# FINITE DIFFERENCE METHODS


Dr. Sreenivas Jayanti

Department of Chemical Engineering

IIT-Madras

# THE CFD APPROACH

- Assembling the governing equations
- Identifying flow domain and boundary conditions
- Geometrical discretization of flow domain
- *Discretization of the governing equations*
- *Incorporation of boundary conditions*
- Solution of resulting algebraic equations
- Post-solution analysis and reformulation, if needed

# OUTLINE

- Basics of finite difference (FD) methods
- FD approximation of arbitrary accuracy
- FD formulas for higher derivatives
- Application to an elliptic problem
- FD for time-dependent problems
- FD on non-uniform meshes
- Closure

# Basics of Finite Difference Methods

- FD methods are quite old and somewhat dated for CFD problems but serve as a point of departure for CFD studies
- The principal idea of CFD methods is to replace a governing partial differential equation by an *equivalent and approximate* set of algebraic equations
- Finite difference techniques are one of several options for this *discretization* of the governing equations
- In finite difference methods, each derivative of the pde is replaced by an equivalent finite difference approximation

# Basics of Finite Difference Methods

- The basis of a finite difference method is the Taylor series expansion of a function.

- Consider a continuous function f(x). Its value at neighbouring points can be expressed in terms of a Taylor series as

**(1)** $f(x+\Delta x) = f(x) + df/dx\,(\Delta x) + d^2f/dx^2\,(\Delta x^2)/2! + .. + d^nf/dx^n\,(\Delta x^n)/n! + ..$

- The above series converges if $\Delta x$ is small and f(x) is differentiable

- For a converging series, successive terms are progressively smaller

# FD Approximation for a First Derivative

- The terms in the Taylor series expansion can be rearranged to give

$$df/dx = [f(x+ \Delta x) - f(x)] / \Delta x - d^2f/dx^2 (\Delta x)/2! - \ldots - d^nf/dx^n (\Delta x^{n-1})/n! - \ldots$$

Or

**(2)** $$df/dx \approx [f(x+ \Delta x) - f(x)] / \Delta x + O(\Delta x)$$

- Here $O(\Delta x)$ implies that the leading term in the neglected terms of the order of $\Delta x$, i.e., the error in the approximation reduces by a factor of 2 if $\Delta x$ is halved.

- Equation (2) is therefore a first order-accurate approximation for the first derivative.

# Other Approximations for a First Derivative

- Other approximations are also possible. Writing the Taylor series expansion for $f(x- \Delta x)$, we have

**(3)** $f(x-\Delta x) = f(x) - df/dx\ (\Delta x) + d^2f/dx^2\ (\Delta x^2)/\ 2! - .. + d^nf/dx^n\ (\Delta x^n)/n! +$

- Equation (3) can be rearranged to give another first order approximation :

**(4)** $df/dx \approx [f(x) - f(x-\Delta x)] / \Delta x + O(\Delta x)$

- Subtracting (3) from (1) gives a second order approximation for df/dx :

**(5)** $df/dx \approx [f(x+ \Delta x) - f(x-\Delta x)] / (2\Delta x) + O(\Delta x^2)$

# FD Approximations on a Uniform Mesh

- Consider a uniform mesh with a spacing of $\Delta x$ over an interval [0, L]
- Denoting the mesh index by i, we can write

$$f_i = f(x_i) = f(i\ \Delta x) \quad \text{and} \quad fi+1 = f\,[(i+1)\ \Delta x] \quad \text{and so on}$$

- Then

**(2)** $\Rightarrow$ $df/dx \approx [f(x+ \Delta x) - f(x)] / \Delta x \quad = (fi+1 - fi)/\Delta x + O(\Delta x)$

**(3)** $\Rightarrow$ $df/dx \approx [f(x) - f(x- \Delta x)] / \Delta x \quad = (fi- fi-1)/\Delta x + O(\Delta x)$

**(5)** $\Rightarrow$ $df/dx \approx [f(x+ \Delta x) - f(x-\Delta x)] / (2\Delta x) \quad = (fi+1- fi-1)/(2\Delta x) + O(\Delta x^2)$

are the

| | |
|---|---|
| forward | “one-sided” |
| backward | “one-sided” |
| central | “symmetric” |

differencing formulas, respectively, for df/dx at x or node i

- One-sided formulas are necessary at ends of domains

# Higher Order Accuracy

- Higher order of accuracy of approximation can be obtained by including more number of adjacent points

- Let us seek a third-order, one-sided approximation for u(x). This requires four points and will be of the form

**(6)** $du/dx)i = [aui + bui+1 + cui+2 + dui+3]/ \Delta x + O(\Delta x^3)$

- This is equivalent to writing (6) as

**(7)** $du/dx)i = [aui + bui+1 + cui+2 + dui+3]/ \Delta x + (0)\, d^2u/dx^2(\Delta x)$
$+ (0)d^3u/dx^3(\Delta x^2) + (e)\, d^4u/dx^4(\Delta x^3)$

or

**(8)** $aui + bui+1 + cui+2 + dui+3 = + du/dx\, (\Delta x) + (0)\, d^2u/dx^2(\Delta x^2)$
$+ (0)d^3u/dx^3(\Delta x^3) - (e)\, d^4u/dx^4(\Delta x^4)$

- How to find a, b, c and d?

# Third-Order, One-sided Formula

- Expand ui+1, ui+2 and ui+3 in Taylor series about ui:

**(9a)** $u_{i+1} = u(x+ 1\Delta x) = u(x) + du/dx\,(\Delta x) + d^2u/dx^2\,(\Delta x)^2/ 2! + \ldots$

**(9b)** $u_{i+2} = u(x+ 2\Delta x) = u(x) + du/dx\,(2\Delta x) + d^2u/dx^2\,(2\Delta x)^2/ 2! + \ldots$

**(9c)** $u_{i+3} = u(x+ 3\Delta x) = u(x) + du/dx\,(3\Delta x) + d^2u/dx^2\,(3\Delta x)^2/ 2! + \ldots$

- Find { a ui + b (9a) + c(9b) + d (9c) } and rearrange to get

**(10)** $a u_i + b u_{i+1} + c u_{i+2} + d u_{i+3} = pu + q\, du/dx\,(\Delta x) + r\, d^2u/dx^2(\Delta x^2)$
$+ s\, d^3u/dx^3(\Delta x^3) + t\, d^4u/dx^4(\Delta x^4)$

- Compare the coefficients of (8) and (10) to get

  a = -11/6 b = 3 c = -3/2 d = 1/3

or

**(11)** $du/dx)_i = [-11 u_i + 18 u_{i+1} - 9 u_{i+2} + 2 u_{i+3}]/ (6\Delta x) + O(\Delta x^3)$

# Higher Derivatives

- Finite difference approximation for second derivative:

$$d^2u/dx^2)_i = [d/dx( du/dx)]_i$$
$$\approx [ (du/dx)_{i+1/2} - (du/dx)_{i-1/2} ] / \Delta x$$
$$\approx [ (ui+1 - ui)/ \Delta x - (ui - ui-1)/ \Delta x ] / \Delta x$$

or

**(12)** $$d^2u/dx^2)_i \approx [ (ui+1 -2 ui + ui-1) ] / \Delta x^2$$

- Taylor series evaluation of equation (11) shows that the approximation is second order accurate; thus,

**(12a)** $$d^2u/dx^2)_i = [ (ui+1 -2 ui + ui-1) ] / \Delta x^2 + O(\Delta x^2)$$

- Note that use of central differences for the second derivative requires *three* points, viz., (i-1), i, (i+1), for a second order accurate formula

# Other Formulas for Higher Derivatives

- Using forward differencing throughout, one can get the following first order accurate formula involving three points for the second derivative:

$$d^2u/dx^2)_i = [d/dx( du/dx)]_i \approx [ (du/dx)_{i+1} - (du/dx)_i] / \Delta x$$
$$\approx [ (ui+2 - ui+1)/ \Delta x - (ui+1 - ui)/ \Delta x ] / \Delta x$$

or

**(13)** $$d^2u/dx^2)_i \approx [ (ui+2 -2 ui+1 + ui) ] / \Delta x^2 + O(\Delta x)$$

- A central, second order scheme for the third derivative needs four points:

**(14)** $$d^3u/dx^3)_i = [ (ui+2 -2 ui+1 + 2 ui-1 - ui-2) ] / (2\Delta x^3) + O(\Delta x^2)$$

- If p = order of derivative, q = order of accuracy and n = no of points, then

  n = p + q -1 for central schemes

  n = p + q for one-sided schemes

# Mixed Derivatives

- Mixed derivatives can occur as a result of coordinate transformation to a non-orthogonal system (for example, to take account of non-regular shape of the flow domain).

- Straightforward application of the method for higher derivatives:

$$(\partial^2 u/\partial x \partial y)_{i,j} = [\partial/\partial x\ (\partial u/ \partial y)]_{i,j}$$

$$\approx [(\partial u/\partial y)_{i+1,j} - (\partial u/\partial y)_{i-1,j}] / (2\Delta x)$$

$$\approx [ (u_{i+1,j+1} - u_{i+1,j-1})/ 2\Delta y - (u_{i-1,j+1} - u_{i-1,j-1})/ 2\Delta y ] / (2\Delta x)$$

or

**(15)** $$(\partial^2 u/\partial x \partial y)_{i,j} \approx [(u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1})] / (4 \Delta x \Delta y) + O(\Delta x^2, \Delta y^2 )$$

- A large variety of schemes possible

# Example: 2-D Poisson Equation

**(16)** $\partial^2u/\partial x^2 + \partial^2u/\partial y^2 = f$ $\quad 0 \le x \le L$ and $0 \le y \le W$

with Dirichlet boundary condtions: $u(x,y) = g(x,y)$ on boundary

• Write $\partial^2u/\partial x^2)_{i,j} \approx [(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] / (\Delta x^2) + O(\Delta x^2)$

and $\partial^2u/\partial y^2)_{i,j} \approx [(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})] / (\Delta y^2) + O(\Delta y^2)$

and substitute in (16) to get

**(17)**
$$[(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] / (\Delta x^2) + [(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})] / (\Delta y^2) = f_{ij} + O(\Delta x^2, \Delta y^2)$$

- With Dirichlet boundary conditions, equation (17) would be valid for
  $2 \le i \le N_i \qquad 2 \le j \le N_j$
- Results in $(N_i-1) \times (N_j-1)$ algebraic equations to be solved for $u(i,j)$

# Poisson Equation: Other Boundary Conditions

- Normal gradient specified, e.g, $du/dy = c1$ for all i at $j = Nj$
- Values of $u(I, Nj)$ not known and have to be determined
- For these boundary points,

$$du/dy \approx (u_{i,Nj-1} - u_{i,Nj})/ \Delta y = c1 \quad \text{"first order accurate"}$$

or

$$u_{i,Nj} - u_{i,Nj-1} = c1 * \Delta y$$

- Equations for the interior points remain the same
- For second order accuracy, one can write

$$du/dy \approx (a u_{i,Nj-2} + b u_{i,Nj-1} + c u_{i,Nj})/ \Delta y = c1$$

- Convective boundary condition: $q''_w = h*(u_{inf} - u_w)$; h and uinf given
- This can be implemented by noting that $q''_w = -k\, du/dy$
- Thus, $h*(u_{inf} - u_{i,Nj}) = -k*(a u_{i,Nj-2} + b u_{i,Nj-1} + c u_{i,Nj})/ \Delta y$
  which gives the necessary algebraic equation for the boundary point.
- $(Ni-1) \times (Nj)$ algebraic equations to be solved for $u(i,j)$

# Discretization of Time Domain

- Consider the unsteady heat conduction problem: $\partial T/\partial t = \partial^2 T/\partial x^2$ **(18)**

- Denote $T(x,t) = T(i\,\Delta x, n\,\Delta t) = T_i^n = T_{i,n}$

- We seek discretization of eqn. (18) of the form

**(19)** $\partial T/\partial t\,)_{i,n} = \partial^2 T/\partial x^2)_{i,n}$

- Evaluate LHS of (19) using forward differencing as

**(20)** $\partial T/\partial t\,)_{i,n} = (T_{i,n+1} - T_{i,n}) / \Delta t + O(\Delta t)$

- But several options for RHS even if we choose, say, central differencing for $\partial^2 T/\partial x^2$

# Explicit and Implicit Schemes

- Put $\partial^2 T/\partial x^2)_{i,n} = (T_{i+1,n} - 2\, T_{i,n} + T_{i-1,n})/\ \Delta x^2 + O(\Delta x^2)$ **(21)**

  and substitute (20) and (21) in (19) to get

- Explicit equation for $T_{i,n+1}$:

$$(T_{i,n+1} - T_{i,n})/\ \Delta t = (T_{i+1,n} - 2\, T_{i,n} + T_{i-1,n})/\ \Delta x^2$$

or

$$T_{i,n+1} = T_{i,n} + \Delta t/\Delta x^2\ (T_{i+1,n} - 2\, T_{i,n} + T_{i-1,n}) + \mathbf{O(\Delta t, \Delta x^2)} \quad \mathbf{(22)}$$

- Put $\partial^2 T/\partial x^2)_{i,n} = (T_{i+1,n+1} - 2\, T_{i,n+1} + T_{i-1,n+1})/\ \Delta x^2 + O(\Delta x^2)$ **(23)**

  and substitute (20) and (21) in (19) to get

- Implicit equation for $T_{i,n+1}$:

$$(T_{i,n+1} - T_{i,n})/\ \Delta t = (T_{i+1,n+1} - 2\, T_{i,n+1} + T_{i-1,n+1})/\ \Delta x^2 \quad \text{or}$$

$$(1+ 2\, \Delta t/\Delta x^2)\ T_{i,n+1} = T_{i,n} + \Delta t/\Delta x^2\ (T_{i+1,n+1}+ T_{i-1,n+1}) + \mathbf{O(\Delta t, \Delta x^2)} \quad \mathbf{(24)}$$

# Other Schemes

- Put $T_{i,n} = (T_{i,n-1} + T_{i,n+1})/2$ to get

- the DuFort-Frankel scheme:

**(25)** $$(T_{i,n+1} - T_{i,n-1})/(2\Delta t) = (T_{i+1,n} - T_{i,n+1} + T_{i,n-1} + T_{i-1,n})/\Delta x^2 + O(\Delta t^2, \Delta x^2)$$

  – Explicit, second order accurate and unconditionally stable

- Evaluate RHS at (n+1/2) as (RHS,n + RHS,n+1)/2 and put in (19) to get

- the Crank-Nicolson scheme:

**(26)** $$-r T_{i-1,n+1} + (2+2r) T_{i,n+1} - r T_{i+1,n+1} = r T_{i-1,n} + (2-2r) T_{i,n} + r T_{i+1,n}$$

where $r = \Delta t/\Delta x^2$

  – Implicit, second order accurate and unconditionally stable

# Explicit vs Implicit Methods

- Explicit methods are simple to program and allow marching forward in time point by point from given initial condition

- Implicit methods are more difficult to program and require simultaneous solution of algebraic equations at each time step to get the solution

- Explicit methods are generally less stable than implicit methods and may give unphysical solutions if the marching time step is too large

- Too-large a time step, which is possible with implicit methods, may result in less accuracy and instability in non-linear problems

# Non-uniform Meshes

- The methods discussed above can be extended to non-uniform meshes but may have to be done with care so as not to lose an order of accuracy :

- $du/dx)_I = \{u_{i+1} (\Delta x_{i-1})^2 - u_{i-1}(\Delta x_i)^2 + u_i[(\Delta x_{i-1})^2 - (\Delta x_i)^2]\}$

  $/ \{(\Delta x_{i-1}) (\Delta x_i) (\Delta x_{i-1} + \Delta x_i)\} + O\{(\Delta x_i)^2\}$

  where $\Delta x_i = x_{i+1} - x_i$ etc

- Highly non-uniform and distorted meshes should be avoided where possible

# Closure

- Finite difference techniques can be used to systematically develop approximate formula for derivatives of a function on a structured mesh
- Formulas of desired accuracy level can be obtained for a derivative provided sufficient number of points are included in the formula
- Discretized equations of  partial differential equations can be obtained by replacing the derivatives with finite difference formulas
- The resulting equations are coupled of algebraic equations which may be non-linear for non-linear equations and require simultaneous solution
- The issue of stability is brought into play for time-dependent problems

Thank you for your attention.

E-mail me at

sjayanti@iitm.ac.in

in case of queries