# Employing Random Butterfly Transformations in Sparse Direct Solvers

François-Henry Rouet

Lawrence Berkeley National Laboratory

Joint work with: Xiaoye. S. Li (LBNL) and Marc Baboulin (INRIA)

Sparse Days, June 5–6, 2014

# Introduction

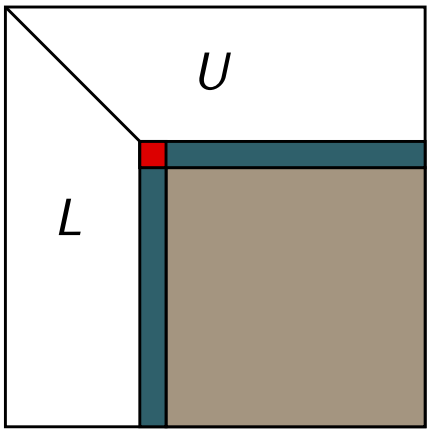
$LU$ decomposition (Gaussian Elimination) for the solution of $Ax = b$

**for** $k = 1$ **to** $n$ **do**

$a_{k+1:n,k} \leftarrow \frac{a_{k+1:n,k}}{a_{kk}}$

$a_{k+1:n,k+1:n} \leftarrow a_{k+1:n,k+1:n} - a_{k+1:n,k} \times a_{k,k+1:n}$

**end for**




- Stability issue: $a_{kk}$ may be small or zero $\Rightarrow$ large element growth $\Rightarrow$ elements of normal size lost in summation.
- Numerical pivoting as a cure: swap rows/columns so that each $a_{kk}$ is large.
  E.g., Partial Pivoting: row $k$ is exchanged with row $p$ such that $|a_{pk}| = \max_{j \geqslant k} |a_{jk}|$
  Eventually, $PA = LU$ ($P$ permutation matrix).

# Introduction

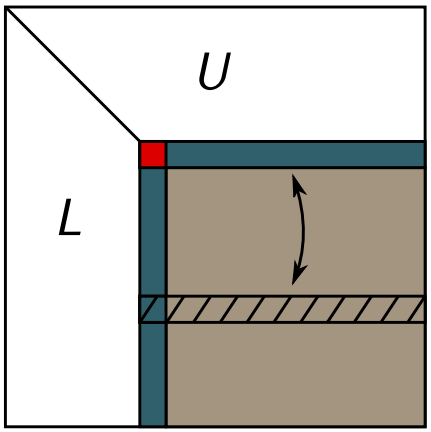
$LU$ decomposition (Gaussian Elimination) for the solution of $Ax = b$

**for** $k = 1$ **to** $n$ **do**

$$a_{k+1:n,k} \leftarrow \frac{a_{k+1:n,k}}{a_{kk}}$$

$$a_{k+1:n,k+1:n} \leftarrow a_{k+1:n,k+1:n} - a_{k+1:n,k} \times a_{k,k+1:n}$$

**end for**




- Stability issue: $a_{kk}$ may be small or zero $\Rightarrow$ large element growth $\Rightarrow$ elements of normal size lost in summation.
- Numerical pivoting as a cure: swap rows/columns so that each $a_{kk}$ is large.
  E.g., Partial Pivoting: row $k$ is exchanged with row $p$ such that $|a_{pk}| = \max_{j \geqslant k} |a_{jk}|$
  Eventually, $PA = LU$ ($P$ permutation matrix).

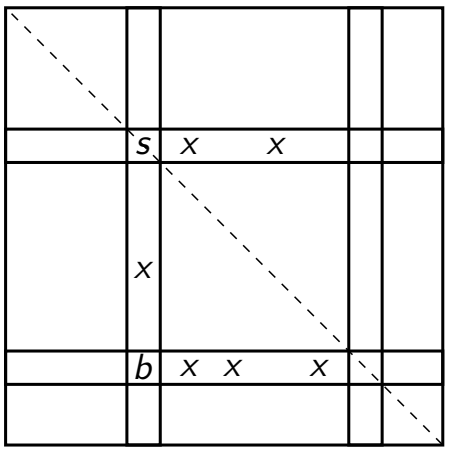
# Permutations – Dense vs. sparse GE

**Dense:** $PA = LU$, $PAQ^T = LU$

- Complete pivoting, partial pivoting, tournament pivoting, etc.
- Data movement. Random matrix with MAGMA: 20-40% penalty.
- Scalability. No $LDL^T$ in ScaLAPACK, MAGMA, PLASMA.

**Sparse:** $PAQ^T = LU$

- $P$ and $Q$ chosen to maintain stability, preserve sparsity, increase parallelism.
- Dynamic pivoting causes dynamic change of $L$ & $U$ structures.
- Example: NLPKKT80, MUMPS factorization with 128 processes:
  - Partial threshold pivoting: 639 seconds.
  - No pivoting (modified to be diagonally dominant): 87 seconds.



# Relaxed strategies in sparse cases

Trade-off: better sparsity at the expense of larger pivot growth.

- Threshold pivoting [Duff/Erisman/Ried '86]
- Compressed threshold pivoting [Hogg/Scott '03]
- Restricted pivoting [Schenk/Gartner '04]
- Static pivoting [Li/Demmel '98]:
  - Column permutation: place large elements on the diagonal (e.g., with MC64).
  - During the factorization, static pivoting: small pivots are replaced by an arbitrary value (e.g., $\sqrt{\varepsilon}||A||$).
  - Iterative refinement applied to the solution.

None of them are very scalable.

# Objectives

- We consider a randomization technique that eliminates the need for pivoting.
- Recent success in the dense case (M. Baboulin et al.).
- This talk presents preliminary results for the sparse case.

# Random Butterfly matrices (D.S. Parker, 1995)

Butterfly matrix: $n \times n$ matrix of the form:

$$B^{<n>} = \frac{1}{\sqrt{2}} \begin{bmatrix} R_0 & R_1 \\ R_0 & -R_1 \end{bmatrix}$$

where $R_0$ and $R_1$ are random diagonal $\frac{n}{2} \times \frac{n}{2}$ matrices.

Recursive Butterfly matrix is a product of butterfly matrices, $n = 2^d$:

$$W^{<n,d>} = \begin{bmatrix} B_1^{<n/2^{d-1}>} & & \\ & \ddots & \\ & & B_{2^{d-1}}^{<n/2^{d-1}>} \end{bmatrix} \cdot W^{<n,d-1>}, \text{ with } W^{<n,1>} = B^{<n>}$$

# Random Butterfly matrices (D.S. Parker, 1995)

Butterfly matrix: $n \times n$ matrix of the form:

$$B^{<n>} = \frac{1}{\sqrt{2}} \begin{bmatrix} R_0 & R_1 \\ R_0 & -R_1 \end{bmatrix}$$

where $R_0$ and $R_1$ are random diagonal $\frac{n}{2} \times \frac{n}{2}$ matrices.

Recursive Butterfly matrix is a product of butterfly matrices, $n = 2^d$:

$$W^{<n,d>} = \begin{bmatrix} B_1^{<n/2^{d-1}>} & & \\ & \ddots & \\ & & B_{2^{d-1}}^{<n/2^{d-1}>} \end{bmatrix} \cdot W^{<n,d-1>}, \text{ with } W^{<n,1>} = B^{<n>}$$

**Remarks:**

- $W^{<n,d>}$ becomes denser with increasing $d$.
- Parker's theory is based on the complete depth $d = \log_2 n$.
  In practice, we also consider a partial depth $d < \log_2 n$.

# Random Butterfly Transformations (RBT)

**Theorem (Parker, 1995)**

Let $U$ and $V$ be two Recursive Butterfly matrices with $d = \log_2 n$. With probability 1, $U^T AV$ can be factorized without pivoting.

(Probability $1 \equiv 1 - O(2^{-t})$ in finite-precision arithmetic with $t$ bits.)

- **Proof:** show that, if $A$ is nonsingular, $\widetilde{A} = U^T AV$ is block nondegenerate, i.e., any principal submatrix $\widetilde{A}(i : i + k, i : i + k)$ is nonsingular.
- **Notation:**
  - $S_k^n$: strictly increasing sequences of length $k$ of $\{1, \ldots, n\}$;
  - $CS_k^n$: end-around consecutive sequences (e.g., $\{n-2, n-1, n, 1, 2\}$).

# RBT: Sketch of proof

- Binet-Cauchy Theorem: for two square $n \times n$ matrices $X, Y$, any sequences $\kappa, \mu \in S_k^n$,

$$\det (X \cdot Y)(\kappa, \mu) = \sum_{\lambda \in S_k^n} \det X(\kappa, \lambda) \cdot \det Y(\lambda, \mu).$$

- For a consecutive sequence $\alpha = 1 : k \in CS_k^n$,

$$\det \widetilde{A}(\alpha, \alpha) = \sum_{\kappa \in S_k^n} \sum_{\lambda \in S_k^n} \det U^T(\alpha, \kappa) \cdot \det A(\kappa, \lambda) \cdot \det V(\lambda, \alpha)$$

It is a polynomial of degree one in the butterfly random variables, which cannot be zero because:

1. The variable polynomials $\det U(\kappa, \alpha) \cdot \det V(\lambda, \alpha)$ are nonzero due to special structure.
2. The coefficients, $\det A(\kappa, \lambda)$, are not all zero, otherwise, $\det A = 0$ by Laplace expansion for determinants.

# Previous work: RBT for dense systems

Baboulin et al. studied practical aspects and high-performance implementations:

- In practice, $d = 1$ or 2 is enough. Iterative refinement can recover lost digits.
- The 2-norm condition number of the matrix is almost unchanged.
- They proposed:
  - A hybrid CPU-GPU implementation for the unsymmetric case.
    $\sim 20\%$ gain over $LU$ with partial pivoting in MAGMA.
  - A multicore implementation for the symmetric case.
    2–4x faster than $LDL^T$ based on LAPACK+multithreaded BLAS.

# New work: RBT for sparse systems

Primary issue: $\widetilde{A} = U^T A V$ has more nonzeros than $A$.

Assuming $d = 1$, i.e., $U = \begin{bmatrix} U_0 & U_1 \\ U_0 & -U_1 \end{bmatrix}, V = \begin{bmatrix} V_0 & V_1 \\ V_0 & -V_1 \end{bmatrix}$,

$$\widetilde{A} = \begin{bmatrix} U_0(A_{11} + A_{21} + A_{12} + A_{22})V_0 & U_0(A_{11} + A_{21} - A_{12} - A_{22})V_1 \\ U_1(A_{11} - A_{21} + A_{12} - A_{22})V_0 & U_1(A_{11} - A_{21} + A_{12} + A_{22})V_1 \end{bmatrix}$$

- Worst case: $\widetilde{A}$ has 4x as many nonzeros as $A$.
- Best case: $A_{11} = A_{22}, A_{12} = A_{21}$ : $\widetilde{A}$ is block-diagonal with the same number of nonzeros as $A$.

# Preserve sparsity: (1) one-sided RBT

Consider a one-sided approach $\widetilde{A} = U^T A$ (vs two-sided $U^T AV$):

- Maximum increase in nonzeros is 2x.
- However, no more "cancellation" effect.

**Informal argument:**

- Recall, 2-sided $\widetilde{A} = U^T AV$, for $\alpha \in CS_k^n$:

$$\det \widetilde{A}(\alpha, \alpha) = \sum_{\kappa \in S_k^n} \sum_{\lambda \in S_k^n} \det U^T(\alpha, \kappa) \cdot \det A(\kappa, \lambda) \cdot \det V(\lambda, \alpha)$$

- Now, 1-sided $\widetilde{A} = U^T A$:

$$\det \widetilde{A}(\alpha, \alpha) = \sum_{\kappa \in S_k^n} \det U^T(\alpha, \kappa) \cdot \det A(\kappa, \alpha)$$

It is not possible for $\det A(\kappa, \alpha)$ to be zero for all $\kappa$.

# Preserve sparsity: (2) combine sparsity ordering

Two strategies:

- Strategy 1: First apply sparsity ordering, then apply RBT:

$$\widetilde{A} = U^T (QAQ^T) V$$

- Strategy 2: First apply RBT, then apply sparsity ordering:

$$\widetilde{A} = Q(U^T AV) Q^T$$

Strategy 2 may preserve sparsity better but...

# Strategy 2 pitfall

$\widetilde{A} = Q(U^T AV)Q^T$ not guaranteed to be factorizable without pivoting, because $Q$ permutes some nonprincipal submatrix (possibly singular) to a leading principal one.

**Example:** Let $A$ be a $4 \times 4$ matrix, written in $2 \times 2$ block form:

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

Let $U$ and $V$ be two RB matrix with size 4 and degree $d = 2$.
Let $q$ be the permutation vector $q = [1\ 3\ 2\ 4]$.

If $\sum A_{11} = \sum A_{22} = \sum A_{12} = \sum A_{21}$ (element-wise summation), the leading submatrix $\widetilde{A}(1:2, 1:2)$ is singular (thus the factorization fails at the second pivot.

# Evaluation

- 90 sparse matrices from UFL ($n \leqslant 10,000$, nonsingular, unsymmetric).
- Preprocessing: diagonal scaling to equilibrate the system.
- Compare with SuperLU:
  - Sparsity ordering $Q$ with AMD on the graph of $A^T A$.
  - Partial pivoting (dynamic): $P(AQ^T) = LU$.
- RBT:
  - Sparsity ordering $Q$ with AMD on the graph of $A^T + A$.
  - Factorization without pivoting.

# Overall summary (1/2)

- Increase in nonzeros: $\mathrm{nnz}(\widetilde{A})/\mathrm{nnz}(A)$.
- Increase in factors: $\mathrm{nnz}(LU(\widetilde{A}))/\mathrm{nnz}(LU(A))$.
- Mean = geometric mean.

| Strategy | | Success rate | Increase in nonzeros | | | Increase in factors | | |
|---|---|---|---|---|---|---|---|---|
| | | | min | mean | max | min | mean | max |
| Strat. 1 | $d = 1$ | 81.1% | 1.00 | 2.97 | 3.99 | 1.12 | 9.92 | 362.32 |
| | $d = 2$ | 92.2% | 2.01 | 9.53 | 15.79 | 1.14 | 19.35 | 635.84 |
| Strat. 2 | $d = 1$ | 82.2% | 1.00 | 2.02 | 4.00 | 0.03 | 1.55 | 20.42 |
| | $d = 2$ | 80.0% | 1.50 | 4.95 | 15.01 | 0.06 | 2.96 | 144.49 |

# Overall summary (2/2)

**Strategy 2, $d = 1$. 90 matrices.**

Factor size compared to GEPP:

- 37 have smaller size.
  - Absence of dynamic pivoting allows the sparsity ordering to do a better job.
- 30 have $\leqslant$ 2x increase.
- 23 have 2x-20x increase.

Forward error $\frac{\|x - x_{true}\|}{\|x_{true}\|}$ compared to GEPP:

- 69 lost fewer than 2 digits.
- 9 lost all digits...

Overall: 48 matrices (53%) have:

- Less than 2x increase in factor size.
- And fewer than 2 lost digits.

# Flops?

We mostly focused on the structure of $U^T AV$ and its factors.

**What about flops?** 90 matrices:

- 23 have less flops. Min: 100x decrease!
- 9 have $\leqslant 2$ increase.
- 32 have 2x-10x increase. Mean: 3x increase.
- 26 have 10x+ increase. Max: $10^3$ increase...

## Flop-reducing orderings? (with **E. Ng & B. Peyton**)

Mean minimum local fill: remove the vertex that minimizes deficiency/size(clique). For 4 matrices that exhibit a small increase in factors but large increase in flops:

- 17% flop reduction over AMD.
- 9% factor reduction over AMD.

# Source of errors

The theoretical argument guarantees no zero pivots but doesn't say anything about small pivots.

- In our tests, the mean is a $5 \cdot 10^1$ increase, but 10 matrices have $10^{15}$...
- Pivot growth for a dense random matrix:

| GEPP | One-sided | | | Two-sided | | |
|---|---|---|---|---|---|---|
| | $d = 1$ | $d = 2$ | $d = \log_2 n$ | $d = 1$ | $d = 2$ | $d = \log_2 n$ |
| $4 \cdot 10^1$ | $2 \cdot 10^7$ | $2 \cdot 10^5$ | $9 \cdot 10^3$ | $5 \cdot 10^5$ | $2 \cdot 10^5$ | $5 \cdot 10^2$ |

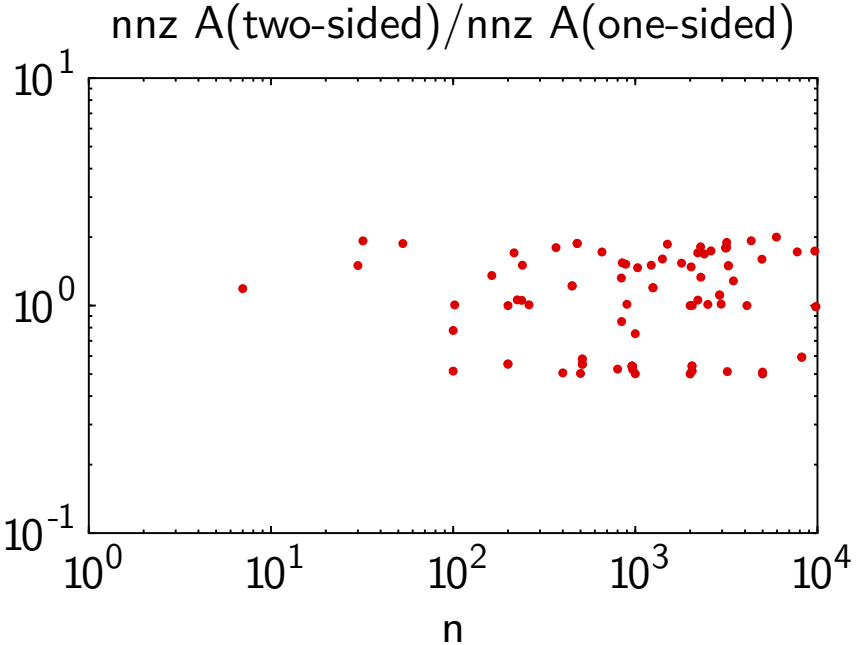
- Wilkinson matrix: pivot with RBT $\simeq 10^1$.

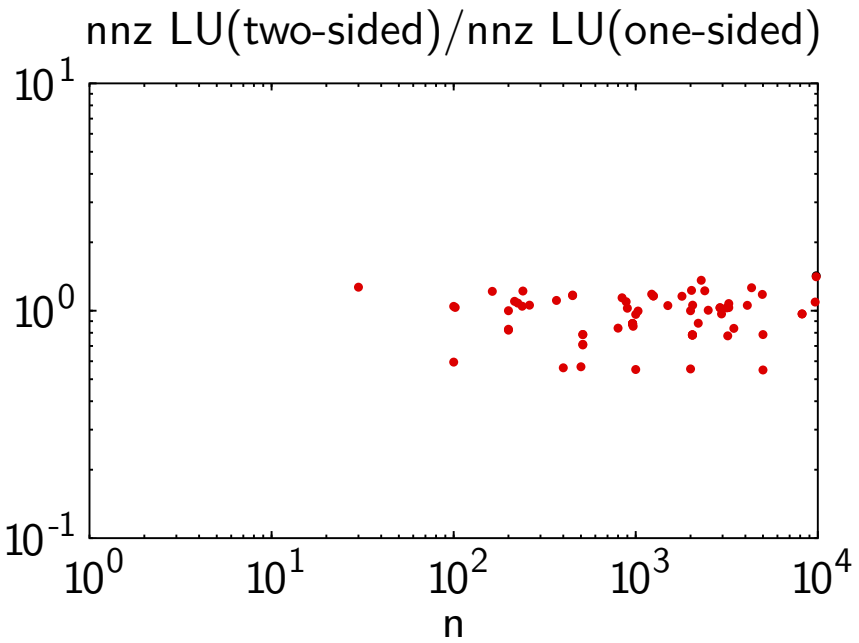
$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 & 1 \\ -1 & \ddots & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ \vdots & & & \ddots & \ddots & 1 \\ -1 & \cdots & \cdots & \cdots & -1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 & 1 \\ -1 & \ddots & \ddots & & \vdots & 2 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & \vdots \\ \vdots & & & \ddots & \ddots & 2^{n-1} \\ -1 & \cdots & \cdots & \cdots & -1 & 2^n \end{bmatrix}$$

# 1-sided vs. 2-sided RBT

**Strategy 2, $\mathbf{d = 1}$.**

The success rates are similar with 1-sided and 2-sided.



# Final remarks

**Results:**

- RBT with partial depth is an attractive alternative to traditional pivoting at extreme scale.
- No simple conclusion yet in sequential evaluation.

# Final remarks

**Results:**

- RBT with partial depth is an attractive alternative to traditional pivoting at extreme scale.
- No simple conclusion yet in sequential evaluation.

**Future work:**

- Parallel implementation.
- Analysis of pivot growth.
- Classification of problems according to various RBT strategies.
- RBT for sparse $LDL^T$ factorization, i.e., $\widetilde{A} = U^T A U$.
  - Augmented or KKT systems.
- RBT kernels in intermediate supernodal or frontal matrices.
  - Advantage: no disruption to sparsity.

# End

Thank you for your attention!

Any questions?