Transient Problems and Stability

Larry Caretto
Mechanical Engineering 692

Computational Fluid Dynamics

April 12, 2010

California State University Northridge

Introduction

- Look at transient problems
- How to handle time derivatives
- Various algorithms
- Examine conduction equation as a sample problem
- Stability analysis (von Neumann)
- Stability of algorithms

California State University Northridge 2

The Time Derivative

- In previous lectures we considered
  - Finite difference approaches
  - Finite volume approaches
  - Finite element approaches
- Although we can treat finite differences in time like any other derivative we have not considered these to date
- Time is typically a one-way coordinate
  - Future events do not affect past ones

California State University Northridge 3

Time Derivatives II

- Treatment of time is typically by finite differences even when the spatial coordinates are done by finite elements or finite volume
  - There is no complex geometry in time that requires complex models
  - Basic approach is to take any complex equation with a time derivative and write it as $\partial\phi/\partial t = \phi(x, y, z, t)$ and then to model it similarly to ordinary differential equations

California State University Northridge 4

Time Derivatives III

- Look to solution of ordinary differential equations for models of time derivatives
  - Have several dependent variables y_1, y_2, …, y_N represented by the vector **y**
  - Solve for the time evolution of each of these variables by a set of ODEs: $d\phi_i/dt = f_i(t, \phi_1, \phi_2, …, \phi_N)$ or $d\phi/dt = \mathbf{f}(t, \phi)$
  - Let ϕ^n represent the values of the variables, ϕ_i at time t_n $\phi^n = [\phi_1^n,…., \phi_N^n]$
  - One time step updates ϕ^n to ϕ^{n+1}

California State University Northridge 5

ODE Algorithms

- Start with initial conditions ϕ^0 at t = 0
- Take one time step, Δt, to ϕ^1
- Repeat process for all other time steps using ϕ^n as initial condition in the algorithm to get to ϕ^{n+1}
- Basic algorithm to solve $d\phi/dt = \mathbf{f}(t,\phi)$ is $\phi^{n+1} = \phi^n + \mathbf{f}_{average}\Delta t$
  - Equation-by-equation basis, $\phi_i^{n+1} = \phi_i^n + f_{i,average}\Delta t$

California State University Northridge 6

ODE Algorithms II

- Explicit Euler Method: $\phi_i^{n+1} = \phi_i^n + f_i^n\Delta t$
- Implicit Euler Method: $\phi_i^{n+1} = \phi_i^n + f_i^{n+1}\Delta t$
- Trapezoid rule: $\phi_i^{n+1} = \phi_i^n + [\, f_i^n + f_i^{n+1}]/2\Delta t$
- Midpoint rule: $\phi_i^{n+1} = \phi_i^n + f_i(t+\Delta t/2,\ \phi^{n+1/2})\Delta t$
- Many other methods including Runge-Kutta predictor corrector methods that use more terms in $f_{average}$
- Approaches listed here usually ones used for partial differential equations which balance spatial and temporal accuracy

California State University Northridge 7

ODE Algorithms III

- Explicit Euler Method: $\phi_i^{n+1} = \phi_i^n + f_i^n\Delta t$
- Implicit Euler Method: $\phi_i^{n+1} = \phi_i^n + f_i^{n+1}\Delta t$
- Trapezoid rule: $\phi_i^{n+1} = \phi_i^n + [\, f_i^n + f_i^{n+1}]/2\Delta t$
- Midpoint rule: $\phi_i^{n+1} = \phi_i^n + f_i(t+\Delta t/2,\ \phi^{n+1/2})\Delta t$
- Key difference is implicit *versus* explicit
  - Explicit algorithms compute ϕ_i^{n+1} from information available at time step n
  - Implicit algorithms, require information from time step n+1, need to be solved by iteration or simultaneous solution of equations

California State University Northridge 8

Numerical PDE Solutions

- Define a finite-difference grid in the independent variables (x, y, z, t)
- Place grid points on region boundary whose values are found from boundary conditions for the problem
- At some grid location convert differential equation into a finite difference equation
  - Observe truncation error in process
  - Neglect truncation error to get set of algebraic equations to solve

California State University Northridge 9

Conduction Equation

- Apply difference formulas derived for ordinary derivatives to partial derivatives
- Use notation to consider different coordinate directions
- Apply to conduction equation $\left[\frac{\partial T}{\partial t} = \alpha\frac{\partial^2 T}{\partial x^2}\right]_i^n$
- Grids $x_i = x_0 + i\Delta t$ and $t_n = t_0 + n\Delta t$
- Try finite difference expressions below to get explicit finite-difference equation

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

California State University Northridge 10

Conduction Equation II

- Substitute finite difference expressions into differential equation

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

- Ignore truncation error, solve for T_i^{n+1}

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

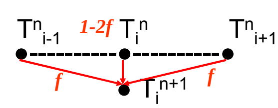
- Obtain temperature at $x = x_i$ and $t = t_{n+1}$ in terms of T values at old time step

California State University Northridge 11

Explicit (FTCS) Method

- Method just derived is called explicit method; can solve one equation at a time

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



$$f \equiv \frac{\alpha\Delta t}{(\Delta x)^2}$$

- T_i^{n+1} does not depend on other T values at the new time step (n+1)

California State University Northridge 12

Explicit Method Example

- Pick α = 1, Δx = 0.25, N_x = 4, Δt = 0.01
- $f = \alpha\Delta t/(\Delta x)^2 = 1(.01)/(.25)^2 = 0.16$
- Pick initial T_i^0 = 1000 and boundaries, $T^n_0 = T^n_4$ = 0 for time > 0 (n ≥ 0)

$$Apply\ T_i^{n+1} = f\left(T_{i+1}^n + T_{i-1}^n\right) + \left(1-2f\right)T_i^n$$

$$T_1^1 = f\left[T_0^0 + T_2^0\right] + (1-2f)T_1^0 = 0.16[0+1000] + 0.68[1000] = 840$$

$$T_2^1 = f\left[T_1^0 + T_3^0\right] + (1-2f)T_2^0 = 0.16[1000+1000] + 0.68[1000] = 1000$$

$$T_3^1 = f\left[T_2^0 + T_4^0\right] + (1-2f)T_3^0 = 0.16[1000+0] + 0.68[1000] = 840$$

- Repeat for subsequent time steps

California State University Northridge 13

Explicit Method Results f = 0.16

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0.00	x = 0.25	x = 0.50	x = 0.75	x = 1.00
	t = 0	1000	1000	1000	1000	1000
n = 0	t = 0+	0	1000	1000	1000	0
n = 1	t = 0.01	0	840	1000	840	0
n = 2	t = 0.02	0	731.2	948.8	731.2	0
n = 3	t = 0.03	0	649	879.2	649	0
n = 4	t = 0.04	0	582	805.5	582	0
n = 5	t = 0.05	0	524.6	734	524.6	0
n = 6	t = 0.06	0	474.2	667	474.2	0
n = 7	t = 0.07	0	429.2	605.3	429.2	0
n = 8	t = 0.08	0	388.7	548.9	388.7	0

California State University Northridge 14

Explicit Method Results f = 0.16

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0.00	x = 0.25	x = 0.50	x = 0.75	x = 1.00
n = 12	t = 0.12	0	262	370.5	262	0
n = 13	t = 0.13	0	237.5	335.8	237.5	0
n = 14	t = 0.14	0	215.2	304.4	215.2	0
n = 15	t = 0.15	0	195	275.8	195	0
n = 16	t = 0.16	0	176.8	250	176.8	0
n = 17	t = 0.17	0	160.2	226.5	160.2	0
n = 18	t = 0.18	0	145.2	205.3	145.2	0
n = 19	t = 0.19	0	131.6	186.1	131.6	0
n = 20	t = 0.20	0	119.2	168.6	119.2	0
Exact	t = 0.20	0	125.1	176.9	125.1	0
Error	t = 0.20	0	5.8	8.2	5.8	0

California State University Northridge 15

Explicit Results f = 0.32

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0.00	x = 0.25	x = 0.50	x = 0.75	x = 1.00
	t = 0	1000	1000	1000	1000	1000
n = 0	t = 0+	0	1000	1000	1000	0
n = 1	t = 0.02	0	680	1000	680	0
n = 2	t = 0.04	0	564.8	795.2	564.8	0
n = 3	t = 0.06	0	457.9	647.7	457.9	0
n = 8	t = 0.16	0	162.2	229.4	162.2	0
n = 9	t = 0.18	0	131.8	186.4	131.8	0
n = 10	t = 0.20	0	107.1	151.4	107.1	0
Exact	t = 0.20	0	125.1	176.9	125.1	0
Error	t = 0.20	0	18	25.4	18	0

California State University Northridge 16

Explicit Results f = 0.64

		i = 0	i = 1	i = 2	i = 3	i = 4
		x = 0.00	x = 0.25	x = 0.50	x = 0.75	x = 1.00
	t = 0	1000	1000	1000	1000	1000
n = 0	t = 0+	0	1000	1000	1000	0
n = 1	t = 0.04	0	360	1000	360	0
n = 2	t = 0.08	0	539.2	180.8	539.2	0
n = 3	t = 0.12	0	-35.3	639.6	-35.3	0
n = 4	t = 0.16	0	419.2	-224.2	419.2	0
n = 5	t = 0.20	0	-260.9	599.3	-260.9	0
Exact	t = 0.20	0	125.1	176.9	125.1	0
Error	t = 0.20	0	385.9	422.5	385.9	0

California State University Northridge 17

What Happened?

- We are seeing effects of instability
- Difference equations may not converge
  - Unstable equations grow without bound
  - May have stable equations that produce incorrect results
  - Conditional stability requires step size less than that needed for accuracy
  - Goal of absolute stability not always possible
  - Discussions of stability complex, can sometimes use physical arguments

California State University Northridge 18

Stability of Explicit Method

- If the values of T_{i+1} and T_{i-1} are fixed an increase in T_i^n should increase T_i^{n+1}
- If f is greater than 0.5, an increase in T_i^n will cause a decrease in T_i^{n+1}
- We can avoid this incorrect result by keeping $f = \alpha\Delta t/(\Delta x)^2 \leq 0.5$
- This imposes a time step limit that may be less than the limit required for accuracy in the solution

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

California State University Northridge 19

Crank-Nicholson Method

- Seek more accurate time derivative
- Provides implicit method
  - Value of T_i^{n+1} depends on T_i^{n+1} and T_i^{n-1}
  - More work per step, but can take longer time steps with this method
  - Apply to diffusion equation at time n + 1/2

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

California State University Northridge 20

Space Derivative at $t_{n+1/2}$

- Take average of space derivative at time steps n and n + 1
- Show average is second order accurate

$$f_{i+1} = f_i + f_i' h + f_i'' \frac{h^2}{2} + f_i''' \frac{h^3}{6} + \ldots$$

$$+ \quad f_{i-1} = f_i - f_i' h + f_i'' \frac{h^2}{2} - f_i''' \frac{h^3}{6} + \ldots$$

$$f_{i+1} + f_{i-1} = 2f_i + 2f_i'' \frac{h^2}{2} + 2f_i'''' \frac{h^4}{24} + \ldots$$

$$f_i = \frac{f_{i+1} + f_{i-1}}{2} - f_i'' \frac{h^2}{4} - f_i'''' \frac{h^4}{48} + \ldots = \frac{f_{i+1} + f_{i-1}}{2} + O(h^2)$$

California State University Northridge 21

Using Space Derivative at $t_{n+1/2}$

- Apply average to space derivative

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_i^{n+\frac{1}{2}} = \frac{1}{2}\left[\left.\frac{\partial^2 T}{\partial x^2}\right|_i^n + \left.\frac{\partial^2 T}{\partial x^2}\right|_i^{n+1}\right] + O[(\Delta t)^2]$$

- Substitute into diffusion equation

$$\left.\frac{\partial T}{\partial t}\right|_i^{n+\frac{1}{2}} - \alpha \left.\frac{\partial^2 T}{\partial x^2}\right|_i^{n+\frac{1}{2}} = \frac{T_i^{n+1} - T_i^n}{\Delta t}$$

$$-\frac{\alpha}{2}\left[\frac{T_{i+1}^{n+1} + T_{i-1}^{n+1} - 2T_i^{n+1}}{(\Delta x)^2} + \frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{(\Delta x)^2}\right] + O[(\Delta t)^2, (\Delta x)^2] = 0$$

California State University Northridge 22

Crank-Nicholson Equation

- Introduce $f = \alpha\Delta t/(\Delta x)^2$ and rearrange
  - Three values at new time step

$$-\frac{f}{2}T_{i-1}^{n+1} + (1+f)T_i^{n+1} - \frac{f}{2}T_{i+1}^{n+1} = \frac{f}{2}\left[T_{i+1}^n + T_{i-1}^n\right] + (1-f)T_i^n$$

$$-fT_{i-1}^{n+1} + 2(1+f)T_i^{n+1} - fT_{i+1}^{n+1} = f\left[T_{i+1}^n + T_{i-1}^n\right] + 2(1-f)T_i^n$$

- Tridiagonal system of equations easily solved by Thomas algorithm

$$-fT_{i-1}^{n+1} + 2(1+f)T_i^{n+1} - fT_{i+1}^{n+1} = R_i^n$$

$$R_i^n = f\left[T_{i+1}^n + T_{i-1}^n\right] + 2(1-f)T_i^n$$

California State University Northridge 23

Crank-Nicholson Equations

- Consider case where boundary temperatures T_0 and T_N are specified
- Rewrite equations in matrix form to show tridiagonal structure

$$\begin{bmatrix} 2(1-f) & -f & 0 & 0 & \cdots & 0 & 0 \\ -f & 2(1-f) & -f & 0 & \cdots & 0 & 0 \\ 0 & -f & 2(1-f) & -f & \cdots & 0 & 0 \\ 0 & 0 & -f & 2(1-f) & & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2(1-f) & -f \\ 0 & 0 & 0 & 0 & \cdots & -f & 2(1-f) \end{bmatrix} \begin{bmatrix} T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ \vdots \\ \vdots \\ T_{N-2}^{n+1} \\ T_{N-1}^{n+1} \end{bmatrix} = \begin{bmatrix} R_1^n + fT_0^n \\ R_2^n \\ R_3^n \\ \vdots \\ \vdots \\ R_{N-2}^n \\ R_{N-1}^n + fT_N^n \end{bmatrix}$$

California State University Northridge 24

Crank Nicholson Results

- Results for $\alpha = 1$, L = 1, $\Delta x = 0.01$, $\Delta t = 0.0005$, $f = \alpha\Delta t/(\Delta x)^2 = 5$

		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	-73.35	423.96	690.85	834.09
n = 2	t = 0.001	0	352.75	305.27	440.73	599.81
n = 3	t = 0.0015	0	25.7	320.81	439.19	533.34
n = 4	t = 0.002	0	203.86	209.57	347.52	473.02
n = 5	t = 0.0025	0	56.79	252.91	334.12	422.43
n = 6	t = 0.003	0	141.46	177.47	298.2	397.48

Crank Nicholson 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	66.73	209.02	279.22	363.26
n = 8	t = 0.004	0	109.4	160.3	263.81	347.29
n = 9	t = 0.0045	0	68.71	179.63	245.68	324.49
n = 10	t = 0.005	0	90.79	148.2	237.92	311.75
n = 11	t = 0.0055	0	67.5	159.07	222.68	296.08
n = 12	t = 0.006	0	78.99	138.51	217.76	285.25
n = 13	t = 0.0065	0	65.08	144.07	205.56	273.92
n = 14	t = 0.007	0	70.94	130.31	201.68	264.62
n = 15	t = 0.0075	0	62.29	132.69	192.04	255.97
n = 16	t = 0.008	0	65.1	123.21	188.58	247.99
n = 17	t = 0.0085	0	59.5	123.75	180.95	241.06

Crank Nicholson 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	60.65	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

Fully Implicit Method

- Discretize diffusion equation at t_{n+1}

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

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

$$-fT_{i-1}^{n+1}+(1+2f)T_i^{n+1}-fT_{i+1}^{n+1}=T_i^n$$

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

Fully Implicit Results

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

		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

		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

		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

Richardson/Leapfrog

- Use two time step central differences

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

- Result is explicit with second order accuracy in time

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

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

DuFort Frankel

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

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

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

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

DuFort Frankel

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

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

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

- Result is explicit for values at time n+1
- Explicit start required to get first set of values at time n-1

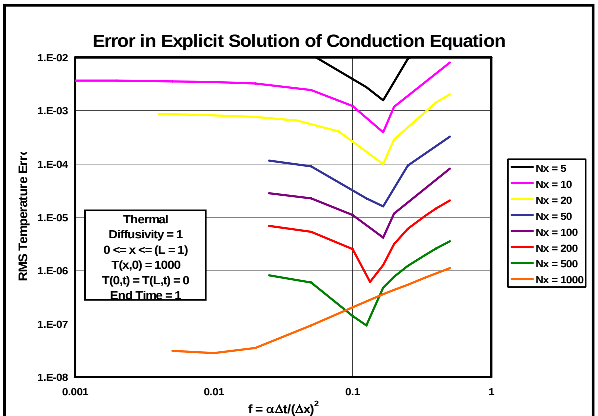
Detailed Truncation Error

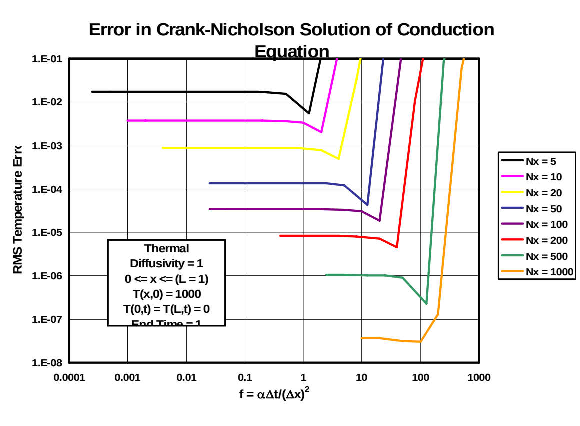
- Look at full infinite series for truncation error [see equations in notes for details]
- Explicit method truncation error [3A-10]

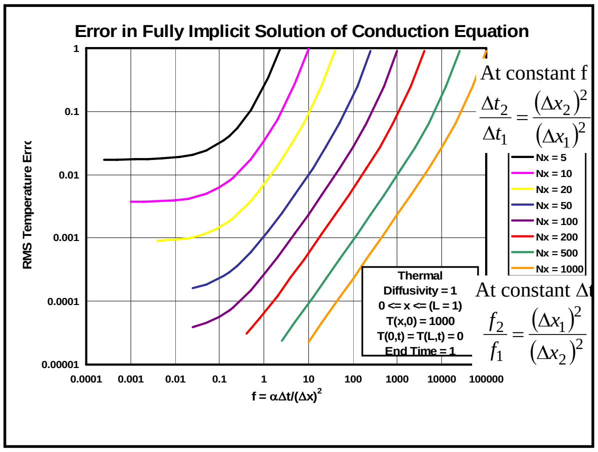
$$TE_i^n = \alpha \sum_{k=2}^{\infty} (\Delta x)^{2k-2}\left[\frac{2}{(2k)!} - \frac{f^{k-1}}{k!}\right]\left.\frac{\partial^{2k} T}{\partial x^{2k}}\right|_i^n$$

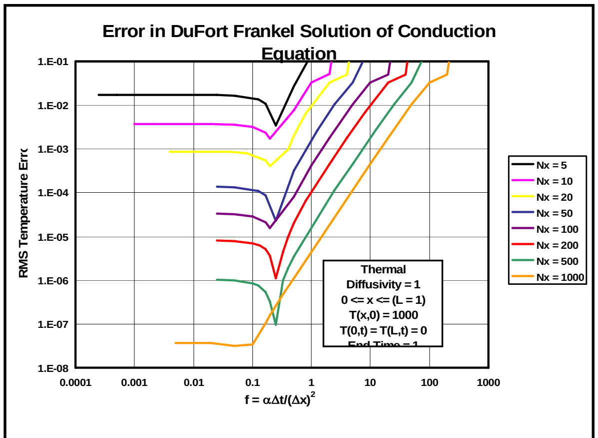
$$TE_i^n = \alpha(\Delta x)^2\left[\frac{2}{4!} - \frac{f}{2!}\right]\left.\frac{\partial^4 T}{\partial x^4}\right|_i^n + \alpha\sum_{k=3}^{\infty}\cdots$$

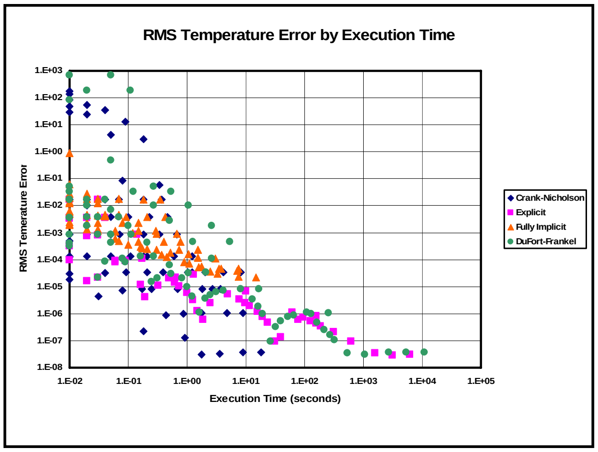
- Lead term vanishes if 2/24 = f/2 or f = 1/6











von Neumann Stability

- Examines stability due to differential equation alone
- Does not consider boundary conditions
- Based on idea that numerical time integration is a series of finite-difference equations that may diverge
- Seeks conditions for which equations will or will not converge
- Use explicit algorithm as example

California State University Northridge

von Neumann Stability II

- Outline of process
  - Start with finite difference equation
  - Define D_k^n as exact solution to finite-difference equation at x_k and t_n
  - Define error, $\varepsilon_k^n = D_k^n - T_k^n$ (or $= D_k^n - \phi_k^n$)
  - Show that error satisfies same finite difference equation as T_k^n (or ϕ_k^n)
  - Model error as discrete complex Fourier series

$$\varepsilon(x,t) = \sum_{m=0}^{M} \varepsilon_m(x,t) \qquad \varepsilon_m(x,t) = e^{at}\, e^{i\beta_m x}$$

$$\beta_m = \frac{m\pi}{L} \qquad m = 0, 1, 2, \ldots M$$

California State University Northridge

Fourier Series

- A general function f(x) can be expressed as an infinite series of sines and cosines

$$f(x) = a_0 + \sum_{n=1}^{\infty} a_n \cos\frac{n\pi x}{L} + \sum_{n=1}^{\infty} b_n \sin\frac{n\pi x}{L}$$

$$a_0 = \frac{1}{L}\int_{-L}^{L} f(x)dx \quad a_n = \frac{1}{L}\int_{-L}^{L} f(x)\cos\frac{n\pi x}{L}dx \quad b_n = \frac{1}{L}\int_{-L}^{L} f(x)\sin\frac{n\pi x}{L}dx$$

- Euler formula

$$e^{i\theta} = \cos\theta + i\sin\theta$$

California State University Northridge 43

Complex Fourier Series

- Write in terms of complex exponentials instead of sines and cosines

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i\beta_n x} \qquad c_n = \int_{-\infty}^{\infty} f(x) e^{i\beta_n x} dx$$

- Detailed derivation from trigonometric series to complex series omitted here
- Note that β_n is like $n\pi/L$
  - Just as higher values of $n\pi/L$ imply higher frequencies so do higher values of β_n

California State University Northridge 44

von Neumann Stability III

- Outline of the process continued
  - From previous chart
    - The error ε_k^n satisfies same FDE as T_k^n
    - Use complex Fourier series for the error
  - Want to ensure that error does not grow with time for a given x
  - Define growth factor, G, for error that should be ≤ 1 for all modes, m

$$G = \left|\frac{\varepsilon_m(x,t_{n+1})}{\varepsilon_m(x,t_n)}\right| = \left|\frac{e^{a(t+\Delta t)}\ e^{i\beta_m x}}{e^{at}\ e^{i\beta_m x}}\right| = \left|e^{a\Delta t}\right| \le 1$$

California State University Northridge 45

von Neumann Stability IV

- Outline of the process concluded
  - Previous charts gave equation for error in Fourier modes and defined growth factor that must be ≤ 1 for all modes, m
  - To apply this substitute error equation into finite difference equation and solve for growth factor, $G = |e^{at}|$
  - See if G is ≤ 1 for all conditions, some conditions or no conditions
  - Equation below is error for mode m of FDE

$$\varepsilon_m(x_k, t_n) = \varepsilon_k^n = e^{at}\ e^{i\beta_m x} = e^{an\Delta t}\ e^{i\beta_m(x_0 + k\Delta x)}$$

California State University Northridge 46

Applying von Neumann

- Example: explicit conduction algorithm
- See notes for more details
- State with FDE for T and substitute error

$$T_k^{n+1} = f\left[T_{k+1}^n + T_{k-1}^n\right] + (1-2f)T_k^n$$

$$\varepsilon_k^{n+1} = f\left[\varepsilon_{k+1}^n + \varepsilon_{k-1}^n\right] + (1-2f)\varepsilon_k^n$$

$$\varepsilon_k^n = e^{an\Delta t}\ e^{i\beta_m(x_0 + k\Delta x)}$$

Substitute error expression into FDE

California State University Northridge 47

Applying von Neumann II

- Result of substituting error into FDE

$$\varepsilon_k^{n+1} = f\left[\varepsilon_{k+1}^n + \varepsilon_{k-1}^n\right] + (1-2f)\varepsilon_k^n$$

$$\varepsilon_k^n = e^{an\Delta t}\ e^{i\beta_m(x_0 + k\Delta x)}$$

$$e^{a(n+1)\Delta t}\ e^{i\beta_m(x_0 + k\Delta x)}$$

$$= f\left[e^{an\Delta t}\ e^{i\beta_m[x_0+(k+1)\Delta x]} + e^{an\Delta t}\ e^{i\beta_m[x_0+(k-1)\Delta x]}\right] +$$

$$(1-2f)e^{an\Delta t}\ e^{i\beta_m(x_0 + k\Delta x)}$$

See common factor of $e^{an\Delta t}e^{i\beta_m[x_0+k\Delta x]}$

California State University Northridge 48

Applying von Neumann III

- Divide out common factor of $e^{an\Delta t}e^{i\beta_m[x_0+k\Delta x]}$

$$\left(e^{an\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}\right)e^{a\Delta t}=\left(e^{an\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}\right)\bullet$$
$$f\left[e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}\right]+(1-2f)\left(e^{an\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}\right)$$

$$e^{a\Delta t}=f\left[e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}\right]+(1-2f)$$

Substitute: $e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}=2\cos(\beta_m\Delta x)$

$$e^{a\Delta t}=2f\cos(\beta_m x)+(1-2f)=2f[\cos(\beta_m x)-1]+1$$

Northridge 49

Applying von Neumann IV

- Use this equation $\cos(\theta)-1=-2\sin^2\left(\frac{\theta}{2}\right)$

$$e^{a\Delta t}=2f[\cos(\beta_m x)-1]+1=2f\left[-2\sin^2\left(\frac{\beta_m x}{2}\right)\right]+1$$

$$e^{a\Delta t}=1-4f\sin^2\left(\frac{\beta_m x}{2}\right)$$

- Take absolute value to get growth factor, G = $|e^{a\Delta t}|$, and see if G ≤ 1

Northridge 50

Applying von Neumann V

- Consider both negative and positive values of absolute value argument

$$G=\left|e^{a\Delta t}\right|=\left|1-4f\sin^2\left(\frac{\beta_m x}{2}\right)\right|\le 1$$

$$1-4f\sin^2\left(\frac{\beta_m x}{2}\right)\le 1 \quad and \quad 4f\sin^2\left(\frac{\beta_m x}{2}\right)-1\le 1$$

Always true since both f and $\sin^2$ are positive

$$f\le\frac{2}{4\sin^2\left(\frac{\beta_m x}{2}\right)}$$

Northridge 51

Applying von Neumann VI

- We can be sure that G ≤ 1 for all $\beta_m x$ if G ≤ 1 when $\sin^2$ has its maximum value

$$f\le\frac{2}{4\sin^2\left(\frac{\beta_m x}{2}\right)}$$ is satisfied if $$f=\frac{\alpha\Delta t}{(\Delta x)^2}\le\frac{1}{2}$$

- This is same result obtained by physical arguments

Northridge 52

Crank Nicholson Stability

- See notes for more details
- State with FDE for T and substitute error

$$-\frac{f}{2}T_{k-1}^{n+1}+(1+f)T_k^{n+1}-\frac{f}{2}T_{k+1}^{n+1}=\frac{f}{2}\left[T_{k+1}^n+T_{k-1}^n\right]+(1-f)T_k^n$$

$$-\frac{f}{2}\varepsilon_{k-1}^{n+1}+(1+f)\varepsilon_k^{n+1}-\frac{f}{2}\varepsilon_{k+1}^{n+1}=\frac{f}{2}\left[\varepsilon_{k+1}^n+\varepsilon_{k-1}^n\right]+(1-f)\varepsilon_k^n$$

$$\varepsilon_k^n=e^{an\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}$$ Substitute error expression into FDE

Northridge 53

Crank Nicholson Stability II

- Substitute error into FDE and eliminate common factor of $e^{an\Delta t}e^{i\beta_m[x_0+k\Delta x]}$

$$-\frac{f}{2}e^{a(n+1)\Delta t}\,e^{i\beta_m[x_0+(k-1)\Delta x]}+(1+f)e^{a(n+1)\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}$$
$$-\frac{f}{2}e^{a(n+1)\Delta t}\,e^{i\beta_m[x_0+(k+1)\Delta x]}=\frac{f}{2}\left[e^{an\Delta t}\,e^{i\beta_m[x_0+(k+1)\Delta x]}\right.$$
$$\left.+e^{an\Delta t}\,e^{i\beta_m[x_0+(k-1)\Delta x]}\right]+(1-f)e^{an\Delta t}\,e^{i\beta_m(x_0+k\Delta x)}$$

Northridge 54

Crank Nicholson Stability III

- Divide out $e^{a n\Delta t}e^{i\beta_m[x_0+k\Delta x]}$ and rearrange

$$e^{a\Delta t}\left[1+f-\frac{f}{2}\left(e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}\right)\right] = \frac{f}{2}\left[e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}\right]+(1-f)$$

$$e^{i\beta_m\Delta x}+e^{-i\beta_m\Delta x}=2\cos(\beta_m\Delta x)$$

$$e^{a\Delta t}\left[1+f-f\cos(\beta_m x)\right]=f\cos(\beta_m x)+(1-f)$$

$$\cos(\beta_m\Delta x)=1-2\sin^2\left(\frac{\beta_m\Delta x}{2}\right)$$

Crank Nicholson Stability IV

$$e^{a\Delta t}\left[1+f-f+2f\sin^2\left(\frac{\beta_m x}{2}\right)\right] = f-2f\sin^2\left(\frac{\beta_m x}{2}\right)+(1-f)$$

$$G=\left|e^{a\Delta t}\right|=\left|\frac{1-2f\sin^2\left(\frac{\beta_m x}{2}\right)}{1+2f\sin^2\left(\frac{\beta_m x}{2}\right)}\right|\le 1$$

Crank Nicholson Stability V

- Growth factor is |(1 – z) / (1 + z)| where z is positive; this is always < 1

$$G=\left|e^{a\Delta t}\right|=\left|\frac{1-2f\sin^2\left(\frac{\beta_m x}{2}\right)}{1+2f\sin^2\left(\frac{\beta_m x}{2}\right)}\right|\le 1$$

- Conclusion: Crank Nicholson method is unconditionally stable
- Large f values produce physically unreasonable solutions, however

Convection Equation

- Look at simplest convection equation with a constant velocity
- Examine stability of FTCS (forward time, central space differences)

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0 \qquad u_i^{n+1}=u_i^n-\frac{c\Delta t}{2\Delta x}\left[u_{i+1}^n-u_{i-1}^n\right]$$

$$G=\left|e^{a\Delta t}\right|=\left|1-\frac{c\Delta t}{\Delta x}\left[1-2\sin^2\left(\frac{\beta_m\Delta x}{2}\right)\right]\right|\le 1$$

Convection Equation II

- Growth factor limit requires an equation that cannot be satisfied and therefore FTCS is unstable
- Lax proposed replacing u_i^n by average of two nearby space nodes (i ± 1)

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

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

Lax’s Method

- Lax’s Method is stable if the Courant number, $N_C = c\Delta x/\Delta t \le 1$

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

$$G=\left|e^{a\Delta t}\right|=\sqrt{\left[1+\left(N_C^2-1\right)\sin^2(\beta_m x)\right]}\le 1$$