- Tri-diagonal matrix

$$l_1 = \frac{a_2}{b_1}$$

$$\begin{pmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

GE

To eliminate a_2,

1 D, 2 M, 2 A

To make U,

$(n-1)$ D, $2(n-1)$ M,

$2(n-1)$ A

GE $\rightarrow$ $\Theta(n)$

After GE

$$\begin{pmatrix} b_1 & c_1 & & 0 \\ & b_2' & c_2' & \\ & & \ddots & \ddots \\ 0 & & & b_n' \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2' \\ \vdots \\ f_n' \end{pmatrix}$$

backward sweep

$x_n = f_n' / b_n'$

$x_j = (f_j' - c_j' x_{j+1}) / b_j'$

for $j = n-1, n-2, \cdots, 1$

$\rightarrow$ n D, $(n-1)$ M, $(n-1)$ A $\rightarrow \Theta(n)$

Total operations : $(2n-1)$ D, $3(n-1)$ M, $3(n-1)$ A

$\Rightarrow$ $\boxed{O(n) \text{ operations for the tri-diagonal matrix system}}$

① Computation of inverse matrix

$AB = I \qquad B = [\underline{b}_1, \underline{b}_2, \cdots, \underline{b}_n]$

$AB = [A\underline{b}_1, A\underline{b}_2, \cdots, A\underline{b}_n] = I$

$$e_j \equiv \begin{array}{c} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \end{array} \begin{array}{c} \\ \\ \\ \leftarrow j \\ \\ \\ \\ \end{array}$$

$\rightarrow A\underline{b}_1 = e_1, \; A\underline{b}_2 = e_2, \cdots, A\underline{b}_j = e_j, \cdots, A\underline{b}_n = e_n$

Augmented matrix $[A \; e_1 \; e_2 \cdots e_n] = [A \; I]$

Perform Gauss-Jordan on this augmented matrix to obtain A^{-1}.

ex) $A = \begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ tri-diagonal matrix $\quad A^{-1} = ?$ $\quad \frac{2}{3}+1 = \frac{5}{3}$

$$\left(\begin{array}{ccc|ccc} -2 & 1 & 0 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{array}\right) \to \begin{pmatrix} -2 & 1 & 0 & 1 & 0 & 0 \\ 0 & -\frac{3}{2} & 1 & \frac{1}{2} & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \to \begin{pmatrix} -2 & 1 & 0 & 1 & 0 & 0 \\ 0 & -\frac{3}{2} & 1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{5}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{pmatrix}$$

$\to I$

$$\to \cdots \to \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{3}{5} & -\frac{1}{5} & \frac{1}{5} \\ 0 & 1 & 0 & -\frac{1}{5} & -\frac{1}{5} & \frac{2}{5} \\ 0 & 0 & 1 & -\frac{1}{5} & \frac{2}{5} & \frac{3}{5} \end{array}\right)$$

A^{-1} full matrix

$Ax = b$
$x = A^{-1}b$

* inverse of a banded matrix is a full matrix.

Operation counts for A^{-1} : n^3 of each M and A $\sim$ similar to that of AB.

$n^3/3$

If A^{-1} is not needed, it should not be computed.

- LU decomposition

$$Ax=b \xrightarrow[GE]{} Ux=c$$

Operation of multiplication of one row by a constant and subtract from another row can be performed by a matrix multiplication.

e.g., multiply 1st row by e_{21} $(= -a_{21}/a_{11})$ and add to 2nd row.

$$E_{21} = \begin{pmatrix} 1 & & & 0 \\ e_{21} & 1 & & \\ 0 & & \ddots & \\ \vdots & & & \\ 0 & 0 & & 1 \end{pmatrix} \quad a_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \rightarrow E_{21}a_1 = \begin{pmatrix} a_{11} \\ 0 \\ a_{31} \\ \vdots \\ a_{n1} \end{pmatrix}$$

$$E_{31} = \begin{pmatrix} 1 & & & 0 \\ 0 & 1 & & \\ e_{31} & & 1 & \\ 0 & & & \ddots \\ \vdots & & & \\ 0 & & & 1 \end{pmatrix} \quad e_{31} = \frac{-a_{31}}{a_{11}} \rightarrow E_{31}a_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ 0 \\ a_{41} \\ \vdots \\ a_{n1} \end{pmatrix}$$

$$E_{31}E_{21} = \begin{pmatrix} 1 & & & & \\ e_{21} & 1 & & O & \\ e_{31} & & 1 & & \\ 0 & & & \ddots & \\ \vdots & O & & & \\ 0 & & & & 1 \end{pmatrix} \rightarrow E_{31}E_{21}a_1 = \begin{pmatrix} a_{11} \\ 0 \\ 0 \\ a_{41} \\ \vdots \\ a_{n1} \end{pmatrix}$$

To eliminate the 1st col.,

$$E_1 = \begin{pmatrix} 1 & & & O & \\ e_{21} & 1 & & & \\ e_{31} & & 1 & & \\ \vdots & O & & \ddots & \\ e_{n1} & & & & 1 \end{pmatrix} \qquad e_{j1} = -\frac{a_{j1}}{a_{11}} \qquad E_1 A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ \vdots & \vdots & & \vdots \\ 0 & a'_{n2} & \cdots & a'_{nn} \end{pmatrix}$$

To eliminate everything below a'_{22}

$$E_2 = \begin{pmatrix} 1 & & & O & \\ 0 & 1 & & & \\ 0 & e_{32} & 1 & & \\ \vdots & \vdots & O & \ddots & \\ 0 & e_{n2} & & & 1 \end{pmatrix} \qquad e_{j2} = -\frac{a'_{j2}}{a'_{22}} \Rightarrow E_2E_1A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a'_{22} & \cdots & a'_{2n} \\ 0 & 0 & a''_{33} & a''_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & a''_{n3} & a''_{nn} \end{pmatrix}$$

$\rightarrow \underbrace{E_{n-1} E_{n-2} \cdots E_2 E_1}_{= E \text{ : lower triangular matrix}} A = U$ (upper triangular matrix)

$A = LU$

$EA = U \rightarrow A = E^{-1} U$?

$$\underset{\overset{\shortparallel}{E_{21}^{-1}}}{E^{*}} E_{21} A = A$$

Let's undo E_{21} process.

$$E_{21} = \begin{pmatrix} 1 & & & 0 \\ e_{21} & 1 & & \\ 0 & & \ddots & \\ \vdots & 0 & & 1 \end{pmatrix} \rightarrow E_{21}^{-1} = \begin{pmatrix} 1 & & & 0 \\ -e_{21} & 1 & & \\ 0 & & \ddots & \\ \vdots & & & 1 \end{pmatrix} \qquad \underbrace{E_{21}^{-1} E_{21}}_{I} A = A$$

multiply 1st row by $-e_{21}$ and add to 2nd row.

$$E_1^{-1} = \begin{pmatrix} 1 & & & \\ -e_{21} & 1 & & O \\ -e_{31} & & 1 & \\ \vdots & O & & \ddots & \\ -e_{n1} & & & & 1 \end{pmatrix}, \quad E_2^{-1} = \cdot \qquad E_3^{-1} = \cdots$$

$O(n^3)$

$$EA = U \rightarrow E_{n-1}E_{n-2}\cdots E_1 A = U$$

$$A = \underbrace{E_1^{-1}E_2^{-1}\cdots E_{n-2}^{-1}E_{n-1}^{-1}}_{\text{lower triangular matrix with 1's on the main diagonal}}U = LU$$

$$Ax = b \rightarrow L\underbrace{Ux}_{=z} = b$$

$$Lz = b \rightarrow \begin{pmatrix} l_{11} & & & O \\ l_{21} & l_{22} & & \\ \vdots & & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

forward sweep: $z_1 = b_1 / \ell_{11}$ — $\Theta(n^2)$

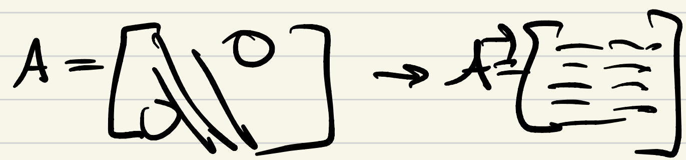
$$z_j = \Big(b_j - \sum_{k=1}^{j-1} \ell_{jk} z_k\Big) / \ell_{jj} \qquad j = 2, 3, \dots, n$$

$Ux = z$: backward sweep → $\Theta(n^2)$ to obtain x.

LU decomposition → $\Theta(n^3)$: expensive

$$Ax = b \rightarrow \begin{cases} x = A^{-1} b \\ LU x = b \end{cases}$$

→ direct solution



↳ LU much easier.

what if $a_{qq} = 0$?

$$\begin{pmatrix} 0 & 4 & -1 \\ 1 & 1 & 1 \\ 2 & -2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

→ row exchange

↓

permutation matrix

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow PA = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 4 & -1 \\ 2 & -2 & 1 \end{pmatrix}$$

$Ax = b$

$PAx = Pb$

$\overline{L}Ux = Pb = b'$

⑤ Round-off error

① Have a well-conditioned matrix, but the solution procedure is bad

→ We can suggest improvement in algorithm.

② Have an ill-conditioned matrix which is close to being singular.

→ no hope if it is due to round-off error.

* Computers store floating point numbers as product of a fractional part f with d digits and exponential part 10^s.

↳ $0.1 \le |f| < 1$

ex) $87.648 \rightarrow \underbrace{0.87648}_{f} \times 10^{2}$ s $d=5$

$100 \rightarrow \underbrace{0.1}_{f} \times 10^{3}$ s $d=1$

rounding is performed on f.

→ round-off error.

ex) $\begin{cases} 0.01x_1 + x_2 = 1 \\ 1x_1 - x_2 = 0 \end{cases}$ → exact sol. $x_1 = x_2 = \frac{1}{1.01} = 0.990099\ldots$

Do this example prob. with a computer that carries 2 significant figures. best sol. $x_1 = x_2 = 0.99$

GE : $\begin{cases} 0.01x_1 + x_2 = 1 \\ -101x_2 = -100 \end{cases}$ → $x_1 = 0$ wrong

$-0.101 \times 10^3 \xrightarrow{\text{rounding}} -0.10 \times 10^3$ → $x_2 = 1$

$l = \frac{a_{21}}{a_{11}}$

But do the same prob.

Pivot

$\begin{cases} 1x_1 - x_2 = 0 \\ 0.01x_1 + x_2 = 1 \end{cases} \xrightarrow{GE} \begin{cases} x_1 - x_2 = 0 \\ 1.01x_2 = 1 \end{cases}$ → $x_1 = 1$

$0.101 \times 10^1 \rightarrow 0.10 \times 10^1$ → $x_2 = 1$

→ choice of pivot is very important. very accurate!

By row changes, make the pivot the largest elt in the column.

ex) $\begin{cases} 1x_1 + 100x_2 = 100 \\ 1x_1 - x_2 = 0 \end{cases} \xrightarrow{GE} \xrightarrow{BS} \left.\begin{matrix} x_1 = 0 \\ x_2 = 1 \end{matrix}\right)$ wrong.

$\Rightarrow$ We need scaling.

Normalize each eq. in advance s.t. the largest elt in each row is 1.

Scaling $\rightarrow \begin{cases} 0.01x_1 + x_2 = 1 \\ x_1 - x_2 = 0 \end{cases} \xrightarrow[\text{pivot}]{} \begin{cases} x_1 - x_2 = 0 \\ 0.01x_1 + x_2 = 1 \end{cases}$

$\rightarrow GE \rightarrow BS \rightarrow x_1 = x_2 =$

② example of ill-conditioned matrix

$\begin{cases} x_1 + x_2 = 2 \\ x_1 + 1.0001x_2 = 2 \end{cases} \xrightarrow{\text{exact sol.}} \left(\begin{matrix} x_1 = 2 \\ x_2 = 0 \end{matrix}\right.$

perturb the RHS

$$\begin{cases} x_1 + x_2 = 2 \\ x_1 + 1.0001\, x_2 = 2.0001 \end{cases} \xrightarrow[\text{sol.}]{\text{exact}} \begin{cases} x_1 = 1 \\ x_2 = 1 \end{cases}$$

slight change in b $\rightarrow$ large change in x

$\rightarrow$ No numerical method can avoid this sensitivity to small perturbations.

Q: Given $Ax=b$, what is the change in x w.r.t. a change in the parameter of the system b?

$Ax=b$

error in b, $\delta b \rightarrow x+\delta x$. find δx.

$Ax=b$ & $A(x+\delta x) = b+\delta b$

$\rightarrow A\delta x = \delta b \rightarrow \delta x = A^{-1}\delta b$

$\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\|\,\|\delta b\|$ (schwartz inequility)

$Ax=b \rightarrow \|b\| = \|Ax\| \leq \|A\|\,\|x\|$

$\|A\|$: norm $= \lambda_{max}(A)$

$$\Rightarrow \frac{\|\delta x\|}{\|A\|\,\|x\|} \leq \frac{\|A^{-1}\|\,\|\delta b\|}{\|b\|}$$

$$\rightarrow \boxed{\frac{\|\delta x\|}{\|x\|} \leq \|A\|\,\|A^{-1}\| \boxed{\frac{\|\delta b\|}{\|b\|}}}$$

amplification factor

condition number of $A = \dfrac{\lambda_{max}}{\lambda_{min}}$

$|A-\lambda I|=0$

$x_1 + x_2 = 2$

$x_1 + 1.0001 x_2 = 2$

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1.0001 \end{pmatrix} \rightarrow \|A\| = 2.00005 \simeq 2$$

$$A^{-1} = \begin{pmatrix} 10001 & -10000 \\ -10000 & 10000 \end{pmatrix} \rightarrow \|A^{-1}\| = 20{,}000$$

$$\Rightarrow \frac{\|\delta x\|}{\|x\|} \leq 2 \times 20000 \frac{\|\delta b\|}{\|b\|} = 40{,}000 \frac{\|\delta b\|}{\|b\|}$$

$\rightarrow$ matrix A is stiff or ill-conditioned.

If the condition number times the order of round-off accuracy of machine is order 1, be very careful!

condition number $= 10^4$

order of round-off error accuracy $= 10^{-4}$

⊙ Cayley-Hamilton theorem

- Every matrix satisfies its own characteristic eq.

$$\underset{n\times n}{A} \ : \quad P(\lambda) = \det(A-\lambda I) = 0$$

$$\rightarrow \lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} + \cdots + c_{n-1}\lambda + c_n = 0$$

$$\Rightarrow \boxed{A^n + c_1 A^{n-1} + c_2 A^{n-2} + \cdots + c_{n-1}A + c_n I = 0}$$

$$P(A) = 0$$

n: integer

$$\underset{2\times 2}{T} = \underset{2\times 2}{f(D)} = \boxed{d_0 I + d_1 D} + d_2 \cancel{D^2} + d_3\cancel{D^3} + d_4\cancel{D^4} + \cdots$$

$\alpha I + \beta D$ (2x2)

- Any power of an nxn matrix can be represented as a polynomial of degree up to n-1.

ex) $A = \begin{pmatrix} 1 & 2 \\ 0 & 4 \end{pmatrix}$, $\sqrt{A} = ?$

$\lambda_1 = 1,\ \lambda_2 = 4$

$$\sqrt{A} = \alpha_0 I + \alpha_1 A$$

$$\rightarrow \left.\begin{array}{l} \sqrt{\lambda_1} = \alpha_0 + \alpha_1\lambda_1 \\ \sqrt{\lambda_2} = \alpha_0 + \alpha_1\lambda_2 \end{array}\right] \rightarrow \begin{array}{l} \alpha_0 = 2/3 \\ \alpha_1 = 1/3 \end{array}$$

$$\rightarrow \sqrt{A} = \frac{2}{3}I + \frac{1}{3}A = \begin{pmatrix} 1 & \frac{2}{3} \\ 0 & 2 \end{pmatrix}$$

- Any function of a matrix $(n \times n)$ can be represented as a polynomial of degree up to $n-1$.

ex) $A = \begin{pmatrix} \pi & 3\pi \\ 2\pi & 2\pi \end{pmatrix}$, $\cos A = ?$ $\lambda_1 = 4\pi$, $\lambda_2 = -\pi$

$$\cos A = \alpha_0 I + \alpha_1 A \rightarrow \left.\begin{matrix} \cos\lambda_1 = \alpha_0 + \alpha_1\lambda_1 \\ \cos\lambda_2 = \alpha_0 + \alpha_1\lambda_2 \end{matrix}\right\} \rightarrow \begin{matrix} \alpha_0 = -\frac{3}{5} \\ \alpha_1 = \frac{2}{5\pi} \end{matrix}$$

$$\cos A = -\frac{3}{5}I + \frac{2}{5\pi}A = \begin{pmatrix} -\frac{1}{5} & \frac{6}{5} \\ \frac{4}{5} & \frac{1}{5} \end{pmatrix}.$$

ex) $A = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, $\sqrt{A} = ?$

$$\sqrt{A} = \alpha_0 I + \alpha_1 A \rightarrow \sqrt{\lambda_1} = \alpha_0 + \alpha_1\lambda_1$$

$\lambda_1 = \lambda_2 = 2$

double root

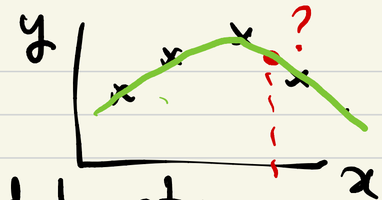
$$P(\lambda) = (\lambda-\lambda_1)^2(\lambda-\lambda_2)\cdots(\lambda-\lambda_{n-1})$$

$$\rightarrow P'|_{\lambda=\lambda_1} = 2(\lambda-\lambda_1)\cdots + \cdots + \cdots = 0$$

$$\frac{1}{2}A^{-1/2} = \alpha_1 \rightarrow \frac{1}{2}\lambda_1^{-\frac{1}{2}} = \alpha_1$$

α_0, α_1 $\rightarrow \sqrt{A} = \begin{pmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{pmatrix}$

Ch. 1 Interpolation



- Discrete data (x_i, y_i) $i = 0, 1, 2, \cdots, n$
  - → Find value of y at a pt. between two data pts.
  - → Fix a smooth curve through the data
- If a data is from a crude experiment with some uncertainty, it is best to use the method of least square errors.

- Polynomial interpolation (x_i, y_i) $i = 0, 1, 2, \cdots, n$ $(n+1)$ pts.

Fix a polynomial of degree n through $n+1$ pts.

$$P = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

Find the coeffs. $a_0, a_1, \cdots, a_n$, by passing the polynomial going through the data pts.

$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n \qquad i = 0, 1, 2, \ldots, n$$

n+1 unknowns n+1 eqs

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \rightarrow O(n^3) \text{ expensive!}$$

A : full matrix $(n+1) \times (n+1)$

For large n, the resulting system of eqs. becomes ill-conditioned in general.

In pracice, we define the polynomial in an explicit way as opposed to solving a matrix system.

- Lagrange polynomial

Define a <u>polynomial of degree n</u> associated with each pt. x_j

$$L_j(x) = \alpha_j (x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)$$

$$= \alpha_j \prod_{\substack{i=0 \\ i\neq j}}^{n} (x-x_i)$$

$$L_j(x_k) = 0 \quad \text{if} \quad k \neq j$$

$$L_j(x_j) = \alpha_j \prod_{\substack{i=0 \\ i\neq j}}^{n} (x_j - x_i) \Rightarrow \alpha_j \equiv 1 \Big/ \prod_{\substack{i=0 \\ i\neq j}}^{n} (x_j - x_i)$$

$$\Rightarrow L_j(x_j) = \begin{cases} 0 & k\neq j \\ 1 & k=j \end{cases}$$

The desired polynomial $\boxed{P(x) = \sum_{j=0}^{n} y_j L_j(x)}$ Lagrange polynomial

$$P(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) + \cdots + y_i L_i(x) + \cdots + y_n L_n(x)$$

ex)

i	0	1	2	3
x_i	1	2	4	8
y_i	1	3	7	11

$y(x=7) = ?$

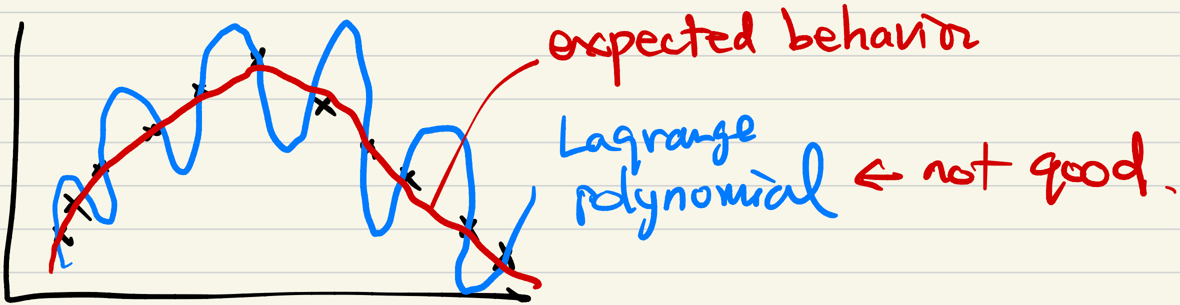
$$P(x) = \underline{1} \cdot L_0(x) + \underline{3} L_1(x) + \underline{7} L_2(x) + \underline{11} L_3(x)$$

$$L_0(x) = \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)}$$

$L_0(7) = 0.714$

$$y(7) = P(7) = \underline{L_0(7)} + 3\,\underline{L_1(7)} + 7\,\underline{L_2(7)} + 11\,\underline{L_3(7)} = 10.857$$

11 pts. → poly. of degree 10

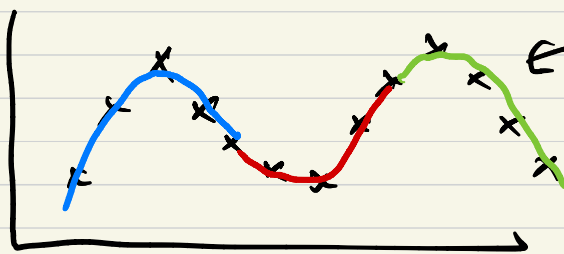


Although the polynomial is tied down at the data pts., it can wander in between. → inaccurate

→ Piecewise Lagrange polynomial interpolation

instead of fitting only one poly. to all the data.

→ Lower-order polys. to sections of data.

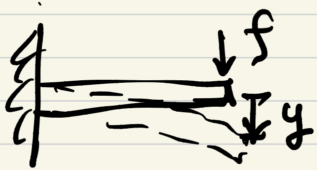


← Piecewise Lagrange polynomial interpolation.

⇒ derivatives at the boundaries of the segments are discontinuous.

⇒ splines!

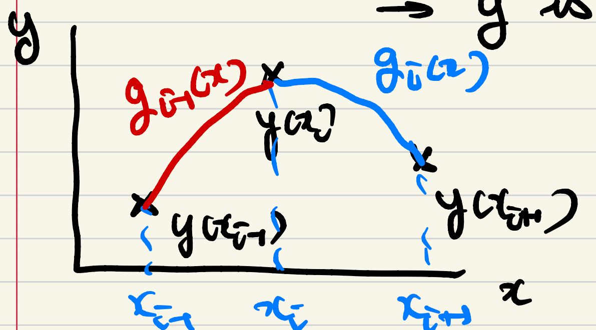
- ## Spline interpolation ✲



$$EIy^{(iv)} = f$$

between the data pts., $f = 0 \Rightarrow EIy^{(iv)} = 0$

$\rightarrow$ y is cuvic in between $x_i < x < x_{i+1}$.



For ith interval $(x_i \le x \le x_{i+1})$

$$g_i(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$g_i(x_i) = y(x_i) = y_i, \quad g_i(x_{i+1}) = y(x_{i+1}) = y_{i+1}$$

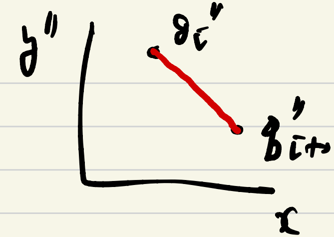
Match the 1st & 2nd derivatives from the adjacent intervals.

$$\begin{cases} g_i'(x_i) = g_{i-1}'(x_i) \\ g_i''(x_i) = g_{i-1}''(x_i) \end{cases}$$

$$g_i' = a_1 + 2a_2 x + 3a_3 x^2$$

$$g_i'' = 2a_2 + 6a_3 x$$

$$\rightarrow g_i''(x) = g''(x_i)\,\frac{x - x_{i+1}}{x_i - x_{i+1}} + g''(x_{i+1})\,\frac{x - x_i}{x_{i+1} - x_i}$$



Integrate twice to get the cubic poly.

$$g_i'(x) = g''(x_i)\,\frac{(x - x_{i+1})^2}{2(x_i - x_{i+1})} + g''(x_{i+1})\,\frac{(x - x_i)^2}{2(x_{i+1} - x_i)} + c_1$$

$$g_i(x) = g''(x_i)\,\frac{(x - x_{i+1})^3}{6(x_i - x_{i+1})} + g''(x_{i+1})\,\frac{(x - x_i)^3}{6(x_{i+1} - x_i)} + c_1 x + c_2$$

obtain c_1 and c_2 by $g_i(x_i) = y_i$ and $g_i(x_{i+1}) = y_{i+1}$

$$\rightarrow g_i(x) = \frac{1}{6} g''(x_i)\left[\frac{(x_{i+1} - x)^3}{(x_{i+1} - x_i)} - (x_{i+1} - x_i)(x_{i+1} - x)\right]$$

$$+ \frac{1}{6} g''(x_{i+1})\left[\frac{(x - x_i)^3}{(x_{i+1} - x_i)} - (x_{i+1} - x_i)(x - x_i)\right]$$

$$+ y_i\,\frac{x_{i+1} - x}{x_{i+1} - x_i} + y_{i+1}\,\frac{x - x_i}{x_{i+1} - x_i}$$

Apply $g_i'(x_i) = g_{i-1}'(x_i)$

$$\rightarrow \frac{\Delta_{i-1}}{6} g''(x_{i-1}) + \frac{\Delta_{i-1}+\Delta_i}{3} g''(x_i) + \frac{\Delta_i}{6} g''(x_{i+1}) = \frac{y_{i+1}-y_i}{\Delta_i} + \frac{y_{i-1}-y_i}{\Delta_{i-1}}$$

$\Delta_i = x_{i+1} - x_i$

$\Delta_{i-1} = x_i - x_{i-1}$

for $i = 1, 2, \cdots, n-1$ $\quad$ $n-1$ eqs.

$g''(x_0), g''(x_1), \cdots, g''(x_n)$ $\quad$ $n+1$ unknowns $\quad$ g_0'' & g_n''

$$\begin{pmatrix} \ddots & \ddots & & 0 \\ & \ddots & \ddots & \\ \frac{\Delta_{i-1}}{6} & \frac{\Delta_{i-1}+\Delta_i}{3} & \frac{\Delta_i}{3} & \\ 0 & & \ddots & \ddots \end{pmatrix} \begin{pmatrix} g_1'' \\ g_2'' \\ g_i'' \\ g_{n-1}'' \end{pmatrix} = \begin{pmatrix} \cdots + g_0'' \\ \\ \\ \cdots + g_n'' \end{pmatrix}$$

tri-diagonal matrix

end conditions ?

Suppl video lecture will be uploaded on eTL tomorrow 5-6pm.