Jiří Eckstein; Jan Zítko

Comparison of algorithms for calculation of the greatest common divisor of several polynomials

In: Jan Chleboun and Petr Přikryl and Karel Segeth and Jakub Šístek and Tomáš Vejchodský (eds.): Programs and Algorithms of Numerical Mathematics, Proceedings of Seminar. Dolní Maxov, June 8-13, 2014. Institute of Mathematics AS CR, Prague, 2015. pp. 64--70.

Persistent URL: http://dml.cz/dmlcz/702664

Programs and Algorithms of Numerical Matematics 17
J. Chleboun, P. Přikryl, K. Segeth, J. Šístek, T. Vejchodský (Eds.)
Institute of Mathematics AS CR, Prague 2015

COMPARISON OF ALGORITHMS FOR CALCULATION OF THE GREATEST COMMON DIVISOR OF SEVERAL POLYNOMIALS

Jiří Eckstein, Jan Zítko

Department of Numerical Mathematics,
Faculty of Mathematics and Physics, Charles University
Sokolovská 83, Prague 8, Czech Republic
jiri.eckstein@gmail.com, jan_zitko@centrum.cz

Abstract

The computation of the greatest common divisor (GCD) has many applications in several disciplines including computer graphics, image deblurring problem or computing multiple roots of inexact polynomials. In this paper, Sylvester and Bézout matrices are considered for this purpose. The computation is divided into three stages. A rank revealing method is shortly mentioned in the first one and then the algorithms for calculation of an approximation of GCD are formulated. In the final stage the coefficients are improved using Gauss-Newton method. Numerical results show the efficiency of proposed last two stages.

1. Introduction

Sylvester matrices (see [1, 3, 5, 6, 10, 11, 14, 15, 16, 17]) or Bézout matrices (see [3, 7, 8, 12]) can be used for the calculation of GCD. We start with Sylvester matrix. The coefficients of GCD of two polynomials f_1 and f_2 can be obtained from a Sylvester subresultant $S_k(f_1, f_2)$ which is formed from the Sylvester matrix $S(f_1, f_2)$ by deleting the last $k-1$ rows, the last $k-1$ columns of the coefficients of f_1 and the last $k-1$ columns of the coefficients of f_2. If $n_i = \deg(f_i)$ for $i = 1, 2$, $n_1 \geq n_2$, and if for a positive integer $d \leq n_2$ the subresultant $S_d(f_1, f_2)$ is the first rank deficient matrix in the sequence

$$S_{n_2}(f_1, f_2), S_{n_2-1}(f_1, f_2), \ldots, S_1(f_1, f_2), \tag{1}$$

then $d = \deg(\mathrm{GCD}(f_1, f_2))$. There are two well-known procedures for calculation of the rank (or rank deficiency) of a matrix. For small dimension the usage of SVD is sufficient (see for example [2] or [9]). The numerical rank revealing algorithm with many robust examples is in detail described in the papers [11, 17]. The whole process is in details, together with the calculation of GCD and rank determination, explained

in the papers [17], [11], [14] and therefore, in the following, we are considering the calculation for m polynomials.

We now consider m real polynomials $f_1,\ f_2,\ldots f_m$. Let n_i =deg(f_i). Denote g =GCD$(f_1, f_2, \ldots, f_m)$. It is assumed that d=deg$(g) > 0$. The objective is to find polynomials $w_1,\ w_2,\ \ldots,\ w_m$ of degrees $n_1 - d,\ n_2 - d, \ldots,\ n_m - d$ respectively, such that $f_i = w_i\, g$ for all considered i, which can be expressed in the form (see [4, 16, 17])

$$C_d(w_i)\vec{g} = \vec{f_i}, \qquad \text{for} \quad i \in \{1, 2, \ldots, m\}, \tag{2}$$

where $C_d(w_i)$ is the Cauchy matrix for the polynomial w_i with $d+1$ columns, i.e., $C_d(w_i) \in \mathbb{R}^{(n_i+1)\times(d+1)}$. The symbol $\vec{g}$ denotes the vector of coefficients of g and the symbols $\vec{f_i}$ and $\vec{w_i}$ have an analogous meaning. The system (2) can be rewritten in the form

$$F(\mathbf{x}) = \mathbf{b}, \text{ where } \quad \mathbf{x} = \begin{bmatrix} \vec{g} \\ \vec{w_1} \\ \vec{w_2} \\ \cdot \\ \cdot \\ \cdot \\ \vec{w_m} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ \vec{f_1} \\ \vec{f_2} \\ \cdot \\ \cdot \\ \cdot \\ \vec{f_m} \end{bmatrix}, \quad F(\mathbf{x}) = \begin{bmatrix} (\vec{r})^T\vec{g} \\ C_d(w_1)\vec{g} \\ C_d(w_2)\vec{g} \\ \cdot \\ \cdot \\ \cdot \\ C_d(w_m)\vec{g} \end{bmatrix} \tag{3}$$

and $\vec{r}$ is a scaling vector (see [16]). Let us remark that $\vec{r}, \vec{g} \in \mathbb{R}^{d+1}$, and $\vec{w_j} \in \mathbb{R}^{n_j-d+1}$. The system (3) represents $\left(\sum_{j=1}^{m} n_j\right)+m+1$ equations with $\left(\sum_{j=1}^{m} n_j\right)+m+1-(m-1)\,d$ unknowns and the least square solution (see [9]) is applied. According to the well known theory (see [2]) we have

$$\text{grad}\left[\frac{1}{2}\,\|F(\mathbf{x}) - \mathbf{b}\,\|^2\right] = (J(\mathbf{x}))^T\,[F(\mathbf{x}) - \mathbf{b}]\,, \tag{4}$$

where $J(\mathbf{x})$ is the Jacobian of F and can be easily calculated as a Gateaux derivative of F. The problem of location of minimum leads to the solution of the system

$$(J(\mathbf{x}))^T[F(\mathbf{x}) - \mathbf{b}] = 0. \tag{5}$$

Let us mention the result formulated in [16]: for every scaling vector $\vec{r}$ satisfying $\vec{r}^T\vec{g} \neq 0$, if GCD$(w_1, w_2, \ldots, w_m) = 1$, then the Jacobi matrix has a full column rank and therefore $F(\mathbf{x}) = \mathbf{b}$. However all these investigations depend on the basic question how to find the rank d. This is well known for $m = 2$ and it is shortly analysed in Section 2. In the next section the algorithm using Sylvester matrices for $m \geq 3$ is discussed. In Section 3, the calculation of the greatest common divisor of several univariate polynomials through Bézout-like matrices is considered. Both strategies are numerically tested in the last section.

2. Calculation of GCD through Sylvester matrices

At the beginning consider the polynomials f_1 and f_2 of degrees n_1 and n_2 respectively, where $n_1 \geq n_2$. According to the previous section we determine an integer d such that $S_d(f_1, f_2)$ is the first rank deficient matrix in (1) and denote the right singular vector of the matrix $S_d(f_1, f_2) = [C_{n_2-d}(f_1), C_{n_1-d}(f_2)]$ corresponding to the smallest singular value $\sigma_{min}\left(S_d(f_1, f_2)\right)$, which is theoretically equal to zero, by $[(\vec{w}_2)^T, -(\vec{w}_1)^T]^T$. We have denoted g =GCD(f_1, f_2). The coefficients of $\vec{g}$ are calculated as the least square solution of the equation

$$C_d(w_2)\vec{g} = \vec{f_2} \qquad \text{or} \qquad C_d(w_1)\vec{g} = \vec{f_1}. \tag{6}$$

One of these equations (usually the first one) is solved and the second one is used for improvement of the result if it is necessary.

However, for three or more polynomials it is impossible to apply an analogous technique for finding the degree of GCD(f_1, f_2, f_3). A consecutive process is usually applied, which can be formally written for three polynomials in the form

$$d = \deg(\mathrm{GCD}(f_3, \mathrm{GCD}(f_1, f_2))).$$

Numerically, the determination of GCD is usually based on some minimisation method which is formally written by (4), (5) and the realization means an infinite iterative process where only finite number of iterations is implemented. Moreover, if the calculation is performed in floating point environment the result is inexact and therefore an approximation is obtained as a result of the above mentioned minimization process. This approximation to GCD will be in this paper entitled *approximate greatest common divisor* - AGCD. This concept is studied and discussed in many papers (see for example [6, 13, 16]). The concept AGCD is mentioned in context with STLN algorithm (see [10, 15, 13, 6]). In this paper AGCD is the result of the least square procedures which is realized by the Gauss-Newton method. Exact coefficients are assumed. Let us consider the system (5). By analogy to [16] and [17] we now present the algorithm for several polynomials. The numerical process will be evident from the following algorithm.

Algorithm 2.1 (*AGCD for m polynomials.*)
Input: Real polynomials f_1, f_2,..., f_m of degrees n_1, n_2,..., n_m respectively, vector **b** defined by (3) and a given tolerance θ. It is assumed that

$$n_1 \geq n_2 \geq \cdots \geq n_m.$$

Output: Polynomial g =AGCD$(f_1, f_2, \ldots, f_m)$
begin
$g := f_{n_m}$
for $j = m, m-1, \ldots, 2$ **do**
Calculate g =AGCD$(g, f_{n_{j-1}})$.

end
for $j = 1, m$ **do**
$w_j(x) := f_j(x)/g(x)$
end
Put $d :=\deg(g)$;
form the vector $(\mathbf{x})^T = [(\vec{g})^T, (\vec{w_1})^T, (\vec{w_1})^T, \ldots, (\vec{w_m})^T]^T$ for the initial approximation of Gauss-Newton iteration.
repeat

$$\mathbf{x}^+ = \mathbf{x} - \begin{bmatrix} \vec{r} & 0 & 0 & 0 \\ C_d(w_1) & C_{n_1-d}(g) & \cdot & \\ \cdot & 0 & \cdot\ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot & 0 \\ C_d(w_m) & 0 & \cdot\ \cdot & C_{n_m-d}(g) \end{bmatrix}^{\dagger} \begin{bmatrix} \vec{r}^T\vec{g} \\ C_d(w_1)\vec{g} - (\vec{f_1}) \\ \cdot \\ \cdot \\ C_d(w_m)\vec{g} - (\vec{f_m}) \end{bmatrix}$$

$\mathbf{x} = \mathbf{x}^+$
until $\|F(\mathbf{x}) - \mathbf{b}\| < \theta$
Once $\|F(\mathbf{x}) - \mathbf{b}\| < \theta$, we extract coefficients of the polynomial $g(x)$ from the vector $\mathbf{x}$.
We now have $g(x) =$GCD $(f_1, f_2, \ldots, f_m)$.
end of algorithm

The matrix is a block matrix, the non-zero blocks are Cauchy matrices. It contains only zero-blocks except for the first column and the diagonal blocks.

3. Calculation of GCD using Bézout matrices

We now present a different approach to computing the GCD of several real univariate polynomials using Bézoutian matrices (see [3], [8]). The size of this kind of matrix depends purely on the degree of one of the polynomials. It will be possible to determine the degree of GCD of a whole set of polynomials at once. Moreover, its coefficients will be computed at the same time. Let p and q be two polynomials,

$$p(x) = a_0x^k + a_1x^{k-1} + \ldots a_{k-1}x + a_k,$$
$$q(x) = b_0x^k + b_1x^{k-1} + \ldots b_{k-1}x + b_k$$

of degrees at most $k > 0$. If $\deg(p) >\deg(q)$ then some of the first coefficients of q equal zero.
The Bézout matrix associated to p and q (see [12]) is

$$B(p, q) = \begin{bmatrix} c_{1,1} & \cdots & c_{1,k} \\ \vdots & & \vdots \\ c_{k,1} & \cdots & c_{k,k} \end{bmatrix},$$

where the coefficients $c_{i,j}$ are defined by the relation

$$\frac{p(x)q(y) - p(y)q(x)}{x - y} = \sum_{i,j=1}^{k} c_{i,j} x^{i-1} y^{j-1}.$$

In the following, the procedure for computing the AGCD of m polynomials is presented. To achieve this, a set of polynomials $f_1, \ldots, f_m$ satisfying

$$k := n_1 = \deg(f_1) > \deg(f_i),\ i = 2, 3, \ldots, m$$

will be assumed. In contrast with Sylvester matrices, all the Bézout matrices $B(f_1, f_i)$, $i = 2, 3, \ldots, m$ are square and of the same dimension. Therefore the matrix

$$B_{f_1}(f_2, \ldots, f_m) = \begin{bmatrix} B(f_1, f_2) \\ B(f_1, f_3) \\ \vdots \\ B(f_1, f_m) \end{bmatrix}$$

can be constructed. Analogously to computation with Sylvester matrices, the degree of the AGCD equals $k - \operatorname{rank}(B_{f_1}(f_2, \ldots, f_m))$. Its coefficients can be computed by determining the linear combinations of column vectors, as described in the algorithm below (for details see [8]). The numerical realization of GCD will be again called AGCD.

Algorithm 3.1 (*AGCD for m polynomials.*)
Input: Real polynomials f_1, f_2,..., f_m of degrees n_1, n_2,..., n_m respectively. It is assumed that $k := n_1 > \max\{n_2, \ldots, n_m\}$.
Output: Polynomial g =AGCD$(f_1, f_2, \ldots, f_m)$.
begin
Determine the $d = k - \operatorname{rank}(B_{f_1}(f_2, \ldots, f_m))$.
Let $\mathbf{t}_1, \ldots, \mathbf{t}_k$ be column vectors of $B_{f_1}(f_2, \ldots, f_m) = [\mathbf{t}_1, \ldots, \mathbf{t}_k]$.
Construct $T_2 = [\mathbf{t}_k, \mathbf{t}_{k-1}, \ldots, \mathbf{t}_{d+1}]$ and $T_1 = [\mathbf{t}_d, \mathbf{t}_{d-1}, \ldots, \mathbf{t}_1]$.
Calculate QR decomposition of T_2, i.e. $T_2 = QR$, where $Q \in \mathbb{R}^{k \times k}$ is orthogonal and $R \in \mathbb{R}^{k \times (k-d)}$ is an upper triangular matrix.
We set $c := (R)^{-1}_{k-d,k-d}$ and compute $w_i^{d+1} = c\left(Q^T T_1\right)_{k-d,i}$, for $i = d, \ldots, 1$.
Setting $h_i := w_{d-i+1}^{d+1},\ i = 1, \ldots, d$ and $h_0 := 1$,
we finally have $g(x) = h_0 x^d + h_1 x^{d-1} + \ldots + h_{d-1} x + h_d = \mathrm{GCD}(f_1, \ldots, f_m)$.
end of algorithm

4. Numerical experiment

To compare the two presented algorithms, let us now have the following polynomials:

$$f_0 = (x-0.9)^5(x-0.8)^5(x-0.7)^5(x+0.3)^5(x+0.5)^5(x+0.7)^5,$$
$$f_1 = (x-2)^5(x-0.9)^5(x-0.8)^5(x+0.5)^5(x+2)^5,$$
$$f_2 = (x-3)^5(x-0.8)^5(x+0.5)^5(x+2)^5 \text{ and}$$
$$f_3 = (x-0.8)^4(x+0.5)^4.$$

It is easily seen, that $\mathrm{GCD}(f_0, \ldots, f_3) = f_3$. Accuracy of these computations is shown in Table 1. The errors made in determining the coefficients are about two orders of magnitude smaller in case of Algorithm 2.1 than in the case of Algorithm 3.1.

Coefficients	Error in coefficients	
GCD	Algorithm 2.1	Algorithm 3.1
1.0000	0.0000e+00	0.0000e+00
-1.2000	-9.1038e-15	1.8050e-12
-1.0600	-6.8834e-15	-7.6916e-13
1.3320	2.6645e-15	-2.6426e-12
0.5361	1.2212e-15	3.3151e-13
-0.5328	-2.6645e-15	1.3358e-12
-0.1696	-2.0539e-15	1.1419e-13
0.0768	-7.2164e-16	-2.3732e-13
0.0256	-3.8164e-16	-5.7697e-14

Table 1: Comparison of computational error in AGCD coefficients produced by Algorithm 2.1 and Algorithm 3.1.

Acknowledgements

This work was supported by the grant prvouk p47. The authors thank for this support.

References

[1] Barnett, S.: *Polynomials and linear control systems.* Marcel Dekker, INC., New York and Basel, 1983.

[2] Björk, Å.: *Numerical method for least square problems.* SIAM, Philadelphia, 1996.

[3] Bini, D. and Pan, V. Y.: *Polynomial and matrix computation, vol. 1 fundamental algorithms.* Birkhuser, 1994.

[4] Corless, R. M., Gianni, P. M., Trager, B. M., and Watt, S. M.: The singular value decomposition for polynomial systems. In: *Proc. ISSAC 95*, pp. 195–200. ACM Press, 1995.

[5] Eliaš, J.: *Problémy spojené s výpočtem největšího společného dělitele*. Bachelor thesis, Charles University, Faculty of Mathematics and Physics, 2009.

[6] Eliaš, J.: *Approximate polynomial greatest common divisor*. Master thesis, Charles University, Faculty of Mathematics and Physics, 2012.

[7] Diaz-Toca, G. M. and Gonzales-Vega, L.: Barnett's theorems about greatest common divisor of several univariate polynomials through Bézout-like matrices. J. Symbolic Computation **34**, (2002), 59–81
doi: 10.1006/jsco. 2002.0542

[8] Diaz-Toca, G. M. and Gonzales-Vega, L. : Computing greatest common divisors and square free decompositions through matrix method: The parametric and approximate cases. Linear Algebra Appl. **412** (2006), 222–246.

[9] Golub, G. H. and Van Loan, C. F.: *Matrix computations*. 3rd Ed. John Hopkins University Press, Baltimore, USA, 1996.

[10] Kaltofen, E., Yang, Z., and Zhi, L.: Structured low rank approximation of Sylvester matrix. Preprint, 2005.

[11] Li, T. Y. and Zeng, Z.: A rank-revealing method with updating, downdating and applications. SIAM J. Matrix Anal. Appl. **26** No. 4 (2005), 918–946.

[12] Pták, V.: Explicit expressions for Bézoutians. Linear Algebra Appl. **59** (1984), 43–54.

[13] Sun, D. and Zhi, L.: Structured low rank approximation of a Bézout matrix. In:*MM Research preprints*, pp. 207–218. KLMM, AMSS, Academia Sinica, Beijing 2006.

[14] Winkler, J. R. and Zítko, J.: Some questions associated with the calculation of the GCD of two univariate polynomials. In: *Winter School and SNA'07*, pp. 130–137. Ostrava, 2007.

[15] Winkler, J. R. and Allan, J. D.: Structured total least norm and approximate GCDs of inexact polynomials. Journal of Comp.and Appl. Math. **215** (2006), 1–13.

[16] Zeng, Z.: The approximate GCD od inexact polynomials, Part I: univariate algorithm. Preprint, 2004.

[17] Zítko, J. and Eliaš, J.: Application of the rank revealing algorithm for the calculation of the GCD. In: *Winter School and SNA'12*, pp. 175–180. Liberec, 2012.