QR-algorithms for eigenvalue computation of structured matrices. The case of low-rank corrections of unitary matrices

L. Gemignani

University of Pisa

gemignan@dm.unipi.it

http://www.dm.unipi.it/~gemignan

The Companion Matrix

$$a(z) = \sum_{i=0}^{n} a_i z^i, \qquad A = \begin{bmatrix} 0 & \dots & 0 & -a_0/a_n \\ 1 & \ddots & \vdots & -a_1/a_n \\ & \ddots & 0 & \vdots \\ & & 1 & -a_{n-1}/a_n \end{bmatrix}$$

- Eigenvalues of A == Zeros of $a(z)$
- A is upper Hessenberg
- MATLAB uses the shifted QR eigenvalue algorithm for the computation of polynomial zeros
- MATLAB only exploits the Hessenberg structure

The (Shifted) QR Algorithm

$$
\begin{aligned}
A_0 &= A \\
A_k - \alpha_k I_n &= Q_k R_k \\
A_{k+1} &:= R_k Q_k + \alpha_k I_n,
\end{aligned}
$$

- Classical Theory: The Hessenberg structure is invariant under QR (A_0 Hessenberg $\Rightarrow$ A_1 Hessenberg)
- $O(n^2)$ flops for each step of QR applied to a Hessenberg matrix
- The MATLAB implementation generally requires $O(n^3)$ flops and $O(n^2)$ storage for computing all the zeros

The Research Problem

To design a QR eigenvalue algorithm for companion matrices which requires $O(n)$ flops and $O(n)$ storage per iteration

- Recent Contributions:
  1. [Bini, Daddi, G.]. (To appear in ETNA) FAST but UNSTABLE
  2. [Chandrasekaran, Gu]. (Talk at the Workshop in Banff, November 2003) www.pims.math.ca/birs/workshops/2003/03w5008 FAST but (probably) UNSTABLE
- The main goal: To achieve both efficiency and stability

The Novel Approach

$$A = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 1 & \ddots & \vdots & 0 \\ & \ddots & 0 & \vdots \\ 0 & & 1 & 0 \end{bmatrix} + \boldsymbol{a}\boldsymbol{e}_n^T = C + \boldsymbol{a}\boldsymbol{e}_n^T$$

- C is a very special unitary Hessenberg matrix
- Exploit the representation of A as a unitary Hessenberg plus a rank-one matrix
- Unitary Hessenberg matrices belong to the class of rank-structured matrices

Unitary Hessenberg Matrices

1. Let H be unitary Hessenberg and $H = QR$ its QR factorization, where R has nonnegative diagonal entries
2. We have $R = I_n$ and, therefore, $H = Q$

$$H = \begin{bmatrix} -\bar{\alpha}_0\alpha_1 & -\bar{\alpha}_0\beta_1\alpha_2 & \dots & \dots & -\bar{\alpha}_0\beta_1\cdots\beta_{n-1}\alpha_n \\ \bar{\beta}_1 & -\bar{\alpha}_1\alpha_2 & \dots & \dots & -\bar{\alpha}_1\beta_2\cdots\beta_{n-1}\alpha_n \\ & \bar{\beta}_2 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \bar{\beta}_{n-1} & -\bar{\alpha}_{n-1}\alpha_n \end{bmatrix}$$

[Gill, Golub, Murray, Saunders , 1974]

The Rank Structure

$$\ell(H) = \max_{1 \le j \le n-1} \text{rank}\,(H[j+1:n, 1:j])$$

$$\wp(H) = \max_{1 \le j \le n-1} \text{rank}\,(H[1:j, j+1:n])$$

- $\ell(H) \le 1$ and $\wp(H) \le 1$ for H unitary Hessenberg
- Depending on some small differences in the definition and in the representation, we may have:

1. Quasiseparable matrices [Eidelman, Gohberg]
2. Matrices with low Hankel rank [Dewilde, Van der Veen]
3. Weakly semiseparable matrices [Tyrtyshnikov]
4. Sequentially/Hierarchically semiseparable matrices [Chandrasekaran, Gu]

Companion Matrices Under QR

Theorem 1 *Let $\{A_k\}$ be the sequence of matrices generated by the shifted QR algorithm applied to the companion matrix $A = A_0$. Then $\ell(A_k) \le 1$ and $\wp(A_k) \le 3$ for all k*

Proof: From $A_k = Q_k A_{k-1} Q_k^H$ and $A_0 = C_0 + \boldsymbol{u}^{(0)} \boldsymbol{w}^{(0)H}$, one gets

$$A_k = C_k + \boldsymbol{u}^{(k)} \boldsymbol{w}^{(k)H}, \quad C_k \text{ unitary Hessenberg}$$

Since A_k is Hessenberg, it also holds

$$c_{i,j}^{(k)} = -u_i^{(k)} \bar{w}_j^{(k)}, \text{ for } i - j \ge 2$$

To conclude: Characterize the rank structure of such a matrix C_k

Unitary Rank-Structured Matrices

Theorem 2 *Let C be a unitary matrix such that tril$(C,-2) =$ tril$(\boldsymbol{u}\boldsymbol{w}^H,-2)$. Then, $\wp(C) \leq 2$. In particular, there exist $n-1$ vectors $\boldsymbol{q}_j \in \mathbb{C}^2$, $1 \leq j \leq n-1$, $n-1$ vectors $\boldsymbol{t}_j \in \mathbb{C}^2$, $2 \leq j \leq n$ and $n-2$ lower triangular matrices $B_j \in \mathbb{C}^{2\times 2}$, $2 \leq j \leq n-1$, such that*

$$c_{i,j} = \boldsymbol{q}_i^T B_{i+1} \cdots B_{j-1} \boldsymbol{t}_j, \text{ for } i+1 \leq j \leq n$$

Proof: Compute the QR factorization of C. The unitary factor Q is the product of $O(n)$ rotations. The rank structure of C coincides with the rank structure of Q

A Representation for A_k

Theorem 3 *For each matrix A_k generated by shifted QR algorithm applied to the companion matrix $A = A_0$, there exist $n-1$ vectors $\boldsymbol{q}_j^{(k)} \in \mathbb{C}^2$, $n-1$ vectors $\boldsymbol{t}_j^{(k)} \in \mathbb{C}^2$ and $n-2$ lower triangular matrices $B_j^{(k)} \in \mathbb{C}^{2\times 2}$ such that*

$$a_{i,j}^{(k)} = \boldsymbol{q}_i^{(k)T} B_{i+1}^{(k)} \cdots B_{j-1}^{(k)} \boldsymbol{t}_j^{(k)} + u_i^{(k)} \bar{w}_j^{(k)}, \text{ for } i+1 \le j \le n;$$

$$a_{i,i}^{(k)} = a_i^{(k)}, \text{ for } 1 \le i \le n;$$

$$a_{i+1,i}^{(k)} = b_i^{(k)}, \text{ for } 1 \le i \le n-1;$$

- Each iterate A_k can be represented by means of $\simeq 11n$ parameters.

The Structured QR Iteration

Let

$$\Phi(\{\boldsymbol{q}_i^{(k)}\}, \{\boldsymbol{t}_i^{(k)}\}, \{B_i^{(k)}\}, \boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k}, \boldsymbol{u}^{(k)}, \boldsymbol{w}^{(k)}) = A_k$$

The structured QR iteration takes in input

$$(\{\boldsymbol{q}_i^{(k)}\}, \{\boldsymbol{t}_i^{(k)}\}, \{B_i^{(k)}\}, \boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{w}^{(k)}, \overbrace{\alpha_k}^{\text{shift}})$$

and returns in output

$$(\{\boldsymbol{q}_i^{(k+1)}\}, \{\boldsymbol{t}_i^{(k+1)}\}, \{B_i^{(k+1)}\}, \boldsymbol{a}^{(k+1)}, \boldsymbol{b}^{(k+1)}, \boldsymbol{u}^{(k+1)}, \boldsymbol{w}^{(k+1)})$$

such that A_{k+1} is equal to

$$\Phi(\{\boldsymbol{q}_i^{(k+1)}\}, \{\boldsymbol{t}_i^{(k+1)}\}, \{B_i^{(k+1)}\}, \boldsymbol{a}^{(k+1)}, \boldsymbol{b}^{(k+1)}, \boldsymbol{u}^{(k+1)}, \boldsymbol{w}^{(k+1)})$$

A Fast and Stable Structured Iteration

INPUT: $(\{\boldsymbol{q}_i^{(k)}\}, \{\boldsymbol{t}_i^{(k)}\}, \{B_i^{(k)}\}, \boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k}, \boldsymbol{u}^{(k)}, \boldsymbol{w}^{(k)}, \overbrace{\alpha_k}^{\text{shift}})$

$\Phi(\{\boldsymbol{q}_i^{(k)}\}, \{\boldsymbol{t}_i^{(k)}\}, \{B_i^{(k)}\}, \boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{w}^{(k)}) = A_k$

$\Gamma(\{\boldsymbol{q}_i^{(k)}\}, \{\boldsymbol{t}_i^{(k)}\}, \{B_i^{(k)}\}, \boldsymbol{a}^{(k)}, \boldsymbol{b}^{(k)}, \boldsymbol{u}^{(k)}, \boldsymbol{w}^{(k)}) = H_k$

1. Compute Q_k such that $A_k - \alpha_k I_n = Q_k R_k$
2. (Expansion Phase) Compute H_{k+1}:
   (a) $H_{k+1} = Q_k^H H_k Q_k$
   (b) $A_{k+1} := R_k Q_k + \alpha_k I_n$, $H_{k+1} = A_{k+1} - \boldsymbol{u}^{(k+1)} \boldsymbol{v}^{(k+1)^H}$
3. (Compression Phase) Compute a QR factorization of H_{k+1}

OUTPUT:

$(\{\boldsymbol{q}_i^{(k+1)}\}, \{\boldsymbol{t}_i^{(k+1)}\}, \{B_i^{(k+1)}\}, \boldsymbol{a}^{(k+1)}, \boldsymbol{b}^{(k+1)}, \boldsymbol{u}^{(k+1)}, \boldsymbol{w}^{(k+1)})$

A Representation for H_{k+1}

Theorem 4 *At the end of the expansion phase the matrix H_{k+1} admits the following representation. There exist vectors $\boldsymbol{r}_i^{(k+1)} \in \mathbb{C}^3$, vectors $\boldsymbol{z}_i^{(k+1)} \in \mathbb{C}^3$ and matrices $F_i^{(k+1)} \in \mathbb{C}^{3\times 3}$ such that*

$$h_{i,j}^{(k+1)} = \boldsymbol{r}_i^{(k+1)^T} F_{i+1}^{(k+1)} \cdots F_{j-1}^{(k+1)} \boldsymbol{z}_j^{(k+1)}, \text{ for } j-i \geq 1$$

$$h_{i,j}^{(k+1)} = -u_i^{(k+1)} \bar{w}_j^{(k+1)} \text{ for } i-j \geq 2.$$

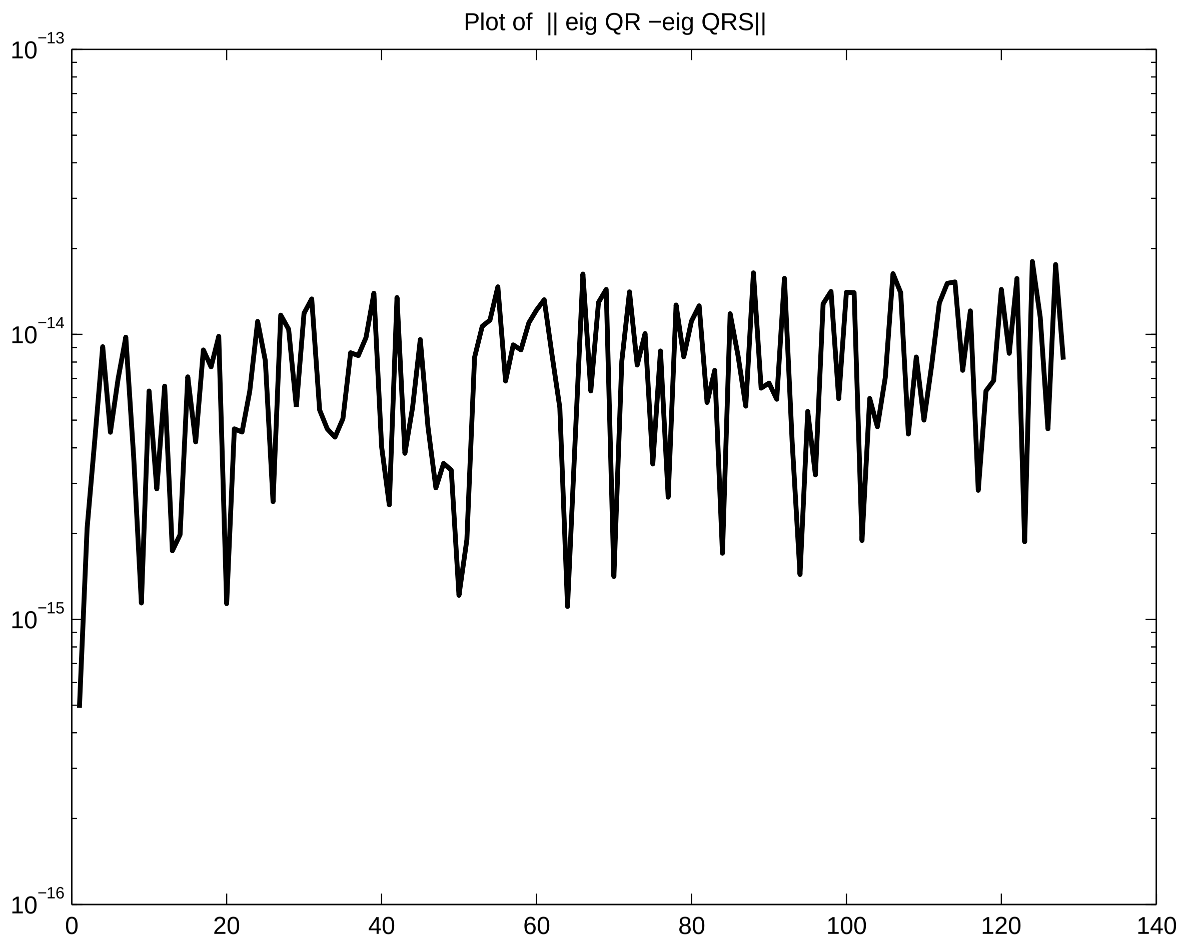
- $\wp(H_{k+1}) \leq 3 \Rightarrow$ redundant representation
- QR factorization computed in linear time [Eidelman, Gohberg]

Computational Analysis

- Computational cost: $\simeq 200n$ flops. Maybe can be slightly reduced
- Linear storage: all the matrices are stored in linear data structures
- No divisions $\Rightarrow$ absolute errors are small as possible. Perturbation results for eigenvalues depend on the magnitude of the absolute perturbation
- Unitary transformations $\Rightarrow$ no significant growth of intermediate results.
- NUMERICALLY STABLE BEHAVIOR

A Numerical Experiment

Companion matrix of size $n = 64$ with random entries



Topics for Future Researches

1. Incorporate scaling and balancing techniques
2. Compare theoretically and experimentally with the fast QR algorithms for unitary Hessenberg matrices designed by [Gragg & coworkers]
3. Investigate the problem of reconstructing a rank-structured matrix given a partial description and some additional requirements [Dym, Gohberg for invertible linear operators]
4. Backward stability analysis for the structured QR algorithm [Tisseur for the standard QR algorithm and Stewart for the fast unitary Hessenberg QR algorithms]
5. Block companion matrices (Matrix Polynomials)

Scaling and Balancing

- Backward stability for companion matrices gives poor bounds for polynomials

1. Balancing can not be used
2. Consider the QZ algorithm applied to the companion matrix pencil $A - \lambda B$ suitably normalized.
   From [Van Dooren & Dewilde] $\rightarrow$ backward stability: The computed roots are exact for $\tilde{p}(z)$ with

$$\| \tilde{\boldsymbol{p}} - \boldsymbol{p} \|_2 \leq k(n) \| \boldsymbol{p} \|_2 \, \epsilon$$

- Experimentally: The rank structure is invariant under the QZ algorithm

Software: The State of Art

- Only MATLAB implementations are available
- Need Fortran and C++ routines to compare timing and accuracy with the LAPACK programs
- Software development is time consuming but I think

  > This should be the time to create a software package concerning fast methods for rank-structured eigenvalue problems

- Collaborations are welcome!!!

References

1. D. A. Bini, L. Gemignani, V. Y. Pan. *QR-like algorithms for generalized semiseparable matrices*. Submitted to Numer. Math, 2003.
2. D. Fasino, L. Gemignani, N. Mastronardi, M. Van Barel. *Orthogonal rational functions and structured matrices.* To appear in SIAM J. Matrix Anal., 2003.
3. D. A. Bini, F. Daddi, L. Gemignani. *The shifted QR algorithm for companion matrices.* To appear in ETNA, 2004.
4. D. A. Bini, L. Gemignani, V. Y. Pan. *Computing polynomial roots in $O(N)$ memory and $O(N^2)$ time by a numerically reliable algorithm based on QR iteration.* To appear in ETNA, 2004.
5. L. Gemignani. *A QR-based method for solving the unitary Hessenberg eigenvalue problem via semiseparable matrices.* Submitted to Comput. Appl. Math.