WEDNESDAY

- quiz on FRI: floating-pt arithmetic & GE.
- computing report #1: GE errors for large matrices.

LAST DAY

- simple STRATEGIES for REDUCING fl-pt error
  1) math formulas matter, some better for computing.
  (avoid subtractions!)

$$\log a - \log b = ?$$

  2) reduce # of operations (fewer truncations)

$$x^{4000} = (4000 * ( ))$$

  3) respect magnitudes in sums (for accuracy)
  + products (avoid OVER/UNDERFLOW)

$$(D_1 \cdot 10 \quad)(D_2 \cdot 10 \quad)(D_3\, 10^{-}\quad)$$

- GAUSSIAN ELIMINATION
  - exact failure mode → PIVOT (some =)
  - numerical failure mode → - · (of #)
  - FIXED by

SFU
muraki

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

```
>> GEdemo

aa =

    -2     3    -2    -4
    -1     1    -1    -1
     5    -1    -6     1
    -4    -5     0    -4


bb =

    -1
     0
    -4
     2

partial pivoting? (y/n):y

Begin Row Reduction with Augmented system:
    -2     3    -2    -4   -1
    -1     1    -1    -1    0
     5    -1    -6     1   -4
    -4    -5     0    -4    2


Swap rows 1 and 3;  new pivot = 5

After Row Reduction in column 1 with pivot = 5.000000
    5.0000   -1.0000   -6.0000    1.0000   -4.0000
         0    0.8000   -2.2000   -0.8000   -0.8000
         0    2.6000   -4.4000   -3.6000   -2.6000
         0   -5.8000   -4.8000   -3.2000   -1.2000


Swap rows 2 and 4;  new pivot = -5.8

After Row Reduction in column 2 with pivot = -5.800000
    5.0000   -1.0000   -6.0000    1.0000   -4.0000
         0   -5.8000   -4.8000   -3.2000   -1.2000
         0         0   -6.5517   -5.0345   -3.1379
         0    0.0000   -2.8621   -1.2414   -0.9655


After Row Reduction in column 3 with pivot = -6.551724
    5.0000   -1.0000   -6.0000    1.0000   -4.0000
         0   -5.8000   -4.8000   -3.2000   -1.2000
         0         0   -6.5517   -5.0345   -3.1379
         0    0.0000         0    0.9579    0.4053


x =

   -0.7308
   -0.1538
    0.1538
    0.4231
```

- Matlab's backslash "\" uses partial pivoting
  → this is sufficient to make GE
- but there is MORE: scaled partial pivoting
  full pivoting (exch rows + cols, p382)

Scaled Partial Pivoting

Scaled partial pivoting (or *scaled-column pivoting*) is needed for the system in the Illustration. It places the element in the pivot position that is largest relative to the entries in its row. The first step in this procedure is to define a scale factor s_i for each row as

$$s_i = \max_{1\le j\le n} |a_{ij}|.$$

If we have $s_i = 0$ for some i, then the system has no unique solution since all entries in the ith row are 0. Assuming that this is not the case, the appropriate row interchange to place zeros in the first column is determined by choosing the least integer p with

$$\frac{|a_{p1}|}{s_p} = \max_{1\le k\le n} \frac{|a_{k1}|}{s_k}$$

and performing $(E_1) \leftrightarrow (E_p)$. The effect of scaling is to ensure that the largest element in each row has a *relative* magnitude of 1 before the comparison for row interchange is performed.

- this issue can be partly addressed by (type 1 fix)
  - if instead of E_2 above we rewrite as
    then solution is same, but PIVOT changes...
  - sometimes happens IRL when wrong are used!

NUMERICAL GE

- 2 phases: ROW REDUCTION & BACK-SUBSTITUTION
- pivot operation: $-(-) + E_k \to E_k$

$$\begin{array}{l} E_j: \\ \\ E_k: \end{array} \quad \left[\begin{array}{cccccc|c} 0 & \cdots & 0 & a_{jj} & \cdots & a_{jN} & b_j \\ \\ 0 & \cdots & 0 & a_{kj} & \cdots & a_{kN} & b_k \end{array}\right]$$

- why is small pivot a_{jj} problematic?
  - linear system is SINGULAR when 2 Rows of the matrix are the (rows)
    - if a_{kj}/a_{jj} really large

      then $-\left(\frac{a_{kj}}{a_{jj}}\right) E_j + E_k \to E_k \approx -( )$

      if large... digits in
  - partial pivoting ensures (sect 6.2) $|-| <$ for all $k>j$

OPERATION COUNTS FOR finite-prec. GE (p369-71)

- total number of $+, -, \times, \div$ in $N \times N$ GE.
-

$$\sum_{j=1}^{m} 1 = m, \quad \sum_{j=1}^{m} j = \frac{m(m+1)}{2}, \quad \text{and} \quad \sum_{j=1}^{m} j^2 = \frac{m(m+1)(2m+1)}{6},$$

$$= \frac{m^2}{2}\left(1+\frac{1}{m}\right) \qquad = \frac{m^3}{3}\left(1+\frac{3}{2m}+\frac{1}{2m^2}\right)$$

$$= O(m^3) \to \text{"O" notation}$$

means $\lim_{m\to\infty} \left(\sum_1^m j^2\right) \Big/ m^3$ exists

• op count for

$$\begin{array}{c} j=N \\ \uparrow \\ j=1 \end{array} \qquad \begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{NN} \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix} - \begin{bmatrix} a_{12} & \cdots & a_{1N} \\ & \ddots & \vdots \\ & & a_{N-1,N} \end{bmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$$

$$\underbrace{\sum_{j=1}^{N}}_{\text{ops for each starting at bottom}} \Big(\underbrace{1}_{\text{div by}} + \overset{*\text{ and }-}{2} \underbrace{(j-1)}_{\#\text{ of used on right-side}} \Big) = \sum_{j=1}^{N} (2j-1) = N^2 \quad (p371)$$

• op count for

$$\underbrace{\sum_{j=1}^{N}}_{\text{ops for each, starting at top}} \underbrace{(N-j)}_{\#\text{ to reduce}} \Big\{ \underbrace{1}_{\text{factor } a_{kj}/a_{jj}} + \overset{*\text{ and }-}{2} \underbrace{(N-j+1)}_{\#\text{ to subtract}} \Big\}$$

WolframAlpha computational knowledge engine

sum (N-j)*(1 + 2*(N-j+1)), j = 1 to N

Examples Random

Sum:

$$\sum_{j=1}^{N} (N-j)\,(2\,(-j+N+1)+1) = \frac{1}{6}\,N\,(4\,N^2+3\,N-7) \quad \sim \quad \left(1 + \frac{3}{4N} - \frac{7}{4N^2}\right)$$

roughly then, matrix size → #ops & compute time

SFU
muraki

• extra for PARTIAL PIVOTING.

$$\underbrace{\sum_{j=1}^{N}}_{\text{top to bot}} \underbrace{(N-j)}_{\text{\# of poss } j^{th} \text{ col}} = -\left(1-\frac{1}{N}\right)$$

very !

• extra OPS for SCALED PART PIV: $+\,O\left(\frac{N^3}{3}\right)$
• " " " FULL PIV : $+\,O\left(\frac{2N^2}{3}\right)$ (p362 + 3)

GE SUMMARY

• → due to finite-prec truncation
  • minimized by (part, scaled, full)

• → op counts are $O(N^3)$
  • only can do better for special matrices (-GE algo)
    (, matrices, etc)
    mostly in matrix

• → pivoting avoids PIVOTS
  • still need to worry if matrix is -s
    OR a " MATRIX"

A TEST FOR BAD GE MATRICES ?

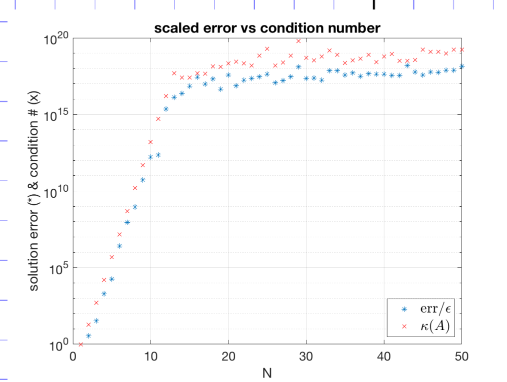
- GOOD MATRIX, discrete Fourier transform, dftmtx(N)

$$[A]_{N\times N} = [a_{jk}] = \left[e^{i2\pi(j-1)(k-1)/N}\right]$$

  - observations:

- BAD MATRIX, Hilbert matrix, hilb(N)

$$[A]_{N\times N} = [a_{jk}] = \left[\frac{1}{j+k-1}\right]$$



- solution error to

$$[A]\vec{x} = [A]\underbrace{(\vec{1})}_{\text{vector of ones}}$$

$$\text{error} = \max_{j=1,N} |x_j - 1|$$

  - observations:

- the mathematical term for a bad matrix for a linear solve is ILL-CONDITIONED. → it is a ___ of the ___ (not a ___ of GE)

SFU

muraki

CONDITION NUMBER of $[A]$

- defined by the eigenvalues of matrix $\underbrace{[A^*]}_{\text{HERMITIAN CONJUGATE}}[A]$.

$$[A] = [a_{jk}] \quad \text{then} \quad [A^*] = [\quad]$$

```
>> A

A =

  1.0000 + 1.0000i   0.0000 + 1.0000i
  2.0000 + 0.0000i   3.0000 + 2.0000i

>> A'

ans =

  1.0000 - 1.0000i   2.0000 + 0.0000i
  0.0000 - 1.0000i   3.0000 - 2.0000i

>> cond(A)

ans =

   6.1623

>> eps =

ans =

  2.2204e-16
```

$$K(A) = \sqrt{\frac{\lambda\;(\quad)}{\lambda\;(\quad)}}$$

- estimate of solution error

$$\text{err} \simeq K(A) \cdot \underbrace{\quad}_{\text{last digit of precision}}$$

"machine epsilon"

muraki