Finite Difference (FD) Approximation

Consider $u \in C^l$.
Goal: Approximate derivative by finitely many function values:

$$\frac{\partial^k u}{\partial x^k}(x_0) \approx \sum_{i=0}^{m} a_i u(x_i) \qquad (k \leq l)$$

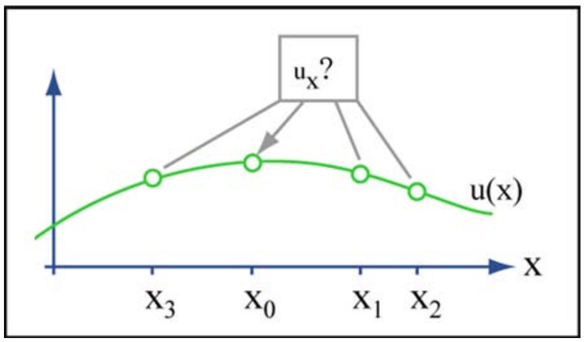

Image by MIT OpenCourseWare.

Vector of coefficients $a = (a_0, a_1, \cdots, a_m)$ is called FD stencil.
How to get stencil?

Taylor expansion

In 1D: $u(x) = u(x_0) + u_x(x_0) \cdot (x - x_0) + \frac{1}{2} u_{xx}(x_0) \cdot (x - x_0)^2 + \frac{1}{6} u_{xxx}(x_0) \cdot (x - x_0)^3 + O(|x - x_0|^4)$

Name $\bar{x}_i = x_i - x_0$

$$\Longrightarrow u(x_i) = u(x_0) + u_x(x_0) \cdot \bar{x}_i + \frac{1}{2} u_{xx}(x_0) \cdot \bar{x}_i^2 + \frac{1}{6} u_{xxx}(x_0) \cdot \bar{x}_i^3 + O(|\bar{x}_i|^4)$$

$$\Longrightarrow \sum_{i=0}^{m} a_i u(x_i) = u(x_0) \cdot \left(\sum_{i=0}^{m} a_i \right) + u_x(x_0) \cdot \left(\sum_{i=0}^{m} a_i \bar{x}_i \right) + u_{xx}(x_0) \cdot \left(\frac{1}{2} \sum_{i=0}^{m} a_i \bar{x}_i^2 \right) + O(h^3) \quad \text{where } \bar{x}_i \leq h \, \forall i.$$

Match coefficients:

$$\sum_{i=0}^{m} a_i u(x_i) \approx u_x(x_0) \Rightarrow \sum_i a_i = 0, \sum_i a_i \bar{x}_i = 1 \qquad \left[\sum a_i \bar{x}_i^2 \text{ small} \right]$$

$$\sum_{i=0}^{m} a_i u(x_i) \approx u_{xx}(x_0) \Rightarrow \sum_i a_i = 0, \sum_i a_i \bar{x}_i = 0, \sum_i a_i \bar{x}_i^2 = 2 \left[\sum a_i \bar{x}_i^3 \text{ small} \right]$$

etc.

Vandermonde matrix

$$V = V(x_0, x_1, \cdots, x_m) = \begin{bmatrix} 1 & \cdots\cdots & 1 \\ \bar{x}_0 & \cdots\cdots & \bar{x}_m \\ \bar{x}_0^2 & \cdots\cdots & \bar{x}_m^2 \\ \vdots & & \vdots \\ \bar{x}_0^k & \cdots\cdots & \bar{x}_m^k \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & \bar{x}_1 & \cdots & \bar{x}_m \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \bar{x}_1^k & \cdots & \bar{x}_m^k \end{bmatrix}$$

Constraints for stencil:

$V \cdot a = b$

linear system

$$k = 1 : \sum a_i u(x_i) \approx u_x(x_0) : \begin{bmatrix} 1 & \cdots & 1 \\ \bar{x}_0 & \cdots & \bar{x}_m \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$k = 2 : \sum a_i u(x_i) \approx u_{xx}(x_0) : \begin{bmatrix} 1 & \cdots & 1 \\ \bar{x}_0 & \cdots & \bar{x}_m \\ \bar{x}_0^2 & \cdots & \bar{x}_m^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

If $m = k \Longrightarrow$ In general one unique stencil a

If $m > k \Longrightarrow$ Multiple stencils

Can add additional criteria, e.g. require higher order.

Ex.: $k = 1, m = 1$

$$\begin{bmatrix} 1 & 1 \\ \hat{x}_0 & \hat{x}_1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$\bar{x}_0 = 0, \bar{x}_1 = h \Longrightarrow a_0 = -\frac{1}{h}, a_1 = \frac{1}{h}$

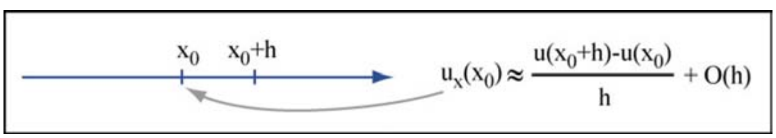

Image by MIT OpenCourseWare.

$\bar{x}_0 = 0, \bar{x}_1 = -h \Longrightarrow a_0 = \frac{1}{h}, a_1 = -\frac{1}{h}$

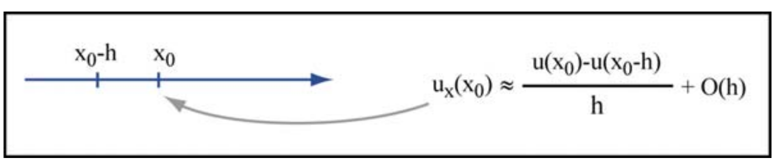

Image by MIT OpenCourseWare.

Ex.: $k = 1, m = 2 \qquad x = (x_0, x_0 + h, x_0 - h)$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & h & -h \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Longrightarrow \left\{ \begin{array}{l} a_1 - a_2 = \dfrac{1}{h} \\ a_0 = -a_1 - a_2 \end{array} \right\}$$

One-parameter family of stencils

Additional criterion: second order accuracy

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & h & -h \\ 0 & h^2 & h^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \implies a_0 = 0, a_1 = \frac{1}{2h}, a_2 = -\frac{1}{2h}$$

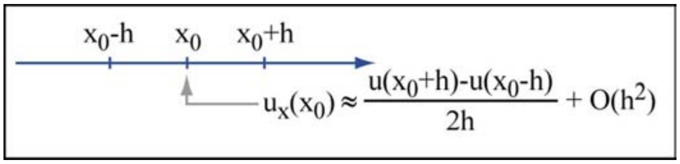

Image by MIT OpenCourseWare.

Ex.: $k = 2, m = 2$

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & \bar{x}_1 & \bar{x}_2 \\ 0 & \bar{x}_1^2 & \bar{x}_2^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} \implies \begin{cases} a_0 = -a_1 - a_2 \\ a_1 = \frac{2}{\bar{x}_1} \cdot (\bar{x}_1 - \bar{x}_2) \\ a_2 = \frac{2}{\bar{x}_2} \cdot (\bar{x}_2 - \bar{x}_1) \end{cases}$$

Equidistant: $x = (x_0, x_0 + h, x_0 - h)$

$a_0 = -\frac{2}{h^2},\ a_1 = a_2 = \frac{1}{h^2}$

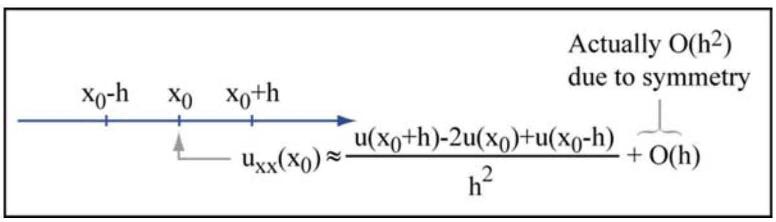

Image by MIT OpenCourseWare.

Higher space dimensions

2D $\quad \vec{x}_i = (x_i, y_i)$

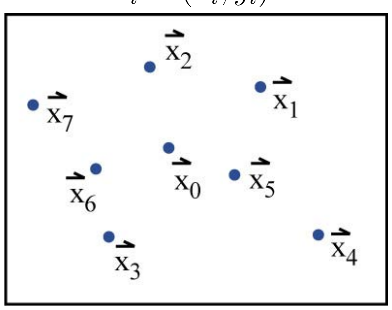

Image by MIT OpenCourseWare.

$$\begin{bmatrix} 1 & 1 & \cdots\cdots & 1 \\ 0 & \bar{x}_1 & \cdots\cdots & \bar{x}_m \\ 0 & \bar{y}_1 & \cdots\cdots & \bar{y}_m \\ 0 & \bar{x}_1^2 & \cdots\cdots & \bar{x}_m^2 \\ 0 & \bar{x}_1\bar{y}_1 & \cdots\cdots & \bar{x}_m\bar{y}_m \\ 0 & \bar{y}_1^2 & \cdots\cdots & \bar{y}_m^2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix} \leftarrow \text{stencil } \vec{a} \text{ for } u_{xx}(x_0)$$

3D

$$\begin{bmatrix} 1 & \cdots\cdots & 1 & \cdots\cdots & 1 \\ 0 & \cdots\cdots & \bar{x}_i & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{y}_i & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{z}_i & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{x}_i^2 & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{y}_i^2 & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{z}_i^2 & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{x}_i\bar{y}_i & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{x}_i\bar{z}_i & \cdots\cdots & \\ 0 & \cdots\cdots & \bar{y}_i\bar{z}_i & \cdots\cdots & \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 2 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leftarrow \text{stencil for } \nabla^2 u(x_0) = u_{xx} + u_{yy} + u_{zz}$$

Ex.: 2D $\nabla^2 u(x_0)$

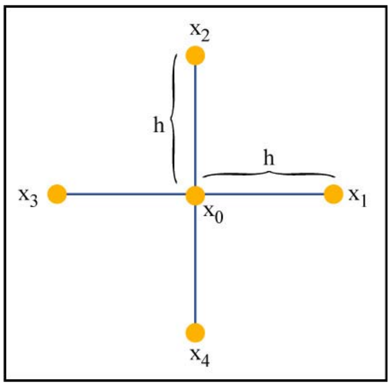

Image by MIT OpenCourseWare.

$$\vec{a} = \left(-\tfrac{4}{h^2}, \tfrac{1}{h^2}, \tfrac{1}{h^2}, \tfrac{1}{h^2}, \tfrac{1}{h^2}\right) = \left(-\tfrac{2}{h^2}, \tfrac{1}{h^2}, 0, \tfrac{1}{h^2}, 0\right) + \left(-\tfrac{2}{h^2}, 0, \tfrac{1}{h^2}, 0, \tfrac{1}{h^2}\right)$$

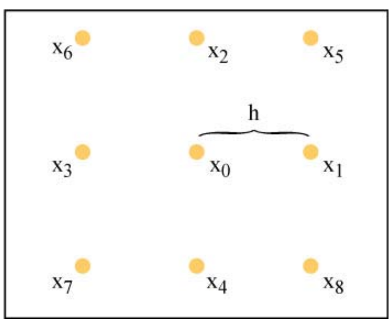

Image by MIT OpenCourseWare.

$\vec{a} = (\cdots \text{ exercise } \cdots)$

Poisson Equation

$$\boxed{\text{1D}}\ \left\{\begin{array}{ll} -u_{xx} = f(x) & \text{in }]0,1[\\ u(0) = a & \\ u_x(1) = c & \end{array}\right\} \begin{array}{l} \\ \leftarrow \text{Dirichlet boundary condition} \\ \leftarrow \text{Neumann boundary condition} \end{array}$$

Discretize on regular grid $\vec{x} = (0, h, 2h, \cdots, nh, 1)$, where $h = \frac{1}{n+1}$

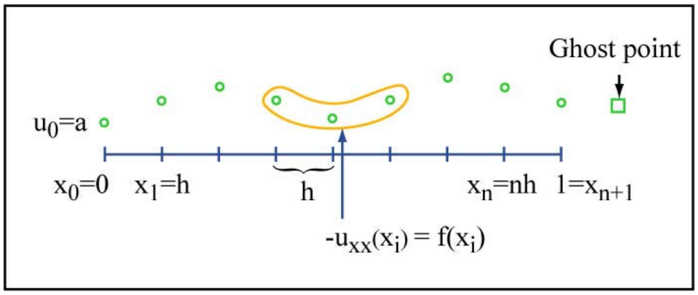

Image by MIT OpenCourseWare.

Interior: $f(x_i) = -u_{xx}(x_i) = \dfrac{-u(x_{i-1}) + 2u(x_i) - u(x_{i+1})}{h^2} + O(h^2)$

$$= \left(-\tfrac{1}{h^2}, \tfrac{2}{h^2}, -\tfrac{1}{h^2}\right) \cdot \begin{pmatrix} u_{i-1} \\ u_i \\ u_{i+1} \end{pmatrix} + O(h^2)$$

Dirichlet boundary condition: $u_0 = u(x_0) = a$ (exact)

Neumann boundary condition:

- Naive choice: $c = u_x(1)$

  $= \dfrac{u_{n+1} - u_n}{h} + O(h) = \left(-\dfrac{1}{h}, \dfrac{1}{h}\right) \cdot \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix} + O(h)$

  $O(h)$ on a single cell $\Longrightarrow$ Could preserve $O(h^2)$ globally, or drop accuracy to $O(h)$. Here the bad event happens.

- Second order approximation:

  $c = u_x(1) = \dfrac{u(x_{n+2}) - u(x_n)}{2h} + O(h^2)$

  Obtain u_{n+2} by $\dfrac{-u_n + 2u_{n+1} - u_{n+2}}{h^2} = f(1)$

  $\Longrightarrow \left(-\dfrac{1}{h}, \dfrac{1}{h}\right) \cdot \begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix} = c + \underbrace{\dfrac{h}{2} f(1)}_{\text{right hand side correction yields } 2^{\text{nd}} \text{ order}}$

- Alternative:

  $\left(-\tfrac{1}{2h}, \tfrac{2}{h}, -\tfrac{3}{2h}\right) \cdot \begin{pmatrix} u_{n-1} \\ u_n \\ u_{n+1} \end{pmatrix} = c$

  2^{nd} order one-sided stencil (check by $V \cdot a = b$).

Discretization generates linear system:

$$\underbrace{\begin{bmatrix} 1 & & & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & & -\frac{1}{h} & \frac{1}{h} \end{bmatrix}}_{A} \cdot \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \vdots \\ u_n \\ u_{n+1} \end{bmatrix}}_{\vec{u}} = \underbrace{\begin{bmatrix} a \\ f(x_1) \\ \vdots \\ \vdots \\ f(x_n) \\ c + \frac{h}{2} f(1) \end{bmatrix}}_{\vec{b}} \qquad (*)$$

Second order approximation (try it yourself!)

<u>Big Question</u>:

How to solve sparse linear systems $A \cdot \vec{u} = \vec{b}$?
$\rightarrow$ lecture 11.

<u>Rem.</u>: $(*) \Leftrightarrow (**)$

$$\begin{bmatrix} \frac{2}{h^2} & -\frac{1}{h^2} & & & & \\ -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{1}{h^2} & \frac{2}{h^2} & -\frac{1}{h^2} \\ & & & & -\frac{1}{h} & \underset{\uparrow}{\frac{1}{h}} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ \vdots \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f(x_1) + \frac{1}{h^2} a \\ f(x_2) \\ \vdots \\ \vdots \\ f(x_{n-1}) \\ f(x_n) + \frac{c}{h} + \frac{f(1)}{2} \end{bmatrix} \quad (**)$$

$\frac{1}{h^2}$ from Neumann boundary conditions

$$u_{n+1} = u_n + h\left(c + \frac{h}{2} f(1)\right)$$
$$\Longrightarrow -\frac{1}{h^2} u_{n+1} = \frac{1}{h^2} u_n + \frac{1}{h}\left(c + \frac{h}{2} f(1)\right)$$

Advantages:
- fewer equations
- matrix <u>symmetric</u>

MIT OpenCourseWare
http://ocw.mit.edu

18.336 Numerical Methods for Partial Differential Equations
Spring 2009

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.