The Science of Programming High-Performance Linear Algebra Libraries

John A. Gunnels

IBM T.J. Watson Research Center
Yorktown Heights, NY
gunnels@us.ibm.com

June 22, 2002

In collaboration with
Paolo Bientinesi (UT-Austin),
Fred Gustavson (IBM T.J. Watson),
Greg Henry (Intel)
Margaret Myers (UT-Austin),
Enrique Quintana-Ortí (UJI,Spain), and
Robert van de Geijn (UT-Austin)

$\boxed{\text{Motivation}}$

Consider matrix-matrix multiplication

- Problem: Compute $C = AB + C$
  - ⋆ Sequential algorithm:

$$
\begin{array}{l}
\textbf{for } i = 1:m \\
\quad \textbf{for } j = 1:n \\
\quad\quad \textbf{for } p = 1:k \\
\quad\quad\quad \gamma_{ij} = \alpha_{ip}\beta_{pj} + \gamma_{ij} \\
\quad\quad \textbf{endfor} \\
\quad \textbf{endfor} \\
\textbf{endfor}
\end{array}
$$

- Requirements:
  - ⋆ High performance
  - ⋆ Distributed memory parallel
  - ⋆ All cases: AB, A^TB, AB^T, A^TB^T
- Solution: Implemented as part of a well known package

Problem:

Unmanageable Complexity?

- For a complete dense linear algebra library we need a substantial library of routines:
  - ⋆ Functionality.
    - ◇ LAPACK and beyond.
  - ⋆ Accomodating multiple layers of memory hierarchy:
    - ◇ Sequential at each level of memory hierarchy
    - ◇ Shared memory parallel
    - ◇ Distributed memory parallel
    - ◇ Out-of-core (parallel)
    - ◇ Hierarchical data structures
  - ⋆ Variants
    - ◇ Three or more variants per operation per layer of hierarchy

Requirements

- Correctness
- Coverage
- Performance
  - ⋆ A variant for all occasions
  - ⋆ Parameterized cost estimates
  - ⋆ Strategy for picking variants
- Numerical Stability

Solution

- A formal approach to deriving all algorithms
  - ⋆ Correctness
  - ⋆ Coverage
- A framework for coding that mirrors the derived algorithms
  - ⋆ No need for debugging and testing
  - ⋆ No need for documentation
- Performance
  - ⋆ Some variants perform better than others depending on machine and problem parameters.

Overview

- Formal Derivation of Linear Algebra Algorithms
- Impact on Implementation of Linear Algebra Algorithms
- Impact on Performance
- Impact on Library Development and Maintainability
- Current Research/Future Directions
- Conclusion

$\boxed{\text{Formal Derivation Linear Algebra Algorithms}}$

Example: LU Factorization without pivoting

$$A \rightarrow LU$$

where

- A, L and U are $n \times n$,
- L is unit lower triangular, and
- U is upper triangular.

A typical derivation of the right-looking algorithm:

Partition and equate

$$\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right) = \left(\begin{array}{c|c} L_{11} & 0 \\ \hline L_{21} & L_{22} \end{array}\right) \left(\begin{array}{c|c} U_{11} & U_{12} \\ \hline 0 & U_{22} \end{array}\right) = \left(\begin{array}{c|c} L_{11}U_{11} & L_{11}U_{12} \\ \hline L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{array}\right)$$

where A_{11}, L_{11}, and U_{11} are $b \times b$. Then the following equalities must hold:

$$\begin{array}{rcl|rcl} A_{11} & = & L_{11}U_{11} & A_{12} & = & L_{11}U_{12} \\ \hline A_{21} & = & L_{21}U_{11} & A_{22} & = & L_{21}U_{12} + L_{22}U_{22} \end{array}$$

giving the algorithm

- $$\begin{array}{c||c|c} & & \\ \hline\hline & A_{11} \leftarrow \{L \backslash U\}_{11} = LU(A_{11}) & A_{12} \leftarrow U_{12} = L_{11}^{-1} A_{12} \\ \hline & A_{21} \leftarrow L_{21} = A_{21} U_{11}^{-1} & A_{22} \leftarrow A_{22} - L_{21}U_{12} \end{array}$$
- recursively factor $A_{22} \rightarrow L_{22}U_{22}$

A derivation of the left-looking algorithm:

Partition A, L, and U conformally

$$A = \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right), L = \left(\begin{array}{c||c|c} L_{00} & 0 & 0 \\ \hline\hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array}\right), \text{ and } U = \left(\begin{array}{c||c|c} U_{00} & U_{01} & U_{02} \\ \hline\hline 0 & U_{11} & U_{12} \\ \hline 0 & 0 & U_{22} \end{array}\right)$$

where $\{A, L, U\}_{00}$ are $k \times k$ and $\{A, L, U\}_{11}$ are $b \times b$. Update current contents of A from $A^{(k)} = \left(\begin{array}{c||c|c} \{L\backslash U\}_{00} & A_{01} & A_{02} \\ \hline\hline L_{10} & A_{11} & A_{12} \\ \hline L_{20} & A_{21} & A_{22} \end{array}\right)$ to $A^{(k)} = \left(\begin{array}{c||c|c} \{L\backslash U\}_{00} & U_{01} & A_{02} \\ \hline\hline L_{10} & \{L\backslash U\}_{11} & A_{12} \\ \hline L_{20} & L_{21} & A_{22} \end{array}\right)$.

Algorithm:

- $$\begin{array}{c||c|c} & A_{01} \leftarrow U_{01} = L_{00}^{-1} A_{01} & \\ \hline\hline & A_{11} \leftarrow \{L\backslash U\}_{11} = LU(A_{11} - L_{10}U_{01}) & \\ \hline & A_{21} \leftarrow (A_{21} - L_{20}U_{01})U_{11}^{-1} & \end{array}$$
- Repartition and continue

Observation

- The two derivations appear to be quite different.
- Pictorially they appear similar.
- We now show how to unify the approach using formal derivation.
- The key is to ask the question "What is in the matrix at a given stage?"
- Would like to derive all important variants

Formal Derivation Worksheet

We will show how to derive a family of LU factorization algorithms. The derivation

- is constructive and
- guarantees correctness.

Worksheet

Step	Annotated Algorithm: $[D, E, F, \ldots] = \mathrm{op}(A, B, C, D, \ldots)$
1a	$\{P_{\mathrm{pre}}\}$
4	**Partition** **where**
2	$\{P_{\mathrm{inv}}\}$
3	**while** G **do**
2,3	$\{(P_{\mathrm{inv}}) \wedge (G)\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\{P_{\mathrm{inv}}\}$
	enddo
2,3	$\{(P_{\mathrm{inv}}) \wedge \neg (G)\}$
1b	$\{P_{\mathrm{post}}\}$

Step	Annotated Algorithm: $A := L\backslash U = \mathsf{LU}(A)$
1a	$\{P_{\rm pre}\}$
4	**Partition** **where**
2	$\{P_{\rm inv}\}$
3	**while** G **do**
2,3	$\{(P_{\rm inv}) \wedge (G)\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\{P_{\rm inv}\}$
	enddo
2,3	$\{(P_{\rm inv}) \wedge \neg (G)\}$
1b	$\{P_{\rm post}\}$

Step	Annotated Algorithm: $A := L\backslash U = \mathsf{LU}(A)$
1a	$\left\{ A = \hat{A} \right\}$
4	**Partition** **where**
2	$\{P_{\rm inv}\}$
3	**while** G **do**
2,3	$\{(P_{\rm inv}) \wedge (G)\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\{P_{\rm inv}\}$
	enddo
2,3	$\{(P_{\rm inv}) \wedge \neg (G)\}$
1b	$\left\{ A = L\backslash U \wedge \hat{A} = LU) \right\}$

Determine a loop invariant

Partition A, L, U, and $\hat{A}$ into regions:

$$A = \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right), L = \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array}\right), U = \left(\begin{array}{c||c} U_{TL} & U_{TR} \\ \hline\hline 0 & U_{BR} \end{array}\right), \text{ and } \hat{A} = \left(\begin{array}{c||c} \hat{A}_{TL} & \hat{A}TR \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)$$

where A_{TL}, L_{TL}, U_{TL}, and $\hat{A}_{TL}$ are all square matrices of equal dimension. Here "TL", "TR", "BL" and "BR" represent where we are in the matrix.

Plugging partitioned matrices into the postcondition we find

$$\left(\begin{array}{c||c} \hat{A}_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right) = \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array}\right) \left(\begin{array}{c||c} U_{TL} & U_{TR} \\ \hline\hline 0 & U_{BR} \end{array}\right) = \left(\begin{array}{c||c} L_{TL}U_{TL} & L_{TL}U_{TR} \\ \hline\hline L_{BL}U_{TL} & L_{BL}U_{TR} + L_{BR}U_{BR} \end{array}\right)$$

and thus the following equalities must hold:

$$\begin{array}{c||c} \hat{A}_{TL} = L_{TL}U_{TL} & \hat{A}_{TR} = L_{TL}U_{TR} \\ \hline\hline \hat{A}_{BL} = L_{BL}U_{TL} & \hat{A}_{BR} = L_{BL}U_{TR} + L_{BR}U_{BR} \end{array}$$

Question: What can be in the matrix at an intermediate step?

$$\begin{array}{c||c} \hat{A}_{TL} = L_{TL}U_{TL} & \hat{A}_{TR} = L_{TL}U_{TR} \\ \hline\hline \hat{A}_{BL} = L_{BL}U_{TL} & \hat{A}_{BR} - L_{BL}U_{TR} = L_{BR}U_{BR} \end{array}$$

Possible conditions (loop-invariants): $\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) =$

Eager

$\{L\backslash U\}_{TL}$	U_{TR}
L_{BL}	$\hat{A}_{BR} - L_{BL}U_{TR}$

Lazy

$\{L\backslash U\}_{TL}$	$\hat{A}_{TR}$
$\hat{A}_{BL}$	$\hat{A}_{BR}$

Row-lazy

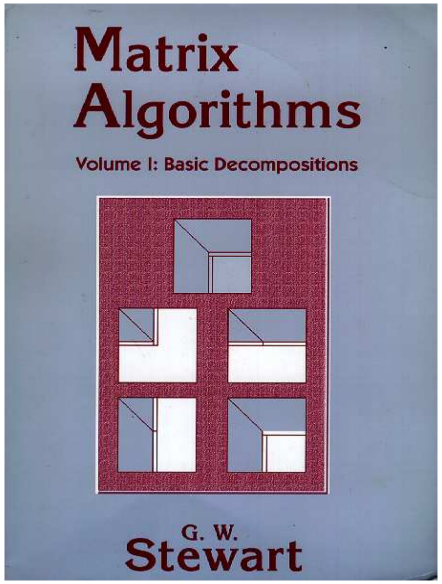
$\{L\backslash U\}_{TL}$	U_{TR}
$\hat{A}_{BL}$	$\hat{A}_{BR}$

Column-lazy

$\{L\backslash U\}_{TL}$	$\hat{A}_{TR}$
L_{BL}	$\hat{A}_{BR}$

Row-Column-lazy

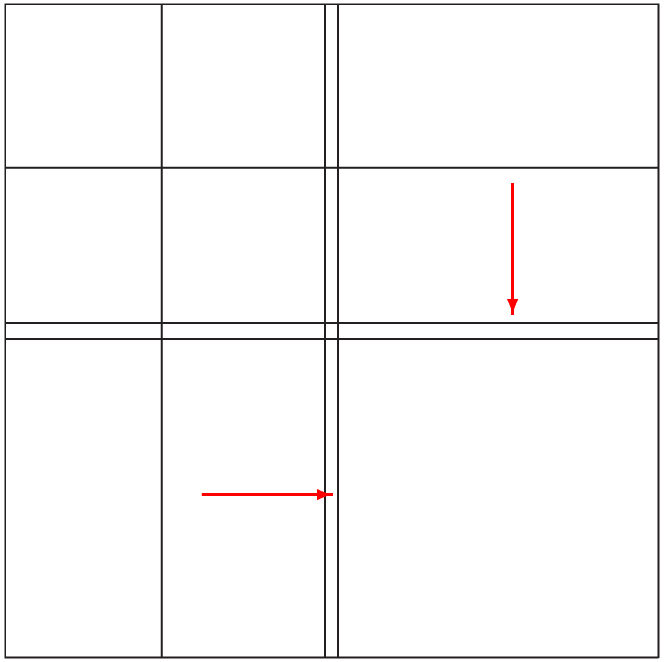
$\{L\backslash U\}_{TL}$	U_{TR}
L_{BL}	$\hat{A}_{BR}$



Step	Annotated Algorithm: $A := L\backslash U = \mathrm{LU}(A)$
1a	$\left\{A = \hat{A}\right\}$
4	**Partition** **where**
2	$\left\{\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)\right\}$
3	**while** G **do**
2,3	$\left\{\left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)\right) \wedge (G)\right\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\left\{\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)\right\}$
	enddo
2,3	$\left\{\left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)\right) \wedge \neg (G)\right\}$
1b	$\left\{A = L\backslash U \wedge \hat{A} = LU)\right\}$

Step	Annotated Algorithm: $A := L\backslash U = \mathsf{LU}(A)$
1a	$\left\{A = \hat{A}\right\}$
4	**Partition** **where**
2	$\left\{\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR}\end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR}\end{array}\right)\right\}$
3	**while** $\neg\mathsf{SameSize}(A_{TL}, A)$ **do**
2,3	$\left\{\left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR}\end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR}\end{array}\right)\right) \wedge (\neg\mathsf{SameSize}(A_{TL}, A))\right\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\left\{\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR}\end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR}\end{array}\right)\right\}$
	enddo
2,3	$\left\{\left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR}\end{array}\right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR}\end{array}\right)\right) \wedge \neg(\neg\mathsf{SameSize}(A_{TL}, A))\right\}$
1b	$\left\{A = L\backslash U \wedge \hat{A} = LU)\right\}$

Step	Annotated Algorithm: $A := L \backslash U = \mathsf{LU}(A)$
1a	$\left\{ A = \hat{A} \right\}$
4	**Partition** $A \rightarrow \left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right)$, $L \rightarrow \left(\begin{array}{c\|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right)$, $U \rightarrow \left(\begin{array}{c\|c} 0 & U_{TR} \\ \hline U_{BL} & U_{BR} \end{array} \right)$, $\hat{A} \rightarrow \left(\begin{array}{c\|c} \hat{A}_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right)$ **where** $A_{TL}, L_{TL}, U_{TL}, \hat{A}_{TL}$ are 0×0
2	$\left\{ \left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$
3	**while** $\neg \mathsf{SameSize}(A_{TL}, A)$ **do**
2,3	$\left\{ \left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge (\neg \mathsf{SameSize}(A_{TL}, A)) \right\}$
5a	**Repartition** **where**
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with**
2	$\left\{ \left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$
	enddo
2,3	$\left\{ \left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge \neg (\neg \mathsf{SameSize}(A_{TL}, A)) \right\}$
1b	$\left\{ A = L \backslash U \wedge \hat{A} = LU) \right\}$



<table>
<tr><th>Step</th><th>Annotated Algorithm: $A := L \backslash U = \mathsf{LU}(A)$</th></tr>
<tr><td>1a</td><td>$\left\{ A = \hat{A} \right\}$</td></tr>
<tr><td>4</td><td>Partition $A \rightarrow \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right)$, $L \rightarrow \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array}\right)$, $U \rightarrow \left(\begin{array}{c||c} 0 & U_{TR} \\ \hline\hline U_{BL} & U_{BR} \end{array}\right)$, $\hat{A} \rightarrow \left(\begin{array}{c||c} \hat{A}_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right)$
where $A_{TL}, L_{TL}, U_{TL}, \hat{A}_{TL}$ are 0×0</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c||c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right) \right\}$</td></tr>
<tr><td>3</td><td>while $\neg \mathsf{SameSize}(A_{TL}, A)$ do</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c||c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right) \right) \wedge (\neg \mathsf{SameSize}(A_{TL}, A)) \right\}$</td></tr>
<tr><td>5a</td><td>Repartition
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c||c|c} L_{00} & 0 & 0 \\ \hline\hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array}\right), \cdots$
where $A_{11}, L_{11}, \ldots$ are $b \times b$</td></tr>
<tr><td>6</td><td>$\{Q_{bu}\}$</td></tr>
<tr><td>8</td><td>S_U</td></tr>
<tr><td>7</td><td>$\{Q_{au}\}$</td></tr>
<tr><td>5b</td><td>Continue with
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array}\right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c||c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline\hline L_{20} & L_{21} & L_{22} \end{array}\right), \cdots$</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c||c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right) \right\}$</td></tr>
<tr><td></td><td>enddo</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) = \left(\begin{array}{c||c} L \backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array}\right) \right) \wedge \neg (\neg \mathsf{SameSize}(A_{TL}, A)) \right\}$</td></tr>
<tr><td>1b</td><td>$\left\{ A = L \backslash U \wedge \hat{A} = LU) \right\}$</td></tr>
</table>

Zooming in on the loop...

Step	Annotated Algorithm: $A := L\backslash U = \mathsf{LU}(A)$
3	**while** $\neg\mathsf{SameSize}(A_{TL}, A)$ **do**
2,3	$\left\{ \left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge (\neg\mathsf{SameSize}(A_{TL}, A)) \right\}$
5a	**Repartition** $\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c\|c\|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c\|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c\|c\|c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$ **where** $A_{11}, L_{11}, \ldots$ are $b \times b$
6	$\{Q_{bu}\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with** $\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c\|c\|c} A_{00} & A_{01} & A_{02} \\ A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c\|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c\|c\|c} L_{00} & 0 & 0 \\ L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$
2	$\left\{ \left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$
	enddo

Step	Annotated Algorithm: $A := L \backslash U = \mathsf{LU}(A)$
3	**while** $\neg\mathsf{SameSize}(A_{TL}, A)$ **do**
2,3	$\left\{ \left(\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge (\neg\mathsf{SameSize}(A_{TL}, A)) \right\}$
5a	**Repartition** $\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c\|c\|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c\|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c\|c\|c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$ **where** $A_{11}, L_{11}, \ldots$ are $b \times b$
6	$\left\{ \left(\begin{array}{c\|c\|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c\|c\|c} L\backslash U_{00} & \hat{A}_{01} & \hat{A}_{02} \\ \hline \hat{A}_{10} & \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{20} & \hat{A}_{21} & \hat{A}_{22} \end{array} \right) \right\}$
8	S_U
7	$\{Q_{au}\}$
5b	**Continue with** $\left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c\|c\|c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c\|c} L_{TL} & 0 \\ \hline L_{BL} & L_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c\|c\|c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$
2	$\left\{ \left(\begin{array}{c\|c} A_{TL} & A_{TR} \\ \hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c\|c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$
	enddo

<table>
<tr><th>Step</th><th>Annotated Algorithm: $A := L\backslash U = \mathsf{LU}(A)$</th></tr>
<tr><td>3</td><td>while $\neg\mathsf{SameSize}(A_{TL}, A)$ do</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c||c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge (\neg\mathsf{SameSize}(A_{TL}, A)) \right\}$</td></tr>
<tr><td>5a</td><td>Repartition
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} L_{00} & 0 & 0 \\ \hline\hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$
where $A_{11}, L_{11}, \ldots$ are $b \times b$</td></tr>
<tr><td>6</td><td>$\left\{ \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c||c|c} L\backslash U_{00} & \hat{A}_{01} & \hat{A}_{02} \\ \hline\hline \hat{A}_{10} & \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{20} & \hat{A}_{21} & \hat{A}_{22} \end{array} \right) \right\}$</td></tr>
<tr><td>8</td><td>S_U</td></tr>
<tr><td>7</td><td>$\left\{ \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c|c||c} L\backslash U_{00} & U_{01} & \hat{A}_{02} \\ \hline L_{10} & L\backslash U_{11} & \hat{A}_{12} \\ \hline\hline \hat{A}_{20} & \hat{A}_{21} & \hat{A}_{22} \end{array} \right) \right\}$</td></tr>
<tr><td>5b</td><td>Continue with
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline\hline L_{20} & L_{21} & L_{22} \end{array} \right), \cdots$</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c||c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$</td></tr>
<tr><td></td><td>enddo</td></tr>
</table>

<table>
<tr><th>Step</th><th>Annotated Algorithm: $A := L\backslash U = \mathrm{LU}(A)$</th></tr>
<tr><td>3</td><td>while $\neg\mathrm{SameSize}(A_{TL}, A)$ do</td></tr>
<tr><td>2,3</td><td>$\left\{ \left(\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c||c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right) \wedge (\neg\mathrm{SameSize}(A_{TL}, A)) \right\}$</td></tr>
<tr><td>5a</td><td>Repartition
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} L_{00} & 0 & 0 \\ \hline\hline L_{10} & L_{11} & 0 \\ \hline L_{20} & L_{21} & L_{22} \end{array} \right), \ldots$
where $A_{11}, L_{11}, \ldots$ are $b \times b$</td></tr>
<tr><td>6</td><td>$\left\{ \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c||c|c} L\backslash U_{00} & \hat{A}_{01} & \hat{A}_{02} \\ \hline\hline \hat{A}_{10} & \hat{A}_{11} & \hat{A}_{12} \\ \hline \hat{A}_{20} & \hat{A}_{21} & \hat{A}_{22} \end{array} \right) \right\}$</td></tr>
<tr><td>8</td><td>$A_{10} := L_{10} = A_{10} U_{00}^{-1}$
$A_{01} := U_{01} = L_{00}^{-1} A_{01}$
$A_{11} := L\backslash U_{11} = \mathrm{LU}(A_{11} - L_{10} U_{01})$</td></tr>
<tr><td>7</td><td>$\left\{ \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right) = \left(\begin{array}{c|c||c} L\backslash U_{00} & U_{01} & \hat{A}_{02} \\ \hline L_{10} & L\backslash U_{11} & \hat{A}_{12} \\ \hline\hline \hat{A}_{20} & \hat{A}_{21} & \hat{A}_{22} \end{array} \right) \right\}$</td></tr>
<tr><td>5b</td><td>Continue with
$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right), \left(\begin{array}{c||c} L_{TL} & 0 \\ \hline\hline L_{BL} & L_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} L_{00} & 0 & 0 \\ \hline L_{10} & L_{11} & 0 \\ \hline\hline L_{20} & L_{21} & L_{22} \end{array} \right), \ldots$</td></tr>
<tr><td>2</td><td>$\left\{ \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) = \left(\begin{array}{c||c} L\backslash U_{TL} & \hat{A}_{TR} \\ \hline\hline \hat{A}_{BL} & \hat{A}_{BR} \end{array} \right) \right\}$</td></tr>
<tr><td></td><td>enddo</td></tr>
</table>

Lazy Algorithm (bordered variant)

Partition $A \rightarrow \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right)$

where A_{TL} is 0×0

while $\neg\mathsf{SameSize}(A_{TL}, A)$ **do**

Repartition

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right)$$

where A_{11} is $b \times b$

$A_{10} := L_{10} = A_{10}U_{00}^{-1}$
$A_{01} := U_{01} = L_{00}^{-1}A_{01}$
$A_{11} := L\backslash U_{11} = \mathsf{LU}(A_{11} - L_{10}U_{01})$

Continue with

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right)$$

enddo

Eager Algorithm

Partition $A \rightarrow \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right)$

where A_{TL} is 0×0

while $\neg\mathsf{SameSize}(A_{TL}, A)$ **do**

Repartition

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

where A_{11} is $b \times b$

$A_{10} := L_{10} = A_{10} U_{00}^{-1}$
$A_{11} := L \backslash U_{11} = \mathsf{LU}(A_{11} - L_{10} U_{01})$
$A_{12} := U_{12} = L_{11}^{-1}(A_{12} - L_{10} U_{02})$

Continue with

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array}\right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array}\right)$$

enddo

Impact on Implementation of Linear Algebra Algorithms

Capturing the Algorithm in Code

Partition $A \rightarrow \left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right)$

where A_{TL} is 0×0

while $\neg$SameSize(A_{TL}, A) **do**

Repartition

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \rightarrow \left(\begin{array}{c||c|c} A_{00} & A_{01} & A_{02} \\ \hline\hline A_{10} & A_{11} & A_{12} \\ \hline A_{20} & A_{21} & A_{22} \end{array} \right)$$

where A_{11} is $b \times b$

Continue with

$$\left(\begin{array}{c||c} A_{TL} & A_{TR} \\ \hline\hline A_{BL} & A_{BR} \end{array} \right) \leftarrow \left(\begin{array}{c|c||c} A_{00} & A_{01} & A_{02} \\ \hline A_{10} & A_{11} & A_{12} \\ \hline\hline A_{20} & A_{21} & A_{22} \end{array} \right)$$

enddo

C code

```
FLA_Part_2x2( A,  &ATL, /**/ &ATR,
                /* ************** */
                  &ABL, /**/ &ABR, /* with */ 0, /* by */ 0, /* submatrix */ FLA_TL );

while ( b = min( FLA_Obj_length( ABR ), nb_alg ) ){

  FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,         &A00, /**/ &A01, &A02,
                      /* ************* */    /* ******************** */
                              /**/              &A10, /**/ &A11, &A12,
                         ABL, /**/ ABR,         &A20, /**/ &A21, &A22,
               /* with */ b, /* by */ b, /* A11 split from */ FLA_BR );

  /* ************************************************************************************************** */







  /* ************************************************************************************************** */

  FLA_Cont_3x3_to_2x2( &ATL, /**/ &ATR,         A00, A01, /**/ A02,
                             /**/               A10, A11, /**/ A12,
                     /* ************** */    /* ***************** */
                       &ABL, /**/ &ABR,         A20, A21, /**/ A22,
               /* with A11 added to submatrix */ FLA_TL );

}
```

Lazy LU in C

```
FLA_Part_2x2( A,  &ATL, /**/ &ATR,
                /* ************** */
                  &ABL, /**/ &ABR, /* with */ 0, /* by */ 0, /* submatrix */ FLA_TL );

while ( b = min( FLA_Obj_length( ABR ), nb_alg ) ){

  FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,         &A00, /**/ &A01, &A02,
                      /* ************* */    /* ******************** */
                              /**/              &A10, /**/ &A11, &A12,
                         ABL, /**/ ABR,         &A20, /**/ &A21, &A22,
               /* with */ b, /* by */ b, /* A11 split from */ FLA_BR );

  /* ****************************************************************************************************** */

  FLA_Trsm( FLA_SIDE_RIGHT, FLA_UPPER_TRIANGULAR, FLA_NO_TRANSPOSE, FLA_NONUNIT_DIAG, ONE, A00, A10 );
  FLA_Trsm( FLA_SIDE_LEFT,  FLA_LOWER_TRIANGULAR, FLA_NO_TRANSPOSE, FLA_UNIT_DIAG,    ONE, A00, A01 );
  FLA_Gemm( FLA_NO_TRANSPOSE, FLA_NO_TRANSPOSE, MINUS_ONE, A10, A01, ONE, A11 );

  FLA_LU_nopivot_level2( A11 );

  /* ****************************************************************************************************** */

  FLA_Cont_3x3_to_2x2( &ATL, /**/ &ATR,         A00, A01, /**/ A02,
                             /**/               A10, A11, /**/ A12,
                    /* ************** */    /* ***************** */
                       &ABL, /**/ &ABR,         A20, A21, /**/ A22,
               /* with A11 added to submatrix */ FLA_TL );

}
```

Eager LU in C

```
FLA_Part_2x2( A,  &ATL, /**/ &ATR,
                /* ************** */
                  &ABL, /**/ &ABR, /* with */ 0, /* by */ 0, /* submatrix */ FLA_TL );

while ( b = min( FLA_Obj_length( ABR ), nb_alg ) ){

  FLA_Repart_2x2_to_3x3( ATL, /**/ ATR,         &A00, /**/ &A01, &A02,
                      /* ************* */    /* ******************** */
                              /**/              &A10, /**/ &A11, &A12,
                         ABL, /**/ ABR,         &A20, /**/ &A21, &A22,
               /* with */ b, /* by */ b, /* A11 split from */ FLA_BR );

  /* ***************************************************************************************************** */

  FLA_LU_nopivot_level2( A11 );

  FLA_Trsm( FLA_SIDE_RIGHT, FLA_UPPER_TRIANGULAR, FLA_NO_TRANSPOSE, FLA_NONUNIT_DIAG, ONE, A11, A21 );
  FLA_Trsm( FLA_SIDE_LEFT,  FLA_LOWER_TRIANGULAR, FLA_NO_TRANSPOSE, FLA_UNIT_DIAG,    ONE, A11, A12 );
  FLA_Gemm( FLA_NO_TRANSPOSE, FLA_NO_TRANSPOSE, MINUS_ONE, A21, A12, ONE, A22 );

  /* ***************************************************************************************************** */

  FLA_Cont_3x3_to_2x2( &ATL, /**/ &ATR,         A00, A01, /**/ A02,
                             /**/               A10, A11, /**/ A12,
                     /* ************** */    /* ***************** */
                       &ABL, /**/ &ABR,         A20, A21, /**/ A22,
               /* with A11 added to submatrix */ FLA_TL );

}
```

But what about pivoting?

Yes, pivoting can be handled elegantly.

- John A. Gunnels, Fred G. Gustavson, Greg M. Henry, and Robert A. van de Geijn. FLAME: Formal Linear Algebra Methods Environment. *ACM Transactions on Mathematical Software*, Dec. 2001.

Parallel Eager LU in C

```
PLA_Obj_partition_2x2( A,  &ATL, /**/ &ATR,
                         /* ************** */
                           &ABL, /**/ &ABR, /* with */ 0, /* by */ 0, /* submatrix */ PLA_TL );

while ( b = min( PLA_Obj_length( ABR ), nb_alg ) ){

  PLA_Obj_repartition_2x2_to_3x3( ATL, /**/ ATR,         &A00, /**/ &A01, &A02,
                             /* ************* */    /* ******************** */
                                     /**/              &A10, /**/ &A11, &A12,
                                ABL, /**/ ABR,         &A20, /**/ &A21, &A22,
                /* with */ b, /* by */ b, /* A11 split from */ PLA_BR );

  /* ****************************************************************************************** */

  PLA_LU_nopivot_level2( A11 );

  PLA_Trsm( PLA_SIDE_RIGHT, PLA_UPPER_TRIANGULAR, PLA_NO_TRANSPOSE, PLA_NONUNIT_DIAG, ONE, A11, A21 );
  PLA_Trsm( PLA_SIDE_LEFT,  PLA_LOWER_TRIANGULAR, PLA_NO_TRANSPOSE, PLA_UNIT_DIAG,    ONE, A11, A12 );
  PLA_Gemm( PLA_NO_TRANSPOSE, PLA_NO_TRANSPOSE, MINUS_ONE, A21, A12, ONE, A22 );

  /* ****************************************************************************************** */

  PLA_Obj_continue_with_3x3_to_2x2( &ATL, /**/ &ATR,         A00, A01, /**/ A02,
                                        /**/                 A10, A11, /**/ A12,
                                /* ************** */    /* ***************** */
                                  &ABL, /**/ &ABR,         A20, A21, /**/ A22,
                /* with A11 added to submatrix */ PLA_TL );

}
```

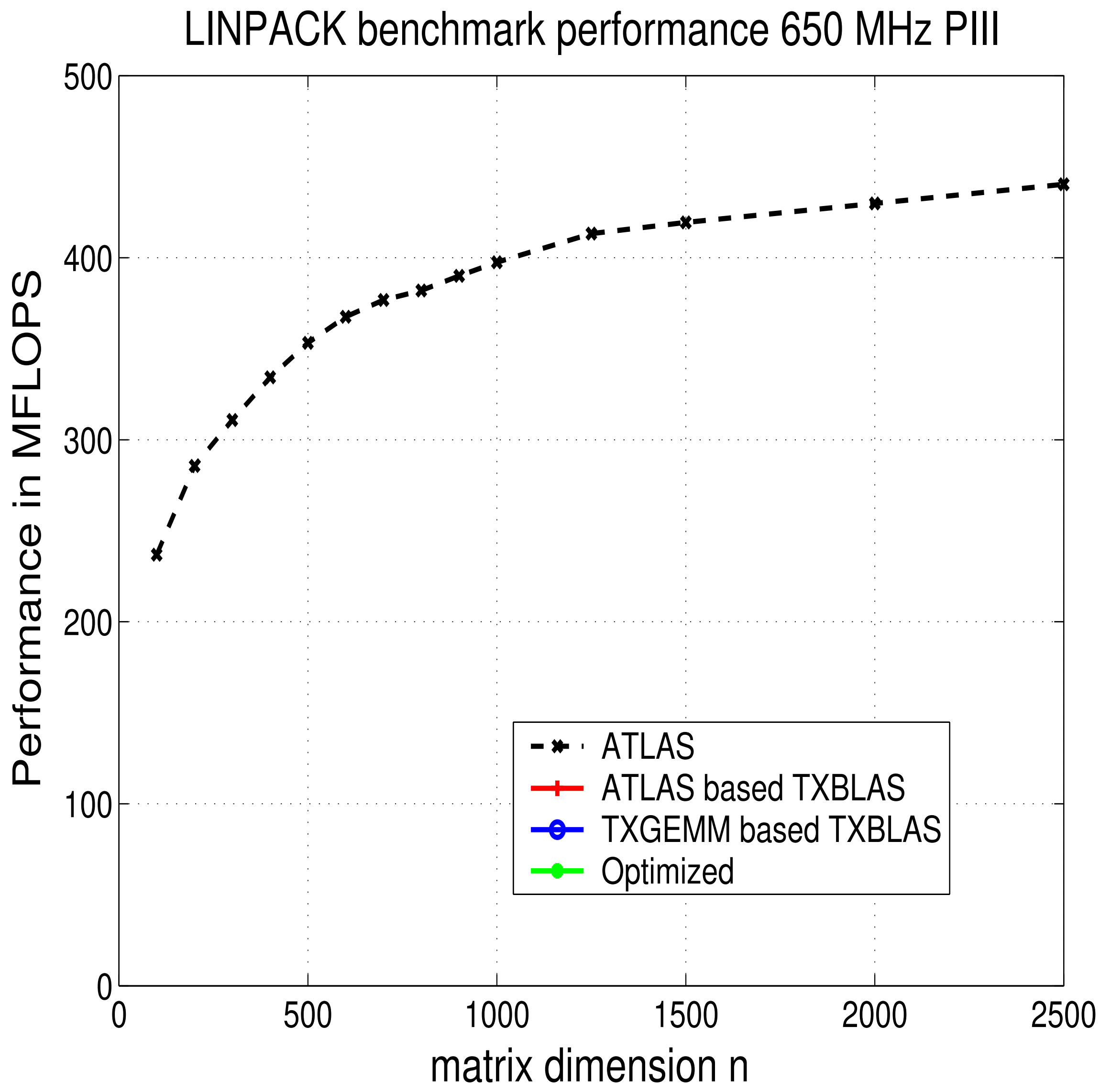
Parallel Out-of-Core Eager LU in C

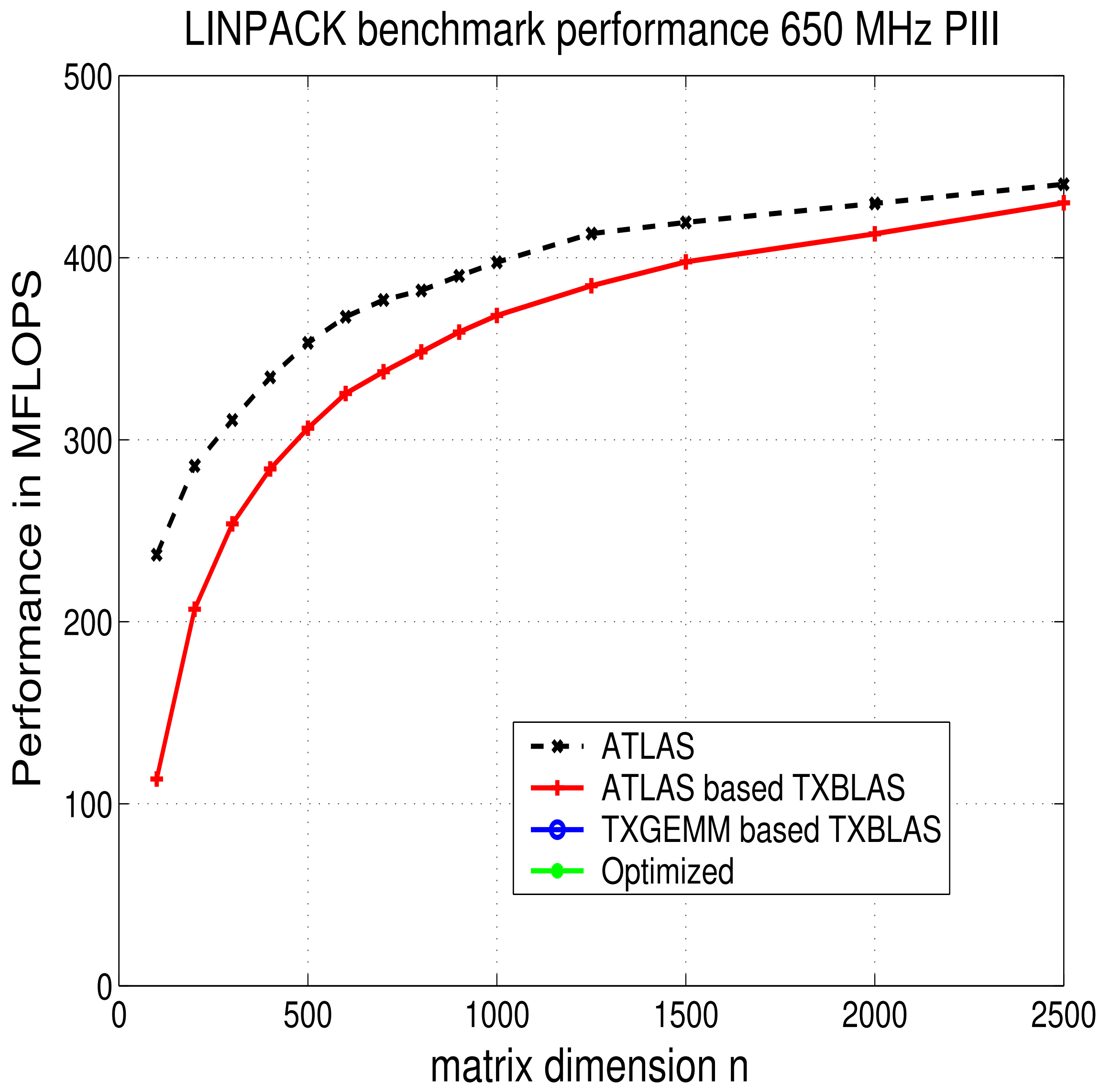
```
POOCLA_Obj_partition_2x2( A,  &ATL, /**/ &ATR,
                            /* ************** */
                              &ABL, /**/ &ABR, /* with */ 0, /* by */ 0, /* submatrix */ PLA_TL );

while ( b = min( PLA_Obj_length( ABR ), nb_alg ) ){

  POOCLA_Obj_repartition_2x2_to_3x3( ATL, /**/ ATR,         &A00, /**/ &A01, &A02,
                                   /* ************* */    /* ******************** */
                                           /**/              &A10, /**/ &A11, &A12,
                                      ABL, /**/ ABR,         &A20, /**/ &A21, &A22,
                /* with */ b, /* by */ b, /* A11 split from */ PLA_BR );

  /* ****************************************************************************************** */

  POOCLA_LU_nopivot_level2( A11 );

  POOCLA_Trsm( PLA_SIDE_RIGHT, PLA_UPPER_TRIANGULAR, PLA_NO_TRANSPOSE, PLA_NONUNIT_DIAG, ONE, A11, A21 );
  POOCLA_Trsm( PLA_SIDE_LEFT,  PLA_LOWER_TRIANGULAR, PLA_NO_TRANSPOSE, PLA_UNIT_DIAG,    ONE, A11, A12 );
  POOCLA_Gemm( PLA_NO_TRANSPOSE, PLA_NO_TRANSPOSE, MINUS_ONE, A21, A12, ONE, A22 );

  /* ****************************************************************************************** */

  POOCLA_Obj_continue_with_3x3_to_2x2( &ATL, /**/ &ATR,         A00, A01, /**/ A02,
                                              /**/               A10, A11, /**/ A12,
                                      /* ************** */    /* ***************** */
                                        &ABL, /**/ &ABR,         A20, A21, /**/ A22,
                /* with A11 added to submatrix */ PLA_TL );

}
```

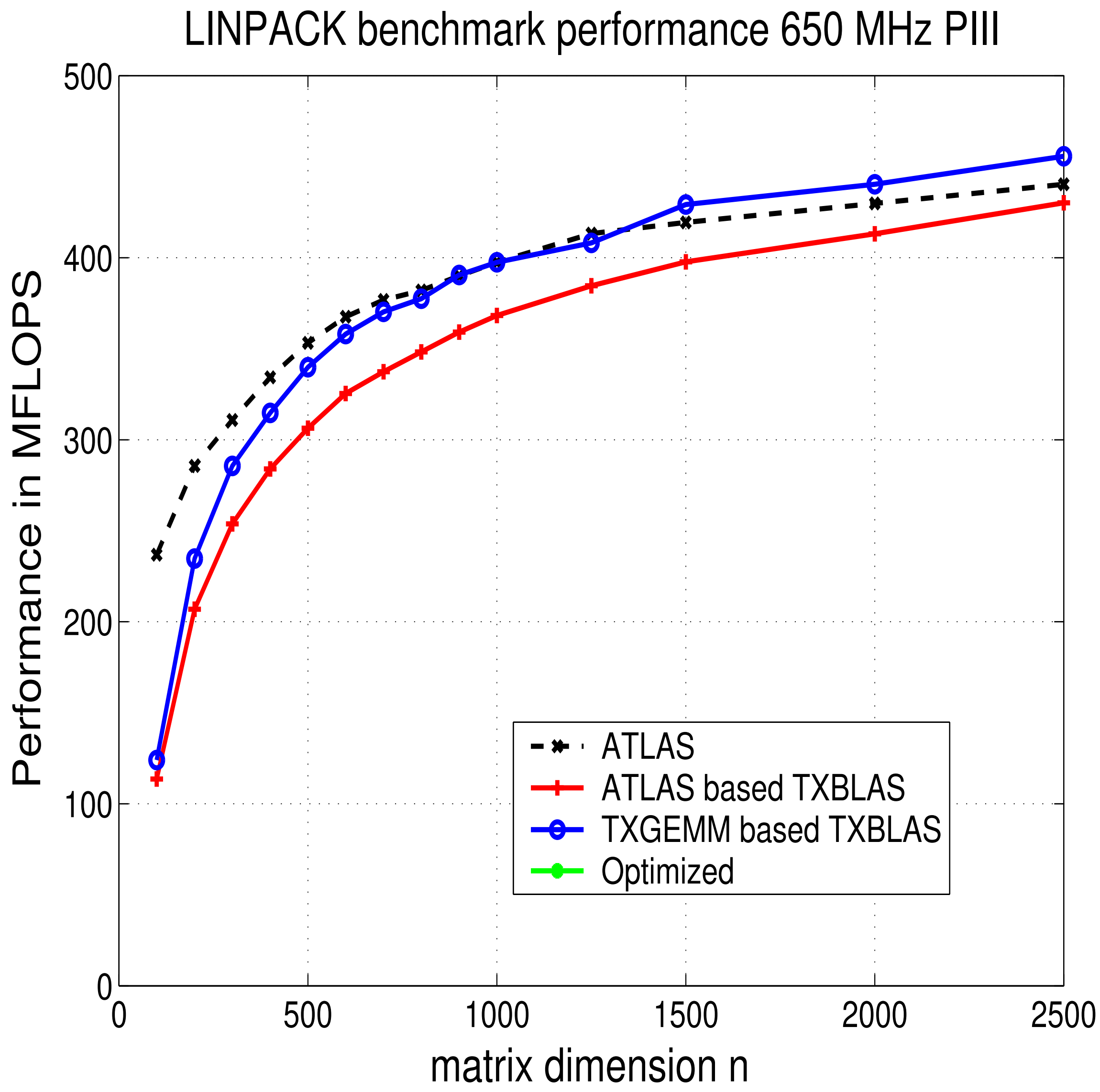
Impact on Performance

LINPACK Benchmark (UNOFFICIAL)

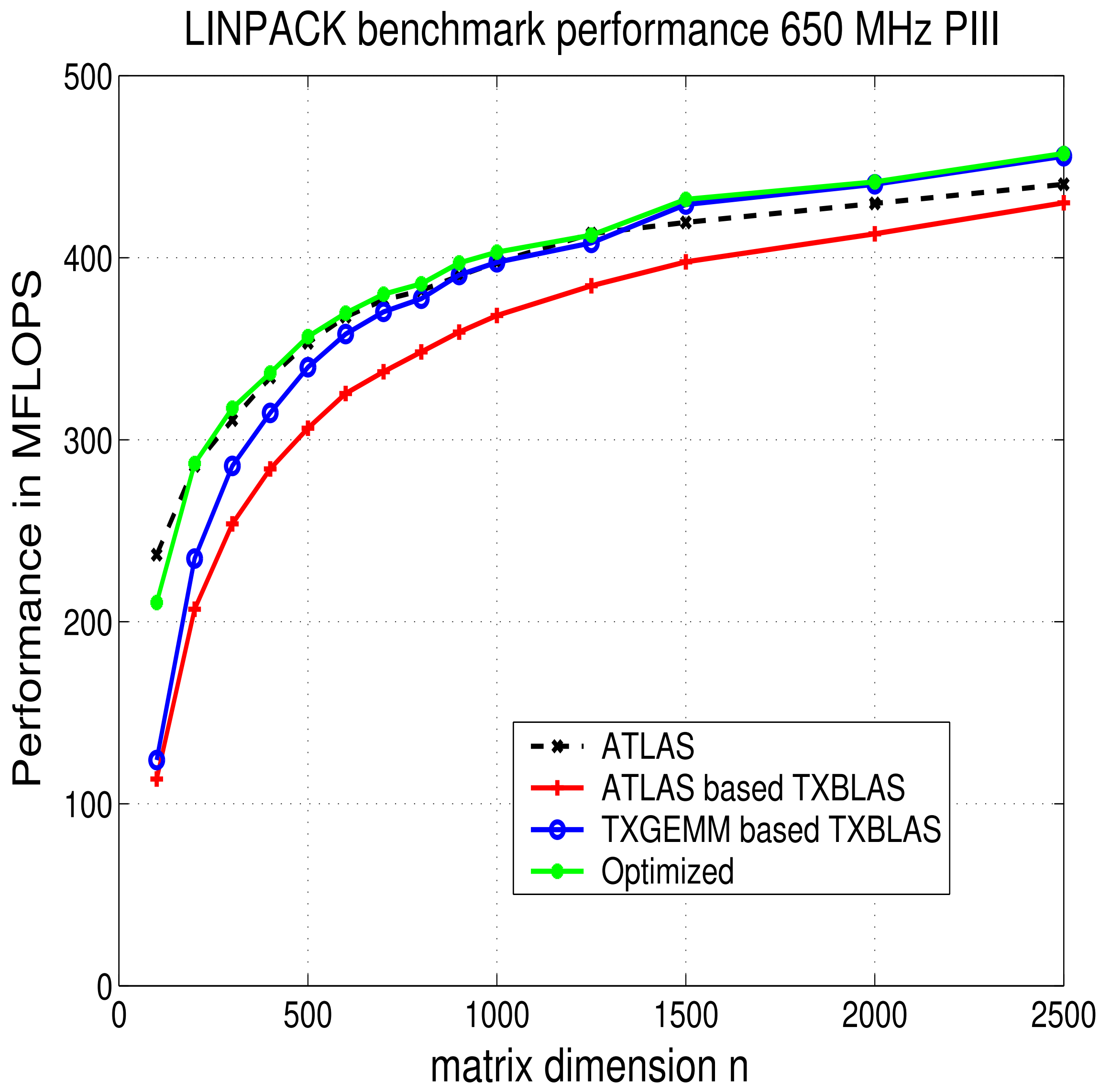
- LU factorization with pivoting followed by forward and backward substitution.
- Double precision real
- Pentium III (650 MHz)
- Linux (Redhat 6.2)
- Reference performance: LU from ATLAS R3.1
- Peak achieved for matrix-matrix multiply:
  - ⋆ Around 490 MFLOPS for ATLAS matrix-matrix multiply


LINPACK benchmark performance 650 MHz PIII
Performance in MFLOPS
matrix dimension n
0
100
200
300
400
500
0
500
1000
1500
2000
2500
ATLAS
ATLAS based TXBLAS
TXGEMM based TXBLAS
Optimized


LINPACK benchmark performance 650 MHz PIII
Performance in MFLOPS
matrix dimension n
0
100
200
300
400
500
0
500
1000
1500
2000
2500
ATLAS
ATLAS based TXBLAS
TXGEMM based TXBLAS
Optimized


LINPACK benchmark performance 650 MHz PIII
Performance in MFLOPS
matrix dimension n
0
100
200
300
400
500
0
500
1000
1500
2000
2500
ATLAS
ATLAS based TXBLAS
TXGEMM based TXBLAS
Optimized

Impact on Performance



Impact on Library Development and Maintainability

Impact on productivity

- Problem: How to measure impact on productivity.
- Solution: Implementation by RvdG of the level-3 BLAS

$C \leftarrow \alpha AB + \beta C$ $C \leftarrow \alpha AB^T + \beta C$	$C \leftarrow \alpha A^T B + \beta C$ $C \leftarrow \alpha A^T B^T + \beta C$
$C \leftarrow \alpha(L + \hat{L}^T)B + \beta C$ $C \leftarrow \alpha B(L + \hat{L}^T) + \beta C$	$C \leftarrow \alpha(U + \hat{U}^T)B + \beta C$ $C \leftarrow \alpha B(U + \hat{U}^T) + \beta C$
$lo(C) \leftarrow \alpha \mathrm{lo}(AA^T) + \beta \mathrm{lo}(C)$ $lo(C) \leftarrow \alpha \mathrm{lo}(A^T A) + \beta \mathrm{lo}(C)$	$up(C) \leftarrow \alpha \mathrm{up}(AA^T) + \beta \mathrm{up}(C)$ $up(C) \leftarrow \alpha \mathrm{up}(A^T A) + \beta \mathrm{up}(C)$
$lo(C) \leftarrow \alpha \mathrm{lo}(AB^T + BA^T) + \beta \mathrm{lo}(C)$ $lo(C) \leftarrow \alpha \mathrm{lo}(A^T B + B^T A) + \beta \mathrm{lo}(C)$	$up(C) \leftarrow \alpha \mathrm{up}(AB^T + BA^T) + \beta \mathrm{up}(C)$ $up(C) \leftarrow \alpha \mathrm{up}(A^T B + B^T A) + \beta \mathrm{up}(C)$
$B \leftarrow \alpha LB$ $B \leftarrow \alpha UB$ $B \leftarrow \alpha BL$ $B \leftarrow \alpha BU$	$B \leftarrow \alpha L^T B$ $B \leftarrow \alpha U^T B$ $B \leftarrow \alpha BL^T$ $B \leftarrow \alpha BU^T$
$B \leftarrow \alpha L^{-1} B$ $B \leftarrow \alpha U^{-1} B$ $B \leftarrow \alpha BL^{-1}$ $B \leftarrow \alpha BU^{-1}$	$B \leftarrow \alpha L^{-T} B$ $B \leftarrow \alpha U^{-T} B$ $B \leftarrow \alpha BL^{-T}$ $B \leftarrow \alpha BU^{-T}$

10 hours including testing.

- John A. Gunnels, Fred G. Gustavson, Greg M. Henry, and Robert A. van de Geijn. FLAME: Formal Linear Algebra Methods Environment. *ACM Transactions on Mathematical Software*, Dec. 2001.

- A Recipe
- Simplicity: Linear-algebra-challenged undergraduates.
- Automating Derivation: Mathematica based system
- Automatic Code/Cost Generation: Produce parallel libraries automatically.
- Solving the Triangular Sylvester Equation
- One-Sweep Matrix Inversion Algorithm
- Parallel Out-of-Core implementation of matrix factorization algorithms
- Fault-Tolerant Linear Algebra Libraries
- High-performance matrix-matrix multiply
- Sparse Iterative Methods: Krylov
- Sparse Direct Matrix Operations: Hierarchical

Conclusion

Summary

- There is a place for formal derivation in scientific computing
- Direct benefits:
  - ⋆ Coverage
  - ⋆ Correctness
  - ⋆ Productivity
  - ⋆ Automation
- Indirect benefits:
  - ⋆ A consistent classification of linear algebra algorithms
  - ⋆ A consistent template for expressing algorithms
  - ⋆ A consistent template for coding algorithms
  - ⋆ Key to high performance (Surprise!)

More information

- http://www.cs.utexas.edu/users/plapack/
- http://www.cs.utexas.edu/users/flame/
- http://www.cs.utexas.edu/users/flame/ITXGEMM/