Restructuring the Symmetric QR Algorithm for Performance

Field Van Zee
Gregorio Quintana-Orti
Robert van de Geijn

THE UNIVERSITY OF TEXAS AT AUSTIN

For details:

Field Van Zee, Robert van de Geijn, and Gregorio Quintana-Orti. “Restructuring the QR algorithm for Performance.” *ACM TOMS.* Accepted (pending minor modifications)

This work was supported by

- UTAustin-Portugal Colab program
- Microsoft
- NSF under grants OCI-0850750, CCF-0917167, and OCI-1148125

Any opinions, findings and conclusions or recommendations expressed in this materialare those of the author(s) and do not necessarily reflect the views of the NationalScience Foundation (NSF).

Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

Symmetric EVD/SVD: 50+ Years of Progress

- Recent progress focuses a lot on the mathematics side
  - Divide & Conquer (Cuppen's) algorithm (D&C)
  - Method of Relatively Robust Representations (MRRR)
- Occasional revisit of Jacobi's method
- Progress on QR has been for non-symmetric problem.
  - Aggressive Early Deflation
  - Multishift

Two Insights

- **WHEN COMPUTING THE DENSE EVD (all eigenvalues and vectors)**, D&C and MRRR have hidden $O(n^3)$ cost
- QR becomes competitive if rotations are applied in batches
  - Classical QR: cast in terms of vector-vector operations
  - Batched application: cast in terms of computation that reuses data in cache, like matrix-matrix operations.

Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

The Hidden Cost of D&C and MRRR

- Start with symmetric, dense *A*
- Reduce to tridiagonal form:

$$A \longrightarrow Q_A T Q_A^T$$

- Compute Spectral Decomposition of *T*:

$$T \longrightarrow Q_T D Q_T^T$$

- Backtransform:

$$A = \underbrace{Q_A Q_T}_{Q} \; D \; \underbrace{(Q_A Q_T)^T}_{Q^T} = Q D Q^T$$

Reduction to Tridiagonal Form

$$A \longrightarrow Q_A T Q_A^T$$

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Reduction to Tridiagonal Form

$$A \longrightarrow Q_A T Q_A^T$$

$$H_{2:5} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} H_{2:5}$$

$$= \begin{pmatrix} \times & \times & & & \\ \times & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{pmatrix}$$

Reduction to Tridiagonal Form

$$A \longrightarrow Q_A T Q_A^T$$

$$H_{3:5} H_{2:5} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} H_{2:5} H_{3:5}$$

$$=$$

$$\begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{pmatrix}$$

Reduction to Tridiagonal Form

$$A \longrightarrow Q_A T Q_A^T$$

$$H_{4:5} H_{3:5} H_{2:5} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} H_{2:5} H_{3:5} H_{4:5}.$$

$$= \begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

Backtransformation

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Backtransformation

$$H_{4:5}\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Backtransformation

$$H_{3:5} H_{4:5} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Backtransformation

$$H_{2:5}H_{3:5}H_{4:5}\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix}$$

Cost of QR algorithm

- Start with symmetric, dense *A*
- Reduce to tridiagonal form:

$$A \longrightarrow Q_A T Q_A^T$$

- Form Q_A
- Compute Spectral Decomposition of *T* while updating Q_A

$$Q_A T Q_A^T \longrightarrow (Q_A Q_T) D (Q_A Q_T)^T$$

Form Q_A

$$\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

Form Q_A

$$H_{4:5} \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} & & & & \\ & & & & \\ & & & & \\ & & & \times & \times \\ & & & \times & \times \end{pmatrix}$$

Form Q_A

$$H_{3:5} H_{4:5} \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} & & & & \\ & & & & \\ & & \times & \times & \times \\ & & \times & \times & \times \\ & & \times & \times & \times \end{pmatrix}$$

Form Q_A

$$H_{2:5}H_{3:5}H_{4:5}\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} & & & & \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{pmatrix}$$

Form Q_A

$$H_{2:5}H_{3:5}H_{4:5}\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

$$=$$

$$\begin{pmatrix} 1 & & & & \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & \times & \times & \times \end{pmatrix}$$

Cost

- Backtransformation: $2\ n^3$ flops
- Form Q_A: $4/3\ n^3$ flops
- Hidden cost of MRRR and D&C: $2/3\ n^3$ flops

EVD OF A DENSE MATRIX!!!
(All eigenvalues and eigenvectors)

Overview

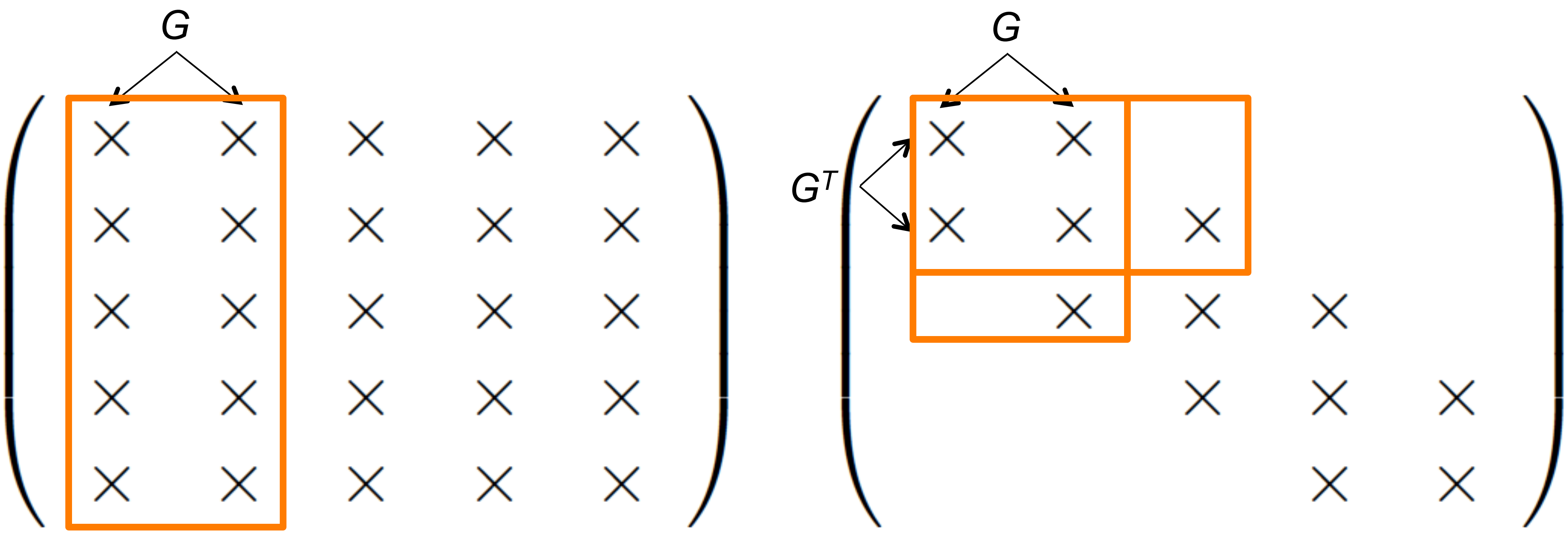
- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

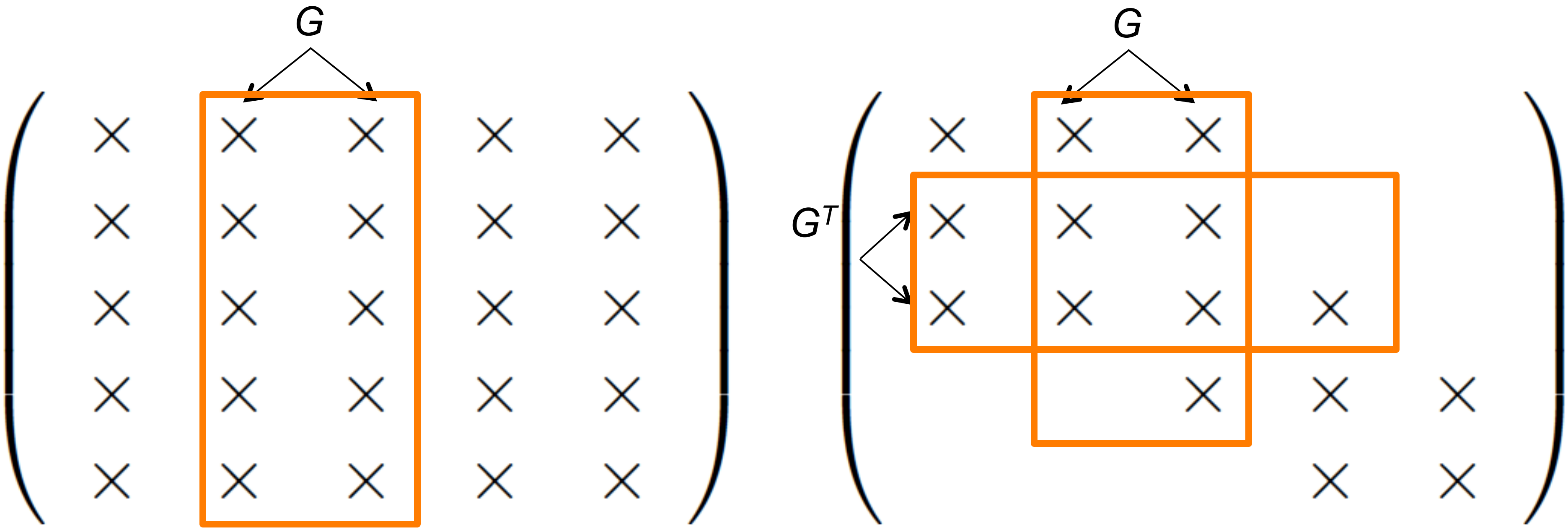
Classical QR algorithm

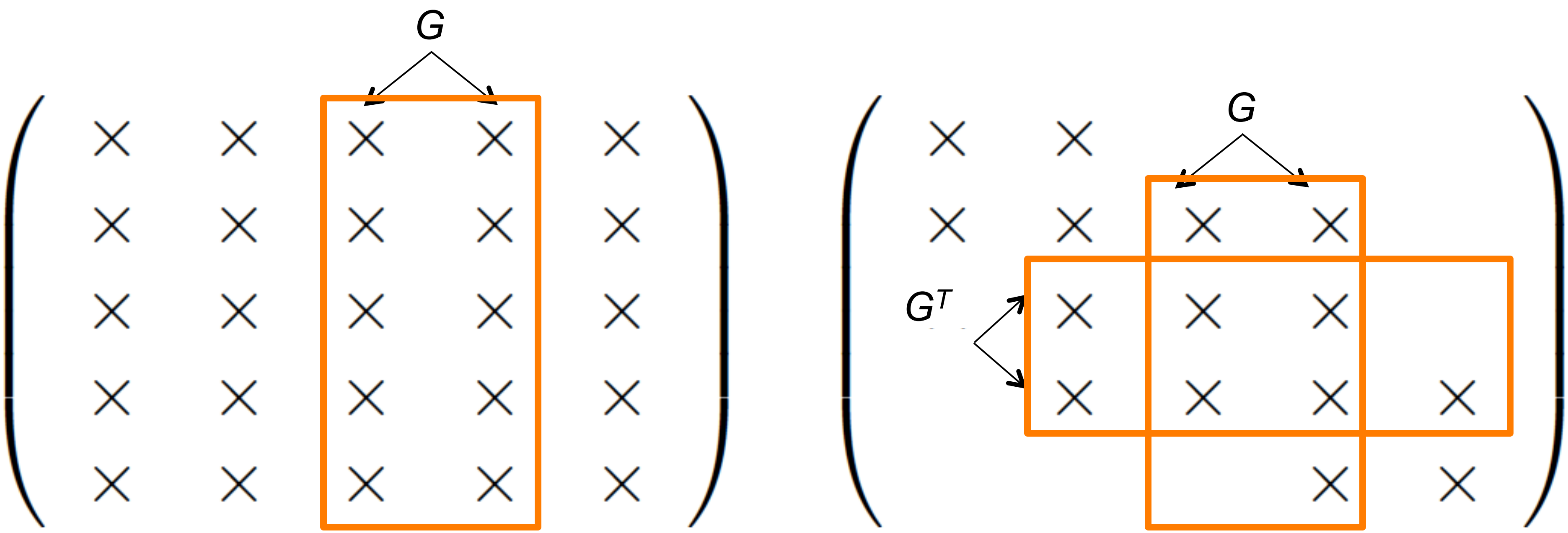
$$Q_A \qquad\qquad\qquad T$$

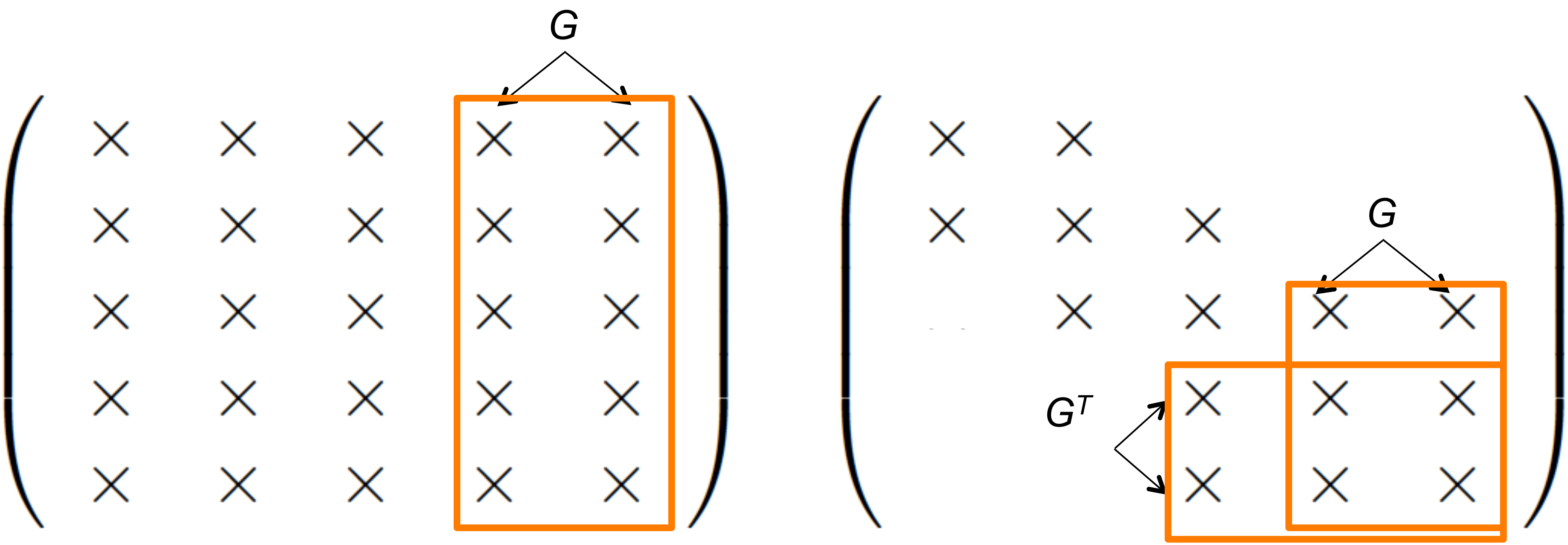
$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \quad \begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

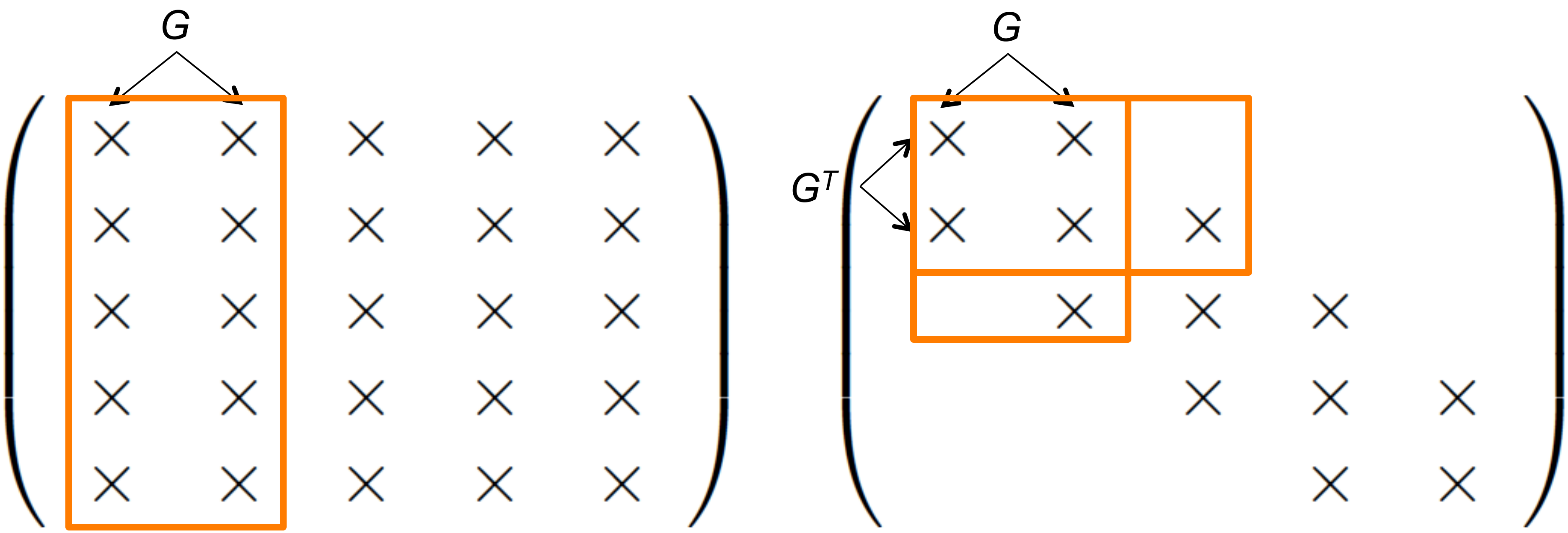
$$\begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} \quad \begin{pmatrix} \times & \times & & & \\ \times & \times & \times & & \\ & \times & \times & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{pmatrix}$$

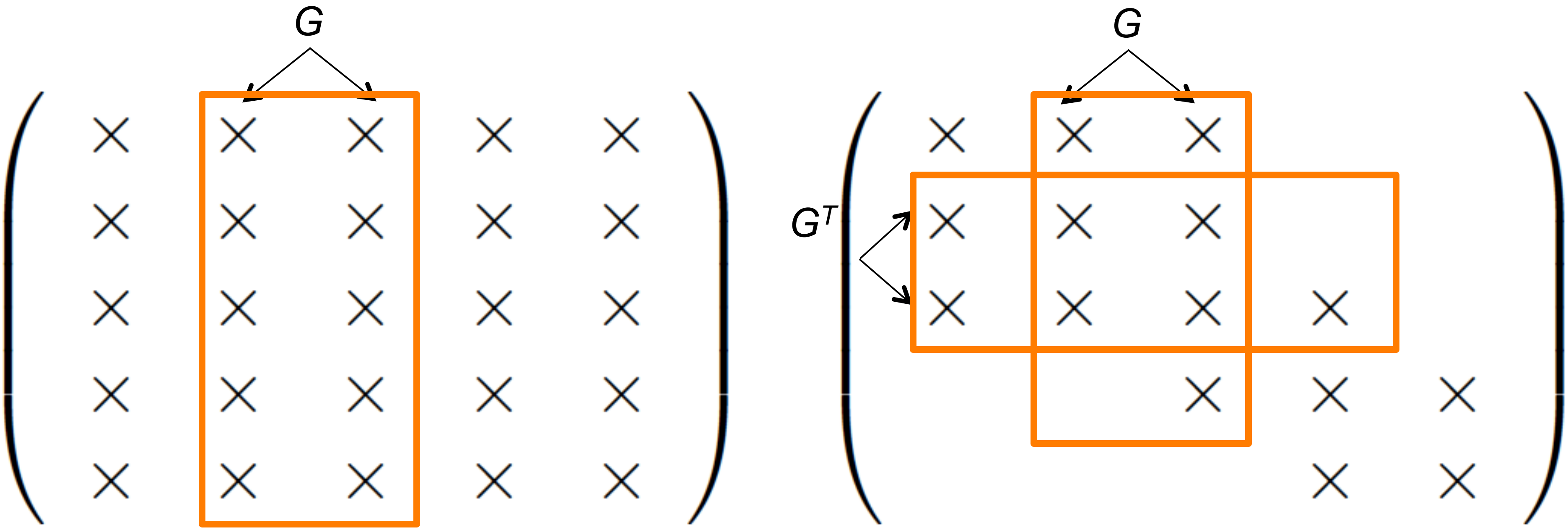

G
G
G^T

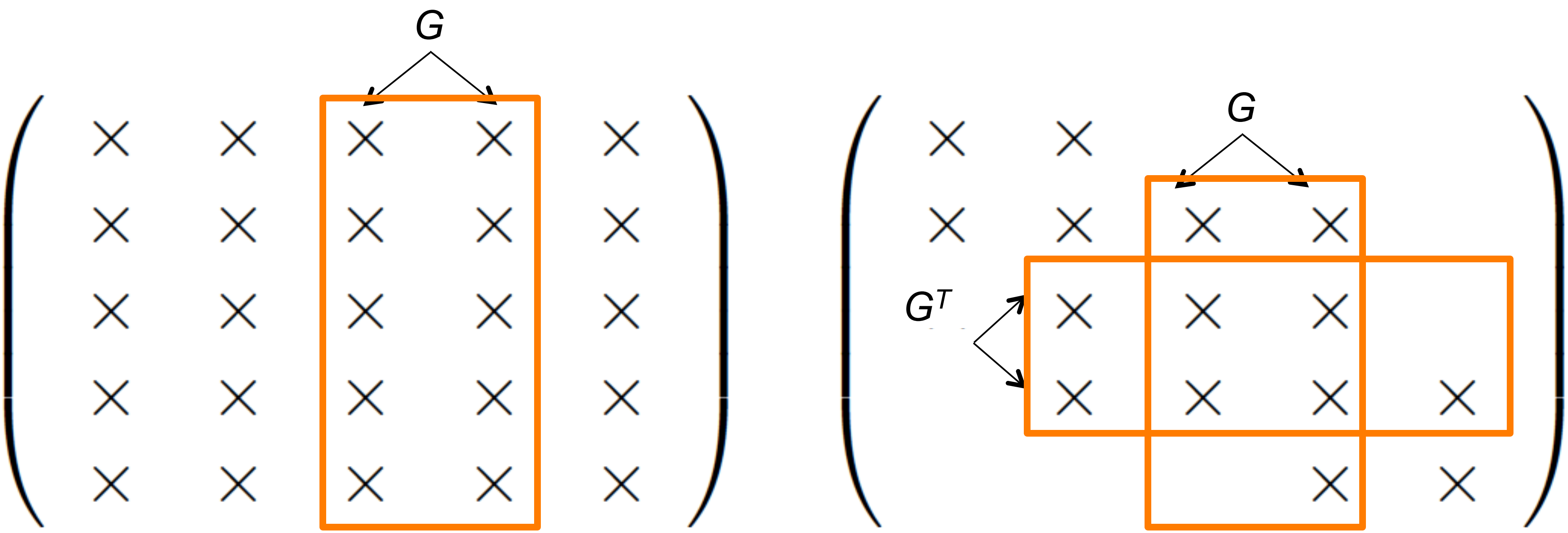

G
G
G^T

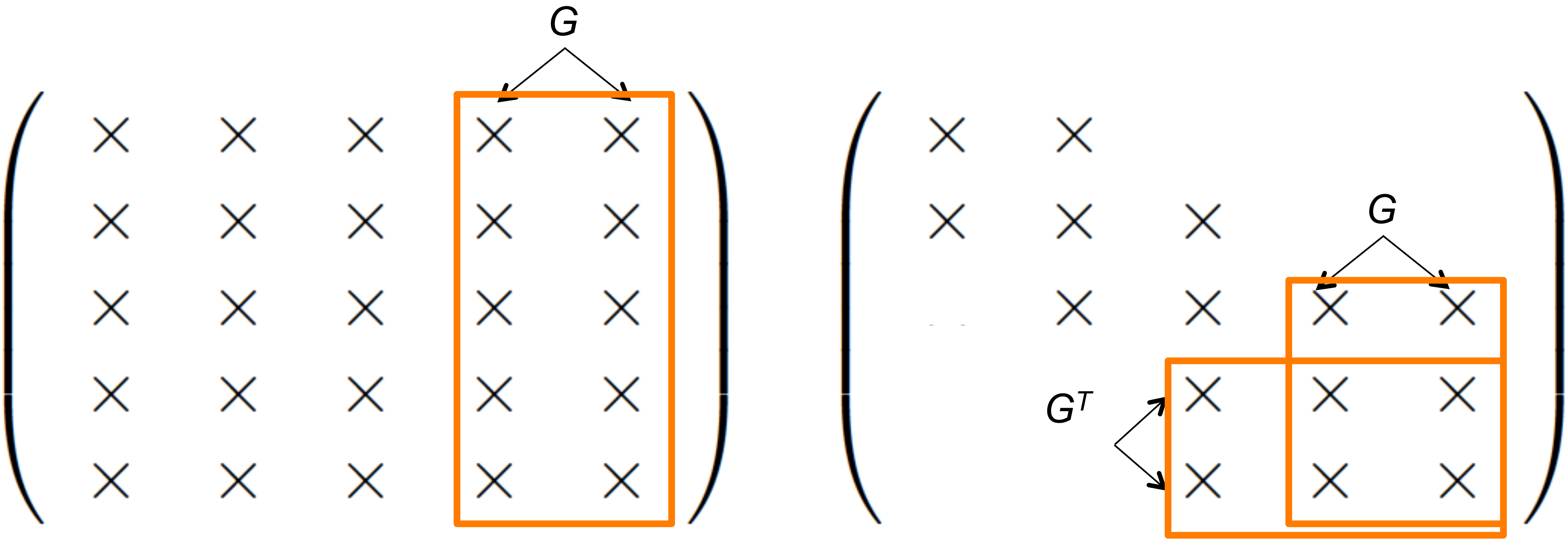

G
G
G^T

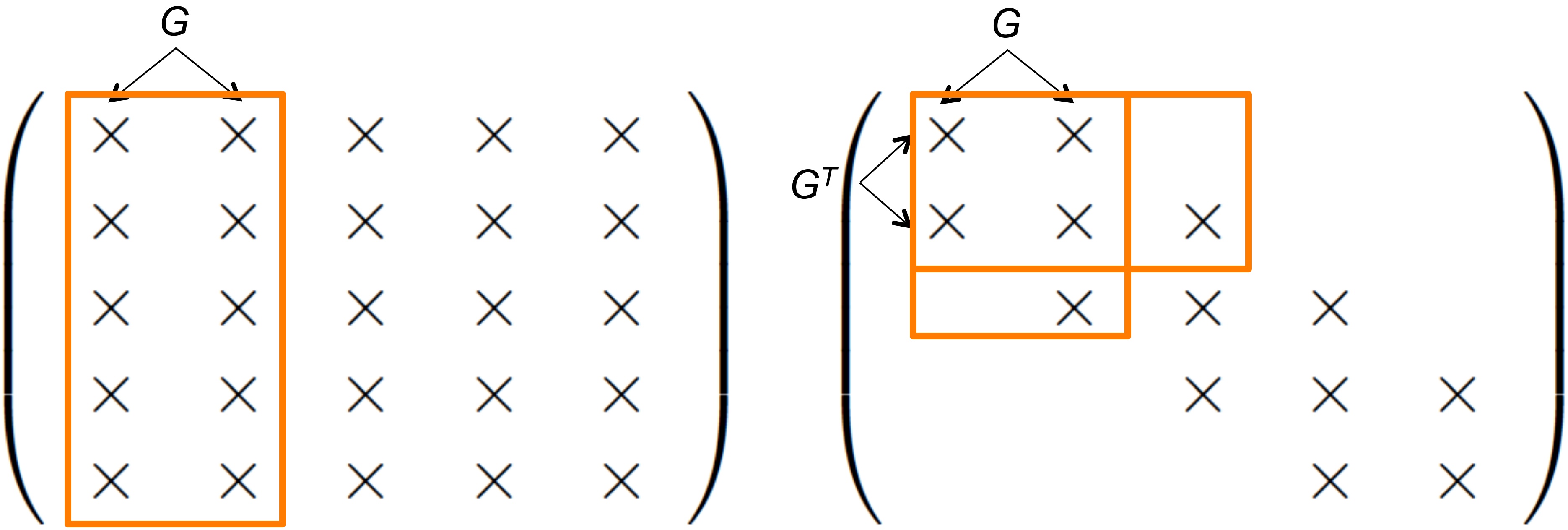

G
G
G^T


G
G
G^T

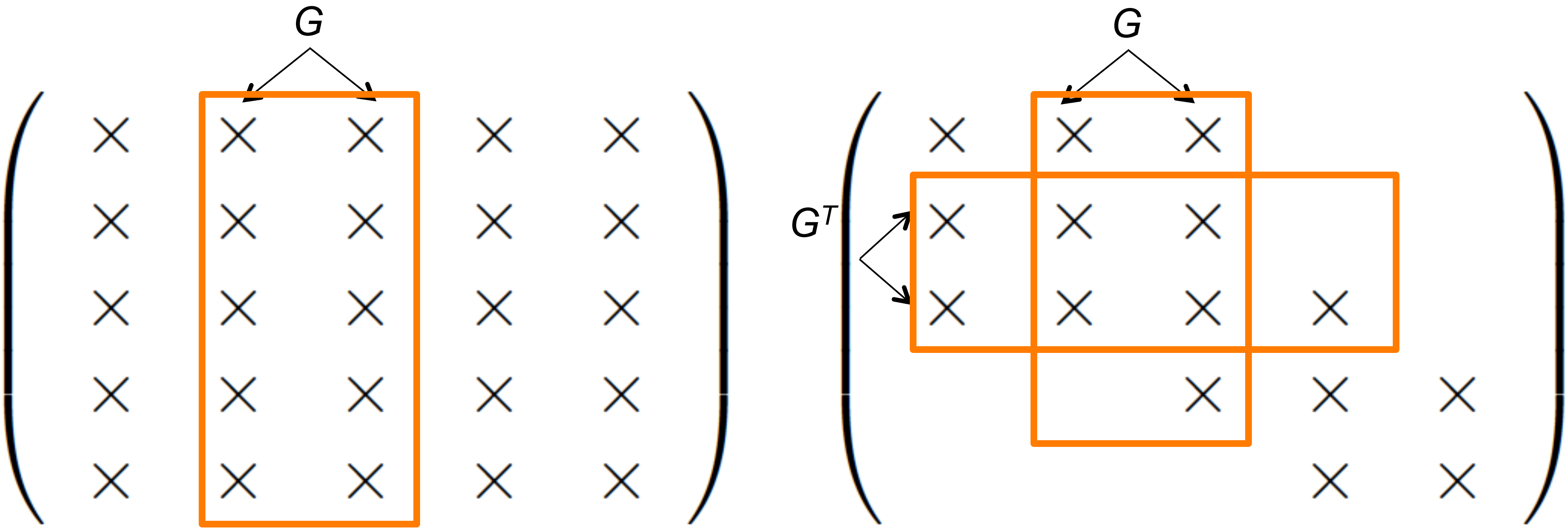

G
G
G^T

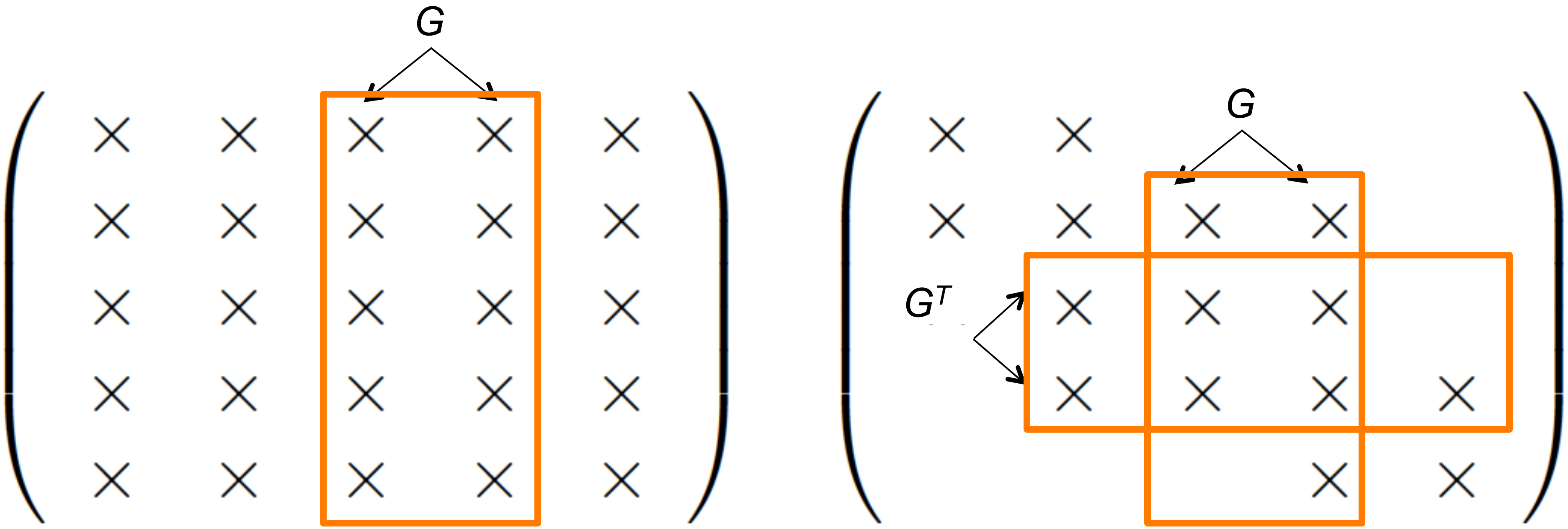

G
G
G^T

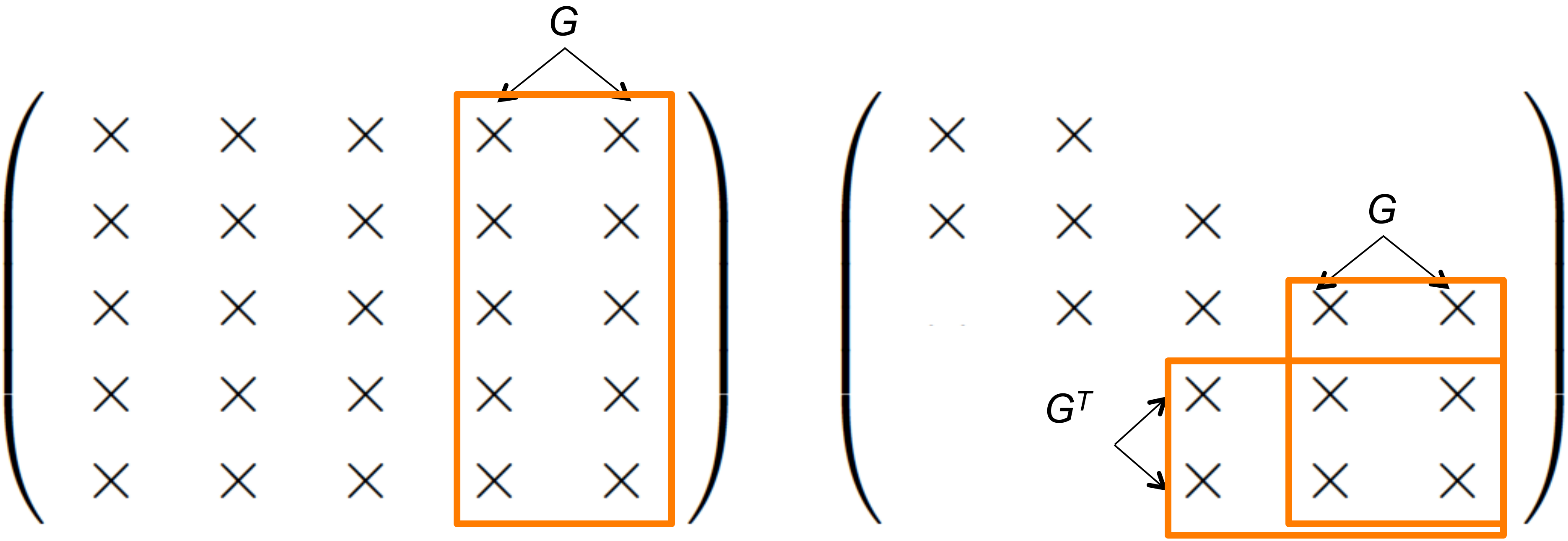

G
G
G^T



G
G
G^T


G
G
G^T

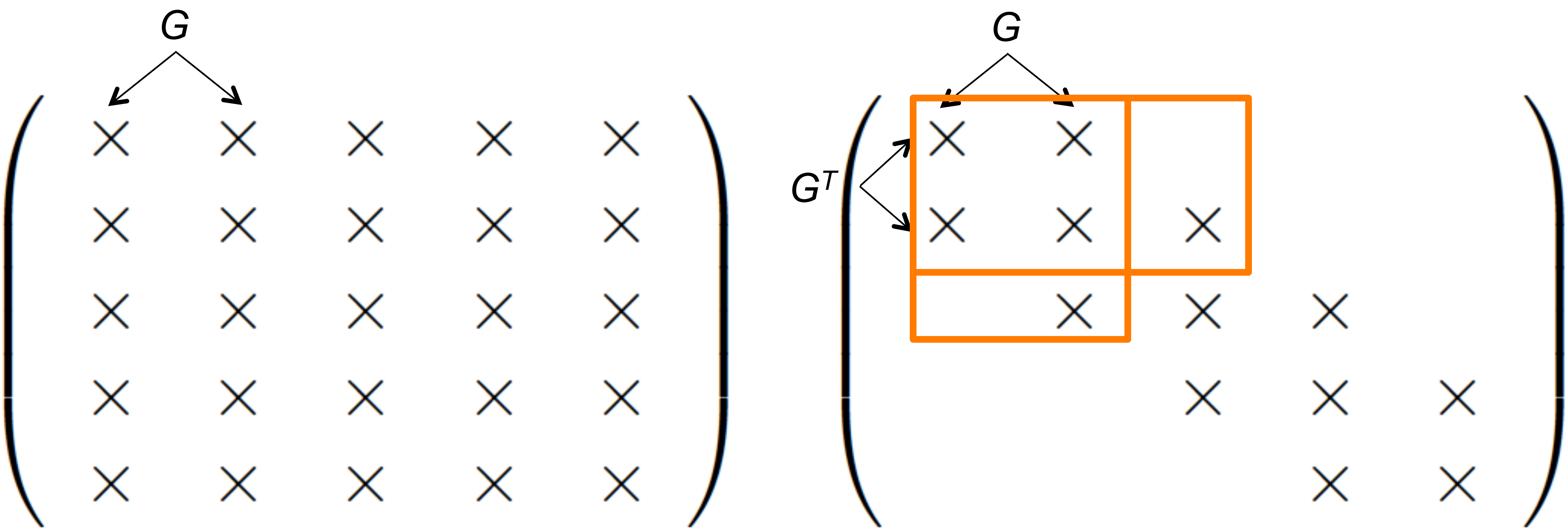

G
G
G^T


G
G
G^T

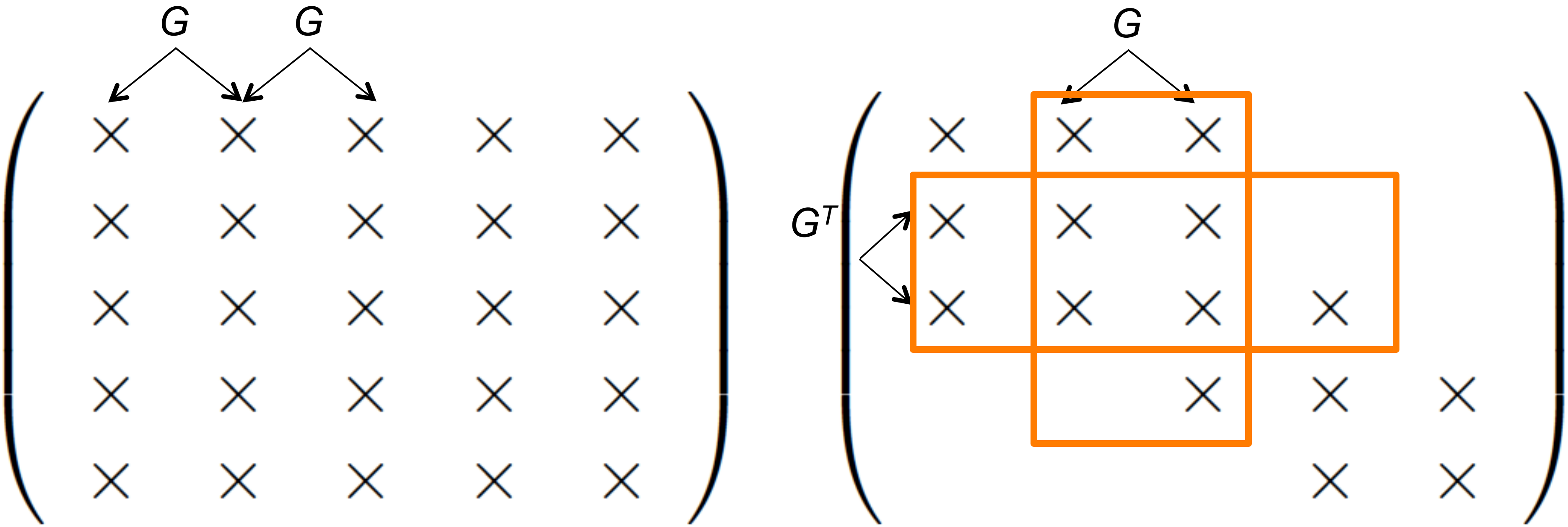
Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

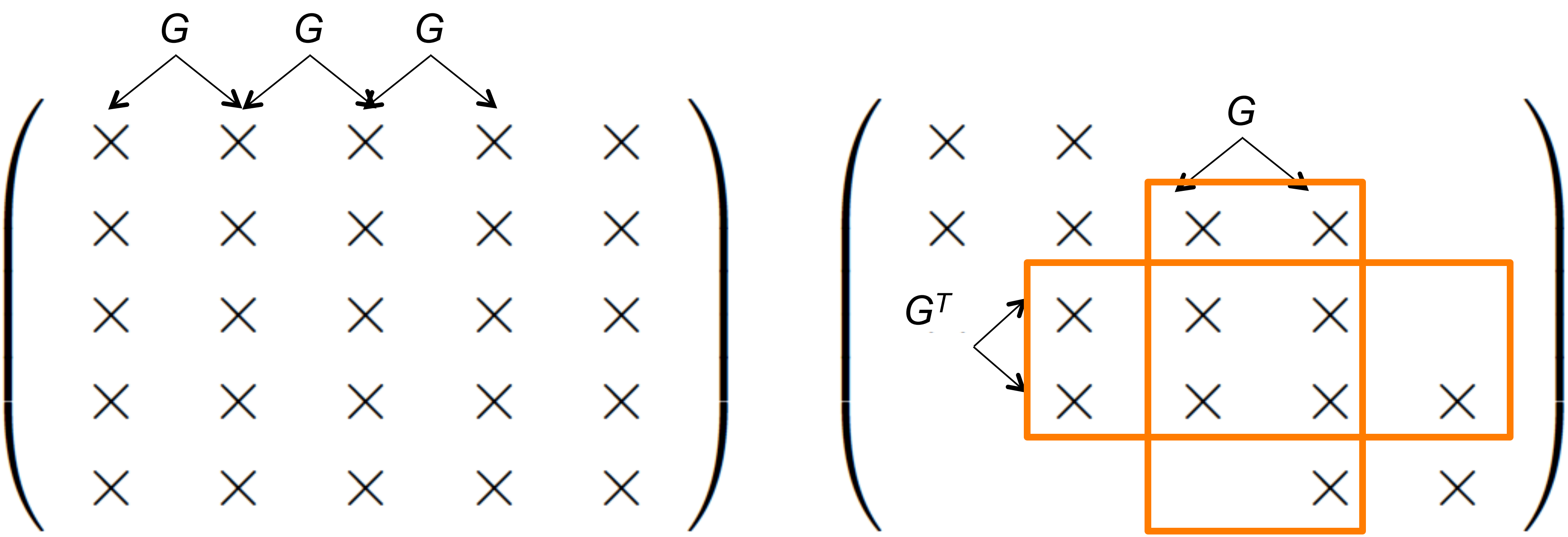
Accumulating Rotations (LAPACK)



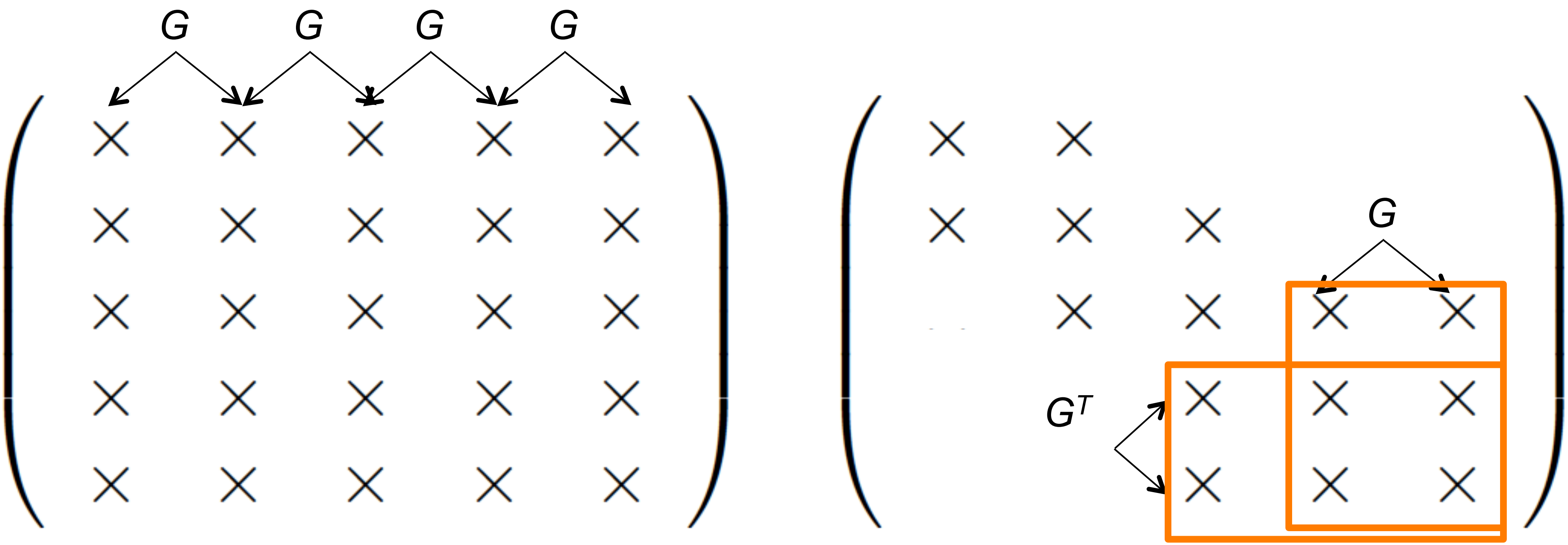
Accumulating Rotations (LAPACK)



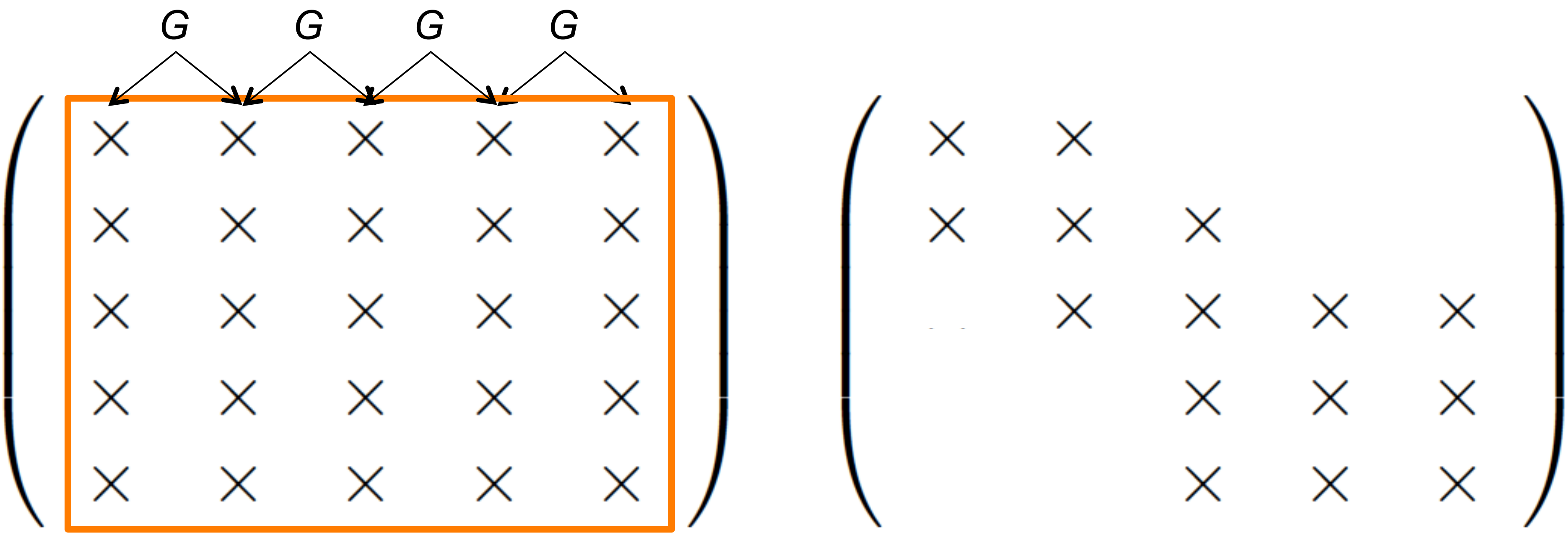
Accumulating Rotations (LAPACK)



Accumulating Rotations (LAPACK)

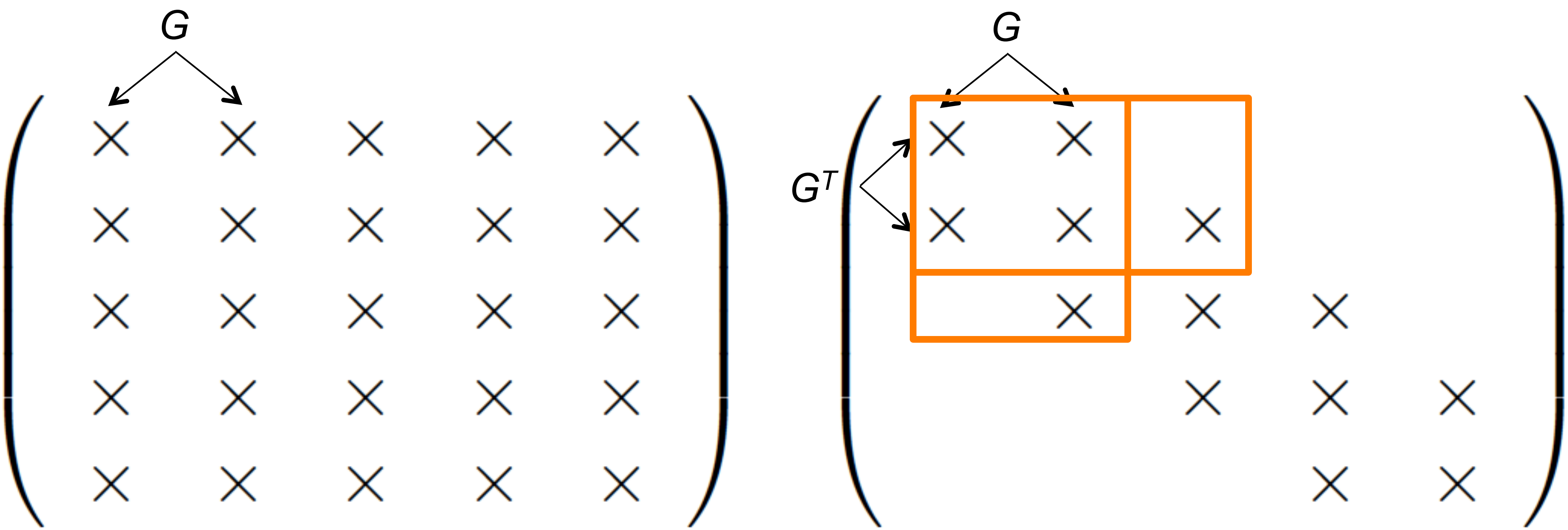


Accumulating Rotations (LAPACK)

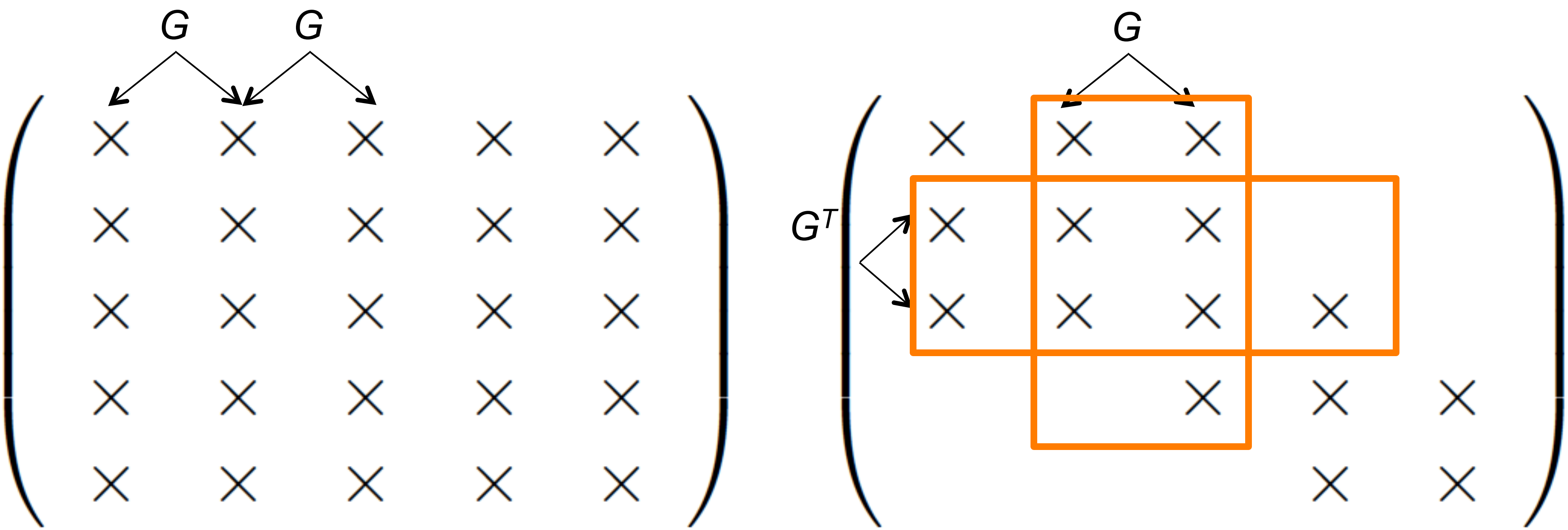


- Apply one sweep worth of rotations.
- Makes application like “level-2 BLAS”

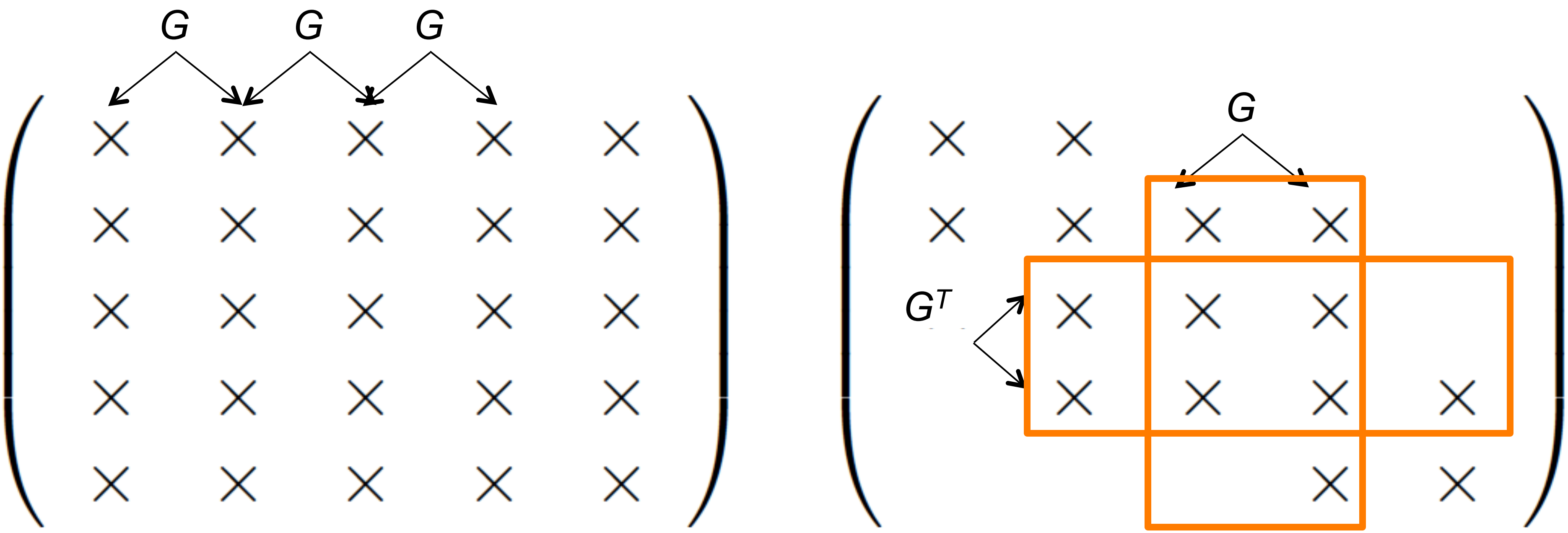
Accumulating Rotations (libflame)



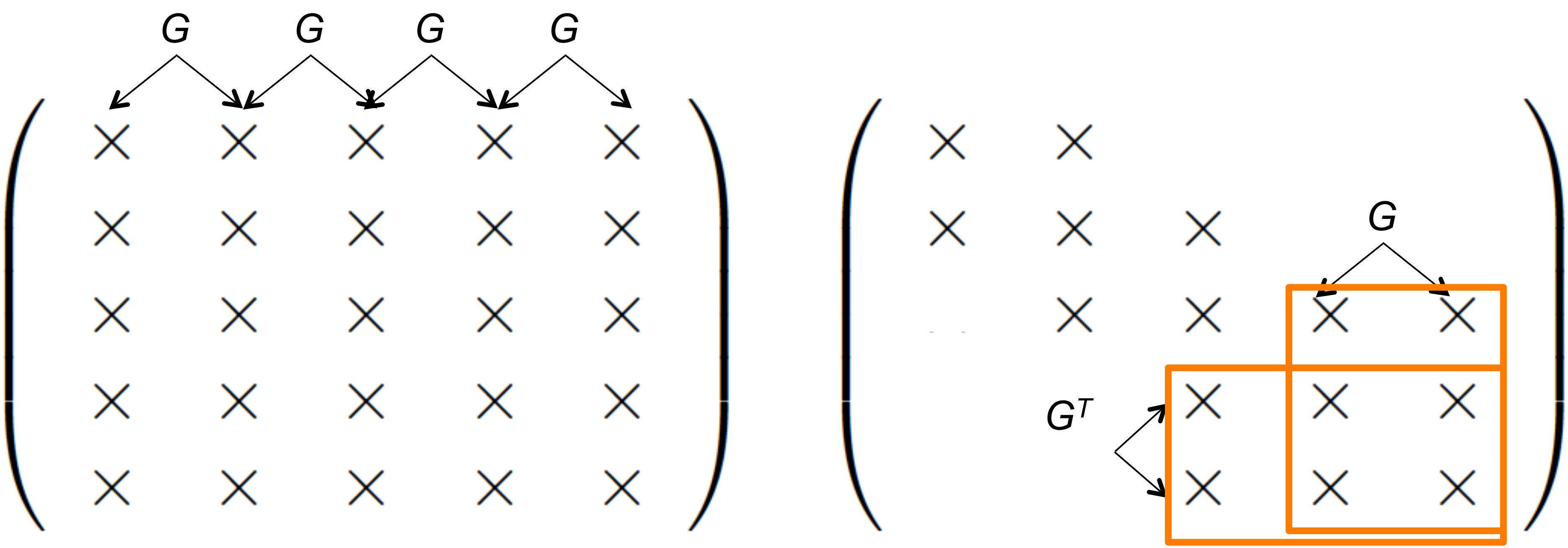
Accumulating Rotations (libflame)



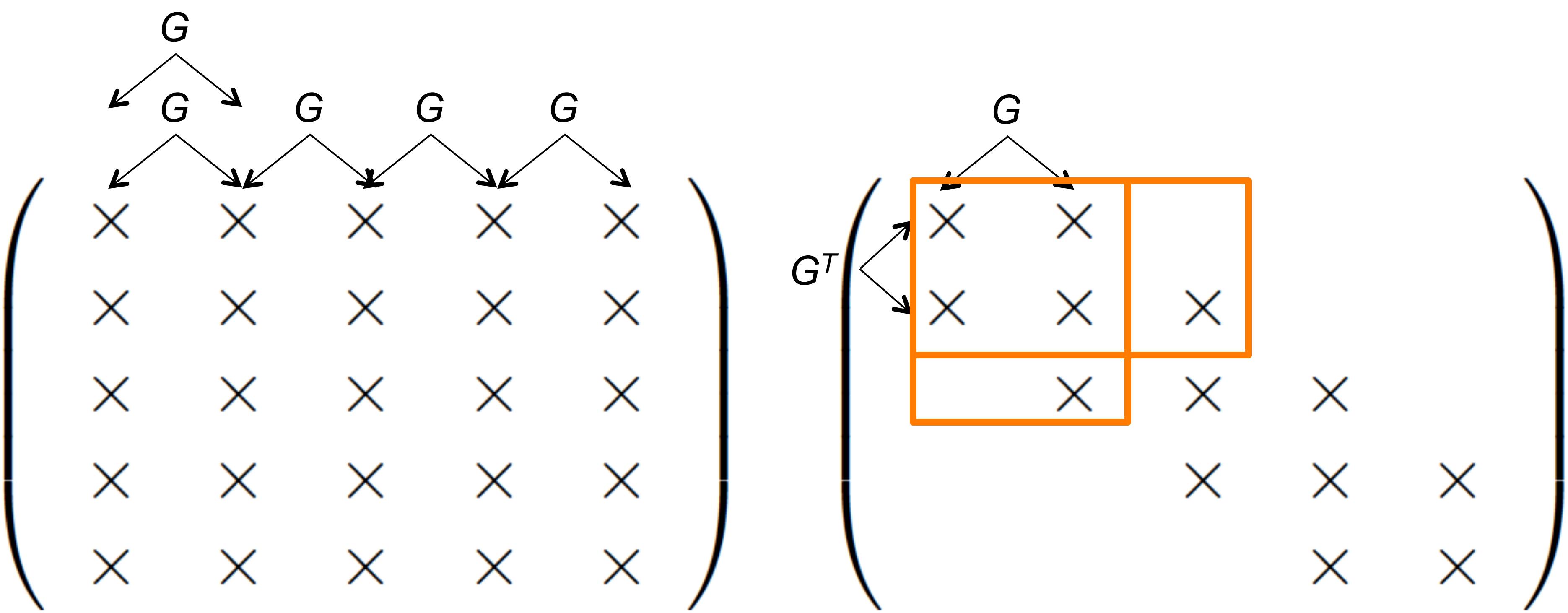
Accumulating Rotations (libflame)



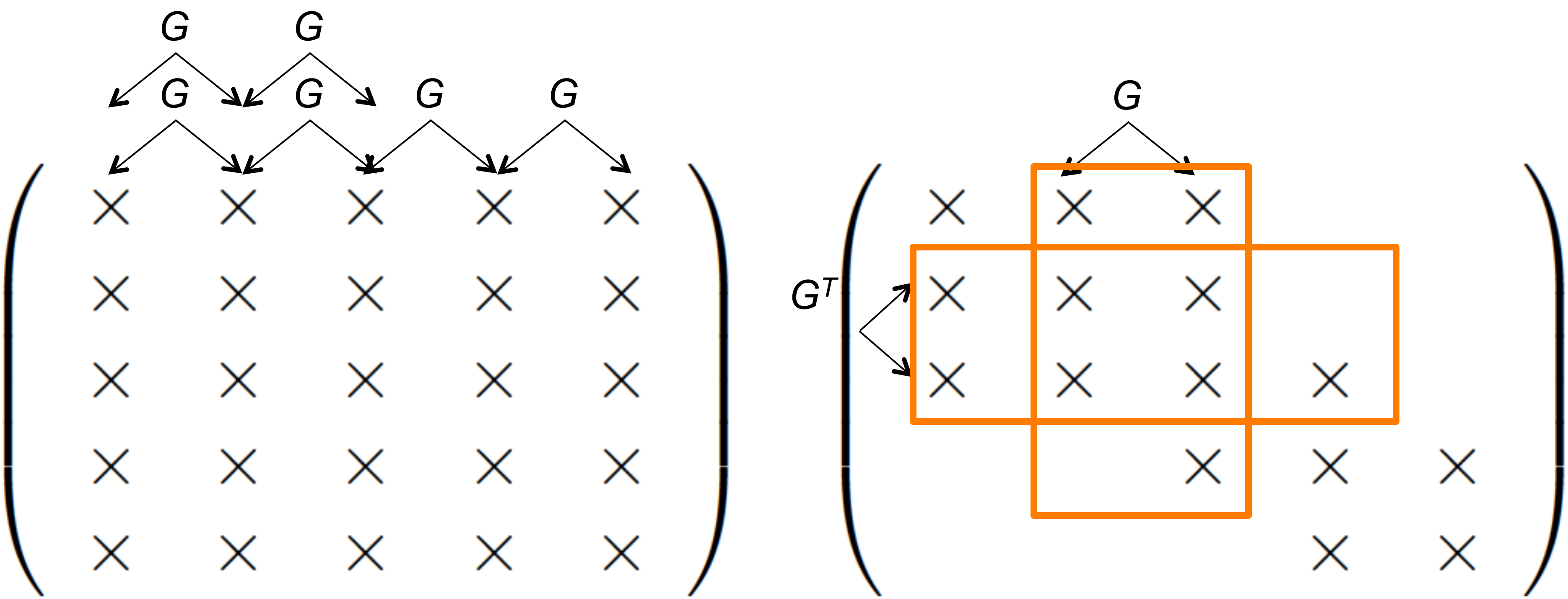
Accumulating Rotations (libflame)



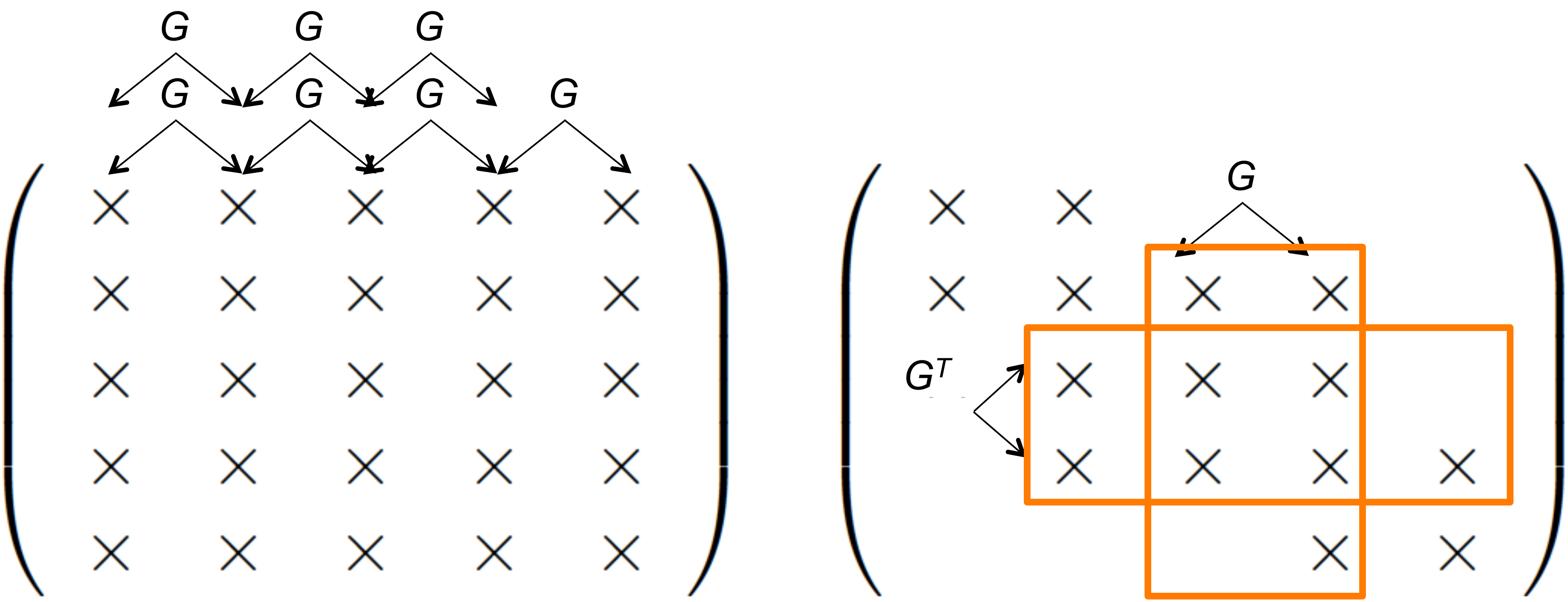
Accumulating Rotations (libflame)



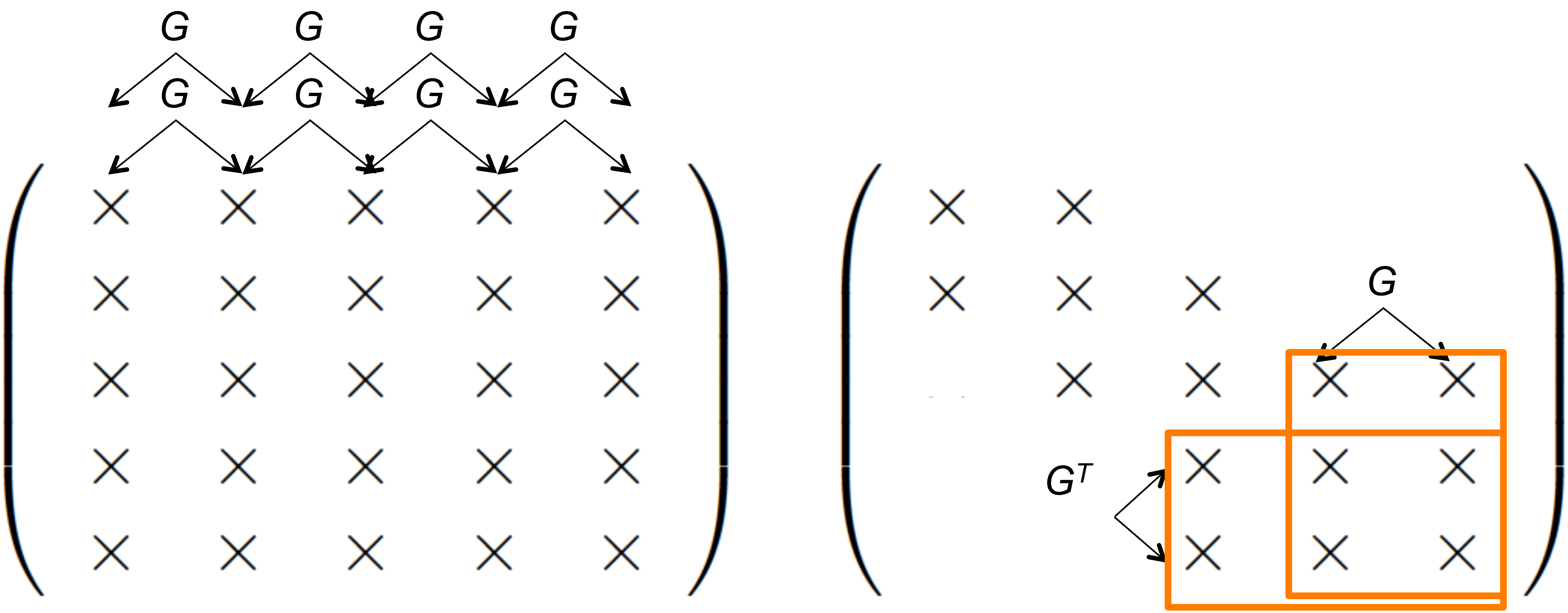
Accumulating Rotations (libflame)



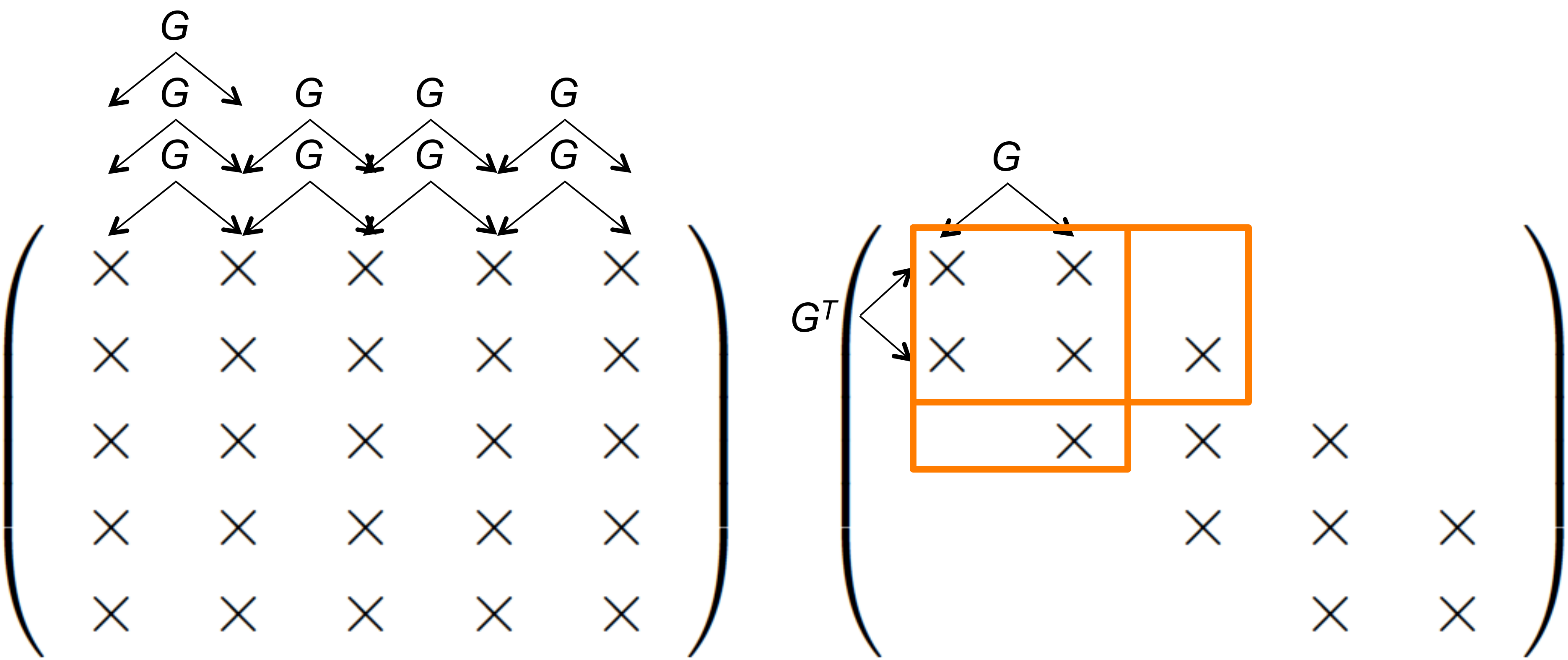
Accumulating Rotations (libflame)



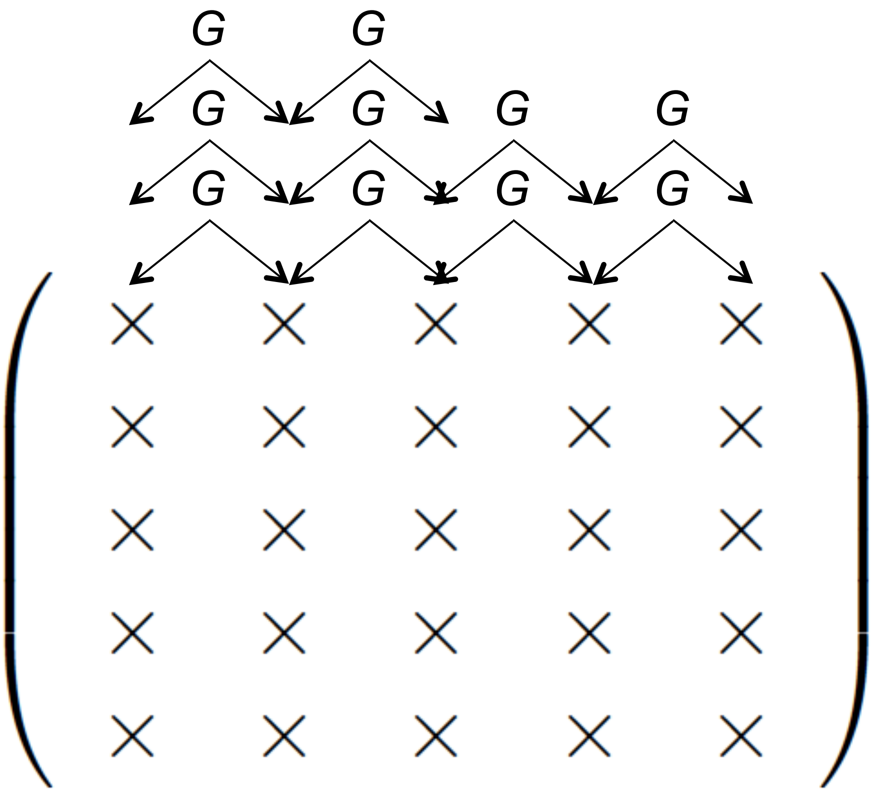
Accumulating Rotations (libflame)

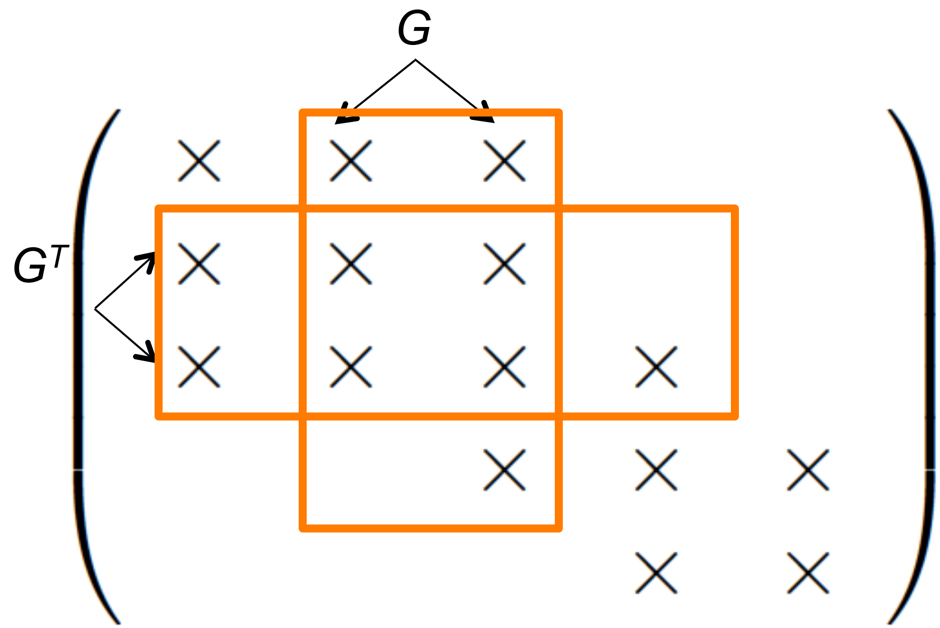


Accumulating Rotations (libflame)

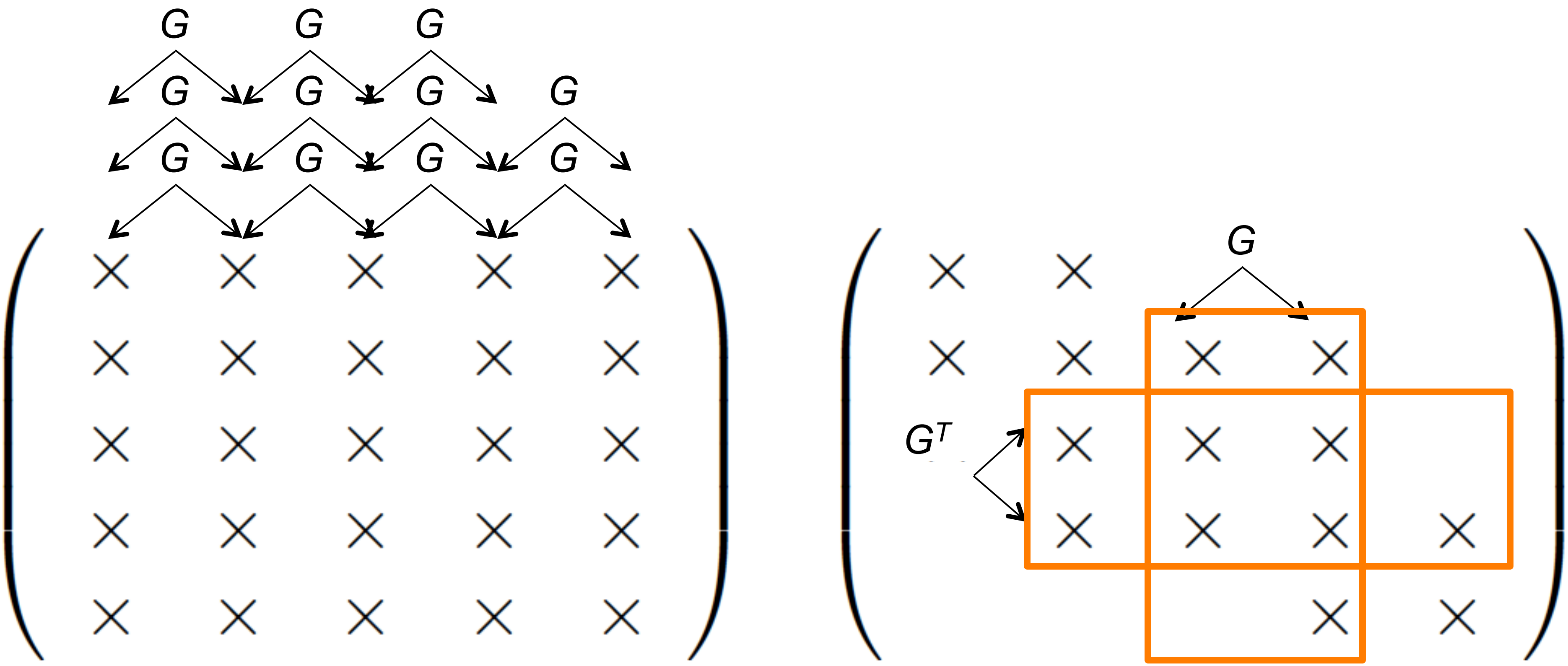


Accumulating Rotations (libflame)

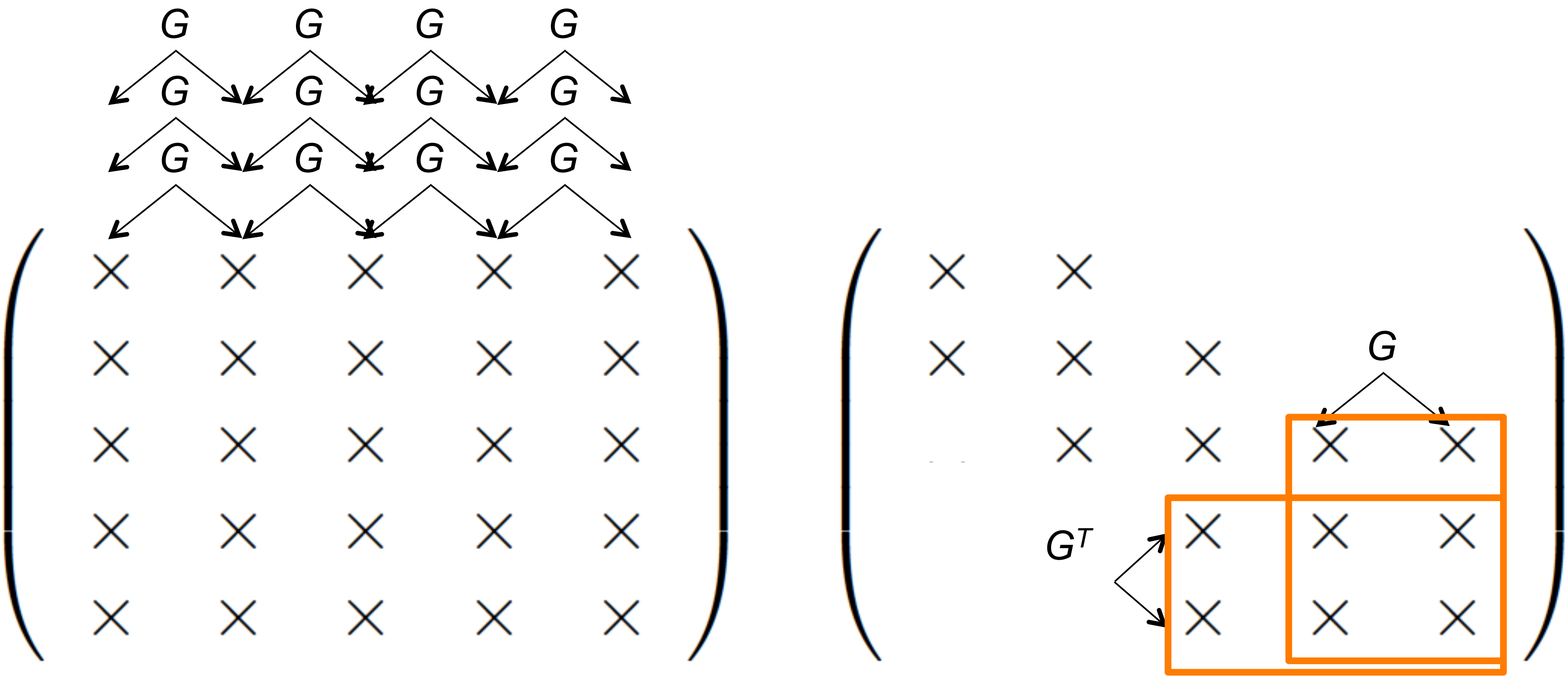




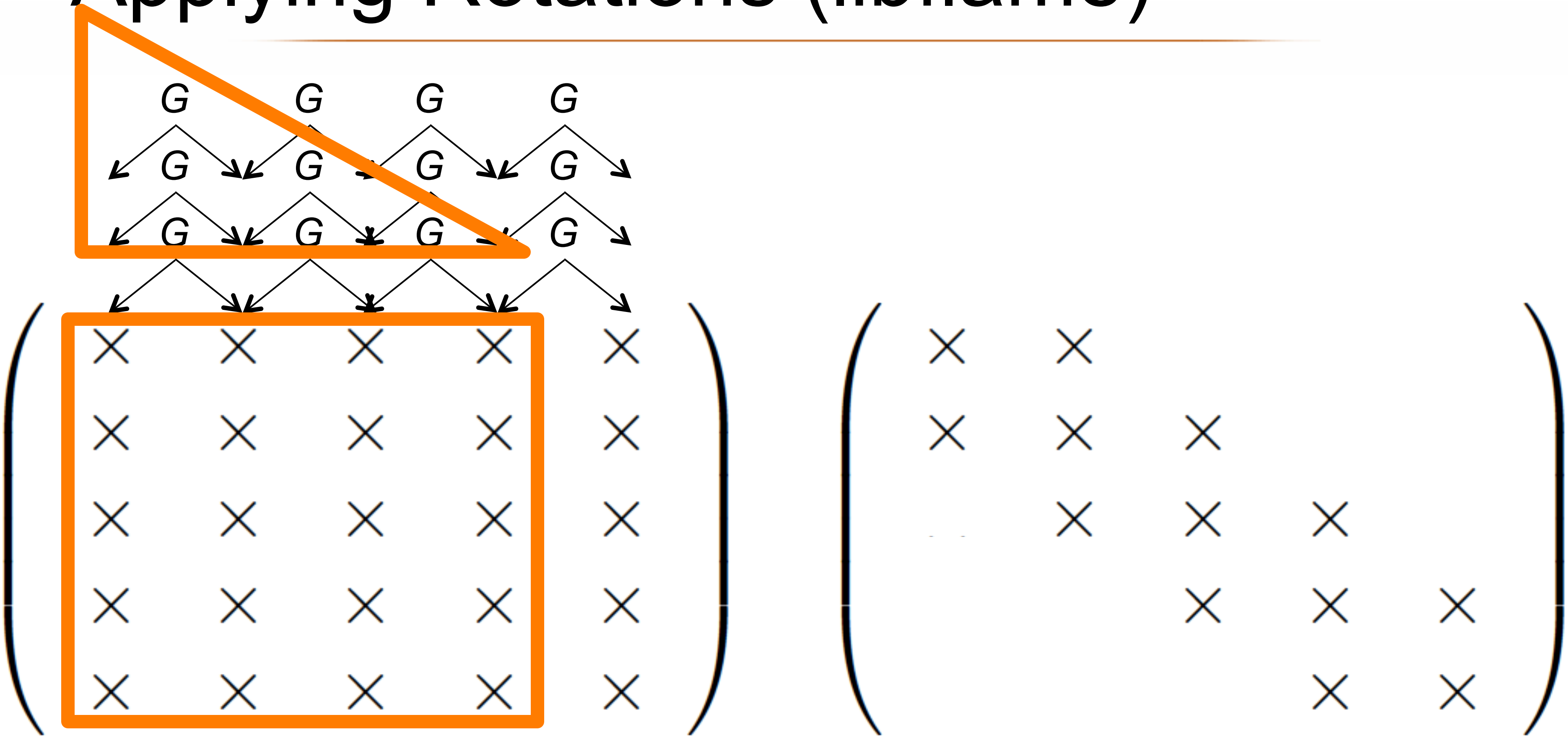
Accumulating Rotations (libflame)



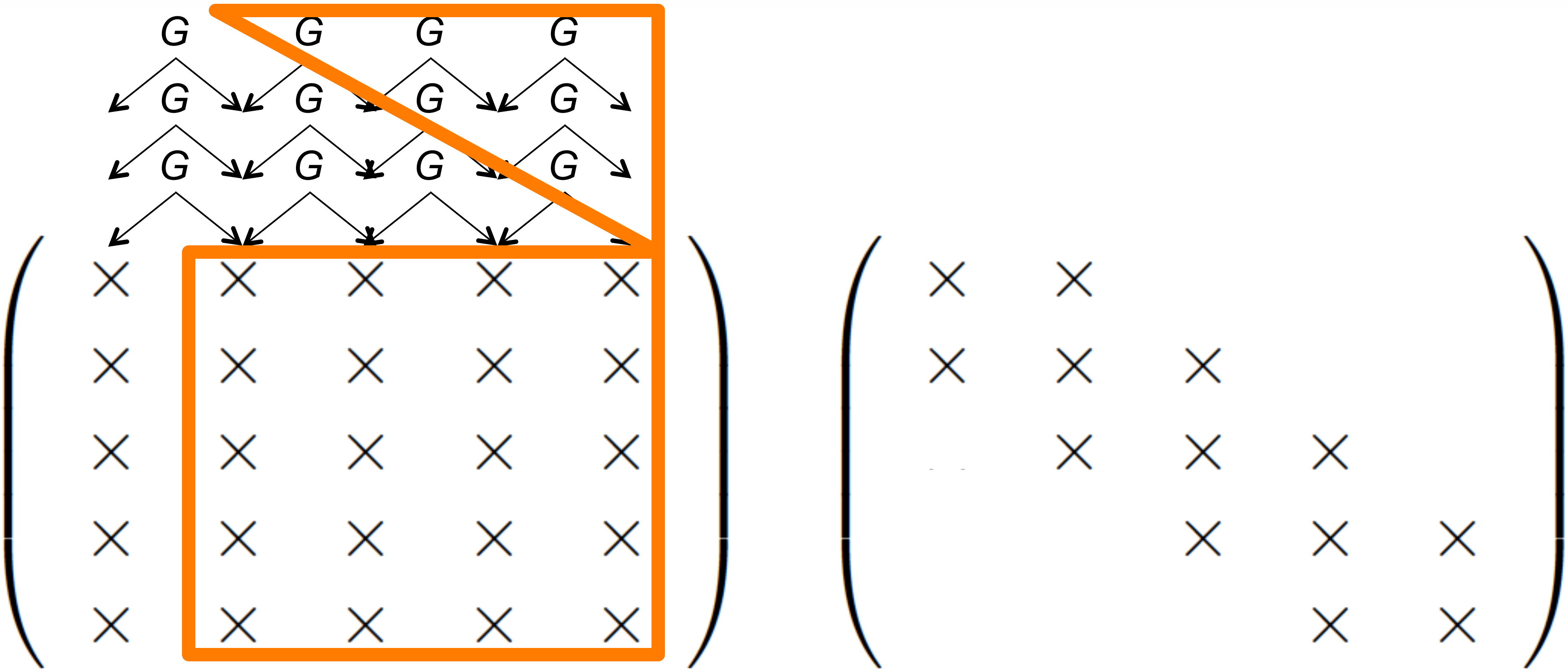
Accumulating Rotations (libflame)



Applying Rotations (libflame)



Applying Rotations (libflame)



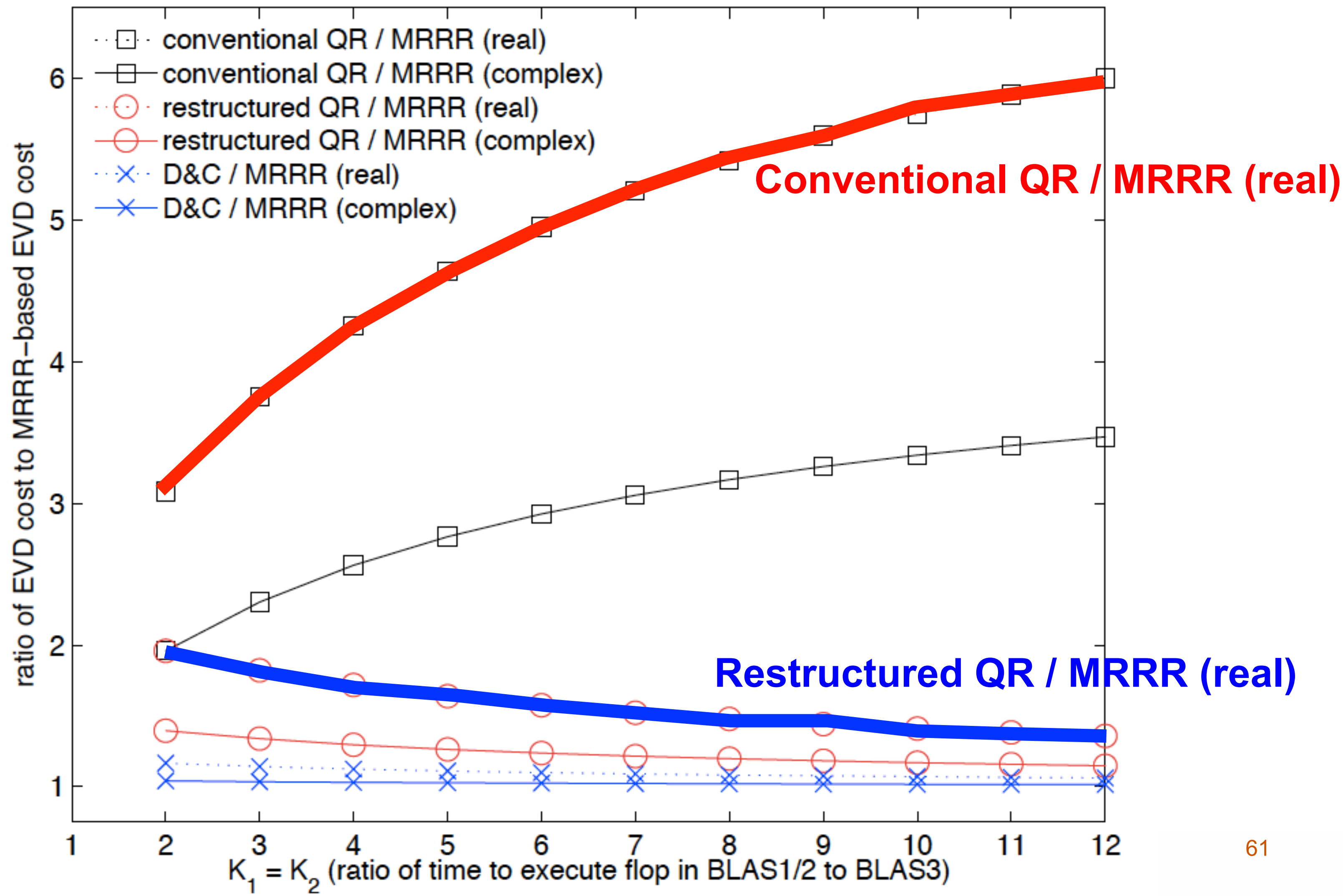
Optimization

- Applying a batch of Givens' rotations:
  - $O(n^2b)$ operations on $O(n^2)$ data.
  - Can attain "level-3 BLAS" performance

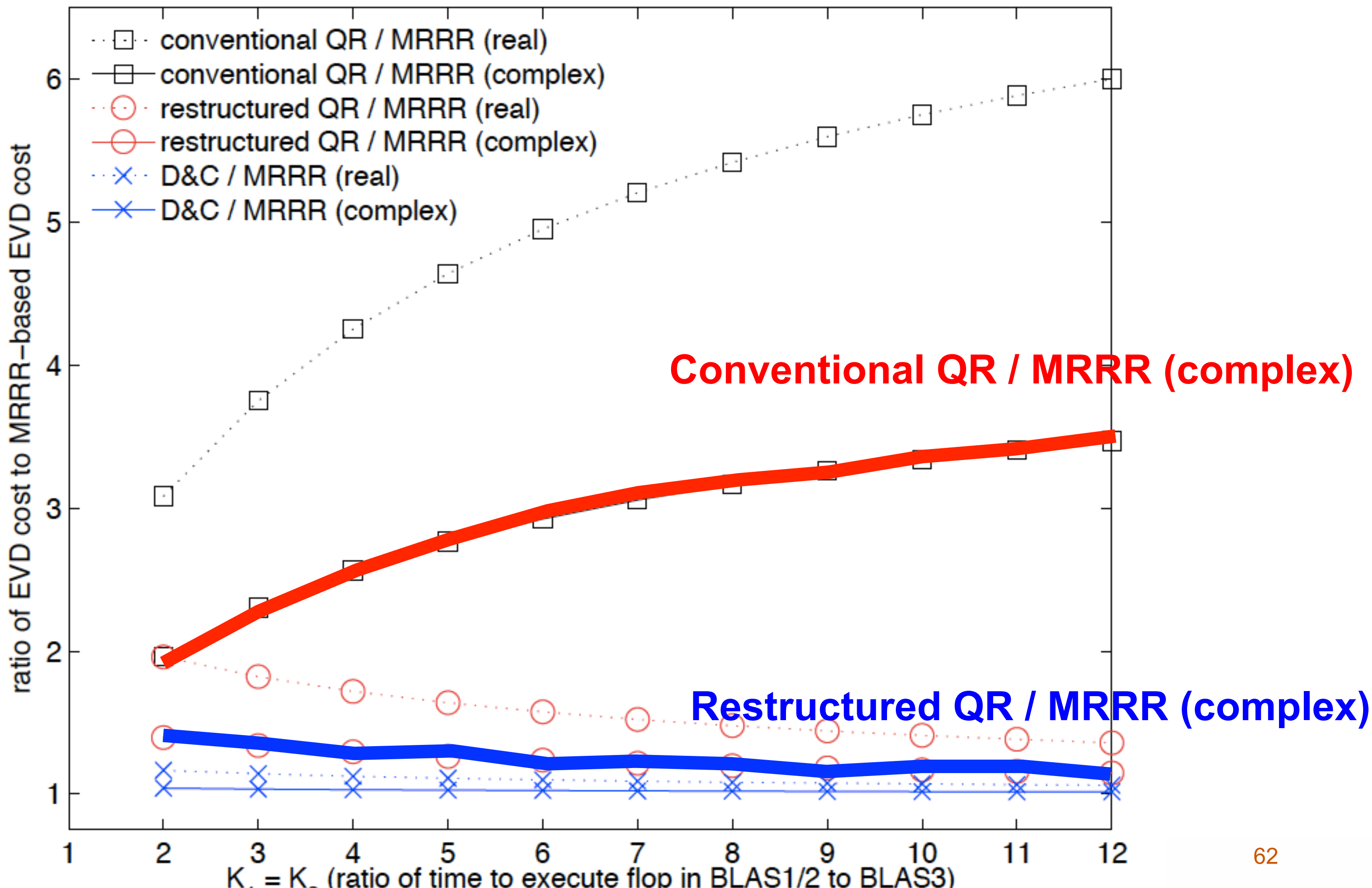
Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

Predicted Performance



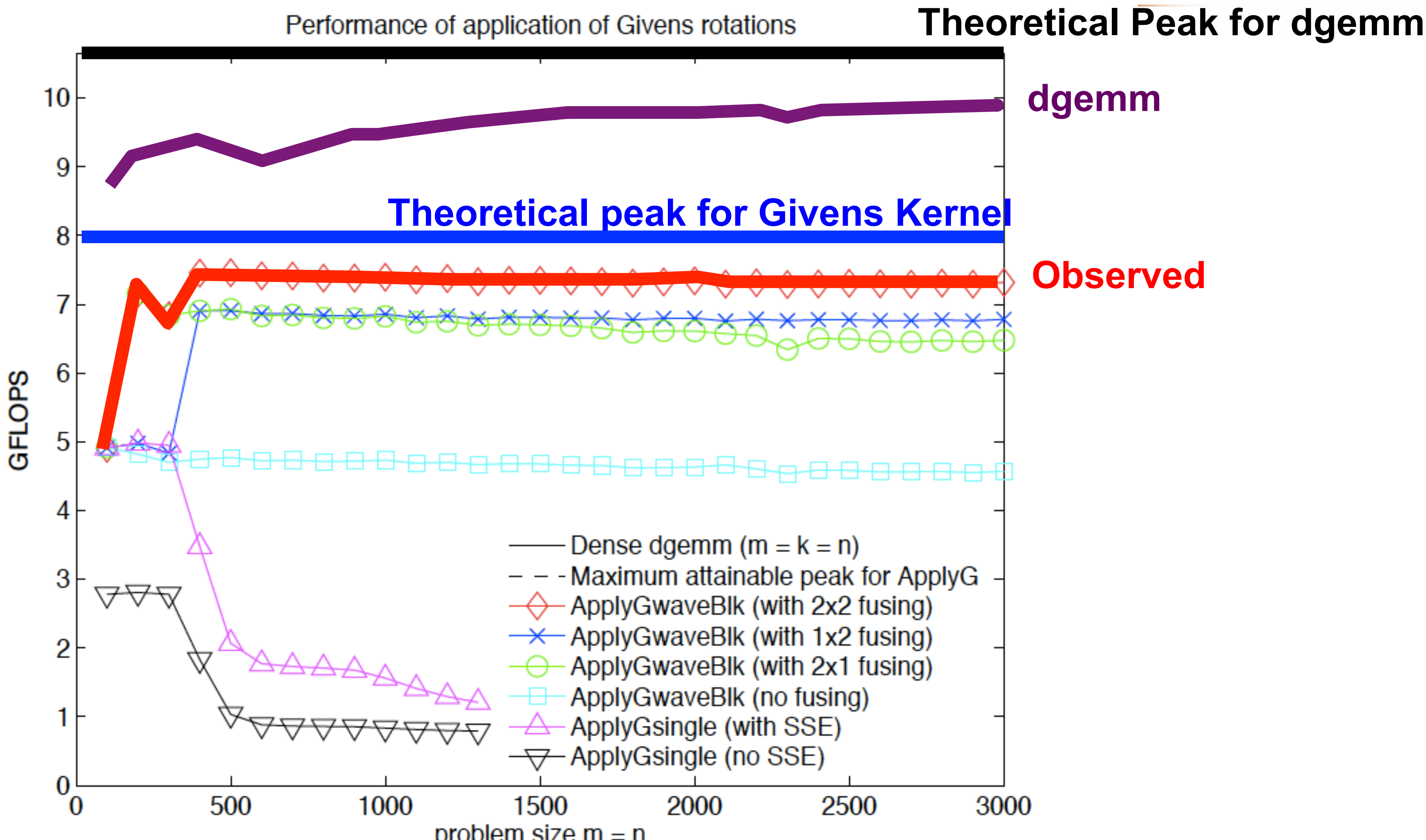
Predicted performance (EVD)



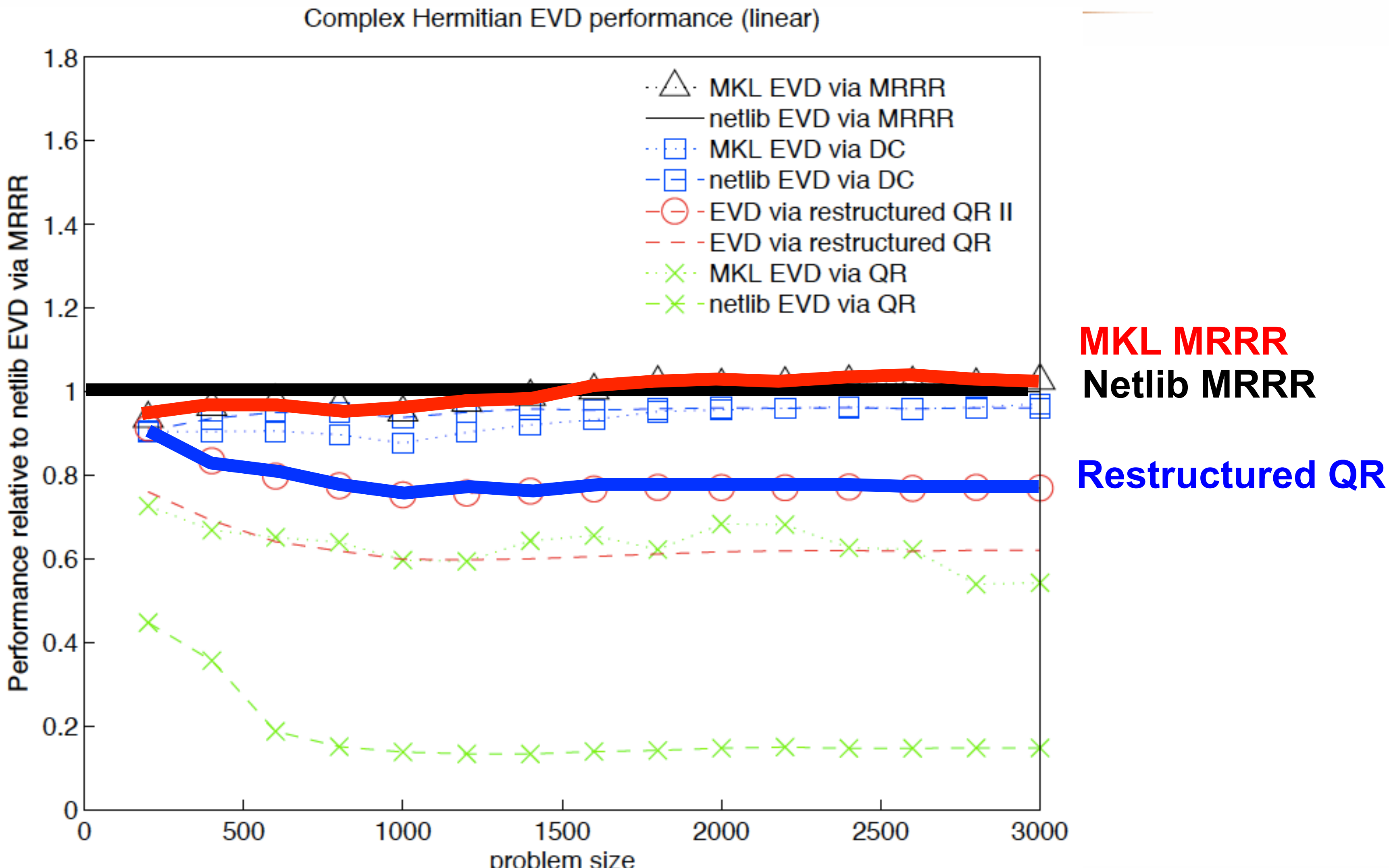
Observed Performance

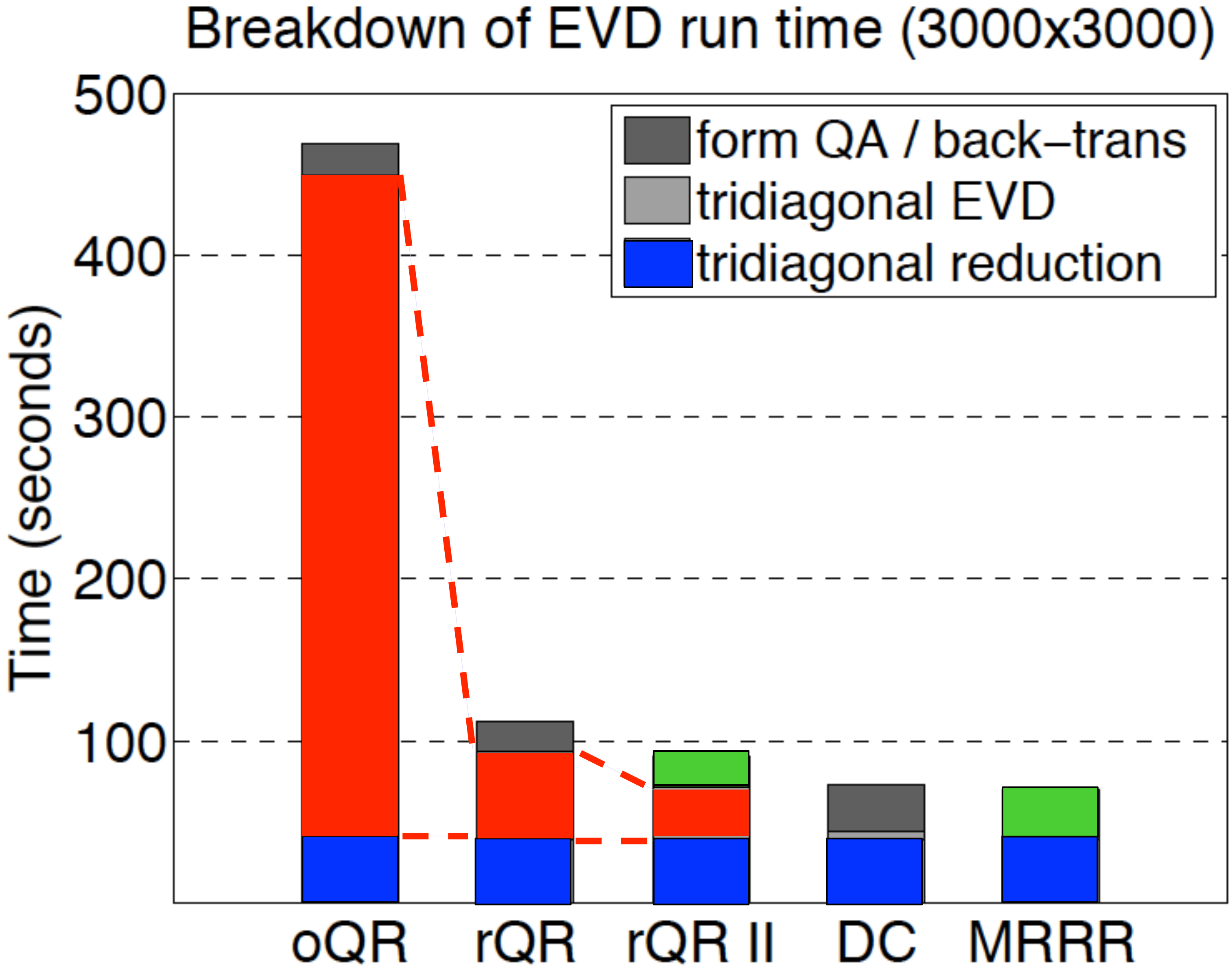
- Target architecture:
- Single core of a Dell PowerEdge R900 server
- 16 megabyte L2 cache/core.
- Single core peak of 10.64 GFLOPS.

Application of Givens rotations



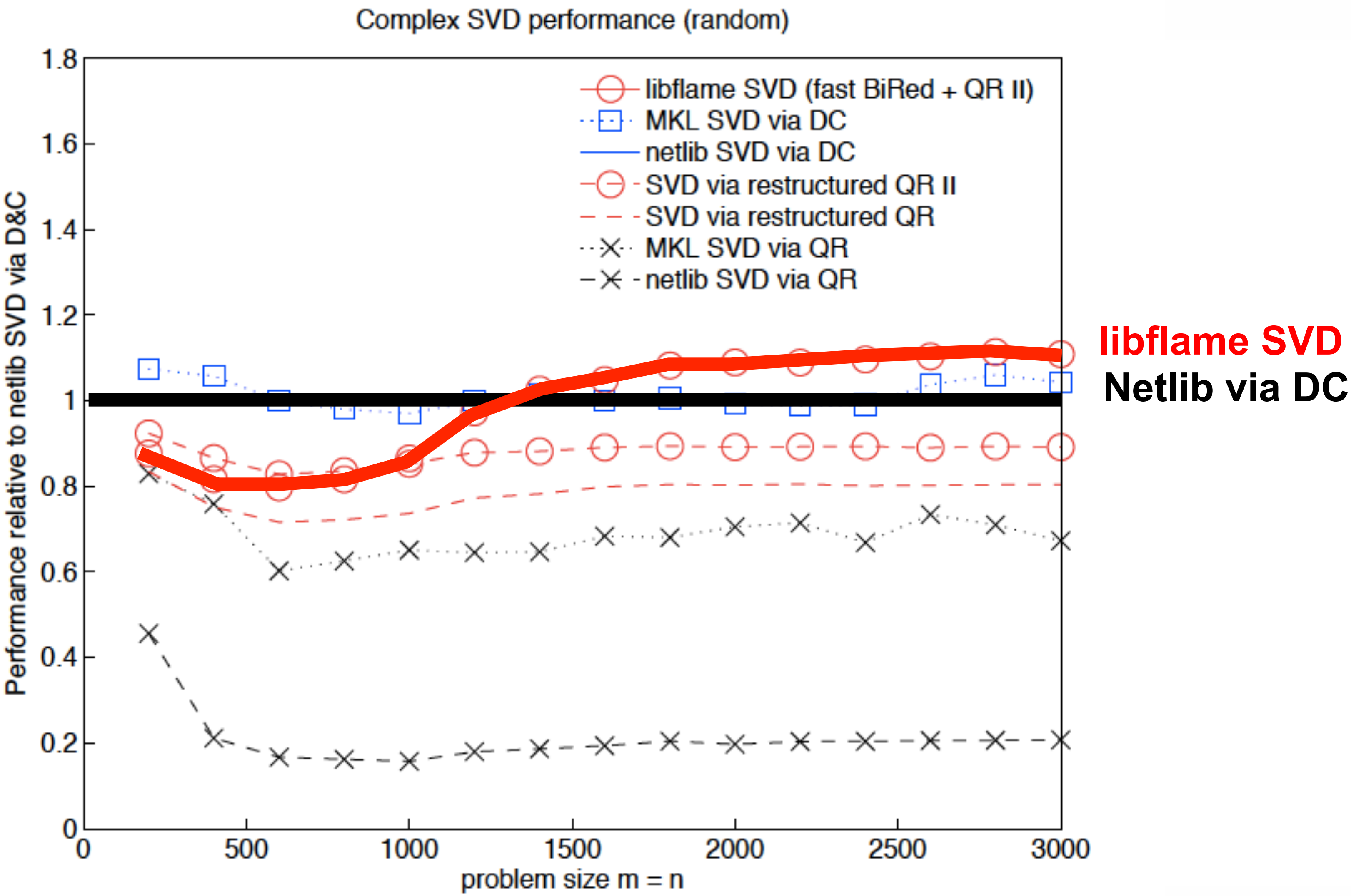
EVD performance (relative to netlib MRRR)



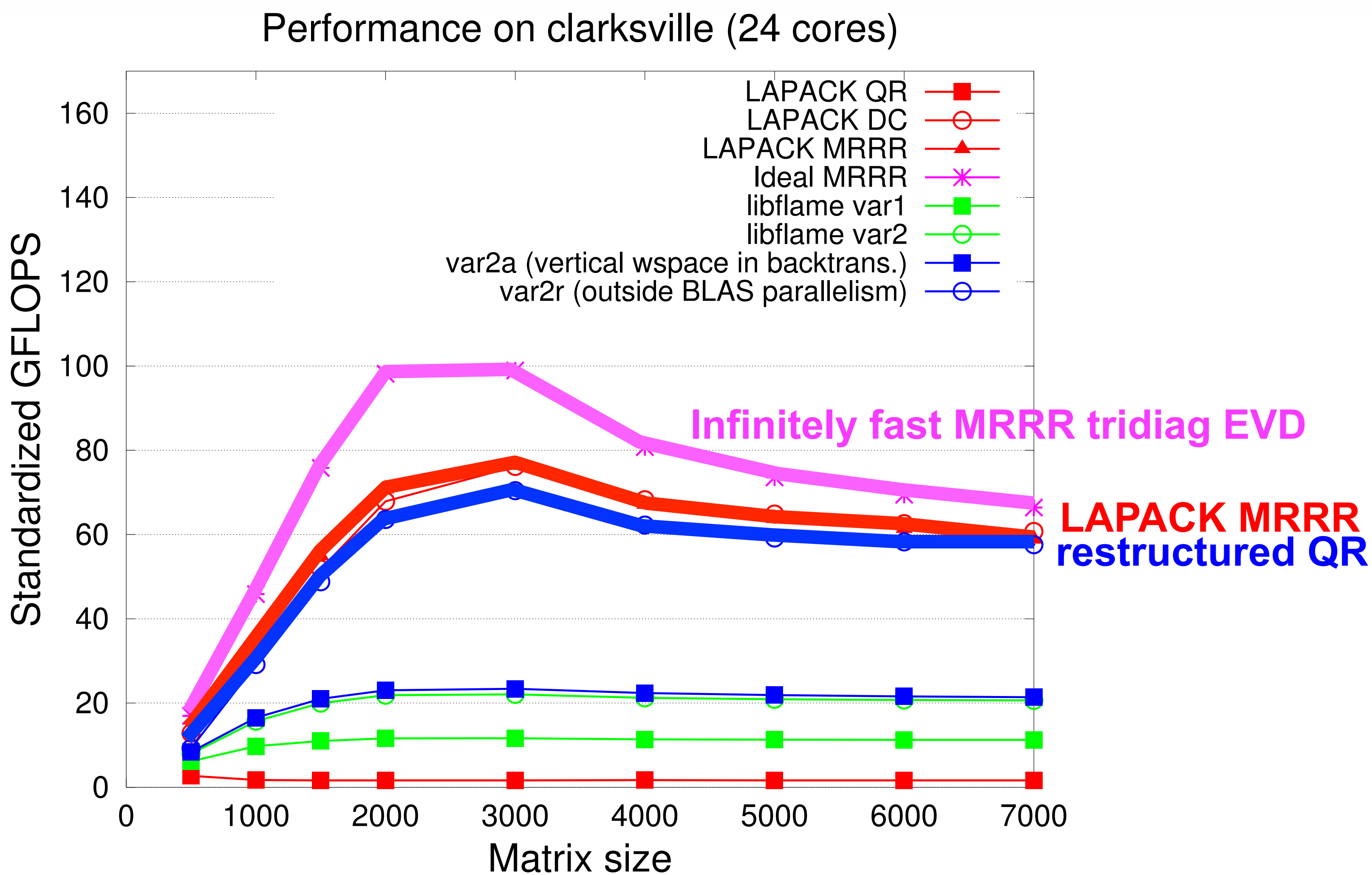


Breakdown of EVD run time (3000x3000)
500
400
300
200
100
Time (seconds)
form QA / back–trans
tridiagonal EVD
tridiagonal reduction
oQR
rQR
rQR II
DC
MRRR

libflame SVD Performance



EVD Parallel Performance (24 cores)



Is your favorite graph missing?

- The paper has an electronic appendix with tons of performance graphs.

Overview

- 50+ years of progress
- The hidden costs of MRRR and D&C
- QR algorithm basics
- Accumulating and applying rotations
- Performance
- Conclusion

Conclusion

- The QR algorithm lives!
- Future directions:
  - Parallelization
  - (multi)GPU
  - Aggressive early deflation