Fast QR iterations for unitary plus low rank matrices

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

Dipartimento di Informatica, Università di Pisa, Italy

NuMA 2018

The Problem

- Eigenvalues computation $Av = \lambda v$
- if we need all the eigenvalues
- Shifted QR algorithm

$$A_k - \mu_k I = QR$$
$$A_{k+1} = RQ + \mu_k I$$

repeat!

- Textbook approach
- First reduce the matrix to Hessenberg form

The implicit QR algorithm

On upper Hessenberg matrices

- Inizialization phase:
  - Pick the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $\tilde{A} = G_1 A G_1^H$.

$\tilde{A}$ is no more in Hessenberg form: a bulge is formed

- Chasing the bulge to restore the Hessenberg structure

The implicit QR algorithm

On upper Hessenberg matrices

- Inizialization phase:
  - Pick the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $\tilde{A} = G_1 A G_1^H$.

$\tilde{A}$ is no more in Hessenberg form: a bulge is formed

- Chasing the bulge to restore the Hessenberg structure

No QR decomposition needed!! Only 2×2 Givens rotations

The implicit QR algorithm

On upper Hessenberg matrices

- Inizialization phase:
  - Pick the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $\tilde{A} = G_1 A G_1^H$.

$\tilde{A}$ is no more in Hessenberg form: a bulge is formed

- Chasing the bulge to restore the Hessenberg structure

No QR decomposition needed!! Only 2×2 Givens rotations
Implicit Q Theorem $\rightarrow$ the same of an explicit QR step

The implicit QR algorithm

On upper Hessenberg matrices

- Inizialization phase:
  - Pick the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $\tilde{A} = G_1 A G_1^H$.

$\tilde{A}$ is no more in Hessenberg form: a bulge is formed

- Chasing the bulge to restore the Hessenberg structure

No QR decomposition needed!! Only 2×2 Givens rotations
Works fabulously with multiple shifts

The implicit QR algorithm

$$A - \mu I = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Apply Q_0: combine the first two rows and the first two columns of A

The implicit QR algorithm

$$A - \mu I = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Apply Q_0: combine the first two rows and the first two columns of A

The implicit QR algorithm

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Chase the Bulge

$$\tilde{A} = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \otimes & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

Chase the Bulge

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \otimes & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

Chase the Bulge

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ {\color{red}0} & \times & \times & \times & \times \\ 0 & {\color{red}\otimes} & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

Chase the Bulge

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & {\color{red}0} & \times & \times & \times \\ 0 & 0 & {\color{red}\otimes} & \times & \times \end{bmatrix}$$

Chase the Bulge

$$A_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & {\color{red}0} & \times & \times \end{bmatrix}$$

The Hessenberg structure is restored

Chase the Bulge

$$A_1 = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

The Hessenberg structure is restored

Each step on unstructured matrix costs $O(n^2)$ flops.
Better results with structured matrices!

The Problem

Efficient computation of the eigenvalues of a Unitary-plus-low-rank

$$p(z) = \sum_{i=0}^{n} a_i \phi_i(z)$$

- If $\Phi = \{1, z, z^2, \dots, z^n\}$ we associate the Companion matrix

$$C = \begin{bmatrix} -\frac{a_{n-1}}{a_n} & -\frac{a_{n-2}}{a_n} & \cdots & -\frac{a_0}{a_n} \\ 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{bmatrix}$$

  the eigenvalues are the roots of $p(z)$.
- Fast algorithms exploit the fact that companion are unitary plus rank-one

$$C = U + e_1 p^T, \quad UU^T = I, U \text{ upper Hessenberg}$$

The Problem

- Companion and block companion matrices
- Fellow matrices (whose eigenvalues are the zeros of a linear combination of Szegö polynomials)
- Perfectly unitary structure corrupted by a low rank error
- Unitary-diagonal matrix plus low rank (interpolation techniques for solving nonlinear eigenproblems)

We assume $A = U + XY^H$ already in Hessenberg form.

The problem

- QR implicit steps on unitary-plus-rank-k
- Need to take advantage from the structure of the matrix otherwise to compute all the eigenvalues we need $O(n^3)$ flops.

$$A = U + XY^H,$$

$U \in \mathbb{C}^{n\times n}$ unitary, $X, Y \in \mathbb{C}^{n\times k}$.

$$\begin{aligned} A &= QR \\ A^{(1)} &= Q^H A Q = Q^H U Q + Q^H X\, Y^H Q = U_1 + X_1 Y_1^H \end{aligned}$$

is still Unitary plus low rank!

- Keyword is representation preserved by QR-steps

Related work

Most authors analyze the companion /block companion case... nothing, to the best of our knowledge, on unitary plus low rank

- 2004 Bini, Daddi, Gemignani (explicit QR)
- 2007 Bini, Eidelman, Gemignani, Gohberg (explicit QR on unitary-plus-rank-1)
- 2007 Chandrasekeran, Gu, Xia, Zhu (implicit QR on a QR factorization of the companion)
- 2012 Delvaux, Frederix, 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

The algorithm

The algorithm consists of

- a preliminary phase (we embed the original matrix into a $(n+k) \times (n+k)$ matrix still unitary +rank-k

$$\hat{A} = \left[\begin{array}{c|c} A & * \\ \hline 0_{kn} & 0_{kk} \end{array}\right] = \hat{U} + \hat{X}\hat{Y}^H$$

- computation of the representation of $\hat{A}$
- iterative phase - actual implicit shifted QR step

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

$A = U + XY^H$, hence

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ \times & \times & \times & \times & \cdot & \cdot \\ \times & \times & \times & \times & \times & \cdot \\ \times & \times & \times & \times & \times & \times \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix}}_{Y^H}$$

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

$A = U + XY^H$, hence

$$\underbrace{\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \boxed{\times} & \times & \times & \cdot & \cdot & \cdot \\ \boxed{\times} & \boxed{\times} & \times & \times & \cdot & \cdot \\ \boxed{\times} & \boxed{\times} & \boxed{\times} & \times & \times & \cdot \\ \boxed{\times} & \boxed{\times} & \boxed{\times} & \boxed{\times} & \times & \times \end{bmatrix}}_{U} + \underbrace{\begin{bmatrix} \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \\ \times & \times \end{bmatrix}}_{X} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times & \times \end{bmatrix}}_{Y^H}$$

that is $\operatorname{rank}(U(h+1:n, 1:h-1)) = k$, $h = 2, \ldots, n-1$

Representation

Consider first the case $k = 1$. $A = U + xy^H$,

- L lower unitary Hessenberg matrix such that $L^H x = \alpha e_1$
- $L^H U$ is a generalized unitary upper Hessenberg with two diagonals

$$L^H U = \begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ 0 & \times & \times & \times & \cdot & \cdot \\ 0 & 0 & \times & \times & \times & \cdot \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix}$$

Representation

$$A = L \left(\underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ 0 & \times & \times & \times & \cdot & \cdot \\ 0 & 0 & \times & \times & \times & \cdot \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix}}_{L^H U} + \underbrace{\begin{matrix} \times & \times & \times & \times & \times & \times \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{matrix}}_{\alpha e_1 y^H} \right)$$

Representation

$$A = L \left(\underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \times & \cdot & \cdot & \cdot \\ 0 & \times & \times & \times & \cdot & \cdot \\ 0 & 0 & \times & \times & \times & \cdot \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix}}_{L^H U} + \underbrace{\begin{matrix} \times & \times & \times & \times & \times & \times \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{matrix}}_{\alpha e_1 y^H} \right)$$

- We can "peal off" another diagonal from $L^H U$

Representation

$$A = L \left(\underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ 0 & \times & \times & \cdot & \cdot & \cdot \\ 0 & 0 & \times & \times & \cdot & \cdot \\ 0 & 0 & 0 & \times & \times & \cdot \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}}_{L^H U R^H} + \underbrace{\begin{matrix} \times & \times & \times & \times & \times & \times \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{matrix}}_{\alpha e_1 z^H} \right) R$$

- $A = L(L^H U R^H + \alpha e_1 z^H) R$
-

Representation

$$A = L \left(\underbrace{\begin{bmatrix} \times & \cdot & \cdot & \cdot & \cdot & \cdot \\ \times & \times & \cdot & \cdot & \cdot & \cdot \\ 0 & \times & \times & \cdot & \cdot & \cdot \\ 0 & 0 & \times & \times & \cdot & \cdot \\ 0 & 0 & 0 & \times & \times & \cdot \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}}_{L^H U R^H} + \underbrace{\begin{matrix} \times & \times & \times & \times & \times & \times \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \\ & & & & & \end{matrix}}_{\alpha e_1 z^H} \right) R$$

- $A = L{\color{red}F}R$
- O(nk) Givens rotations and O(k) vectors

Representation

Skipping all the technicalities about the embedding

Theorem

Let $\hat{A} = \hat{U} + \hat{X}\hat{Y}^H$, then there exist L unitary k-lower Hessenberg, R unitary k-upper Hessenberg, $F = Q + I_{n+k,k}Z^H$ such that

$$\hat{A} = L \cdot F \cdot R. \tag{1}$$

If A is proper ($a_{i+1,i} \neq 0$) and nonsingular also L and R are proper.

An implicit QR step

Single shift case

- Inizialization phase:
  - Compute the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $G_1 A G_1^H$.
- Chasing the bulge to restore the Hessenberg structure

An implicit QR step

Single shift case

- Inizialization phase:
  - Compute the shift μ,
  - Retrieve the vector $x = (A - \mu I)e_1$.
  - x which has only two nonzeros.
  - Compute G_1 such that $G_1 x = \alpha e_1$.
  - Perform the similarity transformation $G_1 A G_1^H$.
- Chasing the bulge to restore the Hessenberg structure

Matrices L, R and F can be represented in terms of Givens rotations acting on two consecutive rows

An implicit QR step

- $A = L\,(Q + e_1 z^H)\,R$

An implicit QR step

- $A = L\,(Q + e_1 z^H)\,R$
- $G_1 A = G_1 L\,(Q + e_1 z^H)\,R = \tilde{L} G_2\,(Q + e_1 z^H)\,R = \tilde{L}(G_2 Q + G_2 e_1 z^H) R = \tilde{L}(\tilde{Q} + e_1 z^H) R$

An implicit QR step

- $A = L\,(Q + e_1 z^H)\,R$
- $G_1 A = G_1 L\,(Q + e_1 z^H)\,R = \tilde{L} G_2\,(Q + e_1 z^H)\,R = \tilde{L}(G_2 Q + G_2 e_1 z^H)R = \tilde{L}(\tilde{Q} + e_1 z^H)R$
- $$\begin{aligned} \tilde{A} = G_1 A G_1^H &= \tilde{L}(\tilde{Q} + e_1 z^H) R G_1^H \\ &= \tilde{L}(\tilde{Q} + e_1 z^H)\tilde{G}_2^H \tilde{R} \\ &= G_2 \hat{L}(\hat{Q} + e_1 \tilde{z}^H)\tilde{R} \end{aligned}$$

An implicit QR step

- $A = L\,(Q + e_1 z^H)\,R$
- $G_1 A = G_1 L\,(Q + e_1 z^H)\,R = \tilde{L} G_2\,(Q + e_1 z^H)\,R =$
  $\tilde{L}(G_2 Q + G_2 e_1 z^H)R = \tilde{L}(\tilde{Q} + e_1 z^H)R$
- $$\begin{aligned}\tilde{A} = G_1 A G_1^H &= \tilde{L}(\tilde{Q} + e_1 z^H) R G_1^H \\ &= \tilde{L}(\tilde{Q} + e_1 z^H)\tilde{G}_2^H \tilde{R} \\ &= G_2 \hat{L}(\hat{Q} + e_1 \tilde{z}^H)\tilde{R}\end{aligned}$$
- To remove the bulge multiply $G_2^H \tilde{A} G_2$. After $n-1$ steps of bulge chasing we recover the Hessenberg structure

Deflation

- The outermost diagonal entries of L and R are always nonzero,
- Deflations can be detected directly on the representation
- $F = Q + I_{nk} Z^H$, Q unitary Hessenberg tends to the triangular form when Q becomes diagonal

$a_{i+1,i} = 0, i = 1, \dots n-1$ iff $q_{i+k,i} = 0$.

Deflation criteria

If $|q_{i+k+1,i+k}| < \varepsilon \prod_{j=1}^{n} |l_{j,j+k}|$ then $|a_{i+1,i}| < \varepsilon$.

Computational cost

- Each step costs $\mathcal{O}(nk)$ flops.
- Total cost of the procedure $\mathcal{O}(n^2k)$.

Applying it to the matrix polynomial eigenvalue problem, with $P(\lambda) = A_0 + A_1\lambda + \cdots + I_m\lambda^d$ and $A_i \in \mathbb{C}^{m\times m}$, we have $n = md$, $k = m$, and we get a cost of $O(m^3d^2)$ which is claimed to be the best achievable by implicit QR.

Backward Stability

The algorithm consists of

- a preliminary phase (we embed the original matrix into a $(n+k)\times(n+k)$ matrix still unitary +rank-k

$$\hat{A} = \left[\begin{array}{c|c} A & * \\ \hline 0_{kn} & 0_{kk} \end{array}\right] = \hat{U} + \hat{X}\hat{Y}^H$$

- computation of the representation
- iterative phase

The preliminary phase is backward stable: error$\approx \varepsilon \|A\|_2$.

Backward Stability: representation

Theorem

- Given $\hat{A} = \hat{U} + \hat{X}\hat{Y}^H$ whose the exact representation is $\hat{A} = LFR$
- Denote by $\tilde{A} = \tilde{L}\tilde{F}\tilde{R}$ the computed one.
- $\tilde{A} = \hat{A} + E$ and $\|E\|_2 \leq c\,\varepsilon\,\|A\|_2$

Backward Stability: implicit QR step

Theorem

Let $\hat{A}^{(1)}$ be the result computed in floating point arithmetic of a QR step on matrix A. Then there exists a perturbation ΔA such that $\hat{A}^{(1)} = P^H(A + \Delta A)P$, and $\|\Delta A\|_2 \approx K_{N,\varepsilon}\mathcal{O}(\varepsilon)\|A\|_2$.

The algorithm is backward stable!

Numerical results

We performed experiments on several classes of matrices

- Scalar polynomials whose roots are known
- Scalar polynomials whose roots are unknown
- Random Fellow matrices with a prescribed norm
- Matrix polynomials from the NLEVP collection
- Fiedler penta-diagonal companion matrices associated to scalar polynomials
- Random unitary plus rank k matrices.
- Random unit diagonal plus random rank k matrices.

Error measures

- Let λ be the "correct" eigenvalue and $\hat{\lambda}$ the closest computed
- Let P_m be the accumulated unitary similarity transformation at the end of the iterative process consisting of m-steps
- A_m is the reconstructed matrix from the representation by the factors L_m, F_m, R_m.

- Forward average error $forw_{err} = \frac{1}{n} \sum_{i=1}^{n} \frac{|\lambda_i - \hat{\lambda}_i|}{|\lambda_i|}$
- Backward error $back_{err} = \frac{\|P_m^H A P_m - A_m\|_\infty}{\|A\|_\infty}$

Error measures

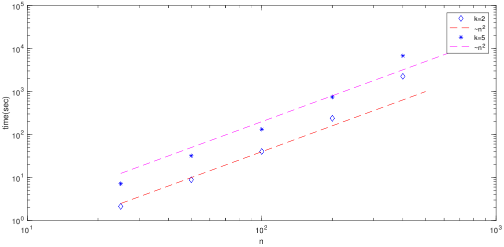
- Let λ be the "correct" eigenvalue and $\hat{\lambda}$ the closest computed
- Let P_m be the accumulated unitary similarity transformation at the end of the iterative process consisting of m-steps
- A_m is the reconstructed matrix from the representation by the factors L_m, F_m, R_m.

- Forward average error $forw_{err} = \frac{1}{n} \sum_{i=1}^{n} \frac{|\lambda_i - \hat{\lambda}_i|}{|\lambda_i|}$
- Backward error $back_{err} = \frac{\|P_m^H A P_m - A_m\|_\infty}{\|A\|_\infty}$

Rule of thumb:

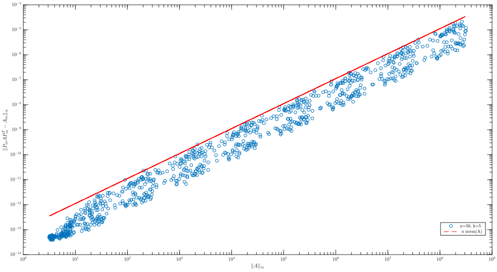
$$\text{forward error} \lesssim \text{condition number} \cdot \text{backward error}$$

Time Complexity



- Confirms that the cost is quadratic in n.
- There is a loss of performance for higher values of n

Backward Error



Dependence of the backward error from the norm of the matrix.

Numerical results

Scalar polynomials

#	degree	$\|\text{condeig}(A)\|_\infty$	forw_{err}	back_{err}	av_{iter}
wil	20	2.47e+27	6.65e-04	2.01e-15	3.29
sswil	20	2.63e+05	7.44e-13	1.93e-15	2.52
proot	20	2.15e+25	4.28e-04	4.61e-15	3.14
2^i	20	6.85e+17	7.34e-05	2.27e-15	4.95
cheb	30	6.31e+10	2.45e-07	2.53e-15	2.61
$\sum x^i$	30	5.57e+00	4.98e-15	2.37e-15	2.64
$(x-1)^n$	10	4.17e+12	4.88e-02	2.36e-15	3.00
clustered	32	4.73e+27	1.58e-01	4.22e-15	3.94
bern	20	9.14e+06	1.86e-13	3.86e-14	3.85
p_1	40	2.08e+01	2.14e-14	2.36e-15	3.00
p_2	29	2.69e+01	5.43e-16	4.23e-15	3.06
p_6	15	2.65e+02	4.00e-16	1.41e-15	3.17
p_7	30	1.87e+05	4.64e-13	9.67e-15	3.32

Table: scalar companion

Numerical results

From the NLEVP collection

#	n	k	$\|\mathrm{condeig}(\mathrm{A})\|_\infty$	$\|A\|_\infty$	$forw_{err}$	$back_{err}$
bicycle	6	2	5.70e+02	9.62e+03	2.60e-15	8.29e-16
clloop	6	2	9.00e+00	3.00e+00	8.99e-16	1.42e-15
qep2	9	3	1.80e+16	4.00e+00	3.31e-09	3.65e-16
spring	15	5	2.33e+00	8.23e+01	3.00e-16	1.93e-15
powplant	24	8	1.72e+05	3.73e+07	7.13e-08	2.68e-15
mtlstrip	27	9	1.71e+02	3.48e+02	7.78e-16	2.42e-15
ocam1	27	9	5.04e+15	1.73e+05	4.03e-07	2.60e-15
acwave1d	30	10	5.96e+01	1.37e+01	1.42e-15	1.02e-14
wiresaw1	30	10	1.57e+01	1.42e+03	6.00e-14	4.20e-15
ocam2	45	15	4.66e+17	6.22e+07	1.16e-02	4.90e-15
orrsom	50	10	1.88e+06	9.67e+00	1.83e-14	6.35e-15
hospital	72	24	4.49e+01	1.11e+04	7.84e-13	2.57e-14
dirac	240	80	2.11e+03	1.38e+03	5.24e-14	1.39e-13
sign1	243	81	3.29e+09	1.53e+01	4.10e-09	5.84e-14
butterfly	320	64	2.97e+01	5.18e+01	5.15e-14	1.29e-13

Numerical results

n	k	$norm(A)$	$\|\text{condeig}(A)\|_\infty$	$back_{err}$
50	1	7.14e+00	4.19e+00	9.45e-15
50	1	9.70e+04	3.52e+01	3.10e-15
50	2	7.19e+00	3.78e+00	9.92e-15
50	2	9.83e+04	3.27e+01	2.91e-15
50	25	8.27e+00	1.00e+15	1.10e-14
50	25	9.98e+04	7.85e+14	3.15e-15
100	1	1.00e+01	1.37e+01	1.79e-14
100	1	9.85e+04	2.04e+02	4.86e-15
100	2	1.01e+01	1.97e+01	1.89e-14
100	2	9.91e+04	1.80e+02	4.78e-15
100	25	1.10e+01	2.18e+07	2.07e-14
100	25	9.99e+04	1.79e+06	5.21e-15

Table: Unitary plus low rank : average on 50 random tests

Numerical results

n	k	$\|\mathrm{condeig(A)}\|_\infty$	$forw_{err}$	$back_{err}$
50	1	2.25e+00	2.66e-16	2.78e-15
50	1	2.25e+00	3.11e-17	2.32e-15
50	2	1.40e+01	1.57e-16	2.78e-15
50	2	2.18e+01	8.83e-14	2.63e-15
50	25	5.59e+01	7.22e-17	2.58e-15
50	25	9.34e+01	8.63e-13	2.21e-15

Table: Unitary diagonal plus low rank: average on 50 random tests

Conclusions and ...

What we have done

- Implicit QR iterations working on the structure
- Algorithm is fast and backward stable
- Everything has been mathematically proved and confirmed by experimental results

What shall be done

- Balancing
- Multishift
- Similar techniques for the QZ