Welcome to Ma221! Lecture 3, Aug 28

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 "rare 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, eg Google TPU (conference now)

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

Axis 4 - Efficiency

(1) Fewest keystrokes
$A \backslash b$
lots of existing software

(2) What does "fewest flops" mean? for $n \times n$ matmul?

Classical: $2n^3$ flops

Strassen (1969) $O(n^{\log_2 7}) = O(n^{2.81})$

Coppersmith-Winograd (1987) $O(n^{2.376})$
"galactic algorithm"

Williams (2013) $O(n^{2.3728642})$
Le Gall (2014) $O(n^{2.3728639})$
Williams + Alman (2020) $O(n^{2.3728596})$

Williams + Xu + Xu + Zhou (2023)

$O(n^{2.37\underline{1552}})$

D., Dumitriu, Holtz (2008):

all standard problems ($Ax=b$, LS, eig) can run as fast as matmul

$O(n^\omega)$ matmul $\rightarrow$ $O(n^\omega)$ for $Ax=b$ etc

numerically stable, sometimes practical

will show how to reorganize $Ax=b$, LS, eig, to do almost all work in matmul

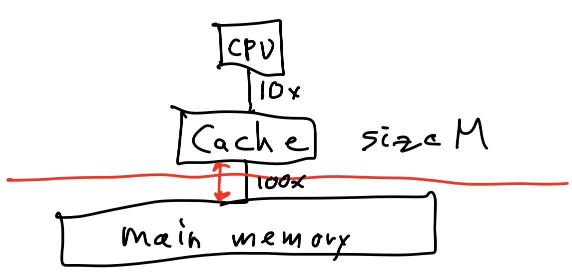
(3) Counting flops not only important metric

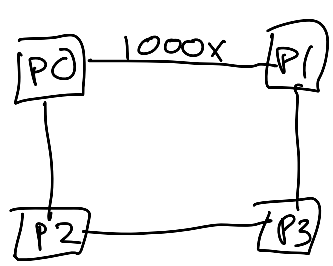
(3.1) "End of Moore's Law" (2004)

$\Rightarrow$ parallelism necessary

sometimes need new math alg

(3.2) Arithmetic cheap compared to moving data





naive 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 $O(n^3)$ matmul is $\Omega\left(\frac{n^3}{\sqrt{M}}\right)$
for $O(n^\omega)$ matmul is $\Omega\left(\frac{n^\omega}{M^{\omega/2-1}}\right)$

Syllabus:

Direct Methods for $Ax=b$, LS

LU: $A = P \cdot L \cdot U$

permutation

LS: $A = Q \cdot R$

orthogonal

Eigenproblems (not Jordan Form)

Schur Form $A = Q R Q^T$

orthogonal $\quad$ eigenvalues on diagonal

$$\text{SVD} \quad A = \underset{\text{orth}}{U}\;\underset{\text{diag}}{\Sigma}\;\underset{\text{orth}}{V^T}$$

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 $f\left(\frac{x_1+x_2}{2}\right)$, keep bisecting until interval narrow enough.

(Matlab demo of bisection)