CSS322 Midterm Summary

System of Linear Equations (L2)

Solve n equations with n unknowns

↳ $Ax = b$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

★ Unique solution

⇒ A is nonsingular matrix

$\det(A) \neq 0$

⇒ solution $x = A^{-1} b$

① L: Lower-Triangular Matrix

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & & \vdots \\ \vdots & & \ddots & 0 \\ l_{n1} & \cdots & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Solve by "forward substitution"

② U: Upper-Triangular Matrix

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Solve by "backward substitution"

LU-factorization / -decomposition

★ $A = LU$ ★

↳ Plain GE (Guassian Elimination)

$$\begin{bmatrix} a_{11} & \cdots & \cdots & \cdots \\ \vdots & a_{22} & & \vdots \\ \vdots & & a_{33} \ddots & \vdots \\ \vdots & \cdots & \cdots & a_{nn} \end{bmatrix}$$

! Pivot may be 0 even if matrix is nonsingular

Solve linear equations →

$Ax = b$

① $A = LU$

② $Lw = b$ → Solve for w → "forward substitution"

③ $Ux = w$ → Solve for x → "backward substitution"

ex
$$\begin{cases} x_1 + 2x_2 + x_3 = -1 \\ -3x_1 + x_2 + x_3 = 0 \\ x_1 \qquad\quad + 3x_3 = 1 \end{cases}$$

$Ax = b$

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

Plain GE ;

$$\begin{array}{r} \\ R_2 - (-3)R_1 \\ R_3 - (1)R_1 \end{array}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 7 & 4 \\ 0 & -2 & 2 \end{bmatrix} \longrightarrow \begin{array}{r} \\ R_2 \\ R_3 - \left(\frac{-2}{7}\right)R_2 \end{array}\begin{bmatrix} 1 & 2 & 1 \\ 0 & 7 & 4 \\ 0 & 0 & 22/7 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -3 & & \\ 1 & & \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 1 & -2/7 & 1 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 1 & -2/7 & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 7 & 4 \\ 0 & 0 & 22/7 \end{bmatrix}$$

② → $Lw = b$

$$\begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 1 & -2/7 & 1 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

→ $w_1 = -1$

→ $-3w_1 + w_2 = 0$

$w_2 = 3w_1 = \underbrace{3(-1)} = -3$

→ $\overset{-1}{\cancel{w_1}} - \frac{2}{7}\overset{-3}{\cancel{w_2}} + w_3 = 1$

$w_3 = 8/7$

$$\Rightarrow w = \begin{bmatrix} -1 \\ -3 \\ 8/7 \end{bmatrix}$$

③ → $Ux = w$

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 7 & 4 \\ 0 & 0 & 22/7 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -3 \\ 8/7 \end{bmatrix}$$

→ $\frac{22}{7}x_3 = \frac{8}{7} \Rightarrow x_3 = \frac{4}{11}$

$7x_2 + 4x_3 = -3$

$7x_2 + \frac{16}{11} = -3$

$x_2 = \left(-3 - \frac{16}{11}\right)/7 = -7/11$

$x_1 + 2x_2 + x_3 = -1$

$x_1 = -1 - 2\left(\frac{-7}{11}\right) - \frac{4}{11} = \frac{-1}{11}$

$$x = \begin{bmatrix} -1/11 \\ -7/11 \\ 4/11 \end{bmatrix}$$ #

How to solve problem of Plain GE?

→ Permutation matrix P → we can exchange rows
↳ only contains 0's and 1's
↳ 1's per row / per column (1)

ex. $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

mat. A perm. mat. P.

left — PA ⇒ exchange rows

AP ⇒ exchange column
right

$PP^T = I$ ⇒ $P^T = P^{-1}$

"$P^T LU$ factorization"

→ GEPP

Guassian Elimination with parital pivoting

→ select pivot at maximum absolute value

$A = P^T LU$ → solve linear equation of $Ax = b$

① $A = P^T LU$
② $\hat{b} = Pb$ ⇒ exchange row of b
③ $Lw = \hat{b}$
④ $Ux = w$ ⇒ x is ans.

ex: $\begin{bmatrix} 1 & 2 & 1 \\ -3 & 1 & 1 \\ 1 & 0 & 3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$

exchange Row 1 & 2

① $\begin{bmatrix} 1 & 2 & 1 \\ -3 & 1 & 1 \\ 1 & 0 & 3 \end{bmatrix} \to \begin{bmatrix} -3 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 3 \end{bmatrix} \xrightarrow[R_3 - \left(\frac{1}{-3}\right)R_1]{R_2 - \left(\frac{1}{-3}\right)R_1} \begin{bmatrix} -3 & 1 & 1 \\ 0 & 7/3 & 4/3 \\ 0 & 1/3 & 10/3 \end{bmatrix} \xrightarrow{R_3 - \frac{1/3}{7/3}R_2} \begin{bmatrix} -3 & 1 & 1 \\ 0 & 7/3 & 4/3 \\ 0 & 0 & 66/21 \end{bmatrix} = U$

$P: \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$L = \begin{bmatrix} 1 & 0 & 0 \\ -1/3 & & \\ -1/3 & & \end{bmatrix}$ $L = \begin{bmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1/3 & 1/7 & 1 \end{bmatrix}$

② $\hat{b} = Pb$ ⇒ $\hat{b} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$

③ $Lw = \hat{b}$ ⇒ $\begin{bmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1/3 & 1/7 & 1 \end{bmatrix}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}$

⇒ $\begin{cases} w_1 = 0 \\ \cancel{-\frac{1}{3}w_1} + w_2 = -1 \Rightarrow w_2 = -1 \\ \cancel{-\frac{1}{3}w_1} + \frac{1}{7}\underset{(-1)}{w_2} + w_3 = 1 \end{cases}$

$w_3 = 1 + \frac{1}{7} = \frac{8}{7}$

$w = \begin{bmatrix} 0 \\ -1 \\ 8/7 \end{bmatrix}$

④ $Ux = w$ $\begin{bmatrix} -3 & 1 & 1 \\ 0 & 7/3 & 4/3 \\ 0 & 0 & 66/21 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 8/7 \end{bmatrix}$

$\frac{66}{21}x_3 = \frac{8}{7} \Rightarrow x_3 = \frac{8}{7} \times \frac{21}{66} = \frac{4}{11}$

$\frac{7}{3}x_2 + \frac{4}{3}x_3 = -1$

$x_2 = \left[-1 - \frac{4}{3}\left(\frac{4}{11}\right)\right] \times \frac{3}{7}$

$= -7/11$

$-3x_1 + x_2 + x_3 = 0$

$x_1 = \left[\left(-\frac{4}{11}\right) - \left(-\frac{7}{11}\right)\right] / -3 = \frac{3}{11} \times \frac{1}{-3} = \frac{-1}{11}$

$\therefore x = \begin{bmatrix} -1/11 \\ -7/11 \\ 4/11 \end{bmatrix}$ ✱

finding inverse matrix?

* $AA^{-1} = I$ * GEPP is the best way to solve $Ax = b$

$A[x_1\ x_2 \cdots x_n] = [e_1\ e_2 \cdots e_n]$

$[Ax_1\ Ax_2 \cdots Ax_n] = [e_1\ e_2 \cdots e_n]$

↳ $Ax_1 = e_1$; $Ax_2 = e_2$, $Ax_3 = e_3$

$$A^{-1} \begin{bmatrix} \vdots & \vdots & & \vdots \\ x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

* factorization ($P^T LU$) of A only once then use it to solve $Ax_i = e_i$ $i = 1, \dots, n$ ⟶ 4 step method

$\hat{b}_i = Pb_i \rightarrow Lw_i = \hat{b}_i \rightarrow Ux_i = w_i$

Norms (L3)

* Triangle inequality : $\|x+y\| \leq \|x\| + \|y\|$

Vector norms

① 2-norms / euclidean norms

$$\|v\|_2 = \sqrt{\sum_{i=1}^{n} v_i^2}$$

② 1-norms

$$\|v\|_1 = \sum_{i=1}^{n} |v_i|$$

③ ∞-norm

$$\|v\|_\infty = \max_{i=1}^{n} \{ |v_i| \}$$

(maximum absolute value)

④ p-norm

$$\|v\|_p = (|v_1|^p + |v_2|^p + \dots + |v_n|^p)^{1/p}$$

; $p \geq 1$

Matrix norms

* Frobenius norm

$$\|B\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^2}$$

* 1-norms

$\|B\|_1$ = maximum absolute column sum

$$A = \begin{bmatrix} 1 & -2 \\ -3 & 4 \end{bmatrix} \Rightarrow \|A\|_1 = \max\{|1|+|-3|, |-2|+|4|\}$$
$$= \max\{4, 6\} = 6$$

* ∞-norm

$\|B\|_\infty$ = maximum absolute row sum

$$\Rightarrow \|A\|_\infty = \max\{|1|+|-2|, |-3|+|4|\}$$
$$= \max\{3, 7\}$$
$$= 7$$

Approximation (L4)

- store decimal number in computer using floating-point format

  $\Rightarrow$ can cause inaccurate / round off / inexactness

error

absolute error = | approx. value - true value |

relative error = $\dfrac{\text{absolute error}}{\text{true value}}$

→ $\hat{x}$ → approximate number → $\dfrac{|\hat{x}-x|}{|x|}$

↳ approximate vector → $\dfrac{\|\hat{x}-x\|}{\|x\|}$

CATASTROPHIC CANCELLATION

arithmetic $+ - \times \div$

- subtracting 2 number having same sign and similar magnitude, result become smaller magnitude

can be resulted in very large error!

floating-point arithmetic

$$X = \underbrace{(d_0 . d_1 d_2 \dots d_{p-1})}_{p \text{ digits}} \times 2^{E}$$

E → exponent.

p digits

↳ precision

float : p = 23

double : p = 52

ex. $18 = (10010)_2 = (1.001)_2 \times 2^4$

$0.875 = \frac{7}{8} = \frac{4}{8} + \frac{2}{8} + \frac{1}{8}$

$= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = (0.111)_2 \times 2^0$

$= (1.11)_2 \times$

unit round off.

↳ $\epsilon_{mach} \Rightarrow 2^{-p}$

ex. $p = 3$

$x = 12 \Rightarrow (1.10)_2 \times 2^3$

$y = 3 \Rightarrow (1.10)_2 \times 2^1$

$z = x - y = 9 \Rightarrow (1.001) \times 2^3$

round off

$\Rightarrow (1.00) \times 2^3 = 8$

relative error $= \dfrac{|8-9|}{|9|} = 1/9$

Conditioning & Stability (L5)

- well-condition → input has a small change, output has a reasonable change.
- ill-condition → output has a very large change

* even in the same problem → it can be both well-/ill-condition at the same time

- Condition number
  → ratio of relative change in solution/output to relative change in input.

$$\text{cond. number} = \frac{\text{relative change in solution}}{\text{relative change in input}}$$

→ close to 1 → "well-condition"
otherwise → "ill-condition"

A some is hard to compute due to complexity, thus upper bound or rough estimation is used instead.

cond. number of nonsingular square matrix ⇒ $\text{cond}(A) = \kappa(A) = \|A\| \cdot \|A^{-1}\|$ → the larger cond(A), the more ill-condition

* A is singular → $\text{cond}(A) = \infty$

- Stability
  - an algorithm is stable if the result it produce is relatively insensitive to perturbation due to approximation during computation

$f(x) + \epsilon_f$

- forward stable, produce nearby result for exact problem

- Backward stable, produce exact result for nearby problem $f(x + \epsilon_b)$
  ↳ tend to use more in practical

* we can expect accurate solution from well-condition problem using stable algorithm

$f(x+\epsilon_b) + \epsilon_f$

- conditionally stable, produce nearby result for nearby problem

* Plain GE is unstable
GEPP is backward stable

Cholesky factorization (L6)

→ any positive definite symmetric matrix A can be factored into $A = LL^T$ ($L \Rightarrow$ lower triangular matrix)

- Symmetric matrix $\Rightarrow A = A^T$

$$A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{21} & a_{22} & & \vdots \\ \vdots & & \ddots & \\ a_{n1} & \cdots & & a_{nn} \end{bmatrix}$$

- The efficient way to calculate $A = LL^T$

$$\underset{A}{\begin{bmatrix} a_{11} & a_{21} \\ a_{21} & a_{22} \end{bmatrix}} = \underset{L}{\begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix}} \underset{L^T}{\begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix}}$$

① $a_{11} = l_{11}^2 \Rightarrow l_{11} = \pm\sqrt{a_{11}}$ → (for the sake of sod, we will use only positive/larger value)

② $a_{21} = l_{11} l_{21} \Rightarrow l_{21} = a_{21}/l_{11}$

③ $a_{22} = l_{21}^2 + l_{22}^2 \Rightarrow l_{22} = \pm\sqrt{a_{22} - l_{21}^2}$

- this concept can be applied to all symmetric $n \times n$ matrix.

$$\underset{A}{\begin{bmatrix} a_{11} & a_{21} & a_{31} \\ & a_{22} & a_{32} \\ & & a_{33} \end{bmatrix}} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & l_{31} \\ & l_{22} & l_{32} \\ & & l_{33} \end{bmatrix}$$

$a_{11} = l_{11}^2$

$a_{21} = l_{11} l_{21}$

$a_{31} = l_{11} l_{31}$

$a_{22} = l_{21}^2 + l_{22}^2$

$a_{32} = l_{21} l_{31} + l_{22} l_{32}$

$a_{33} = l_{31}^2 + l_{32}^2 + l_{33}^2$

- This algorithm is WELL-DEFINED & cheaper than GE/GEPP → the best way to calculate $Ax = b$ when A is symmetric matrix

→ Solving linear system w/ Cholesky factorization ($Ax = b$)

① $A = LL^T$ (factor by using Cholesky factorization)

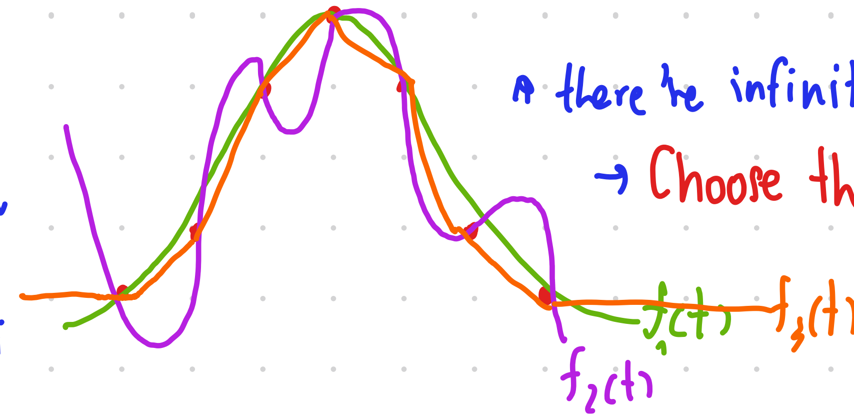
② $Lw = b$ (solve for w, using forward substitution)

③ $L^T x = w$ (solve for x, using back substitution)

Interpolation (L7)

we have n points (t_i, y_i) for $i = 1, \dots, n$

we want to find function that $f(t_i) = y_i$



* there're infinitely many interpolating function

→ Choose the simplest one.

Polynomial Interpolation

simplest / continuous / differentiable / easy to evaluate

monomial basis

n points → polynomial of degree $n-1$

$$P_{n-1}(t) = x_1 + x_2 t + x_3 t^2 + \dots + x_n t^{n-1}$$

→ we need to determine x_i ; $i = 1 \dots n$

$$\underbrace{\begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & & t_3 & \ddots & \vdots \\ 1 & t_n & t_n^2 & \cdots & t_n^{n-1} \end{bmatrix}}_{\text{Vandermonde matrix}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

⇒ solve for x

you can use GEPP!

(& 4 step method)

Vandermonde matrix → if all t_i are distinct

→ non singular matrix

→ improve efficiency by "Horner's method"

calculate faster

$$= P_{n-1}(t) = x_1 + t(x_2 + t(x_3 + t(\dots(x_{n-1} + x_n t)\dots)))$$

ex. $2t^3 + 4t^2 - 3t + 1 = 1 - 3t + 4t^2 + 2t^3$

$= 1 + t(-3 + 4t + 2t^2)$

$= 1 + t(-3 + t(4 + 2t))$

Lagrange Interpolation

$$l_j(t) = \frac{\prod_{k=1, k\neq j}^{n} (t - t_k)}{\prod_{k=1, k\neq j}^{n} (t_j - t_k)}, \quad j = 1, \dots, n.$$

$$\to \ l_1(t) = \frac{(t-t_2)(t-t_3)\cdots(t-t_n)}{(t_1-t_2)(t_1-t_3)\cdots(t_1-t_n)}$$

$$l_2(t) = \frac{(t-t_1)(t-t_3)(t-t_4)\cdots(t-t_n)}{(t_2-t_1)(t_2-t_3)\cdots(t_2-t_n)}$$

→ we rewrite polynomial as

$$P_{n-1}(t) = \sum_{i=1}^{n} x_i l_i(t)$$

▶ Newton Interpolation

→ we write $p_{n-1}(t)$ as $\sum_{i=1}^{n} x_i \pi_i(t)$

where $\pi_i(t) = \prod_{j=1}^{i-1} (t-t_j)$

↓ newtonian basis

$\pi_1(t) = 1$
$\pi_2(t) = (t-t_1)$
$\pi_3(t) = (t-t_1)(t-t_2)$
$\vdots$
$\pi_n(t) = (t-t_1)(t-t_2)\dots(t-t_{n-1})$

→ horner's rules → $p_{n-1}(t) = x_1 + (t-t_1)(x_2 + (t-t_2)(x_3 + \dots x_n(t-t_{n-1})\dots)))$

ex. $5(t+1)(t-1) - 6(t+1)(t-1)(t-2)(t-3)$

$= (t+1)(t-1)(5 - 6(t-2)(t-3))$

• interpolating complex function to simpler polynomial function

→ Sample n points $(t_i, f(t_i))$

• error bound.

$$\underbrace{\max_{t \in [t_1, t_n]} |f(t) - p_{n-1}(t)|}_{\text{error from interpolating}} \le \boxed{\frac{M h^n}{4n}}$$

n → sample how many n points.

h → maximum gap between t.

$h = \max\{t_2 - t_1, t_3 - t_2, t_4 - t_3, \dots, t_n - t_{n-1}\}$

M → $|f^{(n)}(t)| \le M$

↳ derivatives at n order.

first order $f'(x)$
second order $f''(x)$
$\vdots$
n order $f^{(n)}(x)$

- **Numerical Integration (L8)** $\quad I(f) = \int_a^b f(x)\,dx$

3 rules

① Midpoint Rules → 1 point ⇒ degree 1

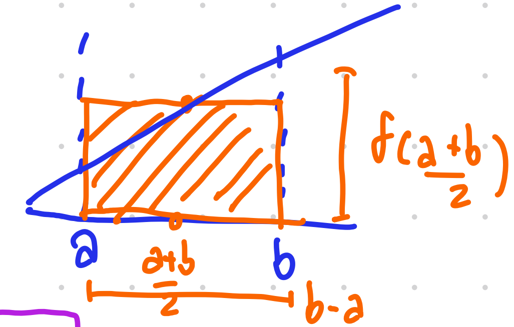
only 1 point → center b/w a & b

$\boxed{n=1}$

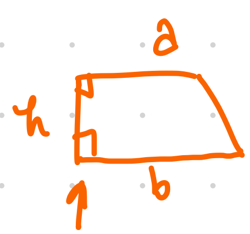
at $\left(\frac{a+b}{2}\right) = x_1$

Polynomial interpolant at $(x_1, f(x_1))$

$P_0(x) = f\left(\frac{a+b}{2}\right)$



$G_1(f) = \boxed{(b-a)\, f\left(\frac{a+b}{2}\right)}$ ⇒ $M(f)$ midpoint rules



→ $\frac{1}{2} \times (a+b) \times h$

Quadrature rule

$Q_n(f) = w_1 f(x_1) + w_2 f(x_2) \dots + w_n f(x_n)$.

* method of undetermined coefficients

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ \vdots & & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} b-a \\ (b^2-a^2)/2 \\ \vdots \\ (b^n-a^n)/n \end{bmatrix}$$

→ solve by GEPP

② Trapezoid Rule → 2 points

$$T(f) = \left(\frac{b-a}{2}\right)(f(a) + f(b))$$

③ Simpson's Rule → 3 points ⇒ degree 3

$$S(f) = \left(\frac{b-a}{6}\right)\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$$

* Midpoint Rule is more accurate than the trapezoid rule & error (twice more accurate)

$$E(f) \approx \frac{T(f) - M(f)}{3}$$

use to approx. the error of integral

* Simpson's rule is much bigger error.

Guassian Quadrature

- for each n, there's a unique n-point Guassian quadrature rule gives the degree of $2n-1$
- highest possible accuracy of n nodes, but difficult to derive.

find w_i's and x_i's so that $\int_a^b dx$, $\int_a^b x\,dx$, ... $\int_a^b x^{2n-1}dx$ computed exactly (2n equations)

↳ will result in nonlinear system of equations

- Weights and nodes derived are specific to the particular interval.
- To integrate over other interval, can transform the new interval into the interval for the rule (say, $[-1, 1]$).

Composite Rules

→ ① Midpoint Composite Rules ($M_k(f)$)

Subdivided. into k equals interval
& use midpoint rules on subinterval

→ ② Trapezoid Composite Rules ($T_k(f)$)

Subdivided. into k equals interval
& use trapezoid rules on subinterval

Numerical Differentiation (L9)

→ function $f(x): \mathbb{R} \to \mathbb{R}$
evaluate. $f'(x)$ for given input x
* Approximate $f'(x)$ for given input x numerically, without computing formula of $f'(x)$

In general
when $n \gg 1$; it's bounded by largest term

$$\Rightarrow O(n^p) = C_p n^p + C_{p-1} n^{p-1} + \dots + C_0$$

when $0 < h \ll 1$; it's bounded by smallest term

$$\Rightarrow O(h^p) = C_p h^p + C_{p+1} h^{p+1} + \dots$$

for small value

- **Forward difference**

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

(has the error of $O(h)$)
first order accuracy

- **Backward Difference**

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

(error is $O(h)$)

- **Centered Difference of $f'(x)$**

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

[second order accuracy] → error of $O(h^2)$

- **Centered difference for $f''(x)$:**

$$f''(x) \approx \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

[second order acc.] error of $O(h^2)$

- another way to derive finite difference formula is to interpolate f & find derivative of interpolant.

$t_i, i = 1, \dots, n \Rightarrow h = t_{i+1} - t_i \Rightarrow$ equally-spare points

let $y_i = f(t_i)$; interpolate 2 points (t_i, y_i) & $(t_{i+1}, y_{i+1}) \rightarrow$ same as forward difference of f'

(t_{i-1}, y_{i-1}) & $(t_i, y_i) \rightarrow$ backward difference of f'

(t_{i-1}, y_{i-1}) & (t_i, y_i) & $(t_{i+1}, y_{i+1}) \rightarrow$ centered difference of f'

To derive higher-order methods or methods for higher derivatives:

1. Add more data points, e.g., (t_{i-2}, y_{i-2}), (t_{i+2}, y_{i+2}), (t_{i-3}, y_{i-3}), (t_{i+3}, y_{i+3}), ...,
2. Find the interpolant, and
3. Take the derivative of the interpolant.

in both integration & differentiation, smaller step size give more accurate result but step can't be too small b/c computation cost & catastrophic cancelation

- Richardson Extrapolation

Suppose $F(h) = a_0 + a_1 h^p + O(h^r)$ as $h \to 0$ for some p & r, w/ $r > p$

thus $a_0 = F(h) + \dfrac{F(h) - F(h/q)}{q^{-p} - 1}$

$F(h)$ value obtained from given method with step h

$\rightarrow p$ is the derivative order

$\rightarrow q$ is some positive integer

$\rightarrow a_0$ is the extrapolated value

(has the error of $O(h^r)$ which smaller than $F(h)$ & $F(h/q)$)