve: $\dfrac{u_i^{n+1} - u_i^n}{\Delta t} = \dfrac{\alpha}{(\Delta x)^2}\left[u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}\right]$
- we can rearrange this into:

$$b_i u_{i-1}^{n+1} + d_i u_i^{n+1} + a_i u_{i+1}^{n+1} = c_i \quad \text{where} \quad a_i = b_i = -\frac{\alpha \Delta t}{(\Delta x)^2};\ c_i = u_i;\ d_i = 1 + \frac{2\alpha \Delta t}{(\Delta x)^2}$$

- This is a system of linear (dependent) equations

$$\begin{array}{cl} & d_1 \quad a_1 \quad \cdots \\ \begin{matrix} b_2 \\ \vdots \end{matrix} & \underbrace{\begin{bmatrix} d_2 & a_2 & 0 & \cdots & \cdots & 0 \\ b_3 & d_3 & a_3 & 0 & \cdots & 0 \\ 0 & b_4 & d_4 & a_4 & 0 & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \\ & & & b_{N-2} & d_{N-2} & a_{N-2} \\ 0 & \cdots & \cdots & \cdots & b_{N-1} & d_{N-1} \end{bmatrix}}_{\mathbf{A}} \end{array} \underbrace{\begin{bmatrix} u_2^{n+1} \\ u_3^{n+1} \\ \vdots \\ u_{N-2}^{n+1} \\ u_{N-1}^{n+1} \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} c_2^* \\ c_3^* \\ \vdots \\ c_{N-2}^* \\ c_{N-1}^* \end{bmatrix}}_{\vec{b}}$$

- we get c_2^* and c_{N-1}^* from our BCs:

$$c_2^* = c_2 - b_2 u_1^{n+1}$$
$$c_{N-1}^* = c_{N-1} - a_{N-1} u_N^{n+1}$$
$$c_i^* = c_i$$

Solving the System of Equations

- In general, solving a system of linear equations requires ~***O***(N³) operations (Gaussian elimination)
- We have a **tridiagnol system** of equations

  -This is a special case which can be solved **very efficiently**
- A few notes about the Matrix **A**:

  -it is diagonally dominant: $|d_i| \geq |b_i| + |a_i|$

  -it has positive elements in the diagonal and negative elements in the off diagonals

  -for these conditions we have a M-matrix and a solution will always exist.
- A good way to solve this type of system is the **Thomas Algorithm**

  (i) First we can transform the matrix into an upper-triangular matrix by:

$$\left.\begin{aligned} d_i' &= d_i - {b_i}/{d_{i-1}'}\, a_{i-1} \\ c_i' &= c_i^* - {b_i}/{d_{i-1}'}\, c_{i-1}' \end{aligned}\right\} \text{for } i = 3 \to N-1$$

  (ii) Now we just need to do a backwards substitution to find our values:

$$\left.\begin{aligned} u_{N-1}^{n+1} &= {c_{N-1}'}/{d_{N-1}'} \\ u_i^{n+1} &= {(c_i' - a_i u_{i+1}^{n+1})}/{d_i'} \end{aligned}\right\} \text{for } i = N-2, N-3 \to 2$$

2-D Heat Equation

- Computational cost of the Thomas Algorithm is ~***O***(N)
- see Ferziger page 95 for more details on the Thomas Algorithm

- **2-D Heat Equation**

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}=\alpha\left[\frac{u_{i+1,j}^{n}-2u_{i,j}^{n}+u_{i-1,j}^{n}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{(\Delta y)^2}\right]$$

•T.E. ~***O***[Δt, $(\Delta x)^2$, $(\Delta x)^2$]

•Stability: $$\left[\underbrace{\frac{\alpha\Delta t}{(\Delta x)^2}}_{d_x}+\underbrace{\frac{\alpha\Delta t}{(\Delta y)^2}}_{d_y}\right]\leq \tfrac{1}{2}$$

-if $\Delta x=\Delta y \Rightarrow d_x=d_y \Rightarrow d\leq \tfrac{1}{4}$

-explicit method in **2-D is twice as restrictive as in 1-D**

Implicit 2-D Heat Equation

- **Implicit Scheme for 2-D heat:** (simple implicit)

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n}}{\Delta t}=\alpha\left[\frac{u_{i+1,j}^{n+1}-2u_{i,j}^{n+1}+u_{i-1,j}^{n+1}}{(\Delta x)^2}+\frac{u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}}{(\Delta y)^2}\right]$$

- We can rearrange this into: (using same idea as for 1-D)

$$a_{i,j}u_{i+1,j}^{n+1}+b_{i,j}u_{i-1,j}^{n+1}+c_{i,j}u_{i,j}^{n+1}+d_{i,j}u_{i,j-1}^{n+1}+e_{i,j}u_{i,j+1}^{n+1}=f_{i,j}$$

-if we drop the indexes and note that $b_{i,j}=d_{i,j}$ and $a_{i,j}=e_{i,j}$ (for a uniform grid) we get a Matrix looks as follows for the given 2D mesh:

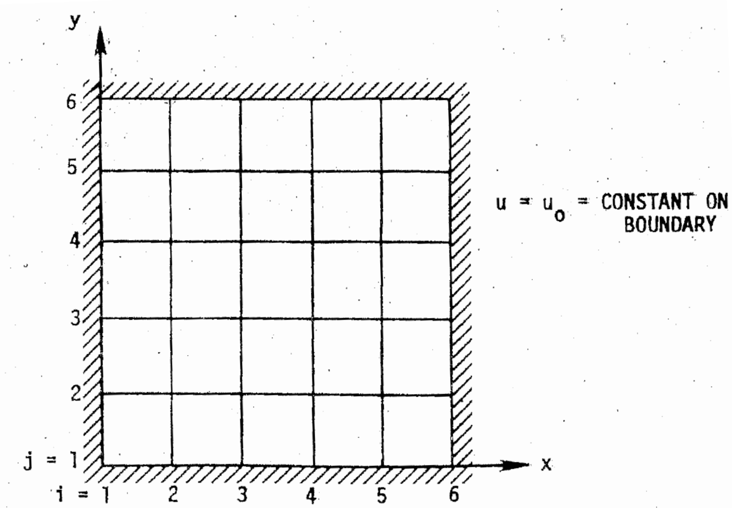
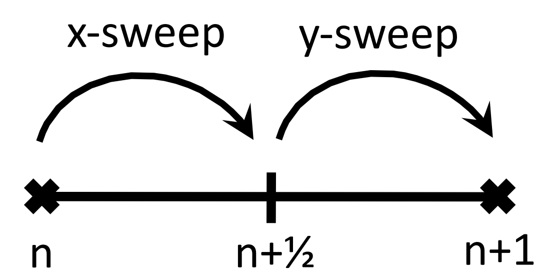


Figure 4.18 Two-dimensional computational mesh.

$$\begin{bmatrix}
c & b & 0 & 0 & a & 0 & & & & & & & & & & 0\\
b & c & b & & & a & & & & & & & & & & \\
0 & b & c & b & & & a & & & & & & & & & \\
0 & & b & c & 0 & & & a & & & & & & & & \\
a & & & 0 & c & b & & & a & & & & & & & \\
0 & a & & & b & c & b & & & a & & & & & & \\
& & a & & & b & c & b & & & a & & & & & \\
& & & a & & & b & c & 0 & & & a & & & & \\
& & & & a & & & 0 & c & b & & & a & & & \\
& & & & & a & & & b & c & b & & & a & & \\
& & & & & & a & & & b & c & b & & & a & 0\\
& & & & & & & a & & & b & c & 0 & & & a\\
& & & & & & & & a & & & 0 & c & b & & 0\\
& & & & & & & & & a & & & b & c & b & 0\\
& & & & & & & & & & a & & & b & c & b\\
0 & & & & & & & & & & 0 & a & 0 & 0 & b & c
\end{bmatrix}
\begin{bmatrix}
u_{2,2}^{n+1}\\ u_{3,2}^{n+1}\\ u_{4,2}^{n+1}\\ u_{5,2}^{n+1}\\ u_{2,3}^{n+1}\\ u_{3,3}^{n+1}\\ u_{4,3}^{n+1}\\ u_{5,3}^{n+1}\\ u_{2,4}^{n+1}\\ u_{3,4}^{n+1}\\ u_{4,4}^{n+1}\\ u_{5,4}^{n+1}\\ u_{2,5}^{n+1}\\ u_{3,5}^{n+1}\\ u_{4,5}^{n+1}\\ u_{5,5}^{n+1}
\end{bmatrix}
=
\begin{bmatrix}
d'''_{2,2}\\ d'_{3,2}\\ d'_{4,2}\\ d'''_{5,2}\\ d''_{2,3}\\ d_{3,3}\\ d_{4,3}\\ d''_{5,3}\\ d''_{2,4}\\ d_{3,4}\\ d_{4,4}\\ d''_{5,4}\\ d'''_{2,5}\\ d'_{3,5}\\ d'_{4,5}\\ d'''_{5,5}
\end{bmatrix}$$

Implicit 2-D Heat Equation

- The matrix for the 2-D heat equation (with simple implicit) is a pentadiagonal matrix (in 3-D we would have even more diagonals).
- As we add diagonals our cost (computationally) goes up considerably
- We need an alternative method



<u>Alternating Direction Implicit</u> (ADI): Ferziger pg 105-107

- Idea: breakup the problem into different directions:

\- <u>x-sweep</u>:

$$\frac{u_{i,j}^{n+1/2}-u_{i,j}^{n}}{\left(\Delta t/2\right)}=\alpha\left[\frac{u_{i+1,j}^{n+1/2}-2u_{i,j}^{n+1/2}+u_{i-1,j}^{n+1/2}}{\left(\Delta x\right)^2}+\frac{u_{i,j+1}^{n}-2u_{i,j}^{n}+u_{i,j-1}^{n}}{\left(\Delta y\right)^2}\right]$$

\- <u>y-sweep</u>:

$$\frac{u_{i,j}^{n+1}-u_{i,j}^{n+1/2}}{\left(\Delta t/2\right)}=\alpha\left[\frac{u_{i+1,j}^{n+1/2}-2u_{i,j}^{n+1/2}+u_{i-1,j}^{n+1/2}}{\left(\Delta x\right)^2}+\frac{u_{i,j+1}^{n+1}-2u_{i,j}^{n+1}+u_{i,j-1}^{n+1}}{\left(\Delta y\right)^2}\right]$$

- This gives us two tridiagnol systems:

$$\left.\begin{aligned}
&* \quad -d_1u_{i-1,j}^{n+1/2}+\left(1+2d_1\right)u_{i,j}^{n+1/2}-d_1u_{i+1,j}^{n+1/2}=d_2u_{i,j+1}^{n}+\left(1-2d_2\right)u_{i,j}^{n}+d_2u_{i,j-1}^{n}\\
&* \quad -d_2u_{i,j-1}^{n+1}+\left(1+2d_2\right)u_{i,j}^{n+1}-d_2u_{i,j+1}^{n+1}=d_1u_{i-1,j}^{n+1/2}+\left(1-2d_1\right)u_{i,j}^{n+1/2}+d_1u_{i-1,j}^{n+1/2}
\end{aligned}\right\}\quad
\begin{aligned}
d_1&=\tfrac{1}{2}d_x\\
d_2&=\tfrac{1}{2}d_y
\end{aligned}$$