OUTLINE

6.1 Components of Numerical Methods

6.2 Introduction to Finite Difference

6-3 Errors Involved in Numerical Solutions

6-4 Example

6.1 Components of numerical methods (1)
Properties-1

- Consistence
  1. The discretization should become exact as the grid spacing tends to zero
  2. Truncation error: Difference between the discretized equation and the exact one
- Stability: does not magnify the errors that appear in the course off numerical solution process.
  1. Iterative methods: not diverge
  2. Temporal problems: bounded solutions
  3. Von Neumann's method
  4. Difficulty due to boundary conditions and non-linearities present.
- Convergence: solution of the discretized equations tends to the exact solution of the differential equation as the grid spacing tends to zero.

6.1 Components of numerical methods (2)

Properties-2

- Conservation

1. The numerical scheme should on both local and global basis respect the conservation laws.
2. Automatically satisfied for control volume method, either individual control volume or the whole domain.
3. Errors due to non-conservation are in most cases appreciable only on relatively coarse grids, but hard to estimate quantitatively

- Boundedness:

1. Numerical solutions should lie within proper bounds.
2. Physically, non-negative quantities (like density, kinetic energy of turbulence) must be always be positive; other quantities, such as concentration, must lie between 0% and 100%.
3. In the absence of sources, some equations (e.g. the heat equation for the temperature when no heat sources are present) require that the minimum and maximum values of the variable be found on the boundaries of the domain.

- Realizability: models of phenomena which are too complex to treat directly (turbulence, combustion, or multiphase flow) should be designed to guarantee physically realistic solutions.
- Accuracy: 1. Modeling error 2. Discretization errors 3. Iterative errors

6.1 Components of numerical methods (3)

Discretization methods (Finite Difference)-1

- First step in obtaining a numerical solution is to discretize the geometric domain→ to define a numerical grid
- Each node has one unknown and need one algebraic equation, which is a relation between the variable value at that node and those at some of the neighboring nodes.
- The approach is to replace each term of the PDE at the particular node by a finite-difference approximation.
- Numbers of equations and unknowns must be equal

6.1 Components of numerical methods (4)
Discretization methods (Finite Difference)-2

- Taylor Series Expansion: Any continuous differentiable function, in the vicinity of x_i , can be expressed as a Taylor series:

$$\Phi(x)=\Phi(x_i)+(x-x_i)\left(\frac{\partial\Phi}{\partial x}\right)_i+\frac{(x-x_i)^2}{2!}\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i+\frac{(x-x_i)^3}{3!}\left(\frac{\partial^3\Phi}{\partial x^3}\right)_i+\ldots+\frac{(x-x_i)^n}{n!}\left(\frac{\partial^n\Phi}{\partial x^n}\right)_i+H$$

$$\left(\frac{\partial\Phi}{\partial x}\right)_i=\frac{\Phi_{i+1}-\Phi_i}{x_{i+1}-x_i}-\frac{x_{i+1}-x_i}{2}\left(\frac{\partial^2\Phi}{\partial x^2}\right)_i-\frac{(x_{i+1}-x_i)^2}{6}\left(\frac{\partial^3\Phi}{\partial x^3}\right)_i+H$$

- Higher order derivatives are unknown and can be dropped when the distance between grid points is small.
- By writing Taylor series at different nodes, x_{i-1}, x_{i+1}, or both x_{i-1} and x_{i+1}, we can have:

$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_{i+1}-\Phi_i}{x_{i+1}-x_i}$$ Forward-FDS

$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_i-\Phi_{i-1}}{x_i-x_{i-1}}$$ Backward-FDS

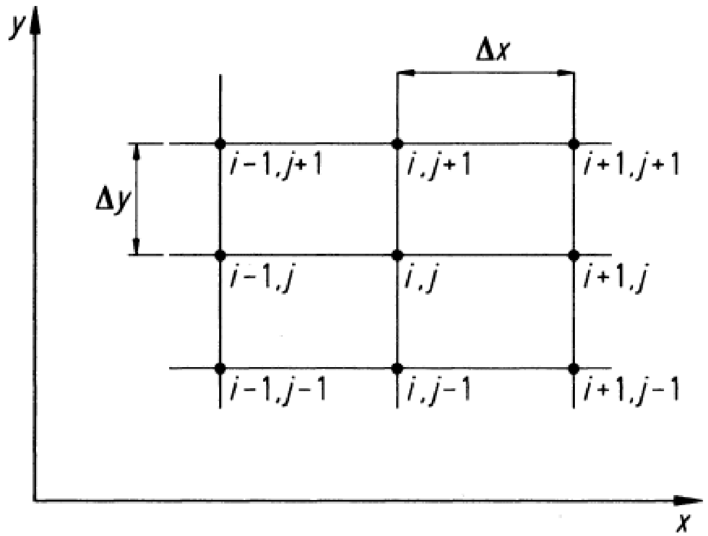
$$\left(\frac{\partial\Phi}{\partial x}\right)_i\approx\frac{\Phi_{i+1}-\Phi_{i-1}}{x_{i+1}-x_{i-1}}$$ Central-FDS

1st order, order of accuracy $P_{kest}=1$

2nd order, order of accuracy $P_{kest}=1$

6-2 Introduction to Finite Difference (1)

- Numerical solutions can give answers at only discrete points in the domain, called grid points.



- If the PDEs are totally replaced by a system of algebraic equations which can be solved for the values of the flow-field variables at the discrete points only, in this sense, the original PDEs have been discretized. Moreover, this method of discretization is called the method of finite differences.

6-2 Introduction to Finite Difference (2)

- A partial derivative replaced with a suitable algebraic difference quotient is called finite difference. Most finite-difference representations of derivatives are based on Taylor's series expansion.
- Taylor's series expansion:

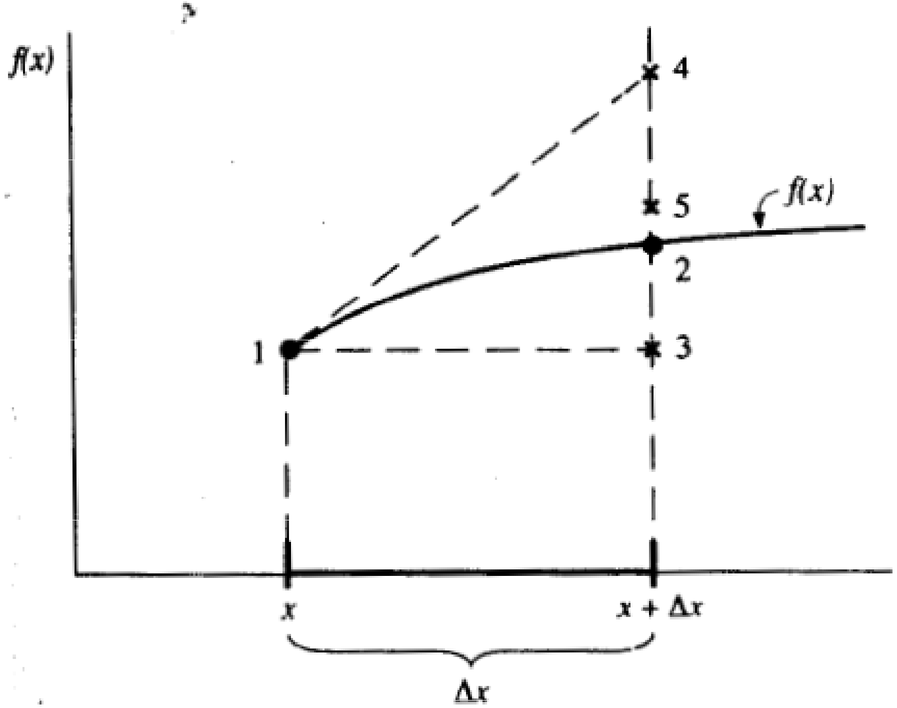
Consider a continuous function of x, namely, f(x), with all derivatives defined at *x*. Then, the value of f at a location can be estimated from a Taylor series expanded about point x, that is, $x+\Delta x$

$$f(x+\Delta x)=f(x)+\frac{\partial f}{\partial x}\Delta x+\frac{1}{2!}\frac{\partial^2 f}{\partial x^2}(\Delta x)^2+\frac{1}{3!}\frac{\partial^3 f}{\partial x^3}(\Delta x)^3+\ldots+\frac{1}{n!}\frac{\partial^n f}{\partial x^n}(\Delta x)^n+\ldots$$

- In general, to obtain more accuracy, additional higher-order terms must be included.

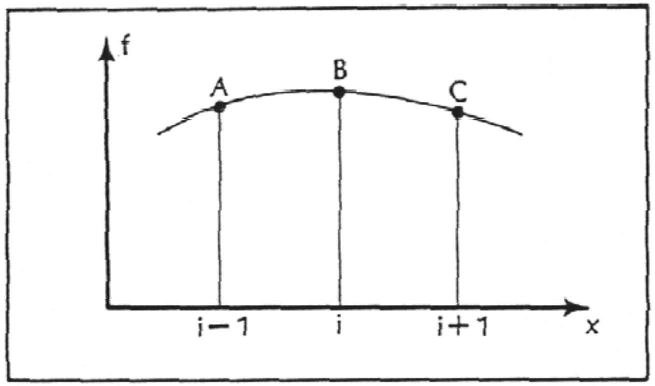
6-2 Introduction to Finite Difference (3)

$$f(x+\Delta x) = \underbrace{f(x)}_{\substack{\text{First guess}\\\text{(not very good)}}} + \underbrace{\frac{\partial f}{\partial x}\Delta x}_{\substack{\text{Add to capture}\\\text{slope}}} + \underbrace{\frac{\partial^2 f}{\partial x^2}\frac{(\Delta x)^2}{2}}_{\substack{\text{Add to account}\\\text{for curvature}}} + \cdots$$



6-2 Introduction to Finite Difference (4)

- Forward, Backward and Central Differences:



(1) Forward difference:

Neglecting higher-order terms, we can get

$$f(x_{i+1}) = f(x_i) + (\frac{\partial f}{\partial x})_i (x_{i+1} - x_i) + \frac{(x_{i+1} - x_i)^2}{2!}(\frac{\partial^2 f}{\partial x^2})_i + \frac{(x_{i+1} - x_i)^3}{3!}(\frac{\partial^3 f}{\partial x^3})_i + \ldots + \frac{(x_{i+1} - x_i)^n}{n!}(\frac{\partial^n f}{\partial x^n})_i + \ldots \quad \text{(a)}$$

$$(\frac{\partial f}{\partial x})_i = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = \frac{f(x_{i+1}) - f(x_i)}{\Delta x_{i+1}}; \quad \Delta x_{i+1} = x_{i+1} - x_i$$

6-2 Introduction to Finite Difference (5)

(2) Backward difference:

Neglecting higher-order terms, we can get

$$f(x_{i-1}) = f(x_i) - (\frac{\partial f}{\partial x})_i (x_i - x_{i-1}) + \frac{(x_i - x_{i-1})^2}{2!}(\frac{\partial^2 f}{\partial x^2})_i - \frac{(x_i - x_{i-1})^3}{3!}(\frac{\partial^3 f}{\partial x^3})_i + \ldots + (-1)^n \frac{(x_i - x_{i-1})^n}{n!}(\frac{\partial^n f}{\partial x^n})_i + \ldots \quad \text{(b)}$$

$$(\frac{\partial f}{\partial x})_i = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} = \frac{f(x_i) - f(x_{i-1})}{\Delta x_i}; \quad \Delta x_i = x_i - x_{i-1}$$

(3) Central difference:

(a)-(b) and neglecting higher-order terms, we can get

$$(\frac{\partial f}{\partial x})_i = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}} \quad \ldots\text{(c)}$$

6-2 Introduction to Finite Difference (6)

(4) If $\Delta x_i = \Delta x_{i+1} = \Delta x$, then (a), (b), (c) can be expressed as

Forward: $$(\frac{\partial f}{\partial x})_i = \frac{f_{i+1} - f_i}{\Delta x} \quad \ldots(d)$$

Backward: $$(\frac{\partial f}{\partial x})_i = \frac{f_i - f_{i-1}}{\Delta x} \quad \ldots(e)$$

Central: $$(\frac{\partial f}{\partial x})_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} \quad \ldots(f)$$

Note:
$$f_{i+1} = f(x_{i+1})$$
$$f_i = f(x_i)$$
$$f_{i-1} = f(x_{i-1})$$

6-2 Introduction to Finite Difference (7)

- Truncation error:

The higher-order term neglecting in Eqs. (a), (b), (c) constitute the truncation error. The general form of Eqs. (d), (e), (f) plus truncated terms can be written as

Forward: $$\left(\frac{\partial f}{\partial x}\right)_i = \frac{f_{i+1} - f_i}{\Delta x} + o(\Delta x)$$

Backward: $$\left(\frac{\partial f}{\partial x}\right)_i = \frac{f_i - f_{i-1}}{\Delta x} + o(\Delta x)$$

Central: $$\left(\frac{\partial f}{\partial x}\right)_i = \frac{f_{i+1} - f_{i-1}}{2\Delta x} + o(\Delta x)^2$$

6-2 Introduction to Finite Difference (8)

- Second derivatives:

  * Central difference:

  If $\Delta x_i = \Delta x_{i+1} = \Delta x$, then (a)+(b) becomes

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_i = \frac{f_{i+1} - 2f_i + f_{i-1}}{(\Delta x)^2} + o(\Delta x)^2$$

  * Forward difference:

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_i = \frac{f_{i+2} - 2f_{i+1} + f_i}{(\Delta x)^2} + o(\Delta x)$$

  * Backward difference:

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_i = \frac{f_i - 2f_{i-1} + f_{i-2}}{(\Delta x)^2} + o(\Delta x)$$

6-2 Introduction to Finite Difference (9)

- Mixed derivatives:

  * Taylor series expansion:

$$f(x+\Delta x, y+\Delta y) = f(x,y) + \Delta x\frac{\partial f}{\partial x} + \Delta y\frac{\partial f}{\Delta y} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f}{\partial x^2} + \frac{(\Delta y)^2}{2!}\frac{\partial^2 f}{\partial y^2} + 2\frac{(\Delta x)(\Delta y)}{2!}\frac{\partial^2 f}{\partial x \partial y} + o[(\Delta x)^3, (\Delta y)^3]$$

  * Central difference:

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} = \frac{f_{i+1,j+1} - f_{i-1,j+1} - f_{i+1,j-1} + f_{i-1,j-1}}{4(\Delta x)(\Delta y)} + o[(\Delta x)^2, (\Delta y)^2]$$

  * Forward difference:

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} = \frac{f_{i+1,j+1} - f_{i,j+1} - f_{i+1,j} + f_{i,j}}{(\Delta x)(\Delta y)} + o[(\Delta x)\ ,(\Delta y)\ ]$$

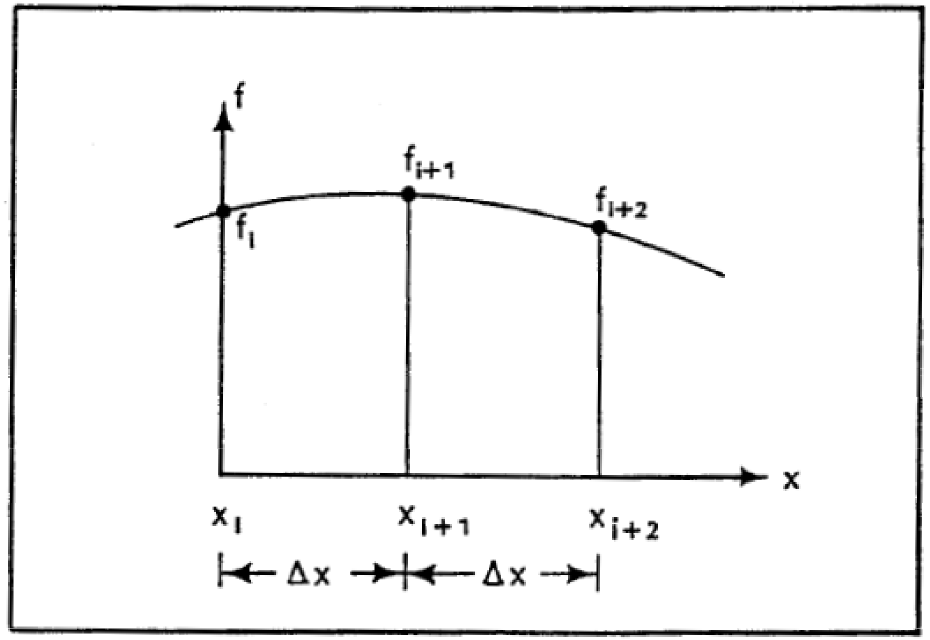
  * Backward difference:

$$\left(\frac{\partial^2 f}{\partial x \partial y}\right)_{i,j} = \frac{f_{i,j} - f_{i-1,j} - f_{i,j-1} + f_{i-1,j-1}}{(\Delta x)(\Delta y)} + o[(\Delta x)\ ,(\Delta y)\ ]$$

6-2 Introduction to Finite Difference (10)

- Polynomial fitting:

Many applications of polynomial fitting are observed in CFD and heat transfer. The technique can be used to develop the entire finite-difference representation for a PDE. However, the technique is perhaps most commonly employed in the treatment of boundary conditions.



6-2 Introduction to Finite Difference (11)

(1) Consider a second-order polynomial, f(x)= Ax^2+ Bx +C

(a) Select the origin at x_i, thus,

$$x_i = 0 \qquad x_{i+1} = \Delta x \qquad x_{i+2} = 2\Delta x$$

(b)

$$\left.\begin{aligned} f_i &= Ax_i^2 + Bx_i + C = C \qquad \because x_i = 0 \\ f_{i+1} &= Ax_{i+1}^2 + Bx_{i+1} + C = A(\Delta x)^2 + B(\Delta x) + C \\ f_{i+2} &= Ax_{i+2}^2 + Bx_{i+2} + C = A(2\Delta x)^2 + B(2\Delta x) + C \end{aligned}\right\} \qquad \begin{aligned} C &= f_i \\ B &= \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2(\Delta x)} \\ A &= \frac{f_{i+2} - 2f_{i+1} + f_i}{2(\Delta x)^2} \end{aligned}$$

(2) For a first-order polynomial f(x)=ax+b

(a)

$$\left.\begin{aligned} f_i &= ax_i + b \\ f_{i+1} &= ax_{i+1} + b \end{aligned}\right\} \Rightarrow \begin{aligned} f_i &= b \\ f_{i+1} &= a(\Delta x) + b \end{aligned} \qquad \Rightarrow \begin{aligned} b &= f_i \\ a &= \frac{f_{i+1} - f_i}{\Delta x} \end{aligned}$$

(b)

$$\left.\frac{\partial f}{\partial x}\right|_i = a = \frac{f_{i+1} - f_i}{\Delta x}$$

6-2 Introduction to Finite Difference (12)

(3) Finite-difference grid near wall

If we assume that the temperature distribution near the boundary is a second-degree polynomial of the form

$$T = a + by + c\,y^2 \quad \& \quad \left.\frac{\partial T}{\partial y}\right|_{y=0} = b$$

$$\Rightarrow a = T_1, \quad b = \frac{-3T_1 + 4T_2 - T_3}{2(\Delta y)}, \quad c = \frac{T_1 - 2T_2 + T_3}{2(\Delta y)^2}$$

Thus, we can evaluate the wall heat flux by the approximation

$$q''_w = -k\left.\frac{\partial T}{\partial y}\right)_{y=0} = -kb = \frac{k}{2(\Delta y)}(3T_1 - 4T_2 + T_3$$

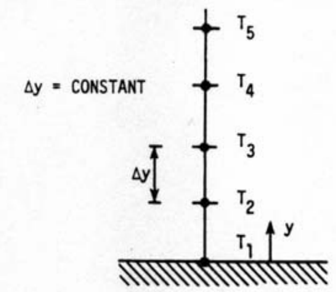


Figure 3.6 Finite-difference grid near wall.

6-2 Introduction to Finite Difference (13)

- Alternatively, we can identify the second-degree polynomial as a truncated Taylor's series expansion about y=0.

  (1) Second-degree polynomial: $T=a+by+cy^2$

  (2) Taylor series: $$T = T(0) + \left.\frac{\partial T}{\partial y}\right|_0 y + \left.\frac{\partial^2 T}{\partial y^2}\right|_0 \frac{y^2}{2!} + \underbrace{\left.\frac{\partial^3 T}{\partial y^3}\right|_0 \frac{y^3}{3!}}_{T.E.} + \ldots$$

  (3) Thus, the approximation $T \approx a+by+cy^2$ is equivalent to utilizing the first three terms of a Taylor series expansion with the resulting T.E. in the expansion for T being $o\left[(\Delta y)^3\right]$

  Solving the Taylor series for an expression for $\left.\frac{\partial T}{\partial y}\right|_0$ involves $\left.\frac{\partial T}{\partial y}\right|_0$

  division by Δy , which reduces the T.E. in the expansion for

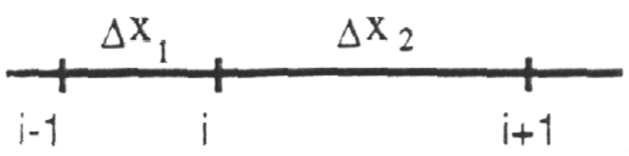
  to $o\left[(\Delta y)^2\right]$

6-2 Introduction to Finite Difference (14)

- The accuracy of changing the mesh size:

  (1) To obtain higher resolution in the region where the gradients are expected to vary rapidly, it is desirable to use a finer mesh over that particular region rather than refining the mesh over the entire domain.

  (2) Consider a change of the mesh size from Δx_2 to Δx_1 at some node *i*.



(3) Approximation of first derivative:

(i) The function *f(x)* is expanded about the node *i* in forward and backward Taylor series, respectively, as

$$f_{i+1} = f_i + (\Delta x_2)\left.\frac{df}{dx}\right|_i + \frac{(\Delta x_2)^2}{2!}\left.\frac{d^2 f}{dx^2}\right|_i + \frac{(\Delta x_2)^3}{3!}\left.\frac{d^3 f}{dx^3}\right|_i + o\left[(\Delta x_2)^4\right]\ldots \quad (a)$$

$$f_{i-1} = f_i - (\Delta x_1)\left.\frac{df}{dx}\right|_i + \frac{(\Delta x_1)^2}{2!}\left.\frac{d^2 f}{dx^2}\right|_i - \frac{(\Delta x_1)^3}{3!}\left.\frac{d^3 f}{dx^3}\right|_i + o\left[(\Delta x_1)^4\right]\ldots \quad (b)$$

(a)-(b)=>

$$\left.\frac{df}{dx}\right|_i = \frac{f_{i+1} - f_{i1}}{\Delta x_2 + \Delta x_1} - \frac{1}{2}\frac{(\Delta x_2)^2 - (\Delta x_1)^2}{\Delta x_2 + \Delta x_1}\left.\frac{d^2 f}{dx^2}\right|_i + o\left[(\Delta x)^2\right]\ldots \quad (c)$$

where $o\left[(\Delta x)^2\right]$ means the largest of $o\left[(\Delta x_1)^2\right]$ or $o\left[(\Delta x_2)^2\right]$

6-2 Introduction to Finite Difference (16)

(ii) Then the finite-difference approximation of the first derivative at the node I where the mesh size is changed from to Δx_1 Δx_2 becomes

$$\left.\frac{df}{dx}\right|_i = \frac{f_{i+1} - f_{i-1}}{\Delta x_2 + \Delta x_1} \cdots \qquad (d)$$

Eq. (c) implies eq. (d) is second-order accurate only if $\Delta x_2 \to \Delta x_1$

(iii) We note that, if the mesh size varies from Δx_1 to Δx_2

abruptly, say $\Delta x_2 = 2\Delta x_1$, then the accuracy of the differencing at *i* deteriorates to first-order.

(4) Approximation of second derivative:

(i) (a)+(b)x $\left(\Delta x_2 / \Delta x_1 \right)^2$

=> $$f_{i+1} + \varepsilon^2 f_{i-1} = \left(1+\varepsilon^2\right) f_i + (1-\varepsilon)(\Delta x_2) \left.\frac{df}{dx}\right|_i + (\Delta x_2)^2 \left.\frac{d^2 f}{dx^2}\right|_i +$$

$$\frac{1}{6}(\Delta x_2 - \Delta x_1)(\Delta x_2)^2 \left.\frac{d^3 f}{dx^3}\right|_i + o\left[(\Delta x)^4\right] \ldots \quad (e)$$

where $\varepsilon = \dfrac{\Delta x_2}{\Delta x_1}$

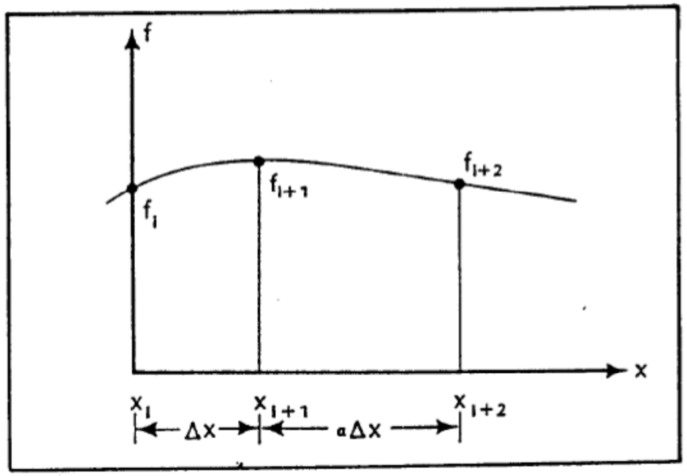
and $o\left[(\Delta x)^4\right]$ means the largest of $o\left[(\Delta x_1)^4\right]$ and $o\left[(\Delta x_2)^4\right]$

6-2 Introduction to Finite Difference (18)

(ii) From eq. (e), we can get

$$\left.\frac{d^2 f}{dx^2}\right|_i = \frac{f_{i+1} - \left(1+\varepsilon^2\right) f_i + \varepsilon^2 f_{i-1}}{\left(\Delta x_2\right)^2} - \frac{1-\varepsilon}{\Delta x_2} \left.\frac{df}{dx}\right|_i + o\left[\left(\Delta x_2 - \Delta x_1\right)\right]$$

The above expression becomes second-order accurate as $\Delta x_2 \rightarrow \Delta x_1$ and it implies that, unless the mesh spacing is changed slowly, the truncation error deteriorates.



6-3 Errors Involved in Numerical Solutions (1)

- In the solution of differential equations with finite differences, a variety of schemes are available for the discretization of derivatives and the solution of the resulting system of algebraic equations.
- In many situations, questions arise regarding the round-off and truncation errors involved in the numerical computations, as well as the consistency, stability and the convergence of the finite difference scheme.
- Round-off errors:computations are rarely made in exact arithmetic. This means that real numbers are represented in "floating point" form and as a result, errors are caused due to the rounding-off of the real numbers. In extreme cases such errors, called "round-off" errors, can accumulate and become a main source of error.

6-3 Errors Involved in Numerical Solutions (2)

- Truncation error: In finite difference representation of derivative with Taylor's series expansion, the higher order terms are neglected by truncating the series and the error caused as a result of such truncation is called the "truncation error".
- The truncation error identifies the difference between the exact solution of a differential equation and its finite difference solution without round-off error.

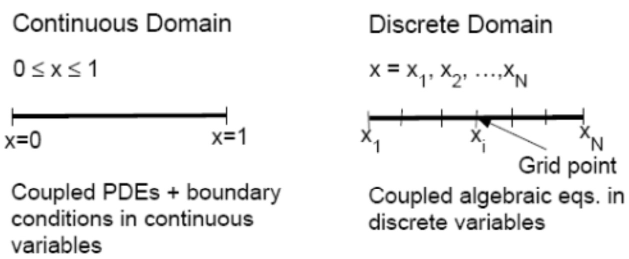
The Strategy of CFD

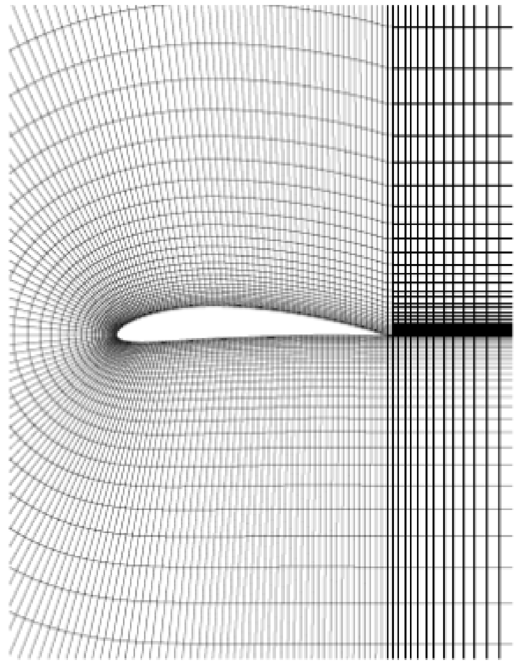
Broadly, the strategy of CFD is to replace the continuous problem domain with a discrete domain using a grid. In the continuous domain, each flow variable is defined at every point in the domain. For instance, the pressure p in the continuous 1D domain shown in the figure below would be given as

$$p = p(x),\ 0 < x < 1$$

In the discrete domain, each flow variable is defined only at the grid points. So, in the discrete domain shown below, the pressure would be defined only at the N grid points.

$$p_i = p(x_i), \quad i = 1, 2, \ldots, N$$





In a CFD solution, one would directly solve for the relevant flow variables only at the grid points. The values at other locations are determined by interpolating the values at the grid points.

The governing partial differential equations and boundary conditions are defined in terms of the continuous variables p, $\vec{V}$ etc. One can approximate these in the discrete domain in terms of the discrete variables p_i, $\vec{V}_i$ etc. The discrete system is a large set of coupled, algebraic equations in the discrete variables. Setting up the discrete system and solving it (which is a matrix inversion problem) involves a very large number of repetitive calculations, a task we humans palm over to the digital computer.

6-4 Example (2)

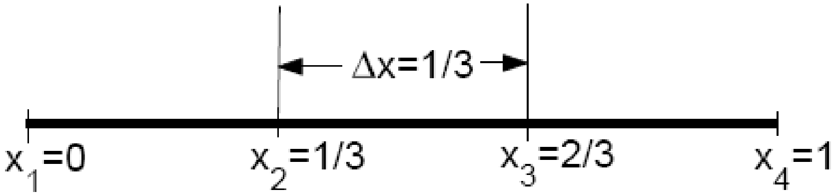
Discretization Using the Finite-Difference Method

To keep the details simple, we will illustrate the fundamental ideas underlying CFD by applying them to the following simple 1D equation:

$$\frac{du}{dx} + u^m = 0; \quad 0 \le x \le 1; \quad u(0) = 1 \tag{1}$$

We'll first consider the case where $m = 1$ when the equation is linear. We'll later consider the $m = 2$ case when the equation is nonlinear.

We'll derive a discrete representation of the above equation with $m = 1$ on the following grid:



6-4 Example (3)

This grid has four equally-spaced grid points with Δx being the spacing between successive points. Since the governing equation is valid at any grid point, we have

$$\left(\frac{du}{dx}\right)_i + u_i = 0 \qquad (2)$$

where the subscript i represents the value at grid point x_i. In order to get an expression for $(du/dx)_i$ in terms of u at the grid points, we expand u_{i-1} in a Taylor's series:

$$u_{i-1} = u_i - \Delta x \left(\frac{du}{dx}\right)_i + O(\Delta x^2)$$

Rearranging gives

$$\left(\frac{du}{dx}\right)_i = \frac{u_i - u_{i-1}}{\Delta x} + O(\Delta x) \qquad (3)$$

Using (3) in (2) and excluding higher-order terms in the Taylor's series, we get the following discrete equation:

$$\frac{u_i - u_{i-1}}{\Delta x} + u_i = 0 \qquad (4)$$

6-4 Example (4)

Assembly of Discrete System and Application of Boundary Conditions

Recall that the discrete equation that we obtained using the finite-difference method was

$$\frac{u_i - u_{i-1}}{\Delta x} + u_i = 0$$

Rearranging, we get

$$-u_{i-1} + (1 + \Delta x)u_i = 0$$

Applying this equation to the 1D grid shown earlier at grid points $i = 2, 3, 4$ gives

$$-u_1 + (1 + \Delta x)\, u_2 = 0 \qquad (i = 2) \tag{6}$$

$$-u_2 + (1 + \Delta x)\, u_3 = 0 \qquad (i = 3) \tag{7}$$

$$-u_3 + (1 + \Delta x)\, u_4 = 0 \qquad (i = 4) \tag{8}$$

The discrete equation cannot be applied at the left boundary (i=1) since u_{i-1} is not defined here. Instead, we use the boundary condition to get

$$u_1 = 1 \tag{9}$$

Equations (6)-(9) form a system of four simultaneous algebraic equations in the four unknowns u_1, u_2, u_3 and u_4. It's convenient to write this system in matrix form:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1+\Delta x & 0 & 0 \\ 0 & -1 & 1+\Delta x & 0 \\ 0 & 0 & -1 & 1+\Delta x \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

6-4 Example (5)

Solution of Discrete System

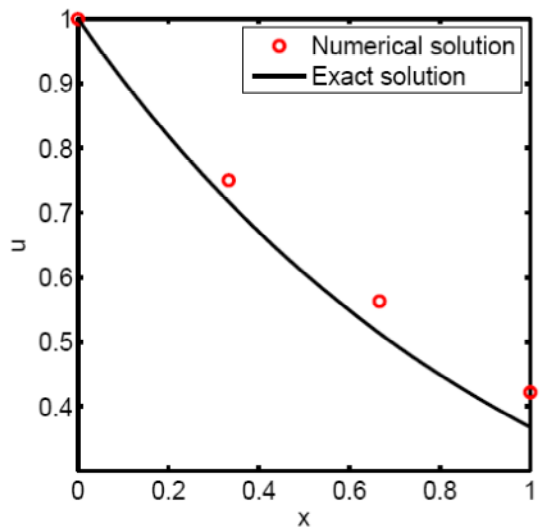
The discrete system (10) for our own humble 1D example can be easily inverted to obtain the unknowns at the grid points. Solving for u_1, u_2, u_3 and u_4 in turn and using $\Delta x = 1/3$, we get

$$u_1 = 1 \quad u_2 = 3/4 \quad u_3 = 9/16 \quad u_4 = 27/64$$

The exact solution for the 1D example is easily calculated to be

$$u_{exact} = \exp(-x)$$

The figure below shows the comparison of the discrete solution obtained on the four-point grid with the exact solution. The error is largest at the right boundary where it is equal to 14.7%.



6-4 Example (6)

Grid Convergence

While developing the finite-difference approximation for the 1D example, we saw that the truncation error in our discrete system is $O(\Delta x)$. So one expects that as the number of grid points is increased and Δx is reduced, the error in the numerical solution would decrease and the agreement between the numerical and exact solutions would get better.

Let's consider the effect of increasing the number of grid points N on the numerical solution of the 1D problem. We'll consider $N = 8$ and $N = 16$ in addition to the $N = 4$ case solved previously. We repeat the above assembly and solution steps on each of these additional grids. The resulting discrete system was solved using MATLAB. The following figure compares the results obtained on the three grids with the exact solution. As expected, the numerical error decreases as the number of grid points is increased.