MASSACHUSETTS INSTITUTE OF TECHNOLOGY
DEPARTMENT OF MECHANICAL ENGINEERING
CAMBRIDGE, MASSACHUSETTS 02139

2.29 NUMERICAL FLUID MECHANICS — SPRING 2015

EQUATION SHEET – Quiz 2

Number Representation

- Floating Number Representation: $x = m\, b^e$, $b^{-1} \le m < b^0$

Truncation Errors and Error Analysis $y = f(x_1, x_2, x_3, ..., x_n)$

- Taylor Series:

$$f(x_{i+1}) = f(x_i) + \Delta x\, f'(x_i) + \frac{\Delta x^2}{2!} f''(x_i) + \frac{\Delta x^3}{3!} f'''(x_i) + ... + \frac{\Delta x^n}{n!} f^n(x_i) + R_n$$

$$R_n = \frac{\Delta x^{n+1}}{n+1!} f^{(n+1)}(\xi)$$

- The Differential Error (general error propagation) Formula: $\varepsilon_y \le \sum_{i=1}^{n} \left| \frac{\partial f(x_1, ..., x_n)}{\partial x_i} \right| \varepsilon_i$
- The Standard Error (statistical formula): $E(\Delta_s y) \simeq \sqrt{\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 \varepsilon_i^2}$
- Condition Number of *f(x)*: $K_p = \left| \frac{\bar{x}\, f'(\bar{x})}{f(\bar{x})} \right|$

Roots of nonlinear equations ($x_{n+1} = x_n - h(x_n) f(x_n)$ **)**

- Bisection: successive division of bracket in half, next bracket based on sign of $f(x_1^{n+1}) f(x_{\text{mid-point}}^{n+1})$
- False-Position (Regula Falsi): $x_r = x_U - \frac{f(x_U)\,(x_L - x_U)}{f(x_L) - f(x_U)}$
- Fixed Point Iteration (General Method or Picard Iteration): $x_{n+1} = g(x_n)$ or $x_{n+1} = x_n - h(x_n) f(x_n)$
- Newton Raphson: $x_{n+1} = x_n - \frac{1}{f'(x_n)} f(x_n)$
- Secant Method: $x_{n+1} = x_n - \frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} f(x_n)$
- Order of convergence p: Defining $e_n = x_n - x^e$, the order of convergence p exists if there exist a constant C≠0 such that: $\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^p} = C$

Conservation Law for a scalar ϕ, in integral and differential forms:

- $$\left\{\frac{d}{dt}\int_{CM}\rho\phi dV=\right\}\quad \frac{d}{dt}\int_{CV_{\text{fixed}}}\rho\phi dV+\underbrace{\int_{CS}\rho\phi\,(\vec{v}.\vec{n})dA}_{\substack{\text{Advective fluxes}\\ \text{(Adv.\& diff. ="convection" fluxes)}}}=\underbrace{-\int_{CS}\vec{q}_{\phi}.\vec{n}\,dA}_{\substack{\text{Other transports}\\ \text{(diffusion, etc)}}}+\underbrace{\sum\int_{CV_{\text{fixed}}}s_{\phi}\,dV}_{\substack{\text{Sum of sources and}\\ \text{sinks terms (reactions, etc)}}}$$
- $$\frac{\partial\rho\phi}{\partial t}+\nabla.(\rho\phi\vec{v})=-\nabla.\,\vec{q}_{\phi}+s_{\phi}$$

Linear Algebraic Systems:

- Gauss Elimination: reduction, $m_{ik}=\dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$, $\quad a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}\,a_{kj}^{(k)}$, $\quad b_{i}^{(k+1)}=b_{i}^{(k)}-m_{ik}\,b_{k}^{(k)}$,

  followed by a back-substitution. $x_k=\left(b_k-\sum_{j=k+1}^{n}a_{kj}^{(k)}x_j\right)\Big/a_{kk}^{(k)}$
- LU decomposition: **A**=**LU**, $a_{ij}=\sum_{k=1}^{\min(i,j)}m_{ik}\,a_{kj}^{(k)}$
- Choleski Factorization: **A**=**R*****R**, where **R** is upper triangular and **R*** its conjugate transpose.
- Condition number of a linear algebraic system: $K(\mathbf{A})=\left\|\mathbf{A}^{-1}\right\|\,\left\|\mathbf{A}\right\|$
- A banded matrix of p super-diagonals and q sub-diagonals has a bandwidth $w=p+q+1$
- $\sum_{i=1}^{n}i=\dfrac{n(n+1)}{2}$ and $\sum_{i=1}^{n}i^2=\dfrac{n(n+1)(2n+1)}{6}$
- Eigendecomposition: $\mathbf{Ax}=\lambda\mathbf{x}$ and $\det(\mathbf{A}-\lambda\mathbf{I})=0$
- Norms:

$$\left\|\mathbf{A}\right\|_1=\max_{1\le j\le n}\sum_{i=1}^{m}\left|a_{ij}\right| \qquad \text{“Maximum Column Sum”}$$

$$\left\|\mathbf{A}\right\|_\infty=\max_{1\le i\le m}\sum_{j=1}^{n}\left|a_{ij}\right| \qquad \text{“Maximum Row Sum”}$$

$$\left\|\mathbf{A}\right\|_F=\sqrt{\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left|a_{ij}\right|^2\right)} \qquad \text{“Frobenius norm” (or “Euclidean norm”)}$$

$$\left\|\mathbf{A}\right\|_2=\sqrt{\lambda_{\max}\left\{\mathbf{A}^*\mathbf{A}\right\}} \qquad \text{“}L-2\text{ norm” (or “spectral norm”)}$$

Iterative Methods for solving linear algebraic systems: $\mathbf{x}^{k+1}=\mathbf{B}\,\mathbf{x}^{k}+\mathbf{c}\qquad k=0,1,2,...$

- Necessary and sufficient condition for convergence:
  $\rho(\mathbf{B})=\max_{i=1...n}\left|\lambda_i\right|<1$, where $\lambda_i=$ eigenvalue$(\mathbf{B}_{n\times n})$
- Jacobi’s method: $\mathbf{x}^{k+1}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{U})\,\mathbf{x}^{k}+\mathbf{D}^{-1}\mathbf{b}$
- Gauss-Seidel method: $\mathbf{x}^{k+1}=-(\mathbf{D}+\mathbf{L})^{-1}\mathbf{U}\,\mathbf{x}^{k}+(\mathbf{D}+\mathbf{L})^{-1}\mathbf{b}$
- SOR Method: $\mathbf{x}^{k+1}=(\mathbf{D}+\omega\mathbf{L})^{-1}[-\omega\mathbf{U}+(1-\omega)\mathbf{D}]\mathbf{x}^{k}+\omega(\mathbf{D}+\omega\mathbf{L})^{-1}\mathbf{b}$
- Steepest Descent Gradient Method: $\mathbf{x}_{i+1}=\mathbf{x}_i+\dfrac{\mathbf{r}_i^T\mathbf{r}_i}{\mathbf{r}_i^T\mathbf{A}\mathbf{r}_i}\mathbf{r}_i$, $\quad\mathbf{r}_i=\mathbf{b}-\mathbf{A}\mathbf{x}_i$
- Conjugate Gradient: $\mathbf{x}_{i+1}=\mathbf{x}_i+\alpha_i\,\mathbf{v}_i$ (α_i such that each $\mathbf{v}_i$ are generated by orthogonalization of residuum vectors and such that search directions are **A**-conjugate).

Finite Differences – PDE types (2nd order, 2D): $A\phi_{xx} + B\phi_{xy} + C\phi_{yy} = F(x, y, \phi, \phi_x, \phi_y)$

$B^2 - AC > 0$: hyperbolic; $B^2 - AC = 0$: parabolic; $B^2 - AC < 0$: elliptic

Finite Differences – Error Types and Discretization Properties ($\mathcal{L}(\phi) = 0$, $\hat{\mathcal{L}}_{\Delta x}(\hat{\phi}) = 0$)

- Consistency: $\left| \mathcal{L}(\phi) - \hat{\mathcal{L}}_{\Delta x}(\phi) \right| \to 0$ when $\Delta x \to 0$
- Truncation error: $\tau_{\Delta x} = \mathcal{L}(\phi) - \hat{\mathcal{L}}_{\Delta x}(\phi) \to O(\Delta x^p)$ for $\Delta x \to 0$
- Error equation: $\tau_{\Delta x} = \mathcal{L}(\phi) - \hat{\mathcal{L}}_{\Delta x}(\hat{\phi} + \varepsilon) = -\hat{\mathcal{L}}_{\Delta x}(\varepsilon)$ (for linear systems)
- Stability: $\left\| \hat{\mathcal{L}}_{\Delta x}^{-1} \right\| <$ Const. (for linear systems)
- Convergence: $\|\varepsilon\| \le \left\| \hat{\mathcal{L}}_{\Delta x}^{-1} \right\| \, \|\tau_{\Delta x}\| \le \alpha \, O(\Delta x^p)$

Finite Differences – General schemes and Higher Accuracy

Higher Order Accuracy Finite-difference based on Taylor Series: $\left(\dfrac{\partial^m u}{\partial x^m} \right)_j - \sum_{i=-r}^{s} a_i \, u_{j+i} = \tau_{\Delta x}$

Newton's interpolating polynomial formulas, equidistant sampling:

$$
\begin{aligned}
f(x) &= f_0 + \frac{\Delta f_0}{h}(x - x_0) + \frac{\Delta^2 f_0}{2! h^2}(x - x_0)(x - x_1) + \cdots \\
&+ \frac{\Delta^n f_0}{n! h^n}(x - x_0)(x - x_1)\cdots(x - x_{n-1}) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0)\cdots(x - x_n)
\end{aligned}
$$

Lagrange polynomial: $f(x) = \sum_{k=0}^{n} L_k(x) f(x_k)$ with $L_k(x) = \prod_{j=0, j\neq k}^{n} \dfrac{x - x_j}{x_k - x_j}$

Hermite Polynomials and Compact/Pade's Difference schemes: $\sum_{i=-r}^{s} b_i \left(\dfrac{\partial^m u}{\partial x^m} \right)_{j+i} - \sum_{i=-p}^{q} a_i \, u_{j+i} = \tau_{\Delta x}$

Finite Differences – Non-Uniform Grids, Grid Refinement and Error Estimation

For a centered-difference approximation of *f'(x)* over a 1D grid, contracting/expanding with a constant factor r_e, $\Delta x_{i+1} = r_e \, \Delta x_i$, the:

- Leading term of the truncation error is: $\tau_{\Delta x}^{r_e} \approx \dfrac{(1 - r_e)\, \Delta x_i}{2} f''(x_i)$
- Ratio of the two truncation errors at a common point is: $R \approx \dfrac{(1 + r_{e,h})^2}{r_{e,h}}$

Grid-Refinement and Error estimation:

- Estimate of the order of accuracy: $p \approx \log\left(\dfrac{u_{2\Delta x} - u_{4\Delta x}}{u_{\Delta x} - u_{2\Delta x}} \right) \Big/ \log 2$

- Discretization error on the grid Δx: $\varepsilon_{\Delta x} \approx \dfrac{u_{\Delta x} - u_{2\Delta x}}{2^p - 1}$

Richardson Extrapolation for the Trapezoidal Rule: $I = I(h_2) + \dfrac{I(h_2) - I(h_1)}{(h_1/h_2)^2 - 1} + O(h^4)$

"Romberg" Differentiation Algorithm: $D_{j,k} = \dfrac{4^{k-1} D_{j+1,k-1} - D_{j,k-1}}{4^{k-1} - 1}$

Finite Differences – Fourier Analysis and Error Analysis

Fourier transform of a generic PDE, $\dfrac{\partial f}{\partial t} = \dfrac{\partial^n f}{\partial x^n}$: With $f(x,t) = \sum_{k=-\infty}^{\infty} f_k(t)\, e^{ikx}$, one obtains:

$$\frac{d f_k(t)}{dt} = (ik)^n\ f_k(t) = \sigma f_k(t) \qquad \text{for } \sigma = (ik)^n$$

Finite Difference Methods – Effective wave number and speed

Effective Wave Number: $\left(\dfrac{\partial\, e^{ikx}}{\partial x}\right)_j = i\, k_{\text{eff}}\ e^{ikx_j}$ (for CDS, 2[nd] order, $k_{\text{eff}} = \dfrac{\sin(k\Delta x)}{\Delta x}$)

Effective Wave Speed (for linear convection eq., $\dfrac{\partial f}{\partial t} + c\dfrac{\partial f}{\partial x} = 0$):

$$\frac{d f_k^{num.}}{dt} = -f_k^{num.}(t)\, c\, i\, k_{\text{eff}} \Rightarrow f_{\text{numerical}}(x,t) = \sum_{k=-\infty}^{\infty} f_k(0)\, e^{ikx - ik_{\text{eff}}\, t} = \sum_{k=-\infty}^{\infty} f_k(0)\, e^{ik(x - c_{\text{eff}}\, t)} \quad \Rightarrow \frac{c_{eff}}{c} = \frac{\sigma_{eff}}{\sigma} = \frac{k_{eff}}{k}$$

Finite Difference Methods – Stability

Von Neumann: $\varepsilon(x,t) = \sum_{\beta=-\infty}^{\infty} \varepsilon_\beta(t)\, e^{i\beta x}$, $\varepsilon_\beta(t)\, e^{i\beta x} \approx e^{\gamma t}\ e^{i\beta x}$ (γ in general complex, function of β)

Strict condition for stability:

$$\left|e^{\gamma\Delta t}\right| \le 1 \ \text{ or for } \xi = e^{\gamma\Delta t},\ |\xi| \le 1 \quad \forall\beta \ \text{ (for the error not to grow in time)}$$

Useful trigonometric relations:

$$e^{ix} + e^{-ix} = 2\cos(x), \quad e^{ix} - e^{-ix} = 2\sin(x) \quad \text{and} \quad 1 - \cos(x) = 2\sin^2(x/2)$$

CFL condition: "Numerical domain of dependence of FD scheme must include the mathematical domain of dependence of the corresponding PDE"

Forward finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

Figure 23.1 Chapra and Canale

Forward Differences

	Error
First Derivative	
$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
$f'(x_i) = \frac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \frac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	$O(h)$
$f''(x_i) = \frac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \frac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	$O(h)$
$f'''(x_i) = \frac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f''''(x_i) = \frac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	$O(h)$
$f''''(x_i) = \frac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

Backward Differences

	Error
First Derivative	
$f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
$f'(x_i) = \dfrac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	$O(h)$
$f''(x_i) = \dfrac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	$O(h)$
$f'''(x_i) = \dfrac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	$O(h)$
$f''''(x_i) = \dfrac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$

FIGURE 23.2
Backward finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

FIGURE 23.3
Centered finite-divided-difference formulas: two versions are presented for each derivative. The latter version incorporates more terms of the Taylor series expansion and is, consequently, more accurate.

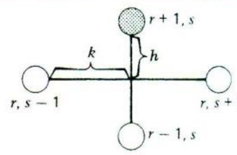
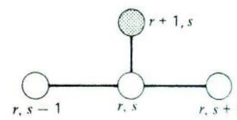
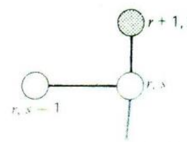
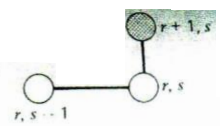
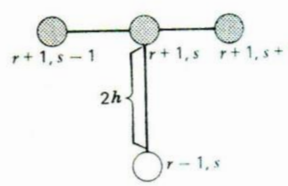
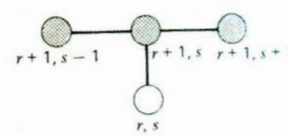
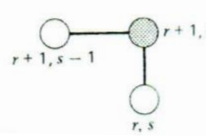
Centered Differences

	Error
First Derivative	
$f'(x_i) = \dfrac{f(x_{i+1}) - f(x_{i-1})}{2h}$	$O(h^2)$
$f'(x_i) = \dfrac{-f(x_{i+2}) + 8f(x_{i+1}) - 8f(x_{i-1}) + f(x_{i-2})}{12h}$	$O(h^4)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2}$	$O(h^2)$
$f''(x_i) = \dfrac{-f(x_{i+2}) + 16f(x_{i+1}) - 30f(x_i) + 16f(x_{i-1}) - f(x_{i-2})}{12h^2}$	$O(h^4)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_{i+2}) - 2f(x_{i+1}) + 2f(x_{i-1}) - f(x_{i-2})}{2h^3}$	$O(h^2)$
$f'''(x_i) = \dfrac{-f(x_{i+3}) + 8f(x_{i+2}) - 13f(x_{i+1}) + 13f(x_{i-1}) - 8f(x_{i-2}) + f(x_{i-3})}{8h^3}$	$O(h^4)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_{i+2}) - 4f(x_{i+1}) + 6f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{h^4}$	$O(h^2)$
$f''''(x_i) = \dfrac{-f(x_{i+3}) + 12f(x_{i+2}) + 39f(x_{i+1}) + 56f(x_i) - 39f(x_{i-1}) + 12f(x_{i-2}) + f(x_{i-3})}{6h^4}$	$O(h^4)$

Finite Difference Methods – Schemes for specific PDE types

Hyperbolic, 1D: $u_x + b\,u_y = 0$

TABLE 6.1. Various Finite Difference Forms for $u_x + bu_y = 0$

Finite Difference Form	Symbolic Difference	Stability	Explicit or Implicit	Computational Molecule
$\frac{u_{r+1,s}-u_{r-1,s}}{2h}+b\frac{u_{r,s+1}-u_{r,s-1}}{2k}=0$	C−C	$\frac{bh}{k}\leq 1$	Explicit	
$\frac{u_{r+1,s}-u_{r,s}}{h}+b\frac{u_{r,s+1}-u_{r,s-1}}{2k}=0$	F−C	Unstable	Explicit	
$\frac{u_{r+1,s}-u_{r-1,s}}{2h}+b\frac{u_{r,s}-u_{r,s-1}}{k}=0$	C−B	Unstable	Explicit	
$\frac{u_{r+1,s}-u_{r,s}}{h}+b\frac{u_{r,s}-u_{r,s-1}}{k}=0$	F−B	$\frac{bh}{k}\leq 1$	Explicit	
$\frac{u_{r+1,s}-u_{r-1,s}}{2h}+b\frac{u_{r+1,s+1}-u_{r+1,s-1}}{2k}=0$	$C-C_{r+1}$	Stable	Implicit	
$\frac{u_{r+1,s}-u_{r,s}}{h}+b\frac{u_{r+1,s+1}-u_{r+1,s-1}}{2k}=0$	$F-C_{r+1}$	Stable	Implicit	
$\frac{u_{r+1,s}-u_{r,s}}{h}+b\frac{u_{r+1,s}-u_{r+1,s-1}}{k}=0$	$F-B_{r+1}$	Stable	Explicit Implicit	

Elliptic PDEs: 2D Laplace/Poisson Eq. on a Cartesian-orthogonal uniform grid

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

SOR, Gauss-Seidel: $u_{i,j}^{k+1} = (1-\omega)\,u_{i,j}^{k} + \omega\dfrac{u_{i+1,j}^{k}+u_{i-1,j}^{k+1}+u_{i,j+1}^{k}+u_{i,j-1}^{k+1}-h^2 g_{i,j}}{4}$

Parabolic PDEs: 2D Heat Conduction Eq. on a Cartesian-orthogonal grid

- Explicit: $\dfrac{T_{i,j}^{n+1}-T_{i,j}^{n}}{\Delta t}=c^2\,\dfrac{T_{i-1,j}^{n}-2T_{i,j}^{n}+T_{i+1,j}^{n}}{\Delta x^2}+c^2\,\dfrac{T_{i,j-1}^{n}-2T_{i,j}^{n}+T_{i,j+1}^{n}}{\Delta y^2}$

- Crank-Nicolson Implicit (for $\Delta x=\Delta y$, with $r=\dfrac{\Delta t\, c^2}{\Delta x^2}$):

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

- ADI:

$$\frac{T_{i,j}^{n+1/2}-T_{i,j}^{n}}{\Delta t/2}=c^2\,\frac{T_{i-1,j}^{n}-2T_{i,j}^{n}+T_{i+1,j}^{n}}{\Delta x^2}+c^2\,\frac{T_{i,j-1}^{n+1/2}-2T_{i,j}^{n+1/2}+T_{i,j+1}^{n+1/2}}{\Delta y^2}$$

$$\frac{T_{i,j}^{n+1}-T_{i,j}^{n+1/2}}{\Delta t/2}=c^2\,\frac{T_{i-1,j}^{n+1}-2T_{i,j}^{n+1}+T_{i+1,j}^{n+1}}{\Delta x^2}+c^2\,\frac{T_{i,j-1}^{n+1/2}-2T_{i,j}^{n+1/2}+T_{i,j+1}^{n+1/2}}{\Delta y^2}$$

(for $\Delta x=\Delta y$): $-rT_{i,j-1}^{n+1/2}+2(1+r)T_{i,j}^{n+1/2}-rT_{i,j+1}^{n+1/2}=rT_{i-1,j}^{n}+2(1-r)T_{i,j}^{n}+rT_{i+1,j}^{n}$

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

Finite Volume Methods: $V\dfrac{d\overline{\Phi}}{dt}+\int_S \vec{F}_\phi.\vec{n}\,dA=S_\phi$, where $\overline{\Phi}=\dfrac{1}{V}\int_V \rho\phi dV$ and $S_\phi=\int_V s_\phi\,dV$

Cartesian grids

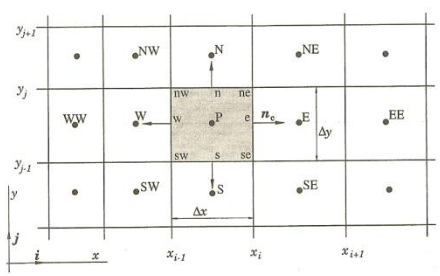


Fig. 4.2. A typical CV and the notation used for a Cartesian 2D grid

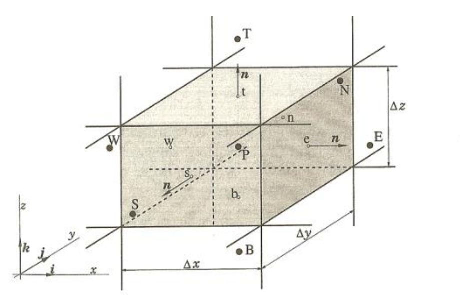


Fig. 4.3. A typical CV and the notation used for a Cartesian 3D grid

- Surface Integrals: $F_e=\int_{S_e} f_\phi\,dA$
  - 2D problems (1D surface integrals)
    - Midpoint rule (2nd order): $F_e=\int_{S_e} f_\phi\,dA=\overline{f}_e S_e=f_e S_e+O(\Delta y^2)\approx f_e S_e$
    - Trapezoid rule (2nd order): $F_e=\int_{S_e} f_\phi\,dA\approx S_e\dfrac{(f_{ne}+f_{se})}{2}+O(\Delta y^2)$
    - Simpson's rule (4th order): $F_e=\int_{S_e} f_\phi\,dA\approx S_e\dfrac{(f_{ne}+4f_e+f_{se})}{6}+O(\Delta y^4)$
  - 3D problems (2D surface integrals)
    - Midpoint rule (2nd order): $F_e=\int_{S_e} f_\phi\,dA\approx S_e f_e \quad +O(\Delta y^2,\Delta z^2)$

- Volume Integrals: $S_\phi=\int_V s_\phi\,dV$, $\overline{\Phi}=\dfrac{1}{V}\int_V \rho\phi dV$
  - 2D/3D problems, Midpoint rule (2nd order): $S_P=\int_V s_\phi\,dV=\overline{s}_P\,V\approx s_P\,V$

- 2D, bi-quadratic (4th order, Cart.): $S_P = \frac{\Delta x\,\Delta y}{36}\left[16 s_P + 4 s_s + 4 s_n + 4 s_w + 4 s_e + s_{se} + s_{sw} + s_{ne} + s_{nw}\right]$

- Interpolations / Differentiations (obtain fluxes "$F_e = f(\phi_e)$" as a function of cell-average values)
  - Upwind Interpolation (UDS): $\phi_e = \begin{cases} \phi_P & \text{if } (\vec{v}.\vec{n})_e > 0 \\ \phi_E & \text{if } (\vec{v}.\vec{n})_e < 0 \end{cases}$
  - Linear Interpolation (CDS): $\phi_e = \phi_E \lambda_e + \phi_P (1-\lambda_e)$ where $\lambda_e = \frac{x_e - x_P}{x_E - x_P}$

$$\phi = \phi_E \lambda + \phi_P (1-\lambda),\ \text{with } \lambda = \frac{x - x_P}{x_E - x_P} \Rightarrow \left.\frac{\partial \phi}{\partial x}\right|_e \approx \frac{\phi_E - \phi_P}{x_E - x_P}$$

  - Quadratic Upwind interpolation (QUICK): $\phi_e = \phi_U + g_1(\phi_D - \phi_U) + g_2(\phi_U - \phi_{UU})$

For uniform grids, $\phi_e = \frac{6}{8}\phi_U + \frac{3}{8}\phi_D - \frac{1}{8}\phi_{UU} - \frac{3\Delta x^3}{48}\left.\frac{\partial^3 \phi}{\partial x^3}\right|_D + R_3$

  - Higher order schemes:

    For example, for $\phi(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$,

    Convective fluxes $\phi_e = \frac{27\phi_P + 27\phi_E - 3\phi_W - 3\phi_{EE}}{48}$

    Diffusive Fluxes, for a uniform Cartesian grid: $\left.\frac{\partial \phi}{\partial x}\right|_e = \frac{27\phi_E - 27\phi_P + \phi_W - \phi_{EE}}{24\,\Delta x}$.

    For a compact high order scheme: $\phi_e = \frac{\phi_P + \phi_E}{2} + \frac{\Delta x}{8}\left(\left.\frac{\partial \phi}{\partial x}\right|_P - \left.\frac{\partial \phi}{\partial x}\right|_E\right) + O(\Delta x^4)$

Solution of the Navier-Stokes Equations

Newtonian fluid + incompressible + constant:

$$\frac{\partial \rho \vec{v}}{\partial t} + \nabla.(\rho \vec{v}\ \vec{v}) = -\nabla p + \mu \nabla^2 \vec{v} + \rho \vec{g}$$

$$\nabla.\vec{v} = 0$$

Strong conservative form, general Newtonian fluid:

$$\frac{\partial \rho v_i}{\partial t} + \nabla.(\rho v_i\ \vec{v}) = \nabla.\left(-p\,\vec{e}_i + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\vec{e}_j - \frac{2}{3}\mu \frac{\partial u_j}{\partial x_j}\vec{e}_i + \rho g_i x_i \vec{e}_i\right)$$

Kinetic energy equation, CV form:

$$\frac{\partial}{\partial t}\int_{CV} \rho \frac{\|\vec{v}\|^2}{2} dV = -\int_{CS} \rho \frac{\|\vec{v}\|^2}{2}(\vec{v}.\vec{n})dA - \int_{CS} p\,\vec{v}.\vec{n}\,dA + \int_{CS} (\vec{\vec{\varepsilon}}.\vec{v}).\vec{n}\,dA + \int_{CV}\left(-\vec{\vec{\varepsilon}} : \nabla\vec{v} + p\,\nabla.\vec{v} + \rho\vec{g}.\vec{v}\right)dV$$

Pressure equation: $\nabla.\nabla p = \nabla^2 p = -\nabla.\frac{\partial \rho \vec{v}}{\partial t} - \nabla.\left(\nabla.(\rho \vec{v}\ \vec{v})\right) + \nabla.\left(\mu \nabla^2 \vec{v}\right) + \nabla.\left(\rho \vec{g}\right)$

For constant μ and ρ: $\nabla.\nabla p = -\nabla.\left(\nabla.(\rho \vec{v}\ \vec{v})\right)$

Pressure-correction Methods

$$H_i = -\frac{\delta(\rho u_i u_j)}{\delta x_j} + \frac{\delta \tau_{ij}}{\delta x_j}$$

Forward-Euler Explicit in Time:
$$\begin{cases} (\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left(H_i^n - \dfrac{\delta p^n}{\delta x_i} \right) \\ \dfrac{\delta}{\delta x_i}\left(\dfrac{\delta p^n}{\delta x_i}\right) = \dfrac{\delta H_i^n}{\delta x_i} \end{cases}$$

Backward-Euler Implicit in Time:
$$\begin{cases} (\rho u_i)^{n+1} - (\rho u_i)^n = \Delta t \left(-\dfrac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \dfrac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \dfrac{\delta p}{\delta x_i}^{n+1} \right) \\ \dfrac{\delta}{\delta x_i}\left(\dfrac{\delta p}{\delta x_i}^{n+1}\right) = \dfrac{\delta}{\delta x_i}\left(-\dfrac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \dfrac{\delta \tau_{ij}}{\delta x_j}^{n+1} \right) \end{cases}$$

Backward-Euler Implicit in Time, linearized momentum update:

$$(\rho u_i)^{n+1} - (\rho u_i)^n = \rho\,\Delta u_i = \Delta t \left(-\frac{\delta(\rho u_i u_j)}{\delta x_j}^n - \frac{\delta(\rho u_i^n \Delta u_j)}{\delta x_j} - \frac{\delta(\rho \Delta u_i u_j^n)}{\delta x_j} + \frac{\delta \tau_{ij}}{\delta x_j}^n + \frac{\delta \Delta\tau_{ij}}{\delta x_j} - \frac{\delta p}{\delta x_i}^n - \frac{\delta \Delta p}{\delta x_i} \right)$$

Steady state solver, matrix notation:

Outer iteration, nonlinear solve: $\mathbf{A}^{\mathbf{u}_i^{m*}} \mathbf{u}_i^{m*} = \mathbf{b}_{\mathbf{u}_i^{m*}}^{m-1} - \dfrac{\boldsymbol{\delta} p}{\boldsymbol{\delta} x_i}^{m-1}$

Outer iteration, pressure update: $\dfrac{\boldsymbol{\delta}\, \tilde{\mathbf{u}}_i^{m*}}{\boldsymbol{\delta} x_i} = \dfrac{\boldsymbol{\delta}}{\boldsymbol{\delta} x_i}\left(\left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \dfrac{\boldsymbol{\delta} p}{\boldsymbol{\delta} x_i}^{m} \right)$, $\qquad \tilde{\mathbf{u}}_i^{m*} = \left(\mathbf{A}^{\mathbf{u}_i^{m*}}\right)^{-1} \mathbf{b}_{\mathbf{u}_i^{m*}}^{m-1}$

Inner iteration, linear solve: $\mathbf{A}^{\mathbf{u}_i^{m*}} \mathbf{u}_i^{m} = \mathbf{b}_{\mathbf{u}_i^{m*}}^{m} - \dfrac{\boldsymbol{\delta} p}{\boldsymbol{\delta} x_i}^{m}$

Steady state solver, matrix notation, pressure-correction schemes:

Based on the above, but introduce $\mathbf{u}_i^m = \mathbf{u}_i^{m*} + \mathbf{u}'$ $p^m = p^{m-1} + p'$ and further simplify to get varied schemes (SIMPLE, SIMPLER, SIMPLEC, PISO, etc.)

Projection Methods, Pressure-Correction Form

Non-Incremental:

$$\left(\rho u_i^*\right)^{n+1} = \left(\rho u_i\right)^n + \Delta t\left(-\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1}\right); \quad \left(\rho u_i^*\right)^{n+1}\Big|_{\partial D} = (\mathrm{bc})$$

$$\frac{\delta}{\delta x_i}\left(\frac{\delta p}{\delta x_i}^{n+1}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^*\right)^{n+1}\right) \quad ; \quad \frac{\delta p}{\delta n}^{n+1}\Bigg|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t\,\frac{\delta p}{\delta x_i}^{n+1}$$

Incremental:

$$\left(\rho u_i^*\right)^{n+1} = \left(\rho u_i\right)^n + \Delta t\left(-\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p^n}{\delta x_i}\right); \quad \left(\rho u_i^*\right)^{n+1}\Big|_{\partial D} = (\mathrm{bc})$$

$$\frac{\delta}{\delta x_i}\left(\frac{\delta\left(p^{n+1}-p^n\right)}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^*\right)^{n+1}\right); \quad \frac{\delta\left(p^{n+1}-p^n\right)}{\delta n}\Bigg|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t\,\frac{\delta\left(p^{n+1}-p^n\right)}{\delta x_i}$$

Rotational Incremental:

$$\left(\rho u_i^*\right)^{n+1} = \left(\rho u_i\right)^n + \Delta t\left(-\frac{\delta(\rho u_i u_j)}{\delta x_j}^{n+1} + \frac{\delta \tau_{ij}}{\delta x_j}^{n+1} - \frac{\delta p^n}{\delta x_i}\right); \quad \left(\rho u_i^*\right)^{n+1}\Big|_{\partial D} = (\mathrm{bc})$$

$$\frac{\delta}{\delta x_i}\left(\frac{\delta\left(\delta p^{n+1}\right)}{\delta x_i}\right) = \frac{1}{\Delta t}\frac{\delta}{\delta x_i}\left(\left(\rho u_i^*\right)^{n+1}\right); \quad \frac{\delta\left(\delta p^{n+1}\right)}{\delta n}\Bigg|_{\partial D} = 0$$

$$\left(\rho u_i\right)^{n+1} = \left(\rho u_i^*\right)^{n+1} - \Delta t\,\frac{\delta\left(\delta p^{n+1}\right)}{\delta x_i}$$

$$p^{n+1} = p^n + \delta p^{n+1} - \mu\frac{\delta}{\delta x_i}\left(\left(u_i^*\right)^{n+1}\right)$$