Welcome to Ma221! Lec 8 Fall24

Recall Induction step of $PA=LU$

$U(1,:)$

$$\begin{array}{cc} & \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{array}{c} 1 \\ m-1 \end{array} & \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \end{array} = \begin{array}{cc} & \begin{array}{cc} 1 & m-1 \end{array} \\ \begin{array}{c} 1 \\ m-1 \end{array} & \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{A_{21}}{A_{11}} & I \end{array}\right] \end{array} \cdot \begin{array}{cc} & \begin{array}{cc} 1 & n-1 \end{array} \\ \begin{array}{c} 1 \\ m-1 \end{array} & \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & \underbrace{A_{22} - \frac{A_{21} \cdot A_{12}}{A_{11}}}_{\text{repeat}} \end{array}\right] \end{array}$$

$L(:,1)$

Express induction steps as code

for $i = 1:n$
    $L(i,i) = 1$, $L(i+1:n,1) = A(i+1:n,1)/A(i,i)$
        ... ignore perm for now
    $U(i,i:n) = A(i,i:n)$
    if $i < n$, $A(i+1:n,i+1:n) = A(i+1:n,i+1:n)$
        $- L(i+1:n,i) \cdot U(i,i+1:n)$

Add permutations
    if $A(i,i) = 0$ and some $A(j,i) \neq 0$ for $j > i$
        swap rows i and j of L and A,
        record in P
    how to choose nonzero $A(j,i)$
        called "pivoting", choices later

Don't waste space: L and U overwrite A

row i of U overwrites row i of A

omit $U(i,i:n) = A(i,i:n)$

col i of L below diagonal overwrites same entries of A, which are available because zeroed out:

change first line to

$$A(i+1:n,i) = A(i+1:n,i)/A(i,i)$$

change last line to

$$A(i+1:n,i+1:n) = A(i+1:n,i+1:n) - A(i+1:n,i) \cdot A(i,i+1:n)$$

Summarize:

for $i = 1$ to $n-1$

if $A(i,i) = 0 \neq A(j,i)$ for $j > i$,

swaps rows i and j

record swap in P

$A(i+1:n,i) = A(i+1:n,i)/A(i,i)$

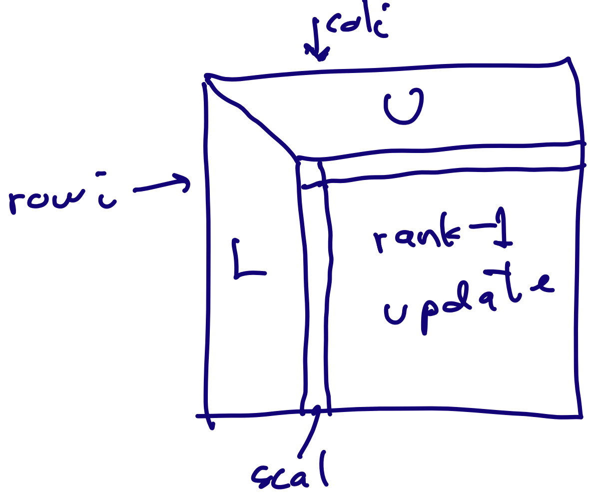
... BLAS 1 scal

$A(i+1:n,i+1:n) = A(i+1:n,i+1:n)$

$- A(i+1:n,i) \cdot A(i,i+1:n)$

... BLAS 2 ger rank-1 update

no data reuse yet, slow



$$\#\text{flops} = \sum_{i=1}^{n-1} (n-i) + 2(n-i)^2 = \frac{2}{3}n^3 + O(n^2)$$

How to pivot, ie. Choose $A(i,i) \neq 0$

Goal: backward stability

$$P \cdot L \cdot U = A + E, \quad \|E\| = O(\varepsilon) \cdot \|A\|$$

not guaranteed by $A(i,i) \neq 0$

Ex: single precision $\varepsilon \sim 10^{-7}$

$$A = \begin{bmatrix} 10^{-8} & 1 \\ 1 & 1 \end{bmatrix} \quad A^{-1} \doteq \begin{bmatrix} -1 & 1 \\ 1 & -10^{-8} \end{bmatrix}$$

$\kappa(A) \sim 2.6$ well-conditioned

$\Rightarrow$ expect accurate answer

$$L = \begin{bmatrix} 1 & 0 \\ \underbrace{10^8}_{fl\left(\frac{1}{10^{-8}}\right)} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 10^{-8} & 1 \\ 0 & \underbrace{fl(1 - 10^8 \cdot 1)}_{= -10^8} \end{bmatrix}$$

Get same L, U if $A(2,2)$ were $.5, -1, \ldots$

because $fl(A(2,2) - 10^8 \cdot 1)$ "forgets"

$A(2,2)$ if small enough, $O(1)$,
so solving $Ax=b$ using this L,U
gives same answer independent of $A(2,2)$
wrong!

Instead: swap rows 1 and 2 $\Rightarrow A(1,1)=1$
and get full accuracy in A^{-1}, $Ax=b$

Intuition: want large entry of A
on diagonal

Recall Q1.10: $C = fl(A\cdot B) = A\cdot B + E$
$|E| \le n\cdot\varepsilon\cdot|A|\cdot|B|$
since $A = P\cdot L\cdot U$, get similar bound:

Thm: (Backward Error of LU)
if P, L, U from Gaussian Elim
$A - E = P\cdot L\cdot U$
$|E| \le n\cdot\varepsilon\cdot P\cdot|L|\cdot|U|$

Cor: Solve $Ax=b$ by GE, forward
substitution with L, backwards with U
computed $\hat{x}$ satisfies
$(A-F)\hat{x} = b$, $|F| \le 3\cdot n\cdot\varepsilon\cdot P\cdot|L|\cdot|U|$

Proof of Cor: assume $P=I$ for simplicity
(imagine running alg on $P^T A$)

Use Q 1.11

Solve $Ly = b$, get $(L+\delta L)\hat{y} = b$ $\quad |\delta L| \le n\cdot\varepsilon\cdot|L|$

Solve $Ux = \hat{y}$, get $(U+\delta U)\hat{x} = \hat{y}$ $\quad |\delta U| \le n\cdot\varepsilon\cdot|U|$

$$
\begin{aligned}
b &= (L+\delta L)\hat{y} = (L+\delta L)(U+\delta U)\hat{x} \\
&= (L\cdot U + \delta L\cdot U + L\cdot\delta U + \delta L\cdot\delta U)\hat{x} \\
&= (A - E + \delta L\cdot U + L\cdot\delta U + \delta L\cdot\delta U)\cdot\hat{x} \quad \text{by Thm} \\
&= (A - F)\hat{x}
\end{aligned}
$$

$$
\begin{aligned}
|F| &\le |E| + |\delta L\cdot U| + |L\cdot\delta U| + |\delta L\cdot\delta U| \\
&\le |E| + |\delta L|\cdot|U| + |L|\cdot|\delta U| + |\delta L|\cdot|\delta U| \\
&\le n\cdot\varepsilon|L|\cdot|U| + n\varepsilon\cdot|L|\cdot|U| + |L|\cdot n\varepsilon|U| \\
&\quad + (n\varepsilon)^2|L|\cdot|U| \\
&\cong 3n\varepsilon|L|\cdot|U| \quad \text{QED of Cor.}
\end{aligned}
$$

Proof sketch of Thm (assume $P = I$)

Trace through alg: how is $U(i,j)$ computed?

when $i \le j$

$$
\begin{aligned}
U(i,j) &= A(i,j) - L(i,1)\cdot U(1,j) \\
&\qquad - L(i,2)\cdot U(2,j) \\
&\qquad \cdots \\
&= A(i,j) - \underbrace{\sum_{k=1}^{i-1} L(i,k)\cdot U(k,j)}_{\text{dot product of row } i \text{ of } L \text{ with col } j \text{ of } U}
\end{aligned}
$$

Use previous analysis of dot prods

when $i>j$ same idea

$$L(i,j) = \frac{A(i,j) - \sum_{k=1}^{j-1} L(i,k)\cdot U(k,j)}{U(j,j)}$$

again use dot product QED

$\Rightarrow$ intuition: want $|L(i,j)|$ to be small

(1) Standard "partial pivoting" (GEPP)

At each step choose largest entry among $A(i:n,i)$

$\Rightarrow |L(k,i)| = |A(k,i)/A(i,i)| \le 1$

Thm: with GEPP, $|L| \le 1$

and $\max(|U(:,i)|) \le 2^{n-1} \max(|A(:,i)|)$

Bad news: attainable in worst case

$\|F\| \le n \cdot \varepsilon \cdot 2^n \|A\|$, lose all precision in single precision for $n \ge 24$

$$\begin{array}{l} \begin{bmatrix} 1 & 0 & 0 & 1 \\ -1\,0 & 1 & 0 & 1\,2 \\ -1\,0 & -1\,0 & 1 & 1\,2 \\ -1\,0 & -1\,0 & -1\,0 & 1\,2 \end{bmatrix} \begin{matrix} \\ \\ 4 \\ 4\ \ 8 \end{matrix} \end{array}$$

Good news: hardly ever happens, GEPP is standard alg

Empirical observation $\frac{\| |L| \cdot |U| \|}{\|A\|} = g \leq n^{2/3}$

g = "growth factor"

If entries of A are random

true with high probability

(2) Complete Pivoting: permute rows and columns so each $A(i,i)$ is largest in all remaining rows and columns

$P_r^T A P_c^T = LU$ called GECP

more stable than GEPP, $O(n^3)$ more expensive

Thm $g < n^{(\log n/4)}$

Empirically $g < n^{1/2}$, not much better than GEPP

rarely used in practice

(3) Tournament Pivoting: needed to hit communication lower bound $O\left(\frac{n^3}{\sqrt{M}}\right)$

(4) Threshold pivoting (sparse case)

tradeoff stability and sparsity, i.e. speed

Error bound for $Ax=b$

$$\frac{\|x-\hat{x}\|}{\|x\|} \leq \kappa(A) \cdot \text{backward error}$$

$$\le \kappa(A) \cdot 3 \cdot n \cdot \varepsilon \cdot g$$

g = growth factor

$$= \kappa(A) \cdot 3 \cdot n \cdot \varepsilon \cdot \frac{\| |L| \cdot |U| \|}{\|A\|}$$

We can estimate $\kappa(A)$ and g in $O(n^2)$ work

What if error bound too large?
or error ok, but too slow, so want to use lower precision?

Try Iterative Refinement, aka Newton's Method

Used mixed precision
- most work ($O(n^3)$ part) in low prec (fast)
- rest, $O(n^2)$, in high prec. (slow)

Low/high could mean
- single/double
- half/single
- double/quad
- other combinations, using 3 precisions, or 5...

Do GEPP to solve $Ax=b$ in low prec
call initial solution $x(1)$

$i = 1$

repeat $r = A \cdot x(i) - b$ in high prec

$O(n^2)$ cost

solve $Ad = r$ in low prec, using

$A = P \cdot L \cdot U$, $O(n^2)$ cost

update $x(i+1) = x(i) - d$ in low prec

$O(n)$ cost

until "convergence"

Testing "convergence" depends on goals:

(1) Getting a small backward in higher prec.

$$\|A x_{comp} - b\| = O(\varepsilon_{high}) \cdot \|A\| \cdot \|x_{comp}\|$$

or get a warning that A too ill-conditioned to converge

Easy to implement because we already have

$$r = A \cdot x_{comp} - b$$

Motivation: use 16-bit accelerators for $O(n^3)$ work

Recent work beyond Newton:
use GMRES instead (Chap 6)

(2) Getting a small relative error in lower precision

$$\frac{\|x - x_{comp}\|}{\|x\|} = O(\varepsilon_{low})$$

or a warning if too ill-conditioned;
convergence criterion complicated, need to avoid being "fooled" by misconvergence
Details in class notes, see LAPACK
sgesvxx

(3) What is value of doing $r = Ax(i) - b$ in low prec?
can get $|E| \le n \cdot \varepsilon \cdot |A|$
i.e. preserve sparsity

Reorganizing GE to Minimize Comm.

Goal: same lower bound as Matmul:
$$\Omega\left(\frac{n^3}{\sqrt{M}}\right)$$

Historically: reorganize GEPP to use BLAS3 (use GEMM i.e. matmul and TRSM, $LX = B$)

Idea: similar induction proof as for GEPP:
do b columns at a time,
apply updates to Schur complement all at once, using GEMM

Ignore pivoting

$$A = \begin{array}{c c} & \begin{array}{cc} b & n-b \end{array} \\ \begin{array}{c} b \\ n-b \end{array} & \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \end{array} = \left[\begin{array}{c|c} L_{11} \cdot U_{11} & A_{12} \\ \hline L_{21} \cdot U_{11} & A_{22} \end{array}\right]$$

where $\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} U_{11}$ using GEPP on b columns

$$= \left[\begin{array}{c|c} L_{11} U_{11} & L_{11} \cdot U_{12} \\ \hline L_{21} U_{11} & A_{22} \end{array}\right]$$

where we solved $A_{12} = L_{11} \cdot U_{12}$ for U_{12} using TRSM

$$= \left[\begin{array}{c|c} L_{11} & 0 \\ \hline L_{21} & I \end{array}\right] \cdot \left[\begin{array}{c|c} U_{11} & U_{12} \\ \hline 0 & \underbrace{A_{22} - L_{21} \cdot U_{12}}_{\text{update } A_{22} \text{ using GEMM}} \end{array}\right] \rightarrow S = \text{Schur complement}$$

repeat on S

Often very fast, but for some combinations of n and M = cache size, can't choose b = block size to reach $O\left(\frac{n^3}{\sqrt{M}}\right)$

Just as for matmul, there is a cache oblivious GEPP that reaches $O\left(\frac{n^3}{\sqrt{M}}\right)$ (1997, Toledo)

High level

Do LU on left half of A

Update right half (U at top, Schur complement at bottom)

Do LU on Schur Complement

Function $[L,U] = RLU(A)$ recursive LU

... assume A $n \times m$, $n \geq m$, m power of 2

if $m = 1$ ... one column

pivot so $A(1,1)$ largest entry, pivot rest of matrix

$L = A / A_{11}$, $U = A_{11}$

else write $A = \begin{array}{c|cc} & m/2 & m/2 \\ \frac{m}{2} & A_{11} & A_{12} \\ n-\frac{m}{2} & A_{21} & A_{22} \end{array}$ $\quad L_1 = \begin{array}{cc} m/2 & \\ \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} & \begin{array}{c} m/2 \\ n-\frac{m}{2} \end{array} \end{array}$

$[L_1, U_1] = RLU\left(\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix}\right)$ ... LU of left half

Solve $A_{12} = L_{11} \cdot U_{12}$ for U_{12} ... update U

$A_{22} = A_{22} - L_{21} \cdot U_{12}$ ... update Schur compl

$[L_2, U_2] = RLU(A_{22})$

$L = \left[L_1, \begin{bmatrix} 0 \\ L_2 \end{bmatrix}\right]^{n \times m}$, $U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}^{m \times m}$

correct by induction

Cost $A(n)$ = recurrence

= # arith ops = $\frac{2}{3}n^3 + O(n^2)$

= same as usual GEPP

$W(n)$ = # words move $= O\left(\frac{n^3}{\sqrt{M}}\right)$

RLU: only minimizes # words moved

not # messages: need more ideas