# NUMERICAL STABILITY OF SYMMETRIC INDEFINITE SOLVERS: DIRECT METHODS

**Miro Rozložník**

*Institute of Computer Science*
*Czech Academy of Sciences, Prague*

joint work with Gil Shklarski and Sivan Toledo

*Seminar of Numerical Analysis SNA09, Ostrava*

February 2-6, 2009

# Outline

Introduction

Bunch-Kaufmann factorization

Parlett-Reid reduction

Aasen's factorization

Numerical stability

Conclusions

# Solution of a symmetric indefinite system of linear equations

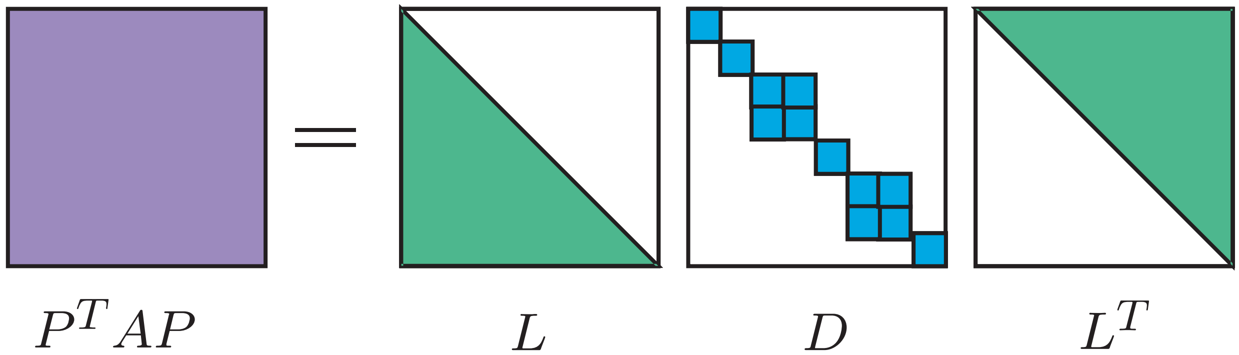
$$Ax = b$$

$A$ is symmetric (indefinite)

$$P^T A P P^T x = P^T b$$

$$LDL^T P^T x = P^T b \qquad\qquad LTL^T P^T x = P^T b$$

# Block Bunch-Kaufmann factorization



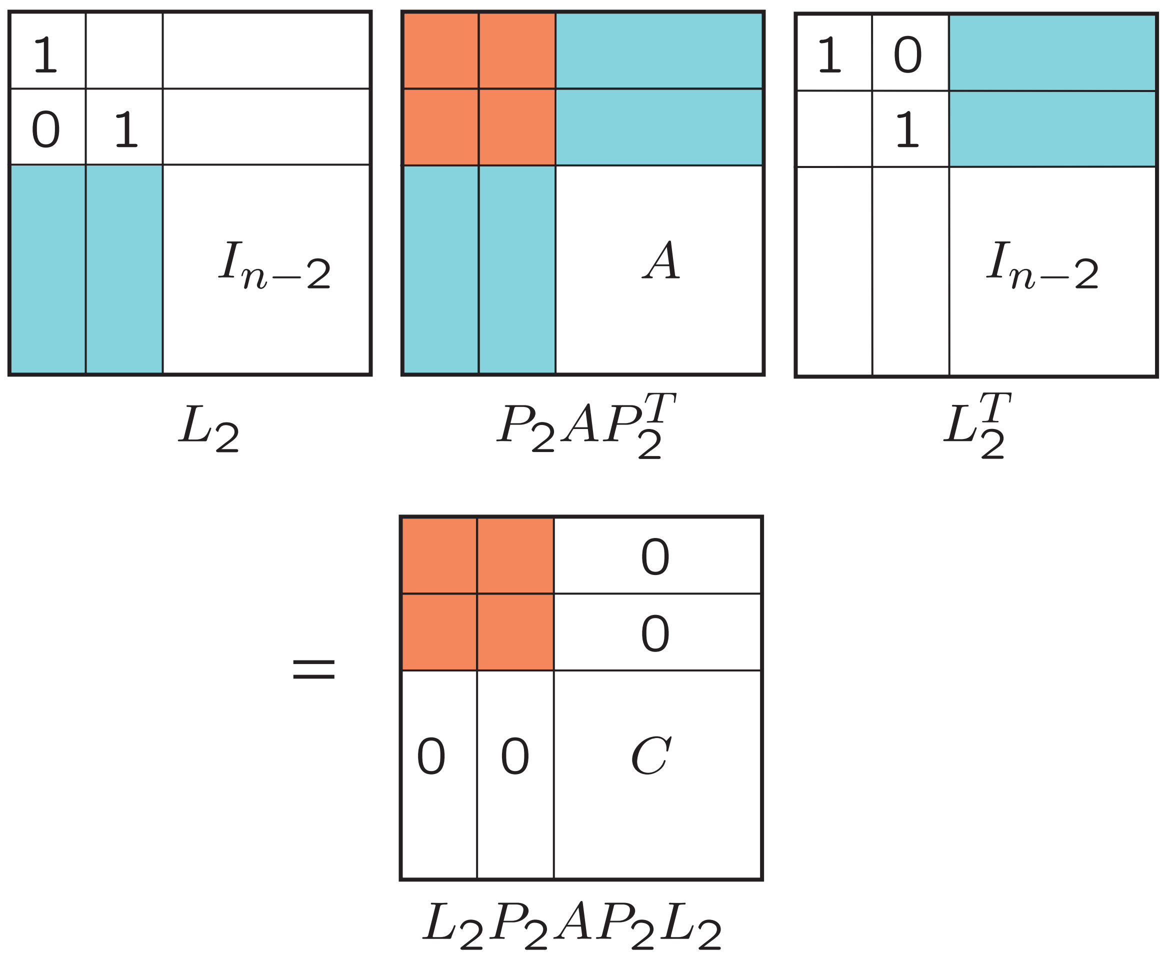
$A$ is symmetric (definite or indefinite)

$L$ is **unit** lower triangular

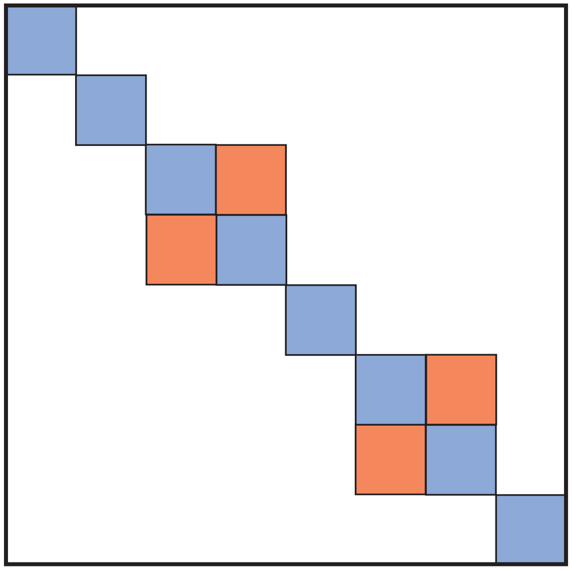
$D$ is **symmetric block diagonal** with 1x1,2x2 blocks

$P$ is a **permutation** matrix

# Bunch-Kaufmann factorization



# Bunch-Kaufmann factorization

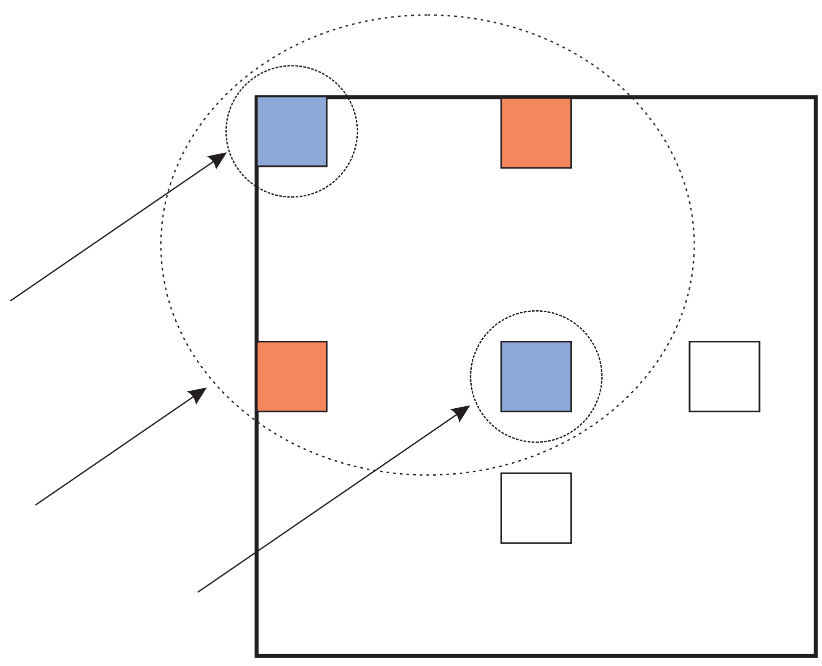


$$\underbrace{L_{n-1}P_{n-1}\ldots L_2P_2L_1P_1AP_1^TL_1^TP_2^TL_2^T\ldots P_{n-1}^TL_{n-1}^T}_{D}$$

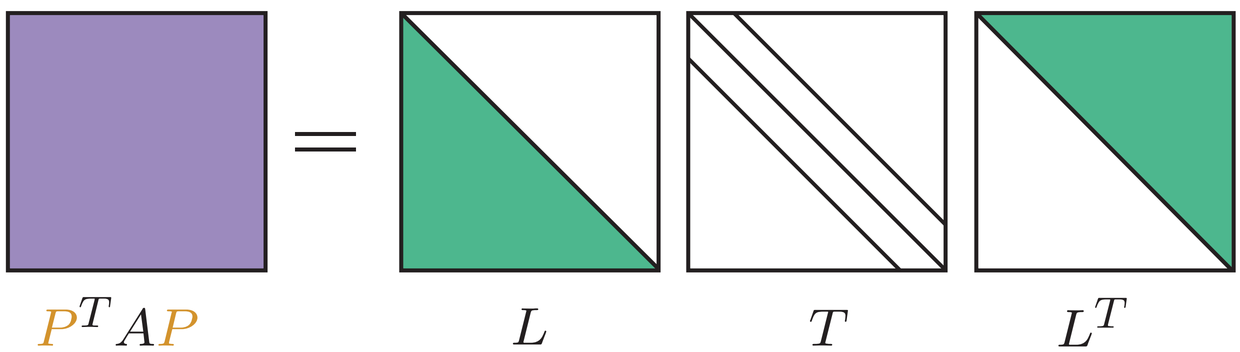
# Bunch-Kaufmann pivoting strategy

- complete pivoting $O(n^3)$ comparisons Bunch, Parlett

- partial pivoting $O(n^2)$ comparisons implemented in LINPACK, LAPACK



# Triangular tridiagonalization




$A$ is symmetric (definite or indefinite)

$L$ is **unit** lower triangular

$T$ is **symmetric** tridiagonal

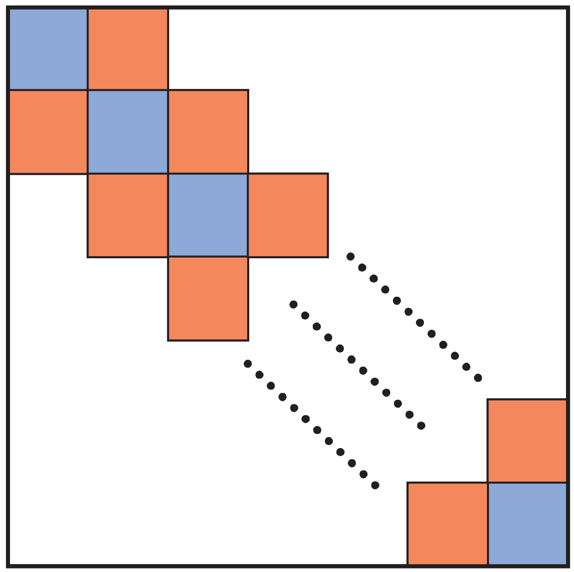
$P$ is a **permutation** matrix

# Parlett - Reid reduction

$$\underbrace{\begin{bmatrix} 1 & & \\ 0 & 1 & \\ 0 & -\frac{\nu}{\alpha} & I_{n-2} \end{bmatrix}}_{L_2} \underbrace{\begin{bmatrix} A_{1,1} & \alpha & \nu^T \\ \alpha & \multicolumn{2}{c}{\multirow{2}{*}{$A_{2:n,2:n}$}} \\ \nu & & \end{bmatrix}}_{P_2{}^T A P_2^T} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ & 1 & -\nu/\alpha \\ & & I_{n-2} \end{bmatrix}}_{L_2^T}$$

$$= \underbrace{\begin{bmatrix} A_{1,1} & \alpha & 0 \\ \alpha & \multicolumn{2}{c}{\multirow{2}{*}{$C$}} \\ 0 & & \end{bmatrix}}_{L_2 P_2{}^T A P_2^T L_2^T}$$

# Parlett - Reid reduction



$$\underbrace{\underbrace{L_{n-1}P_{n_1}\dots L_3P_3L_2P_2}_{L^{-1}P^T}\,A\,\underbrace{P_2^TL_2^TP_3^TL_3^T\dots P_{n-1}^TL_{n-1}^T}_{PL^{-T}}}_{T}$$

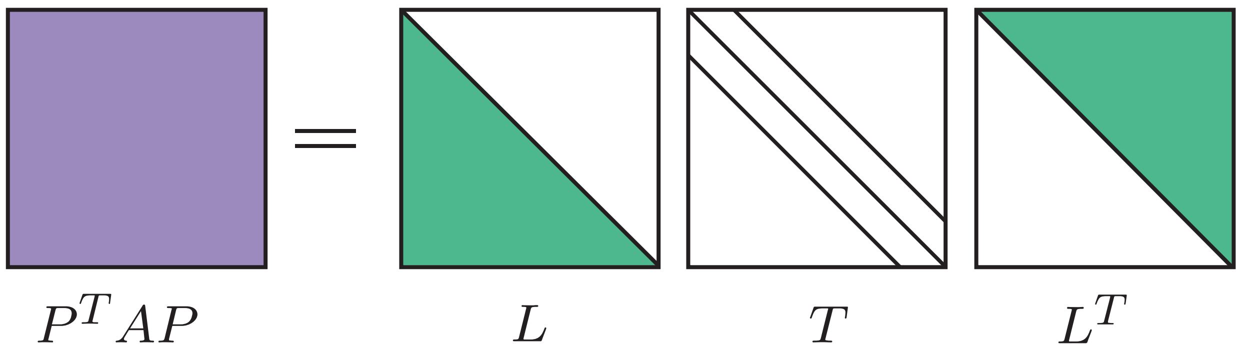
# Parlett - Reid reduction

- The reduced matrix remains symmetric during reduction, the updates are performed on a half of the matrix

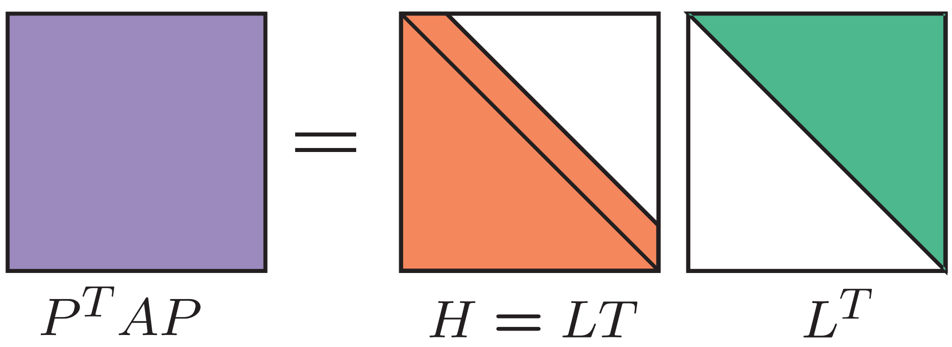
- Complexity: at each step two rank-one updates on half a matrix $2(n-1)^2$; $O(n)$ other operations; total $2/3n^3 + O(n^2)$

$\longrightarrow$ Aasen's factorization

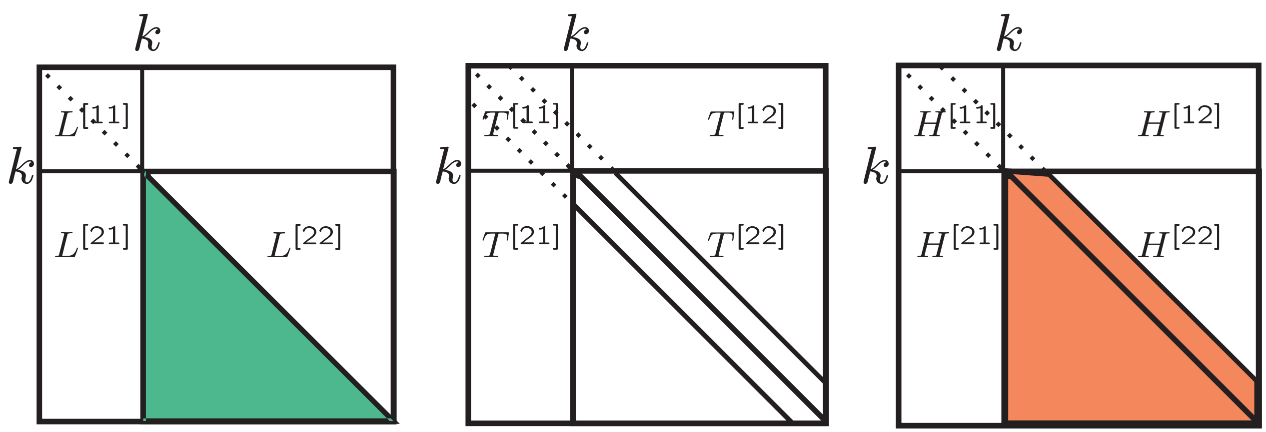
# Parlett - Reid reduction




# Aasen's factorization



# Notation



# Parlett - Reid reduction

works on $L^{[22]}T^{[22]}(L^{[22]})^T$

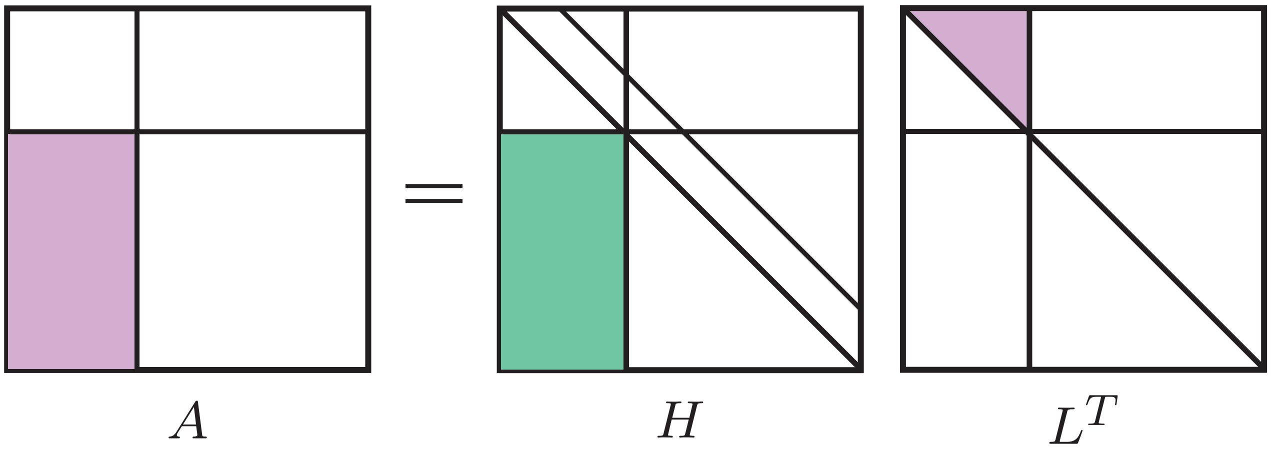
# Aasen's factorization

* for $k > 1$ $L^{[22]}T^{[22]}(L^{[22]})^T \neq H^{[22]}(L^{[22]})^T$

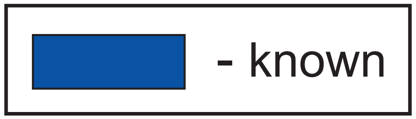
* update of $A^{[22]}$
  compute $(k+1)$-th column $L$ and $k$-th column of $T$ and $H$

* pivoting strategy

# Aasen's factorization – Phase 1

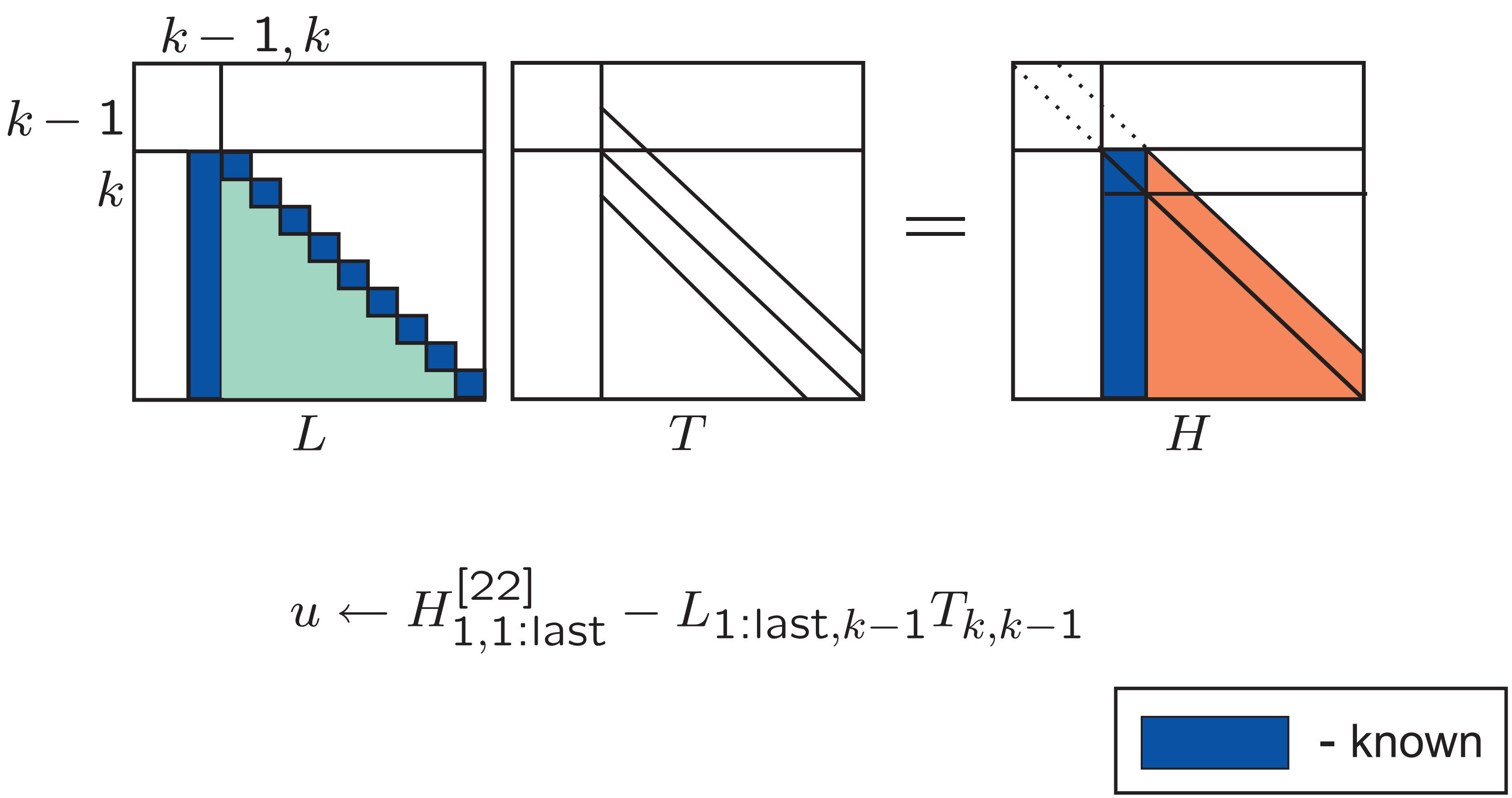



Compute $i$-th column of $H^{[21]}$ from $i$-th column of $A^{[21]}$ and previous columns of $H^{[21]}$ and $L^{[11]}$



# Aasen's factorization – Phase 2

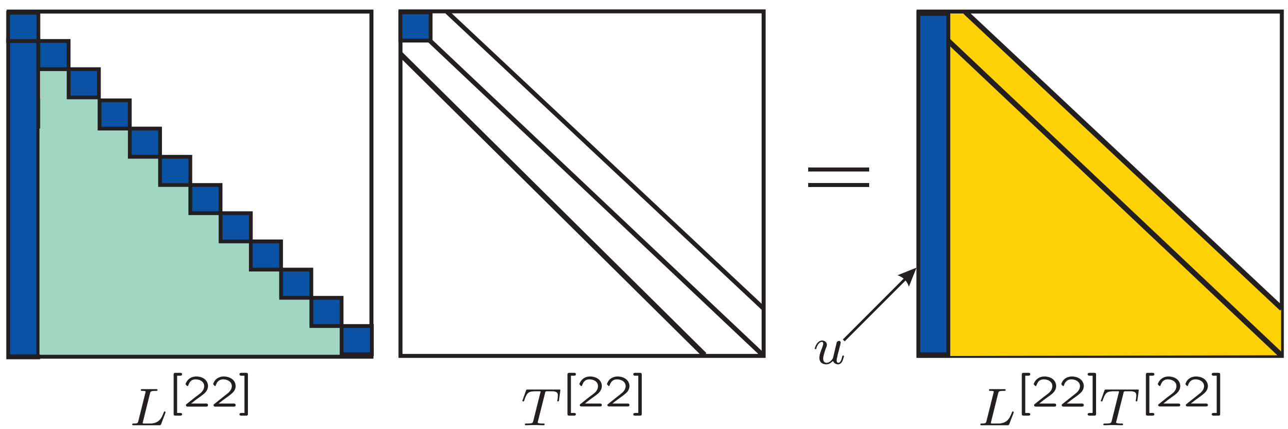
Second phase – extract the first column of $L^{[22]}T^{[22]}$



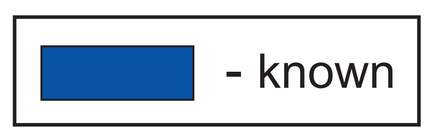

$$u \leftarrow H^{[22]}_{1,1:\text{last}} - L_{1:\text{last},k-1} T_{k,k-1}$$

# Aasen's factorization – Phase 3

Third phase – extract $T^{[22]}_{1,1}$

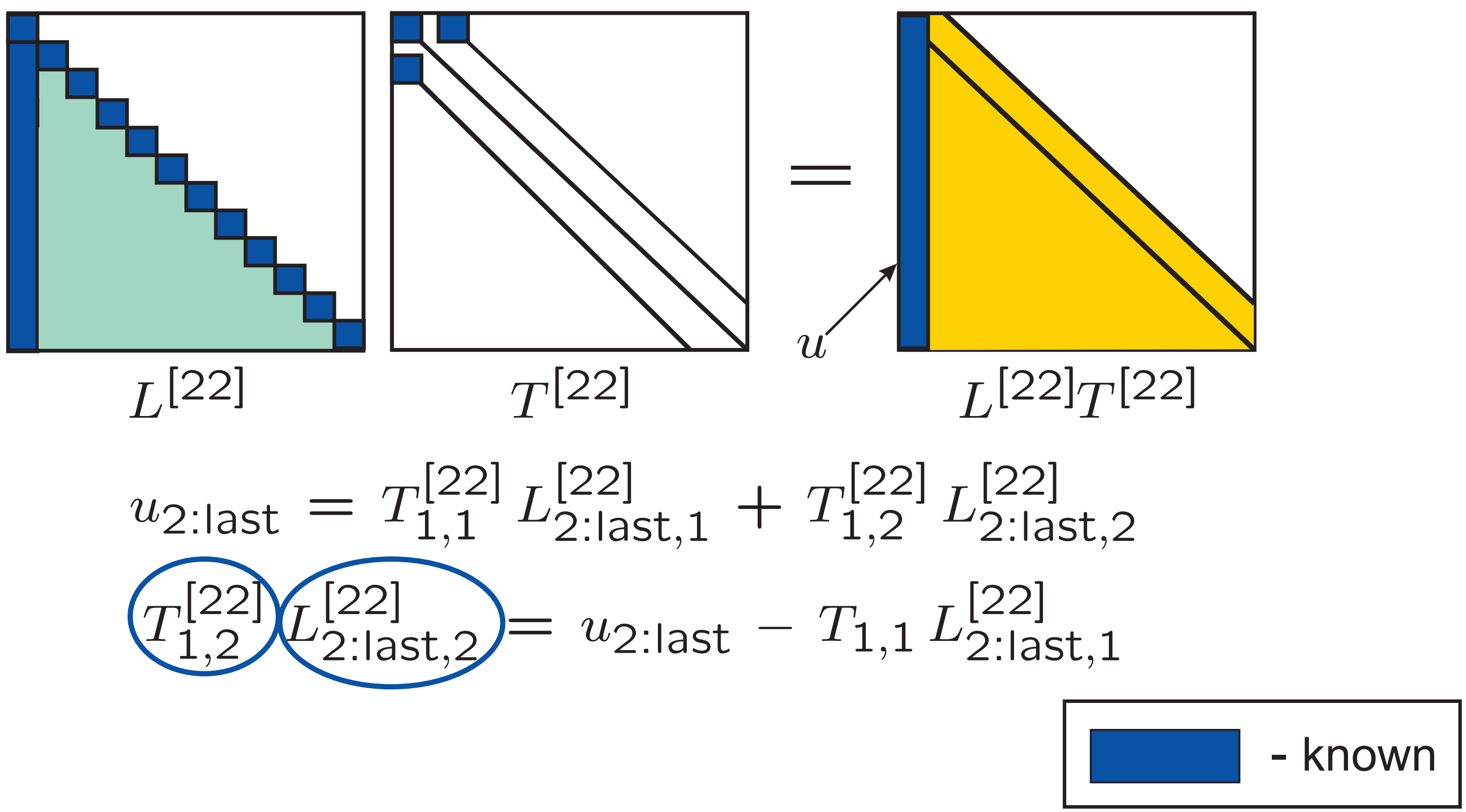



$$T^{[22]}_{1,1} = u_1$$



# Aasen's factorization – Phase 4

Fourth phase – extract $T^{[22]}_{1,2}$ and $L^{[22]}_{2:\text{last},2}$

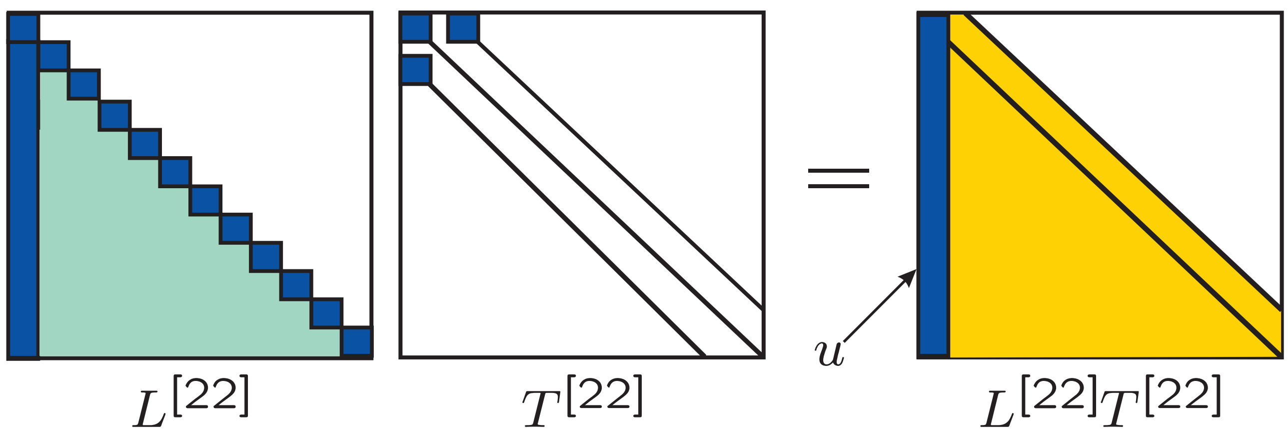



$$u_{2:\text{last}} = T^{[22]}_{1,1}\, L^{[22]}_{2:\text{last},1} + T^{[22]}_{1,2}\, L^{[22]}_{2:\text{last},2}$$

$$T^{[22]}_{1,2}\, L^{[22]}_{2:\text{last},2} = u_{2:\text{last}} - T_{1,1}\, L^{[22]}_{2:\text{last},1}$$

# Aasen's factorization – pivoting strategy

Fourth phase – extract $T_{1,2}^{[22]}$ and $L_{2:\text{last},2}^{[22]}$



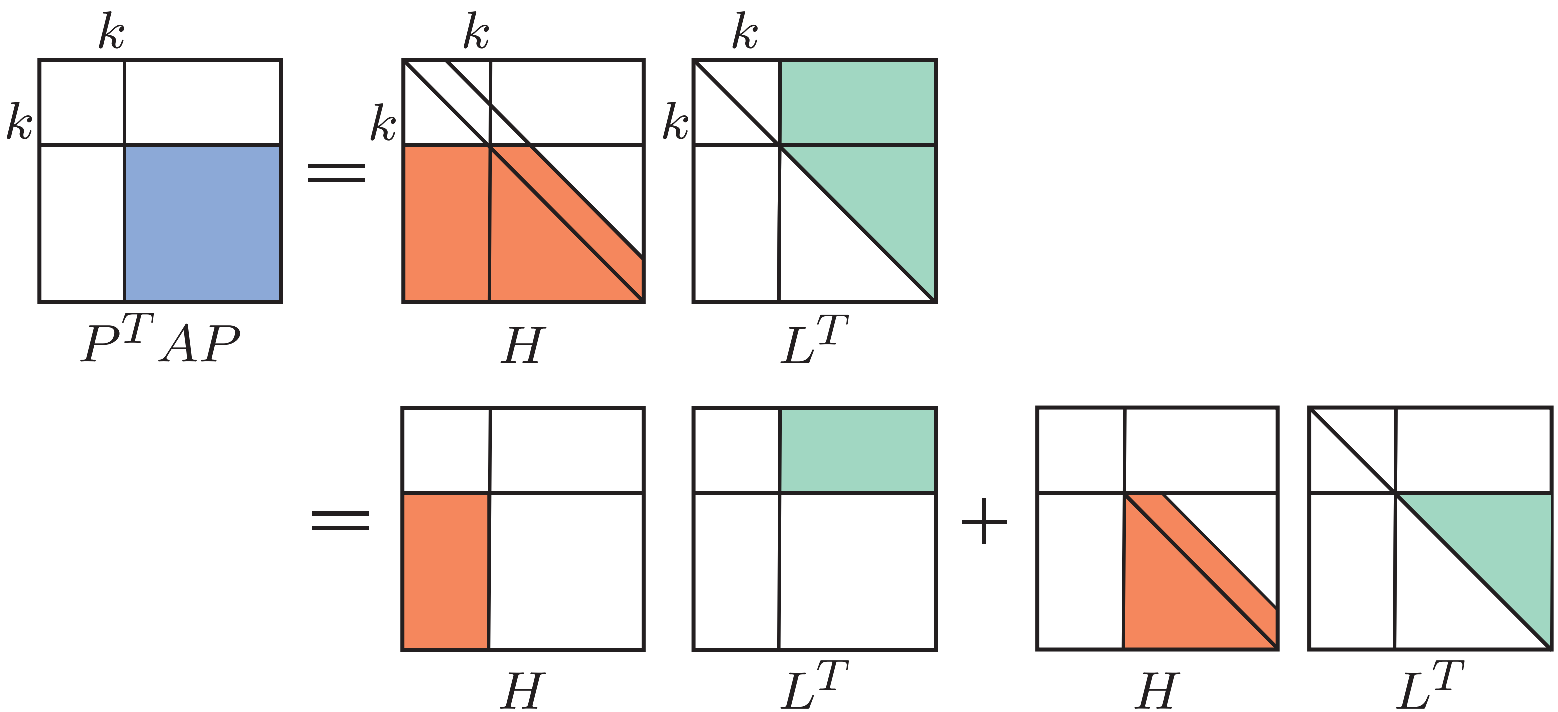

$$u_{2:\text{last}} = T_{1,1}^{[22]}\, L_{2:\text{last},1}^{[22]} + T_{1,2}^{[22]}\, L_{2:\text{last},2}^{[22]}$$

$$T_{1,2}^{[22]}\, L_{2:\text{last},2}^{[22]} = \underbrace{u_{2:\text{last}} - T_{1,1}\, L_{2:\text{last},1}^{[22]}}$$

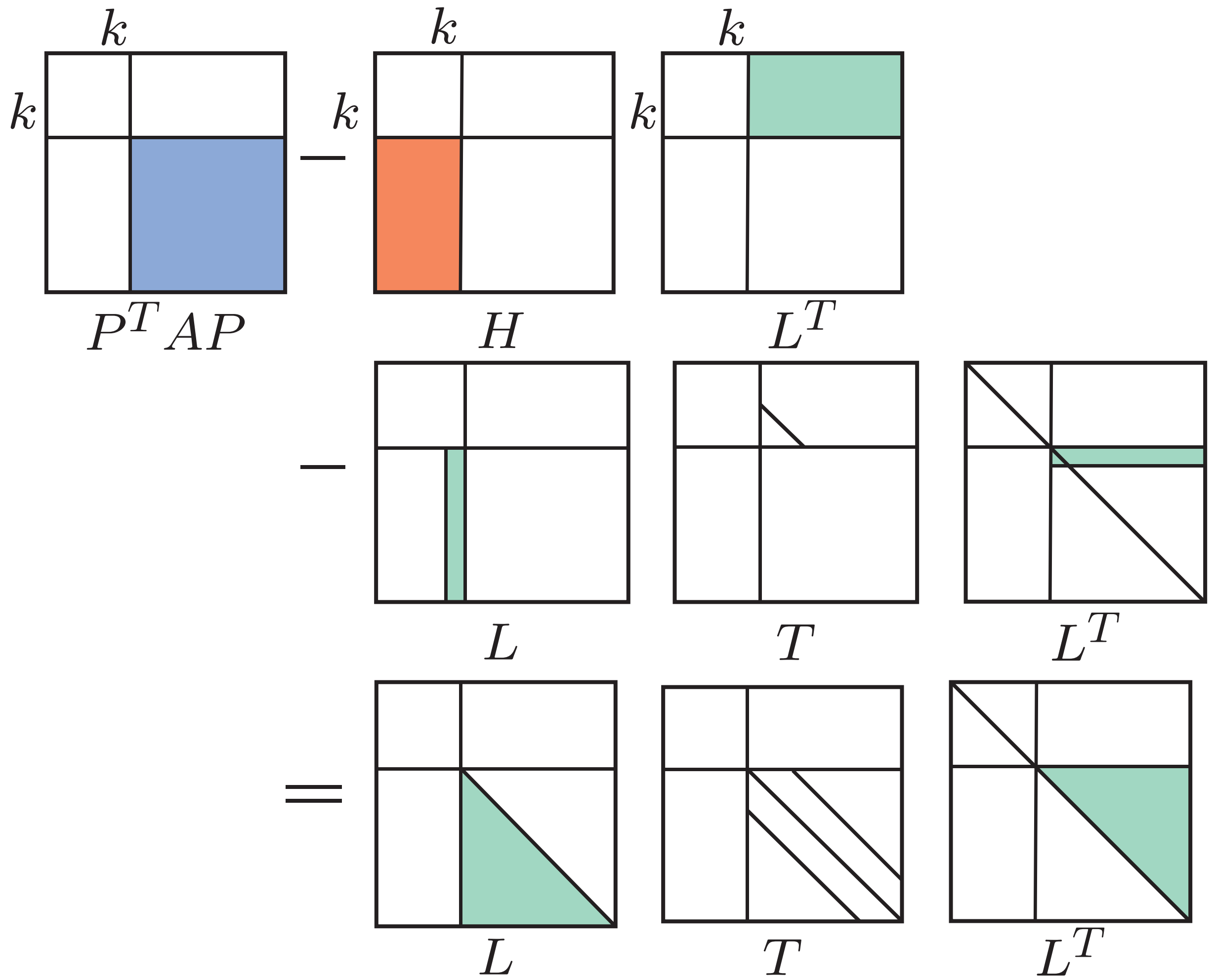
First switch variable $(i+1)$
with the largest index in

- known

# Partitioned factorization



# Partitioned factorization



# Numerical stability – main result

$$A + \Delta A = \overline{L}\,\overline{T}\,\overline{L}^T$$

$$|\Delta A| \le c_3(n,k)\, u|\overline{L}|\,|\overline{T}|\,|\overline{L}|^T$$

$$c_3(n,k) = c_1\left(n + \left\lfloor \frac{n}{k} \right\rfloor + 2\right), \quad c_3(n,1) = c_1(n+3)$$

# Basic assumptions on BLAS

$$X \in \mathbb{R}^{m,k}, \quad Y \in \mathbb{R}^{k,n}, \quad Z = XY \in \mathbb{R}^{m,n}, \quad \overline{Z} = fl(XY)$$

conventional BLAS:

$$|\overline{Z} - Z| \leq c_1(k)u|X|\,|Y| \quad c_1(k) = \frac{k}{1 - ku}$$

Strassen:

$$\|\overline{Z} - Z\| \leq c_3(m,n,k,p)u\|X\|\,\|Y\|$$

# Numerical stability – Proof 1

$$
\begin{aligned}
A^{[11]} + \Delta A^{[11]} &= \overline{L}^{[11]}\,\overline{T}^{[11]}\left(\overline{L}^{[11]}\right)^T \\
\left|\Delta A^{[11]}\right| &\leq c_3(k,1)u\left|\overline{L}^{[11]}\right|\left|\overline{T}^{[11]}\right|\left|\overline{L}^{[11]}\right|^T \\
A^{[21]} + \Delta A^{[21]} &= \overline{H}^{[21]}\left(\overline{L}^{[11]}\right)^T \\
\left|\Delta A^{[21]}\right| &\leq c_1(k)u\left|\overline{H}^{[21]}\right|\left|\overline{L}^{[11]}\right|^T \\
\overline{H}^{[21]} + \Delta H^{[21]} &= \overline{L}^{[21]}\,\overline{T}^{[11]} + \overline{L}^{[22]}_{:,1}\overline{T}^{[21]}_{1,k} \\
\left|\Delta H^{[21]}\right| &\leq c_1(3)\left(\left|\overline{L}^{[21]}\right|\left|\overline{T}^{[11]}\right| + \left|\overline{L}^{[22]}_{:,1}\right|\left|\overline{T}^{[21]}_{1,k}\right|\right)
\end{aligned}
$$

# Numerical stability – Proof 2

$$\overline{C}^{[22]} + \Delta C^{[22]} = A^{[22]} - \overline{H}^{[21]} \left(\overline{L}^{[21]}\right)^T - \overline{L}^{[21]}_{:,k} \overline{T}^{[21]}_{1,k} \left(\overline{L}^{[22]}_{:,1}\right)^T$$

$$\left|\Delta C^{[22]}\right| \leq c_1(k+1)\, u \left(\left|\overline{H}^{[21]}\right| \left|\overline{L}^{[21]}\right|^T + \left|\overline{L}^{[21]}_{:,k}\right| \left|\overline{T}^{[21]}_{1,k}\right| \left|\overline{L}^{[22]}_{:,1}\right|^T\right.$$

$$\overline{C}^{[22]} + \Delta \overline{C}^{[22]} = \overline{L}^{[22]} \overline{T}^{[22]} \left(\overline{L}^{[22]}\right)^T$$

$$\left|\Delta \overline{C}^{[22]}\right| \leq c_3(n-k, k) u \left|\overline{L}^{[22]}\right| \left|\overline{T}^{[22]}\right| \left|\overline{L}^{[22]}\right|^T$$

# Solution of a linear system

$$\text{Assuming } c_4(n) u k_\infty\left(\overline{T}\right) < 1$$

$$\left(A + \widehat{\Delta A}\right)\overline{x} = b + \widehat{\Delta b}$$

$$\left\|\widehat{\Delta A}\right\|_\infty \le c_5(n,k)u\left\|\overline{T}\right\|_\infty, \left\|\widehat{\Delta b}\right\|_\infty \le c_5(n,k)u\left\|\overline{T}\right\|_\infty \|\overline{x}\|_\infty$$

$$\text{growth factor} \qquad \rho_n = \frac{\max_{i,j}\left|\overline{T}_{i,j}\right|}{\max_{i,j}\left|A_{i,j}\right|}$$

$$\max\left\{\frac{\left\|\widehat{\Delta A}\right\|_\infty}{\|A\|_\infty}, \frac{\left\|\widehat{\Delta b}\right\|_\infty}{\|A\|_\infty\|\overline{x}\|_\infty}\right\} \le c_5(n,k)nu\rho_n$$

# Bunch Kaufmann factorization – numerical stability

$$P(A + \Delta A)P^T = \overline{L}\,\overline{D}\,\overline{L}^T$$

$$|\Delta A| \leq c_6(n)u\left(|A| + \left|\overline{L}\right|\left|\overline{D}\right|\left|\overline{L}\right|^T\right)$$

# Solution of a linear systems

Assuming that $c_7(n)uk\left(\overline{D}\right) < 1$

$$\left(A + \widehat{\Delta A}\right)\overline{x} = b$$

$$\left|\widehat{\Delta A}\right| \le c_6(n)u\left(|A| + \left|\overline{L}\right|\,\left|\overline{D}\right|\,\left|\overline{L}\right|^T\right)$$

growth factor $\qquad \rho_n = \dfrac{\max_{i,j,k}|\overline{a}_{ij}^{(k)}|}{\max_{i,j}|a_{ij}|}$

$$\frac{\left\|\widehat{\Delta A}\right\|_\infty}{\|A\|_\infty} \le c_6(n)nu\rho n$$

# Parallel implementation

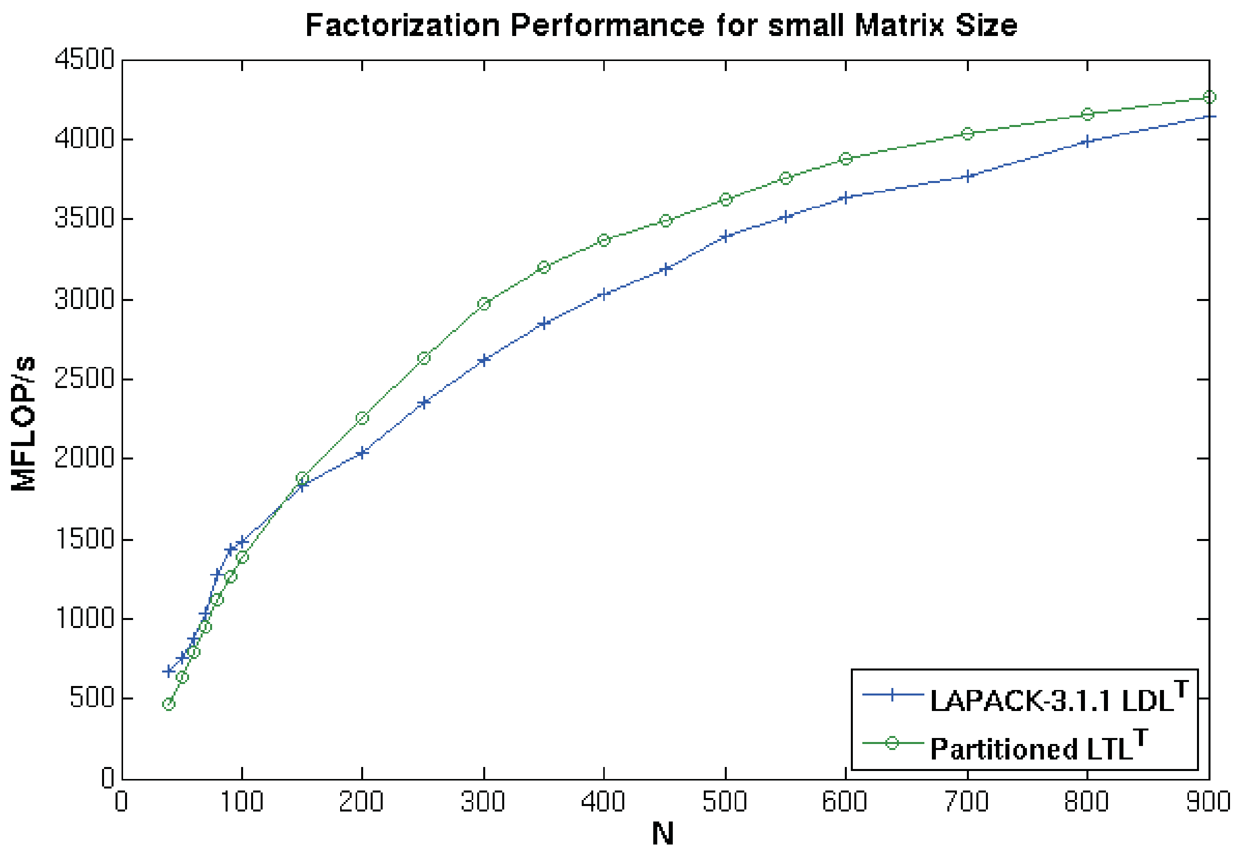
LAPACK uses blocked Bunch-Kaufmann factorization
(Dongarra, Anderson)

Cache-efficient partitioned triangular tridiagonalization
(Shklarski, Toledo – submitted to ACM TOMS)

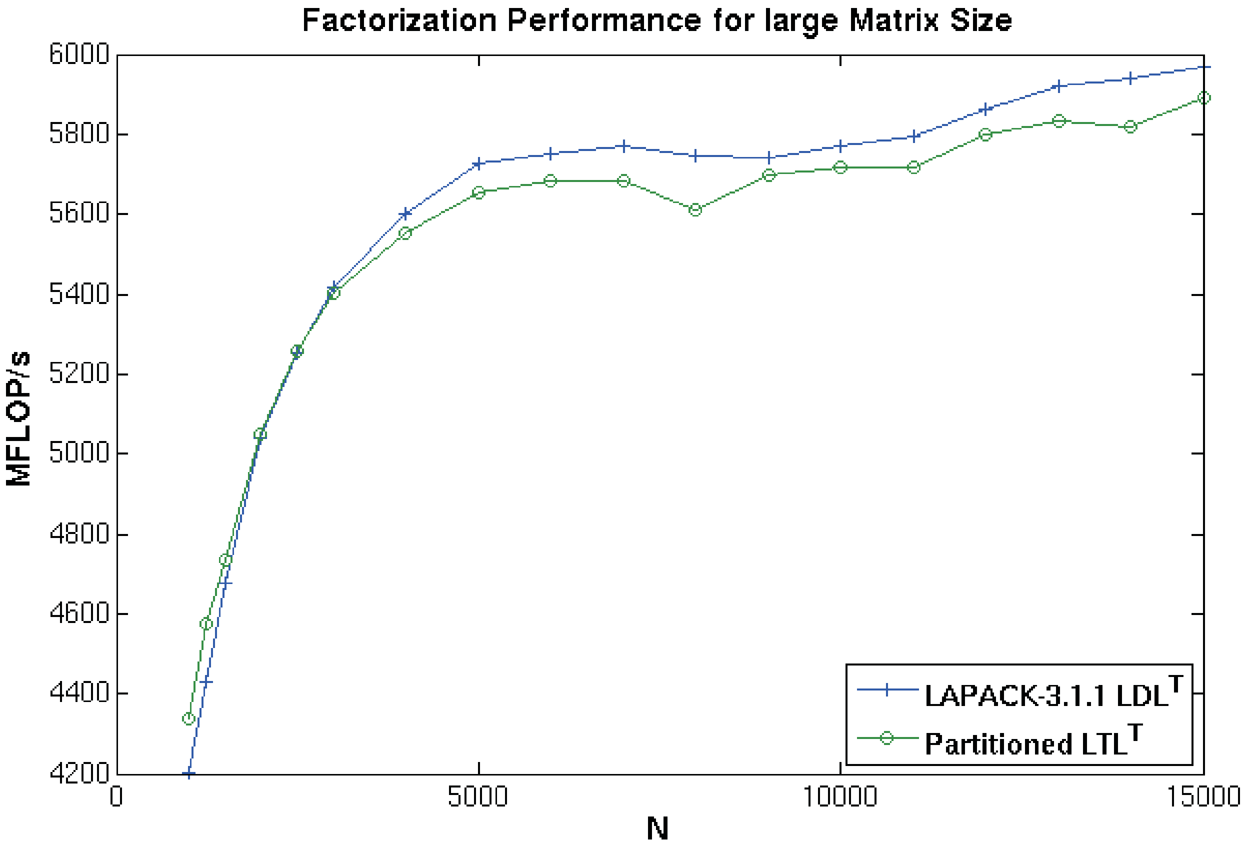
# Numerical examples

- Machine
  - Core 2 Duo 6400, 2.13Ghz, 4GB of main memory
  - Linux x86_64
- Partitioned $LTL^T$
  - C implementation, GCC
  - Batch size $=$ 64
  - Fused $LTL^T$ factor and $QR$ of $T$
- Partitioned $LDL^T$
  - LAPACK 3.1.1 (Bunch-Kaufman)
  - Block size $=$ 64
- BLAS: GOTO BLAS 1.12, confined to a single core
- Matrices:
  - Symmetric matrices, elements uniformly distributed in $(-1, 1)$

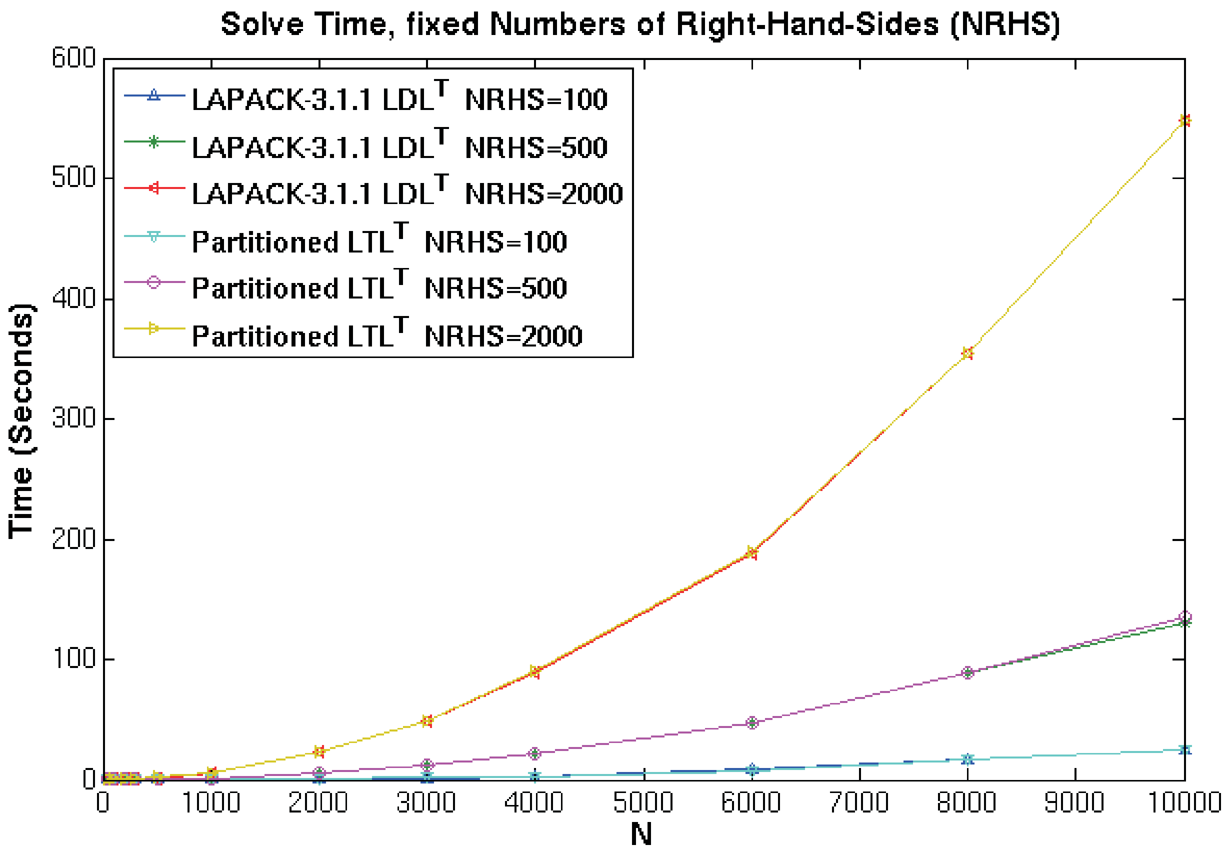
# Numerical examples



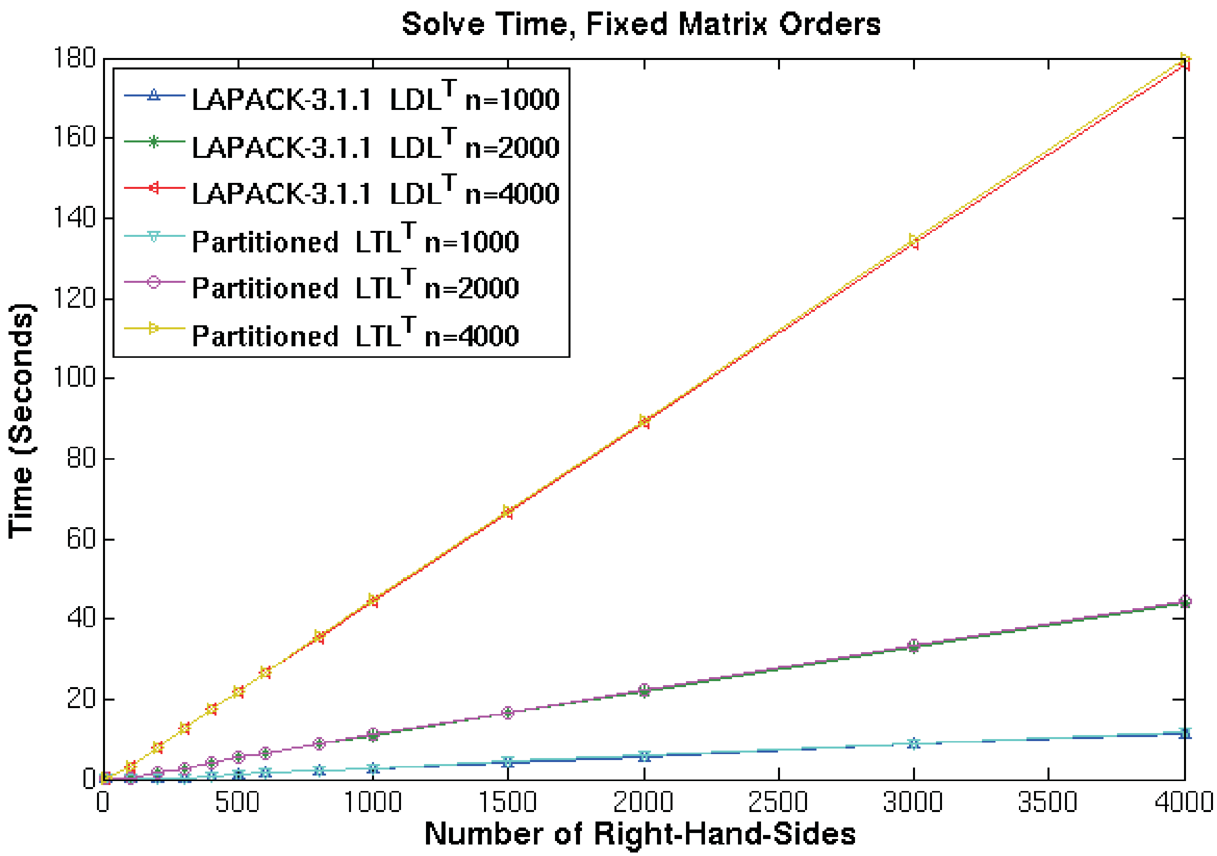
# Numerical examples



# Numerical examples



# Numerical examples



# Conclusions

| $LDL^T$ | $LTL^T$ |
|---|---|
| Reveals inertia | Does not reveal inertia |
| Easy to solve with $D$ | Slightly harder to solve with $T$ |
| Bunch Kauffman Pivoting | Simple Pivoting |
| $L_{i,j}$ can grow | Bounded $L_{i,j}$ |
| Bounded $D$ | $T$ can grow |

# THANK YOU FOR YOUR ATTENTION

R, G. Shklarski, S. Toledo: Partitioned triangular tridiagonalization, submitted to ACM Transactions on Mathematical Software

C and Matlab Codes at

`http://www.tcu.ac.il/~stoledo/research.html`