Parallel, Blocked and Multishift Variants of the QZ Algorithm for Regular Matrix Pairs

Björn Adlerborn and Bo Kågström
{c95ban, bokg}@cs.umu.se
Dept of Computing Science and HPC2N
Umeå University, Sweden

Daniel Kressner
Institut für Mathematik, Technische Universität Berlin
kressner@math.tu-berlin.de

SIAM Applied Linear Algebra ’03, July 15-19, Williamsburg
Motivation (1)

- Want to solve large-scale dense nonsymmetric generalized eigenvalue problems

\[ Ax = sBx \]

on today's HPC systems with deep memory hierarchies
Motivation (2)

Typically, we want

- all eigenvalues and eigenvectors
- or a subset of the eigenvalues with associated left and right eigenspaces
- high accuracy results

Our approach:

- orthogonal transformation methods
- blocked and parallel (distributed) algorithms ("strive for level 3")
Regular Nonsymmetric $Ax = sBx$

Transform $(A, B)$ to generalized Schur form $(S, T)$ using orthogonal equivalence transformations

$S$ quasi-triangular (1x1 and 2x2 blocks on the diagonal)
$T$ upper triangular

- **Real eigenvalues** are given by the diagonal elements $(s_{ii}, t_{ii})$
  - Finite eigenvalues $s_{ii}$ / $t_{ii}$
  - Infinite eigenvalues represented as $(s_{ii}, 0)$

- 2x2 blocks correspond to complex conjugate eigenvalue pairs

Solve Matrix Pencil Systems for several r.h.s. and s

$(A - sB)x = y \iff (H - sT)u = v$ Hessenberg
Outline

- Three-stage reduction to generalized Schur form
  
  \[ (A, B) \rightarrow (H_r, T) \rightarrow (H, T) \rightarrow (S, T) \]

- Stage 1 & 2: Blocked and Parallel Algorithms
- Stage 3: Blocked QZ Algorithm
- Multishift QZ variants
- Aggressive Early Deflation in QZ
- Computational Experiments
- Stage 3: Parallel QZ Algorithm
- Summary and Some References
Unblocked Solution
(presently in LAPACK, based on Moler-Stewart’72)

Original regular $N \times N$ matrix pair $(A, B)$

Triangularize $B$, blocked QR factorization of $B$ and update of $A$ using level 3 operations

**Unblocked** reduction to Hessenberg-triangular form $(H, T)$ by application of Givens rotations to $A$ and $B$ from the left and right hand sides

**Unblocked** reduction to generalized Schur form $(S, T)$:
apply the QZ algorithm iteratively to $(H, T)$
(Givens rotations and Householder transformations of length 3)
Blocked and Parallel Solution

(Dackland-Kågström’99, Adlerborn-D-K’00 and ’02)

Original regular $N \times N$ matrix pair $(A, B)$

Triangularize $B$, blocked QR factorization of $B$ and update of $A$ using matrix-matrix-based operations

**Blocked** two-stage reduction to Hessenberg-triangular form $(H_r, T)$

1: level 3 based reduction to $(H_r, T)$ form
2: blocked reduction of $(H_r, T)$ to $(H, T)$

$H_r$ has $r$ subdiagonals

**Blocked** reduction to generalized Schur form $(S, T)$

Apply blocked QZ sweep iteratively to $(H, T)$
Basic \((H, T)\) Form Reduction

- Two-sided orthogonal transformations to reduce \(A\) to \textbf{Hessenberg form} while preserving \(B\) triangular
  - Givens or (2x2) Householder transformations

\[
\begin{align*}
A_{41} &:= 0 \\
B_{43} &:= 0 \\
Q_{34}A & \\
Q_{34}B & \\
AZ_{34} & \\
BZ_{34} &
\end{align*}
\]
Topology and Data Layout

*ScalAPACK setting:* BLAS, LAPACK, BLACS, PBLAS, MPI etc

- Data distribution of $A$ and $B$ on a 2x2 processor grid

$P_{r \times c}$ processor grid          Square block scattered (cyclic, $rxr$)
**Blocked Stage 1 Reduction:**

From $(A, B)$ to $(H_r, T)$

- Partition the matrices $A$ and $B$ into $r \times r$ blocks

- Orthogonal matrix $U$ triangularizes $B$
  - $U^T B = R$, upper triangular,
  - Apply $U$ to $A$: $A \leftarrow U^T A$
  - Use blocked level 3 QR factorization
  - Updates performed using Level 3 operations

- Orthogonal equivalence transf. reduces $A$ to block upper Hessenberg ($H_r$) form while preserving $B$ triangular
  - Updates performed using Level 3 operations
  - Annihilations performed using Level 1-2 ops.
**Parallel Stage 1:** Level 3 Alg. \((r, p)\)

- Annihilate \(A_{4,1}\) and triangularize \(A_{3,1}\):
  - QR fact \(A_{3:4,1} \leftarrow U^T A_{3:4,1}\)
  - Update \(A_{3:4,2:4} \leftarrow U^T A_{3:4,2:4}\)
  - Update \(B_{3:4,3:4} \leftarrow U^T B_{3:4,3:4}\)

- Re-triangulization of \(B_{3:4,3:4}\):
  - RQ fact \(B_{3:4,3:4} \leftarrow B_{3:4,3:4} V\)
  - Update \(B_{1:2,3:4} \leftarrow B_{1:2,3:4} V\)
  - Update \(A_{3:4} \leftarrow A_{3:4} V\)

- Annihilate \(A_{3,1}\) and triangularize \(A_{2,1}\):
  - Re-triangulization of \(B_{2:3,2:3}\)

- First block column of \(A\) reduced to block Hessenberg form
- *The remaining columns of \(A\) reduced in a similar way*
- \(r = \text{block size} \ p = \#\text{blocks reduced per iteration} \ (p = 2 \ \text{min fill-in})*)}
Blocked Stage 2 Reduction:
From \((H_r, T)\) to \((H, T)\)

Illustration of blocked sweep reducing 3 elements \((r = 4)\) in column 1

“Spiral Blocked Reference Pattern”

Reduce
1. Annihilate 3 elements:
2. Update w.r.t. (1); zero fill-in: col1
3. Update w.r.t. (2)
4. Update w.r.t. (1, 2)
5. Update w.r.t. (5); zero fill-in: row2
6. Update w.r.t. (1, 6)
7. Update w.r.t. (1, 5, 6)
8. Update w.r.t. (1, 5, 6)
9. Update w.r.t. (1, 5, 6)
10. Update w.r.t. (1, 5, 6, 10)
11. Update w.r.t. (1, 5, 6, 10)
12. Update w.r.t. (1, 5, 6, 10)
Generalization to **Super-Sweep** - reducing $m$ columns

**Reduction:**
Reduce $m$ columns of $H_r$; updates of $H_r$ and $T$ are restricted to $r$ consecutive columns at a time.
Store **row** and **column** rotations to enable delayed updates.

**Chasing:**
A super-sweep advances the $m$ sweeps of one block column ($r$ cols) ahead in a **pipelined fashion**, starting with the leading block.

⇒ Reduced # loads; improved data reuse
Parallel Stage 2
partial supersweep \((m = 1, r = 4)\)

1: Zero el’s (in \(A\)) and broadcast
2: Update and zero el’s (in \(B\))

3: Broadcast
4: Update and zero el’s (in \(A\))
Apply $r-1$ row rotations to $B$ (in step 2)

Zero fill-in across processor borders

- **a:** Exchange rows
  - Update row $1_1$

- **b:** Annihilate fill-in col $1_1$
  - Update columns

- **c:** Next row rotation applied row $1_2$

- **d:** Exchange columns
  - Annihilate fill-in col $1_2$
Parallel Performance-Scaled Speedup Configuration

- Fixed local matrix size (1024 x 1024 entries)
- Vary block size $r = NB$
- Vary grid configuration ($P = P_r \times P_c$)
- $p = \max(2, P_r)$ (# blocks reduced/iter. Stage 1)
- Fixed $m = 2$ (supersweep - Stage 2)

Stage 1 + Stage 2 = P_{DGGHRD} - reduction to (H, T) form
Parallel Stage 1 + Stage 2 Implementation - Summary

- New ScALAPACK-style algorithms for $(A, B) \rightarrow (H, T)$
- **Stage 1** speedup $\approx k \times \sqrt{P} (= P_r \times P_c)$
  - Mainly Level 3 operations
- **Stage 2** speedup $\approx \sqrt{P}$
  - Level 1 - 2.5 operations ("fewer flops")
  - More costly communication
- **Stage 1 + 2** speedup $> \sqrt{P}$
  - Overall performance limited by Stage 2
- Performance increases with problem and grid sizes

=> Can solve large scale problems!
QZ Algorithm Overview

- A sequence of single or double shift QZ steps applied to matrix pair in $(H, T)$ form:
  \[ (H, T) \leftarrow Q^T (H, T) Z \]

- Implicit shifts are used for accelerating convergence
  - Eigenvalues of the trailing 2 x 2 block

- QZ step:
  - Introducing an implicit shift (single or double)
  - Bulge chasing of unwanted elements along subdiagonals of $H$ and $T$

- Deflation: Problem splits into two subproblems - one with converged eigenvalues

  Monitoring and decoupling assoc. with deflation surrounds a QZ step.
Deflation Strategies

Small (sub)diagonal convergence criterion:
If $h(i+1,i)$ or $t(i,i)$ gets tiny, it is set to 0!

- **Norm-wise:**
  \[ |h_{i+1,i}| \leq \text{eps} \cdot \| H \|_F \]
  Used by EISPACK (QZIT), LAPACK (DHGEQZ), Dackland-Kågström

- **Neighbor-wise:**
  \[ |h_{i+1,i}| \leq \text{eps} \cdot (|h_{ii}| + |h_{i+1,i+1}|) \]
  Used in state-of-the-art QR implementations.
  Our current Multishift QZ implementation uses nearby-diagonal deflation strategy.
Graded Matrix Pair

\[(H, T) = \left( \begin{bmatrix} 10^0 & 10^{-3} & 0 & 0 \\ 10^{-3} & 10^{-7} & 10^{-10} & 0 \\ 0 & 10^{-10} & 10^{-14} & 10^{-17} \\ 0 & 0 & 10^{-17} & 10^{-21} \end{bmatrix}, I_4 \right) \]

<table>
<thead>
<tr>
<th>Exact eigenvalues</th>
<th>Norm-wise deflation</th>
<th>Neighbor-wise deflation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00000099999991000</td>
<td>1.00000099999991001</td>
<td>1.00000099999990999</td>
</tr>
<tr>
<td>−.899999111128208×10⁻⁰⁶</td>
<td>−.899999111128212×10⁻⁰⁶</td>
<td>−.899999111128213×10⁻⁰⁶</td>
</tr>
<tr>
<td>.2111111558732113×10⁻¹³</td>
<td>.2111111085047986×10⁻¹³</td>
<td>.2111111558732114×10⁻¹³</td>
</tr>
<tr>
<td>−.3736841266803067×10⁻²⁰</td>
<td>0.09999999999999999×10⁻²⁰</td>
<td>−.3736841266803068×10⁻²⁰</td>
</tr>
</tbody>
</table>

Results from Kressner’s Diploma Thesis‘01

- **Nearby-diagonal deflation** more accurate for graded matrices.
Multishift QZ

- Compute the eigenvalues of trailing $M \times M$ submatrix pair of $(H, T)$.
- Choose $MS$ of the $M$ eigenvalues as shifts.
  - Choosing $M > MS$ will probably give better shifts but the computation is more expensive.
- Use the shifts in a pipelined fashion
- Avoid shift blurring by restricting the bulge size.
  - Due to rounding errors the eigenvalues of a certain bulge pencil that carries the shift information tend to be ill-conditioned for a large degree multishift (Watkins-Elsner'94, Watkins'96, '98)
- $BS = 2, 4$ up to 8 can be okay!
Pipelined Bulge Chasing

- Introduce $S = MS/2$ bulges of degree two and chase them down the diagonals of $H$ and $T$

Bottom shifts are applied first - most likely to be closest to the eigenvalues that will deflate next.
Multishift Variants

- Pipelining of independent bulges is earlier proposed in the context of parallel QR algorithms (Van de Geijn'93, Watkins'94, Henry-Van de Geijn'97, Lang'98).

- Pipelining of tightly coupled clusters of bulges
  - Recently and successfully applied to the QR algorithm (Braman-Byers-Mathias'02a).

- Aggressive early deflation strategy for speeding up the convergence of the multishift QR algorithm (Braman-Byers-Mathias'02b).
  - Take advantage of perturbations outside the subdiagonal entries of the Hessenberg matrix in the QR iteration.

Variants generalize to matrix pencils and the QZ algorithm.
Chasing of Tightly Coupled Bulges

- \( MS = 4, \; BS = 2, \; S = MS/2 = 2, \) bulges of degree two are chased down within the “red window”.

All transformations created are bundled and applied to “blue areas” using level 3 operations.
Deflating Perturbation Pair

- $(\mathcal{H}, \mathcal{T})$ in unreduced HT-form, $Q$ and $Z$ unitary
- $P_H$ and $P_T$-perturbation matrices such that

$$ (\hat{\mathcal{H}}, \hat{\mathcal{T}}) \equiv Q^H (\mathcal{H} + P_H, \mathcal{T} + P_T) Z $$

is in reduced HT-form:

$$ \hat{\mathcal{H}} = \begin{bmatrix} \hat{H}_{11} & \hat{H}_{12} \\ 0 & \hat{H}_{22} \end{bmatrix} \quad \hat{\mathcal{T}} = \begin{bmatrix} \hat{T}_{11} & \hat{T}_{12} \\ 0 & \hat{T}_{22} \end{bmatrix} $$

If norm of $(P_H, P_T)$ is tiny, problem split in two (or more) subproblems.

Ideally, we want the (2,2)-pair in $(S, T)$ form.
Restricting the Perturbations?! 

\((P_H, P_T)\) only nonzero in last \(k\) rows and \(k+1\) columns:

\[
P_H = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & P_{32}^{(H)} & P_{33}^{(H)}
\end{bmatrix}
\]

\[
P_T = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & P_{33}^{(T)}
\end{bmatrix}
\]

Block rows and columns are of size \(n-k-1, 1,\) and \(k\)

The problem to find the minimal perturbation \((P_H, P_T)\)

is related to the distance to uncontrollability for the descriptor system

\[
T_{33} \dot{x}(t) = H_{33} x(t) + H_{32} u(t)
\]

\[
\text{rank}([H_{32}, \beta H_{33} - \alpha T_{33}]) = n \text{ for all } (\alpha, \beta) \in \mathbb{C}^2 \quad |\alpha|^2 + |\beta|^2 = 1
\]

\(
\implies \text{system is completely controllable}
\)
**Aggressive Early Deflation in QZ (1)**

Consider \((H, T)\) in unreduced HT-form:

\[
H = \begin{bmatrix}
    H_{11} & H_{12} & H_{13} \\
    H_{21} & H_{22} & H_{23} \\
    0 & H_{32} & H_{33}
\end{bmatrix} \quad T = \begin{bmatrix}
    T_{11} & T_{12} & T_{13} \\
    0 & T_{22} & T_{23} \\
    0 & 0 & T_{33}
\end{bmatrix}
\]

Block rows and columns are of size \(n-k-1, 1, \text{ and } k\)

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

**Generalized Schur decomposition of \(k \times k\) \((H_{33}, T_{33})\)**

\[
(\hat{S}_{33}, \hat{T}_{33}) = Q^H (H_{33}, T_{33}) Z
\]
Aggressive Early Deflation in QZ (2)

Equivalence transformation of \((H, T)\):

\[
\begin{bmatrix}
I & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & Q^H
\end{bmatrix}
\begin{pmatrix}
H \\
T
\end{pmatrix}
= \begin{bmatrix}
I & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & Z
\end{bmatrix}
= \begin{pmatrix}
\hat{H} \\
\hat{T}
\end{pmatrix}
\]

where

\[
\hat{H} = \begin{bmatrix}
H_{11} & H_{12} & H_{13}Z \\
H_{21} & H_{22} & H_{23}Z \\
0 & S & \hat{S}_{33}
\end{bmatrix} \quad \hat{T} = \begin{bmatrix}
T_{11} & T_{12} & T_{13}Z \\
0 & T_{22} & T_{23}Z \\
0 & 0 & \hat{T}_{33}
\end{bmatrix}
\]

\[s = Q^H H_{32} \quad (k \times 1) \quad \text{“the k-spike”}\]
Early Deflation?

- If \( m \) of the trailing components of the vector
  \[
s = Q^H H_{32} \quad (k \times 1)
  \]
  are tiny, we can make a deflation with respect to the trailing \( m \times m \) matrix pair:

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

Block rows and columns are of size \( n-k-1, 1, k-m \), and \( m \), respectively.
Example of Early Deflation

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

Adlerborn-Dackland-Kågström’02

- 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 $k = 5$.

- 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 $10^{-10}$

- Set the trailing $m (= 1, 2, 3)$ components of $s$ to zero.
Retransform to HT-form

\[
\begin{align*}
\tilde{H} &= \begin{bmatrix}
H_{11} & H_{12} & \hat{H}_{13} & \hat{H}_{14} \\
H_{21} & H_{22} & \hat{H}_{23} & \hat{H}_{24} \\
0 & 0 & \hat{H}_{33} & \hat{H}_{34} \\
0 & 0 & 0 & \hat{H}_{44}
\end{bmatrix} \\
\tilde{T} &= \begin{bmatrix}
T_{11} & T_{12} & \hat{T}_{13} & \hat{T}_{14} \\
0 & T_{22} & \hat{T}_{23} & \hat{T}_{24} \\
0 & 0 & \hat{T}_{33} & \hat{T}_{34} \\
0 & 0 & 0 & \hat{T}_{44}
\end{bmatrix}
\end{align*}
\]

- **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:

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

Rank-1 perturbation of an upper triangular matrix

- Apply RQ-updating, \(2(k - m - 1)\) rotations \(\rightarrow Z\)

- **Transform** \((Q^T \hat{H}_{33}Z, Q^T \hat{T}_{33}Z)\) to HT-form using standard algorithm
Computational Experiments

\[(H, T) \rightarrow \text{Stage 3} \rightarrow (S, T)\]

Report on results from testing of Multishift variants (prel.) of the QZ algorithm

Parameters in Multishift QZ

- \(MS\) = multishift size
- \(BS\) = \# shifts/bulge (used in pipelined chasing)
- \(NS\) = \# steps the bulges are chased along the block diagonals = \(3*MS/2 - 2\)
- \(WS\) = window size for aggressive deflation = \(3*MS/2\)

Comparisons with

- LAPACK
- Blocked QZ (Dackland-Kågström’99)

Computer: HPC2N chips

- 4 Power3 (375MHz)
- 4GB of memory
Ex1: Random \((A, B)\) reduced to HT-form
QZ variants w/o Aggressive Deflation

1. LAPACK “dashed line”
   (upper - without, lower - with)

2. Blocked (Dackland-Kågström, TOMS’99) “solid line”

3. Multishift - tightly coupled 2-by-2 bulges

4. Multishift + Aggressive Early Deflation

AD-windows:
1. 40
2. 30 (\(NB = 80\))
3. 3*\(MS/2\)

Time: \((S, T), Q, and Z, \ N = 1000\)

Multishift size
\(MS = 4:4:200\)

3. Best \(MS = 20, 32, 36\)
4. Best \(MS = 68, 72, 76\)
Ex1: Random \((A, B)\) red. to HT-form
QZ variants w/o Aggressive Deflation

1. LAPACK “dashed line”
   (upper - without, lower - with)

2. Blocked (Dackland-Kågström, TOMS’99) “solid line”

3. Multishift - tightly coupled
   2-by-2 bulges

4. Multishift + Aggressive
   Early Deflation

AD-windows:
1. 50
2. 30 \((NB = 72)\)
3. \(3 \times MS/2\)
4. Best \(MS = 28, 32, 36\)
5. Best \(MS = 80, 88, 92\)
Cost for HT- and ST-reductions

Ex 1: Random $(A,B)$  \( N = 1500 \)

Stage 1 & Stage 2: Dackland-Kågström blocked

Stage 3: Multishift + Aggressive Early Deflation
Varying the Bulge Size in the Pipeline Chasing

Ex 1: Random \((A,B)\)  \(N = 1500\)  \(WS = 90\)

Time: \((S,T), Q,\) and \(Z\)

Risk for shift blurring!  
(Watkins-Elsner’94, Watkins’96)
Ex2: \((H_6, T_6)\) for varying \(N\)

1. LAPACK “dashed line” (without aggressive deflation)

2. Blocked (Dackland-Kågström, TOMS’99) “solid line”

3. Multishift - tightly coupled 2-by-2 bulges

4. Multishift + Aggressive Early Deflation

AD-windows:
1. -
2. - \((NB = 48)\)
4. \(3*MS/2\) \((MS = 40)\)

Time: \((S, T), Q,\) and \(Z\)

Problem size \(N = 100:100:2000\)

4. Two orders of magnitude better!
Topography and Data Layout

**ScalAPACK setting:** BLAS, LAPACK, BLACS, PBLAS, MPI etc

- Data distribution of \( A \) and \( B \) on a 2x2 processor grid

\[
\begin{array}{cc}
0 & 1 \\
0 & P_r \times P_c \text{ processor grid} \quad \text{Square block scattered (cyclic, } r \times r) \\
1 & \\
\end{array}
\]

\[
\begin{array}{cccc}
A & & & \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 \\
\end{array}
\]
Stage 3 in Parallel Setting

• Choose #shifts $MS$ large enough, and introduce the bulges with enough space between them

workload can be evenly distributed across the 2D processor grid.

Currently: One bulge per diagonal block
To come:
• One tightly cluster of bulges per diagonal block
• Aggressive early deflation

Parallel Multishift QZ prototype works!
• Accurate results
• Shows speedup with increasing #shifts
Summary and Ongoing Work

Stage 1 \[ (A,B) \rightarrow (H_r,T) \]
Stage 2 \[ (H,T) \rightarrow (S,T) \]

- Efficient blocked and parallel algorithms and software for Stage 1 and Stage 2 completed. Blocked Stage 3 as well.
- Stage 3: New Multishift QZ with aggressive early deflation. Variants are implemented and tested - good results!
- Parallel Multishift QZ prototype implemented!
- Combining the parallel and blocked Stage 1-3 reductions enable effective computation of the generalized Schur form of a regular matrix pair \((A,B)\).
- This work is essential since the reduction to generalized Schur form is a fundamental operation in large-scale control applications.
- Our software is designed for integration in state-of-the-art libraries such as LAPACK, ScaLAPACK and SLICOT.
Some of Our References


B. Kågström and D. Kressner (2003) Multishift Variants of the QZ Algorithm with Aggressive Early Deflation (in prep.)