Rank structured matrices and their role in the design of polynomial rootfinders

D.A. Bini
Joint work with Y. Eidelman, L. Gemignani, I. Gohberg, V. Pan

Dipartimento di Matematica, Università di Pisa
www.dm.unipi.it/~bini

Sophia Antipolis, November 2007

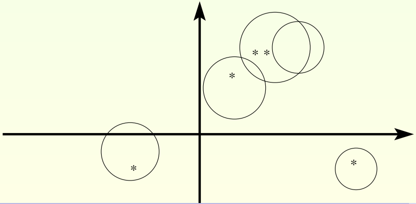
1. Polynomial rootfinders: the available algorithms
   - The problem
   - MPSolve
   - Eigensolve
2. Rank structured matrices and rootfinders
   - Rank structured matrices
   - QR iteration and the rank structure
   - Algorithmic proposals

The Problem

Approximating the roots of a univariate polynomial with relative precision ϵ

For a given $\epsilon > 0$ compute a set of ϵ-inclusion disks:

- the union of the disks contains all the roots
- the relative radius of each disk is bounded by ϵ, i.e., $(\text{radius})/|\text{center}| \le \epsilon$
- Each isolated disk contains only one root each union of k overlapping disks disjoint from the remaining ones contains k roots

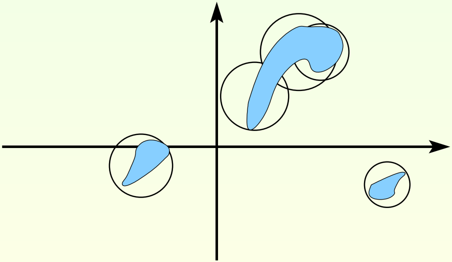


Input/Output

- for approximate input, the union of the disks contains the δ-pseudo roots of $p(x)$

$$\{z \in \mathbb{C} : \quad \widetilde{p}(z) = 0, |\widetilde{p}_i - p_i| \le \delta |p_i|\}$$

Moreover, centers are pseudoroots



Available packages:

- **MPSolve** [Bini, Fiorentino 1999]

  `http://www.dm.unipi.it/cluster-pages/mpsolve`

  Successfully applied to the *partition polynomials* of degree up to 70.000 [R.P Boyer, W.Y. Goh]

  www.math.drexel.edu/~rboyer/talks/MIT_FINAL.pdf

  Included in the library SYNAPS by Bernard Mourrain

Available packages:

- **MPSolve** [Bini, Fiorentino 1999]

  `http://www.dm.unipi.it/cluster-pages/mpsolve`

  Successfully applied to the *partition polynomials* of degree up to 70.000 [R.P Boyer, W.Y. Goh]

  www.math.drexel.edu/~rboyer/talks/MIT_FINAL.pdf

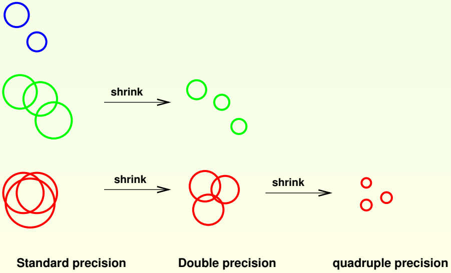
  Included in the library SYNAPS by Bernard Mourrain

- **EigenSolve** [S. Fortune 2001]

  `http://cm.bell-labs.com/who/sjf/eigensolve.html`

Overview of MPSolve

- It generates a sequence of *sets of inclusion disks*
- Disks are shrinked by means of the Ehrlich-Aberth iteration: a fixed-point iteration which acts simultaneously on the approximations to all the roots
- **high-precision arithmetic is used only for shrinking the overlapping disks (adaptivity)**



Overview of MPSolve

- cluster analysis is applied for accelerating convergence
- the number of iterations depends on clusters
- the cost per iteration is $O(n^2)$ ops
- the required memory is $O(n)$
- the polynomial can be assigned as a black box which given z provides $p(z)$

Overview of Eigensolve

1. The monic polynomial can be assigned as a black box which given z provides $p(z)$.

Overview of Eigensolve

1. The monic polynomial can be assigned as a black box which given z provides $p(z)$.
2. Given a set of approximations $z_1, \ldots, z_n$ to the roots of $p(z)$, a generalized companion matrix is constructed

$$C = \begin{bmatrix} z_1 & & & 0 \\ & z_2 & & \\ & & \ddots & \\ 0 & & & z_n \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$$

such that the eigenvalues of C are the roots of $p(z)$

Overview of Eigensolve

1. The monic polynomial can be assigned as a black box which given z provides $p(z)$.
2. Given a set of approximations $z_1, \ldots, z_n$ to the roots of $p(z)$, a generalized companion matrix is constructed

$$C = \begin{bmatrix} z_1 & & & 0 \\ & z_2 & & \\ & & \ddots & \\ 0 & & & z_n \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$$

   such that the eigenvalues of C are the roots of $p(z)$
3. New approximation to the roots are obtained by numerically computing the eigenvalues of C

Overview of Eigensolve

1. The monic polynomial can be assigned as a black box which given z provides $p(z)$.
2. Given a set of approximations $z_1, \ldots, z_n$ to the roots of $p(z)$, a generalized companion matrix is constructed

$$C = \begin{bmatrix} z_1 & & & 0 \\ & z_2 & & \\ & & \ddots & \\ 0 & & & z_n \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}$$

   such that the eigenvalues of C are the roots of $p(z)$
3. New approximation to the roots are obtained by numerically computing the eigenvalues of C
4. if the approximations are satisfactory then stop; otherwise repeat from 2

Overview of Eigensolve

Nice feature: the closer are the approximations to the roots, the better conditioned are the eigenvalues of C

In the limit, the computation of the eigenvalues of C *does not need* high precision.

Higher precision can be needed only to construct C from the polynomial.

Overview of Eigensolve

For polynomials given in terms of the coefficients in the monomial basis, the initial approximations to the roots of $p(z)$ are computed as eigenvalues of the companion (Frobenius) matrix F

$$F = \begin{bmatrix} 0 & \dots & 0 & -a_0 \\ 1 & \ddots & & \vdots \\ & \ddots & 0 & -a_{n-2} \\ 0 & & 1 & -a_{n-1} \end{bmatrix}$$

The QR method of the LAPACK implementation is used for computing eigenvalues of both F and C

Overview of Eigensolve

Complexity analysis

- $O(n^2)$ ops are needed to compute C
- the cost of computing eigenvalues is
  – $O(n^3)$ ops to reduce C to upper Hessenberg form
  – $O(n^2)$ ops per step to apply QR; moreover, $O(n)$ steps are usually required
  – the overall cost is $O(n^3)$ ops
- The memory space is $O(n^2)$

Despite this larger complexity Eigensolve is sometimes faster than MPSolve

Possible improvements

Eigensolve **does not exploit** the specific structure of the generalized companion matrices C and F

There is room to improve Eigensolve by designing new algorithms for computing the eigenvalues of generalized companion matrices

We will prove that exploiting the structures of C and F allows:

- to reduce the complexity from $O(n^2)$ ops per step to $O(n)$ ops per step
- to reduce the memory space from $O(n^2)$ to $O(n)$.

Rank structured matrices

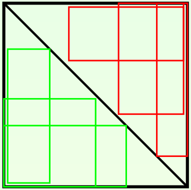
Definition

A matrix A is semiseparable of rank (r,s) if there exist matrices L and U of rank r and s, respectively such that

$$\mathrm{Tril}(A)=\mathrm{Tril}(L), \quad \mathrm{Triu}(A)=\mathrm{Triu}(U),$$

Definition

A matrix A is quasiseparable of rank (r,s) if the submatrices contained in the lower triangular part (upper triangular part) have rank at most r (s, respectively).



any semiseparable matrix is quasiseparable but not conversely

A tridiagonal matrix is quasiseparable of rank $(1,1)$ but it is not semiseparable

$$\begin{bmatrix} * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 \\ 0 & * & * & * & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & 0 & * & * \end{bmatrix}$$

We simply call matrices in these classes *rank structured*

Historical notes

- Many papers starting from the late 1930s
  See [Vandebril, Van Barel, Golub, Mastronardi *Calcolo* vol. 42, 2005] for a detailed commented bibliography

Historical notes

- Many papers starting from the late 1930s
  See [Vandebril, Van Barel, Golub, Mastronardi *Calcolo* vol. 42, 2005] for a detailed commented bibliography
- Partial list: Gantmacher, Krein 37, Asplund 59, Capovani 70,71, Gohberg, Kailath, Koltracht 85, Fiedler 86, Rozsa 86, Bevilacqua, Romani, Eidelman, Van Barel, Vandebril, Delvaux, Tyrtyshnikov, Olshevsky, Mastronardi, Gemignani, Dewilde, van der Veen, Gu, Chandrasekaran, Fasino, Hackbush,.....

Historical notes

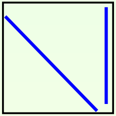
- Many papers starting from the late 1930s
  See [Vandebril, Van Barel, Golub, Mastronardi *Calcolo* vol. 42, 2005] for a detailed commented bibliography
- Partial list: Gantmacher, Krein 37, Asplund 59, Capovani 70,71, Gohberg, Kailath, Koltracht 85, Fiedler 86, Rozsa 86, Bevilacqua, Romani, Eidelman, Van Barel, Vandebril, Delvaux, Tyrtyshnikov, Olshevsky, Mastronardi, Gemignani, Dewilde, van der Veen, Gu, Chandrasekaran, Fasino, Hackbush,.....
- Special issue of the journal Calcolo vol. 42 (2005) dedicated to rank structured matrices

Historical notes

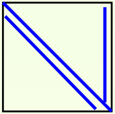
- Many papers starting from the late 1930s
  See [Vandebril, Van Barel, Golub, Mastronardi *Calcolo* vol. 42, 2005] for a detailed commented bibliography
- Partial list: Gantmacher, Krein 37, Asplund 59, Capovani 70,71, Gohberg, Kailath, Koltracht 85, Fiedler 86, Rozsa 86, Bevilacqua, Romani, Eidelman, Van Barel, Vandebril, Delvaux, Tyrtyshnikov, Olshevsky, Mastronardi, Gemignani, Dewilde, van der Veen, Gu, Chandrasekaran, Fasino, Hackbush,.....
- Special issue of the journal Calcolo vol. 42 (2005) dedicated to rank structured matrices
- Many research groups and strong competition (Leuven, Delft, Tel Aviv, Pisa, Bari, Berkley, Santa Barbara, Moskow, ...)

Companion-like matrices are rank structured

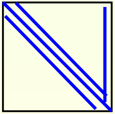
- Diagonal plus rank one: representation in the Lagrange basis
- Frobenius: representation in the power basis



- Modified Frobenius: representation in the Newton basis

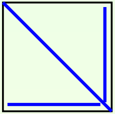


- Comrade: Representation in orthogonal bases [Barnett 75]

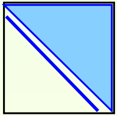


Companion-like matrices are rank structured

- Arrowhead [Golub 73, Fiedler 90]



- Unitary Hessenberg [Gragg 86]



- Fellow [Calvetti, Kim, Reichel 2002] Representation in orthogonal bases on the unit circle
  Unitary Hessenberg+ rank 1

Algorithmic aspects of the rank structure

- The inverse of a rank-structured matrix is rank structured
- Solving systems with rank-structured matrices costs $O(n)$ ops and $O(n)$ memory
- The LU factorization and the QR factorization of a rank structured matrix has rank-structured factors
- Computing these factors costs $O(n)$ ops

Representing rank structured matrices

There are different $O(n)$ representation of a rank structured matrix given in terms of the *generators*, and of *Givens rotations* [Van Barel et al].

A suitable representation given by Eidelman and Gohberg is the *quasiseparable representation* (for simplicity, assume $r = s$)

$$a_{i,j} = \begin{cases} p_i^* E_{i-1} E_{i-2} \cdots E_{j+1} q_j, & \text{for } i > j \\ g_i^* F_{i+1} F_{i+2} \cdots F_{j-1} h_j, & \text{for } i < j \end{cases}$$

where $p_i, q_j, g_i, h_j \in \mathbb{R}^s$, $E_i, F_j \in \mathbb{R}^{s \times s}$

QR factorization and rank structured matrices

QR factorization of a matrix A

$A = QR$, $\quad Q^*Q = I$, $\quad R$ upper triangular

The QR iteration

$$\begin{aligned} &A_1 = A \\ &A_k - \alpha_k I =: Q_k R_k \qquad \text{QR factorization,} \\ &A_{k+1} := R_k Q_k + \alpha_k I \end{aligned}$$

All the matrices A_k are unitarily similar to A

$$A_k = Q_{k-1}^* A_{k-1} Q_{k-1}$$

$$A_k = S_k^* A S_k, \qquad S_k = Q_1 Q_2 \cdots Q_{k-1}, \qquad S_k^* S_k = I$$

The QR iteration

Under mild assumptions A_k converges to (block) triangular form

The diagonal entries of A_k converge to the eigenvalues of A

The computation is backward stable [Tisseur], i.e., the computed eigenvalues are the exact eigenvalues of a sligthly perturbed matrix

The QR iteration is a robust and stable algorithm for computing eigenvalues of a matrix

Question

Can the QR iteration take advantage of the rank structure of A?

In general the structure is not maintained by the QR iteration

QR and rank structure

Theorem

Let $A = U + T$, be such that

- T *has rank* r
- A *is quasiseparable of rank* (p, q)
- U *is unitary, i.e.,* $U^*U = I$, *or Hermitian, i.e.* $A = A^*$.

Then, the matrices A_k *generated by the shifted QR iteration are quasiseparable of rank* $(p, q + 2r)$.

Hermitian: [Bini, Gemignani, Pan, Numer Math 2005], [Vandebril, Van Barel, Mastronardi, 2005], [Eidelman, Gohberg, Olshevsky, LAA 2005]

Unitary: [Bini, Daddi, Gemignani, ETNA 2004], [Bini, Eidelman, Gemignani, Gohberg, TR 2005, SIMAX to appear], [Bini, Eidelman, Gemignani, Gohberg, Math Comp 2007], [Delvaux, Van Barel TR TW164 (2006) KU Leuven], [Gemignani, Calcolo 2005]

Theorem (Bini, Gemignani, Pan, Numer. Math. 2005)

Let $A = D + uv^*$, where D is real diagonal. Then A_k is quasiseparable of rank $(1, 3)$, moreover, given the quasiseparable representation of A_k, the quasiseparable representation of A_{k+1} can be computed with $120n + O(1)$ arithmetic operations and $O(n)$ memory.

Theorem (Bini, Daddi, Gemignani, ETNA 2004, Chandrasekaran, Gu, Xia, Zhu, TR 2006)

Let A be a Frobenius matrix. Then A_k is quasiseparable of rank $(1, 3)$, moreover, given the quasiseparable representation of A_k, the quasiseparable representation of A_{k+1} can be computed with $O(n)$ arithmetic operations and $O(n)$ memory.

Theorem (Bini, Eidelman, Gemignani, Gohberg, SIMAX to appear)

Let $A = H + uv^*$ where A is upper Hesenberg, H is unitary. Then A_k is quasiseparable of rank $(1,3)$ and there exists an algorithm which computes a quasiseparable representation of A_{k+1} given a quasiseparable representation of A_k in $180n + O(1)$ ops.

An implicit QR algorithm for Frobenius matrices is given in Chandrasekaran, Gu, Xia, Zhu, TR 2006.

Outline of the proof technique

The case of Frobenius:

$$\begin{bmatrix} 0 & \dots & 0 & -a_0 \\ 1 & \ddots & & \vdots \\ & \ddots & 0 & -a_{n-2} \\ 0 & & 1 & -a_{n-1} \end{bmatrix} = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 1 & \ddots & & 0 \\ & \ddots & \ddots & \vdots \\ 0 & & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 & -a_0 - 1 \\ \vdots & & \vdots & -a_1 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & -a_{n-1} \end{bmatrix}$$

$$A = U + V$$

U unit circulant (unitary), V rank-one

Since $A_k = S_k^* A S_k$ then

$$A_k = S_k^* U S_k + S_k^* V S_k = U_k + V_k$$

U_k unitary, V_k rank-one

Recall

$$A_k = U_k + V_k$$

therefore

$$A_k^{-1} = U_k^* + \text{Rank-one}$$

Since, A_k is upper Hessenberg, its inverse has rank-one matrices in the lower triangular part

That is, the submatrices in the upper triangular part of U_k have rank at most 2.

The submatrices in the upper triangular part of A_k have rank at most 3.

Algorithms

$O(n)$ complexity algorithms for the QR iteration applied to a rank structured matrix have been designed for

- unitary plus low rank matrices, say, Frobenius
- Hermitian plus low rank, say, diagonal plus rank one

They include

- the linear/quadratic shift
- the (optional) reduction to Hessenberg form
- the implicit QR (partially implemented and still under investigation)
- the deflation of computed roots

Available implementation in matlab or fortran

No multiprecision implementation yet available.

Algorithms

Numerical experiments show

- very good numerical stability
- better performances w.r.t the customary QR if $n \geq n_0$ for $n_0 \approx 100$ in theory, and for $n_0 \in [200, 300]$ in practice, depending on the compiler.

Some results: [Bini, Daddi, Gemignani 2004]

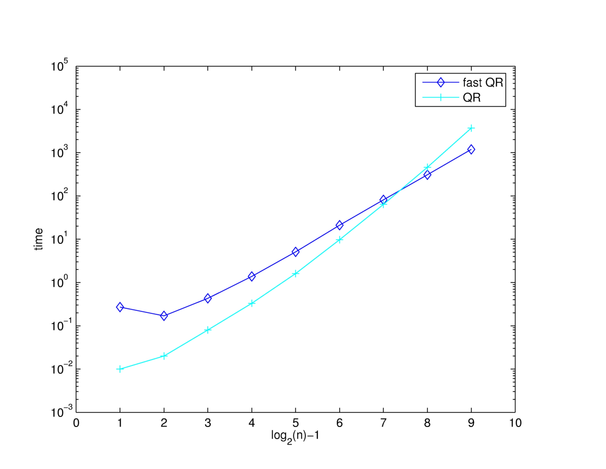
Computing the smallest 40 roots of

$$p(x) = \prod_{i=1}^{40}(x - 2^{-i}) \sum_{i=0}^{n-40} x^i.$$

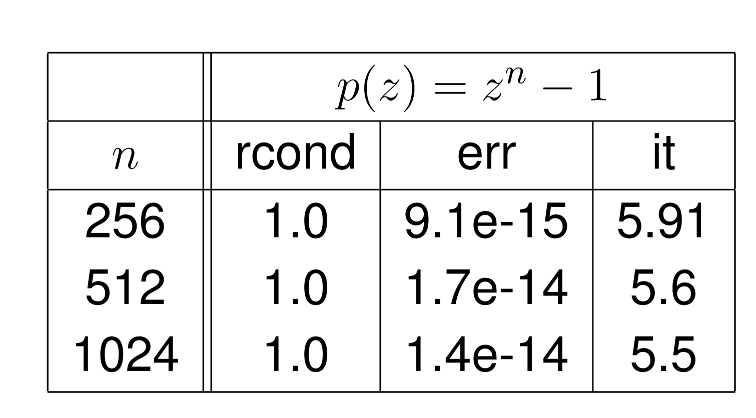
n	cpu	n	cpu
100	0.00	16000	1.56
200	0.01	32000	3.38
400	0.02	64000	6.78
800	0.04	125000	13.02
1600	0.08	250000	28.02
3200	0.16	500000	52.51
6400	0.36	1000000	116.02

Algorithms

Computing all the roots of $p(z) = z^n - 1$, $n = 2^{m+1}$



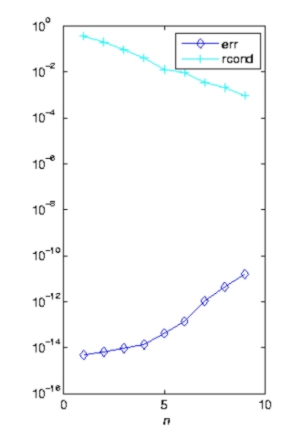
Computing all the roots of $p(z) = z^n - 1$, $n = 2^{m+1}$

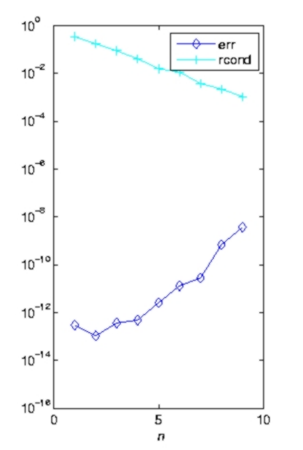


	$p(z) = z^n - 1$		
n	rcond	err	it
256	1.0	9.1e-15	5.91
512	1.0	1.7e-14	5.6
1024	1.0	1.4e-14	5.5

Algorithms

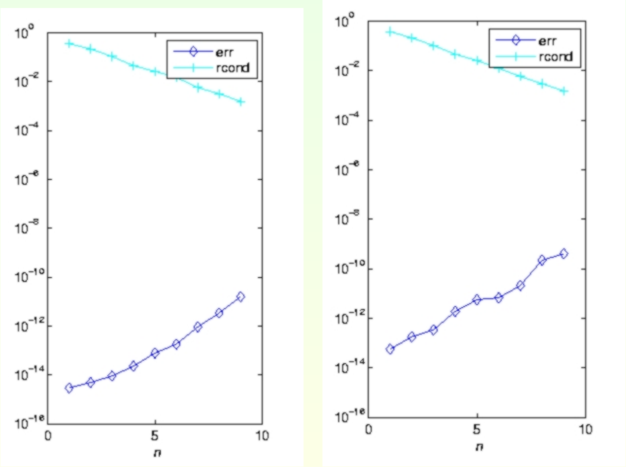
Computing all the roots of a polynomial of degree 2^{m+3}





Algorithms

Computing all the roots of a polynomial of degree 2^{m+3}



Algorithmic proposal

In the package Eigensolve, replace the customary QR algorithm for computing the eigenvalues of the Diagonal plus rank-one matrix with the fast version

Advantage 1: reduction of the complexity from $O(n^2)$ ops per step to $O(n)$ ops per step

Advantage 2: reduction of the memory space from $O(n^2)$ to $O(n)$

Drawback: The fast QR is fast if the diagonal entries are real; if there are h complex roots out of n then A_k is quasiseparable of rank $(h, 3h+1)$ and the cost per step grows to $O(h^2 n)$

The rank structure is not preserved by QR applied to Diagonal plus rank one if the diagonal is complex.

In fact, the structure is preserved for Hermitian plus low rank or Unitary plus low rank.

In the case of complex roots, one may consider a generalized companion matrix of the form

$$A = B + uv^t$$

where B is block diagonal with diagonal blocks of size

1 for real eigenvalues
2 for complex eigenvalues

The 2×2 blocks are of the kind $\begin{bmatrix} a & b \\ -b & a \end{bmatrix}$

The block diagonal matrix B is not Hermitian nor unitary, however, observe that

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix} = \begin{bmatrix} a-b & 0 \\ 0 & a+b \end{bmatrix} + \begin{bmatrix} b & b \\ -b & -b \end{bmatrix}$$

therefore, A can be viewed as a real diagonal plus a rank $h+1$ matrix, where h is the number of complex conjugate pairs

The QR iteration preserves the rank structure; each step costs $O(nh^2)$ operations.

A better approach

The block diagonal matrix B is not generally Hermitian nor unitary. However, there exists a sign matrix $D = \mathrm{diag}(\pm 1)$ such that DB is real symmetric.

DQR iteration [Uhlig, Numer. Math. 1997]

$$A_k - \alpha_k I = DQ_k R_k$$

$$A_{k+1} := R_k D Q_k + \alpha_k I$$

Under mild assumptions the sequence A_k converges to a block upper triangular matrix.

The DQR algorithm can break down due to the use of D-orthogonal factorizations.

Theorem (Gemignani 2007)

The sequence A_k is such that A_k is quasiseparable of rank $(1, 4)$. An algorithm exists for computing the quasiseparable representation of A_{k+1} given the quasiseparable representation of A_k in $O(n)$ ops.

The analysis of this algorithm is still work in place

A different approach

We may construct a new generalized companion matrix A with the following features:

- $A = S + wy^T + uv^T$
- S is real symmetric semiseparable of rank (1,1)

A different approach

We may construct a new generalized companion matrix A with the following features:

- $A = S + wy^T + uv^T$
- S is real symmetric semiseparable of rank (1,1)
- $S + wy^T$ is in block triangular form with blocks of size at most 2×2. The role of this matrix is the same as the role of D in the diagonal plus rank-one representation of C
  - $S + wy^T$ has given eigenvalues
  - u and v are computed such that $\det(\lambda I - A) = p(\lambda)$ at the cost $O(n^2)$

$$S = \left[\begin{array}{cc|cc|cc|cc}
a_1 & b_1 & 0 & 2b_1 & 0 & 2b_1 & 0 & 2b_1 \\
b_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & a_2 & b_2 & 0 & 2b_2 & 0 & 2b_2 \\
2b_1 & 0 & b_2 & a_2 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & a_3 & b_3 & 0 & 2b_3 \\
2b_1 & 0 & 2b_2 & 0 & b_3 & a_3 & 2b_4 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & a_4 & b_4 \\
2b_1 & 0 & 2b_2 & 0 & 2b_3 & 0 & b_4 & a_4
\end{array}\right]$$

$$S = \left[\begin{array}{cc|cc|cc|cc}
a_1 & b_1 & 0 & 2b_1 & 0 & 2b_1 & 0 & 2b_1 \\
b_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & a_2 & b_2 & 0 & 2b_2 & 0 & 2b_2 \\
2b_1 & 0 & b_2 & a_2 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & a_3 & b_3 & 0 & 2b_3 \\
2b_1 & 0 & 2b_2 & 0 & b_3 & a_3 & 2b_4 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & a_4 & b_4 \\
2b_1 & 0 & 2b_2 & 0 & 2b_3 & 0 & b_4 & a_4
\end{array}\right]$$

$$wy^T = - \begin{bmatrix} 0 \\ 2 \\ \vdots \\ 0 \\ 2 \end{bmatrix} \begin{bmatrix} b_1 & 0 & b_2 & 0 & b_3 & 0 & b_4 & 0 \end{bmatrix}$$

A different approach

$$S + wy^T = \left[\begin{array}{cc|cc|cc|cc}
a_1 & b_1 & 0 & 2b_1 & 0 & 2b_1 & 0 & 2b_1 \\
-b_1 & a_1 & -2b_2 & 0 & -2b_3 & 0 & -2b_4 & 0 \\
\hline
0 & 0 & a_2 & b_2 & 0 & 2b_2 & 0 & 2b_2 \\
0 & 0 & -b_2 & a_2 & -2b_3 & 0 & -2b_4 & 0 \\
\hline
0 & 0 & 0 & 0 & a_3 & b_3 & 0 & 2b_3 \\
0 & 0 & 0 & 0 & -b_3 & a_3 & -2b_4 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & a_4 & b_4 \\
0 & 0 & 0 & 0 & 0 & 0 & -b_4 & a_4
\end{array}\right]$$

A different approach

The QR iteration generates quasiseparable matrices of rank $(4, 8)$ at the cost $O(n)$ per step

The Fortune algorithm can be fully implemented in a fast way

Experimentally, the eigenvalues of the matrix $S + wy^T$ are not well conditioned with respect to general perturbations. On the other hand they are very well conditioned for structured perturbations. Is this enough to make this approach effective?

The algorithm is still work in place

Future work

- implementation of the Fortune algorithm with fast QR for polynomials with real roots;
- implementation of the Fortune algorithm with fast DQR for polynomials with complex roots;
- analysis of the numerical stability of DQR;
- analysis and implementation of the method based on the block triangular companion;