Efficient Numerical Methods for the Polynomial Spectral Factorization

Luca Gemignani

Dipartimento di Matematica, Università di Pisa, Italy
gemignan@dm.unipi.it

SC2011
Some (Personal) Motivations

- Polynomial spectral factorization problems are wonderful examples of the power of interplay between polynomial and structured matrix computations.
- I first learned about this problem fifteen years ago from the paper “Spectral factorization of Laurent polynomials” by Goodman, Micchelli, Rodriguez, Seatzu. That paper made clear to me the potential synergies with the “Sardinia group” of numerical linear algebra.
- The numerical solution of these problems is a still–active research field providing the source of many interesting questions and advances.
Let $a(z)$ be a symmetric Laurent polynomial of degree $n$, that is,

$$a(z) = \sum_{i=-n}^{n} a_i z^i, \quad a_i \in \mathbb{R}, \quad a_n \neq 0.$$ 

Observe that (spectral symmetry)

$$a(\xi) = 0, \quad \xi \in \mathbb{C} \setminus \{0\}, \Rightarrow a(\xi^{-1}) = 0$$

- The customary spectral factorization problem is

$$a(z) = b(z) \cdot b(z^{-1}), \quad b(z) = \sum_{i=0}^{n} b_i z^i, \quad b_0 > 0, \quad b(z) \neq 0 \text{ if } |z| < 1$$

- A different spectral-type factorization is

$$a(z) = \prod_{i=1}^{n} (z^{-1} + \gamma_i + z)$$
Plan of the Talk

1. To survey some effective numerical methods for solving the polynomial factorization problems
   - a PCG-based implementation of the Wilson’s method for spectral factorization
   - a structured QR iteration for the splitting into quadratic factors

2. To present a generalization of the splitting technique for matrix polynomials

3. To discuss current researches concerning
   - computing the spectral factorization of matrix polynomials;
   - computing approximate spectral factors of bivariate polynomials.
Polynomial Spectral Factorization

- \( a(z) = b(z) \cdot b(z^{-1}) \) is a system of nonlinear equations. Then we can apply the Newton-Raphson method \( \Rightarrow \) Wilson’s method.

- The Jacobian matrix is

\[
\mathcal{J}^T = \begin{bmatrix}
b_0 & \cdots & b_n \\
\vdots & \ddots & \vdots \\
b_n & \cdots & b_0
\end{bmatrix} + \begin{bmatrix}
b_0 & \cdots & \cdots & b_n \\
\vdots & \ddots & \vdots & \vdots \\
\vdots & \ddots & \ddots & \\
b_n & \cdots & \cdots & b_0
\end{bmatrix}
\]

- Each iteration reduces to

\[
b_k(z) \cdot b_{k+1}(z^{-1}) + b_k(z^{-1}) \cdot b_{k+1}(z) = a(z) + b_k(z) \cdot b_k(z^{-1})
\]
How to perform this efficiently?

- Suppose that $a(z) > 0$ for $|z| = 1$
- By rewriting the equation as
  $$\frac{b_{k+1}(z^{-1})}{b_k(z^{-1})} + \frac{b_{k+1}(z)}{b_k(z)} = 1 + \frac{a(z)}{b_k(z) \cdot b_k(z^{-1})}$$

  we reduce to computing the central coefficients of the Laurent series of the function $\frac{a(z)}{b_k(z) \cdot b_k(z^{-1})}$

- The method proposed in “Proceeding of SPIE, 2002” by Bacciardi and G. uses the PCG method to compute the first column of the inverse of a positive definite Schur-Cohn matrix defined by
  $$\mathcal{P}_k = b_k(Z_{2n+1})^T \cdot b_k(Z_{2n+1}) - \hat{b}_k(Z_{2n+1}) \cdot \hat{b}_k(Z_{2n+1})^T$$

- The method proposed in “Geophysical Prospecting, 2003” by Fomel, Sava, Rickett and Claerbout uses evaluation/interpolation techniques.
Some Numerical Results

\[ a(z) = b(z) \cdot b(z^{-1}) \quad b(z) = \sum_{i=0}^{n} (i + 1)z^i, \ m = 2n \]

<table>
<thead>
<tr>
<th>m</th>
<th>steps</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>10</td>
<td>0.05</td>
</tr>
<tr>
<td>128</td>
<td>12</td>
<td>0.13</td>
</tr>
<tr>
<td>256</td>
<td>15</td>
<td>0.31</td>
</tr>
<tr>
<td>512</td>
<td>15</td>
<td>0.77</td>
</tr>
<tr>
<td>1024</td>
<td>16</td>
<td>2.05</td>
</tr>
<tr>
<td>2048</td>
<td>17</td>
<td>7.16</td>
</tr>
<tr>
<td>4096</td>
<td>18</td>
<td>24.15</td>
</tr>
<tr>
<td>8192</td>
<td>19</td>
<td>109.85</td>
</tr>
<tr>
<td>16384</td>
<td>20</td>
<td>326.49</td>
</tr>
</tbody>
</table>
Some numerical Results

- \( a(z) = b(z) \cdot b(z^{-1}) \)
- \( b(z) = \sum_{i=0}^{n} (i + 1)z^i, \quad m = 2n \)

<table>
<thead>
<tr>
<th>m</th>
<th>PCG</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>5</td>
<td>74</td>
</tr>
<tr>
<td>128</td>
<td>5</td>
<td>138</td>
</tr>
<tr>
<td>256</td>
<td>5</td>
<td>253</td>
</tr>
<tr>
<td>512</td>
<td>5</td>
<td>486</td>
</tr>
<tr>
<td>1024</td>
<td>9</td>
<td>941</td>
</tr>
<tr>
<td>2048</td>
<td>9</td>
<td>1856</td>
</tr>
</tbody>
</table>

- \( a(z) = z^{-m} + \left( \frac{m}{m+1} + \frac{m+1}{m} \right) + z^m \)

- \( m = 2046, \quad \text{time}(\text{Wilson}) = 1.7 \quad \text{time}(\text{FFT}) = 20.133 \)
Factoring into Quadratic Factors

For solving $a(z) = \prod_{i=1}^{n}(z^{-1} + \gamma_i + z)$ we take a different route. Observe that

$$a(z) = \gamma(y) = \sum_{i=0}^{n} a_i \phi_i(y), \quad y = y(z) = z + z^{-1}$$

$\phi_0(y) = 2; \phi_1(y) = y; \; \phi_j(y) = \phi_{j+1}(y) + \phi_{j-1}(y), \; j \geq 1$

A companion pencil associated with $\gamma(y)$ has the form $L(y) = yE + F$ where

$$
\begin{pmatrix}
1 \\
\vdots \\
1 \\
a_n
\end{pmatrix}
\begin{pmatrix}
y \\
-\sqrt{2} \\
\vdots \\
a_0 \sqrt{2}
\end{pmatrix} + 
\begin{pmatrix}
0 & -\sqrt{2} & \ddots & \ddots & \ddots \\
-\sqrt{2} & 0 & -1 & \ddots & \ddots \\
\ddots & \ddots & \ddots & \ddots & \ddots \\
\ddots & \ddots & \ddots & -1 & 0 & -1 \\
a_0 \sqrt{2} & \ldots & a_{n-3} & a_{n-2} - a_n & a_{n-1}
\end{pmatrix}
$$
1. The QR method applied to \((E^{-1} \cdot F)^T\) produces a sequence of matrices with the property of having low rank submatrices in the upper triangular portion.

2. The recipe for making computationally usable the algebraic property of the rank is to define a condensed representation for the matrices employing \(O(n)\) parameters.

3. The QR algorithm in “Numer. Algorithms, 2008” by Eidelman, G., Gohberg is backward stable and computationally efficient (\(O(n)\) flops and storage per step).
\[ a(z) = z^{-m} + \left( \frac{m}{m+1} + \frac{m+1}{m} \right) + z^m \]
Solving the T-Palindromic EigenProblem

- Let us consider a T-palindromic eigenproblem

\[ P(\lambda)x = 0, \quad P(\lambda) = \lambda^{-k} \sum_{j=0}^{2k} P_j \lambda^j, \quad P_j^T = P_{2k-j} \]

- The symmetry of the coefficients implies the symmetry of the spectrum

\[ \lambda_i \in \text{spectum}(P(\lambda)), \quad \lambda_i \neq 0 \Rightarrow \frac{1}{\lambda_i} \in \text{spectum}(P(\lambda)) \]

- We want to compute the generalized eigenvalues in pairs \((\lambda, 1/\lambda)\)
We have

\[ P(\lambda) = P_0 + \sum_{j=1}^{k} \left[ \frac{P_j + P_j^T}{2}(\lambda^j + \lambda^{-j}) + \frac{P_j - P_j^T}{2}(\lambda^j - \lambda^{-j}) \right] \]

\[ y = y(\lambda) = \lambda + \lambda^{-1}, \quad w = w(\lambda) = \lambda - \lambda^{-1}, \quad w^2 = y^2 - 4 \]

\[ P(\lambda) = B(y) + w \cdot C(y) = Q(y, w) \]

\[ B(y) = B(y)^T, \quad C(y) = -C(y)^T \rightarrow Q(y, w)^T = B(y) - w \cdot C(y) \]

\[ M(y) = \begin{pmatrix} B(y) & w^2 C(y) \\ C(y) & B(y) \end{pmatrix} \]

\[ \det(M(y)) = [\det(Q(y, w))]^2 = [\det(P(\lambda))]^2 = q(y)^2 \]
In a joint paper with V. Noferini we propose to compute the roots of \( q(y) = 0 \) by means the Erlich-Aberth method.

It is worth noticing that \( q(y) \) is unknown but \( M(y) \) is not and

\[
\frac{\det(M(y))}{(\det(M(y)))'} = \frac{1}{2} \frac{q(y)}{q'(y)}
\]

The rank-structure of the linearization of \( M(y) \) is used to compute efficiently the ratio

\[
\frac{q'(y)}{q(y)} = \text{trace}(L^{-1}(y)E)
\]
Numerical Results

- **MSS1.** Consider the mass-spring example in the nonoverdamped case. This is

\[ P(\lambda) = C(\eta(\lambda)) = M\lambda^{-2} + C\lambda^{-1} + K + 2M + C\lambda + M\lambda^2. \]

where \( M = I_n \), \( C = \tau \text{tridiag}(-1, 3, -1) \), 
\( K = \kappa \text{tridiag}(-1, 3, -1) \) and \( \kappa = 5 \), \( \tau = 3 \), \( n = 50 \).
Can we compute a matrix spectral factorization by using the generalized eigenvalues $\xi_j$? We elaborate on a method proposed in "IEEE Trans. Autom. Control, 1985" by F. Callier.

$$\hat{P}(\lambda) \leftarrow (T(\lambda))^{-1} \cdot P(\lambda) \cdot (T(1/\lambda))^{-T}, \quad \text{det } T(\lambda) = \lambda - \xi_j.$$ 

Under some assumptions we can first reduce $P(\lambda)$ to a unimodular $\tilde{P}(\lambda)$ and then to a constant matrix $K = U^T \cdot U$. In this way

$$W(\lambda) = U \cdot T_j(\lambda) \cdots T_1(\lambda)$$

is the desired spectral factor.
The root-finding approach encounters some numerical difficulties near $\lambda = \pm 1$ due to the ill-conditioning of computing the roots from the coefficients of the quadratic factor.

Symmetric division for palindromic polynomials

$$p(z) = q(z)(1 - z\xi)(1 - \frac{z}{\xi}) + r(z^{n-1} + z)$$

Then we can refine by applying the Newton method to

$$r(\xi) = 0$$
The Bivariate Case

1. In the paper in “Geophysical Prospecting, 2003” the authors discuss how to compute an approximate triangular decomposition of the bidimensional Laplacian matrix using the polynomial spectral factorization approach. The idea is simply to consider the Laurent polynomial associated with

\[\ldots, 0, -1, 0, \ldots, -1, 4, -1, 0, \ldots, -1, 0, \ldots\]\n
2. Differently saying, the product of bivariate polynomials can be computed by using the univariate multiplication algorithm

\[a(x, y) = (3 + x + y) \cdot (3 + x^{-1} + y^{-1})\]
\[a(x, x^3) = q(x) \cdot q(x^{-1})\]
\[q(1) = 3.0000e+00, q(2) = 1.0000e+00\]
\[q(3) = -2.4980e-16, q(4) = 1.0000e+00\]