Commun. Optim. Theory 2016 (2016), Article ID 7

MATHRES
Communications in Optimization Theory
Available online at http://cot.mathres.org

SOLVING OSCILLATORY/PERIODIC ORDINARY DIFFERENTIAL EQUATIONS WITH DIFFERENTIAL EVOLUTION ALGORITHMS

ASHIRIBO SENAPON WUSU*, MOSES ADEBOWALE AKANBI

Department of Mathematics, Lagos State University, Nigeria

Abstract. In this paper, the differential evolution algorithm for solving optimization problems is adapted for the solution of oscillatory/periodics ordinary differential equations. We propose a new scheme for solving second–order two–point boundary value problems by converting the differential equation to an optimization problem and the solution is obtained via differential evolution algorithm. The parameters of the proposed scheme are approximated using this optimization technique. Numerical examples show the accuracy of the proposed method compared with some existing classical methods.

Keywords. Optimization technique; Ordinary differential equation; Differential evolution; Boundary value problem.

1. Introduction

Many problems from science and engineering are modelled by Ordinary Differential Equations (ODEs) whose solutions describe the temporal evolution of the modelled processes. In most cases, however, the arising equations possess special properties like stiff and periodic behaviour and are too complex to be studied analytically. Consequently, their solutions have to be approximated by numerical methods.

There are many numerical methods available for the step-by-step integration of ordinary differential equations. However, the quest for more accurate and efficient methods is on the rise.

*Corresponding author

E-mail addresses: wussy_ash@yahoo.com (A.S. Wusu), akanbima@gmail.com (M.A. Akanbi)

Received March 18, 2016

In recent times, the idea of solving differential equations with evolutionary algorithm is gaining more attention. In this light, differential equation problems are converted to optimization problems and then solved using optimization techniques.

In the work of [2], the author showed that the classical genetic algorithm can be used to obtain approximate solutions of second–order initial value problems. For the first–order initial value problem, the concept of solution via the combination of collocation method (finite elements) and genetic algorithms was proposed by the author in [7]. In a later work, the author in [8] combined the genetic algorithm with the Nelder-Mead method for solving the second–order initial value problem $y'' = f(x,y)$. The idea of adapting neural network for the solution of second–order initial value problems was also proposed in [5]. The use of continuous genetic algorithm for the solution of the two–point second–order ordinary differential equation was discussed by the authors in [9]. In an early work by the authors in [11], the adaptation of the 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 this paper, the second-order two-point boundary value problem of the form

$$u'' = f(t,u), \;\; u(a) = \eta_1, \; u(b) = \eta_2 \tag{1}$$

with oscillatory/periodic behaviour is considered. Here, it is assumed that (1) satisfies the existence and uniqueness conditions established by Betty [12] and Petrysyhn and Yu [13]. The periodicity of (1) means that its solution can be written as a linear combination of members of the set

$$\left\{1, t, \cdots, t^K, \exp(\pm\omega t), t\exp(\pm\omega t), \cdots, t^P \exp(\pm\omega t)\right\}, \quad K, P \in \mathbb{Z}^+, \tag{2}$$

where ω is the frequency. In this paper, we show that the differential evolution algorithm can also be used to find very accurate solutions of the periodic problems (1).

2. Basic notions of differential evolution algorithms

Formally, let $f : \mathbb{R}^n \to \mathbb{R}$ be the function which must be optimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the fitness of the given candidate solution. The gradient of f is not known. The goal is to find a solution m for which $f(m) \leq f(p)$ for all p in the search-space, which would mean m is the global minimum. Maximization can be performed by considering the function $h := -f$ instead.

Let $\mathbf{x} \in \mathbb{R}^n$ designate a candidate solution (agent) in the population. The basic differential evolution algorithm can then be described as follows:

- Initialize all agents $\mathbf{x}$ with random positions in the search-space.
- Until a termination criterion is met (e.g. number of iterations performed, or adequate fitness reached), repeat the following:
  - For each agent $\mathbf{x}$ in the population do:
    - Pick three agents $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$ from the population at random, they must be distinct from each other as well as from agent $\mathbf{x}$
    - Pick a random index $R \in \{1, \ldots, n\}$ (n being the dimensionality of the problem to be optimized).
    - Compute the agent's potentially new position $\mathbf{y} = [y_1, \ldots, y_n]$ as follows:
      - For each i, pick a uniformly distributed number $r_i \equiv U(0,1)$
      - If $r_i <$ CR or $i = R$ then set $y_i = a_i + F(b_i - c_i)$ otherwise set $y_i = x_i$
      - (In essence, the new position is outcome of binary crossover of agent $\mathbf{x}$ with intermediate agent $\mathbf{z} = \mathbf{a} + F(\mathbf{b} - \mathbf{c})$.)
    - If $f(\mathbf{y}) < f(\mathbf{x})$ then replace the agent in the population with the improved candidate solution, that is, replace $\mathbf{x}$ with $\mathbf{y}$ in the population.
- Pick the agent from the population that has the highest fitness or lowest cost and return it as the best found candidate solution.

Note that $F \in [0,2]$ is called the differential weight and CR $\in [0,1]$ is called the crossover probability, both these parameters are selectable by the practitioner along with the population size NP ≥ 4.

3. Construction of the proposed scheme

Since the boundary value problem (1) is oscillatory/periodic, then its solution can be written as a linear combination of members of the set (2). We write the solution of (1) as

$$u(t)=\sum_{i=0}^{K}\alpha_i t^i+\sum_{j=0}^{P}\beta_j t^j\exp(\omega t)+\sum_{l=0}^{Q}t^l\chi_l\exp(-\omega t),\quad K,P,Q\in\mathbb{Z}. \tag{3}$$

Any component of (3) can be set to zero by setting any of the bounds K,P,Q to -1 as the case may be.

The second derivative of (3) gives

$$\begin{aligned}u''(t) &= \sum_{i=0}^{K}(i-1)i\alpha_i t^{i-2}+\sum_{j=0}^{P}\beta_j\left((j-1)jt^{j-2}e^{\omega t}+2j\omega t^{j-1}e^{\omega t}+\omega^2t^je^{\omega t}\right)\\ &+ \sum_{l=0}^{Q}\chi_l\left((l-1)lt^{l-2}e^{-\omega t}-2l\omega t^{l-1}e^{-\omega t}+\omega^2t^le^{-\omega t}\right).\end{aligned} \tag{4}$$

Substituting (3) and (4) into (1) results in

$$\begin{aligned}\sum_{i=0}^{K}(i-1)i\alpha_i t^{i-2} &+ \sum_{j=0}^{P}\beta_j\left((j-1)jt^{j-2}e^{\omega t}+2j\omega t^{j-1}e^{\omega t}+\omega^2t^je^{\omega t}\right)\\ &+ \sum_{l=0}^{Q}\chi_l\left((l-1)lt^{l-2}e^{-\omega t}-2l\omega t^{l-1}e^{-\omega t}+\omega^2t^le^{-\omega t}\right)=f(t,u(t)).\end{aligned} \tag{5}$$

Using the boundary conditions $u(a)=\eta_1,\ u(b)=\eta_2$, we have the constraints that

$$\left[\sum_{i=0}^{K}\alpha_i t^i+\sum_{j=0}^{P}\beta_j t^j\exp(\omega t)+\sum_{l=0}^{Q}t^l\chi_l\exp(-\omega t)\right]_{t=a}=\eta_1, \tag{6}$$

and

$$\left[\sum_{i=0}^{K}\alpha_i t^i+\sum_{j=0}^{P}\beta_j t^j\exp(\omega t)+\sum_{l=0}^{Q}t^l\chi_l\exp(-\omega t)\right]_{t=b}=\eta_2. \tag{7}$$

Using (5), at each node point t_n, the associated error term is given as

$$\mathscr{E}_n(t)=\left[\begin{array}{l}\sum_{i=0}^{K}(i-1)i\alpha_i t^{i-2}+\\ \sum_{j=0}^{P}\beta_j\left((j-1)jt^{j-2}e^{\omega t}+2j\omega t^{j-1}e^{\omega t}+\omega^2t^je^{\omega t}\right)+\\ \sum_{l=0}^{Q}\chi_l\left((l-1)lt^{l-2}e^{-\omega t}-2l\omega t^{l-1}e^{-\omega t}+\omega^2t^le^{-\omega t}\right)-\\ f(t,u(t))\end{array}\right]_{t=t_n}. \tag{8}$$

The associated sum of square of error terms is thus given as

$$\sum_{n=1}^{N}\mathscr{E}_n^2(t), \tag{9}$$

where $N = \frac{b-a}{h}$ and h is the steplength.

To find the values of the parameters in (3) which minimizes (9), we formulate the problem as an optimization problem in the following way:

$$
\begin{aligned}
(10) \qquad & \text{Minimize :} \quad && \sum_{n=1}^{N} \mathscr{E}_n^2(t) \\
& \text{Subject to :} && \left[\sum_{i=0}^{K} \alpha_i t^i + \sum_{j=0}^{P} \beta_j t^j \exp(\omega t) + \sum_{l=0}^{Q} t^l \chi_l \exp(-\omega t) \right]_{t=a} = \eta_1, \\
& && \left[\sum_{i=0}^{K} \alpha_i t^i + \sum_{j=0}^{P} \beta_j t^j \exp(\omega t) + \sum_{l=0}^{Q} t^l \chi_l \exp(-\omega t) \right]_{t=b} = \eta_2.
\end{aligned}
$$

Using the differential evolution algorithm we are able to obtain the values of the parameters which minimizes the expression $\sum_{n=1}^{N} \mathscr{E}_n^2(t)$. We shall refer to this proposed method as *"Differential Evolution for BVP ODEs (DEBVPODEs)"*.

4. Numerical examples

To confirm the theoretical expectations of the proposed scheme, two numerical examples are considered. The propose scheme is compared with one of the best known method: *the fourth–order Numerov Method* for solving (1). The table of *"CPU-time"* and the maximum absolute error of all computations are also given. To implement the differential evolution algorithm, the following parameters are used for all computations:
Cross Probability = 0.5; *Initial Points = Automatic*; *Penalty Function = Automatic*; *PostProcess = Automatic*; *RandomSeed* = 0; *ScalingFactor* = 0.6; *SearchPoints = Automatic*; *Tolerance = 0.0*.

Problem 1.

Consider the two–point boundary value problem

$$(11) \quad y'' - y = 2\exp(t), \;\; y(-1) = -\exp(-1), \; y(1) = \exp(1) \quad \mathrm{Exact: y(t) = t\exp(t)}.$$

Implementing the proposed scheme with $K=-1, P=1$ and $Q=-1$, we obtain the parameters of the scheme as

$$\beta_0 = \frac{8}{5739455713973277301}, \beta_1 = 1, \omega = \frac{717431772650876680}{717431772650876681}$$

and the associated approximate solution $y_{Apprx}(t)$ is

$$y_{Apprx}(t) = \frac{8}{5739455713973277301} e^{\frac{717431772650876680}{717431772650876681} t} + t e^{\frac{717431772650876680}{717431772650876681} t}. \tag{12}$$

Clearly, the proposed scheme produced a very good approximation to the exact solution.

	Maximum Absolute Error		CPU-Time (Seconds)	
i	Numerov Method	*DEBVPODEs*	Numerov Method	*DEBVPODEs*
3	2.531397E-06	1.393860E-18	1.060807E-2	3.120020E-4
4	1.583091E-07	1.393860E-18	2.402415E-2	6.240040E-4
5	9.895856E-09	1.393860E-18	5.584836E-2	1.560010E-3
6	6.185383E-10	1.734723E-18	1.394649E-1	3.432022E-3
7	3.876066E-11	1.734723E-18	4.043546E-1	7.176046E-3
8	2.767286E-12	1.734723E-18	1.246448E00	1.591210E-2

TABLE 1. Maximum Absolute Error and *CPU-time* in seconds for Problem 11 with step-size $h = 2^{-i}, i = 3(1)8$.

Problem 2.

The second problem considered in this paper is the boundary value problem

$$y'' = y - 4t\, exp(t),\ \ y(0) = 0,\ y(1) = 0 \quad \mathrm{Exact: y(t) = t(1-t)\ exp(t)}. \tag{13}$$

Implementing the proposed scheme with $K=-1, P=2$ and $Q=-1$, we obtain the parameters of the scheme as

$$\beta_0 = 0, \beta_1 = \frac{8129562136218680}{8129562136218679}, \beta_2 = -\frac{8129562136218680}{8129562136218679}, \omega = \frac{20829781963889533}{20829781963889534}$$

and the associated approximate solution $y_{Apprx}(t)$ is

$$y_{Apprx}(t) = \frac{8129562136218680}{8129562136218679} t(1-t) e^{\frac{20829781963889533}{20829781963889534} t} \tag{14}$$

again, the proposed scheme gave excellent approximation to the exact solution.

	Maximum Absolute Error		CPU-Time (Seconds)	
i	Numerov Method	*DEBVPODEs*	Numerov Method	*DEBVPODEs*
3	5.861163E-06	1.110223E-16	5.928038E-3	0.000000E00
4	3.723985E-07	3.330669E-16	1.216808E-2	6.240040E-4
5	2.327992E-08	4.440892E-16	2.776818E-2	9.360060E-4
6	1.455035E-09	4.649059E-16	6.208840E-2	1.872012E-3
7	9.090573E-11	4.649059E-16	1.500730E-1	3.432022E-3
8	5.550560E-12	7.077672E-16	4.174587E-1	8.112052E-3
9	1.155631E-12	7.077672E-16	1.281704E00	1.747211E-2

TABLE 2. Maximum Absolute Error and *CPU-time* in seconds for Problem 13 with step-size $h = 2^{-i}, i = 3(1)9$.

From the numerical results above, it is clear that the proposed scheme is quite efficient and accurate compared with the Numerov method.

5. Conclusion

In this paper, we have been able to construct and implement a scheme for converting a two–point boundary value problem to an optimization problem. The resulting optimization problem was in turn solved using the differential evolution techniques. Clearly, the theoretical expectations: *accuracy* and *efficiency* of the proposed scheme are satisfied. The proposed scheme gave excellent results compared with the Numerov method. Other evolutionary techniques can be exploited as well.

Acknowledgements

The authors are grateful to the reviewers for useful suggestions which improve the contents of this article.

REFERENCES

[1] J.C. Butcher, Numerical Methods for Ordinary Differential Equations. Wiley, (2008).

[2] D.M. George, On the appliaction of genetic algorithms to differential equations, Romanian J. Economic Forecasting 2 (2006).

[3] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 2nd Edition, (1989)

[4] H.J. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor, MI, University of Michigan press, (1975)

[5] Junaid A., Raja A.Z., Qureshi I.M. Evolutionary Computing Approach for the Solution of Initial Value Problems in Ordinary Diffential Equations, World Academic of Science, Engineering and Tecnology, 31 (2009).

[6] J.D. Lambert, Computational Methods in ODEs, John Wiley & Sons, New York, (1973).

[7] N.E. Mastorakis, Numerical Solution of Non-Linear Ordinary Differential Equations via Collocation Method (Finite Elements) and Genetic Algorithms, Proceedings of the 6th WSEAS Int. Conf. on Eolutionary Computing. Lisbon, Portugal. June 16-18, (2005), 36–42.

[8] N.E. Mastorakis, Unstable Ordinary Differential Equations: Solution via Genetic Algorithms and the method of Nelder-Mead, Proceedings of the 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation. Elounda, Greece. August 21-23, (2006), 1–6.

[9] A.A. Omar, A. Zaer, M. Shaher, S. Nabil, Solving singular two-point boundary value problems using continuous genetic algorithm, Abst. Appl. Anal. 2012 (2012), 205391.

[10] Z. Michalewiz, Genetic Algorithm + Data Structure = Evolution Programs, 2nd Edition, New York: Springer-Verlag, Berlin (1994)

[11] B.O. Fatimah, W.A. Senapon, A.M. Adebowale, Solving ordinary differential equations with evolutionary algorithms, Open J. Optim. 4 (2015), 69-73.

[12] T. Betty, Uniqueness of initial value problems, Divulgaciones Matemmaticas, 5(1/2), (1997), 39–41.

[13] W.V. Petryshyn, Z.S. Yu, Periodic solutions of nonlinear second-order differential equations which are not solvable for the highest derivative, J. Math. Anal. Appl. 89 (1982), 462–488.