Welcome to Ma221. Lecture3, Aug28

Math problem
structure
accuracy

1) Error Bounds: even if $A$ close to your input, if $A$ close to singular, could still get wrong answe
We will derive “cheap” error bounds
"condition number" = machine epsilon
eg $10^{-16}$ double

$$\text{condition number} = \frac{1}{\text{distance to nearest ill posed problem}}$$

2) “Guaranteed correct” except in “raro cases”: take a few steps of Newton to make error smaller
cheap for $Ax=b$, LS
popular again because of 16-bit floating point accelerators, e.g. Google TPU (now)

3) “Accuracy” = getting bitwise identical answers every time you run code.
   because floating point addition not associative.

Axis 4 - Efficiency

(1) Fewest keystrokes
   A/b
   lots of existing software

(2) What does “fewest fIops” mean?
   for n x n matrix?
   Classical: $2n^3$ fIops
   Strassen (1969) $O(n^{\log_2 7}) = O(n^{2.81})$
   Coppersmith-Winograd (1987) $O(n^{2.376})$
   “galactic algorithm”
   Williams (2013) $O(n^{2.37286412})$
   Le Gall (2014) $O(n^{2.3728439})$
   Williams-Afshin (2020) $O(n^{2.3728594})$
Williams, Xu, Xu, Zhou (2023)
\( O(n^{2.371552}) \)

all standard problems \( Ax = b, L_S, e_{ig} \) can run as fast as \( \text{matmul} \)
\( O(n^w) \) \( \text{matmul} \rightarrow O(n^w) \) for \( Ax=b \)

numerically stable
sometimes practical

will slow how to reorganize \( Ax = b, L_S, e_{ig} \),
to do almost all work in \( \text{matmul} \)

(3) Counting flops not only important (metric)

(3.1) "End of Moore's Law" (2004)

\( \Rightarrow \) parallelism necessary

sometimes need new math alg

(3.2) Arithmetic cheap compared to moving data

naive \( \text{matmul} \) (3 nested loops) can be
100x slower than optimized version
Lots of recent algorithms attaining lower bounds, sometimes same ops in different order, sometimes different math alg.

(3.3) Minimizing Energy
Turns out minimizing data movement
minimizes energy

Communication lower bounds
for O(n³) matmul is \( \Omega \left( \frac{n^3}{VM} \right) \)
for \( O(n^n) \) matmul is \( \Omega \left( \frac{n^n}{M^{n/2-1}} \right) \)

Syllabus,

Direct Methods for \( Ax = b \), LS
\[ LU : A = P \cdot L \cdot U \]
permutation

\[ LS : A = QR \]
orthogonal

Eigenproblems (not Jordan Form)
Schur Form \( A = QRQ^T \)
SVD \[ A = U \Sigma V^T \]

Iterative:
- Jacobi
- Gauss-Seidel
- Conjugate gradients
- Multigrid...

Common idea: Do many matrix-vector multiples, find "best solution" in subspace spanned by vectors

Randomized

Lecture 2: Floating Point Arith.
- Error Analysis
- Beyond basics
  - Exceptions
  - Variable precision
  - Reproducibility
  - Interval Arithmetic

Ex: Polynomial evaluation + zero finding

Recall bisection

Want to solve \( f(x) = 0 \)

Start with interval \([x_1, x_2]\)

where \( f(x_1) \cdot f(x_2) < 0 \)
evaluate \( f(\frac{x_1 + x_2}{2}) \), keep bisecting until interval narrow enough.

(Matlab demo of bisection)