# On the computation of the GCD (LCM) of 2-d polynomials

P. Tzekis[1] N. P. Karampetakis[2] H. K. Terzidis[1]
[1]Technological Educational Institution of Thessaloniki
School of Sciences - Department of Mathematics
P.O. Box 14561, GR-541 01 Thessaloniki , Greece
email : tzekis@math.auth.gr
[2]Department of Mathematics,
Aristotle University of Thessaloniki
Thessaloniki 54006, Greece
email : karampet@math.auth.gr

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

## I. Introduction

Two basic problems of algebraic computations are the computations of the greatest common divisor (GCD) and least common multiple (LCM) of a set of polynomials. The GCD is usually linked with the characterisation of zeros of a polynomial matrix description of a system whereas LCM is connected with the derivation of minimal realizations of the transfer function representation of a system. The problem of finding the GCD of a set of $n$ polynomials on $R[x]$ of maximal degree $q$, is a classical problem that has been considered before [6]. Numerical methods for the LCM [7] and GCD ([8], [9]) have also been developed. Due to the difficulty of finding the exact GCD and LCM of a set of polynomials, approximate algorithms have also been developed [10]. Comparison of algorithms for calculation of GCD of polynomials is given in [11].

The main disadvantage of many algorithms is their complexity. In order to overcome these difficulties we may use other techniques such as interpolation methods. [13] for example, 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 Transforms (DFT) techniques or better Fast Fourier Transforms (FFT) techniques. Some of the advantages of the DFT based algorithms are that there are very efficient algorithms available both in software and hardware and that there are greatly benefitted by the existence of a parallel environment (through symmetric multiprocessing or other techniques). [5] for example used DFT techniques to compute the minimal polynomial of a polynomial matrix and [12] used FFT methods for the computation of the determinant of a polynomial matrix.

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

## II. DFT calculation of a GCD of two-variable polynomials

Consider $n$ polynomials of the form

$$A_k(x,y) = \sum_{m=0}^{M_1}\sum_{j=0}^{M_2} A_{k,m,j}x^m y^j \in R[x,y] \tag{1}$$

$$k = 1,2,...,n-1 \tag{2}$$

where $M_1$ (resp. $M_2$) is the greatest power of $x$ (resp. $y$) in $A_k(x,y)$. Consider the finite sequence $X(k_1,k_2)$ and $\tilde{X}(r_1,r_2)$, $k_i, r_i = 0,1,..,M_i$. 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 :

$$\tilde{X}(r_1,r_2) = \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2} X(k_1,k_2)W_1^{-k_1r_1}W_2^{-k_2r_2} \tag{3}$$

$$X(k_1,k_2) = \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2} \tilde{X}(r_1,r_2)W_1^{k_1r_1}W_2^{k_2r_2} \tag{4}$$

where

$$W_i = e^{\frac{2\pi j}{M_i+1}}, \forall i = 1,2 \tag{5}$$

$$R = (M_1+1)(M_2+1) \tag{6}$$

Relation (3) is the forward Fourier transform of $X(k_1,k_2)$ while (4) the inverse Fourier transform of $\tilde{X}(r_1,r_2)$.

In the following we will propose a new algorithm for the computation of the GCD and LCM of the $A_k(x,y)$, $k = 0,1,...,n-1$ 2-d polynomials using discrete Fourier transform. The greatest powers of the variables $x,y$ in CGD (LCM) $p(x,y)$ are for GCD

$$\begin{aligned}\deg_x(p(x,y)) = b_0 &:= \left(\leq \min_{k=0,1,\dots,n-1}\{\deg_x(A_k)\}\right)\\ \deg_y(p(x,y)) = b_1 &:= \left(\leq \min_{k=0,1,\dots,n-1}\{\deg_y(A_k)\}\right)\end{aligned} \tag{7}$$

and for LCM

$$\begin{array}{l}\deg_x\left(p(x,y)\right)=b_0:=\left(\leq\sum_{k=0}^{n-1}\deg_x\left(A_k\right)\right)\\ \deg_y\left(p(x,y)\right)=b_1:=\left(\leq\sum_{k=0}^{n-1}\deg_y\left(A_k\right)\right)\end{array}$$

Thus, the polynomial $p(x,y)$ can be written as

$$p(x,y)=\sum_{k_0=0}^{b_0}\sum_{k_1=0}^{b_1}\left(p_{k_0,k_1}\right)\left(x^{k_0}y^{k_1}\right) \tag{8}$$

and can be numerically computed via interpolation using the following $R_1$ points

$$\begin{array}{c}u_i(r_j)=W_i^{-r_j};i=0,1\text{ and }r_j=0,1,...,b_i\\ W_i=e^{\frac{2\pi j}{b_i+1}}\end{array} \tag{9}$$

where

$$R_1=(b_0+1)(b_1+1) \tag{10}$$

In order, to evaluate the coefficients $p_{k_0,k_1}$ define

$$\tilde{p}_{x,r_1}=p\left(x,u_1(r_1)\right) \tag{11}$$

where we use an algorithm for the computation of the GCD (LCM) of the above polynomials. Then, by applying $u_0(r_0)$ in the above polynomial, we take

$$\tilde{p}_{r_0,r_1}=p\left(u_0(r_0),u_1(r_1)\right) \tag{12}$$

From (8), (9), (12) we get

$$\tilde{p}_{r_0,r_1}=\sum_{l_0=0}^{b_0}\sum_{l_1=0}^{b_1}\left(p_{l_0,l_1}\right)\left(W_0^{-r_0l_0}W_1^{-r_1l_1}\right)$$

Notice that $[p_{l_0,l_1}]$ and $[\tilde{p}_{r_0,r_1}]$ form a DFT pair and thus using (4) we derive the coefficients of (8) i.e.

$$p_{l_0,l_1}=\frac{1}{R_1}\sum_{r_0=0}^{b_0}\sum_{r_1=0}^{b_1}\tilde{p}_{r_0,r_1}W_0^{r_0l_0}W_1^{r_1l_1} \tag{13}$$

where $l_i=0,\ldots,b_i$.

*Algorithm 1:* DFT computation of the gcd of two-variable polynomials

Step 1. Calculate the number of interpolation points $b_i$ by (7).

Step 2. Compute the $R_1$ points $u_i(r_j)$ in (9).

Step 3. Determine the gcd of the polynomials $A_k\left(x,u_1(r_1)\right)$, $i=0,1,...,n-1$.Then, apply the points $u_1(r_1)$ in order to construct the values $\tilde{p}_{r_0,r_1,r_2}$ in (12).

Step 4. Use the inverse DFT (13) for the points $\tilde{p}_{r_0,r_1}$ in order to construct the values $p_{l_0,l_1}$.

*Example 2:* Consider two polynomials $A\left(x,y\right)=x^2y+3xy+x+3$ and $B\left(x,y\right)=xy^2+3xy+y+3$. Then by applying the DFT algorithm 1 we have

*Step 1.* Calculate the number of interpolation points $b_i$ by (7).

$$\begin{array}{l}b_0=\min\left\{\deg_x A(x,y),\deg_x B(x,y)\right\}=\min\left\{2,1\right\}=1\\ b_1=\min\left\{\deg_y A(x,y),\deg_y B(x,y)\right\}=\min\left\{1,2\right\}=1\end{array}$$

*Step 2.* Compute the

$$R_1=\prod_{i=0}^{1}(b_i+1)=(1+1)\left(1+1\right)=4$$

points $u_i(r_j)=W_i^{-r_j},W_i=e^{\frac{2\pi j}{b_i+1}},i=0,1$ and $r_j=0,1,..,b_i$ in (9).

$$\begin{array}{c}u_0\left(0\right)=u_1\left(0\right)=W_0^0=1\\ u_0\left(1\right)=u_1\left(1\right)=W_0^{-1}=e^{-\frac{2\pi j}{1+1}}=e^{-\pi j}=-1\end{array}$$

*Step 3.* Determine the GCD of the polynomials $A\left(x,u_1(r_1)\right)$, $B\left(x,u_1(r_1)\right)$

$$\begin{array}{c}p\left(x,u_1\left(0\right)\right)=1+x\\ p\left(x,u_1\left(1\right)\right)=1-x\end{array}$$

and then the values at $u_0(r_0)$ of each polynomial

$$\begin{array}{c}\tilde{p}_{0,0}=p\left(u_0\left(0\right),u_1\left(0\right)\right)=\left(1+x\right)_{x=1}=2\\ \tilde{p}_{1,0}=p\left(u_0\left(1\right),u_1\left(0\right)\right)=\left(1+x\right)_{x=-1}=0\end{array}$$

$$\begin{array}{c}\tilde{p}_{0,1}=p\left(u_0\left(0\right),u_1\left(1\right)\right)=\left(1-x\right)_{x=1}=0\\ \tilde{p}_{1,1}=p\left(u_0\left(1\right),u_1\left(1\right)\right)=\left(1-x\right)_{x=-1}=2\end{array}$$

and thus construct the values $\tilde{p}_{r_0,r_1}$ in (12).

Step 4. Use the inverse DFT (13) for the points $\tilde{p}_{r_0,r_1}$ in order to construct the values $p_{l_0,l_1}=\frac{1}{4}\sum_{r_0=0}^{1}\sum_{r_1=0}^{1}\tilde{p}_{r_0,r_1}W_0^{r_0l_0}W_1^{r_1l_1}$.

$$\begin{array}{c}p_{0,0}=1,p_{0,1}=0\\ p_{1,0}=0,p_{1,1}=1\end{array}$$

and thus the *GCD* is

$$p\left(x,y\right)=xy+1$$

In case where we are interested for the LCM of the polynomails $A\left(s\right),B(s)$ we need to change the steps as follows :

*Step 1a.* Calculate the number of interpolation points $b_i$ by (7).

$$\begin{array}{l}b_0=\deg_x A(x,y)+\deg_x B(x,y)=2+1=3\\ b_1=\deg_y A(x,y)+\deg_y B(x,y)=1+2=3\end{array}$$

*Step 2a.* Compute the

$$R_1=\prod_{i=0}^{1}(b_i+1)=(3+1)\left(3+1\right)=16$$

points $u_i(r_j)=W_i^{-r_j},W_i=e^{\frac{2\pi j}{b_i+1}},i=0,1$ and $r_j=0,1,..,b_i$ in (9).

$$u_0(0) = u_1(0) = W_0^0 = 1$$
$$u_0(1) = u_1(1) = W_0^{-1} = e^{-\frac{2\pi j}{3+1}} = e^{-\frac{\pi j}{2}}$$
$$u_0(2) = u_1(2) = W_0^{-2} = e^{-2\frac{2\pi j}{3+1}} = e^{-\pi j}$$
$$u_0(3) = u_1(3) = W_0^{-3} = e^{-3\frac{2\pi j}{3+1}} = e^{-\frac{3\pi j}{2}}$$

*Step 3a* Determine the LCM of the 1-d polynomials $A(x, u_1(r_1)), B(x, u_1(r_1))$

$$p(x, u_1(0)) = 4x^2 + 16x + 12$$
$$p(x, u_1(1)) = -(1+3j)x^2 - 10jx + 9 - 3j$$
$$p(x, u_1(2)) = -2x^2 - 4x + 6$$
$$p(x, u_1(3)) = -(1-3j)x^2 + 10jx + 9 + 3j$$

and then the values at $u_0(r_0)$ of each polynomial

$$\tilde{p}_{0,0} = 32, \tilde{p}_{1,0} = 8 - 16j, \tilde{p}_{2,0} = 0, \tilde{p}_{3,0} = 8 + 16j$$
$$\tilde{p}_{0,1} = 8 - 16j, \tilde{p}_{1,1} = 0, \tilde{p}_{2,1} = 8 + 4j, \tilde{p}_{3,1} = 20$$
$$\tilde{p}_{0,2} = 0, \tilde{p}_{1,2} = 8 + 4j, \tilde{p}_{2,2} = 8, \tilde{p}_{3,2} = 8 - 4j$$
$$\tilde{p}_{0,3} = 8 + 16j, \tilde{p}_{1,3} = 20, \tilde{p}_{2,3} = 8 - 4j, \tilde{p}_{3,3} = 0$$

and thus construct the values $\tilde{p}_{r_0,r_1}$ in (12).

*Step 4a.* Use the inverse DFT (13) for the points $\tilde{p}_{r_0 r_1}$ in order to construct the values $p_{l_0,l_1} = \frac{1}{16}\sum_{r_0=0}^{3}\sum_{r_1=0}^{3}\tilde{p}_{r_0,r_1}W_0^{r_0 l_0}W_1^{r_1 l_1}$.

$$p_{0,0} = 9, p_{0,1} = 3, p_{0,2} = 0, p_{03} = 0$$
$$p_{1,0} = 3, p_{1,1} = 10, p_{1,2} = 3, p_{1,3} = 0$$
$$p_{2,0} = 0, p_{2,1} = 3, p_{2,2} = 1, p_{2,3} = 0$$
$$p_{3,0} = 0, p_{3,1} = 0, p_{3,2} = 0, p_{3,3} = 0$$

and thus the *LCM* is

$$p(x, y) = x^2y^2 + 3xy^2 + 3x^2y + 10xy + 3y + 3x + 9$$

Of course if we want to calculate the LCM, is better to calculate the GCD and then the LCM is the multiplication of the two polynomials divided by the GCD. But this happens in the case we have two polynomials. If we have more than two polynomials this is more complicated.

Let now see where the above algorithm suffers, by giving two special cases :

*Example 3:* Consider two polynomials $A(x, y) = (y + x)(x + 1)y$ and $B(x, y) = (xy + 1)(x + 1)$. Then by applying the DFT algorithm 1 we have

*Step 1.* Calculate the number of interpolation points $b_i$ by (7).

$$b_0 = \min\left\{\begin{array}{c}\deg_x\left[\left(y^2 + xy\right)(x+1)\right],\\ \deg_x\left[(xy+1)(x+1)\right]\end{array}\right\} = 2$$
$$b_1 = \min\left\{\begin{array}{c}\deg_y\left[\left(y^2 + xy\right)(x+1)\right],\\ \deg_y\left[(xy+1)(x+1)\right]\end{array}\right\} = 1$$

*Step 2. Compute the*

$$R_1 = \prod_{i=0}^{1}(b_i + 1) = (2+1)(1+1) = 6$$

*points* $u_i(r_j) = W_i^{-r_j}, W_i = e^{\frac{2\pi j}{b_i+1}}, i = 0, 1$ *and* $r_j = 0, 1, .., b_i$ *in (9).*

$$u_0(0) = W_0^0 = 1$$
$$u_0(1) = W_0^{-1} = e^{-\dfrac{2\pi j}{2+1}} = e^{-\dfrac{2\pi j}{3}}$$
$$u_0(2) = W_0^{-2} = e^{-2\dfrac{2\pi j}{2+1}} = e^{-\dfrac{4\pi j}{3}}$$

$$u_1(0) = W_1^0 = 1$$
$$u_1(1) = W_1^{-1} = e^{-\dfrac{2\pi j}{1+1}} = e^{-\dfrac{2\pi j}{2}} = -1$$

*Step 3. Determine the GCD of the polynomials* $A(x, u_1(r_1))$*,* $B(x, u_1(r_1))$

$$p(x, u_1(0)) = (1+x)(1+x)$$
$$p(x, u_1(1)) = (1+x)(1-x)$$

In reality, as we can see, the gcd of polynomials $A(x, y)$, $B(x, y)$ is $1 + x$. Here we have $p(x, u_1(0)) = (1+x)(1+x)$ and $p(x, u_1(1)) = (1+x)(1-x)$. If we will continue the algorithm, we shall take the result $(1 + x)(1 + xy)$ which is wrong. This happens because for $u_1(0) = W_1^0 = 1$, we have $A(x, u_1(0)) = (1+x)(1+x)$, $B(x, u_1(0))= (1+x)(1+x)$ and for $u_1(1) = W_1^{-2} = e^{-2\frac{2\pi j}{3+1}} = e^{-\pi j} = -1$, we have $A(x, u_1(1)) = (1+x)(1-x)$, $B(x, u_1(1))= (1+x)(1-x)$ and so, $u_1(i)$ creates temporarly extra common divisors for these two polynomials. To solve this problem, we can just determine two random real numbers $c_1, c_2$ and multiply the points $W_i = e^{\frac{2\pi j}{b_i+1}}, i = 0, 1$ and $j = 0, 1, .., b_i$ with $c_1$and $c_2$ respectively and thus we have

$$\widetilde{W}_1 = c_1W_1 \qquad \widetilde{W}_2 = c_2W_2 \tag{14}$$

$$\tilde{X}(r_1, r_2) = \sum_{k_1=0}^{M_1}\sum_{k_2=0}^{M_2}X(k_1, k_2)\widetilde{W}_1^{-k_1r_1}\widetilde{W}_2^{-k_2r_2} \tag{15}$$

$$X(k_1, k_2) = \frac{1}{R}\sum_{r_1=0}^{M_1}\sum_{r_2=0}^{M_2}\tilde{X}(r_1, r_2)\widetilde{W}_1^{r_1k_1}\widetilde{W}_2^{r_2k_2} \tag{16}$$

where

$$\widetilde{W}_i = c_ie^{\frac{2\pi j}{M_i+1}} \ \forall i = 1, 2 \tag{17}$$
$$R = (M_1 + 1)(M_2 + 1) \tag{18}$$

For example, let $c_1 = c_2 = 1.55$. Then we will have

$$u_0(0) = 1.55 W_0^0 = 1.55$$
$$u_0(1) = 1.55 W_0^{-1} = 1.55 e^{-\frac{2\pi j}{3}}$$
$$u_0(2) = 1.55 W_0^{-2} = 1.55 e^{-\frac{4\pi j}{3}}$$

$$u_1(0) = 1.55 W_1^0 = 1.55$$
$$u_1(1) = 1.55 W_1^{-1} = 1.55 e^{-\frac{2\pi j}{2}} = -1.55$$

*Step 3.* Determine the GCD of the polynomials $A(x, u_1(r_1))$, $B(x, u_1(r_1))$

$$p(x, u_1(0)) = 1 + x$$
$$p(x, u_1(1)) = 1 + x$$

and then the values at $u_0(r_0)$ of each polynomial

$$\tilde{p}_{0,0} = 2.55, \tilde{p}_{1,0} = 0.225 - 1.34234j$$
$$\tilde{p}_{2,0} = 0.225 + 1.34234j$$
$$\tilde{p}_{0,1} = 2.55, \tilde{p}_{1,1} = 0.225 - 1.34234j$$
$$\tilde{p}_{2,1} = 0.225 + 1.34234j$$

and thus construct the values $\tilde{p}_{r_0,r_1}$ in (12).

Step 4. Use the inverse DFT (13) for the points $\tilde{p}_{r_0 r_1}$ in order to construct the values $p_{l_0,l_1} = \frac{1}{6}\sum_{r_0=0}^{2}\sum_{r_1=0}^{1} \tilde{p}_{r_0,r_1} \widetilde{W}_0^{r_0 l_0} \widetilde{W}_1^{r_1 l_1}$.

$$p_{0,0} = 1, p_{0,1} = 0$$
$$p_{1,0} = 1, p_{1,1} = 0$$
$$p_{2,0} = 0, p_{2,1} = 0$$

and thus the *GCD* is

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

Another problem that we have to solve appears in the below example:

*Example 4:* Consider two polynomials $A(x, y) = (x + y)(y + 1)$ and $B(x, y) = (xy + 1)(y + 1)$. Then by applying the DFT algorithm 1 we have

*Step 1.* Calculate the number of interpolation points $b_m$ by (7).

$$b_0 = \min\left\{ \begin{array}{c} \deg_x[(x+y)(y+1)], \\ \deg_x[(xy+1)(y+1)] \end{array} \right\} = 1$$
$$b_1 = \min\left\{ \begin{array}{c} \deg_y[(x+y)(y+1)], \\ \deg_y[(xy+1)(y+1)] \end{array} \right\} = 2$$

*Step 2. Compute the*

$$R_1 = \prod_{m=0}^{1} (b_m + 1) = (1 + 1)(2 + 1) = 6$$

*points* $u_i(r_j) = W_i^{-r_j}, W_i = e^{\frac{2\pi j}{b_i+1}}, i = 0, 1$ *and* $r_j = 0, 1, .., b_i$ *in (9).*

$$u_0(0) = W_1^0 = 1$$
$$u_0(1) = W_1^{-1} = e^{-\frac{2\pi j}{1+1}} = e^{-\frac{2\pi j}{2}} = -1$$

$$u_1(0) = W_0^0 = 1$$
$$u_1(1) = W_0^{-1} = e^{-\frac{2\pi j}{2+1}} = e^{-\frac{2\pi j}{3}}$$
$$u_1(2) = W_0^{-2} = e^{-2\frac{2\pi j}{2+1}} = e^{-\frac{4\pi j}{3}}$$

*Step 3. Determine the GCD of the polynomials* $A(x, u_1(r_1))$, $B(x, u_1(r_1))$

$$p(x, u_1(0)) = 1$$
$$p(x, u_1(1)) = 1$$
$$p(x, u_1(2)) = 1$$

But we know that the gcd of polynomials $A(x, y)$, $B(x, y)$ is $(y + 1)$. This happens because when we replace

the interpolation points in the variable $y$, the factor $y+1$ gives us a constant value (let say $u_1(k) + 1$) and the polynomials will be

$$\begin{aligned} A(x, u_1(k)) &= (x + u_1(k))(u_1(k) + 1) = \\ &= C_1(x + u_1(k)) \\ B(x, u_1(k)) &= (u_1(k)x + 1)(u_1(k) + 1) = \\ &= C_2\left(x + \frac{1}{u_1(k)}\right) \end{aligned}$$

which they have gcd 1. Basicaly, the problem exists in case where appear one or more common factors only of variable $y$.

Let now say that we have $n$ polynomials $p_i(x, y) \in R[x, y]$ $i = 0, 1, ..., n - 1$. These polynomials can be rewritten as

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

where $p_{i,x}(x) \in R[x]$ are prime each other, $p_{i,y}(y) \in R[y]$ are prime each other, $p_{i,x,y}(x, y) \in R[x, y]$ are prime each other with no factors only of $x$ or $y$, $g_x(x)$ 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 $p'_{i,x,y}(x,y)$. We use the previous algorithm with one difference: when we calculate the gcd in variable $x$ (in 1-d case) we take also the product of constants. So, when we are using the inverse DFT, we take as result the polynomial $Q_1(x,y) = g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,y}(y)$. If we do the same for the variable $y$, we will take the polynomial $Q_2(x,y) = g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x)$. Then, if we choose an arbitrary number $c$ and we replace the variable $y$ with $c$, we will take $Q_1(x,c) = g_x(x)g_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,y}(c) = g_x(x)g_{x,y}(x,c)C_1$ and $Q_2(x,c) = g_y(c)g_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x) = C_2 g_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x)$. Then we have

$$\frac{Q_2(x,c)}{Q_1(x,c)} = \frac{C_2 g_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x)}{g_x(x)g_{x,y}(x,c)C_1} = \frac{C_2}{C_1}\frac{\prod_{i=0}^{n-1} p'_{i,x}(x)}{g_x(x)} = = \frac{C_2}{C_1} p_{0,x}(x)\prod_{i=1}^{n-1} p'_{i,x}(x)$$

where $Q_1(x,c)$ and $Q_2(x,c)$ are polynomials only of variable $x$. Thus, the gcd will be

$$G(x,y) = Q_2(x,y)\frac{Q_1(x,c)}{Q_2(x,c)} =$$
$$= \left[g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x)\right]\left[\frac{C_1}{C_2}\frac{1}{p_{0,x}(x)\prod_{i=1}^{n-1} p'_{i,x}(x)}\right] =$$
$$= \frac{C_1}{C_2} g_y(y)g_{x,y}(x,y)g_x(x)p_{0,x}(x)\times$$
$$\times\prod_{i=1}^{n-1} p'_{ix}(x)\frac{1}{p_{0,x}(x)\prod_{i=1}^{n-1} p'_{i,x}(x)} =$$
$$= \frac{C_1}{C_2} g_y(y)g_{x,y}(x,y)g_x(x)$$

The division that take place in the above equations, is the usualy 1-d division over the variable $x$. Of course, the number of interpolation points of $x$ and $y$ are for GCD

$$\begin{array}{c} \deg_x(p(x,y)) = b_0 := \left(\le \min_{k=0,1,\ldots,n-1}\{\deg_x(A_k)\}\right) \\ \deg_y(p(x,y)) = b_1 := \left(\le \sum_{k=0}^{n-1}\{\deg_y(A_k)\}\right) \end{array} \quad (19)$$

Note that the points $u_i(r_j) = c_i^{-1}W_i^{-r_j}$ are arbitrary and the polynomials $p_{i,x}(x)$, $p_{i,y}(y)$ and $p_{i,x,y}(x,y)$ are not creating temporary common factors. For example, in our application, if we know that the variables have common zeroes in a special area we can choose $c_1, c_2$ such that the points $u_i(r_j) = c_i^{-1}W_i^{-r_j}$ belong outside of this area. We summarize the above results by the following extended algorithm :

*Algorithm 5: DFT computation of the gcd of two-variable polynomials*

Step 1. Compute the $Q_1(x,y) = g_x(x)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,y}(y)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $y$.

Step 2. Compute the $Q_2(x,y) = g_y(y)g_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $x$.

Step 3. Compute the 1-d polynomials $Q_1(x,c)$ and $Q_2(x,c)$.

Step 4. The CGD is $G(x,y) = Q_2(x,y)\frac{Q_1(x,c)}{Q_2(x,c)} = Q_1(x,y)\frac{Q_2(c,y)}{Q_1(c,y)}$.

The same algorithm is also applied for the computation of the LCM:

When we are using the inverse DFT, we take as result the polynomial $Q_1(x,y) = l_x(x)l_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,y}(y)$. If we do the same for the variable $y$, we will take the polynomial $Q_2(x,y) = l_y(y)l_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x)$. Then, if we choose an arbitrary number $c$ and we replace the variable $y$ with $c$, we will take $Q_1(x,c) = l_x(x)l_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,y}(c) = l_x(x)l_{x,y}(x,c)C_1$ and $Q_2(x,c) = l_y(c)l_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x) = C_2 l_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x)$.Then we have

$$\frac{Q_2(x,c)}{Q_1(x,c)} = \frac{C_2 l_{x,y}(x,c)\prod_{i=0}^{n-1} p'_{i,x}(x)}{l_x(x)l_{x,y}(x,c)C_1} = \frac{C_2}{C_1}\frac{\prod_{i=0}^{n-1} p'_{i,x}(x)}{l_x(x)} \in R[x]$$

where $Q_1(x,c)$ and $Q_2(x,c)$ are polynomials only of variable $x$. Thus, the lcm will be

$$Q_2(x,y)\frac{Q_1(x,c)}{Q_2(x,c)} =$$
$$= \left[l_y(y)l_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,x}(x)\right]\left[\frac{C_1}{C_2}\frac{l_x(x)}{\prod_{i=0}^{n-1} p'_{i,x}(x)}\right] =$$
$$= \frac{C_1}{C_2} l_y(y)l_{x,y}(x,y)l_x(x) = L(x,y)$$

*Algorithm 6: DFT computation of the lcm of two-variable polynomials*

Step 1. Compute the $Q_1(x,y) = l_x(x)l_{x,y}(x,y)\prod_{i=0}^{n-1} p'_{i,y}(y)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $y$.

Step 2. Compute the $Q_2(x,y) = l_y(y) l_{x,y}(x,y) \prod_{i=0}^{n-1} p'_{i,x}(x)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $x$.

Step 3. Compute the 1-d polynomials $Q_1(x,c)$ and $Q_2(x,c)$.

Step 4. The CGD is $G(x,y) = Q_2(x,y)\frac{Q_1(x,c)}{Q_2(x,c)} = Q_1(x,y)\frac{Q_2(c,y)}{Q_1(c,y)}$.

The Algorithm 5 is applied in the following example in order to compute the GCD of 2 two-variable polynomials.

*Example 7:* Consider two polynomials $A(x,y) = (x+1)(x+1)(x+y)y$ and $B(x,y) = (x+1)(x+2)(x+y)y$. Then by applying the DFT algorithm 5 we have

*Step 1. Compute $Q_1(x,y)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $x$:*

$$Q_1(x,y) = (x+1)(x+y)y^2$$

*Step 2. Compute $Q_2(x,y)$ by (1) with the interpolation points multiplied by arbitrary real numbers $c_1$ and $c_2$, first by the variable $y$:*

$$Q_2(x,y) = (x+1)^2(x+y)y$$

*Step 3. Compute the values of the polynomials $Q_1(x,c)$* and $Q_2(x,c)$ where $c$ is an arbitrary number*:*

$$\begin{aligned} Q_1(x,1.27) &= 1.6129(1+x)(1.27+x) \\ Q_2(x,1.27) &= 1.27(1+x)^2(1.27+x) \end{aligned}$$

*Step 4. The GCD we are looking for, is*

$$\begin{gathered} G(x,y) = Q_2(x,y)\frac{Q_1(x,1.27)}{Q_2(x,1.27)} = \\ = (x+1)^2(x+y)y\frac{1.6129(1+x)(1.27+x)}{1.27(1+x)^2(1.27+x)} = \\ = (x+y)y\frac{1.6129(1+x)}{1.27} = \frac{1.6129}{1.27}(1+x)(x+y)y = \\ = 1.27(1+x)(x+y)y \end{gathered}$$

We present here only the computation of $Q_2(x,y)$. The computation of $Q_1(x,y)$ is similar.

*Step 1.* Calculate the number of interpolation points $b_m$ by (7).

$$\begin{gathered} b_0 = \min\left\{ \begin{array}{c} \deg_x[(x+1)(x+1)(x+y)y], \\ \deg_x[(x+1)(x+2)(x+y)y] \end{array} \right\} = 3 \\ b_1 = \deg_y[(x+1)(x+1)(x+y)y] + \\ + \deg_y[(x+1)(x+2)(x+y)y] = 4 \end{gathered}$$

*Step 2. Compute the*

$$R_1 = \prod_{i=0}^{1}(b_i+1) = (3+1)(4+1) = 20$$

Let $c_1 = c_2 = 5$. Then we will have

$$\begin{aligned} u_0(0) &= 5W_0^0 = 5 \\ u_0(1) &= 5W_0^{-1} = 5e^{-\frac{2\pi j}{4}} = 5j \\ u_0(2) &= 5W_0^{-2} = 5e^{-\frac{4\pi j}{4}} = -5 \\ u_0(3) &= 5W_0^{-3} = 5e^{-\frac{6\pi j}{4}} = -5j \end{aligned}$$

$$\begin{aligned} u_1(0) &= 5W_0^0 = 5 \\ u_1(1) &= 5W_0^{-1} = 5e^{-\frac{2\pi j}{5}} \\ u_1(2) &= 5W_0^{-2} = 5e^{-\frac{4\pi j}{5}} \\ u_1(3) &= 5W_0^{-3} = 5e^{-\frac{6\pi j}{5}} \\ u_1(4) &= 5W_0^{-4} = 5e^{-\frac{8\pi j}{5}} \end{aligned}$$

*Step 3.* Determine the GCD of the polynomials $A(u_0(r_0),y)$, $B(u_0(r_0),y)$ multiplied by the coefficients of the greatest power of $y$ of polynomials $A(u_0(r_0),y)$, $B(u_0(r_0),y)$

$$\begin{gathered} A(5,y) = 36y^2 + 180y, B(5,y) = 42y^2 + 210y \Rightarrow \\ \Rightarrow p(5,y) = (y^2+5y)\cdot 36 \cdot 42 = 1512y^2 + 7560y \\ A(5j,y) = -(24-10j)y^2 - (50+120j)y, B(5,y) = \\ = -(23-15j)y^2 - (75+115j)y \Rightarrow \\ \Rightarrow p(5j,y) = (y^2+5jy)\cdot(-24+10j)\cdot(-23+15j) = \\ = (402-590j)y^2 + (2950+2010j)y \\ A(-5,y) = 16y^2 - 180y, B(5,y) = 12y^2 - 60y \Rightarrow \\ \Rightarrow p(-5,y) = (y^2-5y)\cdot 16 \cdot 12 = 192y^2 - 960y \\ A(-5j,y) = -(24+10j)y^2 - (50-120j)y, B(5,y) = \\ = -(23+15j)y^2 - (75-115j)y \Rightarrow \\ \Rightarrow p(-5j,y) = (y^2-5jy)\cdot(-24-10j)\cdot(-23-15j) = \\ = (402+590j)y^2 + (2950-2010j)y \end{gathered}$$

and then the values at $u_0(r_0)$ of each polynomial

$$\begin{array}{cc} \tilde{p}_{0,0} = 75600 & \tilde{p}_{1,0} = 24800 - 4700i \\ \tilde{p}_{0,1} = -18900. - 58168.2i & \tilde{p}_{1,1} = -2684.33 - 4896.7i \\ \tilde{p}_{0,2} = -18900. + 13731.7i & \tilde{p}_{1,2} = 11107.9 - 11800.3i \\ \tilde{p}_{0,3} = -18900. - 13731.7i & \tilde{p}_{1,3} = -28762.7 - 13576.9i \\ \tilde{p}_{0,4} = -18900. + 58168.2i & \tilde{p}_{1,4} = -4460.91 + 34973.9i \\ \tilde{p}_{2,0} = 0 & \tilde{p}_{3,0} = 24800 + 4700i \\ \tilde{p}_{2,1} = -5366.56 + 1743.7i & \tilde{p}_{3,1} = -4460.91 - 34973.9i \\ \tilde{p}_{2,2} = 5366.56 + 7386.44i & \tilde{p}_{3,2} = -28762.7 + 13576.9i \\ \tilde{p}_{2,3} = 5366.56 - 7386.44i & \tilde{p}_{3,3} = 11107.9i + 11800.3i \\ \tilde{p}_{2,4} = -5366.56 - 1743.7i & \tilde{p}_{3,4} = -2684.33 + 4896.7i \end{array}$$

and thus construct the values $\tilde{p}_{r_0,r_1}$ in (12).

Step 4. Use the inverse DFT (13) for the points $\tilde{p}_{r_0,r_1}$ in order to construct the values $p_{l_0,l_1} = \frac{1}{20}\sum_{r_0=0}^{3}\sum_{r_1=0}^{4}\tilde{p}_{r_0,r_1}\widetilde{W}_0^{r_0 l_0}\widetilde{W}_1^{r_1 l_1}$.

$$\begin{aligned} &p_{0,0}=0, p_{0,1}=0, p_{0,2}=0, p_{0,3}=1, p_{0,4}=0\\ &p_{1,0}=0, p_{1,1}=0, p_{1,2}=1, p_{1,3}=1, p_{1,4}=0\\ &p_{2,0}=0, p_{2,1}=0, p_{2,2}=1, p_{2,3}=0, p_{2,4}=0\\ &p_{3,0}=0, p_{3,1}=0, p_{3,2}=0, p_{3,3}=0, p_{3,4}=0 \end{aligned}$$

and thus $Q_2(x,y)$ is

$$\begin{aligned} Q_2(x,y) &= \sum_{k_0=0}^{3}\sum_{k_1=0}^{4} p_{k_0,k_1}x^{k_0}y^{k_1} = \\ &= y^3 + xy^2 + xy^3 + x^2y^2 = \\ &= (x+1)(x+y)y^2 \end{aligned}$$

## III. Conclusions

An algorithm for the computation of the GCD and LCM of two-variable polynomials have been developed, based on DFT technicues. It has the main advantage of speed and robustness that all DFT techniques have. The proposed algorithm has been implemented in the *Mathematica* computer programming language and can be requested by the corresponding author. The above algorithm can be extended to the 2-D polynomial matrix case by using existing methods of computation of GCD of 1-D polynomial matrices.

## References


[1] Atiyah M.F. and McDonald I.G., 1964, Introduction to Commutative Algebra (Reading, MA:Addison-Wesley).

[2] Galkowski Krzysztof, 1996, Matrix Description of Multivariable Polynomials, Linear Algebra and its Applications, Vol.234, 209-226.

[3] Gantmacher F.R., 1959, The theory of matrices, New York, Chelsea.

[4] Givone D. D. and Roesser R.P., 1973, Minimization of multidimensional linear iterative circuits, IEEE Trans. Comput., Vol.C-22. pp. 673-678.

[5] Karampetakis N. P., Tzekis P., 2005, On the computation of the minimal polynomial of a polynomial matrix, Int. J. Appl. Math. Comput. Sci., Vol. 15, No. 3, 339–349.

[6] 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

[7] Karcanias N. and Mitrouli, M., Numerical computation of the least common multiple of a set of polynomials, Reliable Computing, Issue 4, Vol. 6 (2000) pp. 439-457.

[8] Karcanias N. and Mitrouli M., A Matrix Pencil Based Numerical Method for the Computation of the GCD of Polynomials, IEEE Trans. Autom. Cont., Vol. 39 (1994), 977-981.

[9] Mitrouli M. and Karcanias N., Computation of the GCD of polynomials using Gaussian transformation and shifting, Int. Journ. Control 58 (1993), 211-228.

[10] Noda M. and Sasaki T., Approximate GCD and its applications to ill-conditioned algebraic equations, Jour. of Comp. and Appl. Math. 38 (1991), 335-351.

[11] Pace I. S. and Barnett S., Comparison of algorithms for calculation of GCD of polynomials, Int. Journ. System Scien. 4 (1973), 211-226.

[12] Paccagnella, L. E. and Pierobon, G. L., 1976, FFT calculation of a determinantal polynomial, IEEE Trans. on Automatic Control, June, pp.401-402.

[13] Schuster, A. and Hippe, P., 1992, Inversion of polynomial matrices by interpolation, IEEE Trans. Automat. Control, Vol.37, No.3, pp.363-365.