Structured matrix methods for the computation of multiple roots of univariate polynomials

John D. Allan and Joab R. Winkler

Department of Computer Science

The University of Sheffield, United Kingdom

Overview

- Introduction
- Structured and unstructured condition numbers
- A root finder that computes root multiplicities
- The Sylvester resultant matrix
- The method of structured total least norm
- The principle of minimum description length
- A polynomial root solver
- Conclusions

1. Introduction

Classical methods yield unsatisfactory results for the computation of multiple roots of a polynomial.

Example Consider the fourth order polynomial

$$x^4 - 4x^3 + 6x^2 - 4x + 1 = (x-1)^4$$

whose root is $x = 1$ with multiplicity 4. The `roots` function in MATLAB returns

$$1.0002, \quad 1.0000 \pm 0.0002i, \quad 0.9998$$

which shows that roundoff errors are sufficient to cause a relative error in the solution of 2×10^{-4}. □

Example The roots of the polynomial $(x-1)^{100}$ were computed by the `roots` function in MATLAB.

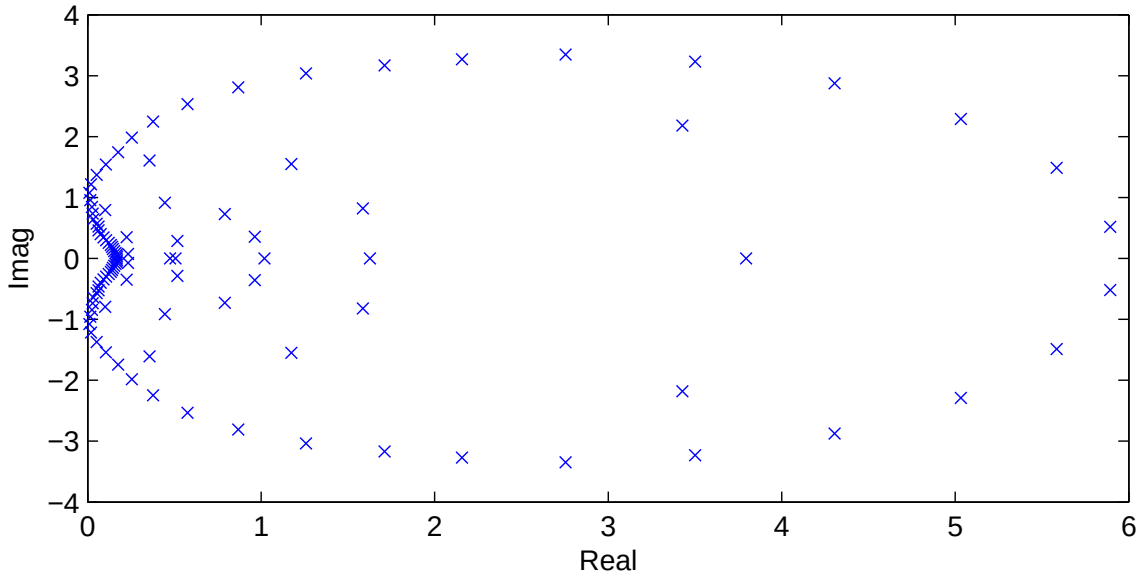

Figure 1: The computed roots of $(x-1)^{100}$.

Example Consider the computed roots of $(x-1)^n, n = 1, 6, 11, \ldots,$ when the constant term is perturbed by 2^{-10}.

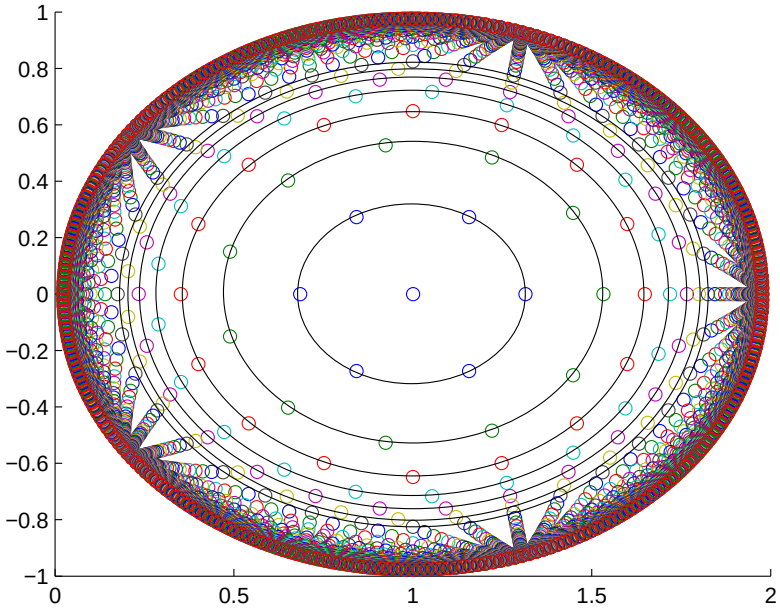

Figure 2: The computed roots of $(x-1)^n, n = 1, 6, 11, \ldots,$ when the constant term is perturbed.

2. Structured and unstructured condition numbers

The componentwise and normwise condition numbers of a root x_0 of the polynomial

$$f(x) = \sum_{i=0}^{m} a_i \phi_i(x)$$

are

$$\kappa_c(x_0) = \frac{1}{\varepsilon_c^{1-\frac{1}{m}}} \frac{1}{|x_0|} \left(\frac{m!}{\left| f^{(m)}(x_0) \right|} \sum_{i=0}^{n} \left| a_i \phi_i(x_0) \right| \right)^{\frac{1}{m}}$$

and

$$\kappa_n(x_0) = \frac{1}{\varepsilon_n^{1-\frac{1}{m}}} \frac{1}{|x_0|} \left(\frac{m!}{\left| f^{(m)}(x_0) \right|} \|a\|_2 \|\phi(x_0)\|_2 \right)^{\frac{1}{m}}$$

Compare with the structured condition number of $(x - x_0)^n$ that preserves the multiplicity of the root, but not the value of the root

$$\rho(x_0) = \frac{\Delta x_0}{\Delta f} = \frac{1}{n\,|x_0|} \frac{\|(x - x_0)^n\|_2}{\|(x - x_0)^{n-1}\|_2} = \frac{1}{n\,|x_0|} \left(\frac{\sum_{i=0}^{n} \binom{n}{i}^2 (x_0)^{2i}}{\sum_{i=0}^{n-1} \binom{n-1}{i}^2 (x_0)^{2i}} \right)^{\frac{1}{2}}$$

where

$$\Delta f = \frac{\|\delta f\|_2}{\|f\|_2} \qquad \text{and} \qquad \Delta x_0 = \frac{|\delta x_0|}{|x_0|}$$

Example The condition number $\rho(1)$ of the root $x_0 = 1$ of the polynomial $(x-1)^n$ is

$$\rho(1) = \frac{1}{n}\left(\frac{\sum_{i=0}^{n}\binom{n}{i}^2}{\sum_{i=0}^{n-1}\binom{n-1}{i}^2}\right)^{\frac{1}{2}} = \frac{1}{n}\sqrt{\frac{\binom{2n}{n}}{\binom{2(n-1)}{n-1}}} = \frac{1}{n}\sqrt{\frac{2(2n-1)}{n}} \approx \frac{2}{n}$$

if n is large.

Compare with the componentwise and normwise condition numbers for random (unstructured) perturbations

$$\kappa_c(1) \approx \frac{|\delta x_0|}{\varepsilon_c} \qquad \text{and} \qquad \kappa_n(1) \approx \frac{|\delta x_0|}{\varepsilon_n}$$

□

Conclusion

- A multiple root of a polynomial is ill-conditioned, and the ill-conditioning increases as its multiplicity increases
- If the multiplicity is sufficiently large, its condition numbers are proportional to the signal-to-noise ratio
- A multiple root of a polynomial is well-conditioned with respect to perturbations that preserve its multiplicity

3. A root finder that computes root multiplicities

If there is prior evidence that the theoretically exact form of the polynomial has multiple roots, then it is desirable to compute these multiple roots

- Given an inexact polynomial, what is the nearest polynomial that has multiple roots?
- How can multiple roots be computed because roundoff error is sufficient to cause them to break up into simple roots?

Example Let $w_i(x), i = 1, \ldots, m_{max}$, be the product of all factors of degree i of $f(x)$

$$f(x) = w_1(x) w_2^2(x) w_3^3(x) \cdots w_{m_{\max}}^{m_{\max}}(x)$$

and thus

$$r_0(x) = \text{GCD}\left(f(x), f^{(1)}(x)\right) = w_2(x) w_3^2(x) w_4^3(x) \cdots w_{m_{\max}}^{m_{\max}-1}(x)$$

Similarly

$$\begin{aligned}
r_1(x) &= \text{GCD}\left(r_0(x), r_0^{(1)}(x)\right) &&= w_3(x) w_4^2(x) w_5^3(x) \cdots w_{m_{\max}}^{m_{\max}-2}(x) \\
r_2(x) &= \text{GCD}\left(r_1(x), r_1^{(1)}(x)\right) &&= w_4(x) w_5^2(x) w_6^3(x) \cdots w_{m_{\max}}^{m_{\max}-3}(x) \\
r_3(x) &= \text{GCD}\left(r_2(x), r_2^{(1)}(x)\right) &&= w_5(x) w_6^2(x) \cdots w_{m_{\max}}^{m_{\max}-4}(x) \\
&\vdots
\end{aligned}$$

and the sequence terminates at $r_{m_{\max}-1}(x)$, which is a constant.

A sequence of polynomials $h_i(x), i = 1, \ldots, m_{\max}$, is defined such that

$$\begin{aligned}
h_1(x) &= \frac{f(x)}{r_0(x)} &&= w_1(x)w_2(x)w_3(x)\cdots \\
h_2(x) &= \frac{r_0(x)}{r_1(x)} &&= w_2(x)w_3(x)\cdots \\
h_3(x) &= \frac{r_1(x)}{r_2(x)} &&= w_3(x)\cdots \\
&\vdots \\
h_{m_{\max}}(x) &= \frac{r_{m_{\max}-2}}{r_{m_{\max}-1}} &&= w_{m_{\max}}(x)
\end{aligned}$$

and thus all the functions, $w_1(x), w_2(x), \cdots, w_{m_{\max}}(x)$, are determined from

$$w_1(x) = \frac{h_1(x)}{h_2(x)}, \quad w_2(x) = \frac{h_2(x)}{h_3(x)}, \quad \cdots \quad , w_{m_{\max}-1}(x) = \frac{h_{m_{\max}-1}(x)}{h_{m_{\max}}(x)}$$

until

$$w_{m_{\max}}(x) = h_{m_{\max}}(x)$$

The equations

$$w_1(x) = 0, \quad w_2(x) = 0, \quad \cdots \quad , w_{m_{\max}}(x) = 0$$

contain only simple roots: In particular

- All the roots of w_i are roots of multiplicity i of $f(x)$

□

The algorithm requires repeated GCD and polynomial division computations:

- The GCD computations are achieved by using the Sylvester matrix and its sub-resultant matrices
- The polynomial division computations are implemented by a deconvolution operation

4. The Sylvester resultant matrix

The Sylvester resultant matrix $S(f, g)$ of the polynomials

$$f(x) = \sum_{i=0}^{m} a_i x^{m-i} \qquad \text{and} \qquad g(x) = \sum_{j=0}^{n} b_j x^{n-j}$$

has two important properties:

- The determinant of $S(f, g)$ is equal to zero if and only if $f(x)$ and $g(x)$ have a non-constant common divisor
- The rank of $S(f, g)$ is equal to $(m + n - d)$, where d is the degree of the greatest common divisor (GCD) of $f(x)$ and $g(x)$

The matrix $S(f,g)$ is

$$\left[\begin{array}{ccccc|ccccc}
a_0 & & & & & b_0 & & & & \\
a_1 & a_0 & & & & b_1 & b_0 & & & \\
\vdots & a_1 & \ddots & & & \vdots & b_1 & \ddots & & \\
a_{m-1} & \vdots & \ddots & a_0 & & b_{n-1} & \vdots & \ddots & b_0 & \\
a_m & a_{m-1} & \ddots & a_1 & & b_n & b_{n-1} & \ddots & b_1 & \\
 & a_m & \ddots & \vdots & & & b_n & \ddots & \vdots & \\
 & & \ddots & a_{m-1} & & & & \ddots & b_{n-1} & \\
 & & & a_m & & & & & b_n &
\end{array}\right]$$

The k'th Sylvester matrix, or subresultant, $S_k \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$ is a submatrix of $S(f,g)$ that is formed by:

- Deleting the last $(k-1)$ rows of $S(f,g)$
- Deleting the last $(k-1)$ columns of the coefficients of $f(x)$
- Deleting the last $(k-1)$ columns of the coefficients of $g(x)$

The integer k satisfies $1 \leq k \leq \min(m,n)$, and a subresultant matrix is defined for each value of k.

- Start with $k = k_0 = \min(m,n)$ and decrease by one until a solution is obtained.

Each matrix S_k is partitioned into:

- A vector $c_k \in \mathbb{R}^{m+n-k+1}$, where c_k is the first column of S_k
- A matrix $A_k \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+1)}$, where A_k is the matrix formed from the remaining columns of S_k

$$
\begin{aligned}
S_k &= \left[\begin{array}{c|c} c_k & A_k \end{array}\right] \\
&= \left[\begin{array}{c|c|c} c_k & \text{coeffs. of } f(x) & \text{coeffs. of } g(x) \end{array}\right] \\
&\qquad \underbrace{\qquad}_{1} \quad \underbrace{\qquad\qquad}_{n-k} \quad \underbrace{\qquad\qquad}_{m-k+1}
\end{aligned}
$$

Theorem 1 Consider the polynomials $f(x)$ and $g(x)$, and let k be a positive integer, where $1 \leq k \leq \min(m, n)$. Then

1. The dimension of the null space of S_k is greater than or equal to one if and only if the over determined equation

$$A_k y = c_k$$

possesses a solution.

2. A necessary and sufficient condition for the polynomials $f(x)$ and $g(x)$ to have a common divisor of degree greater than or equal to k is that the dimension of the null space of S_k is greater than or equal to one.

5. The method of structured total least norm

The problem to be solved is:

For a given value of k, compute the smallest perturbations to $f(x)$ and $g(x)$ such that

$$(A_k + E_k)\, y = c_k + h_k$$

has a solution, where

- E_k has the same structure as A_k
- h_k has the same structure as c_k
- Start with $k = \min(m, n)$ and decrease by one until a solution is obtained.

- z_i be the perturbation of $a_i, i = 0, \ldots, m$, of $f(x)$
- z_{m+1+j} be the perturbation of $b_j, j = 0, \ldots, n$, of $g(x)$

The perturbation of the Sylvester resultant matrix is:

$$\left[\begin{array}{c|c} h_k & E_k \end{array}\right] = \begin{bmatrix} z_0 & & & & z_{m+1} & & \\ z_1 & z_0 & & & z_{m+2} & & \\ \vdots & z_1 & \ddots & & \vdots & \ddots & \\ z_{m-1} & \vdots & \ddots & z_0 & z_{m+n} & \ddots & z_{m+1} \\ z_m & z_{m-1} & \ddots & z_1 & z_{m+n+1} & \ddots & z_{m+2} \\ & z_m & \ddots & \vdots & & \ddots & \vdots \\ & & \ddots & z_{m-1} & & \ddots & z_{m+n} \\ & & & z_m & & & z_{m+n+1} \end{bmatrix}$$

The method of STLN is used to solve the following problem:

$$\min_{z} \|Dz\| \quad \text{where} \quad D = \begin{bmatrix} (n-k+1)I_{m+1} & 0 \\ 0 & (m-k+1)I_{n+1} \end{bmatrix}$$

such that

$$(A_k + E_k)\, y = c_k + h_k$$

and

- E_k has the same structure as A_k
- h_k has the same structure as c_k

The simple polynomial root finder described earlier requires successive GCD calculations, so the algorithm above can be used repeatedly. But

- Can the value of k be chosen automatically, rather than selected by the user?

Use the principle of Minimum Description Length

6. The principle of Minimum Description Length (MDL)

The principle of MDL is an information theoretic criterion for the selection of a hypothesis from a set of hypotheses that explains a given set of data.

Let r be the (unknown) rank of the theoretically exact form of a given inexact matrix.

- Assign a Gaussian probability distribution to the first r singular values, and a one-sided distribution to the remaining singular values
- Use the given inexact data and the method of maximum likelihood to estimate the parameters of these distributions
- Use the principle of MDL to derive an expression for the complexity of this model as a function of r, and select the value of r for which the complexity is a minimum

- MDL measures the complexity of a model by the length of its binary representation
  - Complex models require more bits for their description than do simple models
  - Choose the model that requires the fewest bits for its description

The principle of MDL is in accord with Occam's razor because it attempts to find the simplest model that explains the data.

- Apply the principle of MDL to each of the GCD calculations in order to calculate the degree of the GCD
- The sequence of computed values of r cannot increase

7. A polynomial root solver

1. Perform the sequence of GCD computations, using structured matrices and the principle of MDL
2. Perform a sequence of polynomial divisions to obtain a set of polynomials, each of whose roots is simple
3. Use a standard method to compute the roots of these polynomials
4. Use the method of nonlinear least squares to refine the roots, using the values in the previous step as the initial estimates

Example The computed roots of the following polynomial in the absence of noise, and with a signal-to-noise ratio of 10^{10}, were calculated

$$f(x) = (x - 0.1)^{10}(x - 0.5)^{8}(x - 0.9)^{6}$$

No noise	
root	multiplicity
0.1	10
0.5	8
0.9	6

S/N =10^{10}	
root	multiplicity
0.10000000001	10
0.49999999990	8
0.90000000014	6

Example The computed roots of the following polynomial in the absence of noise, and with a signal-to-noise ratio of 10^{10}, were calculated

$$f(x) = (x - 0.1)^{15}(x - 0.2)^{30}$$

No noise	
root	multiplicity
0.1	15
0.2	30

S/N =10^{10}	
root	multiplicity
0.10000000003	15
0.19999999998	30