Parallel Multishift QR and QZ Algorithms with Advanced Deflation Strategies — Recent Progress

Björn Adlerborn[1], Robert Granat[1], **Bo Kågström**[1], Lars Karlsson[1], Daniel Kressner[2] and Meiyue Shao[1,2]

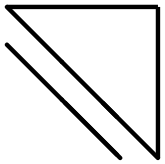
[1]Department of Computing Science and HPC2N, Umeå University, Sweden

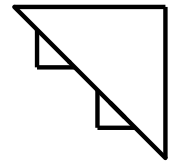
[2]MATHICSE, EPFL Lausanne, Switzerland

SIAM-CSE 2013, Boston, Feb. 25 - Mar. 1, 2013

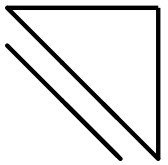
Background – computing Schur forms

$$A \xrightarrow{\textit{Step 1}} H = \quad \xrightarrow{\textit{Step 2}} S =$$

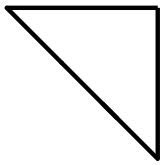




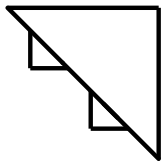
$$(A, B) \rightarrow (H, T) = ($$



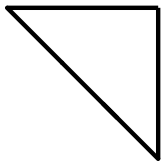
$$,$$



$$) \rightarrow (S, T) = ($$



$$,$$



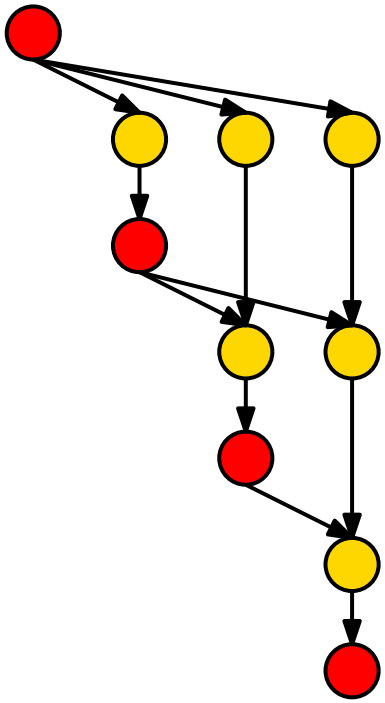
$$)$$

Provide orthogonal bases of subspaces associated with a *specified spectrum*.

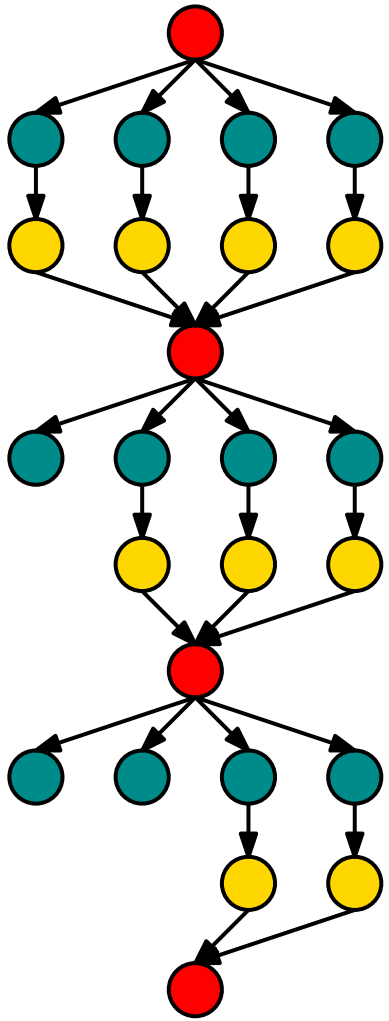
Option: Provide condition estimators!

Task graphs of one-sided and two-sided blocked matrix computations

$Q^T A = R$



$Q^T A Q = H$



Related talks in MS75 - Part I

- Parallel Multishift QR and QZ Algorithms with Advanced Deflation Strategies - Recent Progress
  (presented by Bo Kågström)

- The Parallel Nonsymmetric QR Algorithm with Aggressive Early Deflation
  (presented by Meiyue Shao)

- Towards a Fine-Grained Parallel Implementation of the Nonsymmetric QR Algorithm
  (presented by Bo Kågström - substitute for Lars Karlsson)

Generalized Eigenvalue Problem

Given $n \times n$ matrices A and B, compute all (generalized) eigenvalues λ of $A - \lambda B$ (or $\beta A - \alpha B$), the roots of

$$\det(A - \lambda B) = 0.$$

- If B is nonsingular, coincides with matrix eigenvalue problem $B^{-1}A$.
- If B is singular, some of the eigenvalues are ∞.

Even for nonsingular B, forming $B^{-1}A$ is not advisable!

Some typical applications:

1. DAEs: dynamic systems with algebraic constraints (singular B);
2. Finite element discretizations with nonorthonormal basis functions (sometimes nearly singular B);
3. ... many other ...

Generalized Eigenvalue Problem

Given $n \times n$ matrices A and B, compute all (generalized) eigenvalues λ of $A - \lambda B$ (or $\beta A - \alpha B$), the roots of

$$\det(A - \lambda B) = 0.$$

- If B is nonsingular, coincides with matrix eigenvalue problem $B^{-1}A$.
- If B is singular, some of the eigenvalues are ∞.

Even for nonsingular B, forming $B^{-1}A$ is not advisable!

Some typical applications:

1. DAEs: dynamic systems with algebraic constraints (singular B);
2. Finite element discretizations with nonorthonormal basis functions (sometimes nearly singular B);
3. ... many other ...

Generalized Eigenvalue Problem

Given $n \times n$ matrices A and B, compute all (generalized) eigenvalues λ of $A - \lambda B$ (or $\beta A - \alpha B$), the roots of

$$\det(A - \lambda B) = 0.$$

- If B is nonsingular, coincides with matrix eigenvalue problem $B^{-1}A$.
- If B is singular, some of the eigenvalues are ∞.

Even for nonsingular B, forming $B^{-1}A$ is not advisable!

Some typical applications:

1. DAEs: dynamic systems with algebraic constraints (singular B);
2. Finite element discretizations with nonorthonormal basis functions (sometimes nearly singular B);
3. ... many other ...

The Generalized Schur Decomposition

Given $n \times n$ matrices A and B, there are unitary matrices Q and Z such that

$$S - \lambda T = Q^*(A - \lambda B)Z = \begin{bmatrix} \ddots \\ & \ddots \end{bmatrix} - \lambda \begin{bmatrix} \ddots \\ & \ddots \end{bmatrix}.$$

For A and B real, Q and Z can be chosen orthogonal, resulting in quasi-triangular S (1×1 and 2×2 blocks on the diagonal).

Eigenvalues come in pairs $(\alpha, \beta) = (s_{ii}, t_{ii})$:

- $\lambda_i = s_{ii}/t_{ii}$ (if $t_{ii} \neq 0$)
- $\lambda_i = \infty$ (if $t_{ii} = 0$)

If $s_{ii} = t_{ii} = 0$ for at least one i, the eigenvalue problem is called singular, otherwise regular.

The Generalized Schur Decomposition

Given $n \times n$ matrices A and B, there are unitary matrices Q and Z such that

$$S - \lambda T = Q^*(A - \lambda B)Z = \left[\begin{array}{cc} \diagdown & \\ & \diagdown \end{array}\right] - \lambda \left[\begin{array}{cc} \diagdown & \\ & \diagdown \end{array}\right].$$

For A and B real, Q and Z can be chosen orthogonal, resulting in quasi-triangular S (1×1 and 2×2 blocks on the diagonal).

Eigenvalues come in pairs $(\alpha, \beta) = (s_{ii}, t_{ii})$:

- $\lambda_i = s_{ii}/t_{ii}$ (if $t_{ii} \neq 0$)
- $\lambda_i = \infty$ (if $t_{ii} = 0$)

If $s_{ii} = t_{ii} = 0$ for at least one i, the eigenvalue problem is called singular, otherwise regular.

The Generalized Schur Decomposition

Given $n \times n$ matrices A and B, there are unitary matrices Q and Z such that

$$S - \lambda T = Q^*(A - \lambda B)Z = \left[\begin{array}{cc} \diagdown & \\ & \diagdown \end{array}\right] - \lambda \left[\begin{array}{cc} \diagdown & \\ & \diagdown \end{array}\right].$$

For A and B real, Q and Z can be chosen orthogonal, resulting in quasi-triangular S (1×1 and 2×2 blocks on the diagonal).

Eigenvalues come in pairs $(\alpha, \beta) = (s_{ii}, t_{ii})$:

- $\lambda_i = s_{ii}/t_{ii}$ (if $t_{ii} \neq 0$)
- $\lambda_i = \infty$ (if $t_{ii} = 0$)

If $s_{ii} = t_{ii} = 0$ for at least one i, the eigenvalue problem is called singular, otherwise regular.

Outline

- The generalized eigenvalue problem
- The generalized Schur decomposition

- QZ algorithm – brief review
- Multi-shift QZ variants
- Advanced deflation strategies (AED)
- Parallel multishift QZ algorithm with AED
- Dealing with infinite eigenvalues
- Library software
- Computational experiments

Outline

- The generalized eigenvalue problem
- The generalized Schur decomposition

- QZ algorithm – brief review
- Multi-shift QZ variants
- Advanced deflation strategies (AED)
- Parallel multishift QZ algorithm with AED
- Dealing with infinite eigenvalues
- Library software
- Computational experiments

QZ Algorithm [Moler/Stewart'73]

The purpose of the QZ algorithm is to compute a generalized Schur decomposition.

Ingredients:

1. Initial Hessenberg-triangular (HT) reduction: Compute Q and Z s.t.

$$H - \lambda T = Q^*(A - \lambda B)Z = \left[\begin{array}{cc} \diagdown & \diagdown \\ \diagdown & \end{array}\right] - \lambda \left[\begin{array}{cc} \diagdown & \diagdown \\ & \end{array}\right]$$

2. QZ iterations: Drive subdiagonal entries of H to zero.
3. Deflations: Sufficiently small subdiagonal elements ($\approx 10^{-16} \times \|H\|$) can be set to zero $\rightsquigarrow$

$$H - \lambda T = \begin{bmatrix} H_{11} - \lambda T_{11} & H_{12} - \lambda T_{12} \\ 0 & H_{22} - \lambda T_{22} \end{bmatrix}$$

QZ Algorithm [Moler/Stewart'73]

The purpose of the QZ algorithm is to compute a generalized Schur decomposition.

Ingredients:

1. Initial Hessenberg-triangular (HT) reduction: Compute Q and Z s.t.

$$H - \lambda T = Q^*(A - \lambda B)Z = \left[\begin{array}{cc} \diagdown & \diagdown \\ \diagdown & \end{array}\right] - \lambda \left[\begin{array}{cc} \diagdown & \diagdown \\ & \end{array}\right]$$

2. QZ iterations: Drive subdiagonal entries of H to zero.
3. Deflations: Sufficiently small subdiagonal elements ($\approx 10^{-16} \times \|H\|$) can be set to zero $\rightsquigarrow$

$$H - \lambda T = \begin{bmatrix} H_{11} - \lambda T_{11} & H_{12} - \lambda T_{12} \\ 0 & H_{22} - \lambda T_{22} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$(HT^{-1} - \sigma_1 I)(HT^{-1} - \sigma_2 I)e_1 = \begin{bmatrix} \mathbf{x} \\ \mathbf{x} \\ \mathbf{x} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} \end{bmatrix}$$

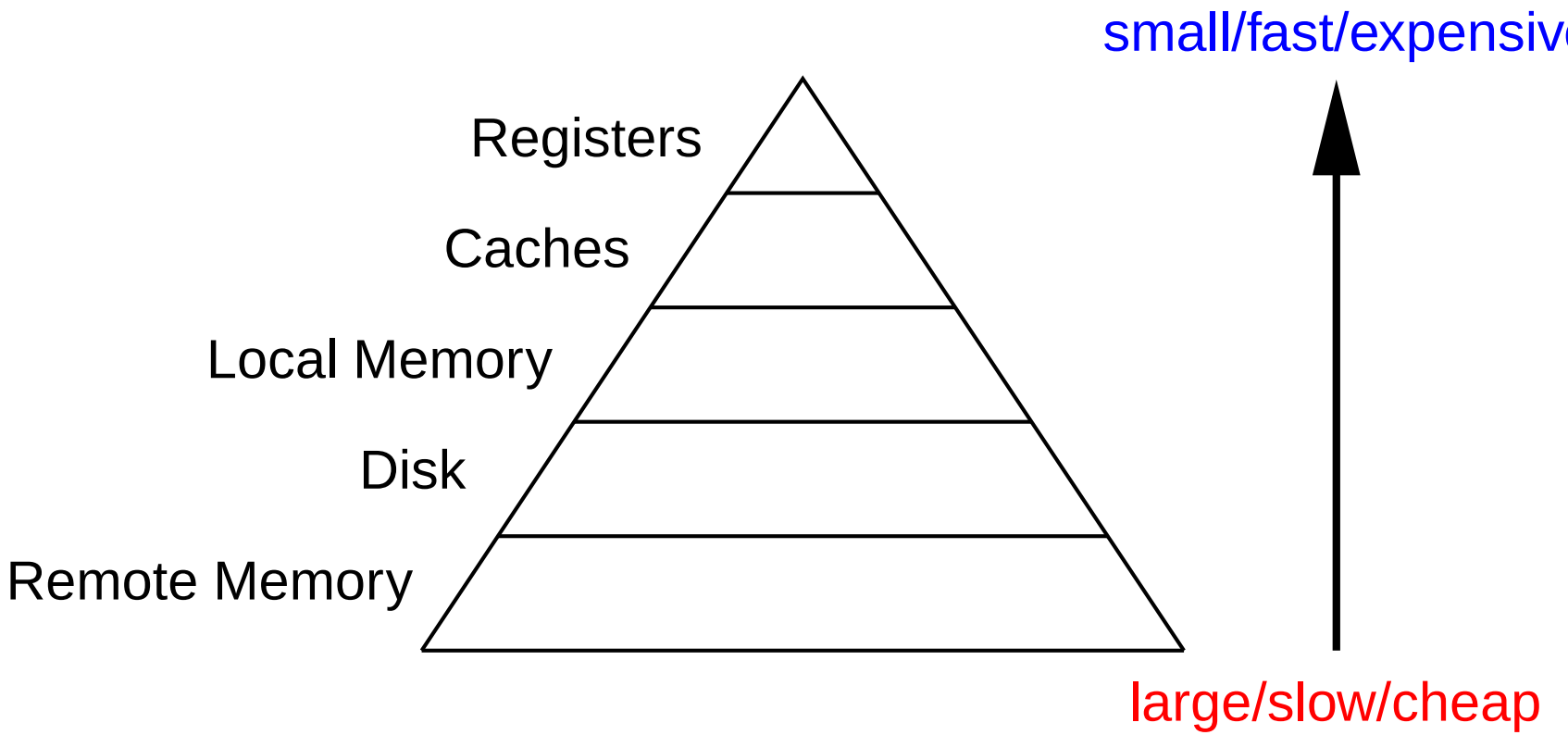
QZ Iterations

Steps of one QZ iteration on $H - \lambda T$:

1. Choose shifts $\sigma_1, \ldots, \sigma_m$ (m is tiny, say $m = 2$ or $m = 4$) as the eigenvalues of the bottom right $m \times m$ block of $H - \lambda T$.
2. Compute unitary Q_0 mapping $\prod(HT^{-1} - \sigma_k I)e_1$ to e_1.
3. Reduce $Q_0^*(H - \lambda T)$ back to Hessenberg-triangular form.

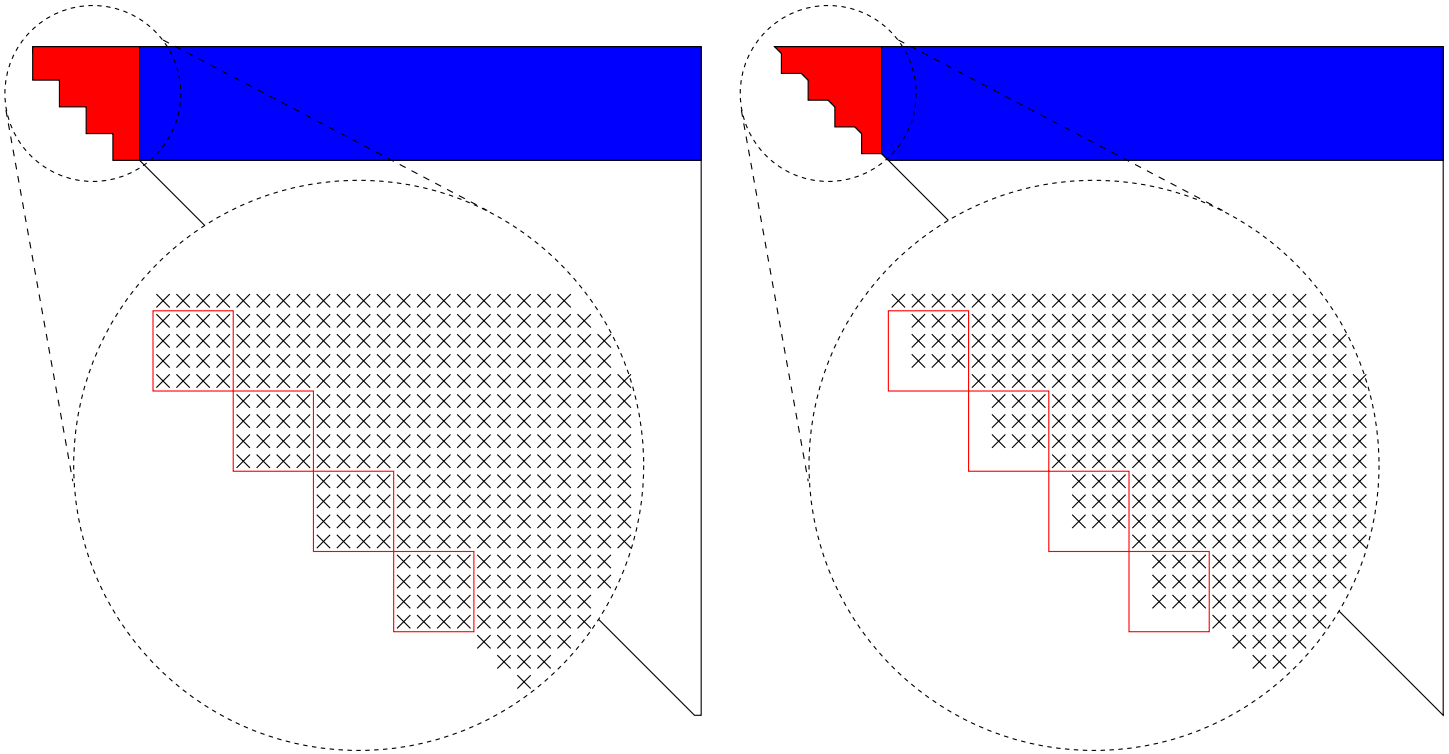
$$H - \lambda T \leftarrow \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{x} \end{bmatrix}$$

Memory Hierarchy and BLAS 3 Paradigm



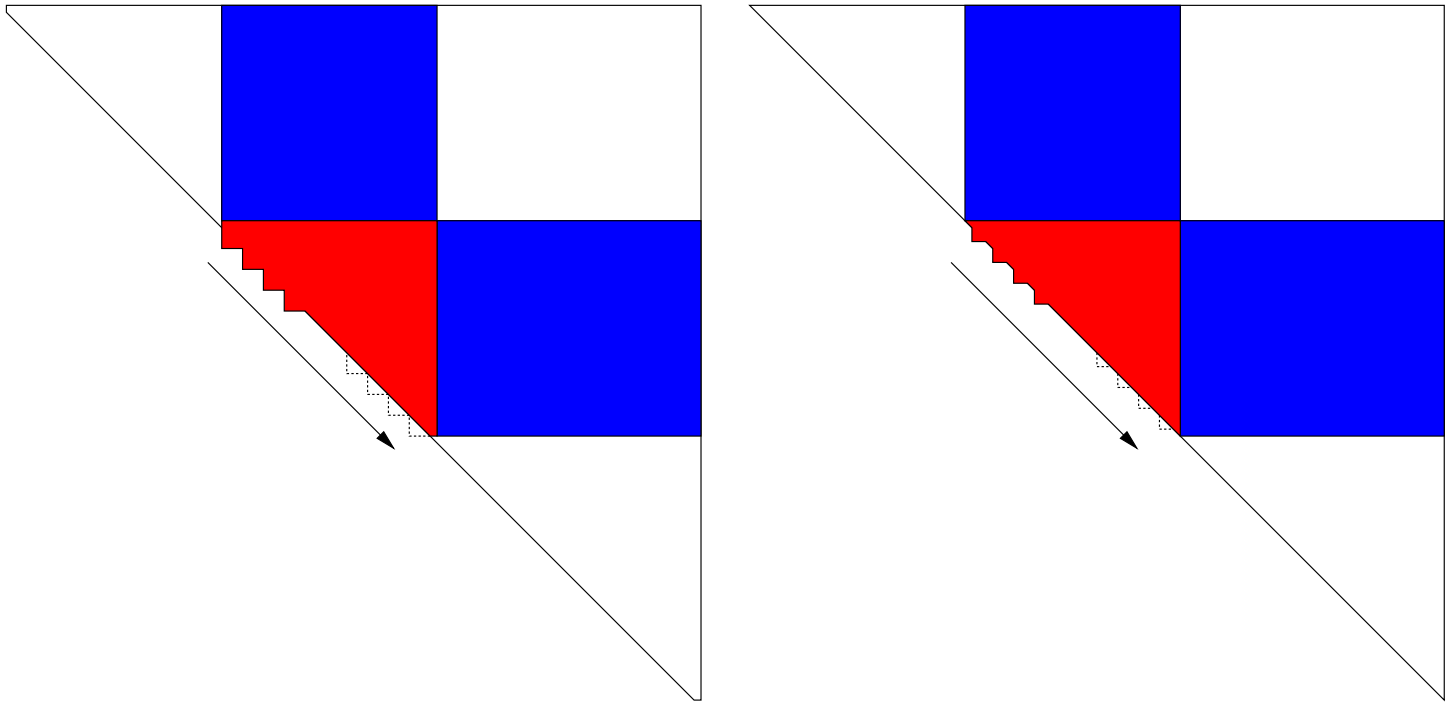
Paradigm: Do as many computations as possible in fast memory before moving data.
Restructure algorithm to increase usage of level 3 BLAS (matrix-matrix multiplies).

Multishift QZ – Introducing Chain



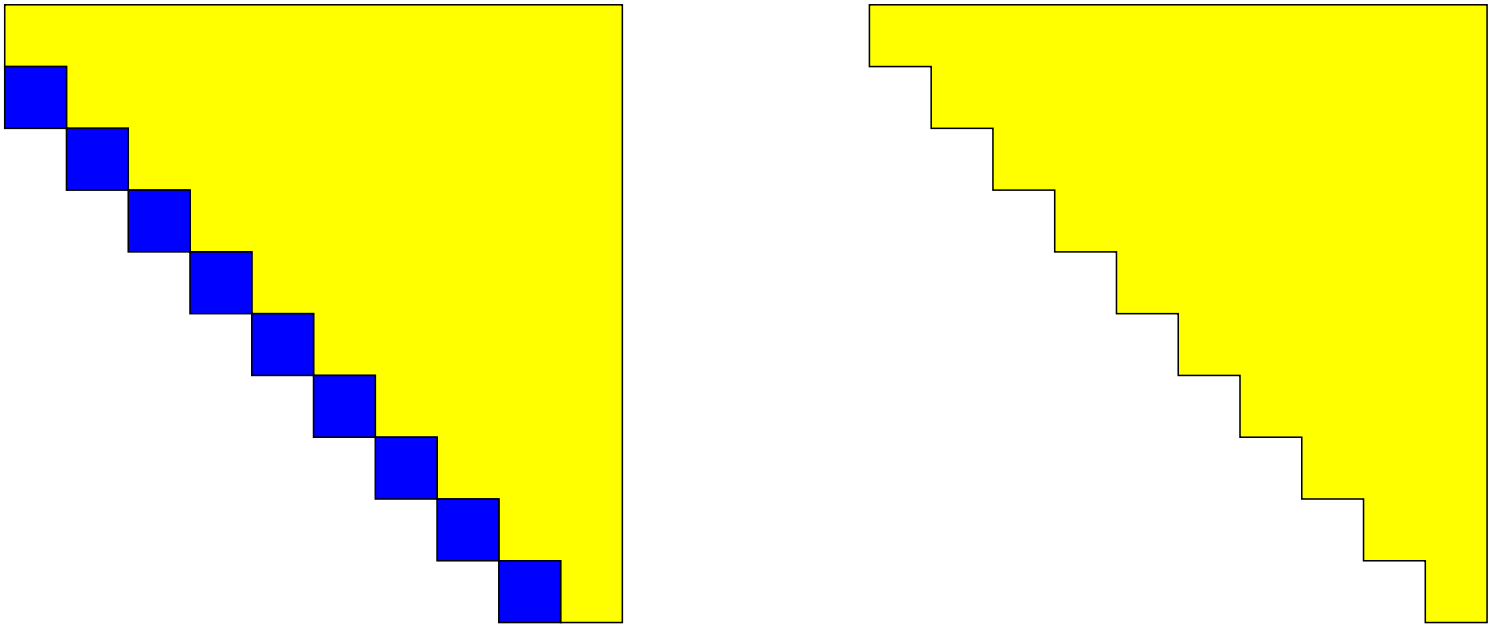
Red area: Updated during introduction.
Blue area: Updated after introduction via matrix-matrix-mult.

Multishift QZ – Chasing Chain



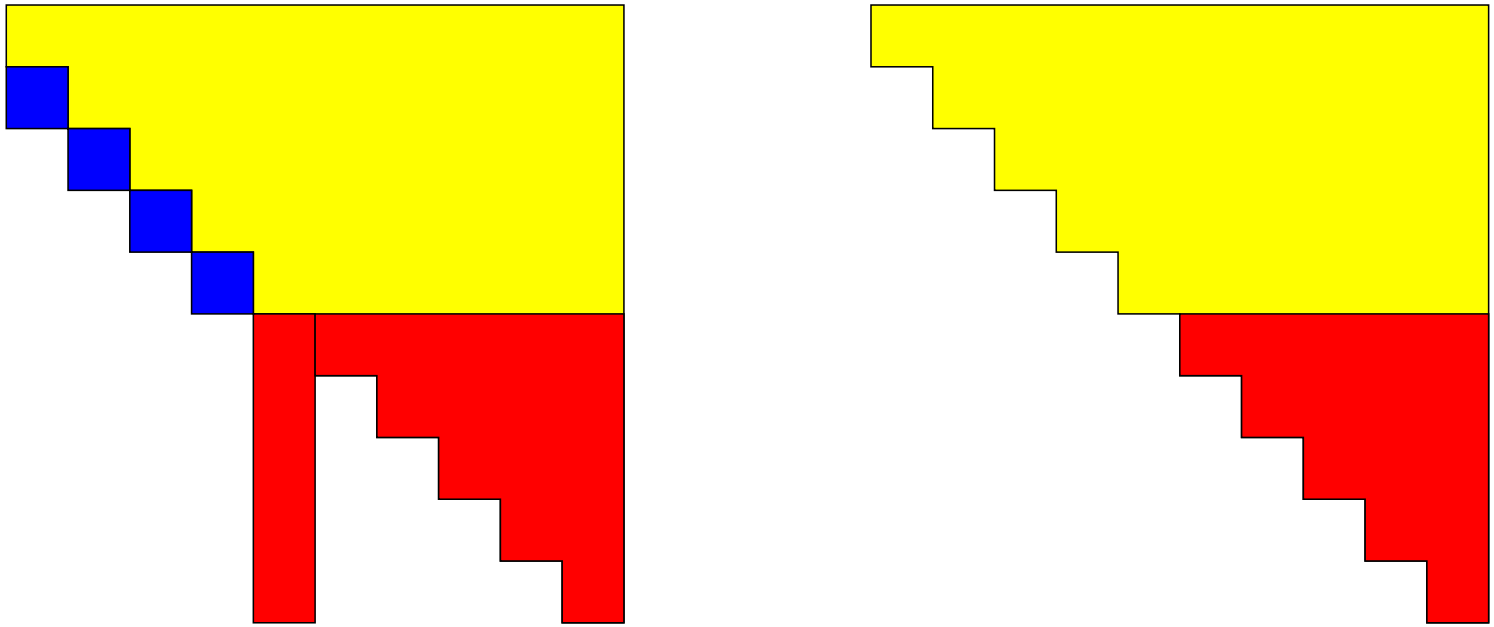
Red area: Updated during chase.
Blue area: Updated after chase via matrix-matrix-mult.

Aggressive Early Deflation in QZ



Standard deflation: Look for $h_{i+1,i} \approx 10^{-16} \times \|H\|$.

Aggressive Early Deflation in QZ



AED: look for small elements outside subdiagonal of H-part.

See [Kågström/Kressner'07], based on [Braman/Byers/Mathias'02].

AED in QZ Algorithm

Let (H, T) be in unreduced form:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix}, \qquad T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ 0 & T_{22} & T_{23} \\ 0 & 0 & T_{33} \end{bmatrix}$$

Block rows and columns are of size $n - n_w - 1, 1$ and n_w.

Deflation window:
Submatrix pair $([H_{32},\ H_{33}], [0,\ T_{33}])$ of size $n_w \times (n_w + 1)$.

Compute generalized Schur decomposition of (H_{33}, T_{33}):

$$(\hat{S}_{33}, \hat{T}_{33}) = Q^H (H_{33}, T_{33}) Z$$

AED in QZ Algorithm

Let (H, T) be in unreduced form:

$$H = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & H_{32} & H_{33} \end{bmatrix}, \qquad T = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ 0 & T_{22} & T_{23} \\ 0 & 0 & T_{33} \end{bmatrix}$$

Block rows and columns are of size $n - n_w - 1, 1$ and n_w.

Deflation window:
Submatrix pair $([H_{32},\ H_{33}], [0,\ T_{33}])$ of size $n_w \times (n_w + 1)$.

Compute generalized Schur decomposition of (H_{33}, T_{33}):

$$(\hat{S}_{33}, \hat{T}_{33}) = Q^H (H_{33}, T_{33}) Z$$

AED in QZ Algorithm

Apply equivalence transformation to (H, T):

$$\begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & Q^H \end{bmatrix} (H, T) \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & Z \end{bmatrix} = (\hat{H}, \hat{T}),$$

where

$$\hat{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13}Z \\ H_{21} & H_{22} & H_{23}Z \\ 0 & Q^H H_{32} & \hat{S}_{33} \end{bmatrix}, \quad \hat{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13}Z \\ 0 & T_{22} & T_{23}Z \\ 0 & 0 & \hat{T}_{33} \end{bmatrix}.$$

$$s = Q^H H_{32} \qquad n_w \times 1 \qquad \text{“the spike”}$$

Early Deflation?

If k of the trailing components of

$$s = Q^H H_{32}$$

are tiny ($< 10^{-16}\|H\|_F$), deflate w.r.t. the trailing $k \times k$ matrix pair:

$$\tilde{H} = \begin{bmatrix} H_{11} & H_{12} & \hat{H}_{13} & \hat{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \hat{s} & \hat{H}_{33} & \hat{H}_{34} \\ 0 & 0 & 0 & \hat{H}_{44} \end{bmatrix} \quad \tilde{T} = \begin{bmatrix} T_{11} & T_{12} & \hat{T}_{13} & \hat{T}_{14} \\ 0 & T_{22} & \tilde{T}_{23} & \tilde{T}_{24} \\ 0 & 0 & \hat{T}_{33} & \hat{T}_{34} \\ 0 & 0 & 0 & \hat{T}_{44} \end{bmatrix}$$

Block rows and columns are of size $n - n_w - 1, 1, n_w - k$ and k, respectively.

BBM/ADK: Best Case Example of AED

Consider 6×6 matrix pair (H_6, T_6), which is a generalization of the motivating example by Braman-Byers-Mathias'02:

$$H_6 = \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 0.001 & 1 & 0 & 0 & 0 & 0 \\ & 0.001 & 2 & 0 & 0 & 0 \\ & & 0.001 & 3 & 0 & 0 \\ & & & 0.001 & 4 & 0 \\ & & & & 0.001 & 5 \end{bmatrix} \quad T_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 0 & 0 & 0 & 0 \\ & & 1 & 0 & 0 & 0 \\ & & & 1 & 0 & 0 \\ & & & & 1 & 0 \\ & & & & & 1 \end{bmatrix}$$

Estimate the distance between (H_6, T_6) and a matrix pair with eigenvalues equal to 5, 4, 3, 2 and 1.

BBM/ADK: Best Case Example of AED

Estimates of the distances ($\|.\|_F$) between (H_6, T_6) and a matrix pair with eigenvalues equal to (h_{ii}, t_{ii}) of (H_6, T_6) for $n_w = 5$.

A matrix pair with ...

- 5 as eigenvalue is within distance 10^{-17}
- 5 and 4 as eigenvalues is within distance 10^{-13}
- 5, 4 and 3 as eigenvalues is within distance 10^{-10}

... by setting the trailing k $(= 1, 2, 3)$ components of s to zero.

Retransform to HT-form

$$\tilde{H} = \begin{bmatrix} H_{11} & H_{12} & \hat{H}_{13} & \hat{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \hat{s} & \hat{H}_{33} & \hat{H}_{34} \\ 0 & 0 & 0 & \hat{H}_{44} \end{bmatrix} \quad \tilde{T} = \begin{bmatrix} T_{11} & T_{12} & \hat{T}_{13} & \hat{T}_{14} \\ 0 & T_{22} & \tilde{T}_{23} & \tilde{T}_{24} \\ 0 & 0 & \hat{T}_{33} & \hat{T}_{34} \\ 0 & 0 & 0 & \hat{T}_{44} \end{bmatrix}$$

- Construct $Q = I - \beta vv^T$ such that $Q^T \hat{s} = ce_1$
- Transform $(Q^T \hat{H}_{33}, Q^T \hat{T}_{33})$ to HT-form:

$$Q^T \hat{T}_{33} = (I - \beta vv^T)\hat{T}_{33} = \hat{T}_{33} - \beta v(\hat{T}_{33} v)^T$$

Rank-1 perturbation of upper triangular matrix

- Apply RQ-updating: $2(n_w - k - 1)$ rotations $\longrightarrow Z$
- Apply standard HT-reduction algorithm to $(Q^T \hat{H}_{33} Z, Q^T \hat{T}_{33} Z)$

Retransform to HT-form

$$\tilde{H} = \begin{bmatrix} H_{11} & H_{12} & \hat{H}_{13} & \hat{H}_{14} \\ H_{21} & H_{22} & \tilde{H}_{23} & \tilde{H}_{24} \\ 0 & \hat{s} & \hat{H}_{33} & \hat{H}_{34} \\ 0 & 0 & 0 & \hat{H}_{44} \end{bmatrix} \quad \tilde{T} = \begin{bmatrix} T_{11} & T_{12} & \hat{T}_{13} & \hat{T}_{14} \\ 0 & T_{22} & \tilde{T}_{23} & \tilde{T}_{24} \\ 0 & 0 & \hat{T}_{33} & \hat{T}_{34} \\ 0 & 0 & 0 & \hat{T}_{44} \end{bmatrix}$$

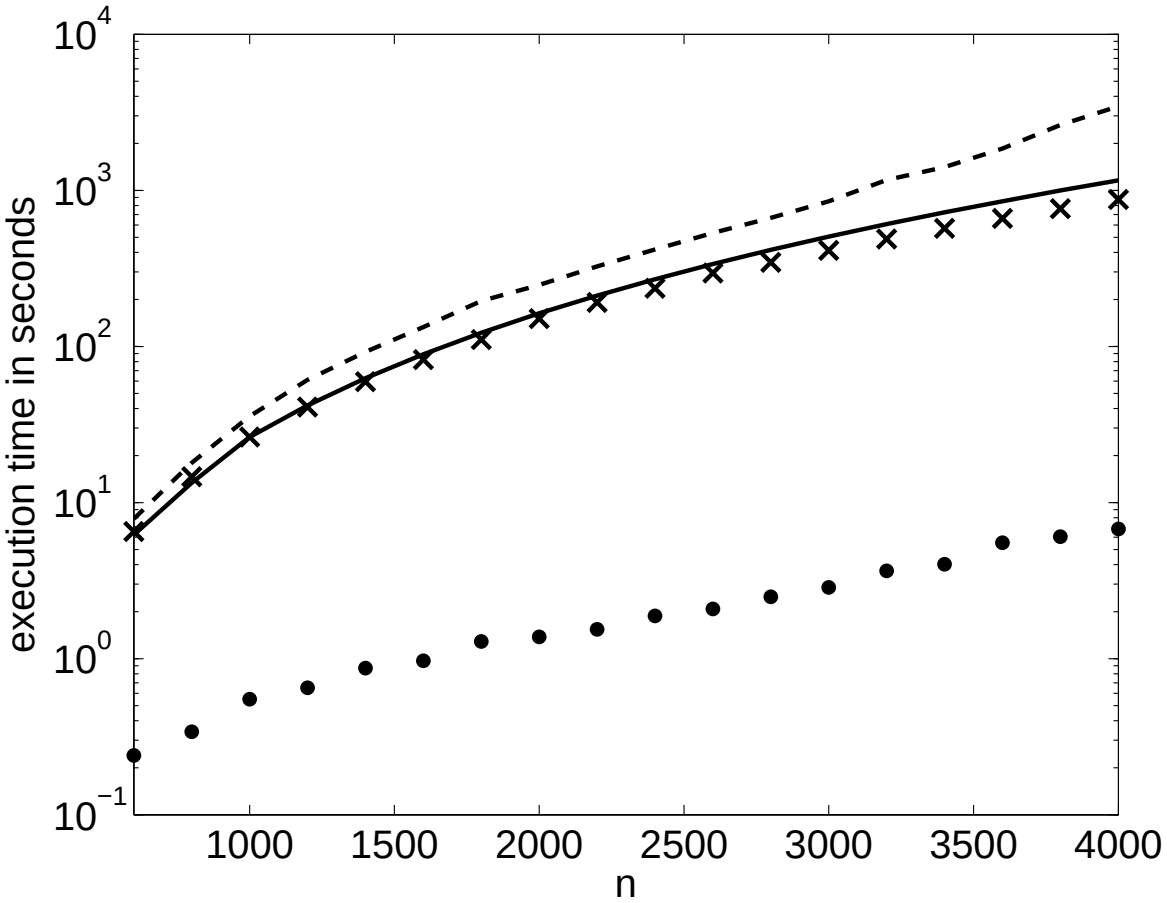
- Construct $Q = I - \beta v v^T$ such that $Q^T \hat{s} = c e_1$
- Transform $(Q^T \hat{H}_{33}, Q^T \hat{T}_{33})$ to HT-form:

$$Q^T \hat{T}_{33} = (I - \beta v v^T)\hat{T}_{33} = \hat{T}_{33} - \beta v (\hat{T}_{33} v)^T$$

Rank-1 perturbation of upper triangular matrix

- Apply RQ-updating: $2(n_w - k - 1)$ rotations $\longrightarrow Z$
- Apply standard HT-reduction algorithm to $(Q^T \hat{H}_{33} Z, Q^T \hat{T}_{33} Z)$

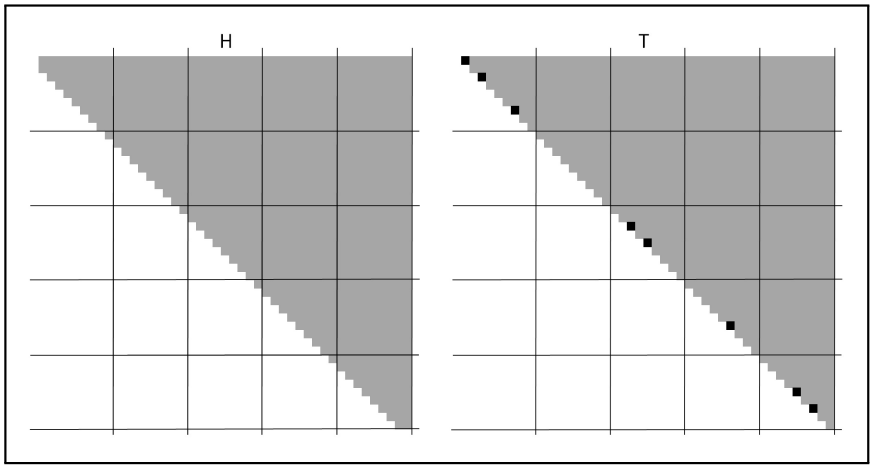
BBM/ADK example for $n = 600 - 4000$



Execution time in logarithmic scale without (top graphs) and with AED ($n_w = n - 1$). Time spent on (multishift) QZ iterations negligible compared to overall time!

Parallel QZ on distributed memory

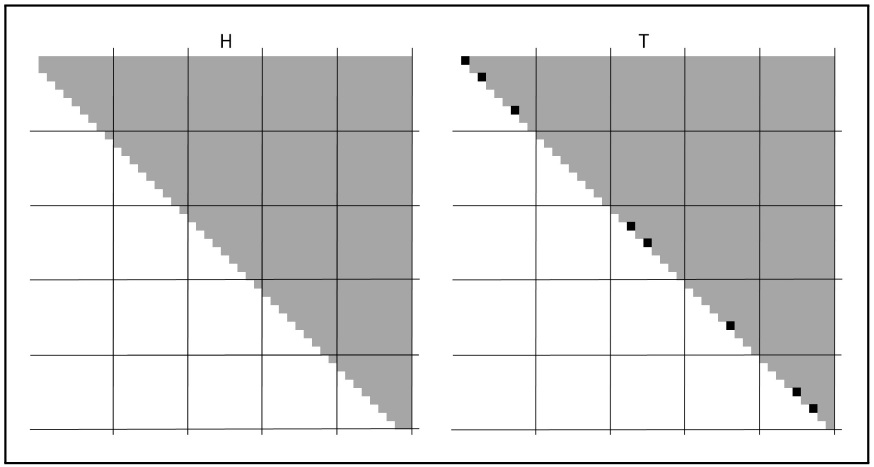
- $P_r \times P_c$ processor grid
- Square block ($NB \times NB$) cyclic data distribution
- Multiple computational windows



- Parallel multishift QZ iterations by chasing several tightly coupled bulge chains—level 3 operations!
- Parallel multi-level AED—faster convergence!
  - Reduce communication costs via data redistribution
  - Computations done on a subgrid
- Explained in context of the // QR algorithm (next talk!)

Parallel QZ on distributed memory

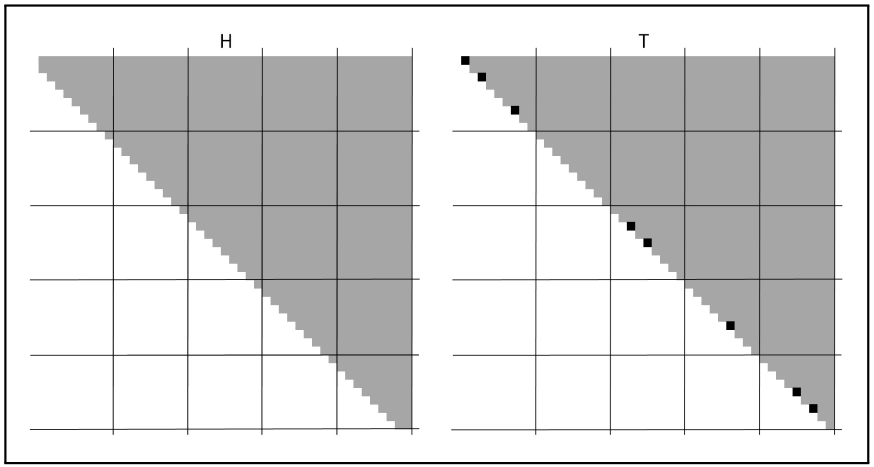
- $P_r \times P_c$ processor grid
- Square block ($NB \times NB$) cyclic data distribution
- Multiple computational windows



- Parallel multishift QZ iterations by chasing several tightly coupled bulge chains—level 3 operations!
- Parallel multi-level AED—faster convergence!
  - Reduce communication costs via data redistribution
  - Computations done on a subgrid
- Explained in context of the // QR algorithm (next talk!)

Parallel QZ on distributed memory

- $P_r \times P_c$ processor grid
- Square block ($NB \times NB$) cyclic data distribution
- Multiple computational windows



- Parallel multishift QZ iterations by chasing several tightly coupled bulge chains—level 3 operations!
- Parallel multi-level AED—faster convergence!
  - Reduce communication costs via data redistribution
  - Computations done on a subgrid
- Explained in context of the // QR algorithm (next talk!)

Treating Infinite Eigenvalues

QZ algorithm is in many aspects similar to the QR algorithm for solving matrix eigenvalue problems.

Fundamental difference: occurence of infinite eigenvalues!

In exact arithmetic, singularity of B implies zero diagonal entry in T:

$$H - \lambda T = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

Zero can be pushed to one of the corners and deflated.

Use windowing technique for deflating many infinite eigenvalues of a large matrix pair ($\Longrightarrow$ blocked algorithm with delayed updates).

Treating Infinite Eigenvalues

QZ algorithm is in many aspects similar to the QR algorithm for solving matrix eigenvalue problems.

Fundamental difference: occurence of infinite eigenvalues!

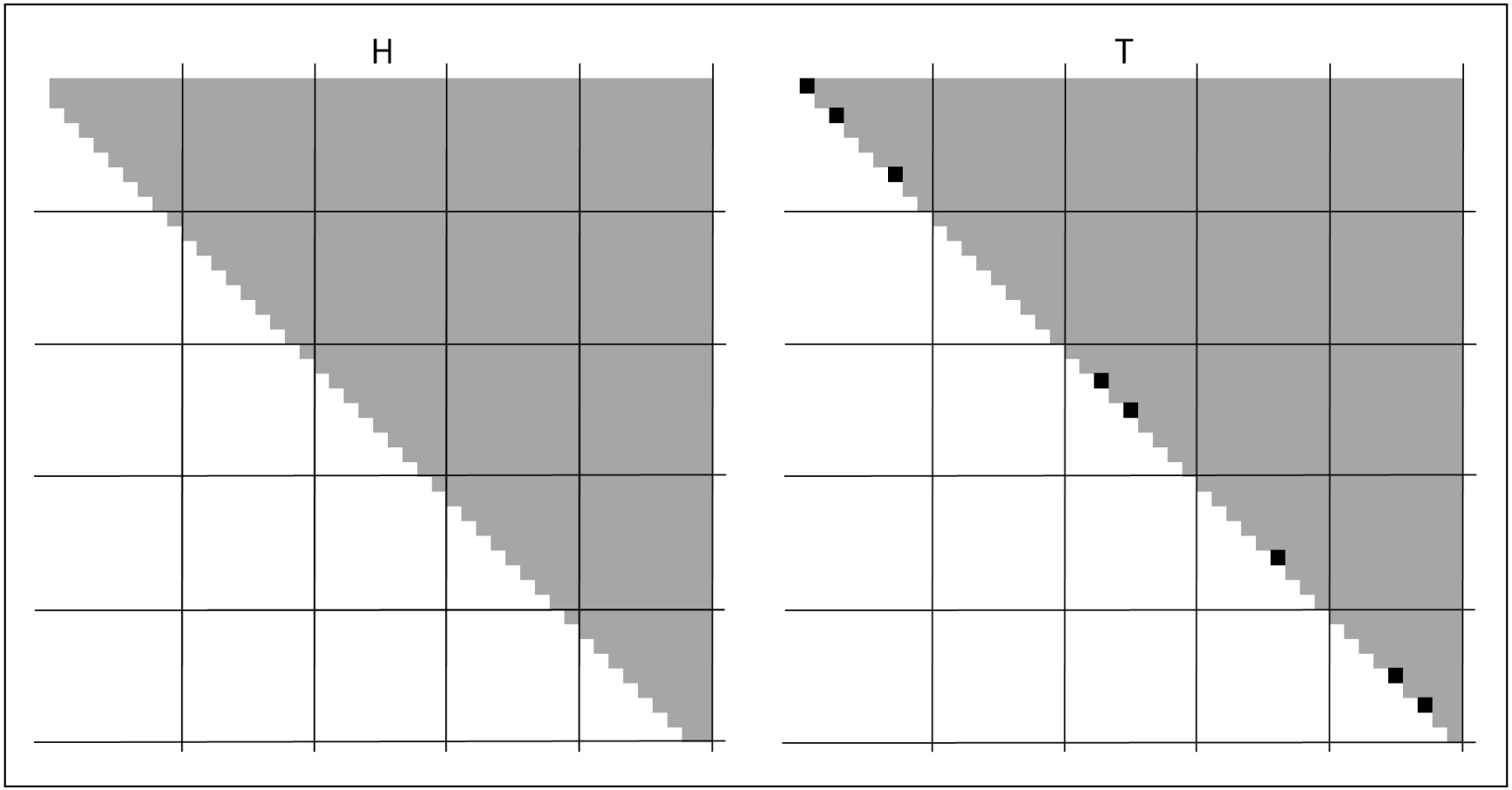
In exact arithmetic, singularity of B implies zero diagonal entry in T:

$$H - \lambda T = \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \end{bmatrix} - \lambda \begin{bmatrix} \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & \mathbf{x} & \mathbf{x} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & \mathbf{0} & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & \mathbf{x} & \mathbf{x} \\ 0 & 0 & 0 & 0 & \mathbf{x} \end{bmatrix}$$

Zero can be pushed to one of the corners and deflated.

Use windowing technique for deflating many infinite eigenvalues of a large matrix pair ($\Longrightarrow$ blocked algorithm with delayed updates).

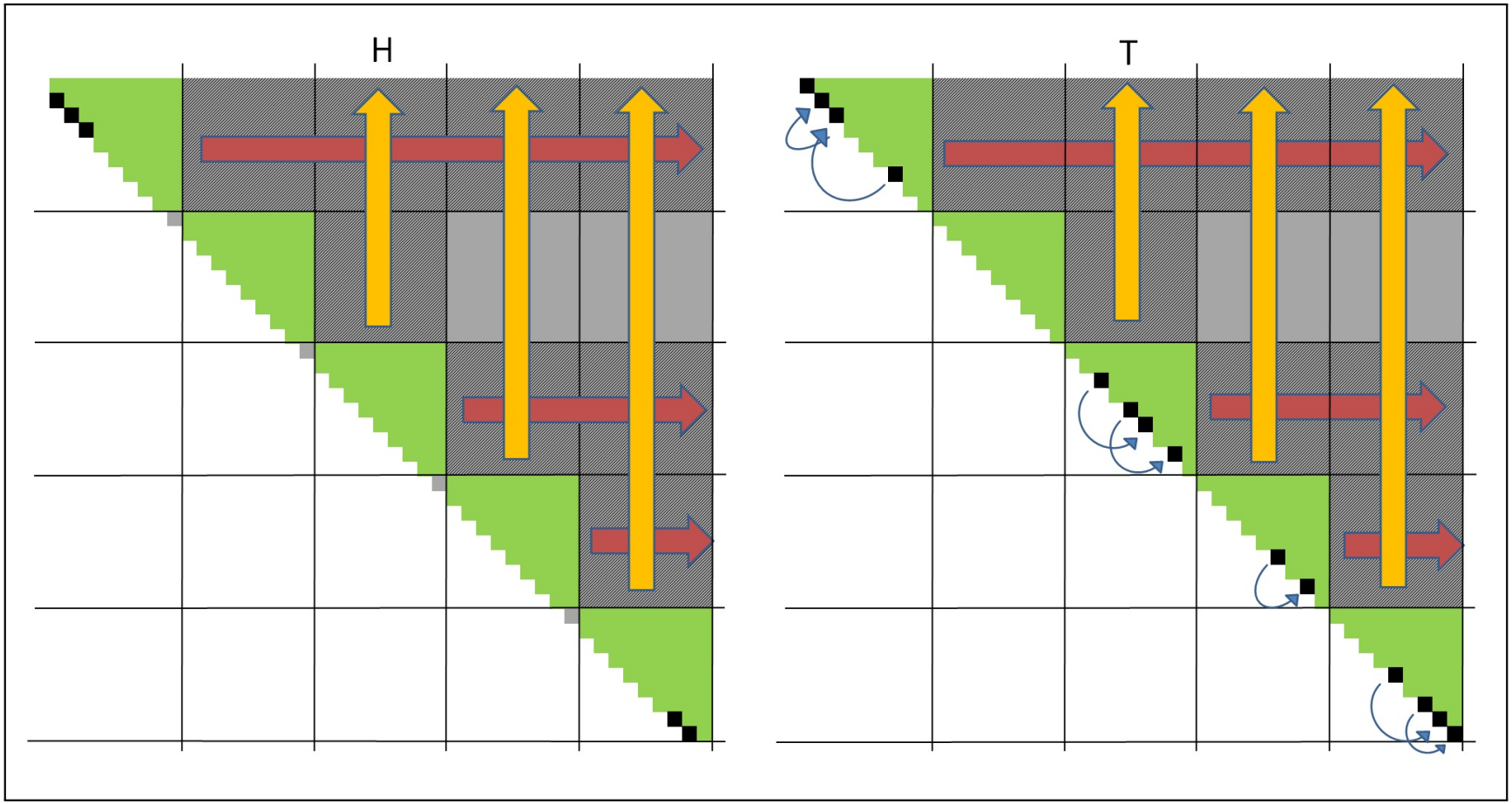
Parallel Deflation of $\lambda = \infty$



Identification of 0-elements in the diagonal of T (black squares).

Assume unreduced H: $h_{i+1,i} \neq 0$

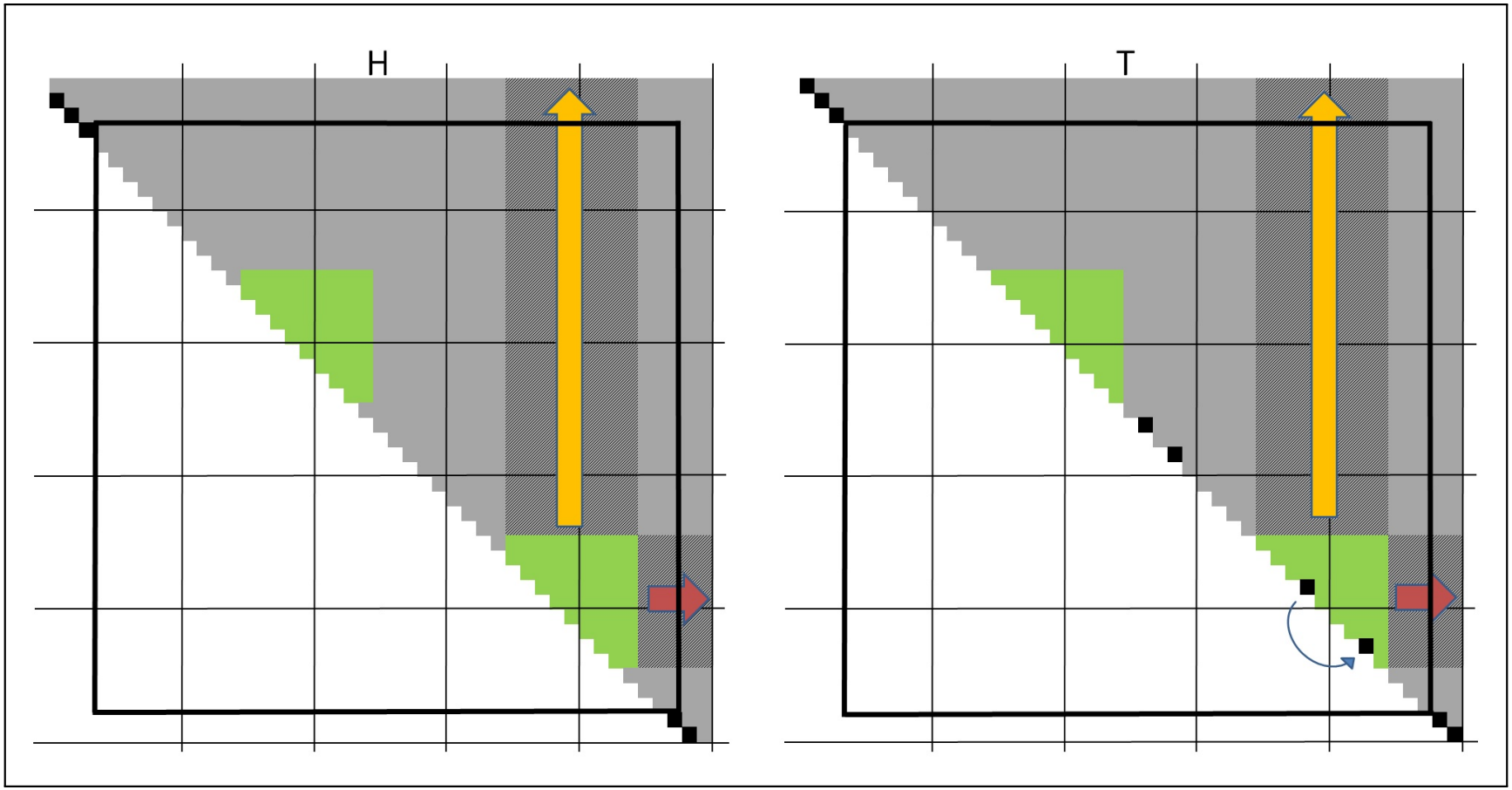
Parallel Deflation of $\lambda = \infty$



Intra-block chasing of zeros in T: Broadcasts horizontal and vertical followed by // updates of off-diagonal blocks in (H, T).

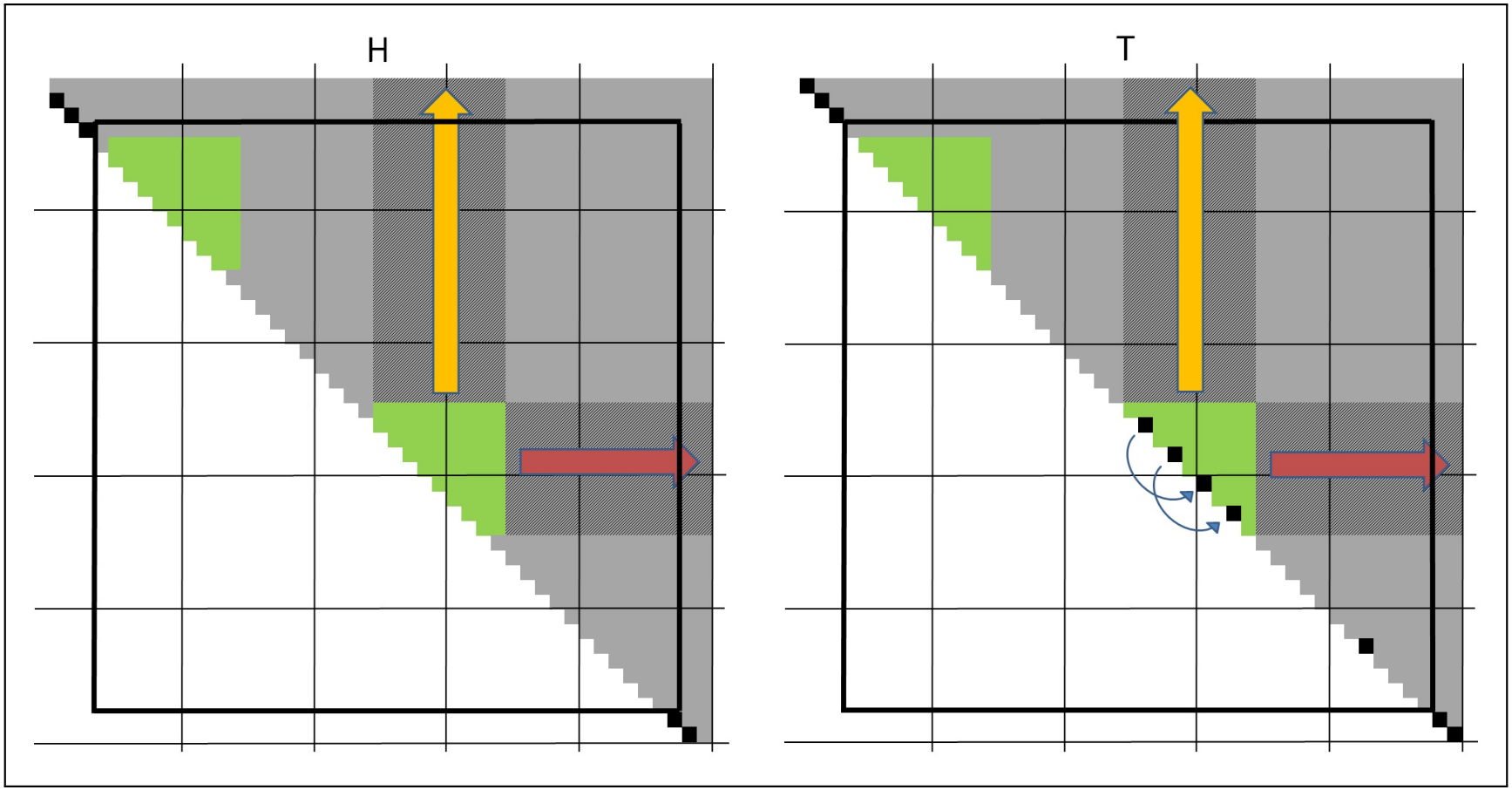
$h_{2,1} = h_{3,2} = h_{4,3} = h_{n-1,n-2} = h_{n,n-1} = 0$ and
$t_{1,1} = t_{2,2} = t_{3,3} = t_{n-1,n-1} = t_{n,n} = 0$

Parallel Deflation of $\lambda = \infty$



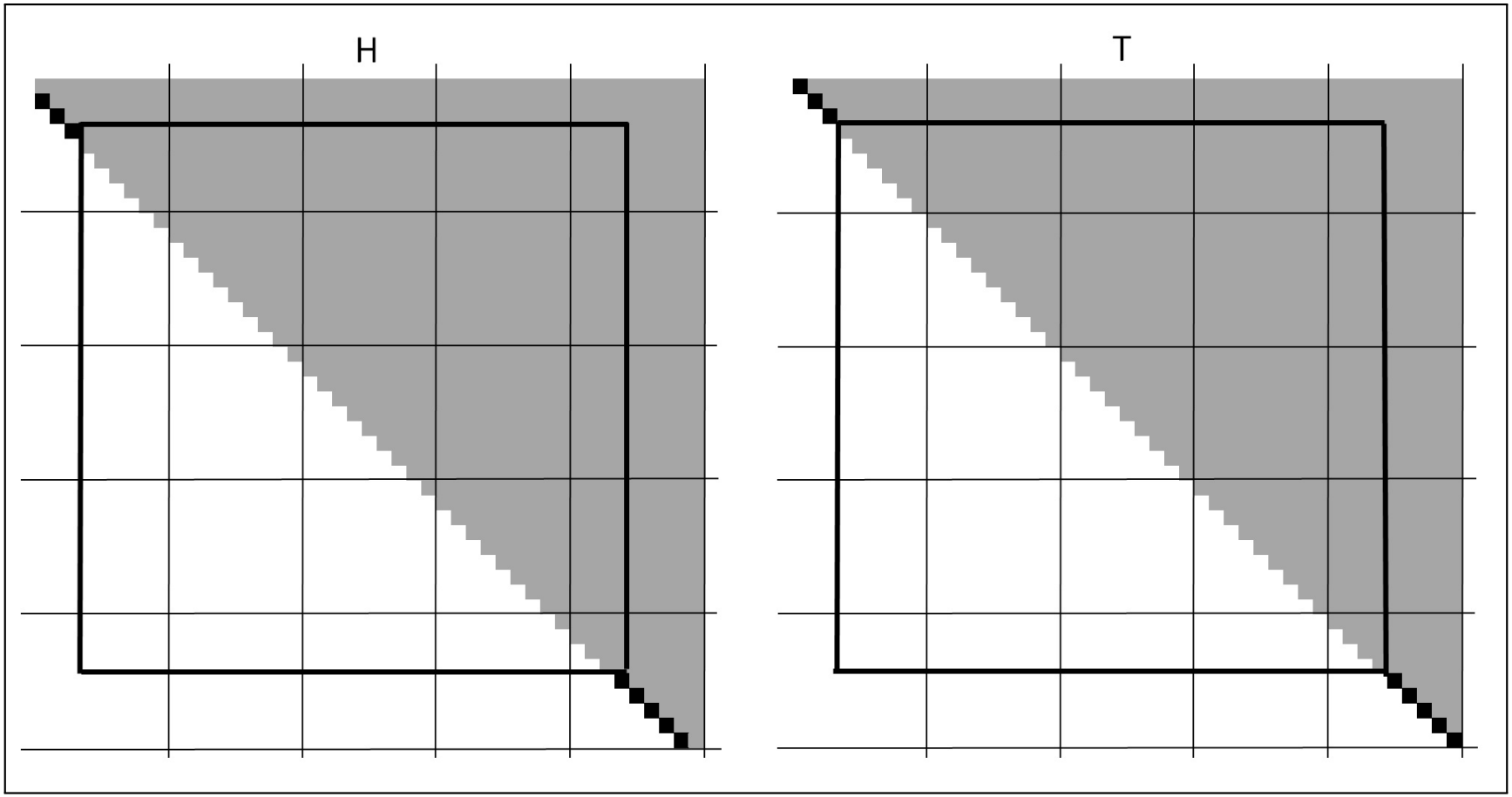
Inter-block chasing of zeros in T: Cross-border chasing for odd-numbered windows. Broadcasts horizontal and vertical followed by // updates of off-diagonal blocks in (H, T).

Parallel Deflation of $\lambda = \infty$



Inter-block chasing of zeros in T: Cross-border chasing for even-numbered windows. Broadcasts horizontal and vertical followed by // updates of off-diagonal blocks in (H, T).

Parallel Deflation of $\lambda = \infty$



Chasing of zero diagonal elements of T completed.

Perform (initial) deflation of infinite eigenvalues ($8 = 3 + 5$)

Beware of Infinite Eigenvalues

In finite-precision arithmetic these zero diagonal entries can be severely perturbed, leading to the $\sqrt{\infty}$-effect.

Example:

$$A - \lambda B = Q^T \left(\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \\ 1 & 3 & 3 & 3 & 3 \\ 0 & 1 & 3 & 3 & 3 \\ 0 & 0 & 1 & 3 & 3 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right) Z$$

with random orthogonal matrices Q and Z.
Two infinite eigenvalues but QZ algorithm computes nonzero pairs (s_{ii}, t_{ii}) with ratio $s_{ii}/t_{ii} \approx \pm 6 \times 10^7 \sqrt{-1}$.

Beware of Infinite Eigenvalues

In finite-precision arithmetic these zero diagonal entries can be severely perturbed, leading to the $\sqrt{\infty}$-effect.

Example:

$$A - \lambda B = Q^T \left(\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \\ 1 & 3 & 3 & 3 & 3 \\ 0 & 1 & 3 & 3 & 3 \\ 0 & 0 & 1 & 3 & 3 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \right) Z$$

with random orthogonal matrices Q and Z.
Two infinite eigenvalues but QZ algorithm computes nonzero pairs (s_{ii}, t_{ii}) with ratio $s_{ii}/t_{ii} \approx \pm 6 \times 10^7 \sqrt{-1}$.

Preprocessing Deflation of $\lambda = \infty$

Exploiting staircase algorithms (e.g., GUPTRI):

$$U^T(A,B)V = \left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{\text{inf}} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{\text{inf}} \end{bmatrix} \right)$$

- U and V orthogonal.
- $(A_{\text{inf}}, B_{\text{inf}})$ reveals the Jordan structure of the infinite eigenvalue.
- (A_{11}, B_{11}) has only finite eigenvalues.

Preprocessing Deflation of $\lambda = \infty$

Exploiting staircase algorithms (e.g., GUPTRI):

$$U^T(A,B)V = \left(\begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{\text{inf}} \end{bmatrix}, \begin{bmatrix} B_{11} & B_{12} \\ 0 & B_{\text{inf}} \end{bmatrix} \right)$$

- U and V orthogonal.
- $(A_{\text{inf}}, B_{\text{inf}})$ reveals the Jordan structure of the infinite eigenvalue.
- (A_{11}, B_{11}) has only finite eigenvalues.

Library software for computing generalized Schur forms of regular $A - \lambda B$

Step	**LAPACK**	**ScaLAPACK** ($\emptyset$)
0: Balancing	xGGBAL	PxGGBAL
1: $(A, B) \rightarrow (H, T)$	xGGHRD	PxGGHRD
2: $(H, T) \rightarrow (S, T)$	xHGEQZ[1]	PxHGEQZ
3: $(S, T) \rightarrow (S_{\text{ord}}, T_{\text{ord}})$	xTGESEN	PxTGESEN[2] PxTGORD

GEP contributions are ongoing work!

[1] No multishift, no AED!

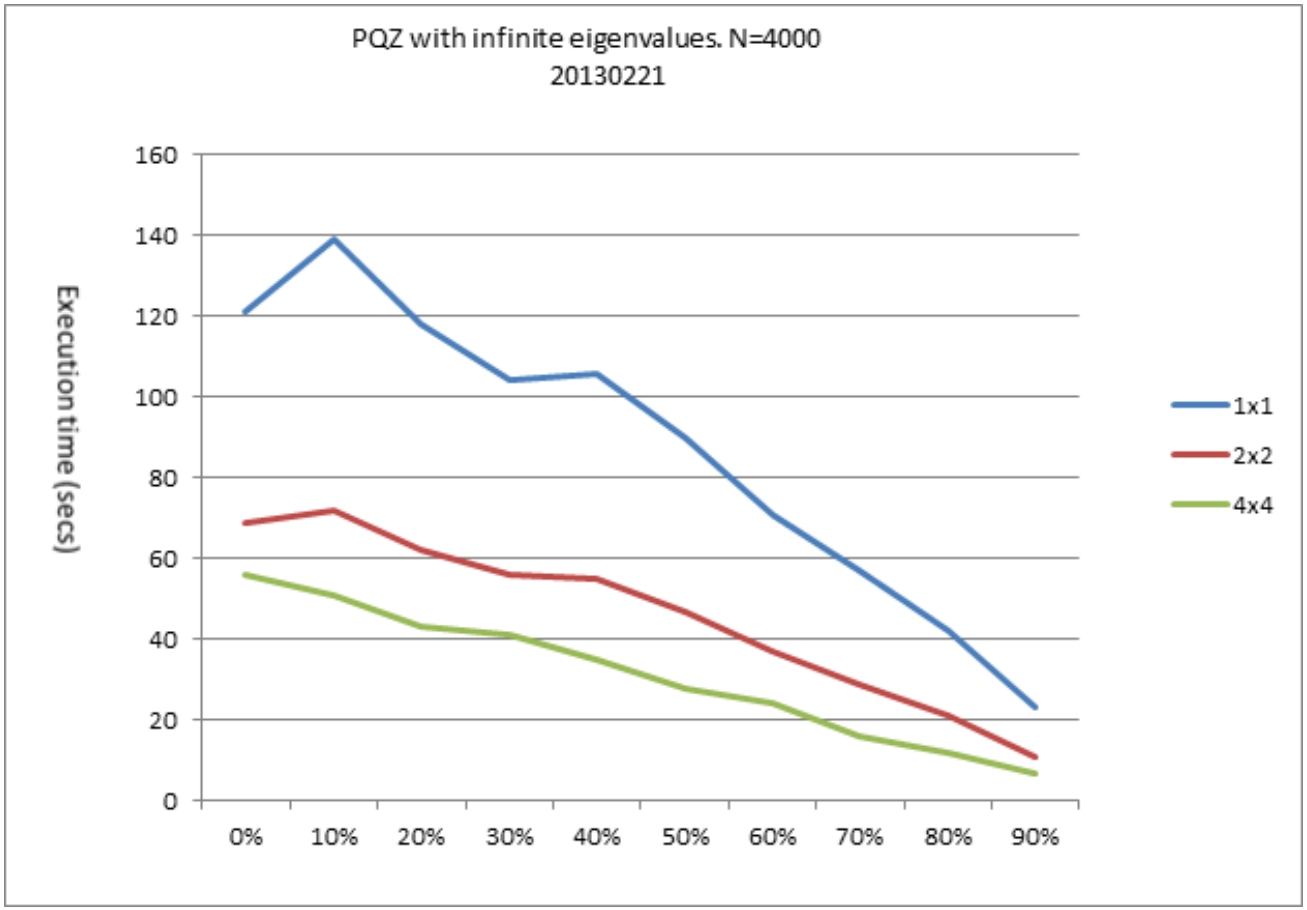
[2] Use parallel routines of SCASY/RECSY—matrix equation solvers

Computational experiments

- Target parallel systems (*abisko*, *akka*)
- Radom matrices—fullrand (A, B), hessrand (H, T)
- Benchmark examples (AKK/BBM, Matrix Market, NEP collection)

Wanted! Large scale (dense) generalized eigenvalue problems!

Varying number of infinite eigenvalues of random H, T



Execution times for hessrand 4000x4000 with varying number of infinite eigenvalues.

PDHGEQZ execution times (in secs) – fullrand (A, B)

PDHGEQZ: parallel multishift QZ algorithm with AED

- two AED-levels, data redistribution, dynamic NIBBLE

$P_r \times P_c$	$n =$ 4 000	8 000	16 000	32 000
1×1	125 (1.1)			*Fullrand*
2×2	73 (1.1)	476 (1.7)		
4×4	48 (1.2)	251 (1.5)	1282 (1.7)	
6×6	45 (1.0)	161 (1.4)	723 (1.6)	
8×8	41 (1.1)	145 (0.9)	602 (1.6)	2640 (1.5)
10×10	40 (0.9)	130 (1.1)	537 (1.8)	2050 (1.5)

(Time PDHGEQZ / Time PDHSEQR)

 # flops (sequential): QZ $\approx 3.5\times$ QR ($Q^T(A,B)Z = (S,T)$)

Fullrand // timings: PDHGEQZ $\approx 0.9 - 1.8\times$ PDHSEQR

PDHGEQZ execution times (in secs) – $100K \times 100K$ hessrand (A, B)

PDHGEQZ: parallel multishift QZ algorithm with AED

- two AED-levels, data redistribution, dynamic NIBBLE
- without tests for infinite eigenvalues

$P_r \times P_c =$	16×16	24×24	32×32
T_{AED}	86%	74%	54%
T_{Sweep}	14%	26%	46%
Total time	25185(2.6)	12913(1.8)	10512(1.7)
#AED	25(0.7)	25(0.8)	33(1.3)
#Sweeps	2(0.5)	4(0.7)	7(0.7)
#Shifts/n	0.07(0.4)	0.13(0.6)	0.27(0.9)

Ratios (PDHGEQZ / PDHSEQR)

Hessrand // timings: PDHGEQZ $\approx 1.7 - 2.6\times$ PDHSEQR

Acknowledgements

- All co-workers and the Umeå research group.
- Rodney James and Julien Langou, UC Denver.
- This research was conducted using the resources of the High Performance Computing Center North (HPC2N).
- Financial support has been provided by the *Swedish Research Council* under grant VR 70625701, the *Swedish Foundation for Strategic Research* under grant A3 02:128, and the eSSENCE Strategic Research Programme.

Abstract PMAA 2012

Key techniques used in our novel parallel QR and QZ algorithms include multi-window bulge chain chasing and distributed aggressive early deflation (AED), which enable level-3 chasing and delayed update operations as well as improved eigenvalue convergence. Mixed MPI-OpenMP coding techniques are utilized for DM platforms with multithreaded nodes, such as multicore processors. Recent progress includes a multi-level recursive approach for performing AED in a parallel environment leading to communication avoiding algorithms via data redistribution. A new performance model of our parallel QR algorithm is presented together with our library software available as part of ScaLAPACK version 2.0. Application and test benchmarks confirm the superb performance of our parallel implementations.

Target parallel systems@HPC2N

akka 64-bit low power Intel Xeon Linux cluster, 672 dual socket quadcore L5420 2.5GHz nodes, 256KB dedicated L1 cache, 12MB shared L2 cache, 16GB RAM per node, Cisco Infiniband and Gigabit Ethernet, 10 GB/sec bandwidth.

abisko 64-bit AMD Opteron L238 12 cores (2.6 GHz) Linux Cluster; (Interlagos)[1 socket 12 cores, 1 NUMA island = 6 cores, 1 module = 2 cores w. common FPU], 48 cores nodes, 15264 cores, 128 GB RAM (312 nodes), 512 GB RAM 8GB (10 nodes), Infiniband QDR - 40Gb/s, 160+ Tflops.

Compilers and software

akka OpenMPI 1.2.6, BLACS 1.1patch3, GOTO BLAS r1.26, LAPACK 3.1.1, ScaLAPACK/PBLAS 1.8.0

sarek MPICH-GM 1.5.2, BLACS 1.1patch3, GOTO BLAS r0.94, LAPACK 3.1.1, ScaLAPACK/PBLAS 1.7.0

On both systems, we use the Portland Group Compiler Suite. ONLY for SISC-paper! Now, Pathscale compiler is used.

Accuracy metrics

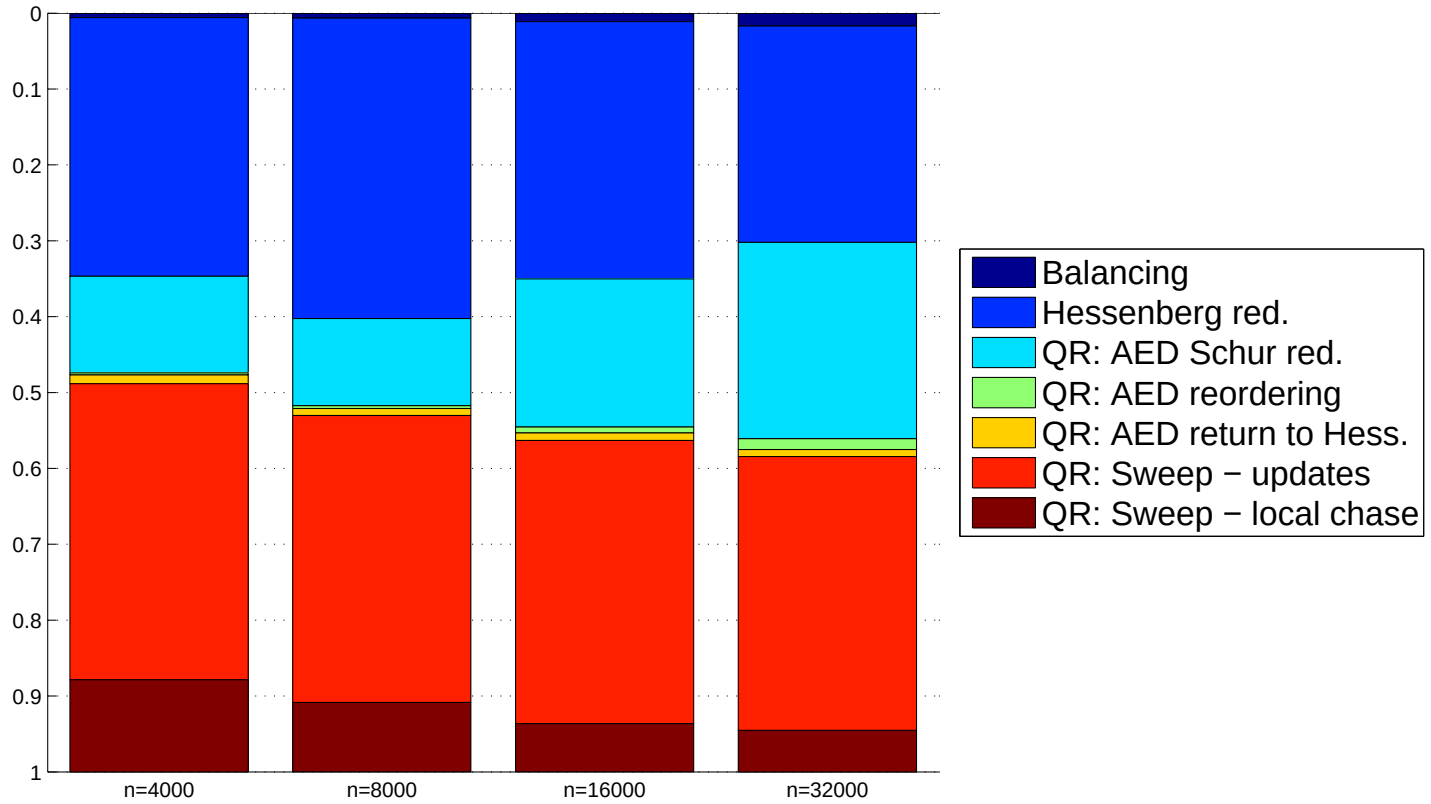
- To validate the output, we compute the following accuracy measures
  - Relative residual norm:

$$R_r = \frac{\|Q^T AQ - S\|_F}{\|A\|_F}$$

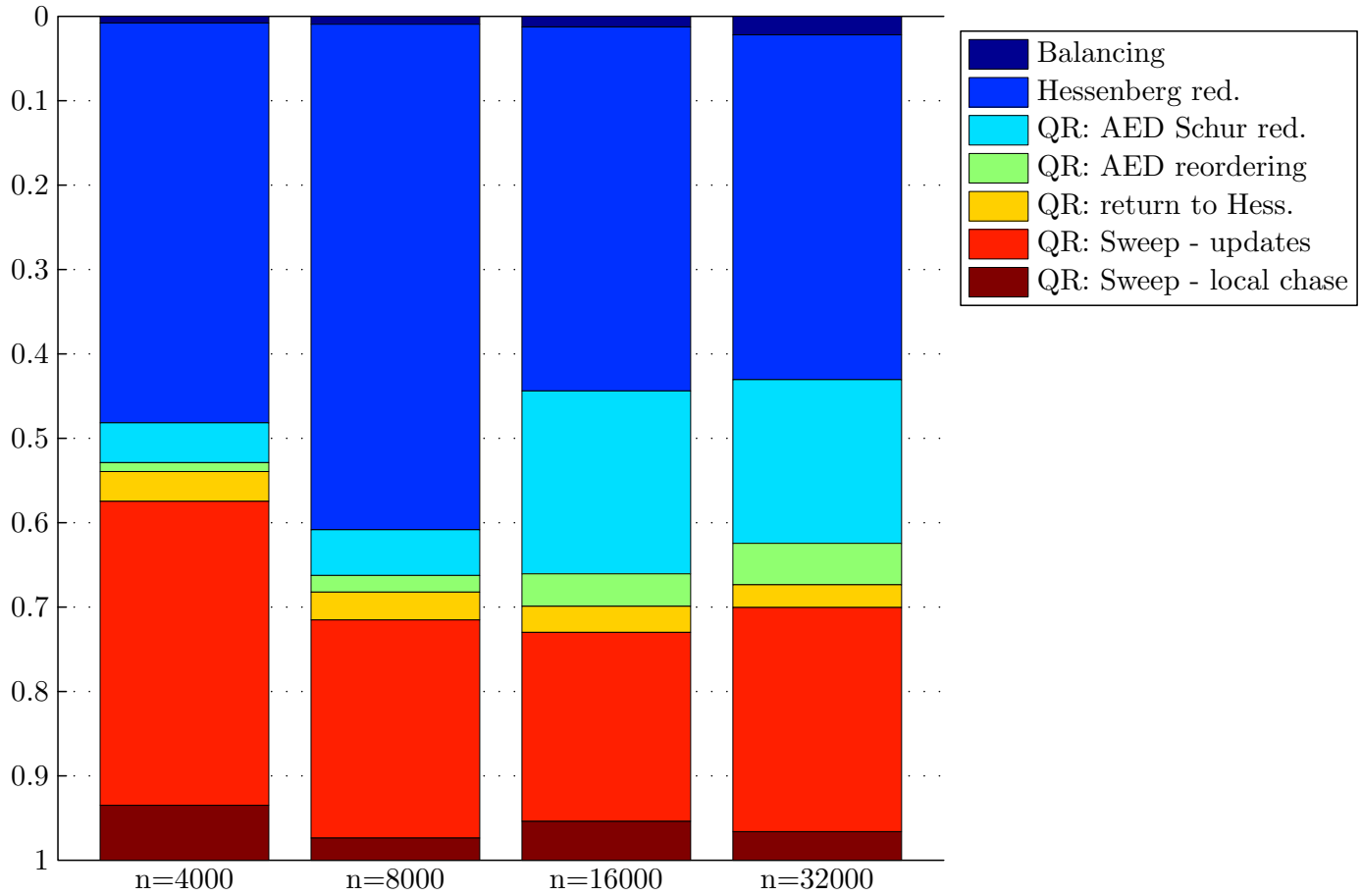
  - Relative orthogonality check:

$$R_o = \frac{\max(\|Q^T Q - I_n\|_F, \|QQ^T - I_n\|_F)}{\mathbf{u}n}$$

- Both ScaLAPACK and novel algorithm perform well on R_o, while our new parallel implementation is usually slightly better for R_r.



Profile of execution time for SISC implementation; memory load corresponds to a $4\,000 \times 4\,000$ submatrix per core.



Profile of execution time for ScaLAPACK 2.0.1 implementation; memory load corresponds to a $4\,000 \times 4\,000$ submatrix per core.

Choice of algorithm parameters

- $P_r \times P_c$: process mesh size; $P_r = P_c$
- nb: data distribution block size; optimal on a few cores:
  - *akka*: $nb = 50$
  - *sarek*: $nb = 160$
- n_w: size of deflation window; varies with problem size n
- n_s: number of shifts; $= 2n_w/3$ (min 10)

n:	590-3000	–6000	–12000	–24000	–48000	–96000	$>$ 96000
n_w:	96	192	384	768	1536	3072	6144
n_s:	64	128	256	512	1024	2048	4096

- `NIBBLE`: dynamic (fixed 14% in LAPACK)
  - If AED detects a high fraction of eigenvalues in the deflation window to be converged, it can be beneficial to skip subsequent QR sweep and perform AED once again on a suitably adjusted deflation window.