Solving Ordinary Differential Equation (ODE) Using Least Square Method: Application of Steepest Descent Method

Siti Farhana Husin, Mustafa Mamat, Mohd Asrul Hery Ibrahim, Mohd Rivaie

Abstract: ***An ordinary differential equation (ODE) is an equation and techniques that is widely used in mathematical modelling and the most mathematical formulations used in physical laws. One of the useful numerical method to solve non-homogeneous second order linear ODE is the least square method (LSM). However, the LSM requires to the use of the inverse matrix to find the solution. Hence to prevent this difficulties, this paper seeks to solve ODE by using LSM with an application of optimization method using steepest descent (SD) method.***

Index Terms: ***ordinary differential equation, least square method, steepest descent method.***

I. INTRODUCTION

An ordinary differential equation (ODE) is an equation that is super useful for modeling and simulating phenomena which involves only ordinary derivatives with respect to a single variable of some unknown function. A simple example that leads to an ODE is Newton's second law of motion applied to a free falling body which defined as

$$m\frac{d^2h}{dt^2} = m\frac{dv}{dt} = -mg$$

where m is the mass of the object, g is the gravitational acceleration, h is the height of the object and $v = \frac{dh}{dt}$ is the velocity.

The ODE can be divided into two main classes which is linear and non-linear equations and among these, linear differential equation is by far play an outstanding role for several reasons. In this study, we focus on the second linear ODE which the general solution form can be written as

$$b_2(x)y'' + b_1(x)y' + b_o(x)y = h(x)$$

where b_0, b_1 and b_o are constants and $h(x) \neq 0$ with unknown dependent variable y and independent variable x

The main aim of the paper is to solve some linear ODE using LSM with an application of steepest descent (SD) method. The issues addressed in the next section begins by the introduction of the steepest descent method applied followed by the methods and algorithms used as a numerical method in this research. The remaining part of this paper are the results and findings of the research and finally, the conclusion gives a brief summary and a little recommendations given.

II. IMPLEMENTATION

Most ODE problems that arise in real-world applications is difficult to solve using theoretical solution because of the complexity of the functions that involve the combination of polynomials, exponents and trigonometry that in need of in-depth understanding and complicated calculations. Therefore, to overcome this difficulties, researchers have come out with an alternative method to solve the ODE by using numerical method. One of the numerical solution that can be used to solve a non-homogeneous second order linear ODE is least square method (LSM).

The LSM is the form of mathematical regression analysis that finds the line of best fit for a data set, providing a visual demonstration of the relationship between the data points. However, in order to solve the second order linear ODE using LSM, there will be a problem when one's dealing with the solution that involves inverse matrix. To overcome this problem, optimization method is adopted specifically for this study we use steepest descent method.

The algorithms below show the calculation steps in getting theoretical solutions using undetermined coefficient method (UCM) and variation of parameter method (VPM) and also algorithm to get numerical solution using LSM.

Algorithm 1: UCM

Step 1: Compute y_c for obtaining

$$y_c = c_1y_1 + c_2y_2 + \ldots + c_ny_n .$$

Step 2: Determine the form particular solution y_p.

Step 3: Determine the unknown coefficient by substituting y_p into the non-homogeneous equation and equating the coefficient.

Step 4: Compute the general solution $y = y_c + y_p$.

Step 5: Find the arbitrary constants remaining in the general solution.

Revised Manuscript Received on February 11, 2019.

Siti Farhana Husin, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.

Mustafa Mamat, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia.

Mohd Asrul Hery Ibrahim, Faculty of Entrepreneurship and Business, Universiti Malaysia Kelantan, Kelantan, Malaysia.

Mohd Rivaie, Department of Computer Sciences and Mathematics, Universiti Teknologi MARA, Terengganu, Malaysia.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Algorithm 2: VPM

Step 1: Compute y_c for obtaining

$$y_c = c_1 y_1 + c_2 y_2 + \ldots + c_n y_n .$$

Step 2: Compute the Wronskian

$$W\left[y_1(x), y_2(x)\right] = \begin{vmatrix} y_1 & y_2 \\ y_1' & y_2' \end{vmatrix}$$

$$W_1 = \begin{vmatrix} 0 & y_2 \\ F(x) & y_2' \end{vmatrix} = -y_2 F(x)$$

$$W_2 = \begin{vmatrix} y_1 & 0 \\ y_1' & F(x) \end{vmatrix} = y_1 F(x)$$

Step 3: Compute for u_1' and u_2'.

Step 4: Find u_1 and u_2 using integration.

Step 5: Solve for $y_p = u_1 y_1 + u_2 y_2$.

Step 6: Compute general solution $y = y_c + y_p$.

Step 7: Find the arbitrary constants remaining in the general solution.

Algorithm 3: LSM

Step 1: Assume the solution is in the cubic form and find the derivative of y' and y''

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

Step 2: Substituting y, y' and y'' into the general ODE

Step 3: Finding the error and compute $\left(E(x)\right)^2$

$$E(x) = \left[\sum_{i=0}^{n} a_i \left[i(i-1)x^{i-2} + ix^{i-1}P(x) + x^i Q(x)\right] - R(x)\right]$$

Step 4: Find partial derivatives with respect to a_1 and a_2, then compute the definite integral from a to b

$$\frac{\partial F}{\partial a_i} = \int_b^a E(x)\,dx = 0$$

Step 5: Solve the system of linear equation $Ax = b$, $\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ using direct inverse.

However, step 5 in the LSM algorithm will lead to solve an inverse matrices and there will be a problem if the matrices involving singular or nearly singular matrices where the inverse does not exist. Hence, to overcome this problem, the SD method is applied to solve the system of linear equation.

III. METHODOLOGY

Steepest descent (SD) method also called as gradient method is the simplest and very well-known method among several algorithms for finding the nearest local minimum of a function which assumes that the gradient of the function can be estimated. Although this method has severe drawback in requiring many iterations to converge towards local minimum, in latter case, many evolutions and modifications have been done for this method [1]–[10].

The general minimization problem of a function is given by

$$\min_{x \in \square} f(x)$$

which has the following iterative form

$$x_{k+1} = x_k + \alpha_k d_k$$

where α_k is the step size which can be calculated using some methods and d_k is the search direction. The standard method to determine the search direction is the SD method which defined as

$$d_k = -g_k \qquad (1)$$

The research to date has tended to focus on modification of the search direction for SD method [4], [6], [9], [10]. In this paper, we will apply and at the same time we want to test the practicality of the direction defined by [9] which known as FMAR1 and denote as

$$d_k^{FMAR1} = -g_k - \theta_k g_{k-1} \qquad (2)$$

where $\theta_k = \dfrac{\|g_k\|^2}{\|g_{k-1}\|^2}$ which will be implement in the LSM to solve the linear ODE. Here and throughout, we use g_k to denote the gradient of f at x_k. We will also use f_k as the abbreviation of $f(x_k)$. The superscript T signifies the transpose.

Recently researchers have examined the application of the optimization method for solving real-data problems that have been transformed into minimization of a function [11]–[16]. The steepest descent algorithm was also applied in most engineering problems including artificial neural networks, neuro fuzzy and an optimized architecture [17]–[19]. Therefore, in this paper, we want to implement the modification of direction in SD method into LSM in order to solve some selected functions from linear ODE.

This paper is focused on two different type of non-homogenous second order linear ODE with boundary condition from 0 to 1 collected from books relate to ODE, [20], [21]. The sample of function and boundary value problem used are as follows:

Problem 1

$y'' + 4y = x^2 + 3e^x$ with BVP: $y(0) = 0$ and $y(1) = 2$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Problem 2

$y'' - 2y' + y = xe^x + 4$ with BVP: $y(0) = 1$ and $y(1) = 2$

For the purpose of comparison, these problems was solved theoretically using undetermined coefficient method (UCM) and variation of parameters method (VPM) respectively to get the exact value. These values was then compared to the solution from the numerical experiments obtained by LSM and SD method, which represent as approximate values and the percentage error will be determine using formula given as

$$\text{Percentage error} = \frac{|\text{Exact value} - \text{Approximate value}|}{|\text{Exact value}|} \times 100\%$$

The algorithm of SD method is given by:

Algorithm 4: SD Method

Step 1: Initialization. Some initial value is chosen and set $k = 0$.

Step 2: Compute the search direction, d_k, by using (1) and (2).

Step 3: Compute the step size, λ_k, by using exact line search procedure.

Step 4: Update new point of iteration, $a_{k+1} = a_k + \lambda_k d_k$.

Step 5: Test the convergent and stopping criteria:

If $a_{k+1} < a_k$ and $|g_k| \le \varepsilon$, then stop. Otherwise, go to Step 1 with $k = k + 1$.

The program was written in the MATLAB 2017a and run on the computer with Intel® Core™ i5 with CPU 2.5GHz and 6.4-bit Operating System.

IV. RESULTS AND DISSCUSSION

This section begins by focusing on two different non-homogeneous linear ODE. To get the exact solution, Problem 1 is solved using UCM while problem 2 is solved using VPM while the approximate value is generated from the numerical solution by using LSM with direct inverse, standard SD and FMAR1 methods. The exact and approximate solutions of both problems is tabulated in Table 1.

Table 1: Exact and approximate solutions for Problem 1 and 2

Problem 1	Function	$y'' + 4y = x^2 + 3e^x$
	Theoretical solution (UCM)	$y = 0.475\cos 2x + 0.05098659\sin 2x + 0.25x^2 - 0.125 + 0.6e^x$
	Numerical solution using LSM with direct inverse	$y = 0.7235047x + 1.61976529x^2 - 0.34326999x^3$
	Numerical solution using LSM with standard SD	$y = 0.72321533x + 1.62021850x^2 - 0.34343383x^3$
	Numerical solution using LSM with FMAR1	$y = 0.7207937327x + 1.624013266x^2 - 0.3448070x^3$
Problem 2	Function	$y'' - 2y' + y = xe^x + 4$
	Theoretical solution (VPM)	$y = -3e^x + 2.09757445xe^x + 0.16666667x^3e^x + 4$
	Numerical solution using LSM with direct inverse	$y = 1 - 0.87447280x - 0.25506554x^2 + 2.21953834x^3$
	Numerical solution using LSM with standard SD	$y = 1 - 0.87447287x - 0.25506558x^2 + 2.21953845x^3$
	Numerical solution using LSM with FMAR1	$y = 1 - 0.8744725436x - 0.2550661411x^2 + 2.129538685x^3$

Table 2 shows the tabulated solution values and Table 3 is the error calculation for Problem 1. To further analysis of this result, Fig. 1 and 2 were plotted to display the solution of the functions given by Problem 1 and Problem 2 respectively. Table 4 shows the tabulated value for Problem 2 and Table 5 is the error calculation for Problem 2.

From the table of the error calculations for both problems, it can be seen that the error is below 7% which the largest error occur at a point $x = 0.5$ in problem 2 by the used of numerical solution from the standard SD method. The sum of relative error from FMAR1 showed the lowest at 14.22% and 30.04% for both problems respectively compared to the other methods used in this research. Based on the graphs above, it shows that all line intercept at both end whereas the lines did not overlap each other at the middle section of the graph. It can be conclude that the LSM is suitable to be used to solve those problems chosen and the implementation of FMAR1 in the LSM give the best result since it has the smallest sum of relative error.

Table 2: Solution values for Problem 1

x	y			
	Exact	Direct Inverse	Steepest Descent	FMAR1
0.1	0.08520022	0.08820485	0.08818028	0.087974699
0.2	0.20019244	0.20674539	0.20670433	0.206360821
0.3	0.34416955	0.353562	0.35351155	0.353089525
0.4	0.51573403	0.52659505	0.52654133	0.526091968
0.5	0.71299215	0.72378492	0.72373306	0.723299308

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

0.6	0.933672	0.94307201	0.94302615	0.942642703
0.7	1.17526107	1.18239668	1.18235999	1.182053312
0.8	1.43515828	1.43969931	1.43967398	1.439462292
0.9	1.71292029	1.712922029	1.71290752	1.712800801
1	2	2	2	2

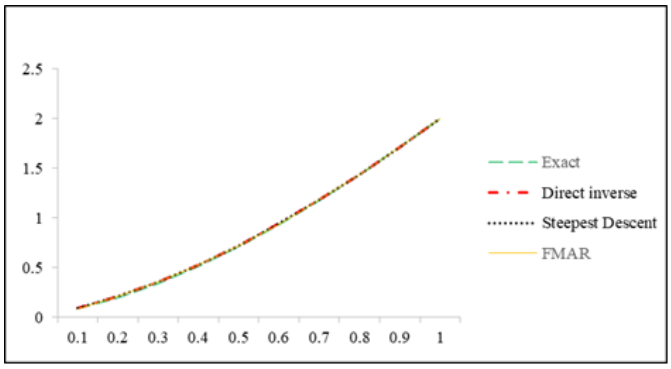

Fig. 1: Solution graph for Problem 1

Table 3: Values of error calculation for Problem 1

x	Errors		
	Direct Inverse	Steepest Descent	FMAR1
0.1	3.526551927	3.497713973	3.25642226
0.2	3.273325406	3.252815141	3.081225795
0.3	2.729018299	2.714359826	2.591738491
0.4	2.105934332	2.09551811	2.008387482
0.5	1.513729148	1.506455576	1.44562011
0.6	1.006778612	1.001866823	0.960798118
0.7	0.607151056	0.604029197	0.577934739
0.8	0.316413183	0.314648221	0.299898071
0.9	0.000101523	0.00074551	0.006975748
1	0	0	0
Sum of relative errors	15.07900349	14.98815238	14.22900081

Table 4: Tabulated values for Problem 2

x	y			
	Exact	Direct Inverse	Steepest Descent	FMAR1
0.1	0.91648927	0.9121316	0.9121316	0.912131623
0.2	0.84981691	0.83193912	0.831939111	0.831939155
0.3	0.80592674	0.7721998	0.772199776	0.772199829
0.4	0.79212405	0.74569085	0.745690821	0.745690876
0.5	0.81734235	0.76518951	0.765189478	0.765189529
0.6	0.89245778	0.84347301	0.843472976	0.843473019
0.7	1.03065877	0.99331858	0.993318547	0.993318579
0.8	1.24788027	1.22750344	1.22750342	1.227503442
0.9	1.563313	1.55880	1.55880482	1.55880483
1	2	2	2	2

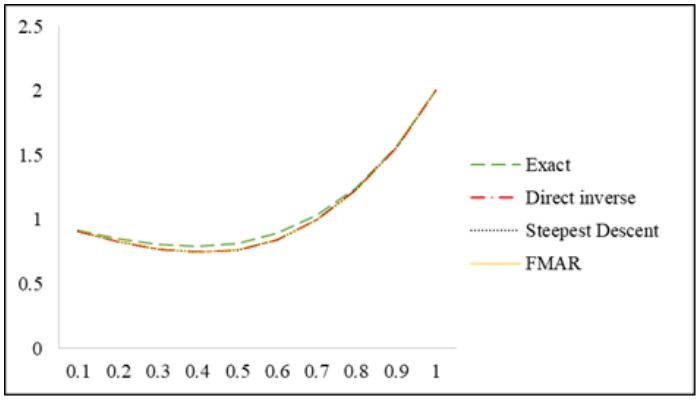

Fig. 2: Solution graph for Problem 2

Table 5: Error calculation for Problem 2

x	Errors		
	Direct Inverse	Steepest Descent	FMAR1
0.1	0.475474197	0.475474197	0.475471699
0.2	2.103722554	2.103723589	2.103718412
0.3	4.184864247	4.184867213	4.184860686
0.4	5.861859642	5.861863252	5.861856385
0.5	6.380782789	6.380786705	6.380780526
0.6	5.488749283	5.488753059	5.488748274
0.7	3.622943993	3.622947205	3.622944052
0.8	1.632915472	1.632917075	1.632915312
0.9	0.28837347	0.288374173	0.288373598
1	0	0	0
Sum of relative errors	30.03968565	30.03970647	30.03966894

From the table of the error calculations for both problems, it can be seen that the error is below 7% which the largest error occur at a point $x = 0.5$ in problem 2 by the used of numerical solution from the standard SD method. The sum of relative error from FMAR1 showed the lowest at 14.22% and 30.04% for both problems respectively compared to the other methods used in this research. Based on the graphs above, it shows that all line intercept at both end whereas the lines did not overlap each other at the middle section of the graph. It can be conclude that the LSM is suitable to be used to solve those problems chosen and the implementation of FMAR1 in the LSM give the best result since it has the smallest sum of relative error.

V. CONCLUSION

Collectively, the main goal of this research is to solve the non-homogeneous second order linear ODE using LSM with an implementation of optimization method specifically modification of SD method introduced by [9]. It is recommended that further research be undertaken especially in solving different type of problem such as higher order ODE or a non-linear equation.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

ACKNOWLEDGMENT

The authors thank the referees for their contributions and also grateful to the Malaysian government and Universiti Sultan Zainal Abidin for funding this research under the Fundamental Research Grant Scheme (FRGS/1/2017/STG06/Unisza/01/1).

REFERENCES

1. G. Zhou and C. Feng, "The steepest descent algorithm without line search for p-Laplacian," Appl. Math. Comput., 224, 2013, pp. 36-45.
2. M. Mamat, A. S. Yee, and I. Mohd, "An efficient algorithm for steepest descent method for unconstrained optimization," J. Sci. Technol. UTHM, 2012, pp. 13-25.
3. Y. Yuan, "A new stepsize for the steepest descent method," Journal of Computational Mathematics, 24(2), 2006, pp. 149–156.
4. Z. Zainal Abidin, M. Mamat, and M. Rivaie, "A new steepest descent method with global convergence properties," AIP Conf. Proc., 1739(1), 2016, pp. 1-8.
5. M. Raydan and B. F. Svaiter, "Relaxed steepest descent and Cauchy-Barzilai-Borwein method," Comput. Optim. Appl., 21(2), 2002, pp. 155-167.
6. R. Johari, M. Rivaie, M. Mamat, and Z. Salleh, "A new scaled steepest descent method for unconstrained optimization with global convergence properties," J. Eng. Appl. Sci, 2018, pp. 5442-5445.
7. C.-S. Liu, J.-R. Chang, and Y.-W. Chen, "A modified algorithm of steepest descent method for solving unconstrained nonlinear optimization problems," J. Mar. Sci. Technol., 23(1), 2015, pp. 88–97.
8. Z. A. Zubai'ah, N. 'Aini, S. F. Husin, M. Rivaie, and M. Mamat, "Hybrid BFGS-ZMRI methods with global convergence properties," AIP Conf. Proc., 1974(1), 2018, pp. 1-8.
9. S. F. Husin, M. Mamat, M. A. H. Ibrahim, and M. Rivaie, "A modification of steepest descent method for solving large-scaled unconstrained optimization problems," Int. J. Eng. Technol., 7(3.28), 2018, pp. 1–13.
10. S. F. Husin, M. Mamat, M. A. H. Ibrahim, and M. Rivaie, "New search direction of steepest descent method for large-scaled unconstrained optimization problem," J. Phys. Conf. Ser., 1132(1), 2018, pp. 1-8.
11. M. Bartholomew-Biggs, Nonlinear Optimization with Financial Applications. Boston/ Dordrecht/ London: Kluwer Academic Publishers, 2005.
12. X. Wang, "Method of steepest descent and its applications," IEEE Microw. Wirel. Components Lett., 37996, 2008, pp. 1–3.
13. C. L. Philip Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Inf. Sci. (Ny)., 275, 2014, pp. 314–347.
14. W. Khadijah, M. Rivaie, M. Mamat, N. Hajar, N. Aini, and Z. Abidin, "An application of conjugate gradient method under strong Wolfe line search for solving unconstrained optimization," International Journal of Engineering and Technology, 7(3.28), 2018, pp. 12–16.
15. N. Shapiee, M. Rivaie, M. Mamat, and P. L. Ghazali, "A new family of conjugate gradient coefficient with application," International Journal of Engineering and Technology, 7(3.28), 2018, pp. 36–43.
16. I. M. Sulaiman, M. Mamat, M. Y. Waziri, and N. S. Amir, "Barzilai-Borwein gradient method for solving fuzzy nonlinear equations," International Journal of Engineering and Technology, 7(3.28), 2018, pp. 80–83.
17. B. P. J and J. V. R. S. P, "Anoptimized architecture for adaptive digital filter," ARPN J. Eng. Appl. Sci., 10(11), 2015, pp. 4964–4970.
18. D. V. M. Chary and J. Amarnath, "Computation of available transfer capability incorporating effect of reactive power and losses using complex neural network," ARPN J. Eng. Appl. Sci., 4(8), 2009, pp. 45–50.
19. N. Austin, P. Senthilkumar, and S. Purushothaman, "Mixed refrigerants suitability analysis using artificial neural networks," ARPN J. Eng. Appl. Sci., 7(5), 2012, pp. 588–592.
20. G. Z. Dennis, Differential Equations with Boundary-Value Problems. Ontario: Nelson Education, 2016.
21. E. B. William and C. D. Richard, Elementary Differential Equations and Boundary Value Problems. New York: Wiley, 2012.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication