

Solution of Second-Order Ordinary Differential Equations via Simulated Annealing

Abdulazeez Bilesanmi¹, Ashiribo Senapon Wusu², Akinwale Lewis Olutimo²

¹Department of General Studies, Petroleum Training Institute, Delta, Nigeria

²Department of Mathematics, Lagos State University, Lagos, Nigeria

Email: bilesanmi_ao@pti.edu.ng, wussy_ash@yahoo.com, ashiribo.wusu@lasu.edu.ng, aolutimo@yahoo.com

How to cite this paper: Bilesanmi, A., Wusu, A.S. and Olutimo, A.L. (2019) Solution of Second-Order Ordinary Differential Equations via Simulated Annealing. *Open Journal of Optimization*, 8, 32-37.

<https://doi.org/10.4236/ojop.2019.81003>

Received: December 1, 2018

Accepted: February 24, 2019

Published: February 27, 2019

Copyright © 2019 by author(s) and

Scientific Research Publishing Inc.

This work is licensed under the Creative

Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

In this paper, we approach the problem of obtaining approximate solution of second-order initial value problems by converting it to an optimization problem. It is assumed that the solution can be approximated by a polynomial. The coefficients of the polynomial are then optimized using simulated annealing technique. Numerical examples with good results show the accuracy of the proposed approach compared with some existing methods.

Keywords

Simulated Annealing, Second-Order Ordinary Differential Equation, Polynomial, Optimization

1. Introduction

The use of techniques that are based on evolutionary algorithms for solving optimization problems has been gaining interests over the last few years. These algorithms use mechanisms inspired by biological evolution, such as reproduction, recombination, mutation, and selection. Since the work of Isaac Newton and Gottfried Leibniz in the late 17th century, differential equations (DEs) have been an important concept in many branches of science. Differential equations arise in physics, engineering, chemistry, biology, economics and a lot of fields. The idea of solving DEs via evolutionary algorithms has been on the increase recently. Approximate solutions of differential equations are obtained by converting the equations to optimization problems and then solved via optimization techniques. The use of classical genetic algorithm to obtain approximate solutions of second-order initial value problems was considered in [1]. In [2], the author combined genetic algorithm with the Nelder-Mead method for solving the second-order initial value problem of the form $y'' = f(x, y)$. In a later work,

approximate solutions of first-order initial value problem was computed via the combination of collocation method and genetic algorithms by the author in [3]. Adaptation of neural network for the solution of second-order initial value problems was proposed by the authors in [4]. Continuous genetic algorithm was used to compute the solution of two-point second-order ordinary differential equation in [5]. The adaptation of differential evolution algorithm for the solution of the second-order initial value problem of the form $y'' + p(t)y' + q(t)y = r(t)$ was proposed in [6]. The authors in [7] considered the approach of using differential algorithm to obtain approximate solutions of the second-order two-point boundary value problem $u'' = f(t, u); u(a) = \eta_1; u(b) = \eta_2$ with oscillatory/periodic behaviour. In this paper we show that the simulated annealing algorithm can also be used to find very accurate approximate solutions of second-order initial value problems of the form

$$y'' = f(t, y); \quad y(t_0) = y_0, \quad y'(t_0) = y'_0, \quad t \in [a, b]. \quad (1)$$

2. Basic Notions of Simulated Annealing Algorithm

Simulated annealing is a simple stochastic function minimizer. It is motivated from the physical process of annealing, where a metal object is heated to a high temperature and allowed to cool slowly. The process allows the atomic structure of the metal to settle to a lower energy state, thus becoming a tougher metal. Using optimization terminology, annealing allows the structure to escape from a local minimum, and to explore and settle on a better, hopefully global, minimum.

At each iteration, a new point, x_{new} , is generated in the neighborhood of the current point, x . The radius of the neighborhood decreases with each iteration. The best point found so far, x_{best} , is also tracked.

If $f(x_{new}) \leq f(x_{best})$, x_{new} replaces x_{best} and x . Otherwise, x_{new} replaces x with a probability $\exp(b(i, \Delta f, f_0))$. Here b is the function defined by *Boltzmann Exponent-exponent of the probability function*, i is the current iteration, Δf is the change in the objective function value, and f_0 is the value of the objective function from the previous iteration. The default definition of the function for b is given as $b(i, \Delta f, f_0) := \frac{-\Delta f \log(i+1)}{10}$.

Simulated annealing uses multiple starting points, and finds an optimum starting from each of them. The default number of starting points, given by the parameter *SearchPoints*, is $\min(2d, 50)$, where d is the number of variables and in this case $d = 1$, since the number of independent variable is one.

3. Proposed Method

Consider the second-order initial value problem (1), assume a solution of the form

$$y(t) = \sum_{i=0}^k \psi_i t^i, \quad k \in \mathbb{Z}^+ \quad (2)$$

where ψ_i are parameters to be determined. Substituting (2) and its second derivative into (1) gives

$$\sum_{i=2}^k i(i-1)\psi_i t^{i-2} = f(t, y(t)) \quad (3)$$

Using the initial conditions we have the constraint that

$$\left[\sum_{i=0}^k \psi_i t^i \right]_{t=t_0} = y_0 \quad \text{and} \quad \left[\sum_{i=1}^k i\psi_i t^{i-1} \right]_{t=t_0} = y'_0 \quad (4)$$

At each node point t_n , we require that

$$\mathcal{E}_n(t) = \left[\sum_{i=2}^k i(i-1)\psi_i t^{i-2} - f(t, y(t)) \right]_{t=t_n} \simeq 0 \quad (5)$$

To solve the above problem, we need to find the set $\{\psi_i \mid i = 0(1)k\}$, which minimizes the expression

$$\sum_{n=1}^{\frac{b-a}{h}} \mathcal{E}_n^2(t) \quad (6)$$

where h is the steplength. We now formulate the problem as an optimization problem in the following way:

$$\text{Minimize: } \sum_{n=1}^{\frac{b-a}{h}} \mathcal{E}_n^2(t) \quad (7)$$

$$\text{Subject to: } \left[\sum_{i=0}^k \psi_i t^i \right]_{t=t_0} = y_0 \quad \text{and} \quad \left[\sum_{i=1}^k i\psi_i t^{i-1} \right]_{t=t_0} = y'_0 \quad (8)$$

Using the simulated annealing algorithm we are able to obtain the set $\{\psi_i \mid i = 0, 1, \dots, k\}$ which minimizes the expression $\sum_{n=1}^{\frac{b-a}{h}} \mathcal{E}_n^2(t)$.

4. Numerical Experiments

We now perform some numerical experiments confirming the theoretical expectations regarding the method we have proposed. Our proposed algorithm shall be compared with the Runge-Kutta Nystrom method in this section. The following parameters needed to implement the simulated annealing are set as follows:

- exponent of the probability function (Boltzmann Exponent = 1).
- set of initial points (Initial Points = 1000).
- maximum number of iterations to stay at a given point (Level Iterations = 50).
- scale for the random jump (Perturbation Scale = 1.0).
- starting value for the random number generator (Random Seed = 0).
- number of initial points (Search Points = 0).
- tolerance for accepting constraint violations (Tolerance = 0.000001).

4.1. Example 1

We examine the following linear equation

$$y''(t) - y(t) = t - 1; \quad y(0) = 2, \quad y'(0) = -2 \quad (9)$$

with the exact solution $y(t) = 1 - t + \exp(-t)$.

Implementing the proposed scheme with $k = 10$, we obtain $\{\psi_i \mid i = 0(1)10\}$ as

$$\left\{ 2, -2, \frac{416995243}{834315644}, -\frac{105164777}{636059757}, \frac{69800031}{1811256752}, -\frac{6535788}{1275358859} \right\}.$$

Using a steplength of $h = 0.01$, the absolute errors obtained by our proposed algorithm are compared with those produced by the well-known Runge-Kutta Nystrom method as shown in **Table 1**. The comparison shows that our approach gave better result compared with the Runge-Kutta Nystrom method.

4.2. Example 2

Consider the equation

$$y''(t) = (1 + t^2)y(t); \quad y(0) = 1, \quad y'(0) = 0 \quad (10)$$

with the exact solution $y(t) = \exp\left(\frac{t^2}{2}\right)$.

Implementing the proposed scheme with $k = 11$, we obtain $\{\psi_i \mid i = 0(1)11\}$ as

$$\left\{ 1, 0, \frac{1306409430}{2612828131}, \frac{29397245}{1713857114397}, \frac{3187969586}{25524507753}, \frac{172091099}{436085951591}, \right. \\ \left. \frac{313833621}{15857243966}, \frac{382909153}{201794651238}, \frac{117010789}{496614906383}, \right. \\ \left. \frac{766119929}{389265107664}, -\frac{130287162}{172534329575}, \frac{125796527}{456658410146} \right\}$$

Table 1. The absolute values of error of $y(t)$ in Problem 9 using the proposed scheme compare with the Runge-Kutta Nystrom method.

t	Runge-Kutta Nystrom	Proposed Scheme
0.00	0	0.
1.00E-1	2.973739E-10	2.591705E-12
2.00E-1	7.050944E-10	5.964562E-12
3.00E-1	1.217025E-9	9.366508E-12
4.00E-1	1.829043E-9	1.286815E-11
5.00E-1	2.538910E-9	1.649259E-11
6.00E-1	3.346159E-9	2.029099E-11
7.00E-1	4.252021E-9	2.428346E-11
8.00E-1	5.259365E-9	2.852540E-11
9.00E-1	6.372666E-9	3.304634E-11
1.00	7.597991E-9	3.792694E-11

Table 2. The absolute values of error of $y(t)$ in Problem 10 using the proposed scheme compare with the Runge-Kutta Nystrom method.

t	Runge Kutta Nystrom	Proposed Scheme
0.00	0	0.
2.00E-1	1.266432E-7	1.76241E-8
4.00E-1	2.923595E-7	3.804534E-8
6.00E-1	5.418410E-7	6.027792E-8
8.00E-1	9.469284E-7	8.585760E-8
1.00	1.627915E-6	1.171072E-7

Table 2 shows the absolute errors of the results obtained by our algorithm compared with the Runge-Kutta Nystrom method. Again, our approach gave minimal absolute errors.

5. Conclusion

In this paper, we have shown how the problem of obtaining approximate solution to (1) can be converted to an optimization problem, and then solved using simulated annealing. The results obtained compete favourably with the Runge-Kutta Nystrom method.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] George, D.M. (2006) On the Application of Genetic Algorithms to Differential Equations. *Romanian Journal of Economic Forecasting*, **3**, 5-9.
- [2] Mastorakis, N.E. (2005) Numerical Solution of Non-Linear Ordinary Differential Equations via Collocation Method (Finite Elements) and Genetic Algorithms. *Proceedings of the 6th WSEAS International Conference on Evolutionary Computing*, Lisbon, 16-18 June 2005, 36-42.
- [3] Mastorakis, N.E. (2006) Unstable Ordinary Differential Equations: Solution via Genetic Algorithms and the Method of Nelder-Mead. *Proceedings of the 6th WSEAS International Conference on Systems Theory & Scientific Computation*, Elounda, 21-23 August 2006, 1-6.
- [4] Junaid, A., Raja, A.Z. and Qureshi, I.M. (2009) Evolutionary Computing Approach for the Solution of Initial Value Problems in Ordinary Differential Equations. *World Academic of Science, Engineering and Technology*, **55**, 578-581.
- [5] Omar, A.A., Zaer, A., Shaher, M. and Nabil, S. (2012) Solving Singular Two-Point Boundary Value Problems Using Continuous Genetic Algorithm. *Abstract and Applied Analysis*, **2012**, Article ID: 205391.
- [6] Bakre, O.F., Wusu, A.S. and Akanbi, M.A. (2015) Solving Ordinary Differential Eq-

uations with Evolutionary Algorithms. *Open Journal of Optimization*, **4**, 69-73.

<https://doi.org/10.4236/ojop.2015.43009>

- [7] Wusu, A.S. and Akanbi, M.A. (2016) Solving Oscillatory/Periodic Ordinary Differential Equations with Differential Evolution Algorithms. *Communications in Optimization Theory*, **2016**, Article ID: 7.