Structured matrix methods for the calculation of the roots of inexact polynomials

Joab R. Winkler, Madina Hasan, Xin Lao

Department of Computer Science

The University of Sheffield

United Kingdom

MOPNET, Nottingham, March 2010

CONTENTS

1. DIFFICULTIES OF COMPUTING POLYNOMIAL ROOTS

There exist many algorithms for computing the roots of a polynomial:

- Bairstow, Graeffe, Jenkins-Traub, Laguerre, Müller, Newton, . . .

These methods yield satisfactory results if:

- The polynomial is of moderate degree
- The roots are simple and well-separated
- A good starting point in the iterative scheme is used

This heuristic has exceptions:

$$f(x) = \prod_{i=1}^{20}(x-i) = (x-1)(x-2)\cdots(x-20)$$

Example 1.1 Consider the polynomial

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

whose root is $x = 1$ with multiplicity 4. MATLAB returns the roots

1.0002, 1.0000 + 0.0002i, 1.0000 - 0.0002i, 0.9998 □

Example 1.2 The roots of the polynomial $(x-1)^{100}$ were computed by MATLAB.

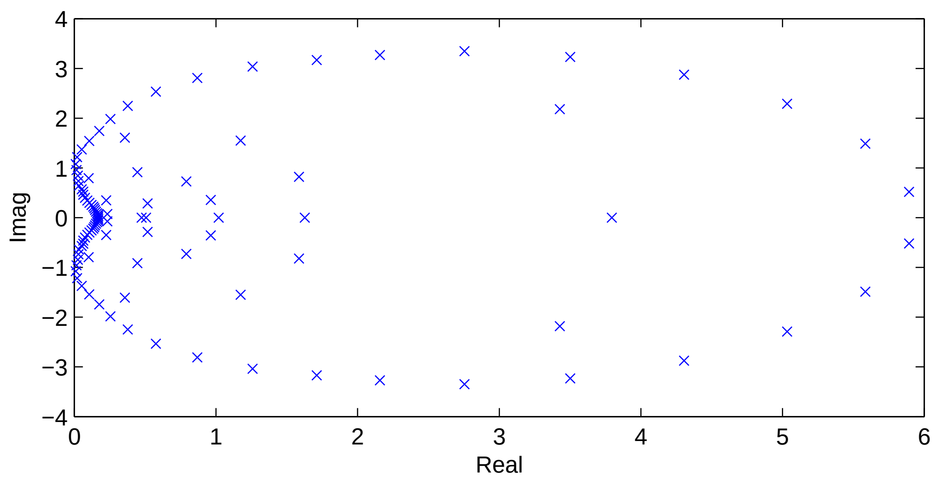

Figure 1.1: The computed roots of $(x-1)^{100}$.

□

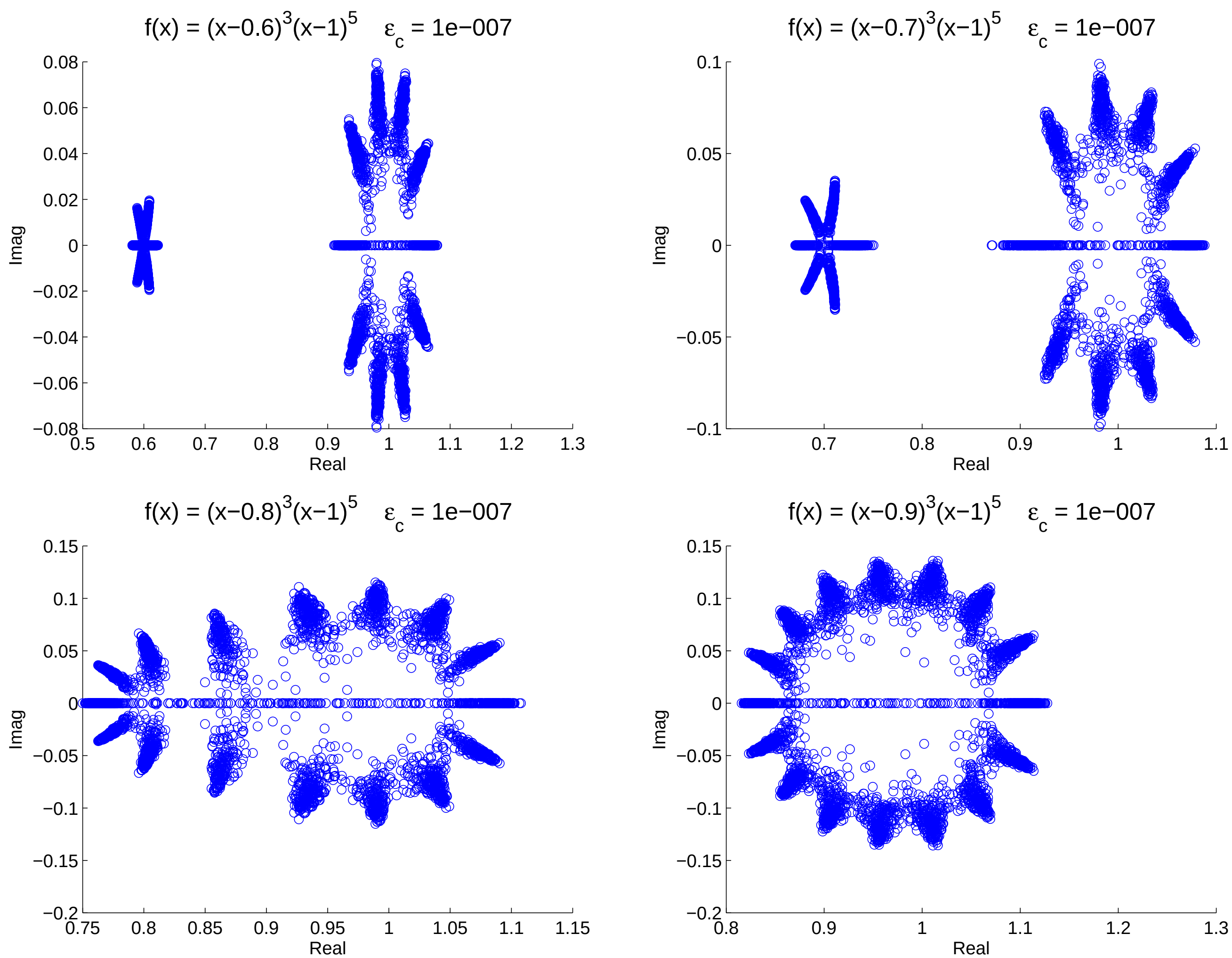

Figure 1.2: The root distribution of four perturbed polynomials.

Example 1.3

$$\frac{\text{componentwise noise amplitude}}{\text{componentwise signal amplitude}} = 10^{-8}$$

exact root	multiplicity	computed root	relative error
-6.7547000000e-1	4	-6.7547000082e-1	1.2139725913e-9
5.7335000000e+0	6	5.7335000822e+0	1.4344694923e-8
2.1747000000e+0	7	2.17469999237+0	3.5069931355e-9
-9.5568000000e+0	10	-9.5567996740e+0	3.4111255034e-8
-6.5553000000e+0	11	-6.5553001701e+0	2.5954947075e-8

- The root multiplicities were calculated correctly

□

Example 1.4

$$\frac{\text{componentwise noise amplitude}}{\text{componentwise signal amplitude}} = 10^{-8}$$

exact root	multiplicity	computed root	relative error
-1.1539000000e+0	4	-1.1539000316e+0	2.7389266674e-8
4.0809000000e+0	5	4.0808998890e+0	2.7198581384e-8
-2.1059000000e+0	6	-2.1058999294e+0	3.2521823798e-8
3.6683000000e+0	7	3.6683000481e+0	1.3110114060e-8
-9.6084000000e+0	13	-9.6084001121e+0	1.1664214687e-8

- The root multiplicities were calculated correctly

□

2. THE GEOMETRY OF ILL-CONDITIONED POLYNOMIALS

- A root x_0 of multiplicity r introduces $(r-1)$ constraints on the coefficients.
- A monic polynomial of degree m has m degrees of freedom.
- The root x_0 lies on a manifold of dimension $(m-r+1)$ in a space of dimension m.
- This manifold is called a pejorative manifold because polynomials near this manifold are ill-conditioned.
- A polynomial that lies on a pejorative manifold is well-conditioned with respect to (the structured) perturbations that keep it on the manifold, which corresponds to the situation in which the multiplicity of the roots is preserved.
- A polynomial is ill-conditioned with respect to perturbations that move it off the manifold, which corresponds to the situation in which a multiple root breaks up into a cluster of simple roots.

Example 2.1 Consider a cubic polynomial $f(x)$ with real roots x_0, x_1 and x_2

$$\begin{aligned}(x-x_0)(x-x_1)(x-x_2) \quad &= \quad x^3-(x_0+x_1+x_2)x^2+ \\ &\quad\quad (x_0x_1+x_1x_2+x_2x_0)x-x_0x_1x_2\end{aligned}$$

- If $f(x)$ has one double root and one simple root, then $x_0 = x_1 \neq x_2$ and thus $f(x)$ can be written as

$$x^3-(2x_1+x_2)x^2+(x_1^2+2x_1x_2)x-x_1^2x_2$$

The pejorative manifold of a cubic polynomial that has a double root is the surface defined by

$$\begin{pmatrix} -(2x_1+x_2) & (x_1^2+2x_1x_2) & -x_1^2x_2 \end{pmatrix} \qquad x_1 \neq x_2$$

- If $f(x)$ has a triple root, then $x_0 = x_1 = x_2$ and thus $f(x)$ can be written as

$$x^3 - 3x_0x^2 + 3x_0^2x - x_0^3$$

The pejorative manifold of a cubic polynomial that has a triple root is the curve defined by

$$\begin{pmatrix} -3x_0 & 3x_0^2 & -x_0^3 \end{pmatrix}$$

$\square$

Theorem 2.1 The condition number of the real root x_0 of multiplicity r of the polynomial $f(x) = (x - x_0)^r$, such that the perturbed polynomial also has a root of multiplicity r, is

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

where $\|\cdot\| = \|\cdot\|_2$ and

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

Example 2.2 The condition number $\rho(1)$ of the root $x_0 = 1$ of $(x-1)^r$ is

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

This expression reduces to

$$\rho(1) = \frac{1}{r}\sqrt{\frac{\binom{2r}{r}}{\binom{2(r-1)}{r-1}}} = \frac{1}{r}\sqrt{\frac{2(2r-1)}{r}} \approx \frac{2}{r} \quad \text{for large } r$$

Compare with the componentwise and normwise condition numbers

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

- $\rho(1)$ is independent of the the noise level (assumed to be small)
- $\rho(1)$ decreases as the multiplicity r of the root $x_0 = 1$ increases □

3. A SIMPLE POLYNOMIAL ROOT FINDER

Let $w_i(x)$ be the product of all factors of degree i of $f(x)$

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

Perform a sequence of greatest common divisor (GCD) computations

$$\begin{aligned}
q_1(x) &= \text{GCD}\left(f(x), f^{(1)}(x)\right) &&= w_2(x)w_3^2(x)w_4^3(x)\cdots w_{r_{\max}}^{r_{\max}-1}(x) \\
q_2(x) &= \text{GCD}\left(q_1(x), q_1^{(1)}(x)\right) &&= w_3(x)w_4^2(x)w_5^3(x)\cdots w_{r_{\max}}^{r_{\max}-2}(x) \\
q_3(x) &= \text{GCD}\left(q_2(x), q_2^{(1)}(x)\right) &&= w_4(x)w_5^2(x)w_6^3(x)\cdots w_{r_{\max}}^{r_{\max}-3}(x) \\
q_4(x) &= \text{GCD}\left(q_3(x), q_3^{(1)}(x)\right) &&= w_5(x)w_6^2(x)w_7^3(x)\cdots w_{r_{\max}}^{r_{\max}-4}(x) \\
&\;\;\vdots
\end{aligned}$$

The sequence terminates at $q_{r_{\max}}(x)$, which is a constant.

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

$$\begin{aligned}
h_1(x) &= \frac{f(x)}{q_1(x)} &= w_1(x)w_2(x)w_3(x)\cdots \\
h_2(x) &= \frac{q_1(x)}{q_2(x)} &= w_2(x)w_3(x)\cdots \\
h_3(x) &= \frac{q_2(x)}{q_3(x)} &= w_3(x)\cdots \\
&\vdots \\
h_{r_{\max}}(x) &= \frac{q_{r_{\max}-2}}{q_{r_{\max}-1}} &= w_{r_{\max}}(x)
\end{aligned}$$

The functions, $w_1(x), w_2(x), \cdots, w_{r_{\max}}(x)$, are determined from

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

until

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

The equations

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

contain only simple roots, and they yield the simple, double, triple, etc., roots of $f(x)$.

- If x_0 is a root of $w_i(x)$, then it is a root of multiplicity i of $f(x)$.

$\square$

Mathematical operations performed in this root finder:

- GCD computations
- Polynomial divisions
- Solution of simple polynomial equations

Example 3.1 Calculate the roots of the polynomial

$$f(x) = x^6 - 3x^5 + 6x^3 - 3x^2 - 3x + 2$$

whose derivative is

$$f^{(1)}(x) = 6x^5 - 15x^4 + 18x^2 - 6x - 3$$

Perform a sequence of GCD computations

$$\begin{aligned} q_1(x) &= \text{GCD}\left(f(x), f^{(1)}(x)\right) &&= x^3 - x^2 - x + 1 \\ q_2(x) &= \text{GCD}\left(q_1(x), q_1^{(1)}(x)\right) &&= x - 1 \\ q_3(x) &= \text{GCD}\left(q_2(x), q_2^{(1)}(x)\right) &&= 1 \end{aligned}$$

The maximum degree of a divisor of $f(x)$ is 3 because the sequence terminates at $q_3(x)$.

The polynomials $h_i(x)$ are:

$$\begin{aligned} h_1(x) &= \frac{f(x)}{q_1(x)} &= x^3 - 2x^2 - x + 2 \\ h_2(x) &= \frac{q_1(x)}{q_2(x)} &= x^2 - 1 \\ h_3(x) &= \frac{q_2(x)}{q_3(x)} &= x - 1 \end{aligned}$$

The polynomials $w_i(x)$ are

$$\begin{aligned} w_1(x) &= \frac{h_1(x)}{h_2(x)} &= x - 2 \\ w_2(x) &= \frac{h_2(x)}{h_3(x)} &= x + 1 \\ w_3(x) &= h_3(x) &= x - 1 \end{aligned}$$

and thus the factors of $f(x)$ are

$$f(x) = (x-2)(x+1)^2(x-1)^3$$

$\square$

3.1 Discussion of method

- The computation of the GCD of two polynomials is an ill-posed problem because it is not a continuous function of their coefficients:
  - The polynomials $f(x)$ and $g(x)$ may have a non-constant GCD, but the perturbed polynomials $f(x) + \delta f(x)$ and $g(x) + \delta g(x)$ may be coprime.
- The determination of the degree of the GCD of two polynomials reduces to the determination of the rank of a resultant matrix, but the rank of a matrix is not defined in a floating point environment.
- Polynomial division is an ill-posed problem:

Even if $\dfrac{f(x)}{g(x)}$ is a polynomial,

$\dfrac{f(x) + \delta f(x)}{g(x) + \delta g(x)}$ is a rational function for arbitrary $\delta f(x)$ and $\delta g(x)$

4. APPROXIMATE GREATEST COMMON DIVISORS

If $f(x)$ is exact and all computations are performed in a symbolic environment, the GCD of $f(x)$ and its derivative $f^{(1)}(x)$ can be computed by the Sylvester resultant matrix $S(f, f^{(1)})$.

The polynomial $f(x)$ is rarely known exactly, and so the given data is

$$\tilde{f}(x) = f(x) + \delta f(x)$$

and $\tilde{f}(x)$ and $\tilde{f}^{(1)}(x)$ are (with probability almost 1) coprime.

- The polynomials $\tilde{f}(x)$ and $\tilde{f}^{(1)}(x)$ have an approximate greatest common divisor (AGCD).

- Use the method of structured total least norm applied to $S(\tilde{f}, \tilde{f}^{(1)})$ to compute the smallest perturbation of $S(\tilde{f}, \tilde{f}^{(1)})$ such that its perturbed form is singular, which implies that the perturbed form $\tilde{f}(x)$ of $f(x)$ has a multiple root.

4.1 The Sylvester resultant matrix

The product of two polynomials is equal to the convolution of their coefficients:

$$\begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{m+n-1} \\ r_{m+n} \end{bmatrix} = \begin{bmatrix} p_0 & & & \\ p_1 & p_0 & & \\ \vdots & p_1 & \ddots & \\ p_m & \vdots & \ddots & p_0 \\ & p_m & \ddots & p_1 \\ & & \ddots & \vdots \\ & & & p_m \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ \vdots \\ q_{n-1} \\ q_n \end{bmatrix}$$

$$\mathbf{r} = C_{n+1}(p)\mathbf{q} = \mathbf{p} \otimes \mathbf{q}$$

$\mathbf{r} \in \mathbb{R}^{m+n+1}$, $\mathbf{p} \in \mathbb{R}^{m+1}$, $\mathbf{q} \in \mathbb{R}^{n+1}$ and $C_{n+1}(p) \in \mathbb{R}^{(m+n+1)\times(n+1)}$

Let:

- $d_k(y)$ be a common divisor of degree k of the exact polynomials $f(y)$ and $f^{(1)}(y)$
- The degree of the GCD of $f(y)$ and $f^{(1)}(y)$ be $\hat{d}$
- $u_k(y)$ and $v_k(y)$ be the quotient polynomials

$$f(y) = u_k(y)d_k(y) \qquad \text{and} \qquad f^{(1)}(y) = v_k(y)d_k(y)$$

Thus

$$f(y)v_k(y) - f^{(1)}(y)u_k(y) = 0 \quad \Leftrightarrow \quad C_{m-k}(f)\mathbf{v}_k - C_{m-k+1}(f^{(1)})\mathbf{u}_k = 0$$

where

$$C_{m-k}(f) \in \mathbb{R}^{(2m-k)\times(m-k)} \qquad \text{and} \qquad \mathbf{v}_k \in \mathbb{R}^{m-k}$$

$$C_{m-k+1}(f^{(1)}) \in \mathbb{R}^{(2m-k)\times(m-k+1)} \qquad \text{and} \qquad \mathbf{u}_k \in \mathbb{R}^{m-k+1}$$

$$\left[\begin{array}{cc} C_{m-k}(f) & C_{m-k+1}(f^{(1)}) \end{array}\right] \left[\begin{array}{c} \mathbf{v}_k \\ -\mathbf{u}_k \end{array}\right] = S_k(f, f^{(1)}) \left[\begin{array}{c} \mathbf{v}_k \\ -\mathbf{u}_k \end{array}\right] = 0$$

- $S_k(f, f^{(1)}) \in \mathbb{R}^{(2m-k)\times(2m-2k+1)}$ and it is rank deficient
- The nullspace vectors yield the coefficients of the quotient polynomials
- Since the degree of the GCD of $f(y)$ and $f^{(1)}(y)$ is $\hat{d}$, these polynomials possess common divisors of degrees $1, 2, \ldots, \hat{d}$, but not a divisor of degree $\hat{d}+1$:

$$\begin{aligned} \text{rank}\, S_k(f, f^{(1)}) &< 2m-2k+1, \qquad k = 1, \ldots, \hat{d} \\ \text{rank}\, S_k(f, f^{(1)}) &= 2m-2k+1, \qquad k = \hat{d}+1, \ldots, m-1 \end{aligned}$$

Calculating the degree of the GCD reduces to estimating the rank of a matrix

Example 4.1 Consider $S_k(f, f^{(1)})$, for $k = 1, 2, 3$, for

$$
\begin{aligned}
f(x) &= (x-1)^2(x-2)(x-3) = x^4 - 7x^3 + 17x^2 - 17x + 6 \\
f^{(1)}(x) &= 4x^3 - 21x^2 + 34x - 17
\end{aligned}
$$

Hence $S_1(f, f^{(1)}) = S(f, f^{(1)})$ is equal to

$$
\begin{bmatrix}
1 & 0 & 0 & 4 & 0 & 0 & 0 \\
-7 & 1 & 0 & -21 & 4 & 0 & 0 \\
17 & -7 & 1 & 34 & -21 & 4 & 0 \\
-17 & 17 & -7 & -17 & 34 & -21 & 4 \\
6 & -17 & 17 & 0 & -17 & 34 & -21 \\
0 & 6 & -17 & 0 & 0 & -17 & 34 \\
0 & 0 & 6 & 0 & 0 & 0 & -17
\end{bmatrix}
$$

and this matrix has a unit loss of rank.

The subresultant matrix $S_2(f, f^{(1)})$ is

$$S_2(f, f^{(1)}) = \begin{bmatrix} 1 & 0 & 4 & 0 & 0 \\ -7 & 1 & -21 & 4 & 0 \\ 17 & -7 & 34 & -21 & 4 \\ -17 & 17 & -17 & 34 & -21 \\ 6 & -17 & 0 & -17 & 34 \\ 0 & 6 & 0 & 0 & -17 \end{bmatrix}$$

and this matrix has full column rank.

The subresultant matrix $S_3(f, f^{(1)})$ is

$$S_3(f, f^{(1)}) = \begin{bmatrix} 4 & 0 & 1 \\ -21 & 4 & -7 \\ 34 & -21 & 17 \\ -17 & 34 & -17 \\ 0 & -17 & 6 \end{bmatrix}$$

and this matrix has full column rank.

It follows that the first rank deficient matrix in the sequence

$$S_3(f, f^{(1)}), S_2(f, f^{(1)}), S_1(f, f^{(1)})$$

is $S_1(f, f^{(1)})$, and thus the degree of the GCD of $f(x)$ and $f^{(1)}(x)$ is one. $\square$

4.2 Pre-processing operations for the computation of an AGCD

The computation of an AGCD of $f(x)$ and $f^{(1)}(x)$ requires that two pre-processing operations be performed:

- $f(x)$ and $f^{(1)}(x)$ must be normalised to balance the Sylvester matrix
- An AGCD of $f(x)$ and $f^{(1)}(x)$ is equal to, up to a scalar multiplier, an AGCD of $f(x)$ and $\alpha f^{(1)}(x)$, where α is an arbitrary non-zero constant.

$$\text{GCD}\left(f, f^{(1)}\right) \sim \text{GCD}\left(f, \alpha f^{(1)}\right), \qquad \alpha \neq 0$$

  - The resultant matrix $S(f, \alpha f^{(1)})$ should be used when it is desired to compute an AGCD of $f(x)$ and $f^{(1)}(x)$
  - How is the optimal value of α computed?

1. Normalisation: Define $f(x)$ and $g(x)$ as

$$f(x) = \sum_{i=0}^{m} \bar{a}_i x^{m-i}, \qquad \bar{a}_i = \frac{a_i}{\left(\prod_{j=0}^{m} |a_j|\right)^{\frac{1}{m+1}}}$$

$$g(x) = \sum_{i=0}^{m-1} \bar{b}_i x^{m-1-i}, \qquad \bar{b}_i = \frac{(m-i)\bar{a}_i}{\left(\prod_{j=0}^{m-1} |(m-j)\bar{a}_j|\right)^{\frac{1}{m}}}$$

Note: $g(x)$ is proportional to $f^{(1)}(x)$

2. The optimal value of α: Use linear programming to calculate α_0, the optimal value of α

4.3 The coprime polynomials and the degree of an AGCD

Recall that

$$S_k(f,g)\begin{bmatrix} \mathbf{v}_k \\ -\mathbf{u}_k \end{bmatrix} = 0$$

- $S_k(f,g) \in \mathbb{R}^{(2m-k)\times(2m-2k+1)}$ and it is rank deficient
- The nullspace vectors yield the coefficients of the quotient polynomials
- If the degree of an GCD of $f(x)$ and $g(x)$ is d, then

$$\begin{aligned} \text{rank } S_k(f,g) &< 2m-2k+1, \qquad k=1,\ldots,d \\ \text{rank } S_k(f,g) &= 2m-2k+1, \qquad k=d+1,\ldots,m-1 \end{aligned}$$

Use the same criterion for the calculation of the degree of an AGCD

- The degree d of an AGCD of $f(x)$ and $g(x)$ is equal to the largest value of k, $k = 1, \ldots, m-1$, such that $S_k(f, g)$ is numerically singular:
  - The SVD of $S_k(f, g)$ cannot be used because $f(x)$ and $g(x)$ are inexact and therefore, with high probability, coprime
  - The property

$$\text{numerical rank } S_k(f, \alpha_0 g) = \text{numerical rank } S_k\left(g, \frac{f}{\alpha_0}\right)$$

    enables a criterion for the calculation of d to be developed

- $S_d(f, \alpha_0 g)$ is numerically rank deficient by one and estimates of the coprime polynomials can be calculated from its nullspace

$$S_d(f, \alpha_0 g) \begin{bmatrix} \mathbf{v}_d \\ -\mathbf{u}_d \end{bmatrix} \approx 0$$

5. STRUCTURED TOTAL LEAST NORM

Recall that $S_d(f, \alpha_0 g)$ is numerically rank deficient by one and

$$S_d(f, \alpha_0 g) \begin{bmatrix} \mathbf{v}_d \\ -\mathbf{u}_d \end{bmatrix} \approx 0$$

If this equation is satisfied exactly, then

- The coprime polynomials are defined by the null space of $S_d(f, \alpha_0 g)$
- $f(y)$ and $g(y)$ have a non-constant common divisor:
  - $f(y)$ has a multiple root
  - $g(y)$ has been moved to a pejorative manifold

The calculation of d determines the column q of $S_d(f, \alpha_0 g)$ such that if

$$S_d(f, \alpha_0 g) = \begin{bmatrix} c_1 & c_1 & \cdots & c_{q-1} & c_q & c_{q+1} & \cdots & c_{2m-2k+1} \end{bmatrix}$$

then the approximate homogeneous equation

$$S_d(f, \alpha_0 g) \begin{bmatrix} \mathbf{v}_d \\ -\mathbf{u}_d \end{bmatrix} \approx 0$$

can be transformed into an approximate linear algebraic equation

$$\begin{bmatrix} c_1 & c_1 & \cdots & c_{q-1} & c_{q+1} & \cdots & c_{2m-2d+1} \end{bmatrix} x \approx c_q$$

or

$$A_d x \approx c_q, \qquad A_d \in \mathbb{R}^{(2m-d)\times(2m-2d)}, \qquad c_q \in \mathbb{R}^{2m-d}$$

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_{q-1} \\ x_{q+1} \\ \vdots \\ x_{2m-2d} \end{bmatrix} \in \mathbb{R}^{2m-2d}, \quad x_q = -1, \quad \begin{bmatrix} x_1 \\ \vdots \\ x_{q-1} \\ -1 \\ x_{q+1} \\ \vdots \\ x_{2m-2d} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_d \\ -\mathbf{u}_d \end{bmatrix}$$

Recall:

- The degree of an AGCD of the given inexact polynomials $f(y)$ and $g(y)$ is d
- The matrix A_d and vector c_q are functions of the coefficients of $f(y)$ and $g(y)$.
- $$A_d x \approx c_q$$

  is an approximate equation because its arguments are the coefficients of inexact polynomials

  Use structured total least norm to solve this approximate equation

Given the inexact polynomials $f(y)$ and $g(y)$, which are assumed to be coprime, calculate the smallest perturbations that must be added to their coefficients such that the perturbed forms of $f(y)$ and $g(y)$ have a non-constant GCD.

Aim:

Compute the Sylvester matrix $S(\delta f, \alpha_0 \delta g)$, such that

$$\|\delta f\|^2 + \|\delta g\|^2$$

is minimised, where

$$S_d(f + \delta f, \alpha_0(g + \delta g)) = S_d(f, \alpha_0 g) + S_d(\delta f, \alpha_0 \delta g)$$

is rank deficient.

The approximate under-determined equation

$$A_d x \approx c_q$$

is corrected by considering the equation

$$\left(A_d(\alpha_0) + E_d(\alpha_0, z)\right) x = c_q(\alpha_0) + h_q(\alpha_0, z)$$

which is non-linear in x and z.

- A_d and E_d have the same structure, and c_d and h_d have the same structure:
  - Use the method of structured total least norm
- The initial vector of perturbations is $z = 0$
- Solve this non-linear under-determined equation subject to the constraint that $\|z\|^2$ is minimised
- This leads to a least squares equality problem

The Sylvester matrix $S_d(f, \alpha_0 g)$ is

$$\begin{bmatrix} \bar{a} & & & \alpha_0 \bar{b} & & \\ \bar{a}_1 & \ddots & & \alpha_0 \bar{b}_1 & \ddots & \\ \vdots & \ddots & \bar{a} & \vdots & \ddots & \alpha_0 \bar{b} \\ \bar{a}_{m-1} & \ddots & \bar{a}_1 & \alpha_0 \bar{b}_{n-1} & \ddots & \alpha_0 \bar{b}_1 \\ \bar{a}_m & \ddots & \vdots & \alpha_0 \bar{b}_n & \ddots & \vdots \\ & \ddots & \bar{a}_{m-1} & & \ddots & \alpha_0 \bar{b}_{n-1} \\ & & \bar{a}_m & & & \alpha_0 \bar{b}_n \end{bmatrix}$$

If the perturbations of the coefficients of $f(y)$ and $\alpha_0 g(y)$ are

$$z_i, i = 0, \ldots, m \qquad \text{and} \qquad \alpha_0 z_{m+1+i}, i = 0, \ldots, n$$

respectively, then $E_d(\alpha_0, z)$ is equal to

$$\begin{bmatrix} z_0 & & & \alpha_0 z_{m+1} & & \\ z_1 & \ddots & & \alpha_0 z_{m+2} & \ddots & \\ \vdots & \ddots & z_0 & \vdots & \ddots & \alpha_0 z_{m+1} \\ z_{m-1} & \ddots & z_1 & \alpha_0 z_{m+n} & \ddots & \alpha_0 z_{m+2} \\ z_m & \ddots & \vdots & \alpha_0 z_{m+n+1} & \ddots & \vdots \\ & \ddots & z_{m-1} & & \ddots & \alpha_0 z_{m+n} \\ & & z_m & & & \alpha_0 z_{m+n+1} \end{bmatrix}$$

- Perform these AGCD computations repeatedly in order to determine the multiplicities of the roots
  - These calculations correspond to the identification of the pejorative manifold on which the theoretically exact polynomial lies

The other stages in the algorithm:

- Use the method of least squares to perform the polynomial division
- Recall it is necessary to solve the polynomial equations

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

  - Calculate the roots of the polynomial by solving these equations

6. EXAMPLES

Example 6.1

$$\frac{\text{componentwise noise amplitude}}{\text{componentwise signal amplitude}} = 10^{-6}$$

exact root	multiplicity	computed root	relative error
8.1031100000e+0	2	8.1031311463e+0	3.8437554630e-6
3.5078000000e+0	8	3.5077983383e+0	4.7372726251e-7
-6.3060000000e-1	8	-6.3060013449e-1	2.1327857935e-7
-5.8211000000e+0	9	-5.8210973315e+0	4.5841110328e-7

- The root multiplicities were calculated correctly

Example 6.2

$$\frac{\text{componentwise noise amplitude}}{\text{componentwise signal amplitude}} = 10^{-8}$$

exact root	multiplicity	computed root	relative error
-7.3132000000e+0	1	-7.3131318042e+0	9.3250329595e-6
9.0183000000e+0	2	9.0182738917e+0	2.8950322472e-6
4.4470000000e+0	3	4.4469987289e+0	2.8582679636e-7
6.6374000000e+0	4	6.6374090279e+0	1.3601587262e-6
-1.9984000000e+0	4	-1.9984000974e+0	4.8743537340e-8
-8.7907000000e+0	6	-8.7907151653e+0	1.7251509523e-6

- The root multiplicities were calculated correctly

□