Int. J. Appl. Math. Comput. Sci., 2007, Vol. 17, No. 4, 463–470
DOI: 10.2478/v10006-007-0038-8

ON THE COMPUTATION OF THE GCD OF 2-D POLYNOMIALS

PANAGIOTIS TZEKIS [*], NICHOLAS P. KARAMPETAKIS [**], HARALAMBOS K. TERZIDIS [*]

[*] Department of Mathematics, School of Sciences
Technological Educational Institution of Thessaloniki
P.O. Box 14561, GR-541 01 Thessaloniki, Greece
e-mail: tzekis@tellas.gr

[**] Department of Mathematics, Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
e-mail: karampet@math.auth.gr

The main contribution of this work is to provide an algorithm for the computation of the GCD of 2-D polynomials, based on DFT techniques. The whole theory is implemented via illustrative examples.

Keywords: greatest common divisor, discrete Fourier transform, two-variable polynomial.

1. Introduction

An interesting problem of algebraic computation is the computation of the greatest common divisor (GCD) of a set of polynomials. The GCD is usually linked with the characterisation of zeros of a polynomial matrix description of a system. The problem of finding the GCD of a set of n polynomials on $\mathbb{R}[x]$ of a maximal degree q is a classical problem that has been considered before (Karcanias *et al.*, 2004). Numerical methods for the GCD (Karcanias *et al.,* 1994; Mitrouli *el al.,* 1993) were also developed. Due to the difficulty in finding the exact GCD of a set of polynomials, approximate algorithms were also developed (Noda *et al.*, 1991). A comparison of algorithms for the calculation of the GCD of polynomials is given in (Pace *et al.*, 1973).

The main disadvantage of many algorithms is their complexity. In order to overcome these difficulties, we may use other techniques such as interpolation methods. For example, Schuster *et al.* (1992) used interpolation techniques in order to find the inverse of a polynomial matrix. The speed of interpolation algorithms can be increased by using Discrete Fourier Transform (DFT) techniques or, better, Fast Fourier Transform (FFT) techniques. Some of the advantages of DFT based algorithms are that there are very efficient algorithms available in both software and hardware and that they are well suited for parallel environments (through symmetric multiprocessing or other techniques). Karampetakis *et al.* (2005) used DFT techniques to compute the minimal polynomial of a polynomial matrix, and Paccagnella *et al.* (1976) used FFT methods for the computation of the determinant of a polynomial matrix.

Here we provide an algorithm for the computation of GCD of 2-D polynomials based on DFT techniques. The proposed algorithm is illustrated via examples.

2. Computation of the GCD of Bivariate Polynomials via Interpolation

Suppose that we have n polynomials $p_i(x,y) \in \mathbb{R}[x,y]$, $i = 0,1,\ldots,n-1$. These polynomials can be rewritten as

$$\begin{aligned} p_i\left(x,y\right) &= \overbrace{p_{i,x}(x)g_x(x)}^{p'_{i,x}(x)}\overbrace{p_{i,x,y}(x,y)g_{x,y}(x,y)}^{p'_{i,x,y}(x,y)}\overbrace{p_{i,y}(y)g_y(y)}^{p'_{i,y}(y)} \\ &= \sum_{m=0}^{M_1}\sum_{j=0}^{M_2} p_{i,m,j}x^m y^j \in \mathbb{R}[x,y], \\ &\qquad\qquad i = 0,1,\ldots,n-1, \quad (1) \end{aligned}$$

where M_1 (resp. M_2) is the greatest power of x (resp. y) in $p_i\left(x,y\right)$, $p_{i,x}(x) \in \mathbb{R}[x]$ are relatively coprime, $p_{i,y}(y) \in \mathbb{R}[y]$ are relatively coprime, $p_{i,x,y}(x,y) \in \mathbb{R}[x,y]$ are relatively factor coprime with no factors only of x or y, i.e., there are no common factors of $p_{i,x,y}(x,y)$, $g_x(x)$ is the GCD of $p'_{i,x}(x)$, $g_y(y)$ is the GCD of $p'_{i,y}(y)$ and $g_{x,y}(x,y)$ is the GCD of $p'_{i,x,y}(x,y)$. The greatest powers of the variables x, y in the CGD of $p(x,y)$ are

$$\begin{aligned}\deg_x(p(x,y)) &= b_1\\ &:= \left(\le \min_{i=0,1,\dots,n-1}\{\deg_x(p_i(x,y))\}\right),\\ \deg_y(p(x,y)) &= b_2\\ &:= \left(\le \min_{i=0,1,\dots,n-1}\{\deg_y(p_i(x,y))\}\right)\end{aligned} \tag{2}$$

for the GCD. Thus,

$$p(x,y) = \sum_{k_1=0}^{b_1}\sum_{k_2=0}^{b_2}(p_{k_1,k_2})\left(x^{k_1}y^{k_2}\right).$$

The computation of the GCD is essentially a univariate problem, since we can compute the multivariate GCD by interpolation. The number of the interpolation points (x_i, y_i) that we need in order to evaluate $p(x,y)$ or, equivalently, its coefficients p_{k_1,k_2} is equal to

$$R_1 = (b_1+1)(b_2+1).$$

A naive approach to evaluate the GCD using interpolation techniques is to make the following steps:

(a) Evaluate the GCDs $\tilde{p}_k(x,y_k)$ of the univariate polynomials $p_i(x,y_k)$ using known techniques (Karcanias *et al.,* 2004; Karcanias *et al.,* 1994; Mitrouli *et al.,* 1993; Noda *et al.,* 1991; Pace *et al.,* 1973), where $y_k, k=0,1,\dots,b_2$ are b_2+1 distinct interpolation points.

(b) Apply the values of b_1+1 distinct interpolation points $x_j, j = 0,1,\dots,b_1$ to the polynomials $\tilde{p}_k(x,y_k)$ in order to get the values of the approximated GCD $\tilde{p}(x,y)$ at (x_j,y_k), i.e.,

$$\tilde{p}(x_j,y_k) = \tilde{p}_k(x_j,y_k), \quad j=0,1,\dots,b_1.$$

(c) Find the 2-D polynomial $\tilde{p}(x,y)$ that passes from R_1 interpolation points $(x_j, y_k, \tilde{p}_k(x_j,y_k))$. Here $\tilde{p}(x,y)$ will be the evaluation of $p(x,y)$ with interpolation.

The main drawbacks of the above approach that we have to overcome are twofold:

(a) There is a case where the polynomials $p_{i,x,y}(x,y)$ might not be zero coprime, i.e., there exist points (x_k,y_k) such that $p_{i,x,y}(x_k,y_k)=0$ for all i. Therefore, for those values of y_k, the GCDs $\tilde{p}_k(x,y_k)$ of the univariate polynomials $p_i(x,y_k)$ will have an extra factor of $(x-x_k)$, since all the polynomials $p_i(x,y_k)$ will vanish for $x=x_k$. Thus, the resulting univariate GCD polynomials $\tilde{p}_k(x,y_k)$ will have extra factors of x, except those of $g_x(x)g_{x,y}(x,y_k)$ that will come from the factors of $p_{i,x,y}(x,y)$, see Example 1.

(b) Since the evaluated univariate polynomials $\tilde{p}_k(x,y_k)$ are monic, we lose information from the factors $p'_{i,y}(y)$, and thus the evaluated GCD of $p(x,y)$ will have the form $p(x,y) = g_x(x)g_{x,y}(x,y)$ instead of $p(x,y) = g_x(x)g_{x,y}(x,y)g_y(y)$, i.e., the part $g_y(y)$ will be missing, see Example 2.

A combination of the above two problems may also exist. In the sequel we give two examples in order to show how the above drawbacks may guide us to unexpected results.

Example 1. Consider the polynomials

$$p_1(x,y) = (x+y)(x+1),$$

$$p_2(x,y) = (xy+1)(x+1).$$

Suppose that we are interested in evaluating the GCD of $p_1(x,y)$ and $p_2(x,y)$ which is clearly $p(x,y) = x+1$.

Step 1. Set

$$\begin{aligned}b_1 &= \min\big\{\deg_x[(x+y)(x+1)],\\ &\quad \deg_x[(xy+1)(x+1)]\big\}\\ &= 2,\\ b_2 &= \min\big\{\deg_y[(x+y)(x+1)],\\ &\quad \deg_y[(xy+1)(x+1)]\big\}\\ &= 1.\end{aligned}$$

Step 2. Take

$$R_1 = \prod_{i=1}^{2}(b_i+1) = (2+1)(1+1) = 6$$

interpolations points: $(-1,1)$, $(-1,-1)$, $(0,1)$, $(0,-1)$, $(1,1)$, $(1,-1)$.

Step 3. Determine the GCDs $\tilde{p}_k(x,y_k)$ of the polynomials $p_1(x,y_k)$, $p_2(x,y_k)$, $k=0,1$, where $y_0 = -1$, $y_1 = 1$:

$$\begin{aligned}\tilde{p}_0(x,-1) &= (x+1)(x-1),\\ \tilde{p}_1(x,1) &= (x+1)(x+1).\end{aligned}$$

However, we know that the GCD of the polynomials $p_1(x,y)$, $p_2(x,y)$ is $x+1$. This happens since the extra factors $(x+y)$ and $(xy+1)$ of $p_1(x,y)$ and $p_2(x,y)$, respectively, are not zero coprime. Note that both $(x+y)$

and $(xy+1)$ vanish at the points $(-1,1)$ and $(1,-1)$, and thus at $y_0 = 1$ (resp. $y_1 = -1$) we have the extra factor $x - x_0 = x + 1$ (resp. $x - x_1 = x - 1$). Thus, if we continue the algorithm by taking the values of $\tilde{p}_0(x,-1), \tilde{p}_1(x,1)$ at $x_j = -1, 0, 1$, i.e.,

$$\begin{aligned} &\tilde{p}_0(-1,-1) = 0, && \tilde{p}_1(-1,1) = 0,\\ &\tilde{p}_0(0,-1) = -1, && \tilde{p}_1(0,1) = 1,\\ &\tilde{p}_0(1,-1) = 0, && \tilde{p}_1(1,1) = 4, \end{aligned}$$

then the polynomial

$$\begin{aligned} \tilde{p}(x,y) = {} & \tilde{p}_{0,0} + \tilde{p}_{0,1}x + \tilde{p}_{1,0}y + \tilde{p}_{1,1}xy \\ & + \tilde{p}_{2,0}x^2 + \tilde{p}_{2,1}x^2y \end{aligned}$$

that will pass from the above interpolation points will satisfy the following equations:

$$\begin{aligned} \tilde{p}(-1,-1) &= \tilde{p}_{0,0} - \tilde{p}_{0,1} - \tilde{p}_{1,0} + \tilde{p}_{1,1} + \tilde{p}_{2,0} - \tilde{p}_{2,1} \\ &= 0, \\ \tilde{p}(0,-1) &= \tilde{p}_{0,0} - \tilde{p}_{1,0} = -1, \\ \tilde{p}(1,-1) &= \tilde{p}_{0,0} + \tilde{p}_{0,1} - \tilde{p}_{1,0} - \tilde{p}_{1,1} + \tilde{p}_{2,0} - \tilde{p}_{2,1} \\ &= 0, \\ \tilde{p}(-1,1) &= \tilde{p}_{0,0} - \tilde{p}_{0,1} + \tilde{p}_{1,0} - \tilde{p}_{1,1} + \tilde{p}_{2,0} + \tilde{p}_{2,1} \\ &= 0, \\ \tilde{p}(0,1) &= \tilde{p}_{0,0} + \tilde{p}_{1,0} = 1, \\ \tilde{p}(1,1) &= \tilde{p}_{0,0} + \tilde{p}_{0,1} + \tilde{p}_{1,0} + \tilde{p}_{1,1} + \tilde{p}_{2,0} + \tilde{p}_{2,1} \\ &= 4, \end{aligned}$$

and thus

$$\tilde{p}_{0,0} = 0,\ \tilde{p}_{1,0} = \tilde{p}_{0,1} = \tilde{p}_{1,1} = \tilde{p}_{2,0} = 1,\ \tilde{p}_{2,1} = 0,$$

or otherwise

$$\tilde{p}(x,y) = x + y + xy + x^2 = (x+1)(x+y),$$

which is different from the real GCD $p(x,y) = x + 1$. ♦

Example 2. Consider two polynomials:

$$\begin{aligned} p_1(x,y) &= (x+1)(y+2), \\ p_2(x,y) &= (x+2y)(y+2). \end{aligned}$$

Suppose that we are interested in evaluating the GCD of $p_1(x,y)$ and $p_2(x,y)$.

Step 1. Let

$$\begin{aligned} b_1 &= \min\big\{ \deg_x[(x+1)(y+2)], \\ &\qquad \deg_x[(x+2y)(y+2)] \big\} \\ &= 1, \\ b_2 &= \min\big\{ \deg_y[(x+1)(y+2)], \\ &\qquad \deg_y[(x+2y)(y+2)] \big\} \\ &= 1. \end{aligned}$$

Step 2 Take

$$R_1 = \prod_{i=1}^{2}(b_i+1) = (1+1)(1+1) = 4$$

interpolations points: $(1,1)$, $(1,0)$, $(-1,1)$, $(-1,0)$.

Step 3. Determine the GCDs $\tilde{p}_k(x,y_k)$ of the polynomials $p_1(x,y_k)$, $p_2(x,y_k)$, $k = 0,1$, where $y_0 = 0, y_1 = 1$:

$$\tilde{p}_0(x,0) = 1, \quad \tilde{p}_1(x,1) = 1.$$

Since the polynomials $\tilde{p}_1(x,y_k)$, $\tilde{p}_2(x,y_k)$ are monic, they do not include information about the factor $y+1$ that is included in the polynomials $p_1(x,y)$ and $p_2(x,y)$. Thus, if we continue the algorithm by taking the values of $\tilde{p}_0(x,0)$ and $\tilde{p}_1(x,1)$ at $x_j = 0,1$, i.e.,

$$\begin{aligned} &\tilde{p}_0(0,0) = 1, && \tilde{p}_1(0,1) = 1,\\ &\tilde{p}_0(1,0) = 1, && \tilde{p}_1(1,1) = 1, \end{aligned}$$

then the polynomial

$$\tilde{p}(x,y) = \tilde{p}_{0,0} + \tilde{p}_{0,1}x + \tilde{p}_{1,0}y + \tilde{p}_{1,1}xy$$

obtained from the above interpolation points will satisfy the following equations:

$$\begin{aligned} \tilde{p}(0,0) &= \tilde{p}_{0,0} = 1, \\ \tilde{p}(1,0) &= \tilde{p}_{0,0} + \tilde{p}_{0,1} = 1, \\ \tilde{p}(0,1) &= \tilde{p}_{0,0} + \tilde{p}_{1,0} = 1, \\ \tilde{p}(1,1) &= \tilde{p}_{0,0} + \tilde{p}_{0,1} + \tilde{p}_{1,0} + \tilde{p}_{1,1} = 1, \end{aligned}$$

and thus

$$\tilde{p}_{0,0} = 1,\ \tilde{p}_{1,0} = 0,\ \tilde{p}_{0,1} = 0,\ \tilde{p}_{1,1} = 0,$$

or otherwise

$$\tilde{p}(x,y) = 1,$$

which is different from the actual GCD $p(x,y) = y + 1$. ♦

In order to overcome the above two drawbacks, we do the following:

(a) We select the interpolation points y_i (resp. x_i) to be as random as possible. To this end, we select the points y_i (resp. x_i) to lie on the vertices of a regular b_2-gone (resp. b_1-gone) that is encircled on a circle of a complex plane with the center at the origin and the radius being a random real number c_2 (resp. c_1). If we consider the *variety* $V(p_{1,x,y}, p_{2,x,y}, \ldots, p_{n,x,y})$ defined by the polynomials $p_{i,x,y}(x,y)$, or otherwise the set of all the solutions (x_k, y_k) of equations

$p_{i,x,y}\left(x_k, y_k\right) = 0$, then the possibility for these points x_k (resp. y_k) to have magnitudes equal to a random real number c_1 (resp. c_2), as the proposed interpolation points, is zero. An additional reason for the proposed selection of interpolation points is that we need to use a discrete Fourier transform algorithm for the computation of the GCD.

(b) When we calculate the univariate GCD polynomials $\tilde{p}_k\left(x, y_k\right)$, we also take into account the product of the highest order coefficients of the polynomials $p_i\left(x, y_k\right)$ in order to retain all the information concerning the polynomial $f_y(y)$, i.e., $f_y(y)$ is the product of all the factors of the polynomials $p_i\left(x, y\right)$ that depend only on y.

In the next section, we present an algorithm based on DFT techniques that use the above two hints in order to overcome the problems presented in the previous two examples.

3. DFT Computation of the GCD of Bivariate Polynomials

Consider the finite sequence $X(k_1, k_2)$ and $\tilde{X}(r_1, r_2)$, $i = 1, 2$ and $k_i, r_i = 0, 1, \ldots, M_i$. In order for the sequence $X(k_1, k_2)$ and $\tilde{X}(r_1, r_2)$ to constitute a DFT pair, the following relations should hold:

$$\tilde{X}(r_1, r_2) = \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1, k_2) W_1^{-k_1 r_1} W_2^{-k_2 r_2}, \quad (3)$$

$$X(k_1, k_2) = \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \tilde{X}(r_1, r_2) W_1^{k_1 r_1} W_2^{k_2 r_2}, \quad (4)$$

where

$$W_i = e^{\frac{2\pi I}{M_i+1}}, \quad \forall i = 1, 2 \quad (5)$$

$$R = \left(M_1 + 1\right)\left(M_2 + 1\right) \quad (6)$$

and I represents the imaginary unit. The relation (3) is the *forward Fourier transform* of $X(k_1, k_2)$ while (4) the *inverse Fourier transform* of $\tilde{X}(r_1, r_2)$.

Determine two random real numbers c_1, c_2 and multiply the points

$$W_i^k = e^{\frac{2\pi k I}{b_i+1}}, \quad i = 1, 2 \text{ and } k = 0, 1, \ldots, b_i$$

by c_1 and c_2, respectively.

Lemma 1. *Let* $\widetilde{W}_i^k = c_i W_i^k$, $i = 1, 2$ *and* $k = 0, 1, \ldots, M_i$ *where the points* W_i^k *are defined by (5). Then the relations*

$$\tilde{X}'(r_1, r_2) = \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1, k_2) \widetilde{W}_1^{-k_1 r_1} \widetilde{W}_2^{-k_2 r_2}, \quad (7)$$

$$X(k_1, k_2) = \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \tilde{X}'(r_1, r_2) \widetilde{W}_1^{r_1 k_1} \widetilde{W}_2^{r_2 k_2} \quad (8)$$

constitute a forward and an inverse Fourier transform, respectively.

Proof. First note that

$$\begin{aligned}
&\tilde{X}'(r_1, r_2) \\
&= \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1, k_2) \widetilde{W}_1^{-k_1 r_1} \widetilde{W}_2^{-k_2 r_2} \\
&= \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1, k_2) \left(c_1 W_1^{k_1}\right)^{-r_1} \left(c_2 W_2^{k_2}\right)^{-r_2} \\
&= c_1^{-r_1} c_2^{-r_2} \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1, k_2) W_1^{-k_1 r_1} W_2^{-k_2 r_2} \\
&= c_1^{-r_1} c_2^{-r_2} \tilde{X}(r_1, r_2), \qquad (9)
\end{aligned}$$

where $\tilde{X}(r_1, r_2)$ is defined in (3). Then, by using (9) and (4), it is easily seen that the relation (8) holds. Indeed,

$$\begin{aligned}
&\frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \tilde{X}'(r_1, r_2) \widetilde{W}_1^{r_1 k_1} \widetilde{W}_2^{r_2 k_2} \\
&= \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \left[c_1^{-r_1} c_2^{-r_2} \tilde{X}(r_1, r_2)\right] \widetilde{W}_1^{r_1 k_1} \widetilde{W}_2^{r_2 k_2} \\
&= \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} c_1^{-r_1} c_2^{-r_2} \tilde{X}(r_1, r_2) \left(c_1 W_1^{k_1}\right)^{r_1} \left(c_2 W_2^{k_2}\right)^{r_2} \\
&= \sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} c_1^{-r_1} c_2^{-r_2} \tilde{X}(r_1, r_2) c_1^{r_1} c_2^{r_2} W_1^{k_1 r_1} W_2^{k_2 r_2} \\
&= \sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \tilde{X}(r_1, r_2) W_1^{k_1 r_1} W_2^{k_2 r_2} \\
&\overset{(4)}{=} X(k_1, k_2)
\end{aligned}$$

■

Therefore, according to Lemma 1, we can always change the interpolation points $y_k = W_2^k$ (resp. $x_k = W_1^k$) belonging to the unit circle with the center at the origin with the points $\tilde{y}_k = c_2 W_2^k$ (resp. $\tilde{x}_k = c_1 W_1^k$) belonging to a circle with the random radius $\|c_1\|$ (resp. $\|c_2\|$) and the same center. Due to the randomness of the new interpolation points $\tilde{y}_k$ (resp. $\tilde{x}_k$), the probability of the polynomials $p_{i,x,y}(x, \tilde{y}_k)$ to have GCDs other than the unity, or otherwise the probability that the interpolation points $(\tilde{x}_k, \tilde{y}_k)$ vanish all the polynomials $p_{i,x,y}(x, y)$ is zero. In that way, we can solve the first problem presented in the previous section.

Consider n polynomials $p_i(x, y) \in \mathbb{R}[x, y]$, $i = 0, 1, \ldots, n-1$ of the form (1) and define $\tilde{b}_i, i = 1, 2$ as

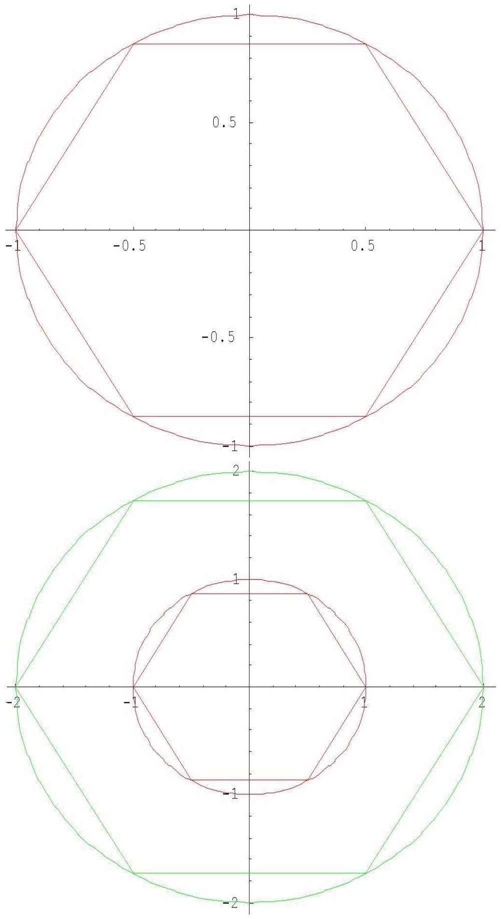

Fig. 1. The values $W_i^k = e^{\frac{2\pi k I}{b_i+1}}$ lie on the unit circle, while the values $2W_i^k = 2e^{\frac{2\pi k I}{b_i+1}}$ lie on a circle with radius 2.

follows:

$$\begin{aligned}\deg_x\left(\bar{p}(x,y)\right) &\le \tilde{b}_1 \\ &:= \left(\min_{i=0,1,\ldots,n-1}\{\deg_x\left(p_i(x,y)\right)\}\right) \\ \deg_y\left(\bar{p}(x,y)\right) &\le \tilde{b}_2 \\ &:= \left(\sum_{i=0}^{n-1}\{\deg_y\left(p_i(x,y)\right)\}\right)\end{aligned} \tag{10}$$

for the GCD. In the following, we shall propose an algorithm for the computation of the polynomial

$$\begin{aligned}\bar{p}(x,y) &= g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,y}(y) \\ &= \sum_{k_1=0}^{\tilde{b}_1}\sum_{k_2=0}^{\tilde{b}_2}\left(\bar{p}_{k_1,k_2}\right)\left(x^{k_1}y^{k_2}\right)\end{aligned} \tag{11}$$

for the GCD that comes from the polynomials $p_i(x,y)$, $i = 0,1,\ldots,n-1$ using the discrete Fourier transform. Here $\bar{p}(x,y)$ can be numerically computed via interpolation using the following $R_1 = (\tilde{b}_1+1)(\tilde{b}_2+1)$ points:

$$\begin{gathered}u_i(r_j) = c_i W_i^{r_j}, \qquad i = 1,2 \text{ and } r_j = 0,1,\ldots,\tilde{b}_i, \\ W_i = c_i e^{\frac{2\pi I}{\tilde{b}_i+1}},\end{gathered} \tag{12}$$

where $c_i, i = 1,2$ are random real numbers. In order to evaluate the coefficients $\bar{p}_{k_1,k_2}$,

(a) we determine the GCD $\tilde{p}_{r_2}\left(x,u_2(r_2)\right), r_2 = 0,1,\ldots,\tilde{b}_2$ of the polynomials $p_i\left(x,u_2(r_2)\right),\ i = 0,1,\ldots,n-1$ by using known algorithms for univariate polynomials, and

(b) we multiply each polynomial $\tilde{p}_{r_2}\left(x,u_2(r_2)\right)$ by the product of the highest order coefficients of the polynomials $p_i\left(x,u_2(r_2)\right)$ and thus get the polynomials $\bar{p}_{r_2}\left(x,u_2(r_2)\right)$. In this way, we succeed to take the product $\prod_{i=0}^{n-1}p'_{i,y}(y)$ in the final result.

Then, applying $u_1(r_1), r_1 = 0,1,\ldots,\tilde{b}_1$ in the above polynomials, we take

$$\tilde{p}_{r_1,r_2} = \bar{p}\left(u_1(r_1),u_2(r_2)\right). \tag{13}$$

From (11) and (13) we get

$$\tilde{p}_{r_1,r_2} = \sum_{l_1=0}^{\tilde{b}_1}\sum_{l_2=0}^{\tilde{b}_2}\left(\bar{p}_{l_1,l_2}\right)\left(W_1^{-r_1l_1}W_2^{-r_2l_2}\right).$$

Note that $[\bar{p}_{l_1,l_2}]$ and $[\tilde{p}_{r_1,r_2}]$ form a DFT pair, and thus using (4) we derive the coefficients of (11), i.e.,

$$\bar{p}_{l_1,l_2} = \frac{1}{R_1}\sum_{r_1=0}^{\tilde{b}_1}\sum_{r_2=0}^{\tilde{b}_2}\tilde{p}_{r_1,r_2}W_1^{r_1l_1}W_2^{r_2l_2}, \tag{14}$$

where $l_i = 0,\ldots,\tilde{b}_i$ and $i = 1,2$.

We can now present the following algorithm:

Algorithm 1. Computation of

$$\bar{p}(x,y) = g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,y}(y).$$

Step 1. Calculate the number $R_1 = (\tilde{b}_1+1)(\tilde{b}_2+1)$ of interpolation points.

Step 2. Compute R_1 points $(u_1(r_1),u_2(r_2))$ for $r_i = 0,1,\ldots,\tilde{b}_i$ and $i = 1,2$ as defined by (12).

Step 3. Determine the polynomials $\bar{p}_{r_2}\left(x,u_2(r_2)\right), r_2 = 0,1,\ldots,\tilde{b}_2$.

Step 4. Determine the values

$$\tilde{p}_{r_1,r_2} = \bar{p}\left(u_1(r_1),u_2(r_2)\right) = \bar{p}_{r_2}\left(u_1(r_1),u_2(r_2)\right)$$

for $r_i = 0,1,\ldots,\tilde{b}_i$ and $i = 1,2$.

Step 5. Use the inverse DFT (14) for the points $\tilde{p}_{r_1,r_2}$ in order to construct the values $\bar{p}_{l_1,l_2}$.

Step 6. Return the polynomial

$$\begin{aligned}\bar{p}(x,y) &= g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,y}(y)\\ &= \sum_{k_1=0}^{\tilde{b}_1}\sum_{k_2=0}^{\tilde{b}_2}\left(\bar{p}_{k_1,k_2}\right)\left(x^{k_1}y^{k_2}\right).\end{aligned}$$

■

Similarly to the previous algorithm, we can determine the polynomial

$$\begin{aligned}\bar{q}(x,y) &= \left(\prod_{i=0}^{n-1}p'_{i,x}(x)\right)g_{x,y}(x,y)g_y(x)\\ &= \sum_{k_1=0}^{\tilde{b}_1}\sum_{k_2=0}^{\tilde{b}_2}\left(\bar{q}_{k_1,k_2}\right)\left(x^{k_1}y^{k_2}\right).\end{aligned}$$

To this end, in Step 3 of the previous algorithm we first determine the polynomials $\tilde{q}_{r_1}\left(u_1(r_1),y\right)$, $r_1 = 0,1,\dots,\tilde{b}_1$ that come from the GCD of the univariate polynomials $p_i\left(u_1(r_1),y\right)$. Then we multiply each polynomial $\tilde{q}_{r_1}\left(u_1(r_1),y\right)$ by the product of the highest order coefficients of the polynomials $p_i\left(u_1(r_1),y\right)$ in order to determine the polynomials $\bar{q}_{r_1}\left(u_1(r_1),y\right)$. Then, by substituting the values of $u_2(r_2)$ in place of y, we get the values $\tilde{q}_{r_1,r_2} = \bar{q}\left(u_1(r_1),u_2(r_2)\right) = \bar{q}_{r_1}\left(u_1(r_1),u_2(r_2)\right)$. Finally, using the inverse DFT, we construct the values $\bar{q}_{k_1,k_2}$.

If we know the polynomials $\bar{p}(x,y)$ and $\bar{q}(x,y)$ defined above, then we can easily determine the GCD of the polynomials $p_i\left(x,y\right)$, as shown in the next result.

Proposition 1. *Consider n 2-D polynomials:*

$$\begin{aligned}&p_i(x,y)\\ &= \overbrace{p_{i,x}(x)g_x(x)}^{p'_{i,x}(x)}\overbrace{p_{i,x,y}(x,y)g_{x,y}(x,y)}^{p'_{i,x,y}(x,y)}\overbrace{p_{i,y}(y)g_y(y)}^{p'_{i,y}(y)},\\ &\qquad i = 0,1,\dots,n-1.\end{aligned}$$

If we define

$$\begin{aligned}\bar{p}(x,y) &= g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,y}(y),\\ \bar{q}(x,y) &= g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,x}(x),\end{aligned}$$

then the GCD is

$$p(x,y) = \bar{q}(x,y)\frac{\bar{p}(x,c)}{\bar{q}(x,c)} = \bar{p}(x,y)\frac{\bar{q}(c,y)}{\bar{p}(c,y)},$$

where c is an arbitrary number.

Proof. Note that

$$\begin{aligned}&\frac{\bar{q}(x,c)}{\bar{p}(x,c)}\\ &\quad= \frac{C_2 g_{x,y}(x,c)\prod_{i=0}^{n-1}p'_{i,x}(x)}{g_x(x)g_{x,y}(x,c)C_1} = \frac{C_2}{C_1}\frac{\prod_{i=0}^{n-1}p'_{i,x}(x)}{g_x(x)}\\ &\quad= \frac{C_2}{C_1}\frac{p'_{0,x}(x)\prod_{i=1}^{n-1}p'_{i,x}(x)}{g_x(x)}\\ &\quad= \frac{C_2}{C_1}\frac{p_{0,x}(x)g_x(x)\prod_{i=1}^{n-1}p'_{i,x}(x)}{g_x(x)}\\ &\quad= \frac{C_2}{C_1}p_{0,x}(x)\prod_{i=1}^{n-1}p'_{i,x}(x),\end{aligned}$$

where $\bar{p}(x,c)$ and $\bar{q}(x,c)$ are polynomials of only the variable x. Thus, the GCD will be

$$\begin{aligned}&p(x,y)\\ &\quad= \bar{q}(x,y)\frac{\bar{p}(x,c)}{\bar{q}(x,c)} = \left[g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,x}(x)\right]\\ &\qquad\left[\frac{C_1}{C_2}\frac{1}{p_{0,x}(x)\prod_{i=1}^{n-1}p'_{i,x}(x)}\right]\\ &\quad= \frac{C_1}{C_2}g_y(y)g_{x,y}(x,y)g_x(x)p_{0,x}(x)\\ &\qquad\prod_{i=1}^{n-1}p'_{i,x}(x)\frac{1}{p_{0,x}(x)\prod_{i=1}^{n-1}p'_{i,x}(x)}\\ &\quad= \frac{C_1}{C_2}g_y(y)g_{x,y}(x,y)g_x(x).\end{aligned}$$

The division that takes place in the above equations is the usual 1-D division over the variable x. ■

Algorithm 2. DFT computation of the GCD of two-variable polynomials.

Step 1. Compute $\bar{p}(x,y)$ by (4):

$$\bar{p}(x,y) = g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,y}(y).$$

Step 2. Compute $\bar{q}(x,y)$ by a modification of (4):

$$\bar{q}(x,y) = g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1}p'_{i,x}(x).$$

Step 3. The CGD is

$$p\left(x,y\right) = \bar{q}(x,y)\frac{\bar{p}(x,c)}{\bar{q}(x,c)} = \bar{p}(x,y)\frac{\bar{q}(c,y)}{\bar{p}(c,y)},$$

where c is an arbitrary real number. ■

Example 3. Consider the polynomials

$$\begin{aligned} &p_1\left(x,y\right)\\ &= \underbrace{1}_{p'_{1,x}(x)\equiv p_{1,x}(x)} \times \underbrace{(x+y)}_{g_{x,y}(x,y)} \times \underbrace{\underbrace{y}_{g_y(y)} \times \underbrace{(y+1)}_{p_{1,y}(y)},}_{p'_{1,y}(y)} \\ &p_2\left(x,y\right)\\ &= \underbrace{(x+1)}_{p'_{2,x}(x)\equiv p_{2,x}(x)} \times \underbrace{(x+y)}_{g_{x,y}(x,y)} \times \underbrace{\underbrace{y}_{g_y(y)} \times \underbrace{y}_{p_{2,y}(y)}}_{p'_{2,y}(y)}. \end{aligned}$$

Applying Algorithm 3, we have

Step 1. Compute $\bar{p}(x,y)$ by (4):

$$\begin{aligned} \bar{p}(x,y) &= g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,y} \\ &= 1 \times (x+y) \times \left[y(y+1) \times y^2\right]. \end{aligned}$$

Step 2. Compute $\bar{q}(x,y)$ by (4):

$$\begin{aligned} \bar{q}(x,y) &= g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x) \\ &= y \times (x+y) \times \left[1 \times (x+1)\right]. \end{aligned}$$

Step 3. Compute the values of the polynomials $\bar{p}(x,c)$ and $\bar{q}(x,c)$, where c is an arbitrary number, i.e., $c = 1.27$,

$$\begin{aligned} \bar{p}(x,1.27) &= 4.64983(1.27+x), \\ \bar{q}(x,1.27) &= 1.27(1+x)(1.27+x). \end{aligned}$$

Step 4. The GCD we are looking for is

$$\begin{aligned} &p\left(x,y\right) \\ &= \bar{q}(x,y)\frac{\bar{p}(x,1.27)}{\bar{q}(x,1.27)} \\ &= \left(x+1\right)\left(x+y\right)y\frac{4.64983(1.27+x)}{1.27(1+x)(1.27+x)} \\ &= \left(x+y\right)y\frac{4.64983}{1.27} \\ &= 3.661\,3\left(x+y\right)y. \end{aligned}$$

We present here only the steps for the computation of $\bar{q}(x,y)$. The computation of $\bar{p}(x,y)$ is similar.

Step 2.1. Calculate the numbers of interpolation points $b_i, i = 1,2$ by (10):

$$\begin{aligned} b_1 &= \deg_x\left[(x+y)y(y+1)\right] \\ &\quad + \deg_x\left[\left(x+1\right)\left(x+y\right)y^2\right] \\ &= 3, \\ b_2 &= \min\big\{\deg_y\left[(x+y)y(y+1)\right], \\ &\quad \deg_y\left[\left(x+1\right)\left(x+y\right)y^2\right]\big\} \\ &= 3. \end{aligned}$$

Step 2.2. Compute the number of interpolation points,

$$R_1 = \prod_{i=1}^{2}(b_i+1) = (3+1)\left(3+1\right) = 16$$

Let $c_1 = c_2 = 5$. Then we have the following interpolation points:

$$\begin{aligned} u_1\left(0\right) &= u_2\left(0\right) = 5W_0^0 = 5, \\ u_1\left(1\right) &= u_2\left(1\right) = 5W_0^1 = 5e^{\frac{2\pi I}{4}} = 5I, \\ u_1\left(2\right) &= u_2\left(2\right) = 5W_0^2 = 5e^{\frac{4\pi I}{4}} = -5, \\ u_1\left(3\right) &= u_2\left(3\right) = 5W_0^3 = 5e^{\frac{6\pi I}{4}} = -5I. \end{aligned}$$

Step 2.3. Determine the GCD of the polynomials $p_1\left(u_1(r_1),y\right)$ and $p_2\left(u_1(r_1),y\right)$ multiplied by the coefficients of the greatest powers of y of the polynomials $p_1\left(u_1(r1),y\right)$ and $p_2\left(u_1(r_1),y\right)$: Since

$$\begin{aligned} p_1\left(5,y\right) &= y(1+y)(5+y), \\ p_2\left(5,y\right) &= 6y^2(5+y), \end{aligned}$$

we get

$$\begin{aligned} \bar{q}_0\left(5,y\right) &= (5+y)y \times 1 \times 6 \\ &= 6(5+y)y. \end{aligned}$$

Since

$$\begin{aligned} p_1\left(5I,y\right) &= y(5I+y)(1+y), \\ p_2\left(5I,y\right) &= (1+5I)y^2(5I+y), \end{aligned}$$

we get

$$\begin{aligned} \bar{q}_1\left(5I,y\right) &= y(5I+y) \times 1 \times (1+5I) \\ &= (1+5I)y(5I+y). \end{aligned}$$

Since

$$\begin{aligned} p_1\left(-5,y\right) &= (-5+y)y(1+y), \\ p_2\left(-5,y\right) &= -4(-5+y)y^2, \end{aligned}$$

we get

$$\begin{aligned} \bar{q}_2\left(-5,y\right) &= (-5+y)y \times 1 \times (-4) \\ &= -4(-5+y)y. \end{aligned}$$

Since

$$\begin{aligned} p_1\,(-5I,y) &= y(-5I+y)(1+y),\\ p_2\,(-5I,y) &= (1-5I)y^2(-5I+y), \end{aligned}$$

we get

$$\begin{aligned} \bar{q}_3\,(-5I,y) &= y(-5I+y) \times 1 \times (1-5I)\\ &= (1-5I)y(-5I+y). \end{aligned}$$

Then the values of each polynomial at $u_2(r_2)$ are

$$\begin{aligned} &\tilde{q}_{0,0} = 300, && \tilde{q}_{1,0} = -100 + 150I,\\ &\tilde{q}_{0,1} = -150 + 150I, && \tilde{q}_{1,1} = -50 - 250I,\\ &\tilde{q}_{0,2} = 0, && \tilde{q}_{1,2} = 150 + 100I,\\ &\tilde{q}_{0,3} = -150 - 150I, && \tilde{q}_{1,3} = 0,\\ &\tilde{q}_{2,0} = 0, && \tilde{q}_{3,0} = -100 - 150I,\\ &\tilde{q}_{2,1} = 100 + 100I, && \tilde{q}_{3,1} = 0,\\ &\tilde{q}_{2,2} = -200, && \tilde{q}_{3,2} = 150 - 100j,\\ &\tilde{q}_{2,3} = 100 - 100I, && \tilde{q}_{3,3} = -50 + 250I, \end{aligned}$$

and thus the values $\tilde{q}_{r_1,r_2}$ in (13) are constructed.

Step 2.4. Use the inverse DFT (14) for the points $\tilde{q}_{r_1,r_2}$ in order to construct the values

$$\bar{q}_{l_1,l_2} = \frac{1}{16}\sum_{r_1=0}^{3}\sum_{r_2=0}^{3}\tilde{q}_{r_1,r_2}\widetilde{W}_1^{r_1 l_1}\widetilde{W}_1^{r_2 l_2}.$$

We get

$$\begin{aligned} &\bar{q}_{0,0} = 0, \quad \bar{q}_{0,1} = 0, \quad \bar{q}_{0,2} = 1, \quad \bar{q}_{0,3} = 0,\\ &\bar{q}_{1,0} = 0, \quad \bar{q}_{1,1} = 1, \quad \bar{q}_{1,2} = 1, \quad \bar{q}_{1,3} = 0,\\ &\bar{q}_{2,0} = 0, \quad \bar{q}_{2,1} = 1, \quad \bar{q}_{2,2} = 0, \quad \bar{q}_{2,3} = 0,\\ &\bar{q}_{3,0} = 0, \quad \bar{q}_{3,1} = 0, \quad \bar{q}_{3,2} = 0, \quad \bar{q}_{3,3} = 0, \end{aligned}$$

Thus,

$$\begin{aligned} \bar{q}(x,y) &= \sum_{k_1=0}^{3}\sum_{k_2=0}^{3} p_{k_1,k_2} x^{k_1} y^{k_2}\\ &= xy + y^2 + xy^2 + x^2 y\\ &= (x+1)\,(x+y)\,y. \end{aligned}$$

■

4. Conclusions

An algorithm for the computation of the GCD of two-variable polynomials have been developed based on DFT techniques. It inherits its main advantages of speed and robustness from DFT techniques. The above algorithm can be extended to the 2-D polynomial matrix case by using existing methods of the computation of the GCD of 1-D polynomial matrices. A similar algorithm can also be applied for the computation of the least common multiple of 2-D polynomials and polynomial matrices.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

Karampetakis N. P., Tzekis P., (2005): *On the computation of the minimal polynomial of a polynomial matrix*. International Journal of Applied Mathematics and Computer Science, Vol. 15, No. 3, pp. 339–349.

Karcanias N. and Mitrouli M., (2004): *System theoretic based characterisation and computation of the least common multiple of a set of polynomials*. Linear Algebra and Its Applications, Vol. 381, pp. 1–23.

Karcanias N. and Mitrouli, M., (2000): *Numerical computation of the least common multiple of a set of polynomials*, Reliable Computing, Vol. 6, No. 4, pp. 439–457.

Karcanias N. and Mitrouli M., (1994): *A matrix pencil based numerical method for the computation of the GCD of polynomials*. IEEE Transactions on Automatic Control, Vol. 39, No. 5, pp. 977–981.

Mitrouli M. and Karcanias N., (1993): *Computation of the GCD of polynomials using Gaussian transformation and shifting*. International Journal of Control, Vol. 58, No. 1, pp. 211–228.

Noda M. and Sasaki T., (1991): *Approximate GCD and its applications to ill-conditioned algebraic equations*. Journal of Computer and Applied Mathematics Vol. 38, No. 1–3, pp. 335–351.

Pace I. S. and Barnett S., (1973): *Comparison of algorithms for calculation of GCD of polynomials*. International Journal of Systems Science Vol. 4, No. 2, pp. 211–226.

Paccagnella, L. E. and Pierobon, G. L., (1976): *FFT calculation of a determinantal polynomial*. IEEE Transactions on Automatic Control, Vol. 21, No. 3, pp. 401–402.

Schuster, A. and Hippe, P., (1992): *Inversion of polynomial matrices by interpolation*. IEEE Transactions on Automatic Control, Vol. 37, No. 3, pp. 363–365.

Received: 17 April 2007
Revised: 25 July 2007
Re-revised: 31 October 2007