Transient CFD

Larry Caretto
Mechanical Engineering 692

Computational Fluid Dynamics

April 21, 2010

Introduction

- Schedule for last two weeks of semester
- Review transient conduction
- Density-based approaches to transient CFD
- Pressure-based approaches to transient CFD
- Fluent options for transient CFD

Remainder of Course

- April 26-28: Non-uniform grids
- May 3-5: Student presentations
  – Have a total of 150 minutes of class time for 14 students
  – Will accept volunteers for May 3 or draw names out of a hat to balance the presentations

Review Conduction Equation

- Apply difference formulas derived for ordinary derivatives to partial derivatives
- 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

$$\left[\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}\right]_i^n$$

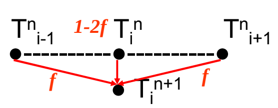
$$\left.\frac{\partial T}{\partial t}\right|_i^n = \frac{T_i^{n+1} - T_i^n}{\Delta t} + O(\Delta t) \quad and \quad \left.\frac{\partial^2 T}{\partial x^2}\right|_i^n = \frac{T_{i+1}^n + T_{i-1}^n - 2T_i^n}{(\Delta x)^2} + O[(\Delta x)^2]$$

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)

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 $f = \alpha\Delta t/(\Delta x)^2 \le 0.5$
- This imposes a time step limit that may be less than the limit required for accuracy in the solution

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

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

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

Crank-Nicholson Equations

- Consider case where boundary temperatures T_0 and T_N are specified
- Rewrite equations in matrix form to show tridiagonal structure $f = \alpha\Delta t/(\Delta x)^2$

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

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

Crank Nicholson Results

- Results for 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	-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 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

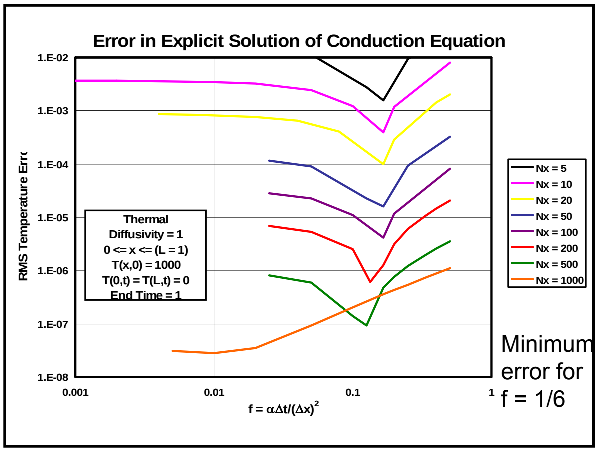
DuFort Frankel

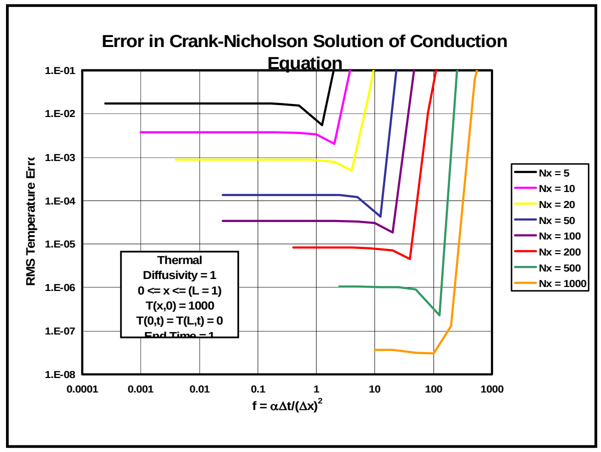
- Rearrange and introduce f = $\alpha\Delta t/(\Delta x)^2$

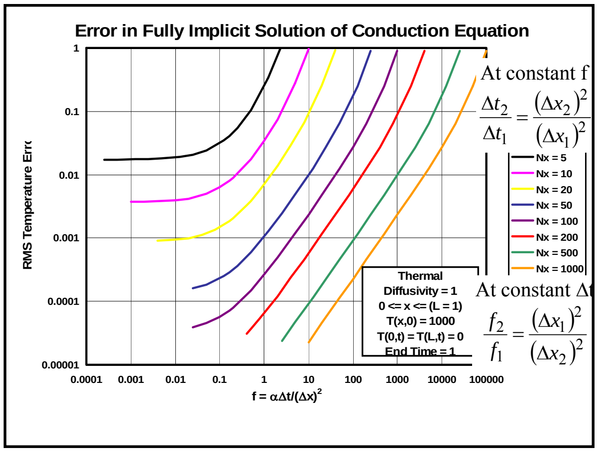
$$T_i^{n+1} - T_i^{n-1} = \frac{2\alpha\Delta t}{(\Delta x)^2}\left(T_{i+1}^n + T_{i-1}^n - T_i^{n+1} - T_i^{n-1}\right) = 2f\left(T_{i+1}^n + T_{i-1}^n - T_i^{n+1} - T_i^{n-1}\right)$$

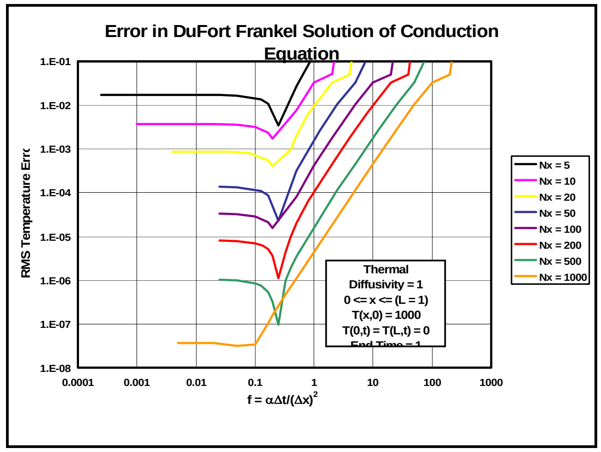
$$(1 + 2f)T_i^{n+1} = T_i^{n-1}(1 - 2f) + 2f\left(T_{i+1}^n + T_{i-1}^n\right)$$

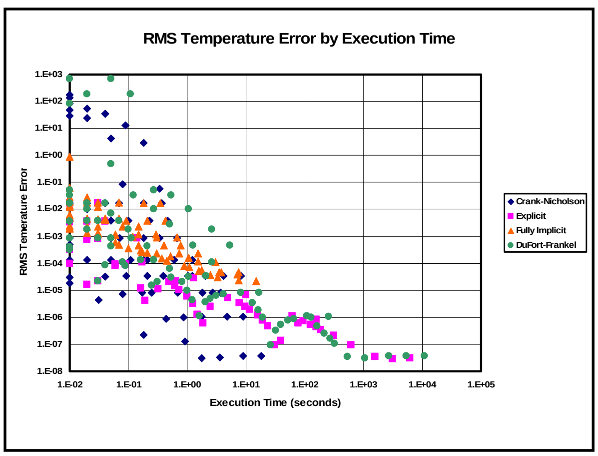
- Result is explicit for values at time n+1
- Explicit start required to get first set of values at time n-1











von Neumann Stability

- Examines stability due to differential equation alone
- 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
  - Determine growth factor G = absolute value of (error at t + Δt) / (error at t)
  - Want G ≤ 1 for stability

California State University Northridge

18

von Neumann Results

- Explicit (FTCS) method

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

  – G ≤ 1 if f ≤ 0.5
- Crank-Nicholson method

Unconditionally stable

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

California State University Northridge 19

Convection Equation $\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$

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

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

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

California State University Northridge 20

Transient Convection Diffusion

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = \alpha\frac{\partial^2 \phi}{\partial x^2}$$

- Stability of FTCS algorithm

$$\phi_k^{n+1} = \phi_k^n - \frac{1}{2}\frac{c\Delta t}{\Delta x}\left(\phi_{k+1}^n - \phi_{k-1}^n\right) + \frac{\alpha\Delta t}{(\Delta x)^2}\left(\phi_{k+1}^n + \phi_{k-1}^n - 2\phi_k^n\right)$$

$$G = 1 - 2\frac{\alpha\Delta t}{(\Delta x)^2}\left(1 - \cos\beta_m x\right) - i\frac{c\Delta t}{\Delta x}\sin\beta_m x$$

- Stability requires $c\Delta t/\Delta x \le 1$, $\alpha\Delta t/(\Delta x)_2 \le 0.5$ and $Pe_{cell} = c\Delta x/\alpha \le 2$
  – Last equation is limit for central difference

California State University Northridge 21

Density-based Solvers

- Typically used for compressible flows in aerodynamics calculations
- Split stress term, σ_{ij}, into sum of pressure and viscous stress, $\tau_{ij} = \sigma_{ij} + P\delta_{ij}$

$$\tau_{ij} = \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right] + (\kappa - \frac{2}{3}\mu)\Delta\delta_{ij}$$

$$\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} + \rho B_j$$

California State University Northridge 22

Density-based Solvers II

- Equation without summation convention
- General direction j equation

$$\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho uu_j}{\partial x} + \frac{\partial \rho vu_j}{\partial y} + \frac{\partial \rho wu_j}{\partial z} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{xj}}{\partial x} + \frac{\partial \tau_{yj}}{\partial y} + \frac{\partial \tau_{zj}}{\partial z} + \rho B_x$$

- x-direction equation

$$\frac{\partial \rho u}{\partial t} + \frac{\partial\left(\rho uu + p - \tau_{xx}\right)}{\partial x} + \frac{\partial\left(\rho vu - \tau_{yx}\right)}{\partial y} + \frac{\partial\left(\rho wu - \tau_{zx}\right)}{\partial z} = \rho B_x$$

California State University Northridge 23

Density-based Solvers III

- Cast continuity, momentum, total energy, and species balance into form of vector equation

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \mathbf{H}$$

- Each conservation equation is one component of the vector equation
- Get components by reviewing equations

California State University Northridge 24

Density-based Solvers IV

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}+\frac{\partial \mathbf{G}}{\partial z}=\mathbf{H}$$

$$\mathbf{U}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho(e+V^2/2)\\ \rho W^{(K)}\end{bmatrix}=\begin{bmatrix}U_1\\U_2\\U_3\\U_4\\U_5\\U_6\end{bmatrix}\quad \mathbf{H}=\begin{bmatrix}0\\ \rho B_x\\ \rho B_y\\ \rho B_z\\ \rho(uB_x+vB_y+wB_z)\\ r^{(K)}\end{bmatrix}=\begin{bmatrix}h_1\\h_2\\h_3\\h_4\\h_5\\h_6\end{bmatrix}$$

Northridge 25

Density-based Solvers V

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}+\frac{\partial \mathbf{G}}{\partial z}=\mathbf{H}$$

$$\mathbf{E}=\begin{bmatrix}\rho u\\ \rho uu+p-\tau_{xx}\\ \rho uv-\tau_{xy}\\ \rho wu-\tau_{xz}\\ u[\rho(e+V^2/2)+p]-u\tau_{xx}-v\tau_{xy}-w\tau_{xz}+q_x\\ \rho uW^{(K)}+j_x^{(K)}\end{bmatrix}=\begin{bmatrix}e_1\\e_2\\e_3\\e_4\\e_5\\e_6\end{bmatrix}$$

Northridge 26

Density-based Solvers VI

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}+\frac{\partial \mathbf{G}}{\partial z}=\mathbf{H}$$

$$\mathbf{F}=\begin{bmatrix}\rho v\\ \rho uv-\tau_{yx}\\ \rho vv+p-\tau_{yy}\\ \rho wv-\tau_{yz}\\ v[\rho(e+V^2/2)+p]-u\tau_{yx}-v\tau_{yy}-w\tau_{yz}+q_y\\ \rho vW^{(K)}+j_y^{(K)}\end{bmatrix}=\begin{bmatrix}f_1\\f_2\\f_3\\f_4\\f_5\\f_6\end{bmatrix}$$

Northridge 27

Density-based Solvers VII

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}+\frac{\partial \mathbf{G}}{\partial z}=\mathbf{H}$$

$$\mathbf{G}=\begin{bmatrix}\rho w\\ \rho uw-\tau_{zx}\\ \rho vw-\tau_{zy}\\ \rho ww+p-\tau_{zz}\\ w[\rho(e+V^2/2)+p]-u\tau_{zx}-v\tau_{zy}-w\tau_{zz}+q_z\\ \rho wW^{(K)}+j_z^{(K)}\end{bmatrix}=\begin{bmatrix}g_1\\g_2\\g_3\\g_4\\g_5\\g_6\end{bmatrix}$$

Northridge 28

Density-based Solvers VIII

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{E}}{\partial x}+\frac{\partial \mathbf{F}}{\partial y}+\frac{\partial \mathbf{G}}{\partial z}=\mathbf{H}$$

- Compute vectors **E**, **F**, **G**, and **H** from flow variables and use numerical integration over time step to get **U**
- Update flow variables from components of **U** vector, U_k.
  - These are ***not*** velocity components
  - Details next chart

Northridge 29

Density-based Solvers IX

$$\rho=U_1$$
$$u=\frac{U_2}{U_1}$$
$$v=\frac{U_3}{U_1}$$
$$w=\frac{U_4}{U_1}$$
$$e=\frac{U_5}{U_1}-\frac{1}{2U_1^2}\left(U_2^2+U_3^2+U_4^2\right)$$
$$W^{(K)}=\frac{U_6}{U_1}$$

Northridge 30

Density-based Solvers X

- Density-based solvers traditionally used for compressible flows
  - Not accurate for low Mach numbers
  - Fluent uses a transformation to allow density based solvers for low Mach number flows
- Density-based solvers can be implicit or explicit
  - Implicit allows longer time steps while preserving stability at higher Courant numbers

California State University Northridge 31

Pressure-based Solvers

- Modify solvers such as those based on SIMPLE to handle transient flows
  - Steady equations used previously

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0$$

$$\frac{\partial \rho uu}{\partial x} + \frac{\partial \rho vu}{\partial y} = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\mu\frac{\partial u}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial u}{\partial y} + S^{(u)}$$

$$\frac{\partial \rho uv}{\partial x} + \frac{\partial \rho vv}{\partial y} = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\mu\frac{\partial v}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial v}{\partial y} + S^{(v)}$$

California State University Northridge 32

Pressure-based Solvers II

- Transient two-dimensional flow equations
  - Have to convert time derivative into finite volume form

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho uu}{\partial x} + \frac{\partial \rho vu}{\partial y} = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\mu\frac{\partial u}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial u}{\partial y} + S^{(u)}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho vv}{\partial y} = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\mu\frac{\partial v}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial v}{\partial y} + S^{(v)}$$

California State University Northridge 33

Pressure-based Solvers III

- Integration over volume and time

●--------------|--------------●-----------------|-----------------●

$x_W = x_{i-1}$ $x_w = x_{i-1/2}$ $x_P = x_i$ $x_e = x_{i+1/2}$ $x_E = x_{i+1}$

$$\int_{\Delta V}\int_t^{t+\Delta t}\frac{\partial \rho\phi}{\partial t}dVdt + \int_{\Delta V}\int_t^{t+\Delta t}\left(\frac{\partial \rho u\phi}{\partial x} + \frac{\partial \rho v\phi}{\partial y}\right)dVdt =$$

$$\int_{\Delta V}\int_t^{t+\Delta t}\left(\frac{\partial}{\partial x}\mu\frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial \phi}{\partial y} + S^{(\phi)}\right)dVdt$$

$$\int_{\Delta V}\int_t^{t+\Delta t}\frac{\partial \rho\phi}{\partial t}dVdt \approx \left[(\rho\phi)_{P,t+\Delta t} - (\rho\phi)_{P,t}\right]\Delta V$$

California State University Northridge 34

Pressure-based Solvers IV

- Integration of spatial derivatives

●--------------|--------------●-----------------|-----------------●

$x_W = x_{i-1}$ $x_w = x_{i-1/2}$ $x_P = x_i$ $x_e = x_{i+1/2}$ $x_E = x_{i+1}$

$$\int_{\Delta V}\int_t^{t+\Delta t}\left(\frac{\partial \rho u\phi}{\partial x} + \frac{\partial \rho v\phi}{\partial y}\right)dVdt = \Delta t\int_{\Delta V}\left(\frac{\partial \overline{\rho u\phi}}{\partial x} + \frac{\partial \overline{\rho v\phi}}{\partial y}\right)dV$$

$$\int_{\Delta V}\int_t^{t+\Delta t}\left(\frac{\partial}{\partial x}\mu\frac{\partial \phi}{\partial x} + \frac{\partial}{\partial y}\mu\frac{\partial \phi}{\partial y} + S^{(\phi)}\right)dVdt =$$

$$\Delta t\int_{\Delta V}\left(\frac{\partial}{\partial x}\overline{\mu}\frac{\partial \overline{\phi}}{\partial x} + \frac{\partial}{\partial y}\overline{\mu}\frac{\partial \overline{\phi}}{\partial y} + \overline{S^{(\phi)}}\right)dV$$

California State University Northridge 35

Pressure-based Solvers V

- Combine all integration results

$$\left[(\rho\phi)_{P,t+\Delta t} - (\rho\phi)_{P,t}\right]\Delta V + \Delta t\int_{\Delta V}\left(\frac{\partial \overline{\rho u\phi}}{\partial x} + \frac{\partial \overline{\rho v\phi}}{\partial y}\right)dV$$

$$= \Delta t\int_{\Delta V}\left(\frac{\partial}{\partial x}\overline{\mu}\frac{\partial \overline{\phi}}{\partial x} + \frac{\partial}{\partial y}\overline{\mu}\frac{\partial \overline{\phi}}{\partial y} + \overline{S^{(\phi)}}\right)dV$$

- Terms multiplied by Δt are almost the same as steady-state terms
  - Here these terms represent a suitable average over the time step

California State University Northridge 36

Pressure-based Solvers VI

- Use steady-state results

$$\frac{\left[(\rho\phi)_{P,t+\Delta t} - (\rho\phi)_{P,t}\right]\Delta V}{\Delta t} = -\int_{\Delta V}\left(\frac{\partial \overline{\rho u \phi}}{\partial x} + \frac{\partial \overline{\rho v \phi}}{\partial y}\right)dV$$
$$+ \int_{\Delta V}\left(\frac{\partial}{\partial x}\overline{\mu}\frac{\partial \overline{\phi}}{\partial x} + \frac{\partial}{\partial y}\overline{\mu}\frac{\partial \overline{\phi}}{\partial y} + \overline{S^{(\phi)}}\right)dV$$

$$\frac{\left[(\rho\phi)_{P,t+\Delta t} - (\rho\phi)_{P,t}\right]\Delta V}{\Delta t} =$$
$$\overline{a_N}\,\overline{\phi_N} + \overline{a_S}\,\overline{\phi_S} + \overline{a_E}\,\overline{\phi_E} + \overline{a_W}\,\overline{\phi_W} - \overline{a_P}\,\overline{\phi_P} + \overline{S^{(\phi)}}$$

California State University Northridge 37

Pressure-based Solvers VII

- Transient finite-volume equation

$$\frac{\left[(\rho\phi)_{P,t+\Delta t} - (\rho\phi)_{P,t}\right]\Delta V}{\Delta t} =$$
$$\overline{a_N}\,\overline{\phi_N} + \overline{a_S}\,\overline{\phi_S} + \overline{a_E}\,\overline{\phi_E} + \overline{a_W}\,\overline{\phi_W} - \overline{a_P}\,\overline{\phi_P} + \overline{S^{(\phi)}}$$

$$\overline{a_N}\,\overline{\phi_N} + \overline{a_S}\,\overline{\phi_S} + \overline{a_E}\,\overline{\phi_E} + \overline{a_W}\,\overline{\phi_W} - \overline{a_{P,transient}}\,\overline{\phi_P} + \overline{S^{(\phi)}_{transient}} = 0$$

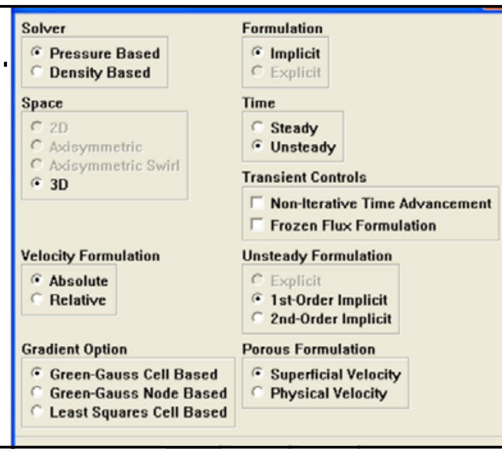
$$\overline{a_{P,transient}} = \overline{a_P} + \frac{\rho_{P,t+\Delta t}\Delta V}{\Delta t} \qquad \overline{S^{(\phi)}_{transient}} = \overline{S^{(\phi)}} + \frac{\rho\phi_{P,t}\Delta V}{\Delta t}$$

California State University Northridge 38

What is Time Average?

- Have same choices used for conduction equation
  – Explicit – use values at old time step
  – Implicit – use values at new time step
  – Crank-Nicholson – use average of values at old and new time steps
- Can also use more accurate time derivatives
- Fluent has various options

California State University Northridge 39



Explicit or Implicit?

- Explicit stability limits on time step (set by the local Courant number, $u\Delta x/\alpha$)
- The Δt required for stability is usually much lower than the Δt for accuracy
- Implicit algorithms will generally take less computer time
- Moving waves (*e. g.* shock waves) require small time steps so that explicit algorithms are preferred here
  – Available in Fluent only with density solver

California State University Northridge 41

Other Fluent Options

- Non-iterative time advancement – simplifies iterations to reduce computer time for solution
  – Does not do “outer” iteration
- Frozen-flux formulation uses a_K coefficients from previous time step
  – Does not update during iterations
  – Another item to save computer time

California State University Northridge 42