An implicit multishift *QR*-algorithm for Hermitian plus low-rank matrices

Gianna M. Del Corso
joint work with Raf Vandebril

Dipartimento di Informatica, Università di Pisa, Italy

1 The Problem

2 An $\mathcal{O}(n)$ representation

3 The algorithm

4 Deflation

5 Numerical Experiments

6 Conclusion

The Problem

- Eigenvalues computation of structured rank matrices
- QR on structures
- Keyword is representation preserved by QR-steps

The Problem

Efficient computation of the eigenvalues of a Hermitian plus a low rank correction

- Perfectly Hermitian structure perturbed by a low rank correction
- Comrade matrices
- Hamiltonian-like matrices

The Problem

Efficient computation of the eigenvalues of a Hermitian plus a low rank correction

- Perfectly Hermitian structure perturbed by a low rank correction
- Comrade matrices
- Hamiltonian-like matrices

The Problem

Efficient computation of the eigenvalues of a Hermitian plus a low rank correction

- Perfectly Hermitian structure perturbed by a low rank correction
- Comrade matrices
- Hamiltonian-like matrices

The Problem

Efficient computation of the eigenvalues of a Hermitian plus a low rank correction

- Perfectly Hermitian structure perturbed by a low rank correction
- Comrade matrices
- Hamiltonian-like matrices

Givens-weight representation

We will represent matrices by means their *QR* factorization.

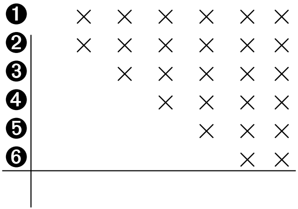
A 6×6 Hessenberg matrix H

$H = G_1 G_2 \dots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

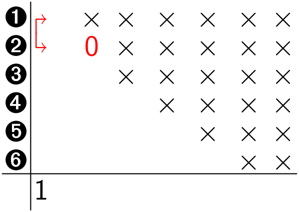


$H = G_1 G_2 \dots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

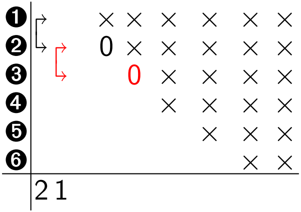


$H = G_1 G_2 \ldots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

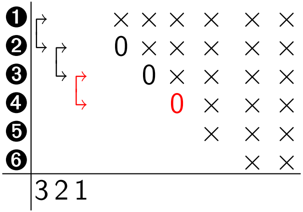


$H = G_1 G_2 \dots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

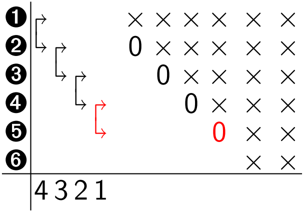


$H = G_1 G_2 \ldots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

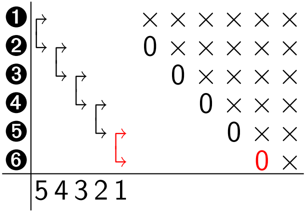


$H = G_1 G_2 \dots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H

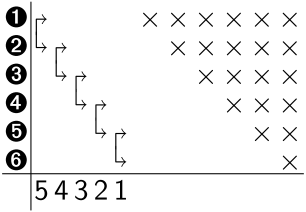


$H = G_1 G_2 \ldots G_5 R.$

Givens-weight representation

We will represent matrices by means their *QR* factorization.

A 6×6 Hessenberg matrix H



$$H = G_1 G_2 \ldots G_5 R.$$

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n\times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H A Q = Q^H S Q + Q^H U V^H Q = \hat{S} + \hat{U}\hat{V}^H$
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n \times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H AQ = Q^H SQ + Q^H UV^H Q = \hat{S} + \hat{U}\hat{V}^H$
  H is still Hermitian plus a rank-m matrix
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n\times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H A Q = Q^H S Q + Q^H U V^H Q = \hat{S} + \hat{U}\hat{V}^H$
  H is still Hermitian plus a rank-m matrix
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n \times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H AQ = Q^H SQ + Q^H UV^H Q = \hat{S} + \hat{U}\hat{V}^H$
  H is still Hermitian plus a rank-m matrix
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n\times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H AQ = Q^H SQ + Q^H UV^H Q = \hat{S} + \hat{U}\hat{V}^H$
  H is still Hermitian plus a rank-m matrix
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

Reduction to Hessenberg

$A = S + UV^H$, where S is Hermitian and $U, V \in C^{n\times m}$

- Reduce A into Hessenberg form.
  If S has some structure the cost is $\mathcal{O}(n^2)$
- $H = Q^H AQ = Q^H SQ + Q^H UV^H Q = \hat{S} + \hat{U}\hat{V}^H$
  H is still Hermitian plus a rank-m matrix
- Let $\mathcal{F}_{n,m}$ the class of Hessenberg matrices which are the sum of a $n \times n$ Hermitian plus rank-m matrix

QR-algorithm

- $\mathcal{F}_{n,m}$ is closed under QR steps
- Multishift QR

$$p_d^{(k)}(H^{(k)}) = Q^{(k)} R^{(k)}$$
$$H^{(k+1)} = {Q^{(k)}}^H H^{(k)} Q^{(k)},$$

- For $d = 2$, $p_2^{(k)}(H^{(k)}) = (H^{(k)} - \mu_1^{(k)} I)(H^{(k)} - \mu_2^{(k)} I)$

QR-algorithm

- $\mathcal{F}_{n,m}$ is closed under QR steps
- Multishift QR

$$p_d^{(k)}(H^{(k)}) = Q^{(k)}R^{(k)}$$
$$H^{(k+1)} = {Q^{(k)}}^H H^{(k)} Q^{(k)},$$

where, for every k, $p_d^{(k)}(x)$ is a monic polynomial of degree d.

- For $d = 2$, $p_2^{(k)}(H^{(k)}) = (H^{(k)} - \mu_1^{(k)} I)(H^{(k)} - \mu_2^{(k)} I)$

QR-algorithm

- $\mathcal{F}_{n,m}$ is closed under QR steps
- Multishift QR

$$p_d^{(k)}(H^{(k)}) = Q^{(k)} R^{(k)}$$
$$H^{(k+1)} = {Q^{(k)}}^H H^{(k)} Q^{(k)},$$

The sequence $\{H^{(k)}\}_k$ converges to the real canonical Schur form of H.

- For $d = 2$, $p_2^{(k)}(H^{(k)}) = (H^{(k)} - \mu_1^{(k)} I)(H^{(k)} - \mu_2^{(k)} I)$

QR-algorithm

- $\mathcal{F}_{n,m}$ is closed under QR steps
- Multishift QR

$$p_d^{(k)}(H^{(k)}) = Q^{(k)} R^{(k)}$$
$$H^{(k+1)} = {Q^{(k)}}^H H^{(k)} Q^{(k)},$$

- For $d = 2$, $p_2^{(k)}(H^{(k)}) = (H^{(k)} - \mu_1^{(k)} I)(H^{(k)} - \mu_2^{(k)} I)$

An implicit method

Perform the transition $H^{(k)} \rightarrow H^{(k+1)}$ implicitly.

- Initialization step Compute $Q_{\mathcal{I}}$ such that

$$Q_{\mathcal{I}}\, p_d(H)\mathbf{e}_1 = \beta\, \mathbf{e}_1, \quad \beta = \|p_d(H)\mathbf{e}_1\|_2,$$

  $H_{\mathcal{I}} = Q_{\mathcal{I}}^H H Q_{\mathcal{I}}$ non Hessenberg. Has a bulge with tip in position $(d+2, 1)$.

- Chasing steps Chase the bulge down to restore the Hessenberg structure. Let $Q_{\mathcal{C}}$ the orthogonal matrix obtained, product of $d(n-2)$ Givens factors.

An implicit method

Perform the transition $H^{(k)} \to H^{(k+1)}$ implicitly.

- Initialization step Compute $Q_\mathcal{I}$ such that

$$Q_\mathcal{I}\, p_d(H)\mathbf{e}_1 = \beta\, \mathbf{e}_1, \quad \beta = \|p_d(H)\mathbf{e}_1\|_2,$$

  $H_\mathcal{I} = Q_\mathcal{I}^H H Q_\mathcal{I}$ non Hessenberg. Has a bulge with tip in position $(d+2, 1)$.

- Chasing steps Chase the bulge down to restore the Hessenberg structure. Let $Q_\mathcal{C}$ the orthogonal matrix obtained, product of $d(n-2)$ Givens factors.

An implicit method

Perform the transition $H^{(k)} \to H^{(k+1)}$ implicitly.

- Initialization step Compute $Q_{\mathcal{I}}$ such that

$$Q_{\mathcal{I}}\, p_d(H)\mathbf{e}_1 = \beta\,\mathbf{e}_1, \quad \beta = \|p_d(H)\mathbf{e}_1\|_2,$$

  $H_{\mathcal{I}} = Q_{\mathcal{I}}^H H Q_{\mathcal{I}}$ non Hessenberg. Has a bulge with tip in position $(d+2, 1)$.

- Chasing steps Chase the bulge down to restore the Hessenberg structure. Let $Q_{\mathcal{C}}$ the orthogonal matrix obtained, product of $d(n-2)$ Givens factors.

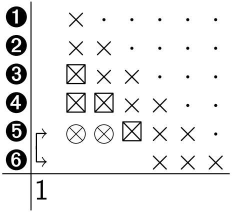
An $\mathcal{O}(n)$ representation

$H = S + UV^H$, hence

$$S = \begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \boxtimes & \times & \times & \cdot & \cdot & \cdot \\ \boxtimes & \boxtimes & \times & \times & \cdot & \cdot \\ \boxtimes & \boxtimes & \boxtimes & \times & \times & \cdot \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \times & \times \end{bmatrix}$$

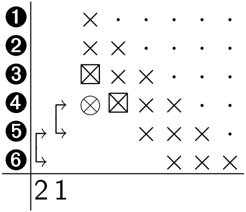
Rank-1 perturbation

If $H = S + \mathbf{u}\mathbf{v}^H$, we can annihilate most of the rank-1 part of S as follows



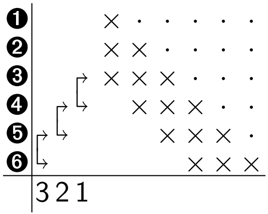
Rank-1 perturbation

If $H = S + \mathbf{uv}^H$, we can annihilate most of the rank-1 part of S as follows



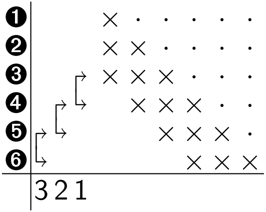
Rank-1 perturbation

If $H = S + \mathbf{uv}^H$, we can annihilate most of the rank-1 part of S as follows



Rank-1 perturbation

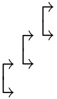
If $H = S + \mathbf{uv}^H$, we can annihilate most of the rank-1 part of S as follows



We represent S as $S = \tilde{V}\tilde{B}$, where $\tilde{V} = G_1 G_2 G_3$, $\tilde{B}$ is the generalized Hessenberg.

Representation of H

$$H = S + \mathbf{u}\mathbf{v}^H = \tilde{V}\left(\tilde{B} + \tilde{\mathbf{u}}\mathbf{v}^H\right)$$

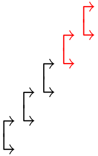
$=$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \cdot \\ & & & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \times \\ \times \\ \\ \\ \\ \end{bmatrix} \mathbf{v}^H \right).$$

Representation of H

To simplify the initialization steps of the implicit QR

$$H = V\left(B + \hat{\mathbf{u}}\mathbf{v}^H\right)$$



$$= \left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \cdot \\ & & & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ 0 \\ 0 \\ \\ \\ \\ \end{bmatrix} \mathbf{v}^H \right)$$

Rank-2 perturbation

If the perturbation is of rank 2,

$$S = H - \mathbf{u}\mathbf{v}^H - \mathbf{x}\mathbf{y}^H$$

We need two sequences of ascending Givens transformations to peel off the rank-2 part in S.

Rank-2 perturbation

If the perturbation is of rank 2,

$$S = H - \mathbf{u}\mathbf{v}^H - \mathbf{x}\mathbf{y}^H$$

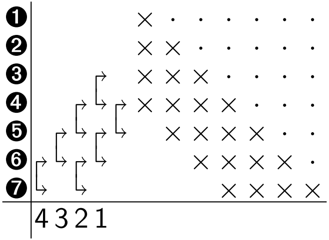
We need two sequences of ascending Givens transformations to peel off the rank-2 part in S.

Rank-2 perturbation

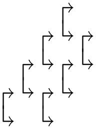
If the perturbation is of rank 2,

$$S = H - \mathbf{u}\mathbf{v}^H - \mathbf{x}\mathbf{y}^H$$

We need two sequences of ascending Givens transformations to peel off the rank-2 part in S.



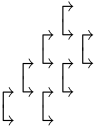
Representation of H

$H =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \otimes_2 \\ \otimes_1 \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}^H + \begin{bmatrix} \times \\ \times \\ \otimes_4 \\ \otimes_3 \\ \\ \\ \\ \end{bmatrix} \mathbf{y}^H \right)$$

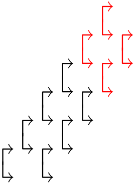
Representation of H

The elements $\otimes$ in the vectors still need to be removed!

$H =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \otimes_2 \\ \otimes_1 \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}^H + \begin{bmatrix} \times \\ \times \\ \otimes_4 \\ \otimes_3 \\ \\ \\ \\ \end{bmatrix} \mathbf{y}^H \right)$$

Representation of H

$H =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ 0 \\ 0 \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}^H + \begin{bmatrix} \times \\ \times \\ 0 \\ 0 \\ \\ \\ \\ \end{bmatrix} \mathbf{y}^H \right)$$

Cost of the representation

In the general rank-m perturbation

$$H = \prod_{i=1}^{m} V^{(\mathbf{u}_i)} \left(B + \sum_{i=1}^{m} \mathbf{u}_i \mathbf{v}_i^H \right),$$

- B is a generalized Hessenberg with $m+1$ subdiagonals
- $\mathbf{u}_i$ are sparse vectors with only the first i entries nonzero
- each $V^{(\mathbf{u}_i)}$ consists of $n-i$ Givens transformations,

- $\sum_{i=1}^{m}(n-i) = \mathcal{O}(nm+m^2)$ Givens transformations;
- $\sum_{i=1}^{m+2}(n+1-i) = \mathcal{O}(nm+m^2)$ entries for the matrix B;
- $\sum_{i=1}^{m} i = \mathcal{O}(m^2)$ entries for the vectors $\mathbf{u}_i$.

Cost of the representation

In the general rank-m perturbation

$$H = \prod_{i=1}^{m} V^{(\mathbf{u}_i)} \left(B + \sum_{i=1}^{m} \mathbf{u}_i \mathbf{v}_i^H \right),$$

- B is a generalized Hessenberg with $m+1$ subdiagonals
- $\mathbf{u}_i$ are sparse vectors with only the first i entries nonzero
- each $V^{(\mathbf{u}_i)}$ consists of $n-i$ Givens transformations,
- $\sum_{i=1}^{m}(n-i) = \mathcal{O}(nm+m^2)$ Givens transformations;
- $\sum_{i=1}^{m+2}(n+1-i) = \mathcal{O}(nm+m^2)$ entries for the matrix B;
- $\sum_{i=1}^{m} i = \mathcal{O}(m^2)$ entries for the vectors $\mathbf{u}_i$.

Cost of the representation

In the general rank-m perturbation

$$H = \prod_{i=1}^{m} V^{(\mathbf{u}_i)} \left(B + \sum_{i=1}^{m} \mathbf{u}_i \mathbf{v}_i^H \right),$$

- B is a generalized Hessenberg with $m+1$ subdiagonals
- $\mathbf{u}_i$ are sparse vectors with only the first i entries nonzero
- each $V^{(\mathbf{u}_i)}$ consists of $n-i$ Givens transformations,
- $\sum_{i=1}^{m}(n-i) = \mathcal{O}(nm+m^2)$ Givens transformations;
- $\sum_{i=1}^{m+2}(n+1-i) = \mathcal{O}(nm+m^2)$ entries for the matrix B;
- $\sum_{i=1}^{m} i = \mathcal{O}(m^2)$ entries for the vectors $\mathbf{u}_i$.

Cost of the representation

In the general rank-m perturbation

$$H = \prod_{i=1}^{m} V^{(\mathbf{u}_i)} \left(B + \sum_{i=1}^{m} \mathbf{u}_i \mathbf{v}_i^H \right),$$

- B is a generalized Hessenberg with $m+1$ subdiagonals
- $\mathbf{u}_i$ are sparse vectors with only the first i entries nonzero
- each $V^{(\mathbf{u}_i)}$ consists of $n-i$ Givens transformations,

- $\sum_{i=1}^{m}(n-i) = \mathcal{O}(nm+m^2)$ Givens transformations;
- $\sum_{i=1}^{m+2}(n+1-i) = \mathcal{O}(nm+m^2)$ entries for the matrix B;
- $\sum_{i=1}^{m} i = \mathcal{O}(m^2)$ entries for the vectors $\mathbf{u}_i$.

The algorithm

- We present the single shift QR applied to $H \in \mathcal{F}_{n,2}$
- The generalization to multishift and/or $m > 2$ has too many technical details ...

... the details can be found in the paper!

The algorithm

- We present the single shift QR applied to $H \in \mathcal{F}_{n,2}$
- The generalization to multishift and/or $m > 2$ has too many technical details ...

... the details can be found in the paper!

Initialization

- Find $Q_{\mathcal{I}}$ such that

$$Q_{\mathcal{I}}^H (H - \mu I)\mathbf{e}_1 = \beta\, \mathbf{e}_1$$

A single Givens is enough for initialization!

$Q_{\mathcal{I}} = G_1$, acting on rows 1 and 2

Initialization

- Find G_1 such that

$$G_1{}^H(H - \mu I)\mathbf{e}_1 = \beta\,\mathbf{e}_1$$

A single Givens is enough for initialization!

$Q_{\mathcal{I}} = G_1$, acting on rows 1 and 2

Inizialization

$$H_2 = G_1^H H_1 G_1$$

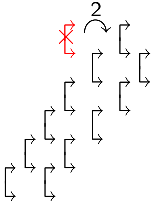
Inizialization

$$H_2 = G_1^H V_1^{(\mathbf{u})} V_1^{(\mathbf{x})} (B_1 + \mathbf{u}_1 \mathbf{v}_1^H + \mathbf{x}_1 \mathbf{y}_1^H) G_1$$

Inizialization

$$H_2 = G_1^H V_1^{(\mathbf{u})} V_1^{(\mathbf{x})} (B_1 + \mathbf{u}_1 \mathbf{v}_1^H + \mathbf{x}_1 \mathbf{y}_1^H)\, G_1$$

$$H_2 =$$

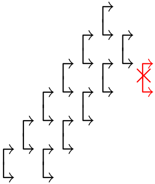


$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_1^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_1^H \right) G_1$$

Inizialization

$$H_2 = G_1^H V_1^{(\mathbf{u})} V_1^{(\mathbf{x})} (B_1 + \mathbf{u}_1 \mathbf{v}_1^H + \mathbf{x}_1 \mathbf{y}_1^H)\, G_1$$

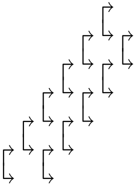
$H_2 =$



$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_1^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_1^H \right) G_1$$

Inizialization

$$H_2 = G_1^H V_1^{(\mathbf{u})} V_1^{(\mathbf{x})} (B_1 + \mathbf{u}_1 \mathbf{v}_1^H + \mathbf{x}_1 \mathbf{y}_1^H)\, G_1$$

$H_2 =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_1^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_1^H \right) G_1$$

$$H_2 = \tilde{V}_1^{(\mathbf{u})} \tilde{V}_1^{(\mathbf{x})} (\tilde{B}_1 + \mathbf{u}_1 \mathbf{v}_1^H + \mathbf{x}_1 \mathbf{y}_1^H) G_1$$

Vectors $\mathbf{u}_1$ and $\mathbf{x}_1$ don't change during the iterations

Transformation on the right

Apply transformation G_1 on the right

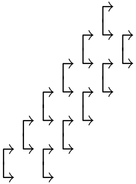
$H_2 =$

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right).$$

Transformation on the right

Apply transformation G_1 on the right

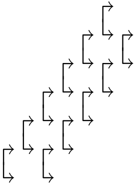
$H_2 =$



$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right).$$

Transformation on the right

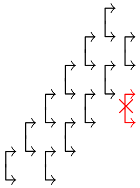
Apply transformation G_1 on the right

$H_2 =$ 
$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ \otimes & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right).$$

We change vectors $\mathbf{v}_1$ and $\mathbf{y}_1$ and a bulge is created in B

Transformation on the right

To remove the undesired $\otimes$ another Givens is needed

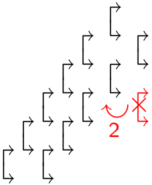
$H_2 =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ 0 & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

Transformation on the right

To remove the undesired $\otimes$ another Givens is needed

$H_2 =$

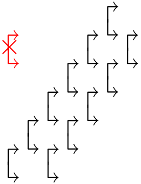


$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ 0 & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

Transformation on the right

To remove the undesired $\otimes$ another Givens is needed

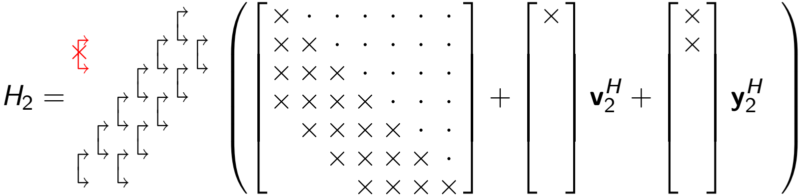
$H_2 =$



$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

Transformation on the right

To remove the undesired $\otimes$ another Givens is needed



The Givens transformation G_2 on the left will determine the next similarity transformation.

$$H_2 = G_2 V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H)$$

The Chasing

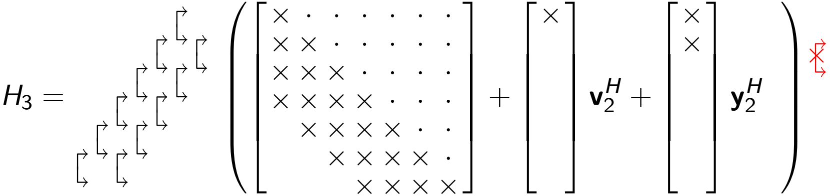
Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$

The Chasing

Idea: move down the Givens on the left!

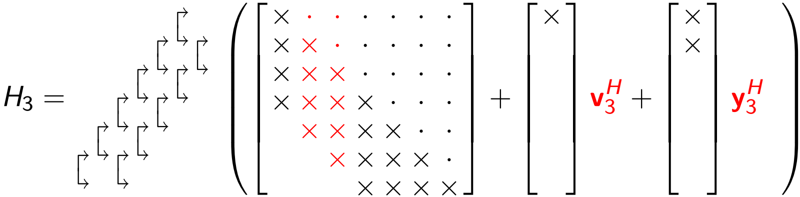
$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) \textcolor{red}{G_2}$$



The Chasing

Idea: move down the Givens on the left!

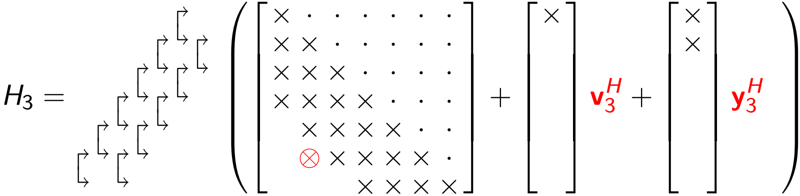
$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$



The Chasing

Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$

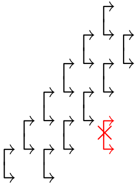


The Chasing

Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$

$H_3 =$

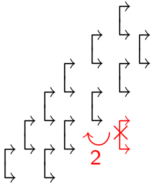


$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & 0 & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_3^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_3^H \right)$$

The Chasing

Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$

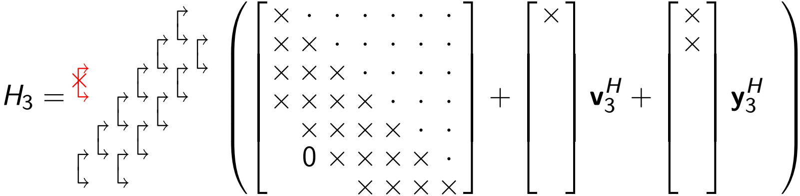
$H_3 =$ 

$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot \\ & \times & \times & \times & \times & \cdot & \cdot \\ & 0 & \times & \times & \times & \times & \cdot \\ & & & \times & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_3^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_3^H \right)$$

The Chasing

Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$

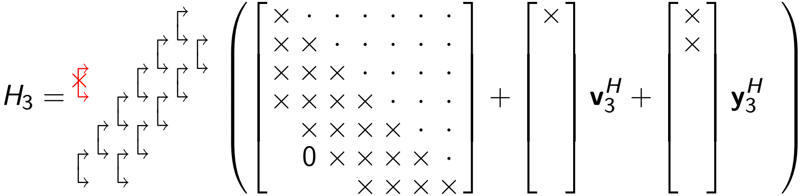


The resulting Givens transformation G_3 acts on rows 3 and 4 and has moved down one position w.r.t. G_2.

The Chasing

Idea: move down the Givens on the left!

$$H_3 = G_2^H H_2 G_2 = V_2^{(\mathbf{u})} V_2^{(\mathbf{x})} (B_2 + \mathbf{u}_2 \mathbf{v}_2^H + \mathbf{x}_2 \mathbf{y}_2^H) G_2$$



$$H_3 = \color{red}{G_3} V_3^{(\mathbf{u})} V_3^{(\mathbf{x})} (B_3 + \mathbf{u}_3 \mathbf{v}_3^H + \mathbf{x}_3 \mathbf{y}_3^H)$$

The last transformations

The implicit algorithm cannot run fully to the end...

The last few columns need to be computed explicitly

Using the representation we can retrieve the columns easily

The last transformations

The implicit algorithm cannot run fully to the end...

The last few columns need to be computed explicitly

Using the representation we can retrieve the columns easily

The last transformations

The implicit algorithm cannot run fully to the end...

The last few columns need to be computed explicitly

Using the representation we can retrieve the columns easily

The multishift strategy

- The multishift strategy proceeds analogously.
- $2(n-1)$ Givens needed for the double shift case.

Deflation

- *QR*-iterations always combined with a deflation technique
- H is numerically of the form

$$H = \left[\begin{array}{c|c} H_1 & \times \\ \hline 0 & H_2 \end{array}\right] = \left[\begin{array}{cccc|cccc} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times & \times \\ \hline & & & {\color{red}0} & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times \\ & & & & & \times & \times & \times \\ & & & & & & \times & \times \end{array}\right]$$

Deflation

- QR-iterations always combined with a deflation technique
- H is numerically of the form

$$H = \left[\begin{array}{c|c} H_1 & \times \\ \hline 0 & H_2 \end{array}\right] = \left[\begin{array}{cccc|cccc} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times & \times \\ \hline & & & 0 & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times \\ & & & & & \times & \times & \times \\ & & & & & & \times & \times \end{array}\right]$$

Decouple the problem and apply the QR to H_1 and H_2

Deflation

- *QR*-iterations always combined with a deflation technique
- H is numerically of the form

$$H = \left[\begin{array}{c|c} H_1 & \times \\ \hline 0 & H_2 \end{array}\right] = \left[\begin{array}{cccc|cccc} \times & \times & \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times & \times \\ & & \times & \times & \times & \times & \times & \times \\ \hline & & & {\color{red}0} & \times & \times & \times & \times \\ & & & & \times & \times & \times & \times \\ & & & & & \times & \times & \times \\ & & & & & & \times & \times \end{array}\right]$$

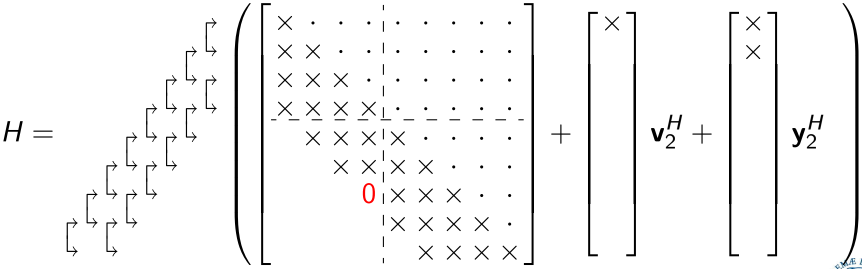
How to detect this using the representation?

Detecting deflation

Let $H = V^{(\mathbf{u})} V^{(\mathbf{x})} \left(B + \mathbf{u}\mathbf{v}^H + \mathbf{x}\mathbf{y}^H \right)$.

Let $W = {V^{(\mathbf{x})}}^H {V^{(\mathbf{u})}}^H H = B + \mathbf{u}\mathbf{v}^H + \mathbf{x}\mathbf{y}^H$.

$H(i+1, i) = 0$ if and only if $W(i+3, i) = 0$



Performing QR-steps

$$H = \left[\begin{array}{c|c|c} H(1:i-1,1:i-1) & \times & \times \\ \hline 0 & H(i:j,i:j) & \times \\ \hline 0 & 0 & H(j+1:n,j+1:n) \end{array}\right]$$

It is possible to perform QR-steps on the different submatrices, even if they interact in the representation

Different strategies for the top, middle and bottom matrices

Performing QR-steps

$$H = \left[\begin{array}{c|c|c} H(1:i-1,1:i-1) & \times & \times \\ \hline 0 & H(i:j,i:j) & \times \\ \hline 0 & 0 & H(j+1:n,j+1:n) \end{array}\right]$$

It is possible to perform QR-steps on the different submatrices, even if they interact in the representation

Different strategies for the top, middle and bottom matrices

Performing *QR*-steps

$$H = \left[\begin{array}{c|c|c} H(1:i-1,1:i-1) & \times & \times \\ \hline 0 & H(i:j,i:j) & \times \\ \hline 0 & 0 & H(j+1:n,j+1:n) \end{array}\right]$$

It is possible to perform *QR*-steps on the different submatrices, even if they interact in the representation

Different strategies for the top, middle and bottom matrices

Performing QR-steps

$$H = \left[\begin{array}{c|c|c} H(1:i-1,1:i-1) & \times & \times \\ \hline 0 & H(i:j,i:j) & \times \\ \hline 0 & 0 & H(j+1:n,j+1:n) \end{array}\right]$$

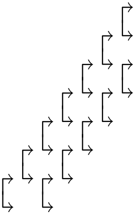
It is possible to perform QR-steps on the different submatrices, even if they interact in the representation

Different strategies for the top, middle and bottom matrices

A QR-step on the middle block

$$H = \left[\begin{array}{cc|ccc|ccc}
\times & \times & \times & \times & \times & \times & \times & \times \\
\times & \times & \times & \times & \times & \times & \times & \times \\
\hline
 & \color{red}{0} & \times & \times & \times & \times & \times & \times \\
 & & \times & \times & \times & \times & \times & \times \\
 & & & \times & \times & \times & \times & \times \\
\hline
 & & & & \color{red}{0} & \times & \times & \times \\
 & & & & & \times & \times & \times \\
 & & & & & & \times & \times
\end{array}\right]$$

A QR-step on the middle block

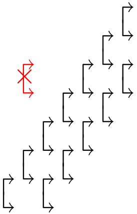


$$H = \left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & {\color{red}0} & \times & \times & \times & \cdot & \cdot & \cdot \\ \hline & & \times & \times & \times & \times & \cdot & \cdot \\ & & & \times & \times & \times & \times & \cdot \\ & & & & {\color{red}0} & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

- Initialization:

$H_2 =$ 

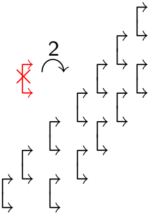
$$\left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & 0 & \times & \times & \times & \cdot & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot & \cdot \\ & & & \times & \times & \times & \times & \cdot \\ & & & & 0 & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_1^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_1^H \right) G_1$$

G_1 acts on rows 3 and 4

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

- Initialization:



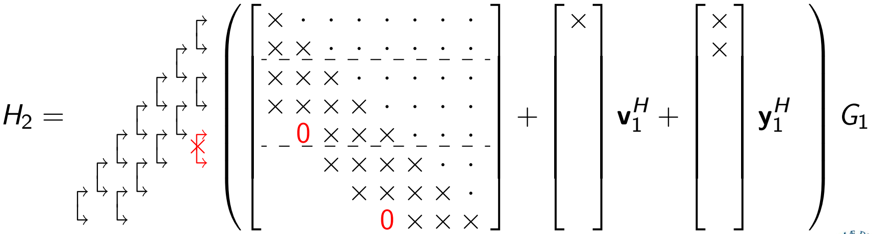
$$H_2 = \left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & 0 & \times & \times & \times & \cdot & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot & \cdot \\ & & & \times & \times & \times & \times & \cdot \\ & & & & 0 & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_1^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_1^H \right) G_1$$

G_1 acts on rows 3 and 4

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

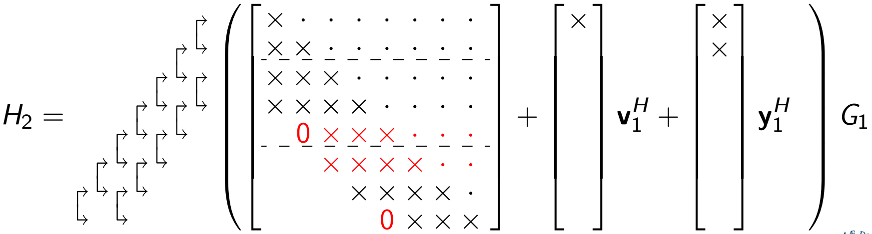
- Initialization:



A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

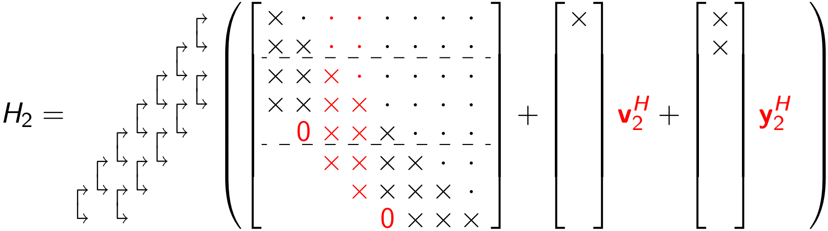
- Initialization:



A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

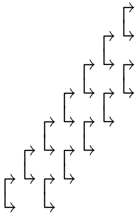
- Initialization:



A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

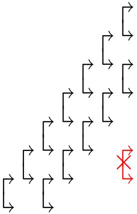
- Initialization:



$$H_2 = \left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & {\color{red}0} & \times & \times & \times & \cdot & \cdot & \cdot \\ \hline & & \times & \times & \times & \times & \cdot & \cdot \\ & & {\color{red}\otimes} & \times & \times & \times & \times & \cdot \\ & & & & {\color{red}0} & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!



$$H_2 = \left(\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & 0 & \times & \times & \times & \cdot & \cdot & \cdot \\ & & \times & \times & \times & \times & \cdot & \cdot \\ & & & \times & \times & \times & \times & \cdot \\ & & & & 0 & \times & \times & \times \end{bmatrix} + \begin{bmatrix} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{v}_2^H + \begin{bmatrix} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \mathbf{y}_2^H \right)$$

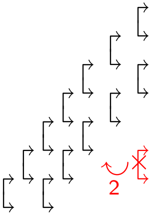
To remove the bulge, we need a Givens acting on rows 6,7

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

- Initialization:

$H_2 =$

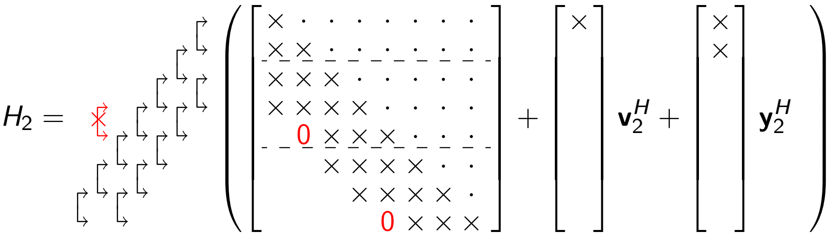


$$\left(\left[\begin{array}{cccccccc} \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline \times & \times & \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot & \cdot & \cdot \\ & 0 & \times & \times & \times & \cdot & \cdot & \cdot \\ \hline & & \times & \times & \times & \times & \cdot & \cdot \\ & & & \times & \times & \times & \times & \cdot \\ & & & & 0 & \times & \times & \times \end{array}\right] + \left[\begin{array}{c} \times \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\right] \mathbf{v}_2^H + \left[\begin{array}{c} \times \\ \times \\ \\ \\ \\ \\ \\ \\ \end{array}\right] \mathbf{y}_2^H \right)$$

A QR-step on the middle block

The middle block is 3×3. We need one Givens for the initialization step and one for the chasing!

- Initialization:



The Givens matrix on the left is the one determinig the chasing step!!

Computational cost

A single *QR*-step

- Inizialization: ≈ 69 flops
- Bulge cheasing: $\approx 134(n-2)$ flops

Error criteria

We considered three different error criteria

- $$E^{(abs)} = \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty = \max_i \left\{ |\lambda_i - \tilde{\lambda}_i| \right\},$$

- $$E^{(rel)} = \max_i \left\{ \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \right\}$$

- $$E^{(rel2)} = \frac{\|\lambda - \tilde{\lambda}\|_\infty}{\|\lambda\|_\infty} = \frac{E^{(abs)}}{|\lambda_{\max}|}$$

Error criteria

We considered three different error criteria

- $$E^{(abs)} = \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty = \max_i \left\{ |\lambda_i - \tilde{\lambda}_i| \right\},$$
- $$E^{(rel)} = \max_i \left\{ \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \right\}$$
- $$E^{(rel2)} = \frac{\|\lambda - \tilde{\lambda}\|_\infty}{\|\lambda\|_\infty} = \frac{E^{(abs)}}{|\lambda_{\max}|}$$

Error criteria

We considered three different error criteria

- $$E^{(abs)} = \|\boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}\|_\infty = \max_i \left\{ |\lambda_i - \tilde{\lambda}_i| \right\},$$
- $$E^{(rel)} = \max_i \left\{ \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|} \right\}$$
- $$E^{(rel2)} = \frac{\|\lambda - \tilde{\lambda}\|_\infty}{\|\lambda\|_\infty} = \frac{E^{(abs)}}{|\lambda_{\max}|}$$

Test suite

- `Type I: Chebyshev-Comrade matrices`, $A = T_n + \mathbf{u}\mathbf{e_n}^T$,

$$T_n = \begin{bmatrix} 0 & \frac{1}{\sqrt{2}} & & & & \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{2} & & & \\ & \frac{1}{2} & \ddots & \ddots & & \\ & & \ddots & \ddots & \frac{1}{2} & \\ & & & \frac{1}{2} & 0 & \frac{1}{\sqrt{2}} \\ & & & & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}.$$

Are involved in the computation of the roots of a polynomial expressed in the first kind Chebyshev polynomial basis

Chebyshev-Comrade matrices

$\mathbf{u} = \alpha\,[1, \ldots, 1]^T$, for increasing values of α.

Table: Errors for Type-I matrices of size $n = 128$, and $\mathbf{u} = \alpha\,[1, \ldots, 1]^T$.

α	1	10^3	10^5	10^7	10^8	10^{11}
$E^{(abs)}$	3.0631e-13	8.6616e-12	1.4552e-10	5.5879e-09	8.9407e-08	1.7243e-06
$E^{(rel)}$	3.0631e-13	8.6722e-12	8.5210e-11	5.5943e-09	1.4885e-08	3.8406e-06
$E^{(rel2)}$	1.6396e-13	8.6555e-15	1.4552e-15	5.5879e-16	8.9407e-16	1.7243e-17
avrgit	2.6371	2.8182	2.8099	2.8099	2.7934	3.1736

Chebyshev-Comrade matrices

Table: Errors for Type-I matrices of the form $T_n + \mathbf{u}\mathbf{e}_\mathbf{n}^\mathbf{T}$, with $\mathbf{u} = \text{rand}(n, 1)$. For different values of n, the absolute and relative errors are shown.

n	$E^{(abs)}$	$E^{(rel)}$	$E^{(rel2)}$
50	1.0947e-13	1.0947e-13	8.6909e-14
100	4.2141e-13	2.4800e-11	3.5758e-13
150	3.4097e-12	2.1823e-12	2.1823e-12
200	3.6731e-12	2.6981e-12	2.6981e-12
300	6.8008e-12	5.0597e-12	5.0597e-12
500	1.7089e-11	1.3405e-11	1.3405e-11

Test suite

- Type-II: Random tridiagonal plus random rank-one permutation. $A = T + \mathbf{u}\mathbf{e_n}^T$, where T is a random symmetric tridiagonal matrix, and $\mathbf{u}$ is a random vector in $[0, 1]$.

Table: Errors for Type-II matrices, and for different values of n. The values reported represent the average error obtained over 50 randomly generated instances.

n	$E^{(abs)}$	$E^{(rel)}$	$E^{(rel2)}$
100	2.2438e-12	1.3879e-12	1.0361e-12
150	3.4440e-10	5.6035e-10	1.5864e-10
200	5.7883e-12	4.9680e-12	2.6019e-12
300	1.5570e-12	7.6508e-12	7.2006e-13
500	4.2696e-12	3.0138e-11	1.8976e-12

Test suite

- `Type-III: Unsymmetric tridiagonal matrices.` We consider an almost symmetric tridiagonal matrix of the form

$$T = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & 1 & 0 & 1 \\ & & & \alpha & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & 1 & 0 & \alpha \\ & & & \alpha & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1-\alpha \\ 0 \end{bmatrix} \mathbf{e_n}^T. \quad (1)$$

When α becomes very large, the dominant eigenvalues are ill-conditioned.

Test suite

- Type-III: Unsymmetric tridiagonal matrices.

Table: Errors for Type-III matrices, n=128, and different values of α.

α	1	10	10^2	10^3	10^5	10^7	10^8
$E^{(abs)}$	5.8842e-14	7.9892e-13	9.8765e-13	1.6662e-12	5.8321e-11	3.1127e-09	1.8700e-08
$E^{(rel)}$	5.8842e-14	4.9782e-13	4.3876e-13	3.1974e-13	5.1728e-11	1.0268e-09	9.2553e-09
$E^{(rel2)}$	2.9430e-14	2.3967e-13	9.8270e-14	5.2664e-14	1.8443e-13	9.8434e-13	1.8700e-12
avrgit	2.9098	2.9268	3.3719	3.3033	2.9016	3.0656	2.9431

Test suite

- Type-IV: Hamiltonian-like matrices.

$$A = \begin{bmatrix} D & 0 \\ 0 & -\Delta \end{bmatrix} + \begin{bmatrix} \mathbf{q} \\ \mathbf{e} \end{bmatrix} \begin{bmatrix} \mathbf{e}^T, -\mathbf{q}^T \end{bmatrix},$$

Table: Errors for Type-IV matrices of different size. The results are obtained for $\alpha = 0.9$ and $\beta = 0.8$.

n	$E^{(abs)}$	$E^{(rel)}$	$E^{(rel2)}$
50	3.1335e-12	1.8983e-12	9.6708e-15
100	6.8070e-12	3.4106e-12	1.0694e-14
150	1.0714e-11	1.0714e-11	1.1290e-14
200	1.8701e-11	1.8701e-11	1.4825e-14
500	7.2838e-11	7.2838e-11	2.3223e-14
1000	8.0850e-10	8.0850e-10	1.2912e-13

Test suite

- `Type-V: Hessenberg form of random matrices.`

Table: Type-V: Hessenberg form of randomly generated matrices.

n	$E^{(abs)}$	$E^{(rel)}$	$E^{(rel2)}$
50	2.0783e-12	2.0783e-12	3.2932e-14
100	9.5799e-12	9.5799e-12	7.6777e-14
150	5.5904e-11	6.2688e-12	2.9514e-13
200	8.2377e-11	8.2377e-11	3.2935e-13
500	9.1855e-11	3.2461e-11	1.4710e-13
1000	3.4718e-09	5.1914e-10	2.7480e-12

Conclusions and future work

- A new implicit-*QR* algorithm on rank-structured matrices
- Givens-weight representation is the keyword
- A new deflation technique
- Effectiveness of the representation on iteration schemes

- Backward error analysis
- Get rid of the explicit computations!

Conclusions and future work

- A new implicit-QR algorithm on rank-structured matrices
- Givens-weight representation is the keyword
- A new deflation technique
- Effectiveness of the representation on iteration schemes

- Backward error analysis
- Get rid of the explicit computations!