Additional Topics in Numerical Solutions of Parabolic Equations

Larry Caretto

Mechanical Engineering 501AB

Seminar in Engineering Analysis

March 23, 2009

Outline

- Complete discussion from last class
  - Fully implicit and DuFort-Frankel methods
- Review numerical diffusion solutions
  - Explicit and Crank-Nicholson methods
  - Thomas algorithm
- General properties of numerical algorithms for PDEs
- Numerical diffusion equation solutions in two or more space dimensions

Fully Implicit Method

- Discretize diffusion equation at t_{n+1}

$$\left.\frac{\partial u}{\partial t}\right|_i^{n+1} = \frac{u_i^{n+1} - u_i^n}{\Delta t} + O(\Delta t) \quad and \quad \left.\frac{\partial^2 u}{\partial x^2}\right|_i^{n+1} = \frac{u_{i+1}^{n+1} + u_{i-1}^{n+1} - 2u_i^{n+1}}{(\Delta x)^2} + O[(\Delta x)^2]$$

$$\left.\frac{\partial u}{\partial t}\right|_i^{n+1} - \alpha \left.\frac{\partial^2 T}{\partial x^2}\right|_i^{n+1} = \frac{u_i^{n+1} - u_i^n}{\Delta t} - \alpha \frac{u_{i+1}^{n+1} + u_{i-1}^{n+1} - 2u_i^{n+1}}{(\Delta x)^2} + O[(\Delta t), (\Delta x)^2] = 0$$

$$- f u_{i-1}^{n+1} + (1 + 2f) u_i^{n+1} - f u_{i+1}^{n+1} = u_i^n$$

- Tridiagonal system of equations
- Almost same work as CN and no spurious oscillations, but less accuracy

$u_0 = u_{10} = 0$ for all times.

Fully Implicit (BTCS) Results

- Same as CN inputs: α = 1, L = 1, Δx = 0.01, Δt = 0.0005, f = $\alpha\Delta t/(\Delta x)^2$ = 5

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0	x = .01	x = .02	x = .03	x = .04
	t = 0	1000	1000	1000	1000	1000
n = 0	t = 0+	0	1000	1000	1000	1000
n = 1	t = 0.0005	0	358.26	588.17	735.71	830.39
n = 2	t = 0.001	0	218.22	408.43	562.69	682.35
n = 3	t = 0.0015	0	166.26	322.13	460.74	578.96
n = 4	t = 0.002	0	139.05	272.65	396.35	507.18
n = 5	t = 0.0025	0	121.84	240.25	352.17	455.26
n = 6	t = 0.003	0	109.75	217.08	319.77	415.99

Fully Implicit Results II

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0	x = .01	x = .02	x = .03	x = .04
n = 7	t = 0.0035	0	100.65	199.49	294.81	385.13
n = 8	t = 0.004	0	93.50	185.57	274.85	360.14
n = 9	t = 0.0045	0	87.68	174.19	258.43	339.38
n = 10	t = 0.005	0	82.82	164.67	244.62	321.81
n = 11	t = 0.0055	0	78.69	156.56	232.81	306.69
n = 12	t = 0.006	0	75.13	149.54	222.55	293.50
n = 13	t = 0.0065	0	72.00	143.38	213.53	281.87
n = 14	t = 0.007	0	69.24	137.93	205.52	271.52
n = 15	t = 0.0075	0	66.77	133.05	198.35	262.22
n = 16	t = 0.008	0	64.55	128.66	191.88	253.82
n = 17	t = 0.0085	0	62.54	124.67	186.01	246.17

Fully Implicit Results III

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0	x = .01	x = .02	x = .03	x = .04
n = 18	t = 0.009	0	60.70	121.03	180.64	239.17
n = 19	t = 0.0095	0	59.02	117.70	175.71	232.74
n = 20	t = 0.01	0	57.47	114.62	171.16	226.79
n = 21	t = 0.0105	0	56.03	111.78	166.95	221.28
n = 22	t = 0.011	0	54.70	109.13	163.04	216.16
n = 23	t = 0.0115	0	53.46	106.67	159.38	211.37
n = 24	t = 0.012	0	52.30	104.36	155.96	206.88
n = 25	t = 0.0125	0	51.21	102.20	152.76	202.67
Exact	t = 0.0125	0	50.43	100.66	150.48	199.72
Error	t = 0.0125	0	0.779	1.542	2.273	2.956

Compare Crank Nicholson

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0	x = .01	x = .02	x = .03	x = .04
n = 18	t = 0.009	0	60.65	117	177.71	234.21
n = 19	t = 0.0095	0	56.86	116.5	171.59	228.43
n = 20	t = 0.01	0	57.1	111.53	168.52	222.53
n = 21	t = 0.0105	0	54.43	110.47	163.53	217.57
n = 22	t = 0.011	0	54.19	106.68	160.64	212.45
n = 23	t = 0.0115	0	52.22	105.35	156.49	208.11
n = 24	t = 0.012	0	51.73	102.36	153.78	203.64
n = 25	t = 0.0125	0	50.21	100.93	150.27	199.78
Exact	t = 0.0125	0	50.43	100.66	150.48	199.72
Error	t = 0.0125	0	0.216	0.272	0.212	0.061

California State University Northridge 7

Richardson/Leapfrog

- Use two time step central differences

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

- Result is explicit with second order accuracy in time

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

- However result is unstable for any f and cannot be used

California State University Northridge 8

DuFort Frankel

- Modification of Richardson method to provide stability
- Replace $2u_i^n$ in second derivative by average at time steps n+1 and n-1
- Introduces another $O[(\Delta t)^2]$ error

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

$$2u_i^n = u_i^{n+1} + u_i^{n-1} + O[(\Delta t)^2]$$

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

California State University Northridge 9

DuFort Frankel

- Rearrange and introduce $f = \alpha\Delta t/(\Delta x)^2$

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

$$(1+2f)u_i^{n+1} = u_i^{n-1}(1-2f) + 2f\left(u_{i+1}^n + u_{i-1}^n\right)$$

- Result is explicit for values at time n+1
- Explicit start required to get first set of values at time n-1
- Can show that this is unconditionally stable

California State University Northridge 10

Truncation Error

- Truncation error for DuFort-Frankel
  - Derivation in appendix to notes on numerical solutions for PDEs
- Set $f^2 = 1/12$ ($f = 0.288675\ldots$) for minimum truncation error

$$TE = \left(\frac{1}{12f} - f\right)\left.\frac{\partial^2 T}{\partial t^2}\right|_i^n \Delta t - \sum_{k=1}^{\infty} \frac{(\Delta t)^{2k}}{(2k+1)!}\left.\frac{\partial^{2k+1} T}{\partial t^{2k+1}}\right|_i^n - 2f\sum_{k=2}^{\infty} \frac{(\Delta t)^{2k-1}}{(2k)!}\left.\frac{\partial^{2k} T}{\partial t^{2k}}\right|_i^n + 2\sum_{k=3}^{\infty} \frac{(\Delta t)^{k-1}}{f^{k-1}(2k)!}\left.\frac{\partial^k T}{\partial t^k}\right|_i^n$$

California State University Northridge 11

Review Diffusion Algorithms

u^{n+1}_{i-1} ●-----● u_i^{n+1} -----● u^{n+1}_{i+1} • New time

u^n_{i-1} ●-----● u_i^n -----● u^n_{i+1} • Old time

- Explicit $u_i^{n+1} = f\left(u_{i+1}^n + u_{i-1}^n\right) + (1-2f)u_i^n$
- Fully implicit $-fu_{i-1}^{n+1} + (1+2f)u_i^{n+1} - fu_{i+1}^{n+1} = u_i^n$
- Crank-Nicholson $-fu_{i-1}^{n+1} + 2(1+f)u_i^{n+1} - fu_{i+1}^{n+1} = f\left[u_{i+1}^n + u_{i-1}^n\right] + 2(1-f)u_i^n = R_i$
- DuFort-Frankel $(1+2f)u_i^{n+1} = u_i^{n-1}(1-2f) + 2f\left(u_{i+1}^n + u_{i-1}^n\right)$

California State University Northridge 12

Review Implicit Equations

- Crank Nicholson and fully implicit have three values at new time step
$$-fu_{i-1}^{n+1} + 2(1+f)u_i^{n+1} - fu_{i+1}^{n+1} = f\left[u_{i+1}^n + u_{i-1}^n\right] + 2(1-f)u_i^n = R_i$$
$$-fu_{i-1}^{n+1} + (1+2f)u_i^{n+1} - fu_{i+1}^{n+1} = u_i^n$$
- Tridiagonal system of equations easily solved by special application of Gauss elimination called Thomas algorithm
  - General tridiagonal form is
$$A_i u_{i-1}^{n+1} + B_i u_i^{n+1} + C_i u_{i+1}^{n+1} = D_i$$

13

Review Thomas Algorithm

- Solution procedure for tridiagonal equations $A_i x_{i-1} + B_i x_i + C_i x_{i+1} = D_i$
- Special application of conventional Gaussian elimination
- Equations for E_i and F_i, i = 0, N
- Get $x_N = F_N$
- Get $x_i = F_i + E_i x_{i+1}$ for i = N – 1 to 1

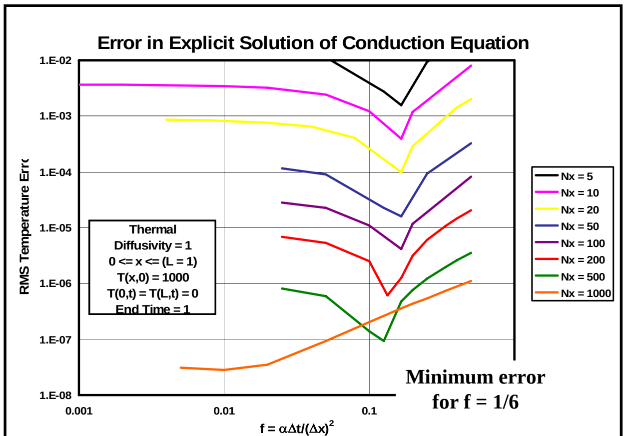
$$\begin{bmatrix} B_0 & C_0 & 0 & 0 & \cdots & 0 & 0 \\ A_1 & B_1 & C_1 & 0 & \cdots & 0 & 0 \\ 0 & A_2 & B_2 & C_2 & \cdots & 0 & 0 \\ 0 & 0 & A_3 & B_3 & & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & B_{N-1} & C_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & A_N & B_N \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ \vdots \\ D_{N-1} \\ D_N \end{bmatrix}$$

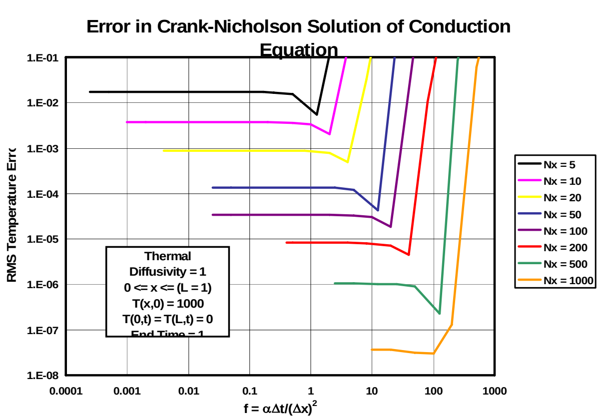
14

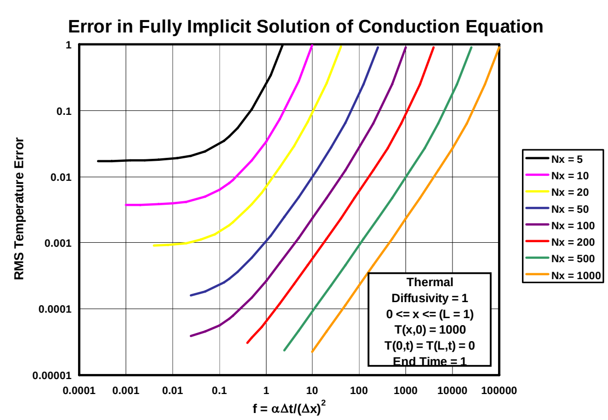
Review Stability Limits

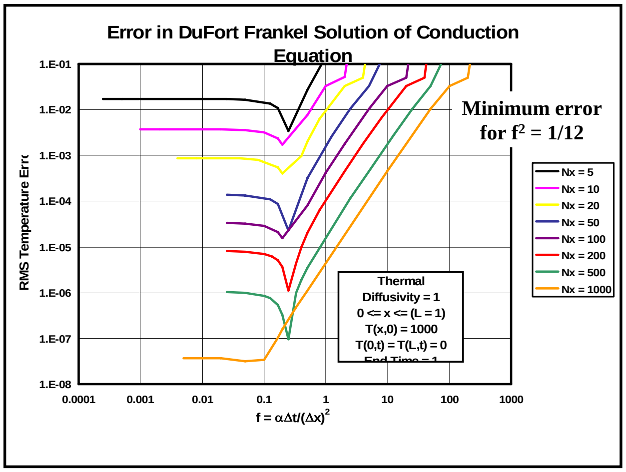
- Saw unexpected oscillations when time step is too large
  - Stable algorithm: oscillations damp out
  - Unstable: oscillations grow without bound
  - Conditional stability: limit on time step
  - Explicit method requires $f = \alpha\Delta t/(\Delta x)^2 < ½$
- Fully implicit, Crank-Nicholson, and DuFort-Frankel are unconditionally stable
  - Have large errors with Δt too large
  - Keep f about 1

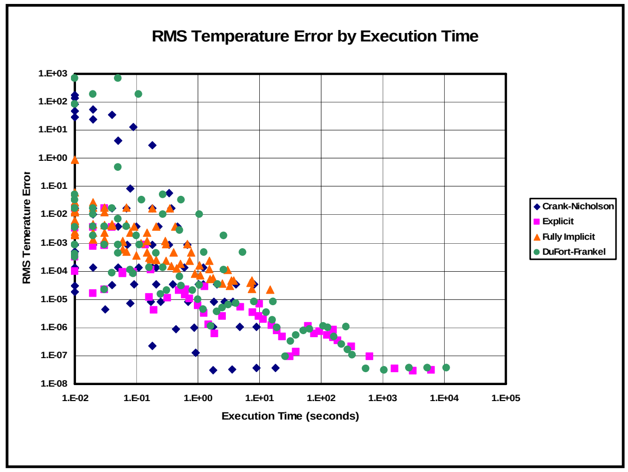
15











Properties of Solutions

- *Consistency* – truncation error becomes zero as step sizes approach zero
- *Stability* – errors remain bounded
- *Convergent* – tends to the exact solution of the differential equation as the grid size tends towards zero
- *Physical reality* – Solutions produce physically realistic results
- *Accuracy* – Many sources of error in numerical solutions.

California State University Northridge 21

Consistency

- Write finite difference expression
- Then write Taylor series for each algebraic term in finite difference expression
- Result will be an equation with derivatives and Δx, Δy, Δz, Δt, etc.
- If we let Δx, Δy, Δz, Δt, etc.approch zero are we left with the differential equation that we are trying to solve?

California State University Northridge 22

Stability

- Solution of difference equations with finite steps gives different results
  - Absolute stability: solution remains bounded for any choice of step size
  - Conditional stability: solution remains bounded for limited step sizes
  - Unconditional instability: Solution is not bounded for any choice of step size
  - Prefer absolute stability but can work with conditional stability

California State University Northridge 23

Convergence

- An algorithm that is consistent and stable is convergent (Lax's theorem)
- Not usually a practical concern to a user of algorithms
- Can run a code with different step sizes and see that solution approaches a consistent value as step size decreases
- DuFort Frankel is neither consistent nor convergent, but gives good results for some chocies of Δx and Δt

California State University Northridge 24

Physical Reality

- Solutions to diffusion equation show uniform decreases – Crank-Nicholson, a stable and convergent algorithm can give oscillations
- Solutions to the wave equation should have no dissipation (an original wave form should propagate with no change in shape) – some algorithms may not give this result

California State University Northridge 25

Problems with Accuracy

- Truncation and round-off errors
- Iterative solution of algebraic equations
- Poor physical models or assumptions
- Blunders
  - Programming errors
  - Incorrect data entries
  - Misinterpretation of results
  - Belief that pretty output must be correct

California State University Northridge 26

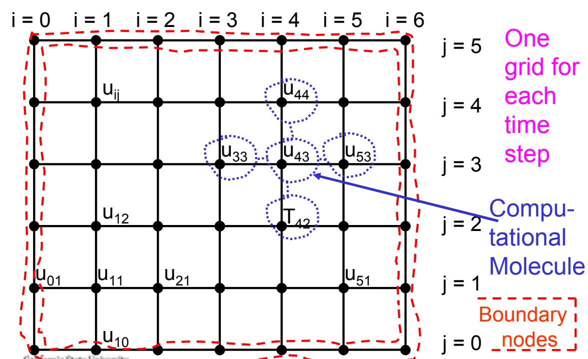
Two Space Dimensions

- Extension of one space dimension
- Have grid in y_j as well as x_i and time
- Explicit method has almost no difference
- Implicit methods harder to solve
- Differential equation shown below
- Explicit method uses same time and space derivatives as in one space dimension

$$\left[\frac{\partial u}{\partial t} = \alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right)\right]_{ij}^n$$

California State University Northridge 27

A Small Grid (N = 6, M = 5)



California State University Northridge 28

Two Space Dimensions II

- Explicit method has error that is second order in space and first order in time

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \alpha \frac{u_{i+1j}^n + u_{i-1j}^n - 2u_{ij}^n}{(\Delta x)^2} + \alpha \frac{u_{ij+1}^n + u_{ij-1}^n - 2u_{ij}^n}{(\Delta y)^2}$$

$$f_x \equiv \frac{\alpha \Delta t}{(\Delta x)^2} \qquad f_y \equiv \frac{\alpha \Delta t}{(\Delta y)^2}$$

$$u_{ij}^{n+1} = f_x\left(u_{i+1j}^n + u_{i-1j}^n\right) + f_y\left(u_{ij+1}^n + u_{ij-1}^n\right) + \left(1 - 2f_x - 2f_y\right)u_i^n$$

California State University Northridge 29

Two Space Dimensions III

- If $\Delta x = \Delta y = h$ so that $f_x = f_y = f$, the explicit equation becomes

$$u_{ij}^{n+1} = f\left(u_{i+1j}^n + u_{i-1j}^n\right) + f\left(u_{ij+1}^n + u_{ij-1}^n\right) + \left(1 - 4f\right)u_i^n$$

- 1 – 4f must be positive for stability

$$f = f_x = \frac{\alpha \Delta t}{(\Delta x)^2} = f_y = \frac{\alpha \Delta t}{(\Delta y)^2} \le \frac{1}{4}$$

- In general $1 - 2f_x - 2f_y$ must be positive

$$f_x + f_y = \frac{\alpha \Delta t}{(\Delta x)^2} + \frac{\alpha \Delta t}{(\Delta y)^2} \le \frac{1}{2}$$

California State University Northridge 30

CN in Two Space Dimensions

- One-dimensional finite-difference equation

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

- Extend this to two dimensions

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

31

CN in Two Space Dimensions II

- Substitute $f_x = \alpha\Delta t/(\Delta x)^2$ and $f_y = \alpha\Delta t/(\Delta y)^2$

$$T_{ij}^{n+1}-T_{ij}^{n}=\frac{f_x}{2}\left[u_{i+1j}^{n+1}+u_{i-1j}^{n+1}-2u_{ij}^{n+1}+u_{i+1j}^{n}+u_{i-1j}^{n}-uT_{ij}^{n}\right]$$
$$\frac{f_y}{2}\left[u_{ij+1}^{n+1}+u_{ij-1}^{n+1}-2u_{ij}^{n+1}+u_{ij+1}^{n}+u_{ij-1}^{n}-2u_{ij}^{n}\right]$$

- Multiply by 2 and rearrange to obtain

$$-f_y u_{ij+1}^{n+1}-f_x u_{i+1j}^{n+1}+2\left(1+f_x+f_y\right)u_{ij}^{n+1}-f_x u_{i-1j}^{n+1}-f_y u_{ij-1}^{n+1}=$$
$$f_x\left(u_{i+1j}^{n}+u_{i-1j}^{n}\right)+f_y\left(u_{ij+1}^{n}+u_{ij-1}^{n}\right)+2\left(1-f_x-f_y\right)u_{ij}^{n}=R_{ij}^{n}$$

32

FI in Two Space Dimensions

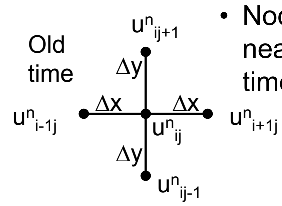
- One-dimensional fully implicit $\quad -fu_{i-1}^{n+1}+(1+2f)u_i^{n+1}-fu_{i+1}^{n+1}=u_i^n$
- Extend this to two dimensions

$$-f_y u_{ij-1}^{n+1}-f_x u_{i-1j}^{n+1}+(1+2f_x+2f_y)u_{ij}^{n+1}-f_x u_{i+1j}^{n+1}-f_y u_{ij+1}^{n+1}=u_{ij}^{n}$$

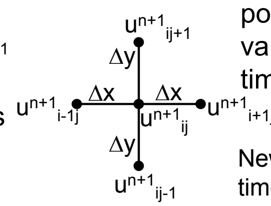
- System of implicit equations is not tridiagonal
- Look at small grid
- Keep same grid for all time steps
  - Have to store one time step only

33

Finite-difference Grid



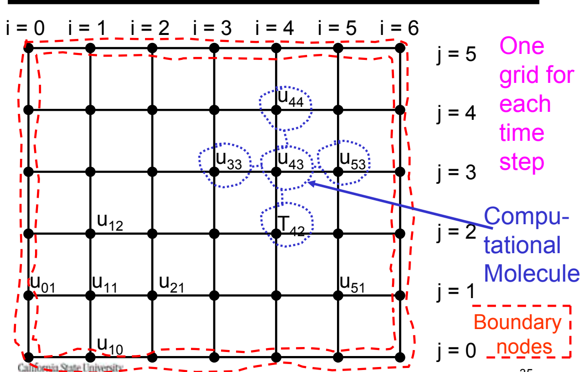
- Node related to four nearest neighbors at both time steps
- CN uses all potential values at old time step



- Discuss iteration methods with Laplace equation

34

A Small Grid (N = 6, M = 5)



35

Alternating Direction Implicit

- ADI is technique for allowing use of Thomas algorithm for simple solution
  - At each time step, treat one direction as explicit and one as implicit
  - Exchange explicit and implicit directions at each time step
  - Keep results at each time step, but always have an even number of time steps
- Different approaches include Peaceman-Radford and Douglas-Radford

36

Peaceman-Radford ADI

- Solve first for approximate u^* at t_{n+1}

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

- Solve next for actual result u^{n+2}

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

- Each equation gives a tridiagonal system

California State University Northridge 37

Peaceman Radford II

- Introduce $f_x = \alpha\Delta t/(\Delta x)^2$ and $f_y = \alpha\Delta t/(\Delta y)^2$ and get tridiagonal forms
- First time step

$$-f_x u_{i-1j}^* + (1+2f_x)u_{ij}^* - f_x u_{i+1j}^* = f_y\left(u_{ij+1}^n + u_{ij-1}^n\right) + (1-2f_y)u_{ij}^n$$

$$-f_x u_{i-1j}^* + (1+2f_x)u_{ij}^* - f_x u_{i+1j}^* = R_{ij}^n$$

- Second time step

$$-f_y u_{ij-1}^{n+2} + (1+2f_y)u_{ij}^{n+2} - f_y u_{ij+1}^{n+2} = f_x\left(u_{i+1j}^* + u_{i-1j}^*\right) + \left(1-2f_x\right)u_{ij}^*$$

$$-f_y u_{ij-1}^{n+2} + (1+2f_y)u_{ij}^{n+2} - f_y u_{ij+1}^{n+2} = R_{ij}^*$$

California State University Northridge 38

Three Dimensional Problem

- Extend explicit two-dimensional finite-difference form to three-dimensions

$$\frac{u_{ijk}^{n+1} - u_{ijk}^n}{\Delta t} = \alpha\frac{u_{i+1jk}^n + u_{i-1jk}^n - 2u_{ijk}^n}{(\Delta x)^2} +$$
$$\alpha\frac{u_{ij+1k}^n + u_{ij-1k}^n - 2u_{ijk}^n}{(\Delta y)^2} + \alpha\frac{u_{ijk+1}^n + u_{ijk-1}^n - 2u_{ijk}^n}{(\Delta z)^2}$$

$$f_x = \frac{\alpha\Delta t}{(\Delta x)^2} \qquad f_y = \frac{\alpha\Delta t}{(\Delta y)^2} \qquad f_z = \frac{\alpha\Delta t}{(\Delta z)^2}$$

$$f_x + f_y + f_z \le \frac{1}{2}$$

California State University Northridge 39

Douglas-Brian Method

- An ADI method in three steps

$$\frac{u_{ijk}^* - u_{ijk}^n}{\Delta t} = \frac{\alpha}{2}\left[\frac{u_{i+1jk}^* + u_{i-1jk}^* - 2u_{ijk}^*}{(\Delta x)^2} + \frac{u_{i+1jk}^n + u_{i-1jk}^n - 2u_{ijk}^n}{(\Delta x)^2}\right]$$
$$\alpha\frac{u_{ij+1k}^n + u_{ij-1k}^n - 2u_{ijk}^n}{(\Delta y)^2} + \alpha\frac{u_{ijk+1}^n + u_{ijk-1}^n - 2u_{ijk}^n}{(\Delta z)^2}$$

$$\frac{u_{ijk}^{**} - u_{ijk}^n}{\Delta t} = \frac{\alpha}{2}\left[\frac{u_{i+1jk}^* + u_{i-1jk}^* - 2u_{ijk}^*}{(\Delta x)^2} + \frac{u_{i+1jk}^n + u_{i-1jk}^n - 2u_{ijk}^n}{(\Delta x)^2}\right] +$$
$$\frac{\alpha}{2}\left[\frac{u_{ij+1k}^{**} + u_{ij-1k}^{**} - 2u_{ijk}^{**}}{(\Delta y)^2} + \frac{u_{ij+1k}^n + u_{ij-1k}^n - 2u_{ijk}^n}{(\Delta y)^2}\right] + \alpha\frac{u_{ijk+1}^n + u_{ijk-1}^n - 2u_{ijk}^n}{(\Delta z)^2}$$

California State University Northridge 40

Douglas-Brian Method II

$$\frac{u_{ijk}^{n+1} - u_{ijk}^n}{\Delta t} = \frac{\alpha}{2}\left[\frac{u_{i+1jk}^* + u_{i-1jk}^* - 2u_{ijk}^*}{(\Delta x)^2} + \frac{u_{i+1jk}^n + u_{i-1jk}^n - 2u_{ijk}^n}{(\Delta x)^2}\right]$$
$$+ \frac{\alpha}{2}\left[\frac{u_{ij+1k}^{**} + u_{ij-1k}^{**} - 2u_{ijk}^{**}}{(\Delta y)^2} + \frac{u_{ij+1k}^n + u_{ij-1k}^n - 2u_{ijk}^n}{(\Delta y)^2}\right]$$
$$+ \frac{\alpha}{2}\left[\frac{u_{ijk+1}^{n+1} + u_{ijk-1}^{n+1} - 2u_{ijk}^{n+1}}{(\Delta z)^2} + \frac{u_{ijk+1}^n + u_{ijk-1}^n - 2u_{ijk}^n}{(\Delta z)^2}\right]$$

- Each step – solving for u^*, u^{**}, and u^{n+1}, has a tridiagonal system of equations
  - Can show that this is unconditionally stable with $O[(\Delta t)^2, (\Delta x)^2, (\Delta y)^2, (\Delta z)^2]$ truncation error

California State University Northridge 41

Operator Splitting

- A method similar to ADI
- Allows use of Thomas algorithm to solve a tridiagonal system of equations
- Uses difference operator notation, $\Delta_x^2\phi_{ij} = \phi_{i+1j} + \phi_{i-1j} - 2\phi_{ij}$ and $\Delta_y^2\phi_{ij} = \phi_{ij+1} + \phi_{ij-1} - 2\phi_{ij}$
- Uses two operator factors and ignores cross product which produces same error as time derivative, $O[(\Delta t)^2]$

California State University Northridge 42

Operator Splitting

- Like ADI, but do not use midpoint step
- Here use difference operator notation, Δ^2

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

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

$$\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)u_{ij}^{n+1}=u_{ij}^{n}$$

California State University Northridge 43

Operator Splitting II

- Use approximate factorization

$$\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)u_{ij}^{n+1}=$$

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

Ignoring this term gives O[$(\Delta t)^2$] error

- Result with O[$(\Delta t)^2$] term neglected

$$\left(1-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}\right)u_{ij}^{n+1}=u_{ij}^{n}$$

California State University Northridge 44

Operator Splitting III

- Define intermediate value u^*_{ij}

$$\left(1-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}\right)u_{ij}^{n+1}=u_{ij}^{n}$$

$$\left(1-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)u_{ij}^{*}=\left(1-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)\left[\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}\right)u_{ij}^{n+1}\right]=u_{ij}^{n}$$

- Solve two sets of tridiagonal equations for each time step

$$\left(1-\alpha\Delta t\frac{\Delta_y^2}{(\Delta y)^2}\right)u_{ij}^{*}=u_{ij}^{n}$$

$$\left(1-\alpha\Delta t\frac{\Delta_x^2}{(\Delta x)^2}\right)u_{ij}^{n+1}=u_{ij}^{*}$$

California State University Northridge 45