Wellcome to MA580-001
CSC.
Numerical analysis and Scientific computing.

Dr. Lee.

An example. Data fitting and Interpolation.

t	t_0	t_1	----	t_m
$y(t)$	y_0	y_1		y_m

$m+1$ data

Find $y(t)$ at any time.

One approach: (polynomial fitting)

$$y(t) = a_0 + a_1 t + \cdots + a_n t^n$$

Want to find $\{a_i\}_{i=0}^{n}$

Hopefully

$$y(t_i) = y_i, \quad i = 0, 1, \cdots, m$$

other trig func.

$\sin nt$
$\cos nt$

$\sum a_n \cos + b_n \sin)$

$y(t) \approx a_0 e^{b_0 t}$

Padé

$t=t_0$: $y_0 = a_0 + a_1 t_0 + a_2 t_0^2 + \cdots + a_n t_0^n$

$t=t_1$: $y_1 = a_0 + a_1 t_1 + a_2 t_1^2 + \cdots + a_n t_1^n$

$Ax=b$

$t=t_i$ $\quad y_i = \sum_{j=0}^{n} a_j t_i^j, \qquad i=0,1,\ldots,m$

Matrix-vector form

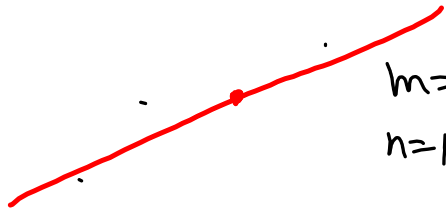
$A\underline{x}=b$

$$\begin{bmatrix} 1 & t_0 & t_0^2 & \cdots & t_0^n \\ 1 & t_1 & t_1^2 & \cdots & t_1^n \\ & & & & \\ 1 & t_m & t_m^2 & \cdots & t_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

$(n+1)\times(n+1)$

$n=0, \quad y(t)=a_0$

$n=1, \quad y(t)=a_0+a_1 t$



$m=4$

$n=1$

$n<m$, 'big data'; simple solution.

Q. Is there a classical solution to the problem? Can we find x^* such that $Ax^*-b=0$ In general, the answer is 'No'. There are too many constraints.

It's called an **over-determined problem.**

We will find the 'best' solution which is least squares solution.

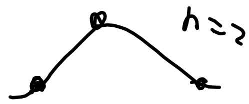
$n=0, \quad y(t)\approx a_0 = \left(\sum_{j=0}^{m} y_j\right)/m+1$ The average?

$n=m$. $A \in R^{(n+1)(n+1)}$, $b \in R^{n+1}$. $Ax=b$.

If $t_i \neq t_j$ if $i \neq j$, distinct sample points.

$\det(A) \neq 0$, $\det(A) = \underline{C\prod_{i\neq j}(t_i - t_j)} \neq 0$

A^{-1} exist. There is a unique solution x^* such that $\underline{Ax^* = b}$



'Interpolation'

$\underline{n > m}$ 'under-determined' $n = 3$

Not enough information to determine the $\{a_i\}$'s uniquely

$$\begin{cases} m = 2 \\ n = 4 \end{cases} \qquad \begin{cases} a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 = y_0 \\ a_0 + a_1 t_1 + a_2 t_1^2 + a_3 t_1^3 + a_4 t_1^4 = y_1 \\ a_0 + a_1 t_2 + a_2 t_2^2 + a_3 t_2^3 + a_4 t_2^4 = y_2 \end{cases}$$

5 unknowns,
3 equations,

$$\begin{bmatrix} \rule{3cm}{0.4pt} \\ \rule{3cm}{0.4pt} \\ \rule{3cm}{0.4pt} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

$\underline{\text{SVD solution}}$ There are infinite number of x^*, x^*_{SVD}: $\min\limits_{Ax^* = b} \|x^*\|_2$, minimizes the two-norm of all possible solution

Introduction: Continue.

What are our tasks in MA580?

A math model: Data fitting & Interpolation

$Ax = b$, $A \in \mathbb{R}^{m \times n}$ $(m+1)(n+1)$

- Wellposedness, existence/uniqueness/sensitivity Linear Algebra
- Numerical Analysis
  Derive numerical methods using $\pm$, $\times/\div$ and logic operation, include some packages
  convergence/complexity
- Scientific computing
  coding/debugging/validation
  including round-off errors
  Use packages, $x = A\backslash b$; in Matlab $m \geq n$
  $x = pinv(A)*b$; $m<n$, $m=n$, $m>n$
  More expensive

$\mathbb{R}$
$\mathbb{C}$ $\Longrightarrow$ Computer number systems: Floating point number system.

$x \in \mathbb{R} \longrightarrow fl(x)$ $\quad \pi \neq fl(\pi)$

$$fl(x) = \underset{\text{sign}}{\pm} . \underset{\text{fraction}}{d_1 d_2 \cdots d_n} \cdot \underset{\text{base}}{\beta}^{s} \longrightarrow \text{exponential}$$

$0 \leq d_i \leq \beta - 1$

$s_{min} \leq s \leq s_{max}$

$\beta = 2$, binary

$\beta = 8$ octal

$\beta = 10$ decimal

$\beta = 16$ hexadecimal

single precision

exponential $\quad 2^3 = 8$

sign	exponential	
1	8	23

$$\begin{array}{r} 23 \\ 8 \\ 1 \\ \hline 32 \text{ bits} \end{array}$$

$2^{-24} \simeq \underline{\underline{6 \times 10^{-8}}}$ $\quad 2^7 = 128$

64-bit

1	11	52

$$\begin{array}{r} 52 \\ 11 \\ 1 \\ \hline 64 \text{ bits.} \end{array}$$

$2^{-52} \simeq \underline{\underline{10^{-16}}}$

Some examples of floating numbers

$-.1011 \cdot 2^5$ binary

$-.31416 \times 10^1$, ✓ $-.031416 \times 10^2$

Not unique! We should use the normalized floating numbers. $x \neq 0$, $fl(x) = sign(x).\, d_1 d_2 \ldots d_n \cdot \beta^s$

$d_1 \neq 0$. In matlab $\tilde{x} = sign(x) \cdot d_1 . d_2 \ldots$ $d_1 \neq 0$

0.00001... format short e

long e

$0 = 0.00 \ldots 0 \beta^0$

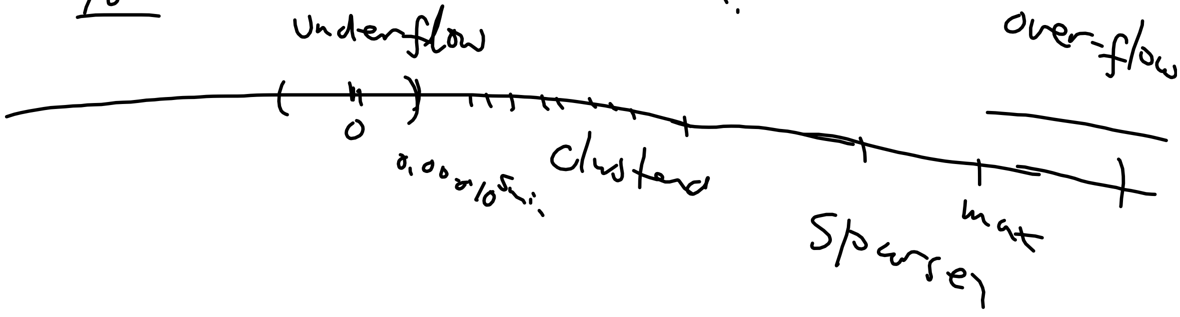
Properties of a computer number system.

- Finite numbers

Total $= 2(\beta - 1)\beta^{n-1}(s_{max} - s_{min} + 1) + 1$ ✓

- has maximum and minimum.

10^{138}



$\underline{fl(x)} \circ fl(y)$ may be out of computer system.

$\circ \quad + \quad - \quad \times \quad \div$

$\underline{10^{134}} \cdot 10^{134}$

Usually underflow is not problem. Simply put it as zero.

overflow means something is wrong. $\frac{y}{x}$ if x is not defined: double check!

Inf NaN. $\log(-5)$ 10^{38}

y/x $\underline{10^{138}}$

Most of time, a computer is adequate enough for many practical problem.

What's the best accuracy can we expect?

Single precision, 10^{-8}

double 〃 10^{-16}

Round-off error analysis,

How do we measure errors.

$x \to fl(x)$

$\pi \to fl(\pi)$

Absolute error $x - fl(x)$ true − approximated

Ex: Error estimate for decimal = absolute error

$x = 0.d_1 d_2 \cdots d_n \, d_{n+1} \cdots \cdot 10^b$

Round-off

$$fl(x) = \begin{cases} 0.d_1 d_2 \cdots d_n \cdot 10^b & \text{if } d_{n+1} \le 4 \quad (d_{n+1} < 5) \\ 0.d_1 d_2 \cdots (d_n+1) \cdot 10^b & (d_{n+1} \ge 5) \end{cases}$$

Truncating

rounding $x - fl(x)$ Varing with x

$|x - fl(x)|$ Upper bound

$$\le \begin{cases} 0.0 \cdots 0 \overset{d_n}{5} 0 \cdots 0 \cdot 10^b & \cdots 4999 \cdots \\ 0.0 \cdots 0 5 0 \cdots 0 \cdot 10^b \end{cases}$$

$\frac{1}{2} 10^{-n} \cdot 10^b$ An upper bound