Fast Eigenvalue Computation of Symmetric Rationally Generated Toeplitz Matrices

Luca Gemignani

Dipartimento di Matematica, Università di Pisa, Italy

gemignan@dm.unipi.it

This is a joint work with K. Frederix & M. Van Barel

Cortona 2008

A Classical Example

- The Kac-Murdock-Szegö Toeplitz matrix

$$T_n = (0.5^{|i-j|})_{i,j=1}^n = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{4} & \cdots \\ \frac{1}{2} & \ddots & \ddots & \ddots \\ \frac{1}{4} & \ddots & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$$

$$t(z) = \sum_{j=-\infty}^{\infty} (\frac{1}{2})^{|j|} z_j = \frac{0.5z}{1-0.5z} + \frac{1}{1-0.5z^{-1}} = \frac{0.75}{(1-0.5z)(1-0.5z^{-1})}$$

- We aim to compute the eigenvalues of T_n efficiently and accurately exploiting the relationships between T_n and its symbol $t(z)$

UNIVERSITÀ DI PISA

Previous Literature

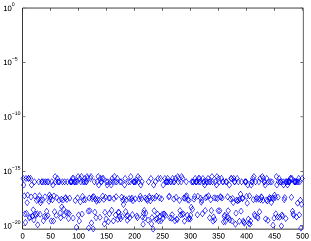
1. Functional iteration methods based on the fast evaluation of the characteristic polynomial and/or associated rational functions [Trench; Bini & Di Benedetto]
   1.1 suited for computing a few eigenvalues
   1.2 accuracy and computational issues
2. Matrix methods based on matrix algebra embeddings and eigenvalue computation of matrices modified by a rank-one correction [Handy & Barlow; Di Benedetto]
   2.1 eigenvector computation can be ill-conditioned depending on the separation of the eigenvalues

$$\min_{i \neq j} |\lambda_i(T_{500}) - \lambda_j(T_{500})| \simeq 10^{-6}$$

Università di Pisa

Our Proposal

- Matrix methods based on the exploitation of the rank structure of T_n



Plot of the 2-nd singular value of the off-diagonal submatrices of T_{500}

- We study efficient methods for the tridiagonalization of T_n by orthogonal similarity

UNIVERSITÀ DI PISA

Condensed Representations

- The rank structure of T_n is induced by condensed representations involving band matrices
  1. If $t(z) = \frac{p(z^{-1})}{a(z^{-1})} + \frac{p(z)}{a(z)}$ then [Dickinson]
  $$T_n = T_a^{-1} \cdot T_p + T_p^T \cdot T_a^T$$
  2. More generally, if $t(z) = \frac{c(z)}{a(z)a(1/z)}$, with $\deg(a(z)) = q$ and $\deg(c(z)) = l$, then
  $$T_n = T_n(s) + T_a^{-1} \cdot T_p + T_p^T \cdot T_a^T,$$
  $$s(z) = \sum_{i=q-l}^{l-q} s_{|i|} z^i, \quad p(z) = p_0 + \ldots + p_q z^q,$$
  $$\mathcal{J}\mathbf{p} = \boldsymbol{\beta}, \quad \mathcal{J} = \begin{bmatrix} a_0 & \ldots & \ldots & a_q \\ & \ddots & & \vdots \\ & & \ddots & \vdots \\ & & & a_0 \end{bmatrix} + \begin{bmatrix} a_0 & \ldots & \ldots & a_q \\ \vdots & & \iddots & \\ \vdots & \iddots & & \\ a_q & & & \end{bmatrix}$$

Università di Pisa

Remarks on the Jury System

- $\mathcal{J}$ is invertible since the zeros of $a(z)$ have modulus greater than 1 [Demeure]
  1. the conditioning of $\mathcal{J}$ depends on the closeness of the zeros to the unit circle
- $\mathcal{J}$ is Toeplitz-plus-Hankel. The representation can be computed in $O(l^2 + q^2)$ flops by using
  1. fast direct methods based on displacement rank techniques
  2. fast iterative methods based on spectral factorization techniques [Demeure; Bacciardi, Gemignani]

UNIVERSITÀ DI PISA

Quasiseparable Representations

- Assume for simplicity $l < q$ and $n = m \cdot q$

1. $\max_{1 \le k \le n-1} \operatorname{rank} T_n(k+1\colon n, 1\colon k) \le q$
2. T_n can be partitioned in a block form as

$$T_n = (T_{i,j}^{(n)})_{i,j=1}^{m}, \quad T_{i,j}^{(n)} \in \mathbb{R}^{q \times q},$$
$$T_{i,j}^{(n)} = A \cdot F_a^{q(i-j-1)} \cdot B, i \ge j, \quad F_a = \mathrm{compan}(z^q a(z^{-1}))$$

3. the matrix $P_n = B_n \cdot T_n \cdot B_n^T$ is a symmetric block tridiagonal matrix, where

$$B_n = \begin{bmatrix} I_q & & & \\ -\Sigma & \ddots & & \\ & \ddots & \ddots & \\ & & -\Sigma & I_q \end{bmatrix}, \quad \Sigma = A^{-1} F_a^q A$$

The Tridiagonal Reduction Algorithm

1. $U \cdot \begin{bmatrix} I_q \\ \Sigma \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}$, $\mathcal{G} = I_{(m-2)q} \oplus U$
2. the multiplication $\mathcal{G} \cdot B_n^{-1}$ creates a bulge

$$\mathcal{G} \cdot B_n^{-1} = \left[\begin{array}{ccc|c} & \star & & 0 \\ \hline \Sigma^{m-2}R & \dots & \Sigma R & \\ 0 & \dots & 0 & I_{2q} \end{array}\right] \cdot Z, \quad Z = (I_{(m-2)q} \oplus \begin{bmatrix} R & U_{1,2} \\ 0 & U_{2,2} \end{bmatrix}$$

3. the multiplication $Z \cdot P_n \cdot Z^T$ creates a bulge in the block tridiagonal structure of P_n
4. this bulge can be chased away by a block Givens transformation which commutes with the first factor of $\mathcal{G} \cdot B_n^{-1}$

- Overall complexity $O(m^2 q^3) = O(n^2 q)$ flops

Università di Pisa

Givens-weight Representations

1. Givens part: An orthogonal matrix Q such that

$$Q^T \cdot T_n = R$$

   where R is lower banded with q subdiagonals. Since

$$Q^T T_n = (T_a Q)^{-1} T_p + (T_p Q)^T T_a^{-T}$$

   Q is the product of (block) Givens transformations determined to convert T_a into upper triangular form
2. Weight part: Elements generated in the factorization around the main diagonal needed to reconstruct the lower part of T_n from $T_n = Q \cdot R$

The Tridiagonal Reduction Algorithm

1. Annihilate the Givens part by multiplying R on the right and on the left by the factors of Q.
2. At intermediate steps the process generates bulges into the band profile of R which can be chased away by standard techniques.
3. As result, at the very end T_n is transformed by orthogonal similarity to banded form with bandwidth q

- Overall complexity $O(m^2q^3) = O(n^2q)$ flops

Numerical Experiments

- We have compared the Matlab implementations of our algorithms
  1. *alg_1* uses the block quasiseparable representation to tridiagonalize T_n
  2. *alg_2* uses the Givens-weight representation to tridiagonalize T_n
- For comparison, T_n is first determined by evaluation-interpolation schemes and then its eigenvalues are computed by the *eig* function

UNIVERSITÀ DI PISA

Numerical Tests

- Example 1

$$T_n = (0.5^{|i-j|})_{i,j=1}^{n}, . \quad t(z) = \frac{0.75}{(1-0.5z)(1-0.5z^{-1})}$$

- Example 2

$$t(z) = \frac{z^{-2} - 3.5z^{-1} + 1.5 - 3.5z + z^2}{a(z)a(z^{-1})}, \quad a(z) = (1-0.1z)(1-0.2z)$$

- Example 3

$$t(z) = \frac{z^{-3} - z^{-2} + 2z^{-1} + 1 + 2z - z^2 + z^3}{a(z)a(z^{-1})}, \quad a(z) = 1-0.4z-0.47z^2+0.21z^3$$

UNIVERSITÀ DI PISA

Numerical Results by *alg_1*

n	Example 1	Example 2	Example 3
10	1.0×10^{-15}	6.4×10^{-16}	1.6×10^{-15}
50	2.0×10^{-15}	1.2×10^{-15}	3.2×10^{-15}
100	4.1×10^{-15}	1.7×10^{-15}	3.3×10^{-15}
500	1.4×10^{-14}	3.5×10^{-15}	1.0×10^{-14}
1000	2.3×10^{-14}	5.6×10^{-15}	1.6×10^{-14}

Table: Numerical results generated by *alg_1* for example 1, 2, 3.

UNIVERSITÀ DI PISA

Numerical Results by *alg_2*

n	Example 1	Example 2	Example 3
10	5.2×10^{-16}	6.6×10^{-16}	1.3×10^{-15}
50	1.1×10^{-15}	1.3×10^{-15}	2.6×10^{-15}
100	1.4×10^{-15}	1.2×10^{-15}	4.1×10^{-15}
500	1.7×10^{-15}	4.1×10^{-15}	8.2×10^{-15}
1000	1.6×10^{-15}	4.0×10^{-15}	1.8×10^{-15}

Table: Numerical results generated by *alg_2* for example 1, 2, 3.

UNIVERSITÀ DI PISA

Some Harder Tests

- Try with larger values of q
- Try for different distribution of the zeros of $a(z)$. This affects the conditioning of the Jury matrix

1. Case 1: $q = 20$ and the zeros of $a(z)$ are approximately uniformly distributed around the unit circle;
2. Case 2: $q = 20$ and some zeros are clustered but there are still zeros at both the sides of the unit circle;
3. Case 3: $q = 20$ and all the zeros are located at one side of the unit circle.

Numerical Results by *alg_1*

n	Case 1	Case 2	Case 3
100	1.3×10^{-15}	5.7×10^{-13}	8.0×10^{-4}
500	4.8×10^{-15}	5.6×10^{-13}	1.3×10^{-3}
1000	5.3×10^{-15}	5.6×10^{-13}	1.4×10^{-3}
$\kappa(\mathcal{J})$	7.5×10^{0}	1.6×10^{4}	1.6×10^{11}

Table: Numerical results generated by *alg_1* for Example $q = 20$ in the three different cases.

Numerical Results by *alg_2*

n	Case 1	Case 2	Case 3
100	1.6×10^{-15}	1.1×10^{-13}	2.0×10^{-4}
500	3.0×10^{-15}	1.3×10^{-13}	4.9×10^{-4}
1000	7.5×10^{-15}	1.7×10^{-13}	6.3×10^{-4}
$\kappa(\mathcal{J})$	7.5×10^{0}	1.6×10^{4}	1.6×10^{11}

Table: Numerical results generated by *alg_2* for Example $q = 20$ in the three different cases.

UNIVERSITÀ DI PISA

Conclusions and Future Work

- The eigenvalue algorithms are fast and as accurate as the computation of the rank structure from the Toeplitz symbol
- The accuracy is comparable for both representations

- Timing comparisons for practical efficiency
- Extensions to generalized eigenvalue problem and block Toeplitz matrices