International Journal of Computer Research ISSN 1535-6698
Volume 12, Number 2, pp. 223-231 © 2003 Nova Publishers, Inc

Separating Two-Dimensional All-Zero Polynomials Using Genetic Algorithms

Ioannis F. Gonos
High Voltage Laboratory, Department of Electrical Engineering
National Technical University of Athens
9 Iroon Politechniou St., Athens,
Greece.
email: igonos@softlab.ntua.gr

George E. Antoniou
Center for Imaging and Optics, Department of Computer Science
Montclair State University
Montclair, New Jersey
USA
email: george.antoniou@montclair.edu

Abstract: In this paper a new approach is presented for the approximate separation of two-dimensional all-zero polynomials. The proposed technique is based on the genetic search algorithm. The implemented genetic algorithm computes the approximate one-dimensional coefficients, the absolute error and useful parameters of the genetic algorithm, using the software package MATLAB. Three examples along with the simulations are presented to illustrate the efficiency of the algorithm.

Key-Words: Genetic algorithm, Two-Dimensions, Polynomials, Optimization

1 Introduction

The study of n-dimensional (n-D) signals and systems continues to attract a large number of researchers from different fields due to the enormous number of applications. Two-dimensional (2-D) systems have two independent variables propagated into vertical and horizontal direction with applications in the areas of signal, and image processing, system theory, geophysics, repetitive processes etc. [1]-[3].

In this paper the exact or approximate separation of 2-D all-zero polynomial is considered using a genetic algorithm. It is well known that this problem does not always have an exact solution due to the absence of fundamental theorem of algebra for polynomials with more than one independent variable [4]. Also it is noted that various techniques applicable to 1-D polynomials/filters cannot be generalized to many dimensions. This problem has been considered in the past. In [5] a Sigma-Pie neural network - based algorithm was proposed, and recently the simulated annealing algorithm was used to compute the separated one-dimensional (1-D) factor coefficients [6]. The proposed technique gives improved error results over the other two approaches requiring no initial estimate of the coefficients.

2 Genetic Algorithms

Genetic algorithms (GAs) are computational models inspired to biological evolution, aimed at probabilistic search. GAs are robust, stochastic and heuristic optimization methods, based on biological reproduction processes. Generally speaking, GAs are search algorithms based on the mechanics of natural selection and natural genetics: reproduction of an original population, performance of crossover and mutation, selection of the best. Artificial reproduction schemes were first developed in the 70's [7] and were more extended during the 80's [8, 9]. The search area for the genetic algorithms is very wide and it usually converges to a point near the global optimum. Fine tuning operations like hill climbing can be used to further refine the near-optimum solution. Simulating the survival of the most suitable among string structures, the optimal string (solution) is searched by randomised information exchange. Since genetic algorithm utilises the coded discrete information of the artificial strings, it can be applied to ill-structured discrete optimisation problems as well as to continuous optimisation ones. They are often used in optimization problems, since they are capable to find solutions of complex problems without carrying out an exhaustive search. Therefore, GAs are very often employed in non-linear problems and multi-objective optimizations.

GAs combine the adaptive nature of the natural genetics or the evolution procedures of organs with functional optimisations. An initial population is provided. This is represented, by bit strings that evolve randomisation through successive generations in order to obtain an optimum for a particular fitness function. In each generation, a new set of artificial strings is created using bits and pieces of the most suitable of the old ones. Solutions with high suitability are mated with other solutions by crossing over parts of solution strings. Strings may also mutate. Solutions with poor fitness are improved by crossover using highly fit solutions.

A simple genetic algorithm relies on the processes of reproduction, crossover and mutation to reach the global or "near-global" optimum. Starting the search, GAs require the initial set of the points. This set is called population, analogous to the biological system. It has a population size P_s. A random number generator creates the initial population. This initial set is converted to a binary system and is considered as chromosomes, actually sequences of "0" and "1". The next step is to form pairs of these points that will be considered as parents for a reproduction. Parents come to reproduction and interchange N_p parts of their genetic material. This is achieved by crossover. Crossover is used to create two new individuals children from two existing individual parents picked from the current population. After the crossover there is a very small probability P_m for mutation. Mutation is the phenomenon where a random "0" becomes "1" or a "1" becomes "0". Mutation is necessary because although reproduction and crossover efficiently search and mix existing strings, occasionally they may result in loss of some potentially useful "genetic" material.

Assume that each pair of "parents" gives N_c children. Thus the genetic algorithm generates the initial layouts and obtains the objective function values. The above operations are carried out and the next generation with a new population of strings is formed. By the reproduction, the population of the "parents" are enhanced with the "children", increasing the original population since new members are added. The parents always belong to the considered population. The new population has now $P_s+N_c \cdot P_s/2$ members. Then the process of natural selection is applied. According to this process only P_s members survive out of the $P_s+ N_c \cdot P_s/2$ members. These P_s members are selected as the members with the higher values of F, if we attempt to achieve maximisation of F, or with the lower values of F, if we attempt to achieve minimisation of F. Repeating the iterations of reproduction under crossover and mutation and natural selection GAs can find the minimum (or maximum) of F. The best values of the population converge at this point. The termination criterion is fulfilled if either the mean value of F in the P_s-members population is no longer improved (maximised or minimised) or the number of iterations is greater than the maximum number of iterations N_{max}.

By random procedure, this population is split into pairs of parents that will be crossed, i.e. they will interchange their genetic material (with N_p parts for crossovers) always under a very small probability p for mutation. By this reproduction, a new population of $P_s + N_c \cdot P_s/2$ members will be formed, since each pair of parents gives birth to k children. The new

International Journal of Computer Research ISSN 1535-6698
Volume 12, Number 2, pp. 223-231 © 2003 Nova Publishers, Inc

population is filtered and only the P_s better members remain into the population. "Better" means here the $2P_s$ lower values of F(A, n). The others are erased. Repeating the iterations of reproduction, under crossover and mutation, and the natural selection, GAs can find the minimum of the function F that is the point where the best values of the population converge.

3. Statement of the Problem and Algorithm

Let us consider the following 2-D all-pole filter/polynomial:

$$f(z_1,z_2)=\sum_{i=0}^{N_1}\sum_{j=0}^{N_2}a_{ij}z_1^i z_2^j \qquad (1)$$

where $a_{ij}\in R$.

The purpose of this paper is to use genetic algorithms to determine the coefficient's b_i and c_j, such that the following equation holds:

$$f(z_1,z_2)\approx f'(z_1,z_2)=\left(\sum_{i=0}^{N_1}b_i z_1^i\right)\left(\sum_{j=0}^{N_2}c_j z_2^j\right) \qquad (2)$$

with $b_i\in R\ \ \forall i:0\le i\le N_1$ and $c_j\in R\ \ \forall j:0\le j\le N_2$

For the above equations, the values of b_i and c_j are calculated using GAs. The GAs minimize the function $\left|f(z_1,z_2)-f'(z_1,z_2)\right|$ for z_1 and $z_2$values ranging from $(-\pi,\pi)$.

This set of equations must be minimized over b_i and c_j. This is why b_i and c_j are converted to the binary system and are considered as parts of a big chromosome. The search starts with a randomly generated population of such $2P_s$ chromosomes. Each coefficient (b_i and c_j) is converted to a t-bits binary number. 2t bits are required for the "chromosome" of b_i and c_j with $-5 < b_i < 5$, $-5 < c_j < 5$.

The algorithm is summarised as follows:

- *Step 1*: Find randomly the P_s initial t bits binary numbers
- *Step 2*: Convert the initial binary numbers to the initial population of P_s members
- *Step 3*: Increase the number of iterations
- *Step 4*: Select randomly the $P_s/2$ pairs from the population
- *Step 5*: From each pair of parents and take k children by crossover. Each bit of each child has probability P_m for a mutation
- *Step 6*: Find the new population $P_s+N_c\cdot P_s/2$ (parents + children)
- *Step 7*: From the new population select the P_s members with the lower values of Step 6.
- *Step 8*: If the number i of iterations is less than the maximum number N_{max} of iterations then go to Step 3, else STOP.
- *Stop*

In the following section three examples are presented illustrating the efficiency of the proposed algorithm.

International Journal of Computer Research ISSN 1535-6698
Volume 12, Number 2, pp. 223-231 © 2003 Nova Publishers, Inc

4 Illustrative Examples

4.1 Example 1

Consider the following all-zero polynomial:

$$F(z_1, z_2) = 15\, z_1 z_2 + 5\, z_1 + 3\, z_2 + 1 \qquad (3)$$

Applying our genetic algorithm the new approximate polynomial takes the following form:

$$F(z_1, z_2) \approx (1 + 5\, z_1)(1 + 3z_2) \qquad (4)$$

For this example the absolute error is 0.0, and the genetic parameters Ps, t, Nc, Pm, Np, Nmax are,

Ps	T	Nc	Pm	Np	Nmax
20	16	4	5 %	6	50

The convergence of the absolute error and the coefficients b_1 and c_1 are given in the Figure 1.

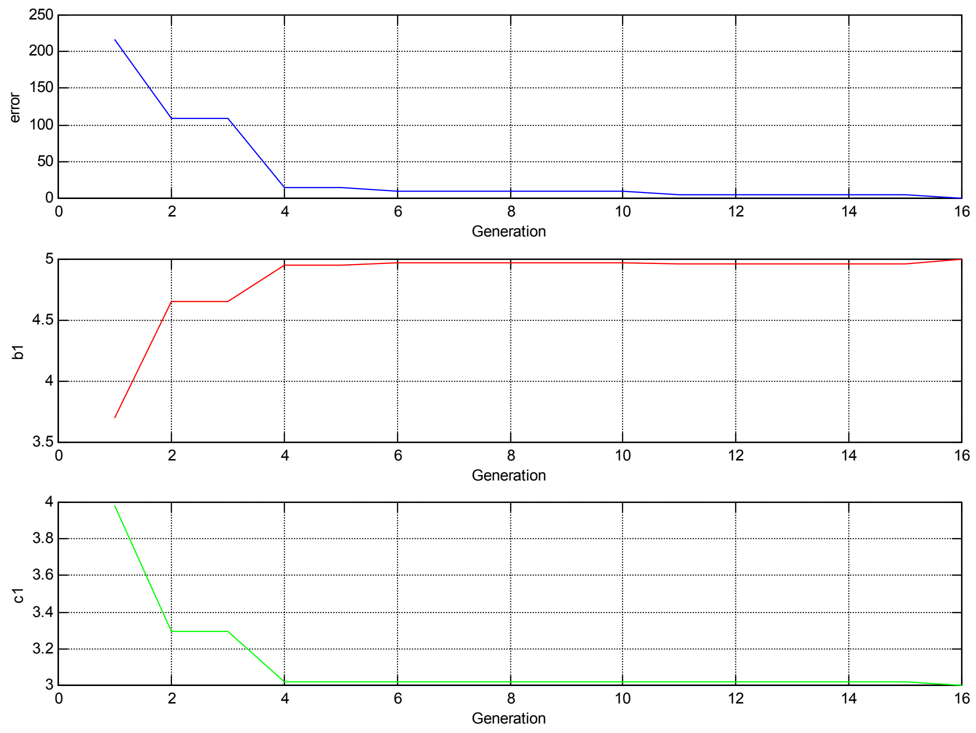


Figure 1. The convergence of the absolute error and the coefficients b_1 and c_1

ISSN 1535-6698
© 2003 Nova Publishers, Inc

4.2 Example 2

Consider the following 2-D all-zero polynomial:

$F(z_{1,} z_2) = 15\, z_1 z_2 + 4\, z_1 + 3\, z_2 + 1$ (5)

Applying our genetic algorithm the new approximate polynomial takes the following form:

$F(z_{1,} z_2) \approx (1 + 4.2954\, z_1)(1 + 3.41207\, z_2)$ (6)

For this example the alsolute error is 52.13 and the genetic parameters Ps, t, Nc, Pm, Np, Nmax are,

Ps	t	Nc	Pm	Np	Nmax
20	16	4	4 %	6	50

The convergence of the absolute error and the coefficients b_1 and c_1 are given in the Figure 2.

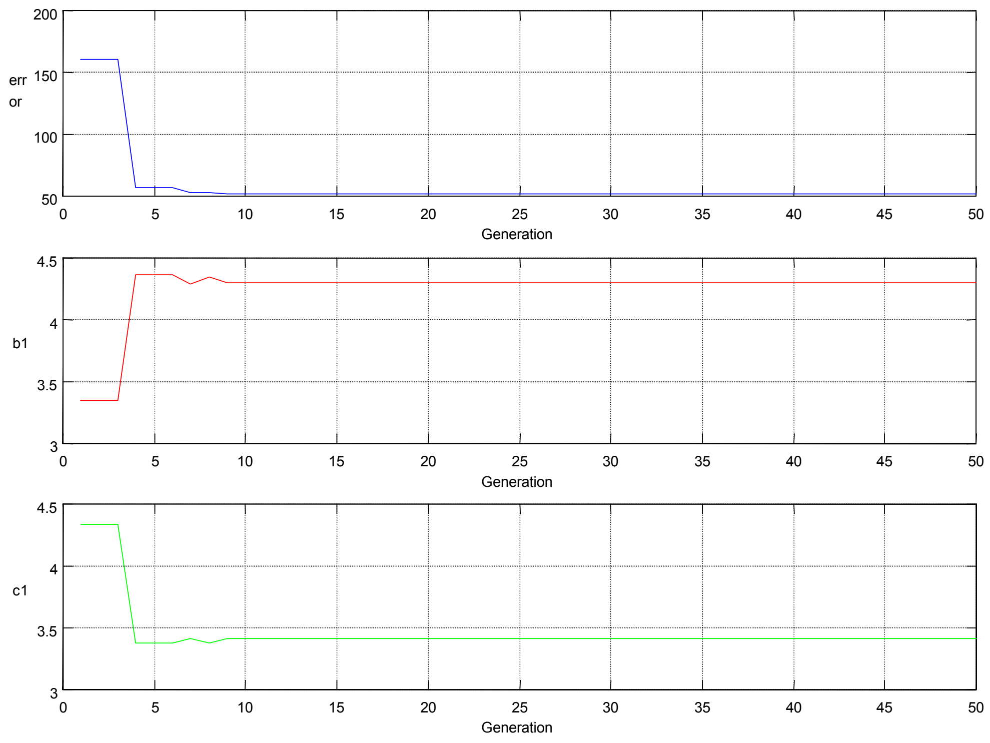


Figure 2. The convergence of the absolute error and the coefficients b_1 and c_1

The mesh and their contours plots for this example are given in Figure 3.

International Journal of Computer Research
Volume 12, Number 2, pp. 223-231

ISSN 1535-6698

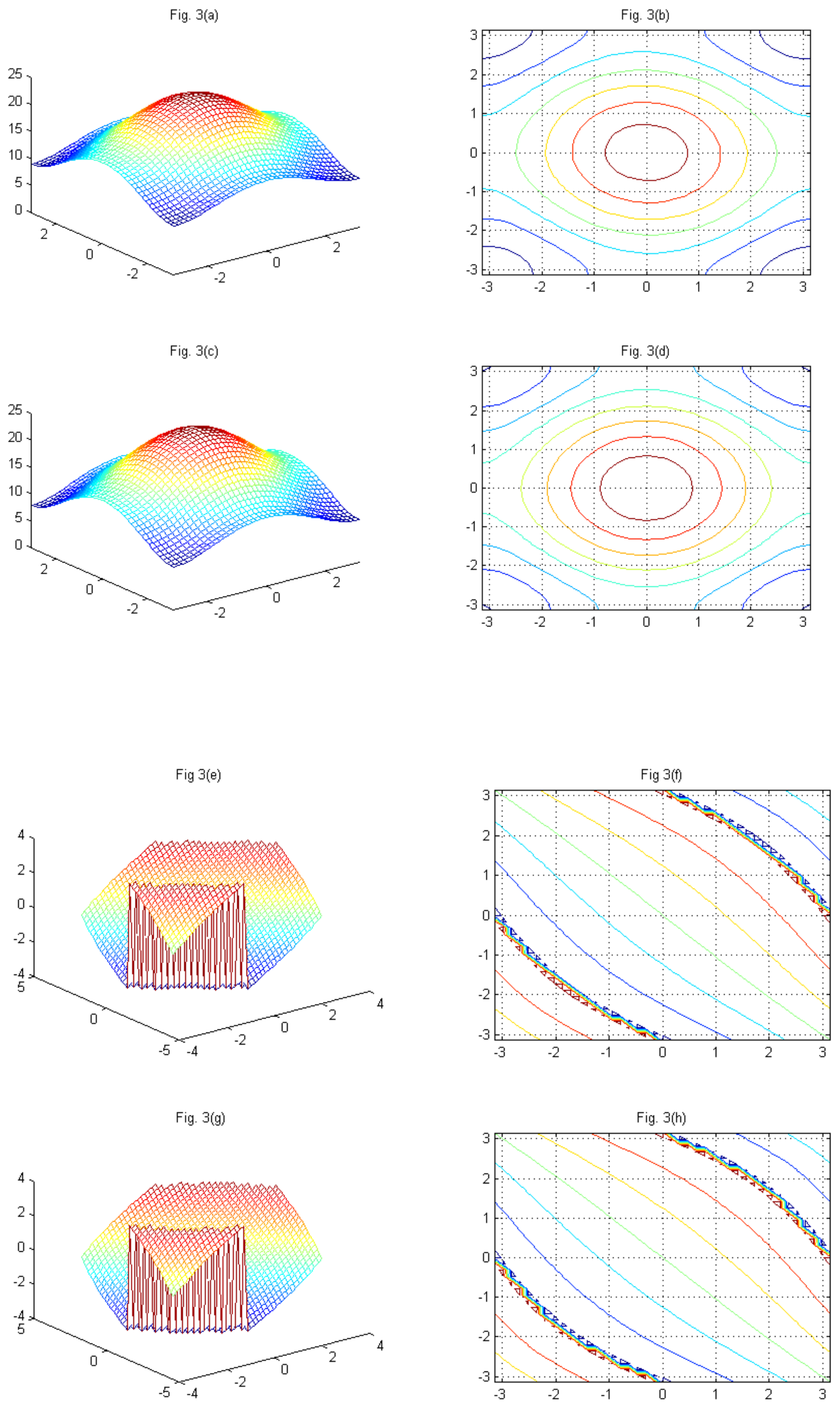


Figure 3. 3(a) and 3(c) show the difference between the mesh plot of the original (5) and aproximated (6) polynomials, and 3(b) and 3(d) show the contour plots for the same.

3(e) and 3(h) show the difference between phase plot of the original and approximated polynomial, and 3(f) and 3(h) show the contour plots for the same.

4.3 Example 3

Consider the folowing 2-D all-zero polynomial:

$F(z_{1,} z_2) = -15 z_1z_2 - 4 z_1 + 3 z_2 + 1$ (7)

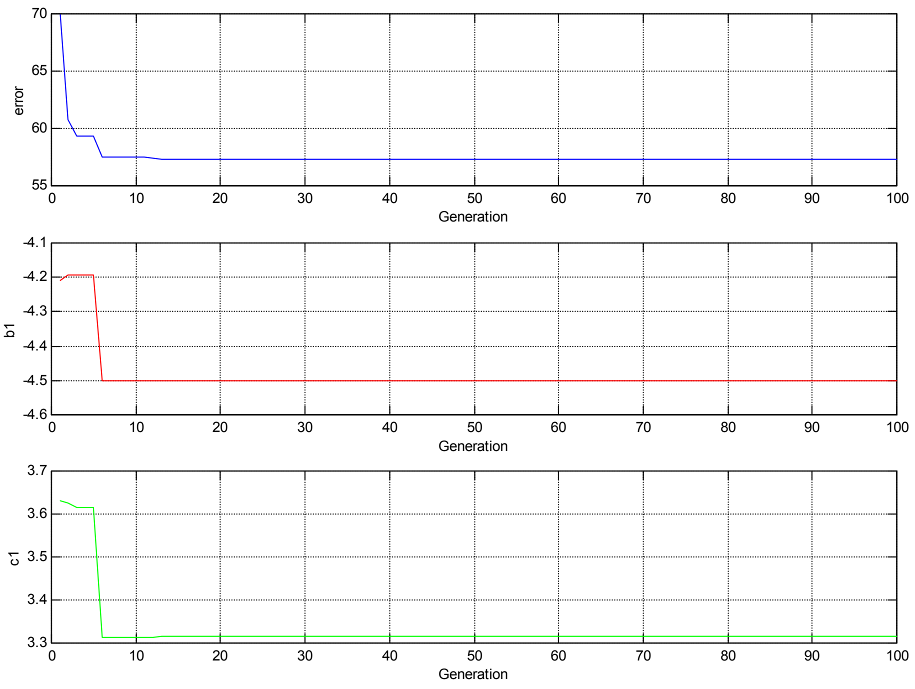
Applying our genetic algorithm the new approximate polynomial takes the following form:

$F(z_{1,} z_2) \approx (1 - 4.50103 z_1)(1+ 3.31556 z_2)$ (8)

For this example the absolute error is 57.29, and the genetic parameters Ps, t, Nc, Pm, Np, Nmax are,

Ps	t	Nc	Pm	Np	Nmax
20	16	4	3 %	6	100

The convergence of the absolute error and the coefficients b_1 and c_1 are given in the Figure



4.

Figure 4. The convergence of the absolute error and the coefficients b_1 and c_1

The mesh and their contours plots for this example are given in Figure 5.

International Journal of Computer Research
Volume 12, Number 2, pp. 223-231

ISSN 1535-6698
© 2003 Nova Publishers, Inc

Fig. 5(a)

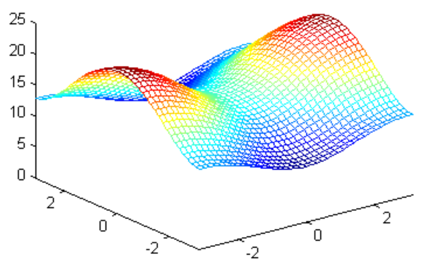


Fig. 5(b)

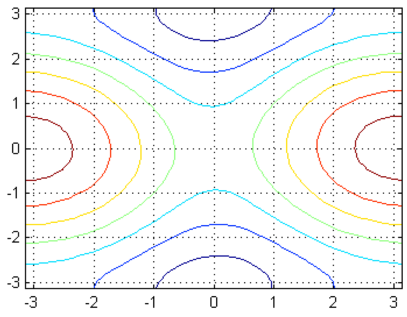


Fig. 5(c)

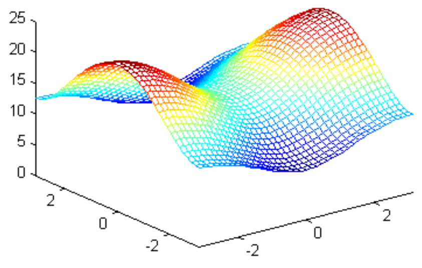


Fig. 5(d)

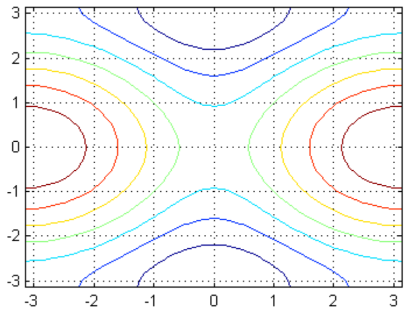


Fig 5(e)

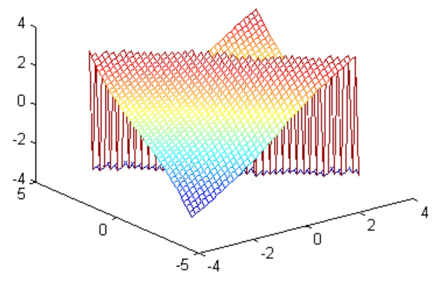


Fig 5(f)

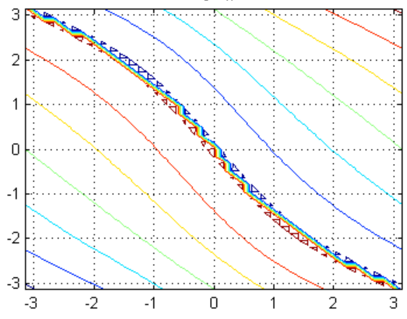


Fig. 5(g)

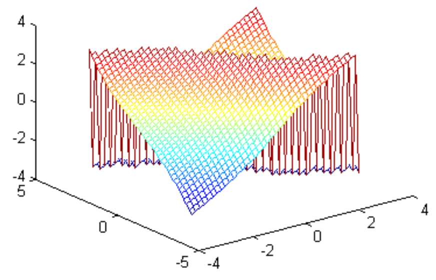


Fig. 5(h)

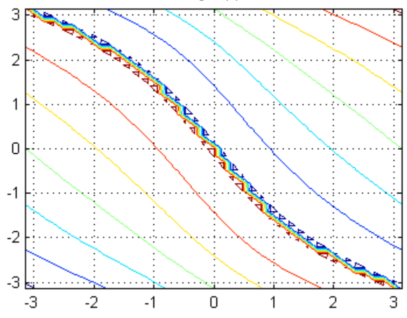


Figure 5. 5(a) and 5(c) show the difference between the mesh plot of the original (7) and aproximated (8) polynomials, and 5(b) and 5(d) show the contour plots for the same.
5(e) and 5(h) show the difference between phase plot of the original and approximated polynomial, and 5(f) and 5(h) show the contour plots for the same.

5 Acknowledgements

George Antoniou thanks Dr. Sokol for supporting his research through the "Margaret and Herman Sokol Faculty Award".

6 Conclusion

In this paper a genetic based algorithm was used for determining the approximate 1-D coefficients of a 2-D all-zero polynomial. The implemented algorithm provides the user with the 1-D approximate coefficients along with the computed absolute error and other useful generic parameters.
The results of this paper can easily be extended to higher dimensions. Also it is noted that the proposed algorithm can be used to find the coefficients of a 2-D all-zero polynomial that can be separated exactly.

References:

[1] Mastorakis, N, (editor) *Recent advances in circuits and systems*, WSEAS, 2000.
[2] Kaczorek, T, *Two dimensional linear systems*, Springer-Verlag, 1985.
[3] Lim, J.E, *Two dimensional signal and image processing*, Prentice-Hall, 1990.
[4] Bose, N.K, *Applied multidimensional systems theory*, Van Nostrand Reinhold, 1982.
[5]Antoniou, G.E., McMahan, L, Dhruve, R. and Stephan, S., *Simulated annealing: application for separation of two-dimensional polynomials*, Proceedings of the International Conference on Imaging Science, Systems, and Technology, 2000, Las Vegas, USA, pp. 387-393.
[6] Hormis Raju, et. al., *Separation of two-dimensional polynomials via a sigma-pie neural network*', Proceedings of the International Conference on Modelling and Simulation, Pittsburg, USA, 1995, pp. 304-306.
[7] Holland H., *Adaptation in natural and artificial systems*, Univ. of Michigan Press. Reprinted in MIT Press, 1992.
[8] Goldberg D.E., *Genetic algorithms in search, optimisation, and machine learning*, Addison- Wesley, 1989.
[9] Krishnakumar K. and Melkote S.N., *Machining fixture layout optimization using the genetic algorithm*, Int. Journal of Machine Tools & Manufacture, **40**, 2000, pp. 579-598.