Welcome to Ma221! Lec 9, Fall24

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

... $A^{n \times m}$ $n \ge m$, $m =$ power of 2

if $m=1$ ... one column

pivot so A_{11} largest entry (GEPP)

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

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

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

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

$A_{22} = A_{22} - L_{21} \cdot U_{12}$

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

$L = \left[L_1 \begin{bmatrix} 0 \\ L_2 \end{bmatrix} \right]^{n \times m}$, $U = \begin{bmatrix} U_1 & U_{12} \\ & U_2 \end{bmatrix}^{m \times m}$

How to use fast matmul, eg Strassen?

Goal: run RLU in $O(n^\omega)$

(1) multiply $L_{21} \cdot U_{12}$ using $O(n^\omega)$ matmul

(2) solve $A_{12} = L_{11} \cdot U_{12}$ for U_{12} using divide and conquer (not as stable as GEPP)

explicitly invert L_{11}, then multiply $(L_{11}^{-1}) \cdot A_{12}$

$$T^{-1} = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}^{-1} = \begin{bmatrix} T_{11}^{-1} & -T_{11}^{-1} \cdot T_{12} \cdot T_{22}^{-1} \\ 0 & T_{22}^{-1} \end{bmatrix}$$

same idea for T upper and lower

recursively invert T_{11} and T_{22}
use $O(n^\omega)$ matmul for $-T_{11}^{-1} \cdot T_{12} \cdot T_{22}^{-1}$

recall $L_{ii} = 1$ $|L_{ij}| \le 1$ so
expect L to be well-conditioned
worst case: $\|L^{-1}\| = O(2^n)$ (rare)

Where to find implementations

Matlab: A\b , or [L,U,P] = lu(A)
rcond, condest to estimate $\kappa(A)$

LAPACK: xGETRF: GEPP x=S/D/C/Z
xGETRF2: GEPP recursively
xGESV: solve $Ax=b$
xGESVX: iterative refinement with residual in prec x
xGESVXX: iterative refinement with residual in prec 2x
xGECON: for condition estimation

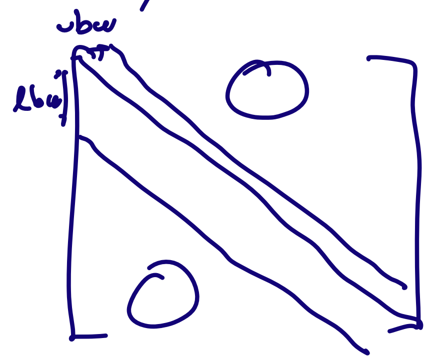
many other libraries
ScaLapack, SLATE, for distributed memory
PLASMA: multicore
MAGMA: GPUs..
⋮

Exploiting Structure

A symmetric positive definite (spd)
- Cholesky, no pivoting, $\frac{1}{2}$ flops of GEPP

A symmetric only
- save half flops, pivoting required for stability, care needed to maintain symmetry

A: band matrix



cost in flops drops from $O(n^3)$ to $O(lbw \cdot ubw \cdot n)$

A: sparse matrices: lots of zeroes

cost, memory drops a lot, depends on nonzero structure, lots of algorithms, software (optimal alg is NP-hard to find, so heuristics)

A: structured matrices: dense but depend on $O(n)$ parameters:

Vandermonde: $V(i,j) = x_i^{j-1}$

Toeplitz: $T(i,j) = t_{i-j}$

many more, discuss most common

Symmetric (Hermitian) Positive Definite

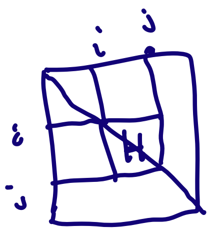
spd or hpd for short

Def: A s.p.d. iff $A = A^T$ and $x^T A x > 0$ if $x \neq 0$

Lemma:

1) X nonsingular $\Rightarrow$ A spd $\Leftrightarrow$ $X^T A X$ spd.

$y^T A y > 0$ $\quad$ $y^T X^T A X y > 0$



2) A spd, $H = A(i:j, i:j)$ "principal submatrix"

$\Rightarrow H$ s.p.d

3) A spd iff $A = A^T$ and all $\lambda_i > 0$

pf: $A = A^T \Rightarrow AQ = Q\Lambda$ $\quad \Lambda = \mathrm{diag}(\lambda_i)$

$Q^T Q = I \Rightarrow Q^T A Q = \Lambda$

$\Rightarrow A$ s.p.d iff Λ s.pd. iff

$x^T \Lambda x = \sum_i \lambda_i x_i^2 > 0 \quad \forall x \neq 0 \Leftrightarrow$ all $\lambda_i > 0$

4) A s.p.d $\Rightarrow A(i,i) > 0$ $(e_i^T A e_i = \lambda_i > 0)$

and $\max_{i,j} |A(i,j)| = \max_i A(i,i)$

pf: suppose $|A(i,j)|$ largest for $i \neq j$

$X =$ all zeros, except $x_i = 1$, $x_j = -\mathrm{sign}(A(i,j))$

$x^T A x = A(i,i) + A(j,j) - 2|A(i,j)| \leq 0$

contradiction

5) A spd iff $A = L \cdot L^T$

L lower triangular, $L_{ii} > 0$

pf: induction on n, show Schur complement s.p.d.

$$A = \begin{array}{c} \\ 1 \\ n-1 \end{array}\begin{array}{c} \begin{array}{cc} 1 & n-1 \end{array} \\ \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] \end{array} = \left[\begin{array}{c|c} \sqrt{A_{11}} & 0 \\ \hline \frac{A_{21}}{\sqrt{A_{11}}} & I \end{array}\right] \left[\begin{array}{c|c} \sqrt{A_{11}} & A_{12}/\sqrt{A_{11}} \\ \hline 0 & S \end{array}\right]$$

$$S = A_{22} - \frac{A_{21} \cdot A_{12}}{A_{11}}$$

$$= \left[\begin{array}{c|c} \sqrt{A_{11}} & 0 \\ \hline \frac{A_{21}}{\sqrt{A_{11}}} & I \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & S \end{array}\right] \left[\begin{array}{c|c} \sqrt{A_{11}} & A_{12}/\sqrt{A_{11}} \\ \hline 0 & I \end{array}\right]$$

$$= X \cdot \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & S \end{array}\right] \cdot X^T$$

$\Rightarrow \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & S \end{array}\right]$ spd by fact 1

$\Rightarrow S$ spd by fact 2

$\Rightarrow$ by induction $S = L_s \cdot L_s^T$

$$\Rightarrow A = \left[\begin{array}{c|c} \sqrt{A_{11}} & 0 \\ \hline \frac{A_{21}}{\sqrt{A_{11}}} & I \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & L_s \end{array}\right] \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & L_s^T \end{array}\right] \left[\begin{array}{c|c} \sqrt{A_{11}} & A_{12}/\sqrt{A_{11}} \\ \hline 0 & I \end{array}\right]$$

$$= \left[\begin{array}{c|c} \sqrt{A_{11}} & 0 \\ \hline \frac{A_{21}}{\sqrt{A_{11}}} & L_s \end{array}\right] \cdot \left[\begin{array}{c|c} \sqrt{A_{11}} & A_{12}/\sqrt{A_{11}} \\ \hline 0 & L_s^T \end{array}\right] = L \cdot L^T$$

Def: A s.p.d then $A = LL^T$, L lower triangular

$L_{ii} > 0$, called Cholesky factorization

Relationship to GE without pivoting

if A spd and $A = L \cdot U$, $L_{ii} = 1$

no pivoting

$$D(i,i) = \sqrt{U(i,i)}$$

$$A = LU = (LD)(D^{-1}U) = L_s \cdot L_s^T$$

is Cholesky

proof: Q 2.7

Fact suggests that any LU alg can be modified to do Cholesky with $\frac{1}{2}$ flops, $\frac{1}{2}$ memory

Simplest Cholesky alg:

for $j = 1:n$

$$L(j,j) = \Big(A(j,j) - \sum_{i=1}^{j-1} L(j,i)^2\Big)^{1/2}$$

$$L(j+1:n,j) = \big(A(j+1:n,j) - L(j+1:n,1:j-1) \cdot L(j,1:j-1)^T\big) / L(j,j)$$

All ideas for speeding up GE apply to Cholesky:

blocking, recursion, comm. lower bound, $O(n^\omega)$ etc.

Error Analysis: Same (simpler) as GE:

$$(A+E)(x+\delta x) = b$$

$$|E| \le 3 \cdot n \cdot \varepsilon \cdot |L| \cdot |L^T| \quad (\text{vs. } P \cdot |L| \cdot |U|)$$

But no pivoting, so how do we bound this?

$$(|L| \cdot |L^T|)_{ij} = \sum_k |L_{ik}| \cdot |L_{jk}|$$

$$\le \| L(i,:) \|_2 \cdot \| L(j,:) \|_2 \quad \text{Cauchy Schwartz}$$

$$= \sqrt{A(i,i)} \cdot \sqrt{A(j,j)}$$

$\le$ largest entry in A (which is on diagonal)

$\Rightarrow$ guaranteed backward stable

$\Rightarrow$ can choose any pivot from diagonal, will do so to maximize sparsity

---

Symmetric Indefinite A

Still possible to save $\frac{1}{2}$ flops, $\frac{1}{2}$ memory but more complicated

Ex: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ can't choose a diagonal pivot

Bunch-Kaufman pivoting

$$A = \left(\prod_i L_i P_i\right) D \left(\prod_i L_i P_i\right)^T$$

$$L_i = \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ 0 & & \vdots & \ddots \\ & & & 1 \end{bmatrix}$$ (column i)

P_i = permutation swaps 1 or 2 columns

D = block diagonal with 1x1 or 2x2 blocks

$$\left(\prod_{i=1}^{n} L_i P_i\right)^{-1} = \left(\prod_{i=n}^{1} P_i^T L_i^{-1}\right)$$

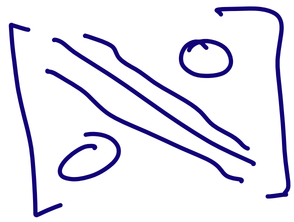
more complicated to use BLAS3
see LAPACK xsytrf

Rook Pivoting: variant of Bunch-Kaufman
searches for a better pivot,
better backward error
xsytrf_rk xsytrf_rook

Aasen Factorization - can hit communication lower bound

$$A = P \cdot L \cdot T \cdot L^T \cdot P^T$$

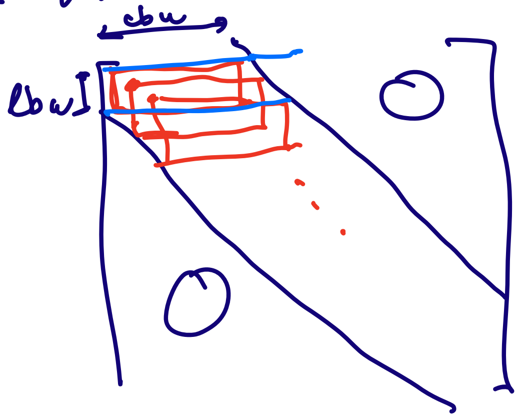
P = perm
L Lower triang

T = tridiagonal 

xsytrf_aa (see class notes)

Sparse matrices

Band matrices



ubw = upper bandwidth

lbw = lower bandwidth

Cost of GE without pivoting

cost per step of GE $= 2 \cdot ubw \cdot lbw$

vs $2n^2$ for dense case

$\Rightarrow$ total cost $= 2n \cdot ubw \cdot lbw + n \cdot lbw$

$= O(n)$ for small bw

Cost with pivoting

$ubw(U) = ubw(A) + lbw(A)$ worst case

"lbw"$(L) = lbw(A)$ but not all adjacent nonzeroes in each column of L

Band matrices arise from discretizing D.E.s

each unknown only depends on nearest neighbors $\Rightarrow$ banded

Ex: Sturm-Liouville problem

$$-y''(x) + q(x)\cdot y(x) = r(x) \quad x \in [0,1]$$

$$y(0) = \alpha, \quad y(1) = \beta \qquad q(x) \geq \bar{q} > 0$$

discretize at $x(i) = ih$ $\quad h = \frac{1}{N+1}$

unknowns are $y(i) = y(x(i))$ for $i = 1..N$

approximate $y''(i) \cong \dfrac{y(i+1) - 2y(i) + y(i-1)}{h^2}$

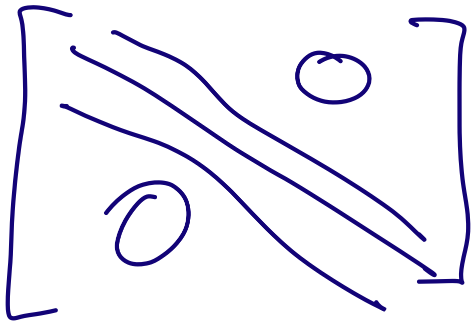
$$q(i) = q(x(i)) \qquad r(i) = r(x(i)) \qquad i = 1 \ldots N$$

$$-\frac{(y(i+1) - 2\cdot y(i) + y(i-1))}{h^2} + q(i)\cdot y(i) = r(i)$$
$i = 1 \ldots N$

or $Ay = r$ for

$$v = r + \left[\frac{\alpha}{h^2}, 0, \ldots, 0, \frac{\beta}{h^2}\right]^T$$

$$A = \operatorname{diag}\left(\frac{2}{h^2} + q(i)\right) \quad \text{... diagonal}$$
$$+ \operatorname{diag}\left(\frac{-1}{h^2}, 1\right) \quad \text{.. above diagonal}$$
$$+ \operatorname{diag}\left(-\frac{1}{h^2}, -1\right) \quad \text{.. below diagonal}$$



Prove A s.p.d to use Cholesky

Gershgorin's Thm: Given any A all its evals lie in circles in complex plane, centers at $A(i,i)$ radii $\sum_{\substack{j=1 \\ j \neq i}}^{n} |A(i,j)|$

proof: $Ax = \lambda x$, $|x(i)|$ largest entry

$$\frac{1}{x(i)}\left[(A(i,i) - \lambda)\, x(i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} A(i,j) \cdot x(j)\right]$$

$$|A(i,i) - \lambda| \leq \sum_{\substack{j=1 \\ j \neq i}}^{n} |A(i,j)| \cdot \underbrace{\left|\frac{x(j)}{x(i)}\right|}_{\leq 1} \quad \text{QED}$$

Apply Gershgorin to Sturm-Liouville:

circles centered at $\frac{2}{h^2} + q(i)$

with radii $\frac{2}{h^2}$, $q(i) \geq \bar{q} > 0$

$\Rightarrow$ all circles in right half plane

$\Rightarrow$ all evals positive $\Rightarrow$ s.p.d