What should convergence criterion be?
Tricky, need to avoid being fooled
by very ill-conditioned A that
looks like it is converging.
See details in typed class notes.

Code available in LAPACK:
sgesvxx, dgesvxx

Ma221 Lecture 5 Segment 5

Return to minimizing communication.
Historically GEPP was written to
perform most work using BLAS3,
Used in LAPACK + ScaLAPACK.
Idea similar to induction proof for
GEPP, but instead of 1 column at a time,
will do b columns at a time, b is
a tuning parameter: For simplicity,
ignore pivoting

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

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

where we have performed GEPP using prior algorithm on

$$\begin{bmatrix} A_{11} \\ A_{21} \end{bmatrix} = \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix} U_{11}$$

(b rows)

$$= \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & A_{22} \end{bmatrix}$$

where have solved $A_{12} = L_{11}U_{12}$ using BLAS3 TRSM

$$= \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & \underbrace{A_{22} - L_{21}U_{12}}_{\text{Schur complement} = S} \end{bmatrix}$$

S computed using BLAS3 GEMM

proceed on S

Most work done in calls to TRSM and GEMM, so should be fast

Often works well, but for some combinations of n and cache size M, can't reach $\Omega\left(\frac{n^3}{\sqrt{M}}\right)$

Just as for matmul, there is a recursive, cache oblivious algorithm (Toledo, 1997)

High Level Algorithm

Do LU on left half of matrix

Update right half of matrix

(U at top, Schur compl. at bottom)

Do LU on Schur Complement

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

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

if $m = 1$ ... one column

pivot so A_{11} largest, update rest of matrix

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

else

... write $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ $L_1 = \begin{bmatrix} L_{11} \\ L_{12} \end{bmatrix}$

$A_{11}, A_{12}, L_{11}, U_1, U_2$ are $m/2 \times m/2$

A_{21}, A_{22}, L_{12} are $n - \frac{m}{2} \times \frac{m}{2}$

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

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

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

Correctness by induction

Recurrences for (m=n)

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

similar recurrence

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

RLU only hits lower bound for #words moved, not # messages

To minimize # messages: either

① Replace partial pivoting by tournament pivoting (discussed later, references in notes)

② Keep GEPP, but more complicated data structure; payoff unclear

How does Strassen etc extend?

Can modify RLU to run in $O(n^{\log_2 7})$ flops

① multiply $L_{21} \cdot U_{12}$ using Strassen

and

② solve $A_{12} = L_{11} U_{12}$ as follows

inverting L_{11} by divide&conquer

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

perform all matmuls using Strassen

Slightly less numerically stable than GEPP

Where to find implementations
(all blocked, unless marked recursive)

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

LAPACK: xGETRF: GEPP where x=S/D/C/Z
xGETRF2 GEPP recursively
xGESV solve Ax=b
xGESVX, condition estimation, iterative refinement with no extra precision
xGESVXX, iterative refinement in extra precision
xGECON for condition estimation alone

ScaLAPACK: PxGETRF etc