www.ijramr.com

International Journal of Recent Advances in Multidisciplinary Research

Vol. 02, Issue 09, pp.0769-0773, September, 2015

ISSN : 2350-0743

RESEARCH ARTICLE

SOLVING SYSTEM OF DIFFERENTIAL EQUATIONS WITH GENETIC PROGRAMMING

Wahed, M.E., *Eldaken, O.M. and Abd-alsalam Z. A.

[1]Department of Computer Sciences, Faculty of computers and information Suez Canal University, Suez Canal, Egypt

[*2,3]Department of Mathematics, Faculty of Science, Suez Canal University, Suez Canal, Egypt

ARTICLE INFO

Article History:

Received 01st June 2015

Received in revised form

24th July, 2015

Accepted 16th August, 2015

Published online 30th September, 2015

Keywords:

Differential equations,

Genetic programming

Grammatical evolution.

ABSTRACT

In this work, a novel hybrid method for the solutions of system of ordinary differential equations is presented here.The method creates trial solutions in genetic programming on grammatical evolution.The trial solutions are enhanced periodically using a local optimization procedure.

INTRODUCTION

Recently there are problems in many scientific fields by using ordinary differential equations in physics (Gear, 1984; Campbell, 1989; Ascher, 1989;), as well as chemistry (Ascher and Petzold, 1991; Campbell *et al.*, 1994), biology (Petzold, 1995; Wazwaz, 1995), economics (Ascher and Lin, 1996), and other fields. A new genetic programming [2], was created to optimize process based on the evolution of a large number of solutions through geneticoperations such as replication, crossover and mutation (Ascher and Lin, 1997). This new methods choose a basis set of functions with adjustable parameters and proceedapproximating the solution by varying these parameters.

Our method offers closed formsolutions, but the variety of the basis functions involved is not determined, rather than constructed dynamically as the solution proceeds and can be of high complexityif required. This feature is the one that makes our method different from others. We have notdeal with any problem of differential equation induction from data. The previous studies are achievedwith the help of grammatical evolution. We used grammatical evolution instead the “classic”tree based genetic programming, because grammatical evolution can produce programs in anylanguage, the genetic operations are fastas well far more convenient to symbolically differentiate expressions. Thecode production can performed by using a mapping process by a grammar expressed. Grammatical evolution has been applied to solve a lot of complex problems suchas symbolic regression (Campbell, 1989).

**Corresponding author: Eldaken,*

Department of Computer Sciences, Faculty of computers and information Suez Canal University, Suez Canal, Egypt

The organized of this article well be as follows: in Section2 abrief description of the grammatical evolution algorithms given followed by analytical description of the proposed method. The test functions used in the experiments followed by the experimental results are outlined.

System Ordinary Differential Equation

The most general form of differential-algebraic equations (DAEs) is given by

$$F(x, y, y') = 0 \qquad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

With initial values

$$y(x_0) = y_0 , \; y'(x_0) = y_1$$

Where F and y is a vector function for which we assumed sufficient differentiability $\frac{\partial F}{\partial y}$ may be singular. The rank and structure of this jacobian matrix may depend, in general, on the solution $y(x)$, and for simplicity we will always assume that it is independent of x . The important special case of semi-explicit differential algebraic equations(DAEs), or an ODE with constraints,

$$\bar{y} = f(x, y, z) \qquad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

$$0 = g(x, y, z) \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

This is a special case of (1). The index is 1 if $\frac{\partial g}{\partial z}$ is nonsingular, because then one differentiation of (3) yields z' in principle. For the semi-explicit index-1 DAE we can distinguish between differential variables $y(x)$ and algebraic variables $z(x)$ [4]. The algebraic variables may be less smooth than the differential variables by one derivative. Differential-algebraic equations

(DAEs) (1) can be written in the semi explicit form (2). These types of systems arise, for example, in circuit analysis, chemical process simulation, power systems, and many other applications. The Genetic programming method has been applied to problems in physics, biology and chemical reactions. Recently, there has been a great deal of interest in applying Genetic programming technique for solving a wide class of nonlinear equations, including algebraic, differential, partial-differential, differential-delay and integral-differential equations (2;Wazwaz, 1995). We applied the Genetic programming method to approximation solution of the system differential-algebraic equations (DAEs).

Method Description

In this section a brief description of the grammatical evolution algorithm is given**.**Themainstepsoftheproposedalgorithmare outlinedwiththestepsforthefitnessevaluationfor thecasesof ODEs.

Grammatical Evolution

Grammatical evolution is an evolutionary technique that can produce code in any programming language requiring the grammar of the target language in BNF syntax and some proper fitness function. This technique has been used with success in many scientific fields such as symbolic regression (Çelik *et al.,* 2002), byreplacing non terminal symbols with the right hand of the selected production rule. Theselection is performed in two steps:

- We read an element from the chromosome (with value ***V***).
- We select the rule according to the scheme

Rule = *V*mod NR

Where NR is the number of rules for the specific non-terminal symbol. The process ofreplacing non terminal symbols with the right hand of production rules is continued untileither a full program has been generated or the end of chromosome has been reached. In thelatter case we can reject the entire chromosome or we can start over (wrapping event) fromthe first element of the chromosome. If the limit of the wrapping events is reached the chromosome is rejected by assigning to it a large fitness value, which prevents the chromosome to be used in the crossover procedure. In the proposed algorithm the limit of wrapping events was set to 2.As an example of the mapping procedure of the grammatical evolution consider the BNF grammar shown in Fig. 1. The number in parentheses denotes the sequence number of the corresponding production rule to be used in the mapping procedure. Consider the chromo some x=[9,8,7,6,16,10,17,23,8,14]. The steps of the mapping procedure are listed in Table1. The final outcome of these steps is the expression 3+sin(x).

Algorithm decription

The proposed method is baased on an evolutionary algorithm ,a stochastic process whose basis lies in the biological evolution The grammar of theproposed method

S: : = <expr> (0)
<expr>: = (<expr> <op> <expr>) (0)
| <func>(<expr>) (1)
| <terminal> (2)
<op> : := + (0)
| - (1)
| * (2)
| / (3)
<func> : := sin (0)
| cos (1)
| exp (2)
| log (3)
<digit >: := 0 (0)
| 1 (1)
| 2 (2)
| 3 (3)
| 4(4)
| 5 (5)
| 6 (6)
| 7 (7)
| 8 (8)
| 9 (9)
<terminal> : :=<xlist> (0)
| <digitlist>.<digitlist>(1)
<xlist>: :=x1(0)
| x2 (1)
| x3 (2)
<digitlist>: :=<digit> (0)
| <digit>.<digitlist> (1)

Table 1. An example of the mapping procedure

string	chromosome	operation
<expr >	9, 8, 7,6,16,10,17,23,8,14	9 mod 7 = 2
<func>(<expr>)	8,7,6,16,10,17,23,8,14	8 mod 4 = 0
sin(<expr >)	7,6,16,10,17,23,8,14	7mod 7 = 0
sin(<expr><op> <expr>)	6, 16,10,17,23,8,14	6 mod 7 = 6
sin(x <op> <expr>)	16,10,17,23,8,14	16 mod 4 = 0
sin(x) + <expr>	10, 17,23,8,14 (1)	10 mod 7 = 3
sin(x) + < digit>	17, 23,8,14	17 mod 10=7
sin(x) + 3	23,8,14	

Algorithm along with a penalty function which is used in order to represent the boundary or initial conditions of the ordinary differential equations, the main steps of the algorithm are as follows:

Algorithm
Input : function ›wer , upper bound [l b], Number of variable N
Repeat
Solve the differential equation by [t,sol]=ode45(f,[l b],N)
[x , f]= Genetic_algorithm($sol, N, [Lb]$)
$Err = f - fold$
Until $Err < \varepsilon$
Plot the result with exact

Example 1: We consider the following system of differential-algebraic

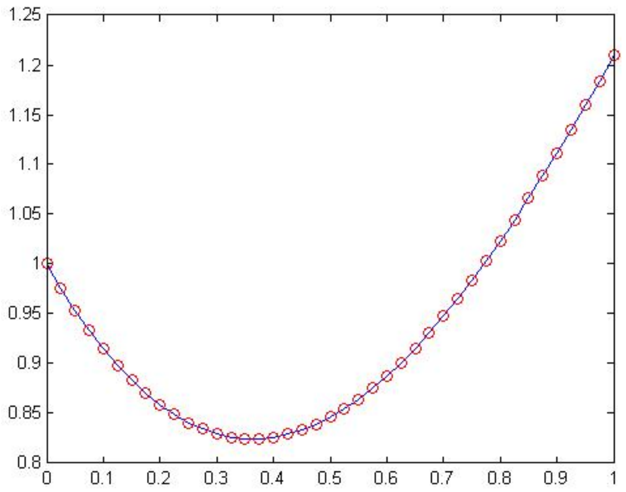
Equations (DAEs)
u' – xv' +u – (1+x)v = 0
v = sin x(10)

With initial condition u(0) = 1, v(0) = 0

The exact solution is u(x) = $e^{-x} + xsinx$

v(x) = sin x
Eq.(10)can be written
u¢ = -*u* + *x* cos *x* + (1+ *x*) sin *x*

X	App by using Genetic	Exact	Error
0	1.0000	1.0000	0
0.0250	0.9759	0.9759	-0.1023
0.0500	0.9537	0.9537	-0.1709
0.0750	0.9334	0.9334	-0.0866
0.1000	0.9148	0.9148	0.0088
0.1250	0.8981	0.8981	-0.0844
0.1500	0.8831	0.8831	-0.1552
0.1750	0.8699	0.8699	-0.0795
0.2000	0.8585	0.8585	0.0164
0.2250	0.8487	0.8487	-0.0670
0.2500	0.8407	0.8407	-0.1389
0.2750	0.8342	0.8342	-0.0725
0.3000	0.8295	0.8295	0.0228
0.3250	0.8263	0.8263	-0.0499
0.3500	0.8247	0.8247	-0.1220
0.3750	0.8246	0.8246	-0.0654
0.4000	0.8261	0.8261	0.0280
0.4250	0.8290	0.8290	-0.0333
0.4500	0.8334	0.8334	-0.1045
0.4750	0.8391	0.8391	-0.0584
0.5000	0.8462	0.8462	0.0321
0.5250	0.8547	0.8547	-0.0172
0.5500	0.8644	0.8644	-0.0866
0.5750	0.8754	0.8754	-0.0512
0.6000	0.8876	0.8876	0.0350
0.6250	0.9009	0.9009	-0.0017
0.6500	0.9154	0.9154	-0.0683
0.6750	0.9310	0.9310	-0.0440
0.7000	0.9475	0.9475	0.0369
0.7250	0.9651	0.9651	0.0131
0.7500	0.9836	0.9836	-0.0498
0.7750	1.0030	1.0030	-0.0367
0.8000	1.0232	1.0232	0.0377
0.8250	1.0442	1.0442	0.0271
0.8500	1.0660	1.0660	-0.0311
0.8750	1.0885	1.0885	-0.0293
0.9000	1.1116	1.1116	0.0375
0.9250	1.1353	1.1353	0.0401
0.9500	1.1595	1.1595	-0.0124
0.9750	1.1842	1.1842	-0.0220
1.0000	1.2094	1.2094	0.0363



$y_1(x)$ **is approximation solution of** $y_1(x)$

Example 2: We consider the following system of differential-algebraicequations (DAEs)

$$\begin{pmatrix} 1 & -x & x^2 \\ 0 & 1 & -x \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} y_1' \\ y_2' \\ y_3' \end{pmatrix}+\begin{pmatrix} 1 & -(x+1) & x^2+2x \\ 0 & -1 & x-1 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}=\begin{pmatrix} 0 \\ 0 \\ \sin(x) \end{pmatrix}$$

With initial condition

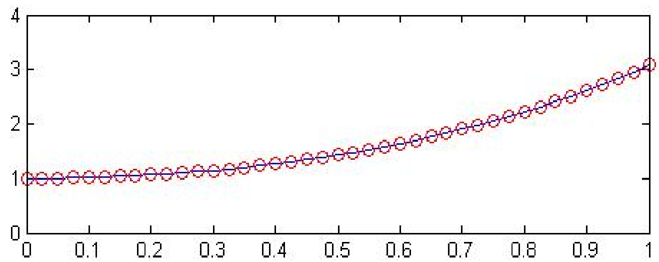
$y_1(-1) = \exp(1)-\exp(-1)$,
$y_2(-1) = \exp(-1)+\sin(1)$,
$y_3(-1) = -\sin(1)$.

The exact solution is given as

$y_1(x) = \exp(x) - x\exp(x)$,
$y_2(x) = \exp(x) + x\sin(x)$,
$y_3(x) = \sin(x)$.

Solution of Differential-Algebraic Equations

X	App $y_1$by using Genetic	Exact solution	Error 1.0e-07
0	1.0000	1.0000	-0.1000
0.0250	1.0009	1.0009	-0.1000
0.0500	1.0038	1.0038	-0.1000
0.0750	1.0086	1.0086	-0.1000
0.1000	1.0154	1.0154	-0.1000
0.1250	1.0241	1.0241	-0.1000
0.1500	1.0350	1.0350	-0.1000
0.1750	1.0479	1.0479	-0.1000
0.2000	1.0630	1.0630	-0.1000
0.2250	1.0803	1.0803	-0.1000
0.2500	1.0998	1.0998	-0.1000
0.2750	1.1216	1.1216	-0.1000
0.3000	1.1458	1.1458	-0.1000
0.3250	1.1723	1.1723	-0.1000
0.3500	1.2014	1.2014	-0.1000
0.3750	1.2329	1.2329	-0.1000
0.4000	1.2670	1.2670	-0.1000
0.4250	1.3038	1.3038	-0.1000
0.4500	1.3434	1.3434	-0.1000
0.4750	1.3857	1.3857	-0.1000
0.5000	1.4309	1.4309	-0.1000
0.5250	1.4790	1.4790	-0.1000
0.5500	1.5302	1.5302	-0.1000
0.5750	1.5846	1.5846	-0.1000
0.6000	1.6421	1.6421	-0.1000
0.6250	1.7029	1.7029	-0.1000
0.6500	1.7671	1.7671	-0.1000
0.6750	1.8349	1.8349	-0.1000
0.7000	1.9062	1.9062	-0.1000
0.7250	1.9813	1.9813	-0.1000
0.7500	2.0601	2.0601	-0.1000
0.7750	2.1429	2.1429	-0.1000
0.8000	2.2298	2.2298	-0.1000
0.8250	2.3208	2.3208	-0.1000
0.8500	2.4161	2.4161	-0.1000
0.8750	2.5159	2.5159	-0.1000
0.9000	2.6202	2.6202	-0.1000
0.9250	2.7293	2.7293	-0.1000
0.9500	2.8432	2.8432	-0.1000
0.9750	2.9621	2.9621	-0.1000
1.0000	3.0862	3.0862	-0.1000



$y_1(x)$ **is approximation solution of** $y_1(x)$

Example 3: Consider the following linear DAE of three variables

$$y_1' = \left(\alpha - \frac{1}{2-x}\right) y_1 + (2-x)\alpha y_3 + \left(\frac{3-x}{2-x}\right) exp(x)$$

$$y_2' = \left(\frac{1-\alpha}{x-2}\right) y_1 - y_2 + (\alpha - 1) y_3 + 2exp(x)$$

$$0 = (x+2)y_1 + (x^2 - 4)y_2 - (x^2 + x - 2)\exp(x)$$

Where α is a positive parameter (take α = 10). For the initial condition

$y_1(-1) = y_2(-1) = \exp(-1)$, $y_3(-1) = -\exp(-1)/3$, the exact solution is given as

$y_1(x) = y_2(x) = \exp(x)$, $y_3(x) = -\exp(x)/(x-2)$.

The numerical results are presentedin Tables 1, 2, and 3.

Table 1. The corresponding error for the second variable y1

X	Genetic App.sol. y_1	Exact solution of y_1	Error 1.0e-07
-1.0000	0.3679	0.36788	-0.1000
-0.9500	0.3867	0.38674	-0.1000
-0.9000	0.4066	0.40657	-0.1000
-0.8500	0.4274	0.42741	-0.1000
-0.8000	0.4493	0.44933	-0.1000
-0.7500	0.4724	0.47237	-0.1000
-0.7000	0.4966	0.49659	-0.1000
-0.6500	0.5220	0.52205	-0.1000
-0.6000	0.5488	0.54881	-0.1000
-0.5500	0.5769	0.57695	-0.1000
-0.5000	0.6065	0.60653	-0.1000
-0.4500	0.6376	0.63763	-0.1000
-0.4000	0.6703	0.67032	-0.1000
-0.3500	0.7047	0.70469	-0.1000
-0.3000	0.7408	0.74082	-0.1000
-0.2500	0.7788	0.7788	-0.1000
-0.2000	0.8187	0.81873	-0.1000
-0.1500	0.8607	0.86071	-0.1000
-0.1000	0.9048	0.90484	-0.1000
-0.0500	0.9512	0.95123	-0.1000
-0.0000	1.0000	1	-0.1000
0.0500	1.0513	1.0513	-0.1000
0.1000	1.1052	1.1052	-0.1000
0.1500	1.1618	1.1618	-0.1000
0.2000	1.2214	1.2214	-0.1000
0.2500	1.2840	1.284	-0.1000
0.3000	1.3499	1.3499	-0.1000
0.3500	1.4191	1.4191	-0.1000
0.4000	1.4918	1.4918	-0.1000
0.4500	1.5683	1.5683	-0.1000
0.5000	1.6487	1.6487	-0.1000
0.5500	1.7333	1.7333	-0.1000
0.6000	1.8221	1.8221	-0.1000
0.6500	1.9155	1.9155	-0.1000
0.7000	2.0138	2.0138	-0.1000
0.7500	2.1170	2.117	-0.1000
0.8000	2.2255	2.2255	-0.1000
0.8500	2.3396	2.3396	-0.1000
0.9000	2.4596	2.4596	-0.1000
0.9500	2.5857	2.5857	-0.1000
1.0000	2.7183	2.7183	-0.1000

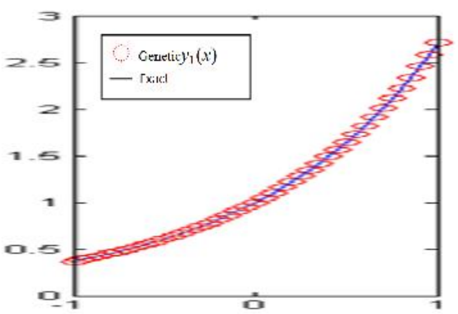


Figure 1. Values of y1 , its Genetic approximation (y'1)

Table 2. The corresponding error for the second variable y2

X	Y2	E2	Exact3
-1.0000	0.3679	-9.9998e-13	0.36788
-0.9500	0.3867	-9.9998e-13	0.38674
-0.9000	0.4066	-9.9998e-13	0.40657
-0.8500	0.4274	-9.9998e-13	0.42741
-0.8000	0.4493	-9.9998e-13	0.44933
-0.7500	0.4724	-9.9998e-13	0.47237
-0.7000	0.4966	-9.9998e-13	0.49659
-0.6500	0.5220	-9.9998e-13	0.52205
-0.6000	0.5488	-9.9998e-13	0.54881
-0.5500	0.5769	-9.9998e-13	0.57695
-0.5000	0.6065	-9.9998e-13	0.60653
-0.4500	0.6376	-9.9998e-13	0.63763
-0.4000	0.6703	-9.9998e-13	0.67032
-0.3500	0.7047	-9.9998e-13	0.70469
-0.3000	0.7408	-9.9998e-13	0.74082
-0.2500	0.7788	-9.9998e-13	0.7788
-0.2000	0.8187	-9.9998e-13	0.81873
-0.1500	0.8607	-9.9998e-13	0.86071
-0.1000	0.9048	-9.9998e-13	0.90484
-0.0500	0.9512	-9.9998e-13	0.95123
-0.0000	1.0000	-9.9998e-13	1
0.0500	1.0513	-1.0001e-12	1.0513
0.1000	1.1052	-1.0001e-12	1.1052
0.1500	1.1618	-1.0001e-12	1.1618
0.2000	1.2214	-1.0001e-12	1.2214
0.2500	1.2840	-1.0001e-12	1.284
0.3000	1.3499	-1.0001e-12	1.3499
0.3500	1.4191	-1.0001e-12	1.4191
0.4000	1.4918	-1.0001e-12	1.4918
0.4500	1.5683	-1.0001e-12	1.5683
0.5000	1.6487	-1.0001e-12	1.6487
0.5500	1.7333	-1.0001e-12	1.7333
0.6000	1.8221	-1.0001e-12	1.8221
0.6500	1.9155	-1.0001e-12	1.9155
0.7000	2.0138	-1.0001e-12	2.0138
0.7500	2.1170	-1.0001e-12	2.117
0.8000	2.2255	-1.0001e-12	2.2255
0.8500	2.3396	-1.0001e-12	2.3396
0.9000	2.4596	-1.0001e-12	2.4596
0.9500	2.5857	-1.0001e-12	2.5857
1.0000	2.7183	-1.0001e-12	2.7183

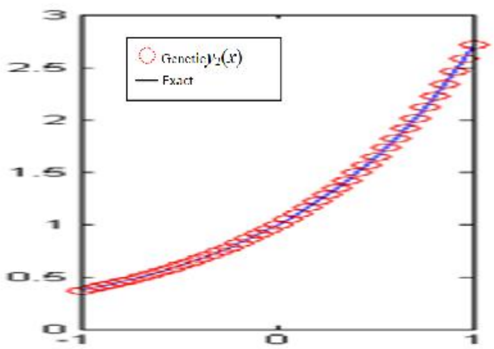


Figure 2. Values of y2 , its Genetic approximation(y'2)

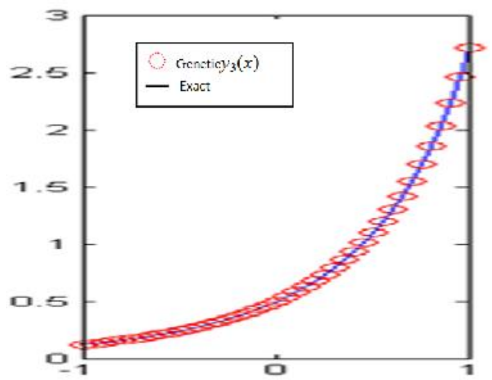


Figure 3. Values of y3 , its Genetic approximation(y'3)

Table 3. The corresponding error for the second variable y3

X	Y3	E3	Exact3
-1.0000	0.12263	-1e-12	0.12263
-0.9500	0.1311	-1e-12	0.1311
-0.9000	0.1402	-9.9998e-13	0.1402
-0.8500	0.14997	-9.9998e-13	0.14997
-0.8000	0.16047	-1e-12	0.16047
-0.7500	0.17177	-1e-12	0.17177
-0.7000	0.18392	-1e-12	0.18392
-0.6500	0.197	-9.9998e-13	0.197
-0.6000	0.21108	-9.9998e-13	0.21108
-0.5500	0.22625	-1e-12	0.22625
-0.5000	0.24261	-1e-12	0.24261
-0.4500	0.26026	-9.9998e-13	0.26026
-0.4000	0.2793	-9.9998e-13	0.2793
-0.3500	0.29987	-9.9998e-13	0.29987
-0.3000	0.32209	-9.9998e-13	0.32209
-0.2500	0.34613	-9.9998e-13	0.34613
-0.2000	0.37215	-9.9998e-13	0.37215
-0.1500	0.40033	-9.9998e-13	0.40033
-0.1000	0.43087	-9.9998e-13	0.43087
-0.0500	0.46401	-9.9998e-13	0.46401
-0.0000	0.5	-9.9998e-13	0.5
0.0500	0.53911	-9.9998e-13	0.53911
0.1000	0.58167	-9.9998e-13	0.58167
0.1500	0.62802	-9.9998e-13	0.62802
0.2000	0.67856	-9.9998e-13	0.67856
0.2500	0.73373	-9.9998e-13	0.73373
0.3000	0.79403	-9.9998e-13	0.79403
0.3500	0.86004	-1.0001e-12	0.86004
0.4000	0.93239	-9.9998e-13	0.93239
0.4500	1.0118	-1.0001e-12	1.0118
0.5000	1.0991	-1.0001e-12	1.0991
0.5500	1.1953	-1.0001e-12	1.1953
0.6000	1.3015	-1.0001e-12	1.3015
0.6500	1.4189	-1.0001e-12	1.4189
0.7000	1.549	-1.0001e-12	1.549
0.7500	1.6936	-1.0001e-12	1.6936
0.8000	1.8546	-1.0001e-12	1.8546
0.8500	2.0345	-1.0001e-12	2.0345
0.9000	2.236	-1.0001e-12	2.236
0.9500	2.4626	-1.0001e-12	2.4626
1.0000	2.7183	-1.0001e-12	2.7183

Conclusion

In this Section, we represent the obtained Genetic programming results for the ODEs problems, and comparison between Genetic programming method and exact solution and calculate the error between it. Fig. 2, Fig. 3, Fig. 4 and Table 3 show that the values of error better in the case of functions which have small values. Obviously using genetic much better of using DAE method. In the future we will try to improve this way when the values are big. Although the error is relatively large in some regions, the advantage of this method for numerical methods is finding a function which give approximate solution for ODEs, and can be handled with this function and study its behavior so this is better in some boundary value problems from numerical methods.

REFERENCES

Ascher, U. M. 1989. "On symmetric schemes and differential-algebraicequations" *SIAM J. Sci. Stat. Comput.*, 10, pp. 937-949.

Ascher, U. M., L.R. Petzold, 1991. "Projected implicit Runge Kutta methods for differential-algebraic equations"*SIAM J. Numer. Anal.*, 28, pp.1097-1120.

Ascher, U. M., P. Lin, 1996. "Sequential regularization methods for higher index differential-algebraic equations with constant singularities. the linear index-2 case"*SIAM J. Numer. Anal.*, 33, pp. 1921-1940.

Ascher, U. M., P. Lin, 1997, "Sequential regularization methods for non-linear higher index differential-algebraic equations"*SIAM J. Sci. Comput.*, 18,pp.160-181.

Biazar, J., E. Babolian, A. Nouri, R. Islam, 2003. "An alternate algorithm for computing Adomian decomposition method in special cases" Applied Mathematics and Computations, 138, pp. 1-7.

Biazar, J., M. Tango, E. Babolian, R. Islam, 2003. "Solution of the kinetic moeling of lactic axi fermentation using Adomian decomposition method" Applied Mathematics and Computations, 139, pp. 249-258.

Campbell, S. L., E. Moore, Y. Zhong, 1994, "Utilization of automatic differentiation in control algorithms"*IEEE Trans. Automat. Control.*, 39, pp.1047-1052.

Campbell, S.L. 1989, A Computational method for general higher index singular systems of differential equations, *IMACS Trans. Sci. Comput.*, 89, pp.555-560.

Çelik, E., E. Karaduman, and M. Bayram, 2002, "Numerical Method to Solve Chemical Differential-Algebraic Equations"*International Journal of Quantum Chemistry*, 89, pp. 447-451.

Çelik, E., M. Bayram, 2003. "On the numerical solution of differential-algebraic equations by Padé series" Applied Mathematics and Computations, 137, pp.151-160.

Gear, C. W., L. R. Petzold, 1984. "ODE systems for the solution of differentialalgebraic systems"*SIAM J. Numer. Anal.*, 21, pp. 716-728.

Petzold, L. R. 1995. "Numerical Solution of Differential-Algebraic Equations" Advances in Numerical Analysis IV.

Wazwaz, A. M. 1995. "A new approach to the nonlinear advection problem: An application of the decomposition technique" *Appl. Math. Comput*, 72, pp. 175-181.

****