

DISPLACEMENT STRUCTURE IN COMPUTING APPROXIMATE GCD OF UNIVARIATE POLYNOMIALS

LIHONG ZHI

*Mathematics Mechanization Research Center
Institute of Systems Science
Chinese Academy of Sciences
Beijing, China 100080
E-mail: lzhi@mmrc.iss.ac.cn*

We propose a fast algorithm for computing approximate GCD of univariate polynomials with coefficients that are given only to a finite accuracy. The algorithm is based on a stabilized version of the generalized Schur algorithm for Sylvester matrix and its embedding. All computations can be done in $O(n^2)$ operations, where n is the sum of the degrees of polynomials. The stability of the algorithm is also discussed.

1. Introduction

Let $f(x)$ and $g(x)$ be given polynomials represented as $f(x) = f_n x^n + f_{n-1} x^{n-1} + \cdots + f_1 x + f_0$, $g(x) = g_m x^m + g_{m-1} x^{m-1} + \cdots + g_1 x + g_0$, where $f_i, g_i \in \mathbb{R}$ and $\|f\|_2 = \|g\|_2 = 1$. Many papers have already discussed the approximate GCD problem [11][17][18][19][20][21][22][25]. There are many different definitions about approximate GCD. In the following text, we make use of the definition in [17]. For a given tolerance ϵ , we are going to find an approximate ϵ -GCD. In paper [12], we have already derived a backward stable method for computing the approximate GCD. The method is based on the classical QR factorization of the Sylvester matrix of f, g and their reversals. Utilizing the special structure of Sylvester matrix, we proposed a combined QR factoring algorithm using Givens rotations and Householder transformations. But the cost of the algorithm is still $O(n^3)$ ($n \geq m$).

Recently, various results on matrix with displacement structure have been reported in [2][3][4][5][6][7][14][16]. It is well-known that the Sylvester matrix is a quasi-Toeplitz matrix with displacement rank at most 2. An algorithm based on fast QR factorization was suggested in [9], but the sta-

bility was not guaranteed. The paper [6] derives a modified fast QR factorization for the matrices with shift structure (e.g., Toeplitz, quasi-Toeplitz). The algorithm is provably both fast and backward stable for solving linear systems of equations, the coefficients matrices of which are structured. This motivates us to extend the stabilized version of the generalized Schur algorithm for computing the approximate GCD efficiently. In the following sections, we first introduce the displacement structure of the Sylvester matrix and its embedding. We then show a fast algorithm for computing the approximate GCD, giving an example that illustrates a good behavior of the algorithm. The backward error and primitiveness test are discussed briefly. All algorithms to be presented in section 4 are also based on fast algorithms for structured matrix. We conclude with a short account on open problems about stability and structured perturbation.

2. Displacement Structure of Sylvester Matrix and its Embedding

The displacement of an $n \times n$ Hermitian matrix R was originally defined by Kailath, Kung, and Morf [15] as

$$\nabla R = R - Z_n R Z_n^T. \quad (2.1)$$

Here and hereafter, Z_i denotes the $i \times i$ lower shift matrix with ones on the first subdiagonal and zeros elsewhere; If ∇R has low rank $r (< n)$ independent of n , then R is said to be structured with respect to the displacement defined by (2.1) and r is referred to as the displacement rank of R .

The Sylvester matrix of $f(x), g(x)$ is:

$$S(f, g) = \begin{bmatrix} f_n & f_{n-1} & \cdots & f_1 & f_0 & & & & & & \\ & f_n & f_{n-1} & \cdots & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & & \ddots & \ddots & & & & \\ & & & f_n & f_{n-1} & \cdots & f_1 & f_0 & & & \\ g_m & g_{m-1} & \cdots & g_1 & g_0 & & & & & & \\ & g_m & g_{m-1} & & \ddots & \ddots & & & & & \\ & & \ddots & \ddots & & \ddots & \ddots & & & & \\ & & & g_m & g_{m-1} & \cdots & g_1 & g_0 & & & \end{bmatrix}. \quad (2.2)$$

Theorem 2.3. *The Sylvester matrix S is a quasi-Toeplitz matrix, i.e., $S - Z_{n+m} S Z_{n+m}^T$ has displacement rank at most 2.*

Proof. It is trivial to note that $S - Z_{m+n}SZ_{m+n}^T = [e_1, e_{n+1}][u, v]^T$ with
 $u = [f_n, \dots, f_0, 0, \dots, 0]^T$, $v = [g_m, \dots, g_1, g_0 - f_n, -f_{n-1}, \dots, -f_1]^T$ (2.4)

Here and hereafter, e_i denotes the i -th column of the $(n+m) \times (n+m)$ identity matrix. \square

In order to compute the QR factorization of the structured matrix S , we need to apply the generalized Schur algorithm to the following properly defined embedding matrix M_5 of the Sylvester matrix.

Theorem 2.5. *The $2(n+m) \times 2(n+m)$ augmented matrix*

$$M_5 = \begin{bmatrix} S^T S & S^T \\ S & 0 \end{bmatrix} \quad (2.6)$$

has displacement rank at most 5.

Proof. We can verify that

$$M_5 - FM_5F^T = GJG^T, \quad (2.7)$$

where $F = Z_{n+m} \oplus Z_{n+m}$, $J = (1 \oplus 1 \oplus -1 \oplus -1 \oplus -1)$ is a signature matrix and

$$G = \begin{bmatrix} x_0^T & y_0^T & x_1^T & y_1^T & 0 \\ e_1 & e_{m+1} & e_{m+1} & 0 & e_1 \end{bmatrix}. \quad (2.8)$$

Here,

$$\begin{aligned} x_0 &= \text{Row}(S, 1), \\ x_1 &= \text{Row}(S, m)Z_{n+m}^T, \\ y_0 &= \text{Row}(S, m+1), \\ y_1 &= \text{Row}(S, n+m)Z_{n+m}^T, \end{aligned} \quad \square$$

It is clear that the generator G can be computed directly from Sylvester matrix S instead of embedding matrix M_5 . As in [6], after first $n+m$ steps of the generalized Schur algorithm applied to (F, G) , we have the partial triangulation as:

$$M_5 = \begin{bmatrix} L \\ U \end{bmatrix} D^{-1} [L^T \ U^T] + \begin{bmatrix} 0 & 0 \\ 0 & -I \end{bmatrix}, \quad (2.9)$$

where L is $(n+m) \times (n+m)$ lower-triangular, and U is an $(m+n) \times (m+n)$ matrix, D is an $2(m+n) \times 2(m+n)$ diagonal matrix. By equating terms on both sides of the above equality we conclude that

$$S^T S = R^T R, \quad S = QR, \quad QQ^T = I, \quad (2.10)$$

where $Q = UD^{-1/2}$, $R = (LD^{-1/2})^T$. The cost of the algorithm is $O(n^2)$. Notice that Q may not be orthogonal matrix even in well-conditioned case[6].

3. M_5 for computing the GCD of univariate approximate polynomials

For exact (or infinite precision) computation, it is well-known that $S(f, g)$ is of rank $m + n - r$ if and only if $\deg(\text{GCD}(f, g)) = r$. Let R be an upper triangular factor of S , then the last non-zero row of R gives the coefficients of $\text{GCD}(f, g)$. See [24] for a proof.

For polynomials with coefficients that are given only to a finite accuracy, the above statement must be used carefully. A backward stable method such as QR factoring using Givens rotations or Householder transformations may not result in an R whose numeric rank equals to the degree of a numeric(approximate) GCD. Although QR factoring with column pivoting can possibly reveal the rank of a matrix, pivoting is forbidden in GCD computation. However, the paper[12] has proved that if all common roots of f, g lie inside the unit circle, the computed R using QR factoring without pivoting gives the coefficients of an approximate GCD; Otherwise, the last “non-zero” row of R will only be a factor of the approximate GCD which includes all common roots inside or close to the unit circle. Other common roots outside of the unit circle can be recovered from the QR factoring of the Sylvester matrix of the reversals of f, g . See[12] for detail.

In[9], a fast QR factorization combined with efficient rank estimator were applied to compute the approximate GCD of univariate polynomials . The method has two unsolved issues. One is that stability of the algorithm was unknown. The other is that the rank estimator has difficulty in deciding the rank in presence of perturbations. Moreover, according to[12], even we can estimate the rank correctly, the computed R may still have different numeric rank as we estimated.

The second issue has been discussed in[12] extensively. So now let us concentrate on the stability problem. Chandrasekaran and Sayed derived a stable and fast solver for non-symmetric systems of linear equations with shift structured coefficient matrices. Can it be extended to solve the approximate GCD problem? We have derived an explicit formula for the generator of M_5 in previous section, now let us see if the stability problem of fast algorithm can also be solved for the approximate GCD computation.

There are two important properties follow from Householder QR or

Givens QR [13] [23]:

$$\|S - QR\|_2 = O(u\|S\|_2), \quad (3.1)$$

$$\|Q^{-1}S - R\|_2 = \|Q^T S - R\|_2 = O(u\|S\|_2) \quad (3.2)$$

where u is the machine precision. The first property shows that the GCD of f, g can be written approximately as a linear combination of the polynomials formed by the rows of R . The second condition tells that any polynomials formed from the rows of R can be written approximately as a polynomial combination of f, g . Now let us check these properties for the R computed by fast QR factoring of M_5 .

Suppose $2(n + m)$ steps of the generalized Schur algorithm can be completed to give

$$\hat{M} = \begin{bmatrix} \hat{R}^T & 0 \\ \hat{Q} & \Delta \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{Q}^T \\ 0 & \Delta^T \end{bmatrix}. \quad (3.3)$$

Since the generalized Schur algorithm is backward stable, The first property can be easily derived. In the well-conditioned case, although \hat{Q} is not orthogonal, we still have that $\Delta^{-1}\hat{Q}$ is numerically orthogonal and $\|\Delta^{-1}\|_2^2$ is bounded by $1/5$. So the second property can be derived as $\|(\Delta^{-1}\hat{Q})^T \Delta^{-1}S - \hat{R}\|_2 = O(u\|S\|_2)$. In the ill-conditioned case, we can not guarantee that Δ is well-conditioned and $\Delta^{-1}\hat{Q}$ is numerically orthogonal. But if we restrict perturbation β which introduced in [6] to be 0, the last $m + n$ negative steps do not fail implies $\lambda_{\min}(\hat{Q}^T \hat{Q}) > u$. So that we have $\|\hat{Q}^{-1}S - \hat{R}\|_2 = O(\sqrt{u}\|S\|_2)$. Though the second property is not guaranteed, we may still obtain useful information from \hat{R} suppose \sqrt{u} is of the size of tolerance.

In practically, we can always perturb the polynomials f, g within tolerance of coefficients to obtain a well-conditioned \hat{Q} even when the perturbed S is still ill-conditioned. Since $\|\hat{Q}\hat{Q}^T - \Delta\Delta^T\|_2 = O(u)$, the condition number of \hat{Q} is very close to the condition number of Δ . Δ is a triangular matrix, its condition number can be estimated efficiently. In the case \hat{Q} or Δ is well-conditioned, we can guarantee the second property.

Example 3.4.

$$\begin{aligned} f := & 0.02077971692 x^{13} + 0.09350872615 x^{12} - 0.2246806892 x^{11} - 0.4552056739 x^{10} \\ & - 0.2659154401 x^9 + 0.2584477292 x^8 + 0.5175448246 x^7 + 0.1149378092 x^6 \\ & - 0.3357223016 x^5 - 0.3237090279 x^4 + 0.2444863572 x^3 + 0.1655883694 x^2 \\ & - 0.09935302169 x + 0.000000006520463574 \end{aligned}$$

$$\begin{aligned}
g := & -0.03804013712 x^{11} - 0.1616705828 x^{10} + 0.3870583952 x^9 + 0.2277653210 x^8 \\
& - 0.2833990216 x^7 + 0.2268143176 x^6 + 0.5102133391 x^5 - 0.1183999265 x^4 \\
& - 0.07227626008 x^3 + 0.5663225421 x^2 + 0.03233411610 x - 0.1940046900
\end{aligned}$$

For Digits= 15, in Maple 8, $\gcd(f, g) = 1$ (Note that if we choose Digits= 10 in Maple, we will get $\gcd(f, g) = x^2 + 0.4623160489 x - 0.5507384540$. We also tried SNAP package in Maple 8, both the QuasiGCD and EpsilonGCD fail for this example). Now let us apply generalized Schur algorithm to the rank 5 generator G and J , the algorithm fails at step 22. It is interesting to check that the 22-th row of computed R gives a polynomial very close to following c_1, c_2 . Since it is not clear how the algorithm works in such case, we would rather to follow [6] to introduce a small perturbation to the matrix M_5 in order to avoiding the early breakdown of the algorithm. We add 10^{-8} to f, g to get \hat{f}, \hat{g} , then both the positive and negative steps of generalized Schur algorithms succeed. We have

$$\|S(\hat{f}, \hat{g}) - \hat{Q}\hat{R}\|_2 = 0.400670 10^{-13}.$$

The condition number of \hat{Q} and Δ is 30.5154. The orthogonality of $\Delta^{-1}\hat{Q}$ can be verified as

$$\|\Delta^{-1}\hat{Q}(\Delta^{-1}\hat{Q})^T - I\|_2 = 0.483682 10^{-10}.$$

Consequently,

$$\|(\Delta^{-1}\hat{Q})^T \Delta^{-1}S - \hat{R}\|_2 = 0.123234 10^{-11}.$$

So the two properties can be achieved even in the ill-conditioned case. The norms of the last two rows of \hat{Q} are less than 10^{-8} . Forming the polynomial from the last third row, we will obtain a monic factor of GCD of f, g as

$$c_1 := x^2 + 0.46231633 x - 0.5507386676$$

Now, if we apply the classical QR factoring to $S(f, g)$, the norms of the last two rows are less than 10^{-10} . The last third row gives

$$c_2 := x^2 + 0.4623160154 x - 0.5507384496$$

It is clear c_1, c_2 are very close to each other.

4. Backward Error Analysis

Although it has been proved that the modified generalized Schur algorithm in[6] is backward stable and fast for solving system of linear equations, it is still not fully proved that the fast QR factorization for the approximate GCD computation is backward stable. So it is important to check the backward error after we obtain a candidate for the approximate GCD. There are two main steps for checking backward error: 1) Approximate polynomial division 2) Test for primitiveness of cofactors.

Approximate Polynomial Division

Let $f(x), c(x)$ be given polynomials have degree $n > n_1$ respectively, suppose $\|f\|_2 = \|c\|_2 = 1$,

$$\begin{aligned} f(x) &= f_n x^n + f_{n-1} x^{n-1} + \cdots + f_1 x + f_0, \\ c(x) &= c_{n_1} x^{n_1} + c_{n_1-1} x^{n_1-1} + \cdots + c_1 x + c_0. \end{aligned} \quad (4.1)$$

We are trying to find the cofactor h which minimizes $\|f - ch\|_2$. There are many methods to solve this least square problem. We present one method based on the displacement structure of the following matrix. Define A as

$$A = \begin{bmatrix} c_{n_1} & & & & & & & \\ \vdots & \ddots & & & & & & \\ \vdots & & c_{n_1} & & & & & \\ c_0 & & & & & & & \\ & & & \ddots & & & & \\ & & & & \vdots & & & \\ & & & & & c_0 & & \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n-n_1+1)}, \quad (4.2)$$

we can write the minimization problem in matrix form as $\min \|AV_h - V_f\|_2$, where V_f and V_h are the coefficient vectors of polynomials f and h respectively. Clearly, A is a generalized Toeplitz matrix. The minimization problem can be solved by normal equations[13].

Theorem 4.3. *Let A be of full rank, then $T = A^T A$ is symmetric positive definite Toeplitz matrix. The difference $T - Z_{n-n_1+1} T Z_{n-n_1+1}^T$ has displacement rank at most 2:*

$$T - Z_{n-n_1+1} T Z_{n-n_1+1}^T = \begin{bmatrix} 1 & 0 \\ t_1 & t_1 \\ \vdots & \vdots \\ t_{n-n_1} & t_{n-n_1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_1 & t_1 \\ \vdots & \vdots \\ t_{n-n_1} & t_{n-n_1} \end{bmatrix}^T = G J G^T$$

$$[1, t_1, \dots, t_{n-n_1}]^T = A^T \text{Column}(A, 1).$$

Apply the modified generalized Schur algorithm to the generator G , we obtain the Cholesky factorization using $O(n^2)$ operations:

$$T = A^T A = R^T R. \quad (4.5)$$

Solve $R^T y = AV_f$ and $RV_h = y$, we can find the cofactor h .

For example 1, the above algorithm finds the backward error $\|f - c_1 \frac{f}{c_1}\|_2 = 0.5566147 \cdot 10^{-7}$, $\|g - c_1 \frac{g}{c_1}\|_2 = 0.1643287 \cdot 10^{-6}$.

Test for Primitiveness

After dividing out the common divisor $c(x)$, it is necessary to check the primitiveness of the polynomials $f/c, g/c$ to guarantee that the computed the approximate GCD $c(x)$ is of the highest degree. As stated in [1], it is equivalent to computing condition number of Sylvester matrix $S(f/c, g/c)$ by solving two systems of linear equations with S as coefficient matrix, i.e,

$$S^T x = b, \quad b \in \mathbb{R}^{m+n} \quad (4.6)$$

where b is the coefficient vector of the polynomial $b(x) = 1$ or $b(x) = x^{m+n-1}$ respectively. S^T is also a quasi-Toeplitz matrix. We can directly apply the fast and stable solver to the following embedding of Sylvester matrix,

$$M_4 = \begin{bmatrix} SS^T & S \\ S^T & 0 \end{bmatrix}. \quad (4.7)$$

Theorem 4.8. M_4 has displacement rank at most 4, i.e.,

$$R = M_4 - FM_4F^T = GJG^T, \quad (4.9)$$

where $F = Z_{n+m} \oplus Z_{n+m}$ and $J = (1 \oplus 1 \oplus -1 \oplus -1)$ The generator can be written as

$$G = [x_1, x_2, y_1, y_2], \quad (4.10)$$

where

$$\begin{aligned} x_1 &= \text{Column}(R, 1), \quad \text{except } x_1[m+1] = 0, \\ y_1 &= \text{Column}(R, 1), \quad \text{except } y_1[1] = 0, y_1[m+1] = 0, \\ x_2 &= \text{Column}(R, m+1), \quad \text{except } x_2[m+1] = 1/2, \\ y_2 &= \text{Column}(R, m+1), \quad \text{except } y_2[m+1] = -1/2, \end{aligned}$$

$x_k[i]$ and $y_k[i]$ mean the i th entry in the vectors x_k and y_k for $k = 1, 2$. Another different rank 4 generator was given by Claude-Pierre Jeannerod in [8]. It is still unknown which generator is better for computing.

Continuing for example 1, we will find the condition number of $S(f/c_1, g/c_1)$ is of order 10^9 . This means that $f/c_1, g/c_1$ are not prime to each other. The classical QR factoring with pivoting running in Matlab tells us the numeric rank of $S(f, g)$ is 21 not 22 as showed by the above QR factoring. Actually, the missing common root of f, g is -5.787684 . For finding this common root, it is necessary to reapply the fast QR factoring algorithm to the Sylvester matrix of the reversals of $f/c_1, g/c_1$. See[12] for details. The approximate GCD c of f and g is:

$$x^3 + 6.250020509 x^2 + 2.125011483 x - 3.187512489$$

The backward errors are:

$$\|f - c \frac{f}{c}\|_2 = 0.119003 10^{-6}, \|g - c \frac{g}{c}\|_2 = 0.285738 10^{-6}.$$

5. Concluding Remarks

This paper proposes a new fast algorithm for computing an approximate GCD of univariate polynomials. The algorithm has been implemented in Maple 8. Some experimental results are included. The work reported here is just a first attempt to use displacement structure on the approximate GCD computations. There are many interesting and important aspects are not explored yet. One of the open problems is whether the fast QR factorization is backward stable for computing the approximate GCD. Another open problem is to find the structured perturbation(of Sylvester type) to avoid the early breakdown of the fast algorithm. We will pursue these problems in a future paper.

Acknowledgments

Lihong Zhi would like to thank Prof. Matu-Tarow Noda, Prof. Robert M. Corless, Prof. Stephen M. Watt, Prof. George Labahn and Dr. Claude-Pierre Jeannerod for useful discussions.

References

1. Bernhard Beckermann and George Labahn, "When are two polynomials relatively prime?," *Journal of Symbolic Computation*, vol. 26, no. 6, pp. 677–689, 1998, Special issue of the JSC on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

2. A.W. Bojanczyk, R.P. Brent and F.De Hoog, "QR factorization of Toeplitz matrices," *Numerische Mathematik*, vol 49, pp. 81-94, 1986.
3. A.W. Bojanczyk, R.P. Brent, F.De Hoog and D.R. Sweet, "On the stability of the Bareiss and related Toeplitz factorization algorithms," *SIAM J. Matrix Anal. Appl.*, vol 16, pp. 40-57, 1995.
4. Richard P. Brent, Adam W. Bojanczyk and Frank R. de Hoog, "Stability analysis of a general Toeplitz systems solver," *Numerical Algorithms*, vol. 10, pp. 225-244, 1995.
5. S. Chandrasekaran and A. H. Sayed, "Stailizing the generalized Schur algorithm," *SIAM J. Matrix Anal. Appl.*, vol 17, pp. 950-983, 1996.
6. S. Chandrasekaran and A. H. Sayed, "A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations," *SIMAX*, vol. 19, no. 1, pp. 107-139, 1998.
7. J. Chun, "Fast array algorithms for structured matrices ", PhD thesis, Stanford university, 1989.
8. Claude-Pierre Jeannerod and Lihong Zhi, "Computing low rank generators of Sylvester matrix embeddings," 2002.
9. Christopher J. Zarowski, Xiaoyan Ma, and Frederick W. Fairman, "A QR-factorization method for computing the greatest common divisor of polynomials with real-valued coefficients," *IEEE Trans. Signal Processing*, vol. 48, pp. 3042-3051, 2000.
10. James W. Demmel, "On condition numbers and the distance to the nearest ill-posed problem," *Numer. Math.*, vol. 51, pp. 251-289, 1987.
11. Robert M. Corless, Patrizia M. Gianni, Barry M. Trager, and Stephen M. Watt, "The Singular Value Decomposition for polynomial systems," in *International Symposium on Symbolic and Algebraic Computation*, A.H.M. Levelt, Ed., Montréal, Canada, 1995, pp. 195-207, ACM.
12. Robert M. Corless, Stephen M. Watt, and Lihong Zhi "QR factoring to compute the GCD of univariate approximate polynomials," submitted, 2002.
13. Gene H. Golub and Charles F. Van Loan, "Matrix computations," Johns Hopkins, 3rd edition, 1996.
14. M.Gu, "Stable and efficient algorithms for structured systems of linear equations," in *SIAM J. Matrix Anal. Appl.*, vol 19, pp. 279-306, 1997.
15. Thomas Kailath, S.Y.Kung, and M.Morf, "Displacement ranks of a matrix," in *Bull. Amer. Math. Soc.*, vol. 1, pp. 769-773, 1979.
16. Thomas Kailath and Ali H. Sayed, "Displacement structure: theory and applications," in *SIAM Review*, vol. 37, no. 3, pp. 297-386, 1995.
17. N. Karmarkar and Y.N. Lakshman, "Approximate polynomial greatest common divisors and nearest singular polynomials," in *International Symposium on Symbolic and Algebraic Computation*, Zürich, Switzerland, 1996, pp. 35-42, ACM.
18. N. K. Karmarkar and Lakshman Y. N., "On approximate GCDs of univariate polynomials," *Journal of Symbolic Computation*, vol. 26, no. 6, pp. 653-666, 1998, Special issue of the JSC on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.
19. Matu-Tarow Noda and Tateaki Sasaki, "Approximate GCD and its appli-

- cation to ill-conditioned algebraic equations” *Journal of Computational and Applied Mathematics*, vol. 38, pp. 335-351, 1991.
20. A. Schönhage, “The fundamental theorem of algebra in terms of computational complexity,” Tech. Rep., University of Tübingen, Math. Dept., 1982.
 21. A. Schönhage, “Quasi-gcd computations,” *Journal of Complexity*, vol. 1, pp. 118-137, 1985.
 22. Lihong Zhi and Matu-Tarow Noda, “Approximate gcd of multivariate polynomials,” *Proc. ASCM*, pp. 9-18, 2000.
 23. Nicholas J. Higham, *Accuracy and Stability of Numerical Algorithms*, 1996.
 24. M. A. Laidacker, “Another theorem relating sylvester’s matrix and the greatest common divisor,” *Mathematics Magazine*, vol. 42, pp. 126-128, 1969.
 25. Hans J. Stetter, “The nearest polynomial with a given zero, and similar problems,” *SIGSAM Bulletin: Communications on Computer Algebra*, vol. 33, no. 4, pp. 2-4, 1999.
 26. Robert M. Corless, Stephen M. Watt, and Lihong Zhi, “A report on the SNAP package in Maple,” Tech. Rep., ORCCA, in preparation 2002.