Floating Point Number Systems

$F(B,P,L,U) = \{\pm d_0.d_1d_2\cdots d_{P-1} \times B^E\} \cup$ {denormalized $E=L-1$, special $E=U+1$}

$d_0.d_1d_2\cdots d_{P-1}$ significant (mantissa) $0 \le d_i < B$

P: precision (control accuracy) B: Base eg 10

E: exponent ($L \le E \le U$), control range

rule: $\boxed{d_0 \ne 0}$ (normalize FP numbers make repr unique)

$fl(0) = \pm 0.00\ldots0 \times B^{L-1}$ ← value is denormalized

- Finite set: # of elements $2 \times (B-1) \times B^{P-1}(U-L+1)$ + denormalized & specials

eg $F(10,5,-10,10) \Rightarrow +0.1230 \times 10^0$

IEEE standard

$52 \times \log_{10}^{2} = 16$ digit

	Single	double precision
B	2	2
P	24	53
L	-126	-1022
U	+127	+1023
HUGE	$+1.11\ldots1 \times 2^{127} \doteq 3.4 \times 10^{38}$	$+1.11\ldots1 \times 2^{1023} \doteq 1.8 \times 10^{308}$
TINY	$+1.000.0 \times 2^{-126} \doteq 1.2 \times 10^{-38}$	$+1.00.0 \times 2^{-1022} \doteq 2.2 \times 10^{-308}$

single:

$\boxed{\# \text{ of } E}$ $127 \times 2 = 254$

$\Rightarrow$ need 8 bit ($2^8 = 256$)

$E = L-1 = -127$ denom (0)

$E = U+1 = 128$ (special, if, NaN)

$\boxed{\text{bit}}$

$1 + \overset{23}{24} + 8 = 32$ bit

sign signif exp

$d_0 = 1$ 不用store

23

$1.xx.x \times B^{U+1}$ at least 1 non 0

$x > HUGE$ $fl(x) = +Inf = 1.00.0 \times B^{U+1}$

$+Inf - (+Inf)$ produce not a number

Overflow & underflow

eg $F(10,3,-10,10)$ compute $z = x^2+y^2$, $x = 4.00 \times 10^5$, $y = 3.00 \times 10^5$

$x^2 = 1.6 \times 10^{11} \rightarrow$ overflow

algo: ① $h = \max(|x|,|y|) * \text{sqrt}\left(1 + \left(\frac{\min(|x|,|y|)}{\max(|x|,|y|)}\right)^2\right)$

underflow 那数变成 0

approximate x in F

① Trunction/chopping ② RtN

Absolute Error $|\tilde{x} - x|$ relative error $\frac{|\tilde{x} - x|}{|x|}$

for round to nearest $\frac{|fl(x) - x|}{|x|} \le \frac{1}{2} B^{1-P}$

chopping $\frac{|fl(x) - x|}{|x|} \le B^{1-P}$ $\frac{max}{min} = \frac{0.05 \times 10^E}{1 \times 10^E} = 0.005$

relative round off error bound (ε_{mach})

IEEE
- single $\varepsilon_{mach} = \frac{1}{2} 2^{1-24} = 2^{-24} \doteq 5.96 \times 10^{-8}$ 7 decimal digits $\frac{1}{2} 2^{1-53}$
- Double $\varepsilon_{mach} = 1.1 \times 10^{-16} \approx 16$ decimal digits

Arithmetic Operations can cause round off error

$\varepsilon_{expr1 + expr2} \le \max\{\varepsilon_{expr1}, \varepsilon_{expr2}\} + |\varepsilon_A|$

$\varepsilon_{expr1 * expr2} \le \varepsilon_{expr1} + \varepsilon_{expr2} + |\varepsilon_B|$

$\varepsilon_{expr1 / expr2} \le \underbrace{\varepsilon_{expr1} + \varepsilon_{expr2}}_{\text{error due to inexact operands}} + \underbrace{|\varepsilon_C|}_{\text{error due to representation}}$

$|\varepsilon_A|, |\varepsilon_B|, |\varepsilon_C| \le \varepsilon_{mach}$

catastrophic Cancellation

occure when subtract two possibly inexact quantities close in size => aviod subtracting nearly equal inexact Q

compute roots of $ax^2+bx+c=0$

$r_1 = \frac{-b - \text{sign}(b) \times \sqrt{b^2-4ac}}{2a}$ $r_2 = \frac{c}{ar_1}$ $\text{sign}(b) = \begin{cases} +1, & b \ge 0 \\ -1, & b < 0 \end{cases}$

$\varepsilon_{expr1 - expr2} \le \left|\frac{expr1}{expr1 - expr2}\right| \varepsilon_{expr1} + \left|\frac{expr2}{expr1 - expr2}\right| \varepsilon_{expr2} + |\varepsilon_D|$

where $|\varepsilon_D| \le \varepsilon_{mach}$

eg $f(x) = \sqrt{1+x} - 1$ when $x \to 0$ ε_{expr1} is magnified

$\Rightarrow f(x) = \sqrt{1+x} - 1 \cdot \frac{\sqrt{1+x}+1}{\sqrt{1+x}+1} = \frac{1+x-1}{\sqrt{1+x}+1} = \frac{x}{\sqrt{1+x}+1}$

$\frac{|expr1|}{|expr1 - expr2|}$ denominato ≈ 1

```
algo:
if x ≥ 3
   f = math.sqrt(1+x) - 1
else:
   f = x / math.sqrt(1+x) + 1
```

numerically stable:

An algo is said to be ~ if the result it produces is the exact result of a "nearby" problem.

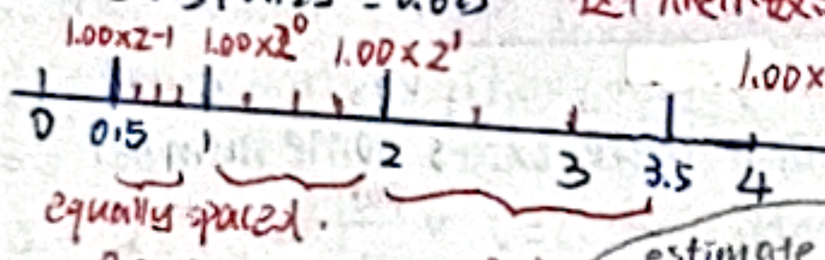
eg 用 r_1, r_2 reconstruct $ax^2+bx+c=0$ $\frac{\Delta a}{a}, \frac{\Delta b}{b}, \frac{\Delta c}{c} \le \varepsilon_{mach}$

change decimal to binary:

$(\frac{2}{3})_{10} = (0.101010\ldots)_2 = 1.0101\ldots \times 2^{-1}$ 一直×2直到0

$(0.101)_2 = 1 \times 2^{-1} + 1 \times 2^{-3} = 0.5 + 0.125 = 0.625$ 这个只是小数点位置

$F(2,3,-1,1)$



estimate $\|A^{-1}\|$ $\max_{\|z\|=1} \|A^{-1}z\|$ $A^{-1}z = y$

$\frac{\text{cond}_2(A)}{n} \le \frac{1}{\text{cond}} \le \text{cond}_2(A)$

When there is cancellation?

$x - y$, and x, y within $(0.1)_B \times B^E$

eg for $B=2$, $x - y \le (0.1)_2 \times 2^E$ ↘ 0.5

cancellation is catastrophic because it repeals the significance / importance of a previously made error.

If x, y are exact, the cancellation is not catastrophic.

mathematical conditioning

⇒ 针对 function

relative error in approximate $f(\bar{x})$ by $f(\hat{x})$ is

$\frac{|f(\hat{x}) - f(\bar{x})|}{|f(\bar{x})|} \doteq \left|\frac{\bar{x} f'(\bar{x})}{f(\bar{x})}\right| |\delta_x|$ $\boxed{\hat{x} = \bar{x}(1+\delta_x) \quad \frac{\hat{x} - \bar{x}}{\bar{x}} = \delta_x}$

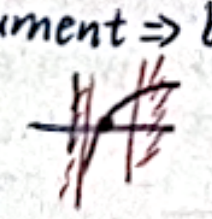
$K_f(x) = \left|\frac{x f'(x)}{f(x)}\right|$ — small: well conditioned; large: ill conditioned. $K_{\sqrt{}}(x) = \frac{1}{2}$

rule of thumb: you loss approx $\approx \log_2 K_f(x)$ bits of accuracy when approx $f(\bar{x})$ by $f(\hat{x})$

numerical stable algo & well conditioned problem ⇒ accurate result

ill conditioned: small Δ in argument ⇒ big Δ result.

eg $y = \ln x$ $K_{\ln}(x) = \left|\frac{x \cdot \frac{1}{x}}{\ln(x)}\right| = \left|\frac{1}{\ln x}\right|$



sol: let $w = x - 1$ $\ln(x) = \ln(w+1)$

$g(w) = \ln(w+1) = \ln(x)$

$K_g(w) = \left|\frac{w \cdot \frac{1}{1+w}}{\ln(1+w)}\right|$ $x \to 1$ $w \to 0$ $K_g(w) \to 1$

$\Rightarrow \ln(1+w)$ is well cond around $w \to 0$

Math.log1p(a) ⇒ $\ln(1+a)$

Evaluate a function:

① mathematical conditioning $K_f(x) = \left|\frac{x f'(x)}{f(x)}\right|$

② algo eg $f(x) = x - \sqrt{x^2-1}$ for $|x| > 1$

$a = x*x$ $\varepsilon_a = \varepsilon_{x*x} = \varepsilon_x + \varepsilon_x + |\varepsilon_a| \le \varepsilon_{mach}$ (assum = 0)

$b = a - 1$ $\varepsilon_b = \varepsilon_{a-1} = \left|\frac{a}{a-1}\right| \varepsilon_a + \left|\frac{1}{a-1}\right| \overset{=0}{\varepsilon_1} + |\varepsilon_b| \le \frac{x^2}{x^2-1} \varepsilon_{mach} + |\varepsilon_b|$

$c = \text{math.sqrt}(b)$ $\varepsilon_c \le \frac{1}{2} \varepsilon_b + |\varepsilon_c|$

$d = x - c$ $\varepsilon_d = \varepsilon_{x-c} \le \left|\frac{x}{x-c}\right| \overset{=0}{\varepsilon_x} + \left|\frac{c}{x-c}\right| \varepsilon_c + |\varepsilon_d|$

$x \to +\infty$ ε_f large $= \left|\frac{1}{\frac{x}{\sqrt{x^2-1}} - 1}\right| \varepsilon_c + |\varepsilon_{IV}|$

so inaccurate result, when f is well conditioned.

```
algo: if x < -1, f = x - math.sqrt(x*x-1)
elif x > 1: f = 1 / (x + math.sqrt(x*x-1))
else: undefined
```

Solving Systems of Linear Equation

- Non singular matrix: $\exists A^{-1}$ s.t. $AA^{-1} = A^{-1}A = I$
- $A\underline{x} = \underline{b}$ has a unique solution iff A is non singular
- A is non singular iff $\det(A) \ne 0$
- A is singular $\Rightarrow \exists z \ne 0$ s.t. $A\underline{z} = 0$

Cramer's rule

$x_i = \det(A_i) / \det(A)$ $A\underline{x} = \underline{b}$ solution:

where $A_i = A$ with ith col $= \underline{b}$

cofactor expansion → n! multiplication

Forward substitution

A is Lower triangular

```
for i=1 to n
    sum = 0
    for j=1 to (i-1)
        sum = sum + a(i,j) * x(j)
    x(i) = (b(i) - sum) / a(i,i)
```

$\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{i-1} 2\text{flops}\right) + 2\right] = n^2 + \theta(n)$

backward substitution

A is upper triangular

```
for i=n to 1 by -1:
    sum = 0
    for j=i+1 to n
        sum = sum + a(i,j) * x(j)
    x(i) = (b(i) - sum) / a(i,i)
```

$\sum_{i=1}^{n}\left[\left(\sum_{j=1}^{i-1} 2\text{flops}\right) + 2\right] = n^2 + \theta(n)$ flops

Gaussian Elimination

$R_j \leftarrow R_j - \frac{a_j}{a_i} R_i$

```
for i=1 to n
    for j=i+1 to (n-1)
        multiplier = a(j,i) / a(i,i)
        for k=i+1 to n
            a(j,k) = a(j,k) - multiplier * (a_i,k)
        b(j) = b(j) - multiplier * b(i)
```

$a_{i,i}$ $a_{i,k}$

$a_{j,i}$ $a_{j,k}$

$\sum_{i=1}^{n-1}\left(\sum_{j=i+1}^{n}\left(1 + \sum_{k=i+1}^{n} 2\text{flops}\right)\right) = \frac{2}{3}n^3 + \theta(n^2)$

Gaussian Transform

$\underset{n \times n}{M_r} = \underset{n \times n}{I} - \underset{n \times 1}{\underline{m}_r} \underset{1 \times n}{\underline{e}_r^T}$, $\underline{m}_r = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{r+1,r} \\ m_{r+2,r} \\ \vdots \\ m_{n,r} \end{bmatrix} \to \frac{a_{r+1,r}}{a_{r,r}}$

$M_r = I - \begin{bmatrix} 0 & & & \\ 0 & & & \\ m_{r+1,r} & 0 & & \\ m_{r+2,r} & & \ddots & \\ m_{n,r} & \cdots & & 0 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ & -m_{r+1,r} & 1 & \\ 0 & -m_{r+2,r} & & \ddots \\ & \vdots & 0 & 1 \end{bmatrix}$ $\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & & & & \\ 0 & m_{r+1,r} & 0 & 0 & \\ 0 & & & & 0 \end{bmatrix}$

- For index $r \le s$, $M_r M_s = I - m_r e_r^T - m_s e_s^T$ (not G.T)
- $M_r^{-1} = I + m_r e_r^T$ (is a G.T)
- For any n-vector $\underline{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$ $M_r \underline{v} = (I - \underline{m}_r e_r^T)\underline{v} = v - v_r \underline{m}_r = \begin{bmatrix} v_1 \\ \vdots \\ v_{r+1} - v_r m_{r+1,r} \\ \vdots \\ v_n - v_r m_{n,r} \end{bmatrix}$

$M_r A$ - 1次 GE

can think of GE as factoring A into LU

Solve $A\underline{x} = \underline{b}$:

① Factor $A = LU$ $\frac{2}{3}n^3 + \theta(n^2)$

$LU\underline{x} = \underline{b}$ let $\underline{y} = U\underline{x}$ ② solve $L\underline{y} = \underline{b}$ Forward sub $n^2 + \theta(n)$

③ solve $U\underline{x} = \underline{y}$ backward sub $n^2 + \theta(n)$

Total: $\frac{2}{3}n^3 + \theta(n^2)$ flops

New prob: $A\underline{z} = \underline{c}$ ① solve $L\underline{y} = \underline{c}$ for. sub

$LU\underbrace{\underline{z}}_{\underline{y}} = \underline{c}$ ② solve $U\underline{z} = \underline{y}$ back. sub

total: $2n^2 + \theta(n)$ flops

permutation matrix

-remember row interchanged

$P_{i,r_i}B$ row i, r_i changed $P_{i,r_i}^{-1} = P_{i,r_i} = P_{i,r_i}^T$

BP_{i,r_i} column i, c_{r_i} changed

$R_i \leftrightarrow R_{r_i}$ where $r_i = i$ where $(i,i) \ne 0$; $r_i > i$ where $(i,i) = 0$

$M_{n-1}P_{n-1,r_{n-1}} \cdots M_2 P_{2,r_2} M_1 P_{1,r_1} A = U$

$\hat{M}_{n-1} \underbrace{P_{n-1} P_{n-2} \cdots P_2 P_1}_{P} A = U$ $P^{-1} = P^T$

$\hat{M}_{n-1} \downarrow L^{-1}$ $\therefore L^{-1}PA = U$

$PA = LU$

Algo:

① find P, L, U, s.t. $PA = LU$

② $A\underline{x} = \underline{b}$ solve $L\underline{y} = P\underline{b}$ for $\underline{y}$

$PA\underline{x} = P\underline{b}$ solve $U\underline{x} = \underline{y}$ for $\underline{x}$

$LU\underline{x} = P\underline{b}$

Pivot vector

eg $Piv = (4,1,3,2)$ $P = \begin{pmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \end{pmatrix}$

indirect address $a(i,j) \to a(Piv(i), j)$

Note: ① $Q_{n \times n}(R_{n \times n} S_{n \times 1})$ $2n \cdot n + 2n \cdot n = 4n^2$

② $\underline{x} = M^{-1}\underline{y} + \underline{z}$ $\begin{cases} \text{let } \underline{c} = M^{-1}\underline{y} & \text{solve } M\underline{c} = \underline{y} \\ \underline{x} = \underline{c} + \underline{z} \end{cases}$

Accuracy

residual vector $\underline{r} = \underline{b} - A\underline{x}$

small residual need not imply on an accurate result

$\hat{\underline{e}} = \hat{\underline{x}} - \underline{x}^*$

$A(\underbrace{\bar{\underline{x}} - \underline{x}^*}_{\bar{\underline{e}}}) = \bar{\underline{b}} - \underline{b} = -\bar{\underline{r}}$ $\bar{\underline{e}} = \bar{\underline{x}} - \underline{x}^* = -A^{-1}\bar{\underline{r}}$

$\|\bar{\underline{e}}\| = \|\bar{\underline{x}} - \underline{x}^*\| = \|-A^{-1}\bar{\underline{r}}\| \le \|A^{-1}\| \|\bar{\underline{r}}\|$

$\|A\underline{x}^*\| = \|\underline{b}\| \Rightarrow \|A\| \|\underline{x}^*\| \ge \|\underline{b}\|$

$\therefore \frac{\|\bar{\underline{x}} - \underline{x}^*\|}{\|\underline{x}^*\|} \le \underbrace{\|A\| \|A^{-1}\|}_{\text{cond}(A) \ge 1} \boxed{\frac{\|\bar{\underline{r}}\|}{\|\underline{b}\|}}$ the norm relative residual

Norm

$\|\underline{x}\| \ge 0$ and $\|x\| = 0$ iff $\underline{x} = 0$

$\|c\underline{x}\| = |c| \|\underline{x}\|$

$\|\underline{x} + \underline{y}\| \le \|\underline{x}\| + \|\underline{y}\|$

$\|\underline{x}\|_1 = |x_1| + |x_2| + \ldots + |x_n|$

$\|\underline{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$

$\|\underline{x}\|_\infty = \max_{1 \le i \le n} |x_i|$

$\|A\| \ge 0$, $\|A\| = 0$ iff $A = 0$

$\|cA\| = |c| \|A\|$

$\|A+B\| \le \|A\| + \|B\|$

$\|AB\| \le \|A\| \|B\|$

induced matrix norm

$\|A\|_1 = \max_{\|x\|_1 = 1} \|A\underline{x}\|_1 = $ max abs col sum

$\|A\|_2 = \max_{\|x\|_2 = 1} \|A\underline{x}\|_2 = \sqrt{\lambda_{max}(A^TA)}$

$\|A\|_\infty = \max_{\|x\|_\infty = 1} \|A\underline{x}\|_\infty = $ max abs row sum $= \max_{1 \le i \le n}\left(\sum_{j=1}^{n} |a_{i,j}|\right)$

Forbenium norm $\|A\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{i,j}^2}$

$M^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

We could loss the $\underline{b}$ info because multiplier was so large (the value on the diagnal was so much smaller than other values in the column)

Alternate interpretation: $A = LU$

cond(L), cond(U) large. we replaced a well conditioned problem with two illed one

Partial Pivoting

r_i that $|a_{r_i,i}| \ge |a_{j,i}|$ for $\forall j \in [i,n]$

$\frac{\|\underline{r}\|}{\|\underline{b}\|} \approx C_n \varepsilon_{mach}$

bits of accuracy in computed sol $\approx P - \lceil 1 - \log_B \text{cond}(A) \rceil$

complete pivoting

$PAQ = LU$ $A\underline{x} = \underline{b}$ $L\underline{y} = P\underline{b}$

$PA\underline{x} = P\underline{b}$ $U\underline{z} = \underline{y}$

$PAQQ^T\underline{x} = P\underline{b}$ $\underline{x} = Q\underline{z}$

$A\underline{x} = \sum_{j=1}^{n} x_j \underline{a}_j$

cond is large when A is close to singular (det=0) (A ill conded) linearly dep row/col

Intropolation — Polynomial Interpolants $(t^{i-1} = t \cdot t^{i-1})$

$P_{m-1}(t) = \sum_{i=1}^{m} c_i t^{i-1}$, operation count: 3m flops

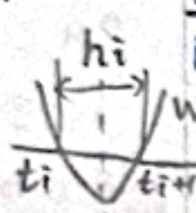
Horner's Rule: opt: 2m-flops

$P_{m-1}(t) = c_1 + t(c_2 + t(c_3 + t(\dots + t(c_{m-1} + t c_m))))$

Generalize: give $S = \{(t_i, y_i)\}_{i=1}^{m}$, find unique

$P_{m-1}(t) = c_1 + c_2 t + c_3 t^2 + \dots + c_m t^{m-1}$ that interplate S

$$\begin{cases} P_{m-1}(t_1) = y_1 \\ P_{m-1}(t_2) = y_2 \\ \vdots \\ P_{m-1}(t_m) = y_m \end{cases} \Rightarrow \begin{cases} c_1 \cdot 1 + c_2 t_1 + c_3 t_1^2 + \dots + c_m t_1^{m-1} = y_1 \\ c_1 \cdot 1 + c_2 t_2 + c_3 t_2^2 + \dots + c_m t_2^{m-1} = y_2 \\ \dots \\ c_1 \cdot 1 + c_2 t_m + c_3 t_m^2 + \dots + c_m t_m^{m-1} = y_m \end{cases} \Rightarrow$$



$$\begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 & \dots & t_1^{m-1} \\ 1 & t_2 & t_2^2 & t_2^3 & \dots & t_2^{m-1} \\ \vdots \\ 1 & t_m & t_m^2 & t_m^3 & \dots & t_m^{m-1} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_m \end{bmatrix}$$

Vandermonde matrix

- t_i is distinct
- V is nonsingular
- solution unique.
- But t=np.linspace(1.0, 2.0, 11) v=np.vander(t)

np.linalg.cond(v) very large, IEEE slove $V\underline{c}=\underline{y}$ 4 dig accuracy

we are using monomial basis $\{1, t, t^2, \dots t^{m-1}\}$

Suppose $\{b_j(t)\}_{j=1}^{m}$ is a basis for P_{m-1}, then $P_{m-1} = \alpha_1 b_1(t) + \alpha_2 b_2(t) + \dots + \alpha_m b_m(t)$ need to solve

want $P_{m-1}(t_i) = y_i$; $B\underline{\alpha} = \underline{y}$

$$\begin{bmatrix} b_1(t_1) & b_2(t_1) & \dots & b_m(t_1) \\ b_1(t_2) & b_2(t_2) & \dots & b_m(t_2) \\ \vdots \\ b_1(t_m) & b_2(t_m) & \dots & b_m(t_m) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

solve this for $\underline{\alpha}$. If $\{b_j(t)\}_{j=1}^{m} = \{1, t, t^2, \dots t^{m}\}$ then $B = V$. Want cond(B)↓, B=I?

Lagrange basis $b_k(t_j) = \begin{cases} 1, & i=j \\ 0, & i \ne j \end{cases} \Rightarrow \alpha_j = y_j$ $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = B$

$b_k(t) = \prod_{j=1, j\ne k}^{m} \frac{(t-t_j)}{(t_k-t_j)}$, $b_k(t)$ labelled $L_k(t)$. $\{b_k(t)\}_{k=1}$ linear Ind. degree<m

$P_{m-1}(t) = \sum_{k=1}^{m} y_k L_k(t)$ 每一项都是 m-1 次方

Issue: evaluate $b_k(t)$ or $L_k(t)$ ~4m flops, there are m of them ⇒ $\Theta(m^2)$ flops

Newton basis let B be lower triangle matrix $\begin{bmatrix} 1 & 0 & & \\ 1 & t_2-t_1 & 0 & \\ 1 & t_3-t_1 & (t_3-t_1)(t_3-t_2) & 0 \\ 1 & t_4-t_1 & (t_4-t_1)(t_4-t_2) & \dots \end{bmatrix}$

$b_k(t) = \prod_{j=1}^{k-1}(t-t_j)$ ⇒ degree k-1. since m polys of unique degree<m, they are linear indep. so form a basis P_{m-1}

$P_{m-1} = c_1 + c_2(t-t_1) + c_3(t-t_1)(t-t_2) + \dots + c_m(t-t_1)(t-t_2)\dots(t-t_{m-1})$

only 2new multiplies per term so total cost to evaluate 3m flops

coefficient $c_i = \left[y_i - \sum_{j=1}^{i-1} c_j \prod_{k=1}^{j-1}(t_i - t_k)\right] / \prod_{k=1}^{i-1}(t_i - t_k)$

$c_1 = y_1$, $c_2 = \frac{y_2 - c_1}{t_2 - t_1}$, $c_3 = \frac{y_3 - [c_1 + c_2(t_3 - t_1)]}{(t_3 - t_1)(t_2 - t_1)}$

summary

	easy to evalu P_{m-1}	easy to $\int, \partial$	find coeff c	add t_{m+1}, y_{m+1} update interplant
M	easy 3m	easy	canbe ill-cond	all coefficient
L	not easy m^2	not easy	super easy (y_i)	add $\frac{t - t_{m+1}}{t_i - t_{m+1}}$ to all & 1 new
N	easy 3m	med easy	easy	add $c_{m+1}(t-t_1)\dots(t-t_m)$

Theorem suppose $P_{m-1}(t)$ interpolate the function $f(t)$ at distinct points $t_1 < t_2 < \dots < t_m$. If $f(t)$ is differentiable on the interval $[t_1, t_m]$. Then for any $t \in [t_1, t_m]$, $f(t) - P_{m-1}(t) = \frac{1}{m!} f^{(m)}(c_t) \cdot W_m(t)$

(c_t is some point in (t_1, t_m) that depends on t, $W_m(t) = \prod_{i=1}^{m}(t - t_i)$)

$\max_{t \in [t_1, t_m]} |f(t) - P_{m-1}(t)| \le \frac{1}{m!} \max_{t \in [t_1, t_m]} |f^{(m)}(t)| \cdot \max_{t \in [t_1, t_m]} |W_m(t)|$

$\Rightarrow \max_{t \in [t_1, t_m]} |f(t) - P_{m-1}(t)| \le \frac{1}{m!} \max_{t \in [t_1, t_m]} |f^{(m)}(t)| \cdot \frac{h^m (m-1)!}{4} = \frac{M h^m}{4m}$

$h = \max_{i=1,\dots,m-1} |t_{i+1} - t_i|$ biggest diff between your interpolation.

Note: Degree m-1 polynomial interpolants do not necessarily converge to the function that they interpolate as m (the # of interpolation points) increase!

Piecewise polynomial Interpolation

construct a set of low degreed polynomial $\{P_{d,i}(t)\}$ (degree, interval #) where each $P_{d,i}(t)$ interpolates a subset of data.

Piecewise linear $\{P_{1,i}(t)\}_{i=1}^{m-1}$, $t_1 < t_2 < \dots < t_m$, $\{t_i\}_{i=1}^{m}$ - knots

Define interpolant $L(t) = \{P_{1,i}(t)\}$, when $t \in [t_i, t_{i+1}]$

Lagrange $P_{1,i}(t) = y_i \frac{t - t_{i+1}}{t_i - t_{i+1}} + y_{i+1} \frac{t - t_i}{t_{i+1} - t_i}$

Newton $P_{1,i}(t) = y_i + \frac{y_{i+1} - y_i}{t_{i+1} - t_i}(t - t_i)$, $i = 1, 2, \dots m-1$

$L(t)$ is continuous. each piece is linear. $P_{1,i}(t_{i+1}) = y_{i+1} = P_{1,i+1}(t_{i+1})$

$L'(t)$ is not continuous. $P'_{1,i}(t) = \frac{y_{i+1} - y_i}{t_{i+1} - t_i}$, $P'_{1,i+1}(t) = \frac{y_{i+2} - y_{i+1}}{t_{i+2} - t_{i+1}}$

Accuracy $t \in [t_i, t_{i+1}]$

$P_{1,i}(t) - f(t) = \frac{1}{2!} f''(c_t) W_2(t)$, $W_2(t) = (t - t_i)(t - t_{i+1})$. $c_t \in [t_i, t_{i+1}]$

$\therefore \max_{t \in [t_i, t_{i+1}]} |W_2(t)| = |\frac{h_i}{2} (-\frac{h_i}{2})| = \frac{h_i^2}{4}$ where $h_i = t_{i+1} - t_i$

$\therefore$ for $t \in [t_1, t_m]$, $|f(t) - L(t)| \le \frac{1}{2!} \cdot \frac{h^2}{4} \cdot \max_{c \in [t_1, t_m]} |f''(c)| = \frac{h^2}{8} \max_{c \in [t_1, t_m]} |f''(c)|$

where $h = \max_i h_i$ $\therefore$ If if you add more points in the interior, h can be made smaller, (derivative 不变) then the approximation error bound will become smaller (eg uniform space $h \to \frac{1}{2}h$, error bound $h^2 \to \frac{1}{4}h^2$) Not smooth

Piecewise Cubic Interpolation (Hermit)

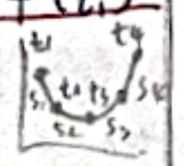
on each $[t_i, t_{i+1}]$, fit a cubic $P_{3,i}(t)$, where $t \in [t_i, t_{i+1}]$

on each interval, $P_{3,i}(t) = \alpha_i + \beta_i(t - t_i) + \gamma_i(t - t_i)^2 + \delta_i(t - t_i)^3$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & h_i & h_i^2 & h_i^3 \\ 0 & 1 & 2h_i & 3h_i^2 \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \\ \gamma_i \\ \delta_i \end{bmatrix} = \begin{bmatrix} f(t_i) \\ f'(t_i) \\ f(t_{i+1}) \\ f'(t_{i+1}) \end{bmatrix}$$

- for each interval $[t_i, t_{i+1}]$
- smoother than linear
- require knowledge of $f'(t_i)$ $t \in [t_i, t_{i+1}]$

Cubic Spline Interpolation ↓ 4(m-1) unknown



$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3$, $i = 1, 2, \dots, m-1$

① S_i to interpolate data: $S_i(t_i) = y_i$, $S_i(t_{i+1}) = y_{i+1}$, $i = 1, 2, \dots m-1$ 2(m-1)

② $S(t)$ be smooth (differentiable): $S'_i(t_{i+1}) = S'_{i+1}(t_{i+1})$ $i = 1, 2, \dots m-2$ m-2

③ $S(t)$ be twice differentiable $S''_i(t_{i+1}) = S''_{i+1}(t_{i+1})$, $i = 1, 2, \dots m-2$ m-2

④ (a) $S''_1(t_1) = 0$, $S''_{m-1}(t_m) = 0$ $\boxed{\text{natural spline}}$ 2

(b) If y' is known at the end point $S'_1(t_1) = y'_1$, $S'_{m-1}(t_m) = y'_m$ $\boxed{\text{clamped spline}}$ [or set=0]

(c) $S'''_1(t_2) = S'''_2(t_2)$ $S'''_{m-2}(t_{m-1}) = S'''_{m-1}(t_{m-1})$

$\boxed{\text{not a knot spline}}$ bc it forces $S_1(t) = S_2(t)$, $S_{m-2}(t) = S_{m-1}(t)$ essentially eliminate t_2 and t_{m-1} as knot.

Accuracy Theorem: let $f \in C^4[a,b]$. If $f(t)$ is a natural, clamped or not-a-knot cubic spline that interpolate $f(t)$ at $a = t_1 < t_2 < \dots < t_m = b$ then $\max_{t \in [a,b]} |S(t) - f(t)| \le C_{method} h^4 \cdot \max_{t \in [a,b]} |f^{(4)}(t)|$

$C_{method} = 5/384$ for clamped, n-a-k, $h = \max_{i=1..m-1} h_i$, where $h_i = |t_{i+1} - t_i|$ indep.

Suppose equal spaced, $h_i = \frac{b-a}{m-1} \to \frac{b-a}{2(m-1)}$ & ↓

error bound reduce by a factor of $(\frac{1}{2})^4 = \frac{1}{16}$

Also, for derivative approximation

$\max_{t \in [a,b]} |f^{(p)}(t) - S^{(p)}(t)| \le C_{method} h^{4-p} \max_{t \in [a,b]} |f^{(4)}(t)|$ p=0,1,2

Compute a cubic spline

Add fictitious piece $S_m(t)$ at the end.

$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3$ $i = 1, \dots, m-1$

$S'_i(t) = b_i + 2c_i(t - t_i) + 3d_i(t - t_i)^2$ & ① ⇒ $a_i = y_i$

$S''_i(t) = 2c_i + 6d_i(t - t_i)$ & ③ ⇒ $y_{i+1} = y_i + b_i h_i + c_i h_i^2 + d_i h_i^3$ ⑤

$S'''_i(t) = 6d_i$ ③ & ⑥ ⇒ $d_i = (c_{i+1} - c_i)/3h_i$ ⑥

⑤ ⇒ $b_i = \frac{y_{i+1} - y_i}{h_i} - \frac{h_i}{3}(2c_i + c_{i+1})$ ⑦

② ⇒ $b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1}$ 把⑥代入 把⑦代入: $b_i + h_i(c_i + c_{i+1}) = b_{i+1}$

$h_i c_i + 2(h_i + h_{i+1}) c_{i+1} + h_{i+1} c_{i+2} = 3\left(\frac{y_{i+2} - y_{i+1}}{h_{i+1}} - \frac{y_{i+1} - y_i}{h_i}\right)$ ⑧ for $i = 1, \dots, m-2$

$$M = \begin{bmatrix} h_1 & 2(h_1+h_2) & h_2 & 0 & & \\ & h_2 & 2(h_2+h_3) & h_3 & & \\ & & h_3 & 2(h_3+h_4) & h_4 & \\ & 0 & & \ddots & & \\ & & & h_{m-2} & 2(h_{m-2}+h_{m-1}) & h_{m-1} \end{bmatrix}_{(m-2) \times m} \quad M\underline{c} = \underline{r}$$

$$\underline{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}_{m \times 1} \quad \underline{r} = 3 \begin{bmatrix} \frac{y_3 - y_2}{h_2} - \frac{y_2 - y_1}{h_1} \\ \vdots \\ \frac{y_m - y_{m-1}}{h_{m-1}} - \frac{y_{m-1} - y_{m-2}}{h_{m-2}} \end{bmatrix}_{(m-2) \times 1}$$

natural cubic splines $S''_1(t_1) = 0 \Rightarrow c_1 = 0$, $S''_m(t_m) = 0 \Rightarrow c_m = 0$

$(m-2) \times (m-2)$ symmetric tridiagnal to solve ⇒ $\theta(m)$ time

remove col 1, m from M; or add 1 at (1,1), (m,m) in M, 0 at 1, m in r

clamped $S'(t_1) = y'_1 \Rightarrow b_1 = y'_1$; ⑦: $h_1(2c_1 + c_2) = 3\left[\frac{y_2 - y_1}{h_1} - y'_1\right]$

$S'(t_m) = y'_m \Rightarrow b_{m-1} + 2c_{m-1}h_{m-1} + 3d_{m-1}h_{m-1}^2 = y'_m$ ②

M 加上 $\left.\begin{array}{l} 2h_1 c_1 + h_1 c_2 = \frac{3}{h_1}(y_2 - y_1) - 3f'(t_1) \\ h_{m-1} c_{m-1} + 2h_{m-1} c_m = 3f'(t_m) - \frac{3}{h_{m-1}}(y_m - y_{m-1}) \end{array}\right]$

not-a-knot $S'''_1(t_2) = S'''_2(t_2) \Rightarrow d_1 = d_2$ | $S'''_{m-2}(t_{m-1}) = S'''_{m-1}(t_{m-1})$

⑥: $h_2 c_1 - (h_1 + h_2) c_2 + h_1 c_3 = 0$ | $h_{m-1} c_{m-2} - (h_{m-2} + h_{m-1}) c_{m-1} + h_{m-2} c_m = 0$

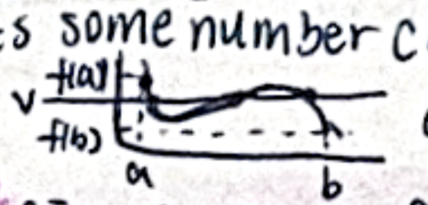
overall algo:
- setup linear system to solve for $\underline{c}$
- use c to determine $\underline{b}$ (⑦), $\underline{d}$ (⑥)
- set $\underline{a} = \underline{y}$

Non linear Equations

Determine the root of a general equation of the form $f(x) = 0$, find x^* such that $f(x^*) = 0$

IVT Intermediate Value

If $f \in [a,b]$ and if v is any number between $f(a), f(b)$. Then there exists some number $c \in [a,b]$ for which $f(c) = v$



C not necessary unique.

Bisection method given continuous $f(x)$, L, R s.t $L < R$, $f(L) f(R) < 0$

```
while ½(R-L) > ABSOL:
    M = L+(R-L)/2
    if sign(f(L)) == sign(f(M)):
        L = M
    else:
        R = M
x* = L+(R-L)/2
```

error in approximating x^* by m_i is $e_i \le \frac{1}{2}|R_i - L_i|$

How quickly converge

After finding n interval, we have $e_n = |m_n - x^*| \le \frac{1}{2^n}|R_1 - L_1|$

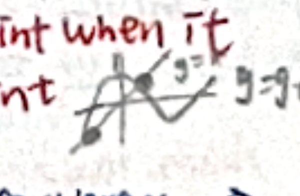
Good: • given initial bracket of the root, method guarantee to converge • only require knowledge of f(x) • can predict in advance how # iteration required to meet error tolerance. • easy to implement.

Bad • requires starting with a bracket of root, which may be difficult to some roots. easy ← diff • converges slowly and not uniformly to root (next itera may be further far away, f(m) bounces around). • does not use all available info about f(x).

Fixed Point Iteration / functional iteration $g(x) = x - f(x)$

$f(x^*) = 0$ is equivalent to find p^*, $g(p^*) = p^*$ where

A point p^* is called a fixed point of a function g(x) if and only if $g(p^*) = p^*$ we see that fixed point of g(x) are roots of f(x) ⇒ g(x) has fixed point when it intersects of y=x. no intersect ⇒ no fixed point

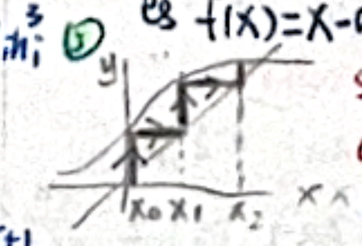


Algo Define the iteration sequence,

$\begin{cases} x_0 \text{ given (guess)} \\ x_{i+1} = g(x_i) \text{ for } i = 0, 1, 2 \dots \end{cases}$ until $|x_{i+1} - x_i| < e$

If it converges → it converge to a fixed point of g(x)

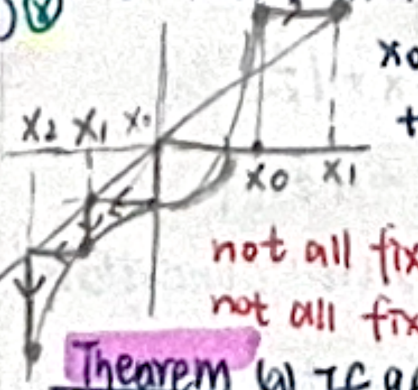
eg $f(x) = x - 0.2\sin x - 0.5 \Rightarrow g(x) = 0.2\sin x + 0.5$ $\because g(x) = x - f(x)$



steps to convergence not depend on accuracy of initial guess.

← 水平找与 y=x 交点, ⇒ 落到x轴新 x_{i+1}

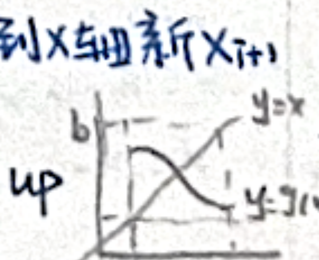
eg $f(x) = x^3 - x - 1 \Rightarrow g(x) = f(x) - x = x^3 - 1$



$x_0 = 0$ or $x_0 = 2$ ⇒ blow up

try $x^3 = x + 1$ & $x_0 = 1$

$x = (x+1)^{1/3}$ $g(x) = (x+1)^{\frac{1}{3}}$



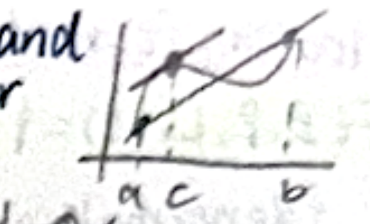
not all fixed points iter solve the problem

not all fixed points can be found by fixed point iter

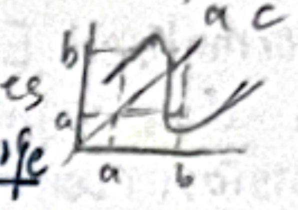
Theorem (a) If $g \in C[a,b]$ with $a < b$, and $\forall x \in [a,b]$, $g(x) \in [a,b]$ then g(x) has a fixed point in [a,b]

(b) if, in addition, $g'(x)$ exists on (a,b) and $|g'(x)| \le k < 1$ for some const k and all $x \in (a,b)$, then fixed point in [a,b] is unique.

Mean Value theorem (MVT) if $f \in [a,b]$ and f is differentiable on (a,b), then a number $c \in (a,b)$ exists with $f'(c) = \frac{f(b) - f(a)}{b - a}$



Note: theorem is sufficient not necessary (可以不满足 满足) eg



when will the fixed point iter converge

It relates to the size of $g'(x^*)$

$x_{i+1} = g(x_i) = g(x^* + h) = g(x^*) + h g'(x^*) + \frac{h^2}{2!} g''(x^*) + \dots \approx g(x^*) + h g'(x^*)$

so $x_{i+1} - g(x^*) = h g'(x^*)$ or $x_{i+1} - x^* = h g'(x^*)$

we know $|x_i - x^*| = h$ so $\underbrace{|x_{i+1} - x^*|}_{\text{error in new approx}} \doteq \underbrace{|x_i - x^*|}_{\text{error in curr approx}} |g'(x^*)|$

attracting fixed point if $|g'(x^*)| < 1$, the fp Iter will converge

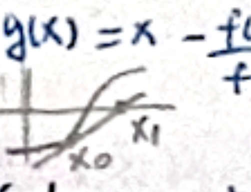
repelling fixed point: if $|g'(x^*)| > 1$, the fp Itera won't converge

Fixed Point theorem let $g \in C[a,b]$ be such that $g(x) \in [a,b]$ for all $x \in [a,b]$. Suppose in addition, g is differentiable on (a,b) and $|g'(x)| \le k < 1$, for some constant k and all $x \in (a,b)$. Then for any $x_0 \in [a,b]$, the fixed point iteration $x_{i+1} = g(x_i)$, $i = 0, 1, 2, \dots$ converges to the fixed point $x^* \in [a,b]$. eg. ① g differentiable & continuous on [-1,1] ② $\forall x \in [-1,1]$, $g(x) \in [-1,1]$ ③ $|g'(x)| \le 0.2 < 1$ for $\forall x \in [-1,1]$ $\therefore x_{i+1} = g(x_i)$ converge for any $x_0 \in [-1,1]$.

Newton's method

it is a fixed point iteration with $g(x) = x - \frac{f(x)}{f'(x)}$

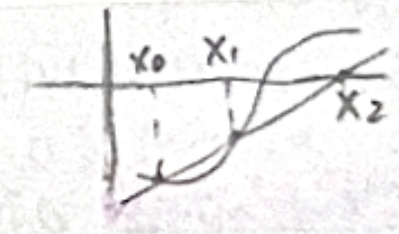
$\begin{cases} x_0 \text{ given, initial guess} \\ x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, i = 0, 1, 2, \dots \end{cases}$



- quadratic convergence (Δx decrease by square factor.
- use f(x), f'(x) ↗ error prone, diff to find when complex f(x)

Secant method

$\begin{cases} x_0, x_1 \text{ given} \\ x_{i+1} = x_i - \frac{f(x_i)(x - x_{i-1})}{f(x_i) - f(x_{i-1})} \end{cases}$



- require one new ~~fxx~~ function per iteration
- ~~require~~ less work per iter than Newton ($f(x_i)$, $f'(x_i)$)
- secant can convergence quickly. (though newton Δx smaller)

rate of convergence Suppose we generate a sequence $x_0, x_1 \dots$ that converge to x^*, let $e_i = |x_i - x^*|$ be the abs error in term x_i. The rate of convergence r of the sequence is the largest value r such that $\lim_{x_i \to x^* (e_i \to 0)} \frac{e_{i+1}}{(e_i)^r} = C \ne 0$ where C is some ≠0 constant.

① **Bisection** $e_{i+1} = \frac{1}{2} e_i$

$\lim_{x_i \to x^*} \frac{e_{i+1}}{e_i} = \frac{1}{2}$ and $r = 1$ ⇒ linear convergence ($r=1$) (when $r > 1$ ⇒ super linear)

② **Newton**: $0 = f(x^*) = f(x_i) + f'(x_i)(x^* - x_i) + \frac{1}{2} f''(\theta)(x^* - x_i)^2$

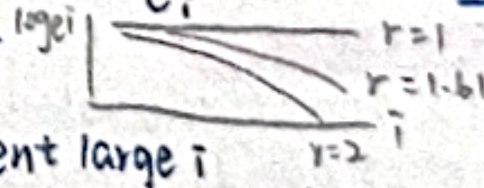
$\Rightarrow x^* - x_{i+1} = \frac{-f''(\theta)}{2f'(x_i)}(x^* - x_i)^2 \Rightarrow e_{i+1} = -\frac{f''(\theta)}{2f'(x_i)} e_i^2$ ($r=2$)

$\lim_{x \to x^*} \frac{e_{i+1}}{e_i^2} = \left|\frac{f''(x^*)}{2f'(x^*)}\right| = \text{const}$ ⇒ quadratic convergence (if $f'(x^*) = 0$, mult root = linear)

③ **Secant** when ② ⑤ converge

$\lim_{x_i \to x^*} \frac{e_{i+1}}{e_i e_{i-1}} = C \ne 0 \Rightarrow \lim_{x_i \to x^*} \frac{e_{i+1}}{e_i^r} = C$, where $\boxed{r = \frac{1+\sqrt{5}}{2} \doteq 1.618}$

interpret $e_{i+1} = C e_i^r$



} converge to root much fast

From data for sufficient large i

$r \doteq \log(e_{i+1}/e_i) / \log(e_i/e_{i-1})$ estimate $e_i \doteq \Delta x = |x_i - x_{i-1}|$

$r_{est} = \log(\Delta x_{i+1}/\Delta x_i) / \log(\Delta x_i/\Delta x_{i-1})$ $C_{est} = \Delta x_{i+1} / \Delta x_i^{r_{est}}$

(suffer from lack of precision as i increases)

Newton • does not always converge • convergence/divergence depends very much on initial guess. • when $f'(x) = 0$ convergence slow, multiple root • convergence when $\left|\frac{f(x) f''(x)}{(f'(x))^2}\right| < 1$ in neighborhood of root / use ① on [b, c] when outside, use ⑪