Orthogonal iterations for structured pencils arising from nonlinear eigenvalue problems

Gianna M. Del Corso
joint work with R. Bevilacqua and L. Gemignani

Dipartimento di Informatica, Università di Pisa, Italy

ILAS2019

Review of Inverse Orthogonal Iterations

Approximating Polynomial

Linearization

The *LFR* format

The algorithm on the structure

Numerical Experiments

Conclusions and further work

The Problem

- Eigenvalues computation of non linear function $T(z)$
- If not a matrix polynomial might have infinitely many eigenvalues
- Computing eigenvalues located only in certain regions

The Problem

$$T : \Omega \to \mathbb{C}^{k \times k}$$

(NEP) find the pair $(\lambda, v), v \neq 0$ such that $T(\lambda)v = 0$.

Consider $\Delta \in \Omega$, and find eigenvalues in Δ.

Possible approaches (see Güttel-Tisseur)

- Newton's method
- Contour integrals
- Approximation of the nonlinear function with a matrix polynomial
- Linearize to obtain a pencil (A, B)

The generalized eigenvalue problem

To find the eigenvalue of (A, B)

- QR on $B^{-1}A$
- QZ on the pencil
- Arnoldi or other Krylov methods
- Since most of the eigenvalues have to be thrown away: orthogonal iterations
- This is important when we have clustered eigenvalues

The generalized eigenvalue problem

To find the eigenvalue of (A, B)

- QR on $B^{-1}A$
- QZ on the pencil
- Arnoldi or other Krylov methods
- Since most of the eigenvalues have to be thrown away: inverse orthogonal iterations
- This is important when we have clustered eigenvalues

Inverse orthogonal iterations

- Generalization of inverse Power iterations
- Approximate the invariant subspace associated with the s eigenvalues with smallest modulus
- Approximate the s eigenvalues with smallest modulus
- Given $A \in \mathbb{C}^{n \times n}$, and $Q_0 \in \mathbb{C}^{n \times s}$ compute

$$\begin{cases} AZ_i = Q_{i-1} \\ Q_i R_i = Z_i \quad \text{economy size QR fact} \end{cases}$$

- Usual hypothesis of convergence

Inverse orthogonal iterations: pencil

- On the pencil (A, B)

$$\begin{cases} AZ_i = BQ_{i-1} \\ Q_i R_i = Z_i \qquad \text{economy size QR fact} \end{cases}$$

- Compute $V_{i-1}S_{i-1} = BQ_{i-1}$, full QR factorization.
- $AZ_i = V_{i-1}S_{i-1}$, that is

$$V_{i-1}^H AZ_i = S_{i-1}, \qquad S_{i-1} \text{ is rectangular upper triang}$$

Consider the RQ factorization of $V_{i-1}^H A$.

- $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$ we get

$$V_{i-1}^H AZ_i = S_{i-1}.$$

- Then $Z_i = \tilde{Q}_i^H \tilde{R}_i^{-1} S_{i-1}$ and $Q_i = \tilde{Q}_i^H(:, 1:s)$.

Inverse orthogonal iterations: pencil

- On the pencil (A, B)

$$\begin{cases} AZ_i = BQ_{i-1} \\ Q_i R_i = Z_i \qquad \text{economy size QR fact} \end{cases}$$

- Compute $V_{i-1}S_{i-1} = BQ_{i-1}$, full QR factorization.
- $AZ_i = V_{i-1}S_{i-1}$, that is

$$V_{i-1}^H AZ_i = S_{i-1}, \qquad S_{i-1} \text{ is rectangular upper triang}$$

Consider the RQ factorization of $V_{i-1}^H A$.

- $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$ we get

$$\tilde{R}_i \tilde{Q}_i Z_i = S_{i-1}.$$

- Then $Z_i = \tilde{Q}_i^H \tilde{R}_i^{-1} S_{i-1}$ and $Q_i = \tilde{Q}_i^H(:, 1:s)$.

Stopping criterion

The usual stopping criterion is

$$\|(I - Q_{i-1}Q_{i-1}^H)Q_i\| < tol$$

In this case

$$Q_i - Q_{i-1}(Q_{i-1}^H Q_i) = E, \quad \|E\| < tol$$

guarantees

$$\frac{\|r(A,B)\|}{\|A^{-1}B\|} = \frac{\|A^{-1}BQ_{i-1} - Q_{i-1}Q_{i-1}^H Q_i R_i\|}{\|A^{-1}B\|} \leq tol$$

Bound on the relative residual!

Back to the original problem

$T : \Omega \to \mathbb{C}^{k\times k}$ holomorphic matrix-valued function, $\Omega \subseteq \mathbb{C}$ open and connected.
Focus on computing eigenvalues in a selected subset $\Delta \in \Omega$
A possible approach

$$(\text{NEP}) \Longrightarrow (\text{PEP}) \Longrightarrow (\text{LIN})$$

$$T(z) \Longrightarrow P(z) \Longrightarrow (A, B)$$

If $T(z)$ is not a polynomial, we seek an approximation $P(z)$ such that

$$\|T(z) - P(z)\| \leq \varepsilon, \qquad z \in \Delta \subseteq \Omega$$

Approximating polynomial

- Taylor polynomial This approximation is appropriate if $\lambda \in \bar{\mathbb{D}}_{\sigma,\rho} \subset \Omega$
- Interpolation on suitable nodes
  - Chebyshev nodes
  - $d+1$ Roots of unity
  - d Roots of unity plus $\sigma = 0$

Approximating polynomial

- Taylor polynomial
  $P_d(z) = T(\sigma) + T'(\sigma)(z-\sigma) + \cdots + T^{(d)}(\sigma)\frac{(z-\sigma)^d}{d!}$. This approximation is appropriate if $\lambda \in \bar{\mathbb{D}}_{\sigma,\rho} \subset \Omega$
- Interpolation on suitable nodes
  - Chebyshev nodes
  - $d+1$ Roots of unity
  - d Roots of unity plus $\sigma = 0$

Approximating polynomial

- Taylor polynomial
  $P_d(z) = T(\sigma) + T'(\sigma)(z-\sigma) + \cdots + T^{(d)}(\sigma)\frac{(z-\sigma)^d}{d!}$. This approximation is appropriate if $\lambda \in \bar{\mathbb{D}}_{\sigma,\rho} \subset \Omega$
- Interpolation on suitable nodes
  - Chebyshev nodes Suitable if the wanted eigenvalues of T lie in or near the interval $[-1; 1]$
  - $d+1$ Roots of unity Suitable if the wanted eigenvalues of T lie in open unit disk
  - d Roots of unity plus $\sigma = 0$ Suitable if the wanted eigenvalues of T lie in open unit disk

Approximating polynomial

For these approximations ...

- We can prove uniform convergence of the interpolant $P_d(z)$ to the $T(z)$ inside "circular" regions in which T is holomorphic.
- If $P_d(z)$ such that $\|T(z) - P_d(z)\| \leq \varepsilon, z \in \Delta$, and (λ, v) is an eigenpair for $P_n(z)$, $\lambda \in \Delta \Longrightarrow \|T(\lambda)v\| < \varepsilon$.
- if $\det(T(\mu)) \neq 0, \mu \in \Delta \Longrightarrow \det(P_d(\mu)) \neq 0$.

Not always easy to choose the degree of the approximating polynomial

Linearizations

Interested in Unitary-plus-low rank pencils...
We tested and compared different linearizations

- Generalized companion
- Unitary diagonal plus low rank
- Arrowed linearization $\rightarrow$ obtained considering the matrix polynomial written in the Lagrange basis

In all these cases we get a pencil (A, B) with A and B unitary plus rank k.

Linearizations

Interested in Unitary-plus-low rank pencils...
We tested and compared different linearizations

- Generalized companion
- Unitary diagonal plus low rank
- Arrowed linearization $\rightarrow$ obtained considering the matrix polynomial written in the Lagrange basis

In all these cases we get a pencil (A, B) with A and B unitary plus rank k.

We can use compressed representations and fast and stable methods

Linearizations

Interested in Unitary-plus-low rank pencils...
We tested and compared different linearizations

- Generalized companion
- Unitary diagonal plus low rank
- Arrowed linearization $\rightarrow$ obtained considering the matrix polynomial written in the Lagrange basis

In all these cases we get a pencil (A, B) with A and B unitary plus rank k.

We can use compressed representations and fast and stable methods

The linearizations considered only need the evaluation of $T(z)$ on the interpolation nodes.

Related work

Most authors analyze the companion /block companion case...

- Since 2004 Bini, Daddi, Gemignani, Eidelman, Gohberg, Boito (explicit QR on unitary plus rank 1)
- 2007 Chandrasekaran, Gu, Xia, Zhu (implicit QR on a QR factorization of the companion)
- 2012 Delvaux, Frederik, Van Barel (block companion, the matrix is stored using the Givens weight representation)
- 2015-2017 Aurentz, Mach, Robol, Vandebril, Watkins different papers where the representation uses only unitary matrices
- 2011 Effenberger, Kresner Chebyshev interpolation points, Krylov methods on the linearized problem
- 2013, Van Beeumen, Meerbergen, Michiels, Hermite interpolation+ rational Krylov
- 2019 Saad, El-Guide, Miedlar Special rational approximation + orthogonal iteration

The *LFR* format

The matrices of the pencil (A, B) are unitary plus low rank.
$A \in \mathcal{U}_k$ can be represented in the *LFR* format

1. $A = LFR$
2. L is the product of k lower Hessenberg matrices
3. R is the product of k upper Hessenberg matrices
4. $F = Q + TZ^H$ is unitary plus rank k,

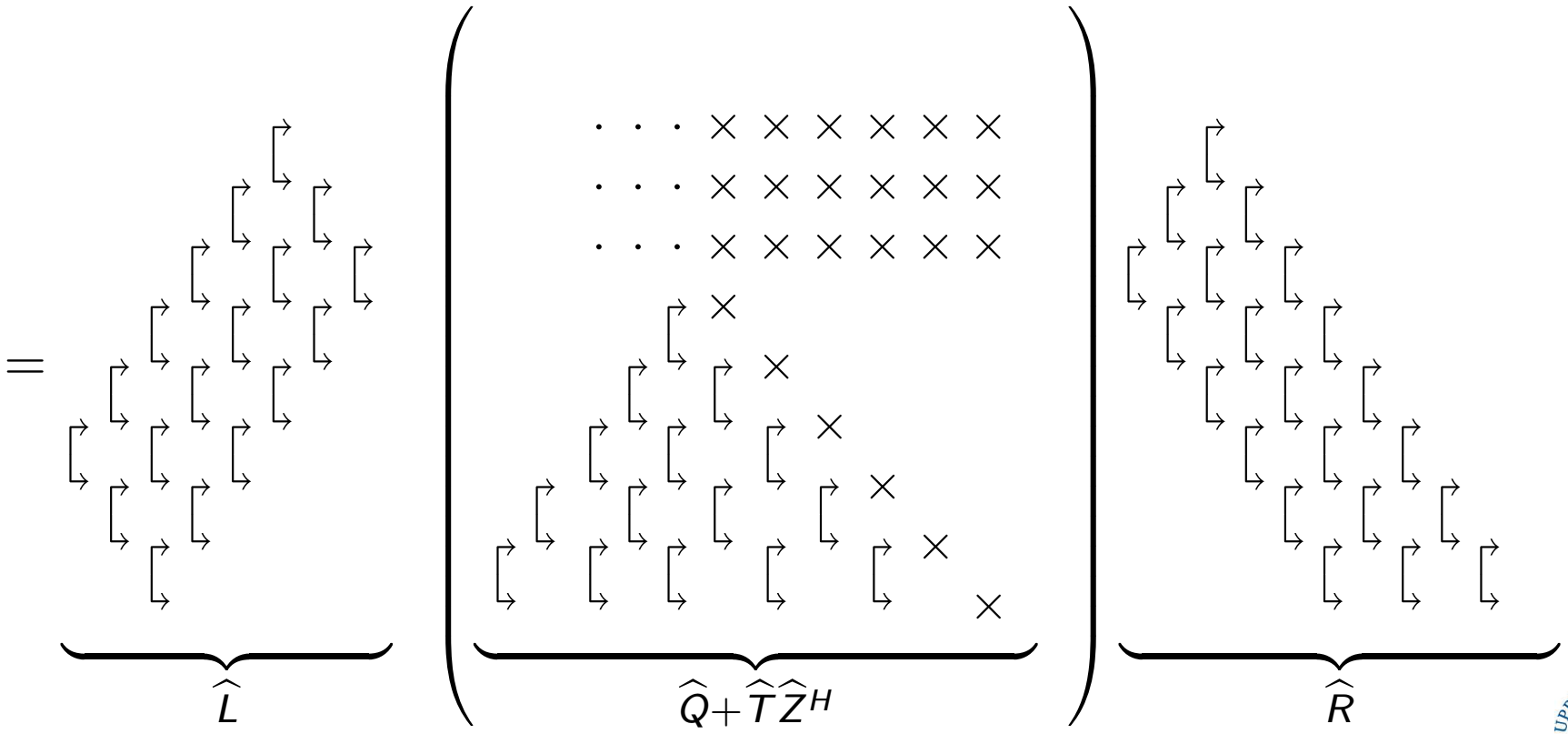
$$Q = \left[\begin{array}{c|c} I_k & \\ \hline & \hat{Q} \end{array}\right]$$

and T upper triangular, so that the correction is in the first k rows

The *LFR* format

A small trick: we construct a larger matrix $\widehat{A} = \widehat{L}\widehat{F}\widehat{R}$ still unitary plus low rank

$$\widehat{A} = \begin{bmatrix} A & * \\ 0_{kn} & 0_{kk} \end{bmatrix} =$$



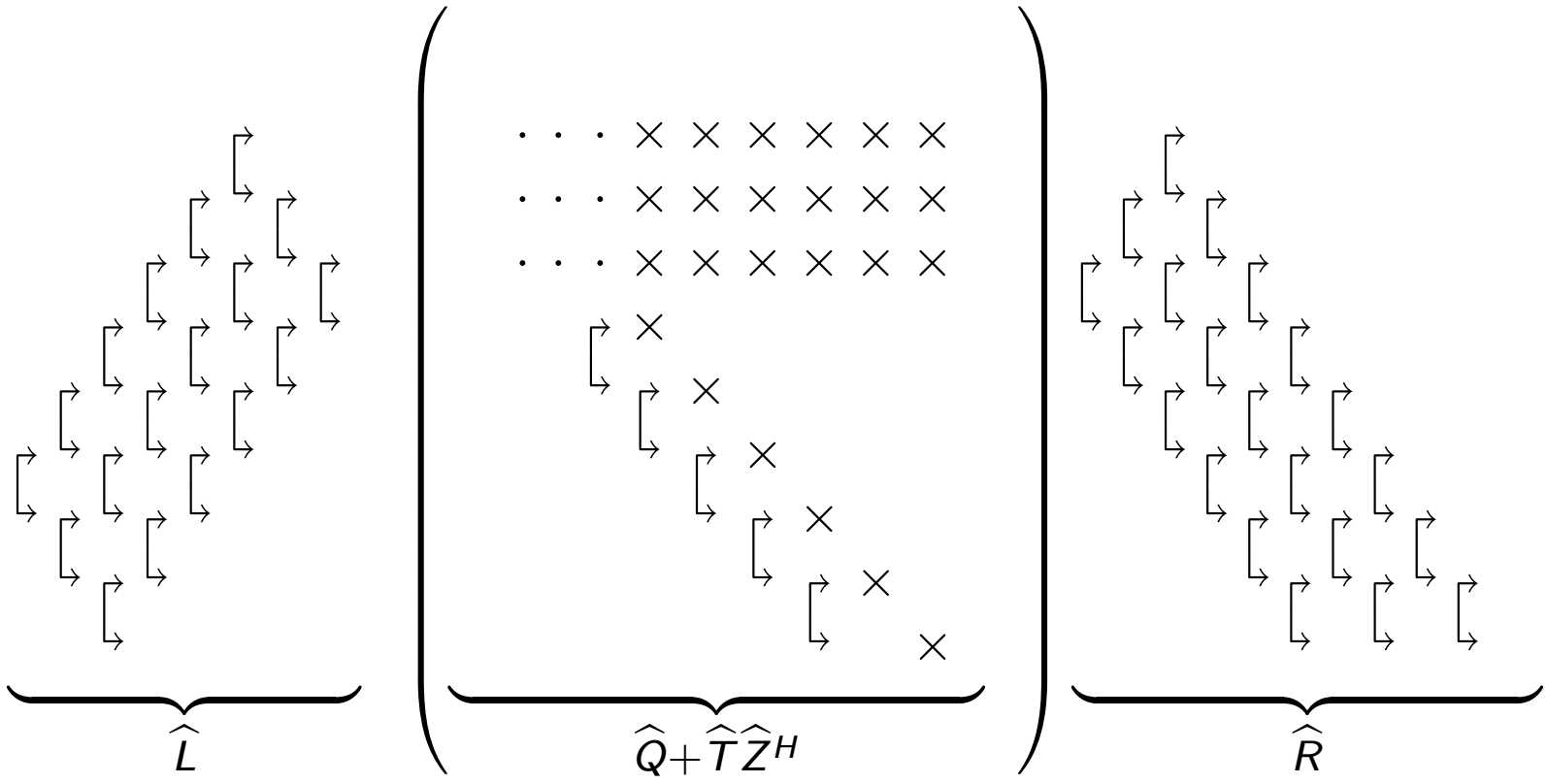
The *LFR* format

- If $Z^H = L(n+1:n+k,:)Q \Leftrightarrow \widehat{A}$ has the last k rows zero
- The structure of $\widehat{F}$ inherits the bandwidth profile of $\widehat{A}$
  1. $\widehat{F}$ is triangular $\Leftrightarrow \widehat{A}$ is triangular
  2. $\widehat{F}$ is Hessenberg $\Leftrightarrow \widehat{A}$ is Hessenberg
  3. $\widehat{F}$ is $h-$Hessenberg $\Leftrightarrow \widehat{A}$ is $h-$Hessenberg

These observations allow to greatly simplify the inverse orthogonal iterations!

Orthogonal iterations on the structure

From a computational point of view it is preferable to work with A in h-Hessenberg form



Orthogonal iterations on the structure

Preprocessing $A = L_A(Q_A + T_A Z_A^H)R_A = L_A(I + T_A \tilde{Z}_A^H)Q_A R_A$

QR is a $(k+h)-$upper Hesenberg matrix

Find $R_A^{(1)}, k-$Hess and $Q_A^{(1)}$ h-Hess such that

$$Q_A R_A = R_A^{(1)} Q_A^{(1)}$$

We have

$$A = \underbrace{L_A(I + T_A \tilde{Z}_A^H)R_A^{(1)}}_{\hat{R}} \underbrace{Q_A^{(1)}}_{\hat{Q}} = \hat{R}\hat{Q}$$

Similarly B is kept factorized as $B = \bar{Q}\bar{R}$, where

$$B = \underbrace{Q_B^{(1)}}_{\bar{Q}} \underbrace{L_B^{(1)}(I + T_B \tilde{Z}_B^H)R_B}_{\bar{R}} = \bar{Q}\bar{R}$$

Orthogonal iterations on the structure

At each step. Q_{i-1} is composed of s orthogonal columns, i.e. is represented as s sequences of ascending Givens rotations

- Compute $V_{i-1}S_{i-1} = BQ_{i-1}$

We can work on the QR representation of B swapping unitary matrices

$$BQ_{i-1} = Q_B^{(1)} L_B^{(1)} (I + T_B \tilde{Z}_B^H) R_B Q_{i-1}$$

We can work on the Schur parametrization of the unitary matrices involved.

Orthogonal iterations on the structure

We have

$$BQ_{i-1} = Q_B^{(1)} L_B^{(1)} (I + T_B \tilde{Z}_B^H) R_B Q_{i-1}$$

Orthogonal iterations on the structure

We have

$$BQ_{i-1} = Q_B^{(1)} L_B^{(1)} (I + T_B \tilde{Z}_B^H) \tilde{Q}_{i-1} \tilde{R}_B$$

Orthogonal iterations on the structure

We have

$$BQ_{i-1} = Q_B^{(1)} L_B^{(1)} \tilde{Q}_{i-1} (I + T_B \tilde{Z}_B^H) \tilde{R}_B$$

Orthogonal iterations on the structure

We have

$$BQ_{i-1} = Q_B^{(1)} \hat{Q}_{i-1} \tilde{L}_B (I + T_B \tilde{Z}_B^H) \tilde{R}_B$$

Orthogonal iterations on the structure

We have

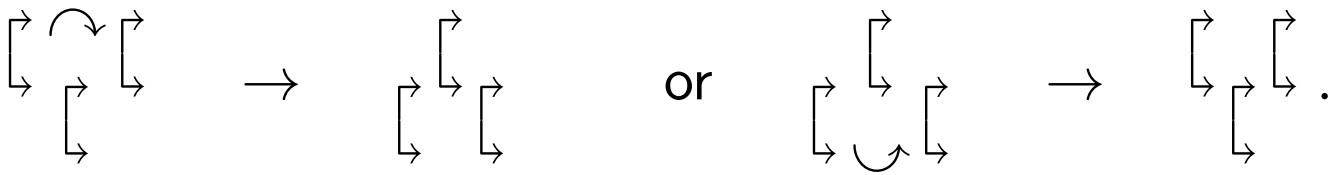
$$BQ_{i-1} = \underbrace{Q_B^{(1)} \hat{Q}_{i-1}}_{V_{i-1}} \underbrace{\tilde{L}_B (I + T_B \tilde{Z}_B^H) \tilde{R}_B}_{S_{i-1}}$$

The cost of retrieving the first s columns of V_{i-1} depends on the structure of $Q_B^{(1)}$.

Basic operations with Givens rotations

Givens transformations can also interact with each other by means of the fusion

The turnover operation allows to rearrange the order of some Givens transformations

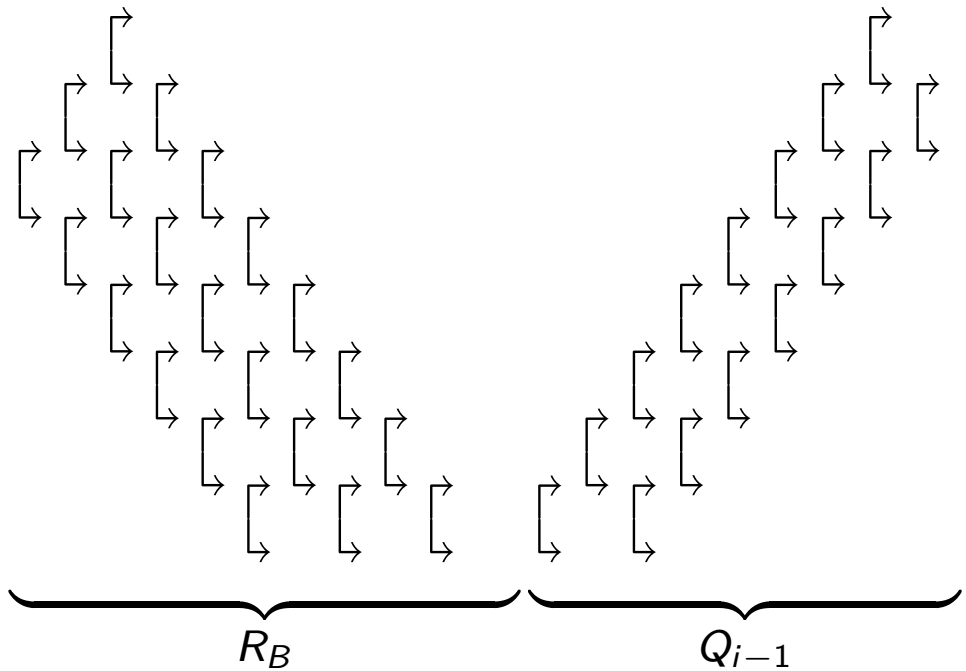


These basic operations require a constant number of flops.

Orthogonal iterations on the structure

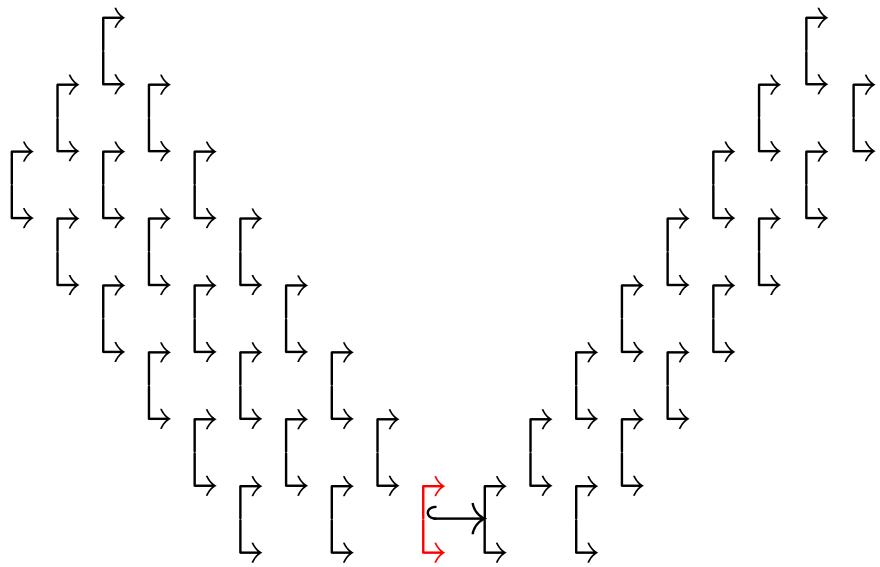
R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

$R_B Q_{i-1} =$



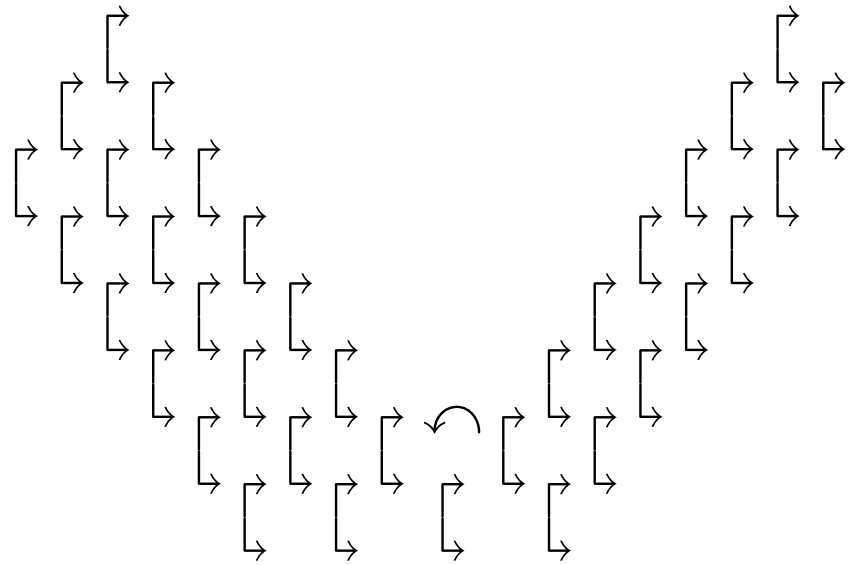
Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

$R_B Q_{i-1} =$ 

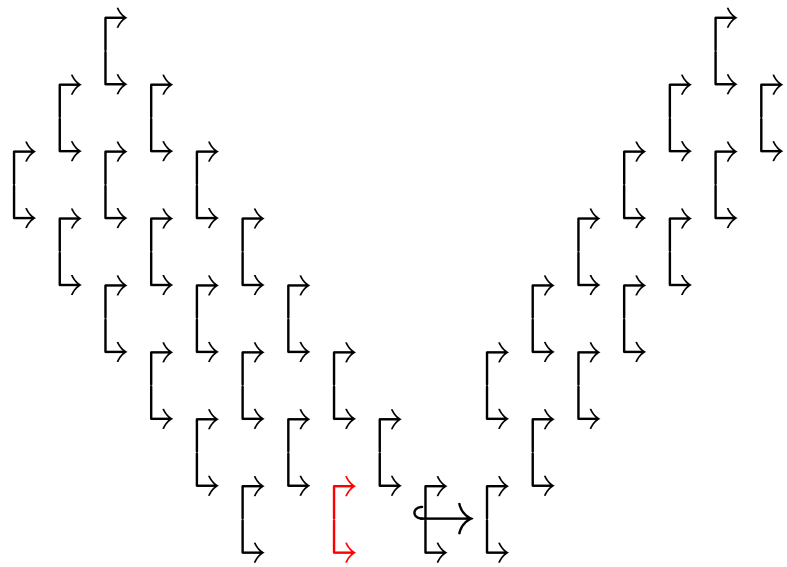
Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

$R_B Q_{k-1} =$ 

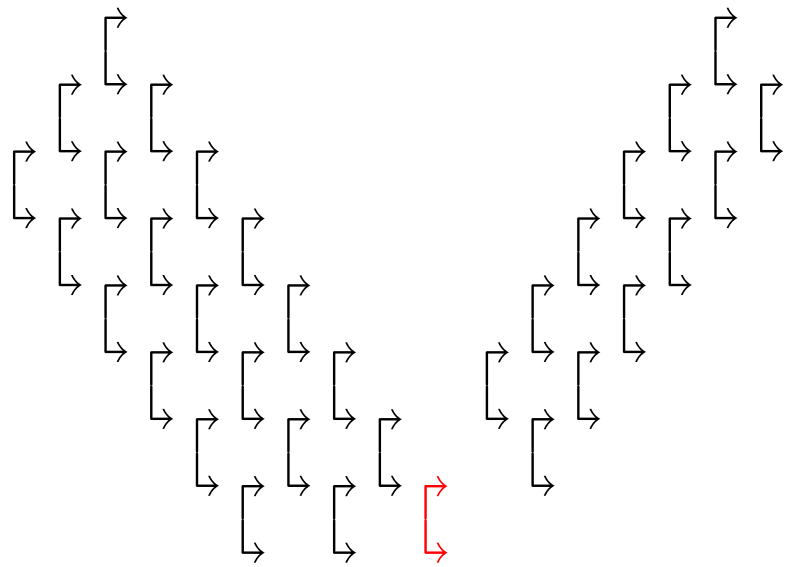
Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

$R_B Q_{k-1} =$ 

Orthogonal iterations on the structure

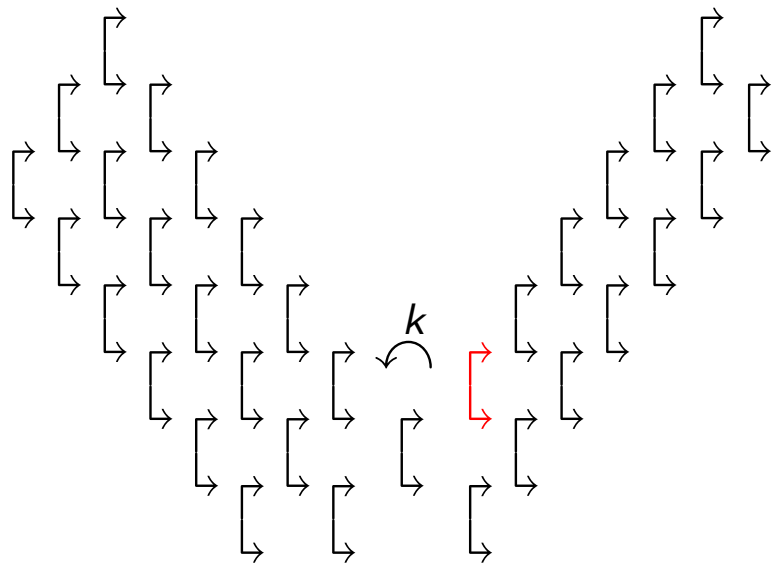
R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

$R_B Q_{k-1} =$ 

Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

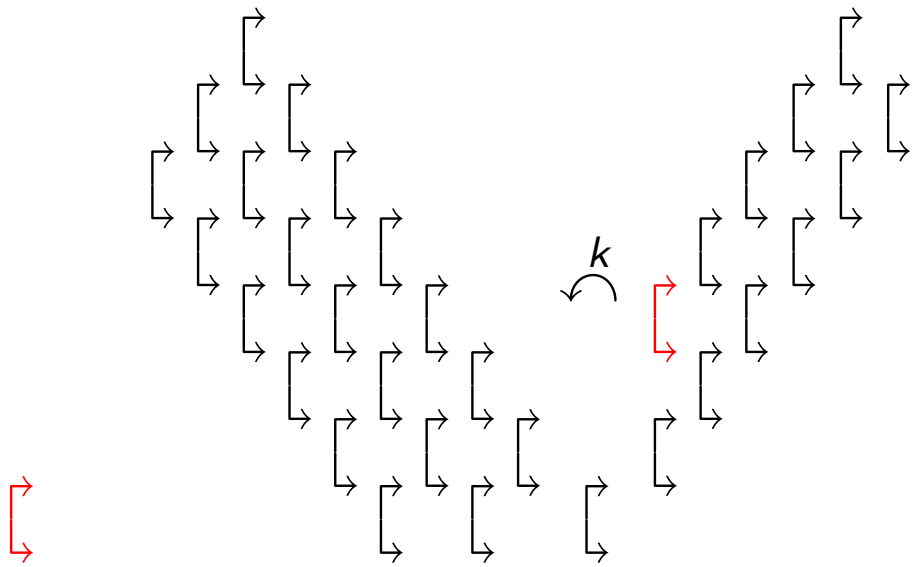
$$R_B Q_{k-1} =$$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

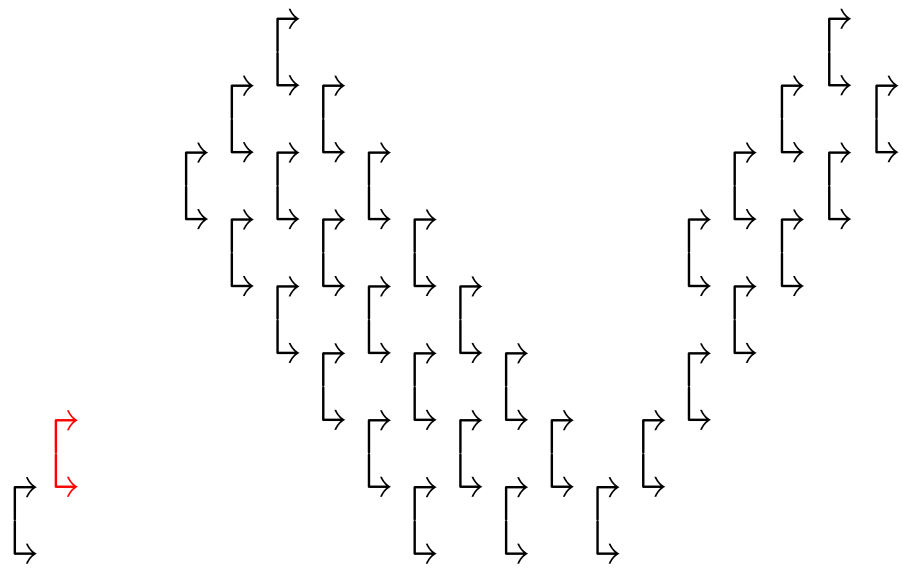
$R_B Q_{k-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

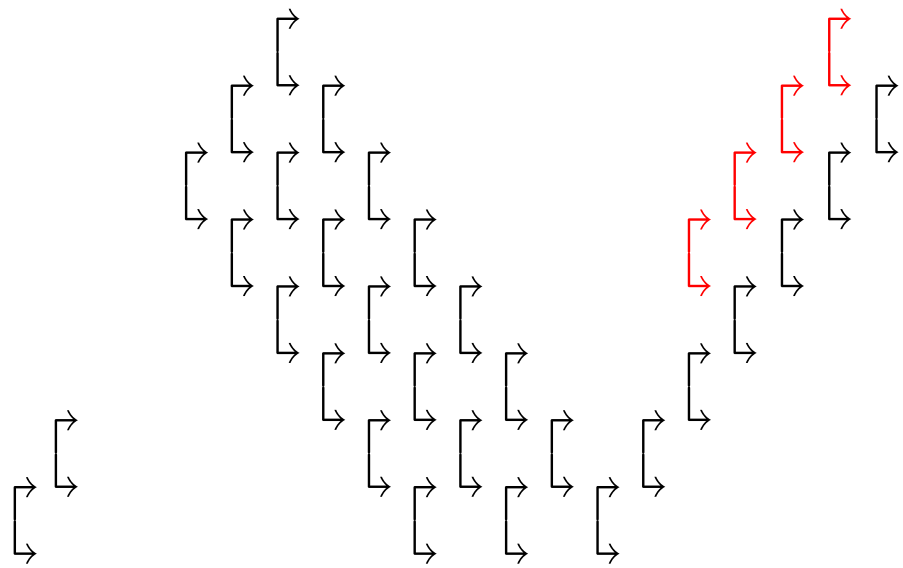
$R_B Q_{k-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

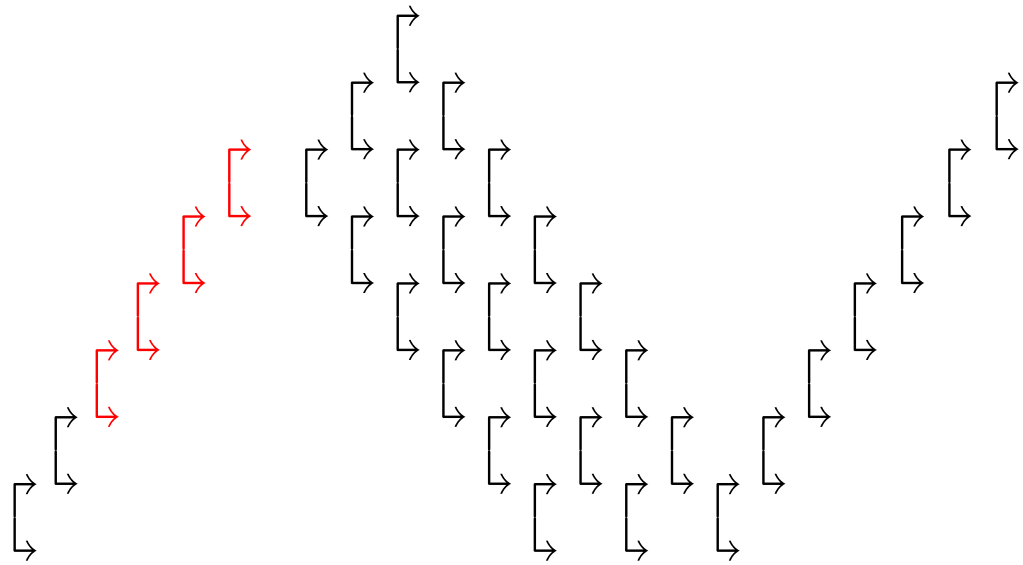
$R_B Q_{k-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

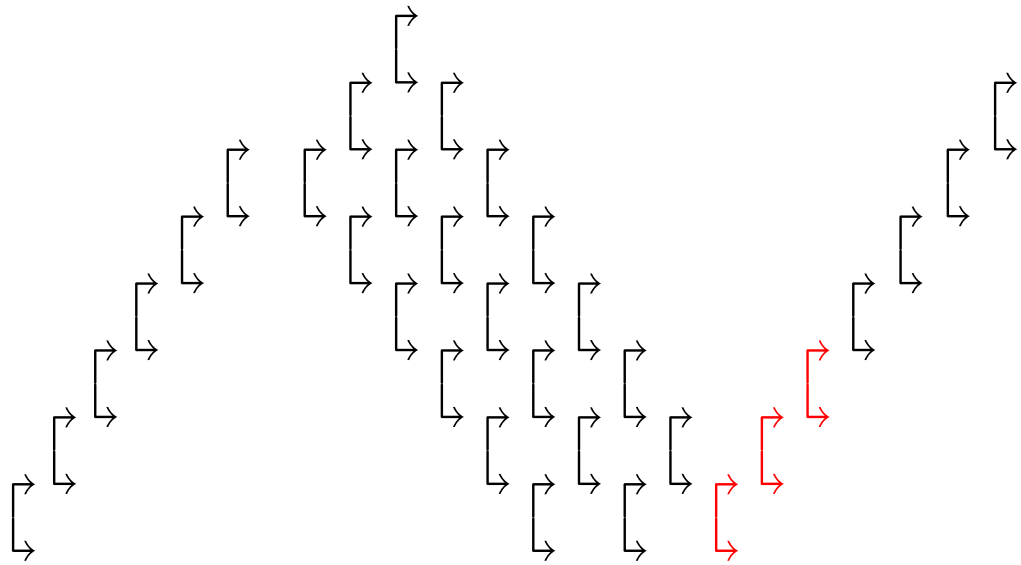
$R_B Q_{k-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

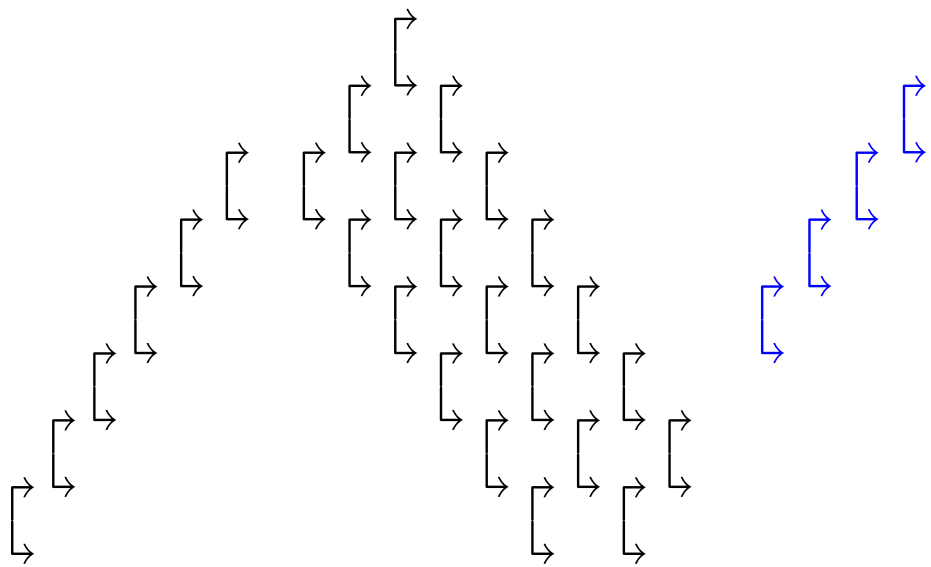
$R_B Q_{i-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

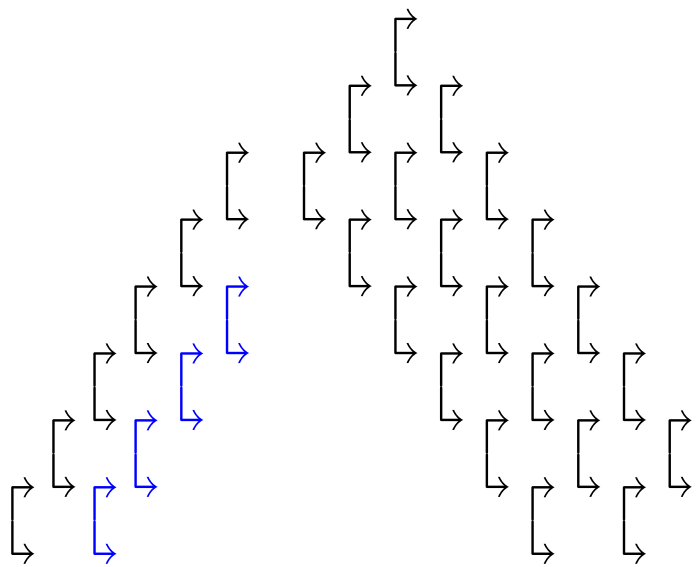
$R_B Q_{i-1} =$



Orthogonal iterations on the structure

R_B is k-upper Hessenberg, Q_{i-1} is s-lower Hessenberg, we can "swap" the two structures

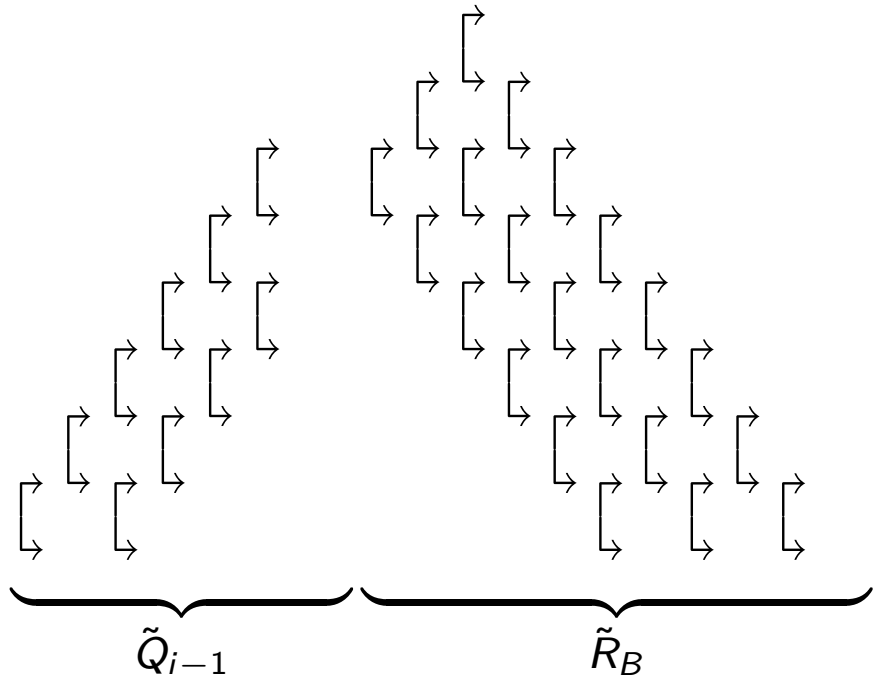
$R_B Q_{i-1} =$



Orthogonal iterations on the structure

We have swapped the structure

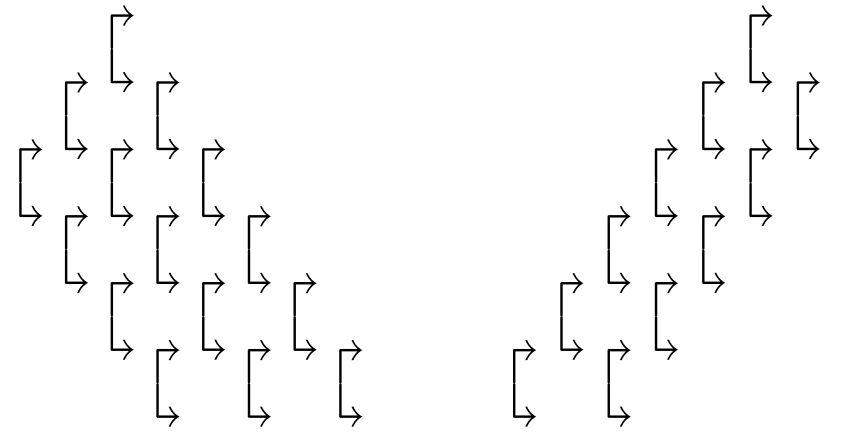
$$R_B Q_{k-1} = \tilde{Q}_{i-1} \tilde{R}_B =$$



This can be performed in $O(nks)$ flops.

Orthogonal iterations on the structure

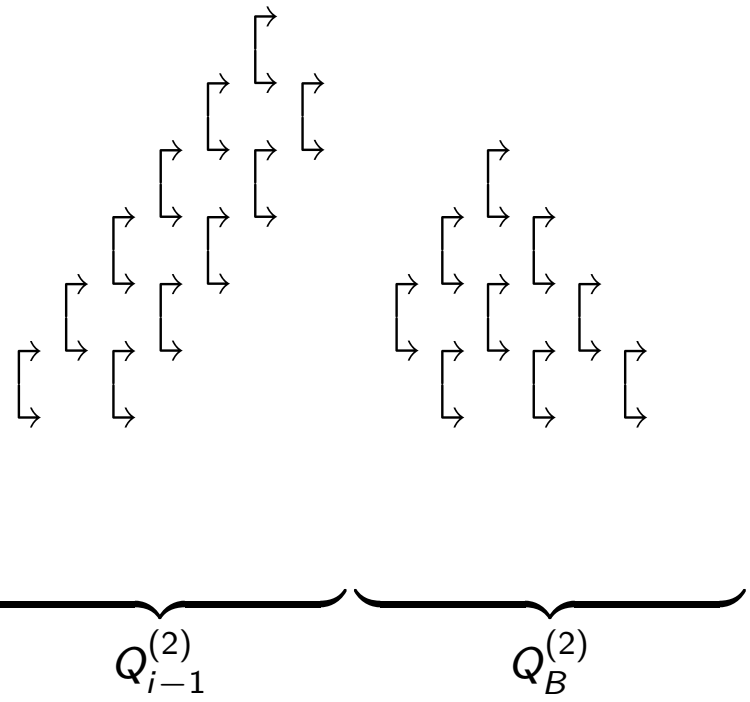
In the case B is k-upper Hessenberg, $Q_B^{(1)}$ has the same structure, we perform a swap of the structure $V_{i-1} = Q_B^{(1)} \hat{Q}_{i-1} = Q_{i-1}^{(2)} Q_B^{(2)}$ we are in the same situation as before

$V_{i-1} =$ 

$Q_B^{(1)}$ $\hat{Q}_{i-1}$

Orthogonal iterations on the structure

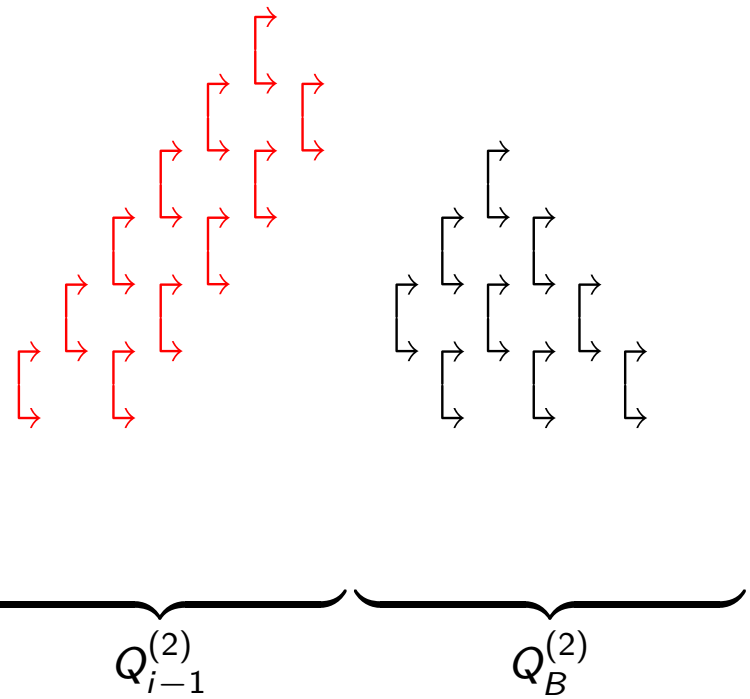
In the case B is k-upper Hessenberg, $Q_B^{(1)}$ has the same structure, we perform a swap of the structure $V_{i-1} = Q_B^{(1)} \hat{Q}_{i-1} = Q_{i-1}^{(2)} Q_B^{(2)}$ we are in the same situation as before

$V_{i-1} =$ 

Orthogonal iterations on the structure

In the case B is k-upper Hessenberg, $Q_B^{(1)}$ has the same structure, we perform a swap of the structure $V_{i-1} = Q_B^{(1)} \hat{Q}_{i-1} = Q_{i-1}^{(2)} Q_B^{(2)}$ we are in the same situation as before

$V_{i-1} =$



The first s columns of V_{i-1} are identified by the s ascending sequences of Givens rotations

Orthogonal iterations on the structure

- Goal $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$
- The same reasoning can be applied to $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$.
- We have to swap factors until the RQ decomposition of $V_{i-1}^H A$ is restored.
- $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$
- At the end we have the new matrix Q_i already factorized as the product of s ascending sequences of Givens transformations.

Orthogonal iterations on the structure

- Goal $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$
- The same reasoning can be applied to $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$.
- We have to swap factors until the RQ decomposition of $V_{i-1}^H A$ is restored.
- $V_{i-1}^H A = \tilde{L}_A \tilde{V}_{i-1}^H (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$
- At the end we have the new matrix Q_i already factorized as the product of s ascending sequences of Givens transformations.

Orthogonal iterations on the structure

- Goal $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$
- The same reasoning can be applied to $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$.
- We have to swap factors until the RQ decomposition of $V_{i-1}^H A$ is restored.
- $V_{i-1}^H A = \tilde{L}_A (I + T_A \tilde{Z}_A^H) \tilde{V}_{i-1}^H R_A^{(1)} Q_A^{(1)}$
- At the end we have the new matrix Q_i already factorized as the product of s ascending sequences of Givens transformations.

Orthogonal iterations on the structure

- Goal $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$
- The same reasoning can be applied to $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$.
- We have to swap factors until the RQ decomposition of $V_{i-1}^H A$ is restored.
- $V_{i-1}^H A = \tilde{L}_A (I + T_A \tilde{Z}_A^H) \tilde{R}_A \bar{V}_{i-1}^H \tilde{Q}_A^{(1)}$
- At the end we have the new matrix Q_i already factorized as the product of s ascending sequences of Givens transformations.

Orthogonal iterations on the structure

- Goal $V_{i-1}^H A = \tilde{R}_i \tilde{Q}_i$
- The same reasoning can be applied to $V_{i-1}^H A = V_{i-1}^H L_A (I + T_A \tilde{Z}_A^H) R_A^{(1)} Q_A^{(1)}$.
- We have to swap factors until the RQ decomposition of $V_{i-1}^H A$ is restored.
- $V_{i-1}^H A = \tilde{L}_A (I + T_A \tilde{Z}_A^H) \tilde{R}_A Q_A^{(2)} Q_i$
- At the end we have the new matrix Q_i already factorized as the product of s ascending sequences of Givens transformations.

Retrieving the eigenvalues

- At convergence construct the $s \times s$ pencil $(Q_i^H A Q_i, Q_i^H B Q_i)$
- The smallest s eigenvalues of (A, B) are approximated by the eigenvalues of $(Q_i^H A Q_i, Q_i^H B Q_i)$
- In our experiments we used Matlab `eig`

Cost and Forward error

This method costs $O(nks)$ flops per iterations if we start from a pencil (A, B) in the LFR format and with A and B as in the linearizations considered.
Similarly the cost is $O(nks)$ per iteration if (A, B) is Hessenberg-Triangular.

We consider the following relative forward error estimate

$$err(\lambda) = \frac{\|T(\lambda)v\|_2}{\|T(\lambda)\|_2\|v\|_2} = \frac{\sigma_k}{\sigma_1}, \text{for } k > 1, \quad err(\lambda) = \sigma_1, \text{for } k = 1,$$

- λ is the approximated eigenvalue inside the unit circle,
- $\sigma_i = \sigma_i(T(\lambda)), 1 = 1, \ldots, k$ are the singular values of $T(\lambda)$
- v is its $k-$th right singular vector of $T(\lambda)$.

Backward Error

A possible measure of the backward error is

$$bk_{err} = \frac{\sigma_{s+1}([AQ_i, BQ_i])}{\sigma_1([AQ_i, BQ_i])}$$

at convergence $AQ_i \approx BQ_i$

Numerical Experiments

From the NLEVP collection, non polynomial problems. We use the stopping condition

$$\|(I - Q_{k-1}Q_{k-1}^H)Q_k\| < 1.0e-14$$

- Time-delay equation $T(z) = z + T_0 + T_1 \exp(6z - 1)$ with

$$T_0 = \begin{bmatrix} 4 & -1 \\ -2 & 5 \end{bmatrix}; \quad T_1 = \begin{bmatrix} -2 & 1 \\ 4 & -1 \end{bmatrix}.$$

  This function has three eigenvalues inside the unit circle.
- Model of cancer growth $T(z) = z - A_0 - A_1 \exp(-rz)$, where This function has three eigenvalues inside the unit circle.
- Neutral functional differential equation $t(z) = -1 + 0.5z + z^2 + hz^2 \exp(\tau z)$. For our choice of the parameters h, τ the function has three eigenvalues inside the unit circle.

Numerical Experiments

From the NLEVP collection

- Spectral abscissa optimization $T(z) = zI_3 - A - B\exp(-z\tau)$, $k = 3$ Abscissa optimization techniques favor multiple roots and clustered eigenvalues with potential numerical difficulties. This function has 4 eigenvalues inside the unit circle.
- Hadeler problem $T(z) = (\exp(z) - 1)A_2 + z^2 A_1 - \alpha A_0$, two real eigenvalues $0 < \lambda < 1$.

Numerical Experiments

Time Delay (k=2, s=3))

nodes	Structure	*deg*	$err(\lambda_1)$	$err(\lambda_2)$	$err(\lambda_3)$	it	bk_{err}
ω^j_{deg+1}	comp	32	8.47e-13	1.20e-12	1.21e-12	50	7.90e-13
$\omega^j_{deg}, 0$	comp	32	8.47e-13	8.48e-13	2.56e-12	53	7.90e-13
$\omega^j_{deg}, 0$	u.diag	40	9.43e-15	4.34e-14	3.43e-13	51	1.06e-12
Cheb	comp	32	5.07e-13	6.25e-10	6.25e-10	53	4.73e-13

Cancer growth (k=2, s=4)

nodes	Structure	*deg*	$err(\lambda_1)$	$err(\lambda_2)$	$err(\lambda_3)$	it	bk_{err}
ω^j_{deg+1}	comp	29	2.65e-14	2.87e-14	8.83e-14	34	9.57e-14
$\omega^j_{deg}, 0$	comp	29	1.67e-15	1.83e-14	2.52e-14	33	9.57e-14
$\omega^j_{deg}, 0$	unit diag	32	8.14e-16	3.47e-15	6.16e-15	36	7.68e-15
Cheb	comp	29	1.77e-15	1.27e-14	4.40e-12	36	7.53e-14

Complex eigenvalues: Interpolation on Chebyshev point is not adequate

Numerical Experiments

Neutral functional differential equation (k=1, s=2)

nodes	Structure	*deg*	$err(\lambda_1)$	$err(\lambda_2)$	it	bk_{err}
ω^j_{deg+1}	comp	60	3.19e-13	3.70e-13	37	4.36e-15
$\omega^j_{deg}, 0$	comp	60	1.71e-13	2.11e-13	37	3.85e-15
$\omega^j_{deg}, 0$	unit diag	64	9.35e-12	9.64e-12	40	3.19e-13
Cheb	comp	62	8.90e-07	8.90e-07	55	5.66e-15

Complex eigenvalues: Interpolation on Chebyshev point is not adequate

Spectral abscissa optimization (k=3, s=6)

nodes	Structure	*deg*	$err(\lambda_1)$	$err(\lambda_2)$	$err(\lambda_3)$	it	bk_{err}
ω^j_{deg+1}	comp	26	7.09e-12	7.09e-12	5.37e-11	89	4.34e-14
$\omega^j_{deg}, 0$	comp	26	1.02e-11	1.02e-11	2.42e-11	90	4.33e-14
$\omega^j_{deg}, 0$	unit diag	48	1.49e-14	1.58e-14	1.73e-14	116	8.20e-13
Cheb	comp	30	6.87e-15	7.08e-15	2.73e-09	116	1.21e-13

Numerical Experiments

Hadeler function (k=8, s=2)

nodes	Structure	*deg*	$err(\lambda_1)$	$err(\lambda_2)$	it	bk_{err}
ω^j_{deg+1}	comp	14	7.51e-14	3.38e-12	75	1.13e-14
$\omega^j_{deg}, 0$	comp	14	7.84e-14	6.98e-13	77	8.39e-15
$\omega^j_{deg}, 0$	unit diag	16	1.76e-15	8.78e-15	81	4.82e-15
Cheb	comp	16	1.35e-15	3.29e-13	80	4.53e-14

Conclusions

- In general we get better approximations than applying the QR method to the same matrices
- For the companion pencil $nks(\#iter)$ versus n^2k for QR/QZ algorithms
- Prove the backward stability of the method
- Accelerating convergence trying incorporating Rayleight-Ritz steps and a sort of deflation
- Design a fast method to reduce (in a backward stable way) (A, B) to (H, T) working on the LFR representation