SFU
muraki

MONDAY

- MATLAB WARM-UP DUE TONIGHT
- COMPUTING WORKSHOP TODAY: $2^{30} - 4^{30}$
- quiz solutions posted later this week
- new questions for quiz #2
- 1st COMPUTING REPORT

LAST DAY

- finite-precision, floating point. REAL #s & ARITHMETIC
  - MACM 316 SPRING 19 uses MATLAB CONVENTION (NOT TEXTBOOK)

+/- d1. d2 d3 d4 d5 d6 d7 d8 … d13 d14 d15 d16 * 10^P

- key ideas to know:
  - format: SIGN, MANTISSA, EXPONENT
  - ROUNDING.
  - CARRY & DROPPING.
  - SYNTHETIC N-digit ARITHMETIC
  - LOSING & GAINING RELATIVE ERROR

$$\frac{|x - x_{exact}|}{|x_{exact}|} \to 10^{-16} \text{ OR LARGER}$$

SFU
muraki

SIMPLE STRATEGIES for REDUCING FINITE-PRECISION ERROR

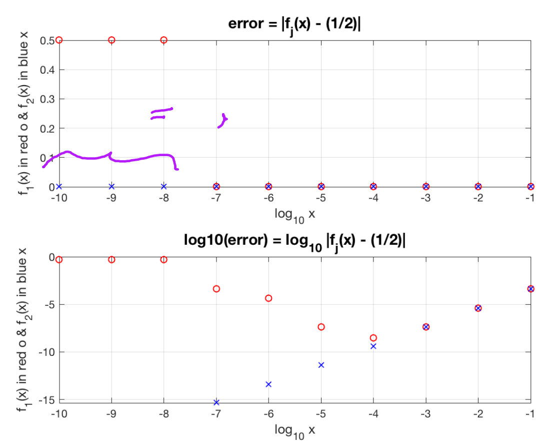
• recall the calculus limit $\lim_{\theta \to 0} \frac{1-\cos\theta}{\theta} = -$

```
>> f1 = @(th) (1-cos(th))./(th.^2)

f1 =

  function_handle with value:

    @(th)(1-cos(th))./(th.^2)
```

```
>> f2 = @(th) (sin(th/2)./(th/2)).^2

f2 =

  function_handle with value:

    @(th)(sin(th/2)./(th/2)).^2
```

1) there is an analytical "fix" for this —

$$\frac{1-\cos\theta}{\theta^2} = \frac{1}{\ }\left(\frac{\theta/}{\theta/}\right)$$



• fu vs fu

• vs (No !)

• vs

• plots !!

• labels + titles !! graphics are communication

• HMWK EXTRA: why is $x = 10^{-4}$ the

2) reducing # of operations?

- how many operations to evaluate polynomial

$$p(x) = \sum_{k=0}^{N} c_k \cdot x^k$$

multiplications per

additions

$$(N + 1 + 2 + 3 \ldots + N) = N + \underline{(\quad)}$$

- NESTED EVALUATION (p24)

$$p(x) = (\cdots((c_N \cdot x + c_{N-1})x + c_{N-2})x + \ldots + c_0)$$

additions + multiplications

3) respecting magnitudes.

ones !

$$s = (1.\ldots)e^{17} + \{1 \qquad \}$$

- SUMMATION (expensive, but)
  - add magnitude quantities first
  - Matlab's command +

```
>> s = 1e17+100

s =

   1.000000000000001e+17

>> s-1e17

ans =

   96
```

SFU
muraki

GAUSSIAN ELIMINATION – the numerical algorithm (s 6.1)

• in this section:

a: GE as row operations
b: exact & numerical failure modes of GE
c: make robust via pivoting operations
d: is GE an algorithm, or a mathematical formula?

· Rows

$$\begin{array}{l} E_1 \to \\ E_2 \to \\ E_3 \to \end{array} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

each corresponds to an E_j, $j=$

• special case: upper triangular form (p363)

$$\begin{array}{l} E_1 \to \\ E_2 \to \\ E_3 \to \end{array} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ & a_{22} & a_{23} \\ & & a_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

• iterative back-substitution for (BSub) x_3, x_2, x_1

$x_3 = \text{—}$, $x_2 = \dfrac{b_{22} - a_{23}}{a_{22}}$ …

• what if $a_{22} =$?

i) · by zero → solution

ii) $= 0$ too !! ⇒ $a_2 =$

SOLUTION NOT

⇒ any $a_{jj} = 0$ (DIAGONAL ELEMENT) means NOT a SOLUTION

• generalizes to $N \times N$

SFU
muraki

ROW REDUCTION ALGORITHM (as seen in tutorial)

- general solution by ROW (RRed)
- 3 ROW o on E_j's (p362)

1) E_j by c
2) add E_j to E
3) interchange $E_i \leftrightarrow E$

ROW REDUCTION ALGORITHM

- AUGMENTED MATRIX notation (examples p363-9)

$$\left[\begin{array}{ccc|c} 0 & 0 & -2 & 12 \\ 2 & 4 & -1 & 7 \\ 2 & 5 & -5 & -1 \end{array}\right]$$

a) $\leftrightarrow$: put non-zero entry in (element, p367)

$$\left[\begin{array}{ccc|c} 2 & 5 & -5 & -1 \\ 2 & 4 & -1 & 7 \\ 0 & 0 & -2 & 12 \end{array}\right]$$

b) $-E_1 + E_2 \to E_2$: form zero in using as PIVOT.

$$\left[\begin{array}{ccc|c} 2 & 5 & -5 & -1 \\ 0 & -1 & 4 & 8 \\ 0 & 0 & -2 & 12 \end{array}\right] \to \text{TRIANGULAR ( ) FORM}$$

SFU
muraki

- in general, final form of row reduction (RRed) is upper Δ form.

$$\left[\begin{array}{ccccc|c} a_{11} & \cdots & \cdots & \cdots & a_{1N} & b_1 \\ & a_{22} & & & \vdots & b_2 \\ & & \ddots & & \vdots & \vdots \\ & \phi & & \ddots & \vdots & \vdots \\ & & & & a_{NN} & b_N \end{array}\right]$$

triangle includes

- step-by-step description of GE process (p365-6)
- complete GE with back-substitution (BSub)

- exact failure of GE: occurs if ANY $= 0$ (p366-7)
  - two distinct failure cases, after RRed

a) $\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 1 & -2 \end{array}\right]$

singular solve: CONSISTENT, solutions

$x_3 = -$

b) $\left[\begin{array}{ccc|c} 1 & 1 & 1 & 4 \\ 0 & 0 & -1 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$

singular solve: CONSISTENT, solutions

$x_3 =$ ANY $x_3 =$ NO!

muraki

NUMERICAL ISSUES IN GE (§6.2)

- row reduction: subtraction of #'s can occur

$$\left[\begin{array}{cccc|c} a_{11} & \cdots & & \cdots & \vdots \\ & \ddots & & & \vdots \\ & & a_{kk} & \cdots & \vdots \\ & & .003 & 59.14 & 59.17 \\ & & 5.291 & -6.130 & 46.78 \end{array}\right]$$

- p 377 does last RRed + BSub in 4-digit arith

$$\begin{array}{l} E_1: \\ E_2: \end{array} \left[\begin{array}{cc|c} .003000 & 59.14 & 59.17 \\ 5.291 & -6.130 & 46.78 \end{array}\right]$$

$$-\left(\underbrace{\frac{5.291}{.003000}}_{1764}\right) E_1 + E_2 \to E_2$$

(attn: do you know how to put 1e-10 into your calculator?)

$$\left[\begin{array}{cc|c} .003000 & 59.14 & 59.17. \\ 0 & 104300. & 104400. \end{array}\right]$$

- BSub gives $x_2 \doteq 1.001$

and $x_1 = \dfrac{59.17 - 59.14(1.001)}{.003} \doteq -10.00$

- Exact answer is $x_1 = +10$, $x_2 = 1$

SFU
muraki

- problem is NEAR-ZERO pivot element (on diagonal during RRed)
  - IDEA: make GE more robust → PARTIAL PIVOTING.

partial pivot rule: exchange remaining rows to use largest magnitude pivot element. (p377)

- row exchange $E_1 \Leftrightarrow E_2$

$$\begin{array}{c} E_1 \rightarrow \\ E_2 \rightarrow \end{array} \left[\begin{array}{cc|c} 5.291 & -6.130 & 46.78 \\ .003 & 59.14 & 59.17 \end{array}\right]$$

- now, $-\left(\frac{.003}{5.291}\right) E_1 + E_2 \rightarrow E_2$

$$\left[\begin{array}{cc|c} 5.291 & -6.130 & 46.78 \\ 0 & 59.14 & 59.17 \end{array}\right]$$

- and BSub

$$x_2 \doteq 1.001$$

$$x_1 \doteq 10.00 \rightarrow \text{correct in 4-digit arith}$$

- there are even more robust versions:

scaled partial pivoting (p379 with example)

complete/full pivoting (p382) → rather costly in CPU time