

# Application of Differential Evolution in the Solution of Stiff System of Ordinary Differential Equations

<sup>1</sup>Ashiribo Senapon Wusu, <sup>2</sup>Olusola Aanu Olabanjo, <sup>3</sup>Benjamin Segun Aribisala

<sup>1</sup>Department of Mathematics, Lagos State University, Ojo, Nigeria;

<sup>2,3</sup>Computer Science Department, Lagos State University, Ojo. Lagos, Nigeria;

<sup>1</sup>ashiribo.wusu@lasu.edu.ng; <sup>2</sup>olusola.olabanjo@lasu.edu.ng; <sup>3</sup>benjamin.aribisala@lasu.edu.ng

## ABSTRACT

In recent times, the adaptation of evolutionary optimization algorithms for obtaining optimal solutions of many classical problems is gaining popularity. To accurately solve stiff systems in Ordinary Differential Equation (ODEs), the use of more accurate numerical techniques is of great importance. In this paper, optimal approximate solutions of initial-valued stiff system of ODEs are obtained by converting the ODE into constrained optimization problem. The latter is then solved via differential evolution algorithm. To illustrate the efficiency of the proposed approach, two numerical examples were considered. This approach showed significant improvement on the accuracy of the results produced compared with existing methods discussed in literature.

**Keywords:** Ordinary Differential Equation; Initial Value Problems; Stiff System; Optimization; Differential Evolution.

## 1 Introduction

Initial value problems involving systems of first-order ordinary differential equation are often encountered in many fields. Most of these systems exhibit a phenomenon known as stiffness.

**Definition 1.1** *The linear system  $\dot{x} = Ax + \phi(t)$  is said to be stiff if  $\text{Re } \lambda_i < 0$ ,  $i = 1, 2, \dots, m$ , and  $\max_{i=1,2,\dots,m} |\text{Re } \lambda_i| \gg \min_{i=1,2,\dots,m} |\text{Re } \lambda_i|$ , where  $\lambda_i, i = 1, 2, \dots, m$  are the eigenvalues of  $A$ . The ratio is called the stiffness ratio. [15]*

$$\left[ \max_{i=1,2,\dots,m} |\text{Re } \lambda_i| \right] : \left[ \min_{i=1,2,\dots,m} |\text{Re } \lambda_i| \right] \quad (1)$$

Such systems are always very difficult to solve when using classical techniques [9]. The difficulty of solving stiff initial value problems was clearly identified in early 1950s when the authors in [9] published one of the first papers in which the problem of stiffness was stated. Subsequently, a whole lot of methods and algorithms have been proposed for solving problems that exhibit stiffness. The author in [6] introduced a class of extended backward differentiation formulae suitable for the integration of stiff systems of autonomous initial value problems. In a later work, [8], the author proposed classes of predictor-corrector method involving the second derivative using the extended backward differentiation formulae. Obtaining the solution of stiff differential equations using exponentially fitted formula was first proposed by the

authors in [16]. Following the work of the authors in [16], several method based on the concept of exponential fitting have been proposed by various authors [1, 3, 6, 7, 8, 10, 12, 19, 20].

In recent years however, interest in the application of evolutionary algorithm for solving many problems is on the increase. Evolutionary algorithm has also been adopted in solving ODE and its application to real life problems is on the increase. Problems in ordinary differential equations are converted to optimization problems and then solved using optimization techniques. In this direction, the classical genetic algorithm was used to obtain approximate solutions of second-order initial value problems by the authors in [11]. The author in [18] computed approximate solutions of first-order initial value problem via the combination of collocation method (finite elements) and genetic algorithms. The combination of genetic algorithm with the Nelder-Mead method for solving the second-order initial value problem of the form  $u'' = f(x, y)$  was proposed in [17]. The use of neural network for the approximate solution of second-order initial value problems was proposed by the authors in [14]. In the work of the authors in [21], the use of continuous genetic algorithm in obtaining the solution of the two-point second-order ordinary differential equation was presented. The use of differential evolution algorithm in obtaining approximate solution of second-order initial value problem of the form  $u'' + p(t)u' + q(t)y = r(t)$  was proposed by the authors in [5]. In [23], 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 was obtained via the application of differential evolution algorithm. In this work, we show that the problem of finding approximate solution of a stiff system of first-order ordinary differential equation can be converted to an optimization problem. The resulting optimization problem is then solved using differential evolution. It is assumed that the solution can be approximated by a linear combination of exponential terms. The differential evolution algorithm is used to optimize the coefficients of the terms of the solutions.

## 2 Differential Evolution Algorithm

Differential evolution is a simple stochastic function minimizer which details can be found in many standard texts. However, we give a brief overview of the algorithm as described in [22]. Formally, let  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  be the function to 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  $R \in \{1, \dots, n\}$  ( $n$  being the dimensionality of the problem to be optimized).

Compute the agent's potentially new position  $\mathbf{y} = [y_1, \dots, y_n]$  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  $NP \geq 4$ .

### 3 Procedure for Proposed Technique

In this work, we considered the system of  $r$  equations

$$\begin{aligned} \frac{d\mathbf{u}}{dt} &= \mathbf{f}(t, u_1, u_2, \dots, u_r), \\ \mathbf{u}(t_0) &= \boldsymbol{\eta}; \quad a \leq t \leq b \end{aligned} \tag{2}$$

where  $\mathbf{u} = [u_1, u_2, \dots, u_r]^T$ ,  $\mathbf{f} = [f_1, f_2, \dots, f_r]^T$ ,  $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_r]^T$ . It is assumed that the solution  $\mathbf{u}(t)$  of (2) can be written as

$$\mathbf{u}(t) = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,r} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,r} \\ \vdots & & \ddots & \vdots \\ \alpha_{r,1} & \alpha_{r,2} & \dots & \alpha_{r,r} \end{pmatrix} \begin{pmatrix} \exp(\omega_1 t) \\ \exp(\omega_2 t) \\ \vdots \\ \exp(\omega_r t) \end{pmatrix} \tag{3}$$

where  $\alpha_{i,j}$ ,  $\omega_i$ ,  $i, j = 1, 2, \dots, r$  are real constants which values are to be determined by the proposed procedure.

Substituting the derivative of (3) into (2) gives

$$\begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,r} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,r} \\ \vdots & & \ddots & \vdots \\ \alpha_{r,1} & \alpha_{r,2} & \dots & \alpha_{r,r} \end{pmatrix} \begin{pmatrix} \omega_1 \exp(\omega_1 t) \\ \omega_2 \exp(\omega_2 t) \\ \vdots \\ \omega_r \exp(\omega_r t) \end{pmatrix} \approx \begin{pmatrix} f_1(t, u_1, u_2, \dots, u_r) \\ f_2(t, u_1, u_2, \dots, u_r) \\ \vdots \\ f_r(t, u_1, u_2, \dots, u_r) \end{pmatrix} \tag{4}$$

Using the initial conditions given in (2) we have,

$$\mathbf{u}(t_0) = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,r} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,r} \\ \vdots & & \ddots & \vdots \\ \alpha_{r,1} & \alpha_{r,2} & \dots & \alpha_{r,r} \end{pmatrix} \begin{pmatrix} \exp(\omega_1 t_0) \\ \exp(\omega_2 t_0) \\ \vdots \\ \exp(\omega_r t_0) \end{pmatrix} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_r \end{pmatrix} \tag{5}$$

From (4), the error expression  $\mathcal{E}_{i,n}(t)$  at each node point  $t_n$  is obtained as

$$\mathcal{E}_{i,n}(t) = \left[ \begin{pmatrix} \alpha_{i,1} & \alpha_{i,2} & \cdots & \alpha_{i,r} \end{pmatrix} \begin{pmatrix} \omega_1 \exp(\omega_1 t) \\ \omega_2 \exp(\omega_2 t) \\ \vdots \\ \omega_r \exp(\omega_r t) \end{pmatrix} - \left( f_i(t, u_1, u_2, \dots, u_r) \right) \right]_{t=t_n} \quad (6)$$

Now we need to find values of  $\alpha_{i,j}$ ,  $\omega_i$ ,  $i, j = 1, 2, \dots, r$ , that minimizes the expression

$$\sum_{i=1}^r \sum_{n=1}^{\frac{b-a}{h}} \mathcal{E}_{i,n}^2(t) \quad (7)$$

where  $h$  is the steplength.

Using (7) and (5), the problem of finding approximate solution of (2) is now formulated as an optimization problem in the following manner:

$$\begin{aligned} \text{Minimize:} &= \sum_{i=1}^r \sum_{n=1}^{\frac{b-a}{h}} \mathcal{E}_{i,n}^2(t) \\ \text{Subject to:} &= \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,r} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,r} \\ \vdots & & \ddots & \vdots \\ \alpha_{r,1} & \alpha_{r,2} & \cdots & \alpha_{r,r} \end{pmatrix} \begin{pmatrix} \exp(\omega_1 t_0) \\ \exp(\omega_2 t_0) \\ \vdots \\ \exp(\omega_r t_0) \end{pmatrix} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_r \end{pmatrix} \end{aligned} \quad (8)$$

In this work, we solved (8) using the differential evolution algorithm. Suitable values of  $\alpha_{i,j}$ ,  $\omega_i$ ,  $i, j = 1, 2, \dots, r$  which satisfies ((8)) were obtained using this technique. We shall refer to this scheme as *DE4stiffODE*.

## 4 Numerical Examples

In this section, the efficiency of the proposed method is investigated on two test problems. The results obtained by our approach are compared with those obtained with the methods proposed by authors in the literature.

### 4.1 Example 1

The two dimensional nonlinear Kaps problem

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} -1002u_1 + 1000u_2^2 \\ u_1 - u_2(1 + u_2) \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (9)$$

with exact solution

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} e^{-2t} \\ e^{-t} \end{pmatrix} \quad (10)$$

is considered. Using our proposed technique, the problem is solved over the interval  $[0,1]$  with a steplength  $h = 0.1$ . As expected, our method approximated the solution almost exactly as

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{pmatrix} \begin{pmatrix} \exp(\omega_1 t) \\ \exp(\omega_2 t) \end{pmatrix} \tag{11}$$

where

$$\left. \begin{aligned} \alpha_{1,1} &= \frac{1100634814035122250906678637333}{1100634814035122250906678649769} \\ \alpha_{1,2} &= \frac{90095597994998040}{7973814067594708980711815298377729856804893} \\ \alpha_{2,1} &= \frac{370946476580559891545427610583}{497552805822304845199269901657} \\ \alpha_{2,2} &= \frac{497552805822304845199269909564}{255655159831828977612244310059} \\ \omega_1 &= -\frac{127827579915914488806122124384}{997456764304870362562667731346} \\ \omega_2 &= -\frac{997456764304870362562667511549}{997456764304870362562667511549} \end{aligned} \right\} \tag{12}$$

A comparison of our results with those obtained by the methods of authors in [4, 13, 24] is presented in Table 1. We shall respectively denote by "SDM10" and "SDM14" the methods of order 10 and order 14 proposed in [24]. Similarly, the method proposed in [4] shall be denoted by "NMTD". The 8th-order method proposed in [13] shall be denoted by "SDAM8".

**Table 1. The absolute error of our "DE4stiffODE" method compared with "SDM10", "SDM14" and "NMTD" methods at some values of t on Example 1.**

t	$y_i$	SDM10	SDM14	NMTD	DE4stiffODE
5	$y_1$	$4.6889 \times 10^{-003}$	$5.8258 \times 10^{-002}$	$9.7751 \times 10^{-004}$	0.0000
	$y_2$	$4.8326 \times 10^{-003}$	$3.2259 \times 10^{-002}$	$1.0556 \times 10^{-006}$	0.0000
50	$y_1$	$1.4156 \times 10^{-002}$	$6.7358 \times 10^{-003}$	$2.6559 \times 10^{-005}$	$2.1793 \times 10^{-048}$
	$y_2$	$1.9419 \times 10^{-002}$	$2.6181 \times 10^{-002}$	$1.1303 \times 10^{-007}$	0.0000
150	$y_1$	$6.3883 \times 10^{-004}$	$2.4686 \times 10^{-006}$	$8.7651 \times 10^{-009}$	$8.1072 \times 10^{-092}$
	$y_2$	$6.1134 \times 10^{-003}$	$5.3608 \times 10^{-004}$	$2.5430 \times 10^{-009}$	0.0000
250	$y_1$	$1.7895 \times 10^{-005}$	$8.1636 \times 10^{-010}$	$2.8923 \times 10^{-012}$	$3.0159 \times 10^{-135}$
	$y_2$	$1.2275 \times 10^{-003}$	$9.7597 \times 10^{-006}$	$6.0129 \times 10^{-011}$	0.0000
500	$y_1$	$1.6011 \times 10^{-009}$	$1.6165 \times 10^{-018}$	$5.7208 \times 10^{-021}$	$8.0500 \times 10^{-244}$
	$y_2$	$1.5267 \times 10^{-005}$	$4.3431 \times 10^{-010}$	$4.2348 \times 10^{-015}$	0.0000

**Table 2. The absolute error of our "DE4stiffODE" method compared with "SDAM" method at t=1 and t=10 on Example**

Absolute Error						
SDAM				DE4stiffODE		
t	h	$u_1(t)$	$u_2(t)$	h	$u_1(t)$	$u_2(t)$
1	0.008	$1.6348 \times 10^{-014}$	0.0000	0.1	0.0000	0.0000
10	0.006	$2.4815 \times 10^{-024}$	$2.0329 \times 10^{-020}$		0.0000	0.0000

It is clear from Table 1 and Table 2 that our proposed approach gave better solution compared with existing methods in literature.

## 4.2 Example 2

The second problem considered is the following system of two linear equations

$$\begin{pmatrix} u_1' \\ u_2' \end{pmatrix} = \begin{pmatrix} -u_1 + 95u_2 \\ -u_1 - 97u_2 \end{pmatrix}, \quad \begin{pmatrix} u_1(0) \\ u_2(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (13)$$

with exact solution

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} -\frac{48}{47}e^{-96t} + \frac{95}{47}e^{-2t} \\ \frac{48}{47}e^{-96t} - \frac{1}{47}e^{-2t} \end{pmatrix}. \quad (14)$$

Problem (13) has been studied by authors in [2, 4, 6, 7, 10, 12]. Using a steplength of  $h = 0.7$ , we solve the problem in the interval  $[0,1]$  to obtain the following solution:

$$\begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} \\ \alpha_{2,1} & \alpha_{2,2} \end{pmatrix} \begin{pmatrix} \exp(\omega_1 t) \\ \exp(\omega_2 t) \end{pmatrix} \quad (15)$$

where

$$\left. \begin{aligned} \alpha_{1,1} &= -\frac{48}{47} \\ \alpha_{1,2} &= \frac{95}{47} \\ \alpha_{2,1} &= \frac{48}{47} \\ \alpha_{2,2} &= -\frac{1}{47} \\ \omega_1 &= -96 \\ \omega_2 &= -2 \end{aligned} \right\} \quad (16)$$

Again, as expected, our approach approximated the solution exactly. For comparison purpose, we shall denote respectively by "ABOT", "NMTD", "Cash4", "Cash5", "SDEBDF", and "J-K" the methods proposed by the authors in [2], [4], [6], [7], [10], and [12]. The efficiency of our proposed scheme can be seen from the choice of the steplength. Even with a larger steplength, our scheme gave a better result compared with the methods discussed in the literature.

**Table 3. Comparison of absolute error of Example 2 at  $t = 1$  with methods discussed in literature**

Steplength	Method	$u_1(t)$	$u_2(t)$
0.0625	"NMTD"	$3.0000 \times 10^{-008}$	$4.0000 \times 10^{-010}$
	"SDEBDF"	$3.0000 \times 10^{-006}$	$4.0000 \times 10^{-007}$
	"ABOT"	$3.0000 \times 10^{-006}$	$4.0000 \times 10^{-007}$
	"Cash5"	$1.0000 \times 10^{-008}$	$4.0000 \times 10^{-007}$
	"Cash4"	$3.0000 \times 10^{-007}$	$4.0000 \times 10^{-007}$
	"J-K"	$3.0000 \times 10^{-007}$	$4.0000 \times 10^{-007}$
.1	"DE4stiffODE"	0.0000	0.0000

## 5 Conclusion

The basic steps involved in converting a stiff system of differential equation problem to an optimization problem and obtaining approximate solution via differential evolution have been shown. Results of numerical implementation show that our approach can be used to obtain almost-exact solution. The comparison of our method with those discussed in literature clearly shows the degree of accuracy of our method.

## REFERENCES

- [1] Abhulimen C.E. and F.O. Otunta, A Sixth Order Multiderivative Multistep Methods for Stiff Systems of Differential Equations. *International Journal of Numerical Mathematics (IJNM)*, 2006. 2(1): p. 248–268
- [2] Abhulimen C.E. and F.O. Otunta, A new class of exponential fitting for initial value problems in ordinary differential equations. *Journal of Nigerian Mathematical Society*, 28, 2009.
- [3] Abhulimen C.E., A class of exponentially-fitted third derivative methods for solving stiff differential equations. *International Journal of Physical Science*, 2008. 3(8): p. 188–193
- [4] Adesanya A.O., R. O. Onsachi, and M. R. Odekunle, New algorithm for first order stiff initial value problems, *FASCICULI MATHEMATICI*, 2017. 58: p. 1–8.
- [5] Bakre O. F., A. S. Wusu and M. A. Akanbi, Solving ordinary differential equations with evolutionary algorithms. *Open Journal of Optimization*, 2015. 4: p. 69–73
- [6] Cash J.R., On the integration of stiff systems of O.D.E.s using extended backward differentiation formulae. *SIAM J. Numerical Math.*, 1980. 34: p. 235–246
- [7] Cash J.R., On exponentially fitting of composite multiderivative Linear Methods. *SIAM J. Numerical Anal.* 1981. 18(5): p. 808–821
- [8] Cash J.R., Second Derivative Extended Backward Differentiation Formulas for the Numerical Solution of stiff Systems. *SIAM J. Numerical Anal.*, 1981. 18: p. 21–36

- [9] Curtiss C.F. and J.D. Hirschfelder, *Integration of stiff equations*. *Proc. Nat. Acad. Sci.*, 1952. 38: p. 235–243
- [10] Ehigie J.O., S.A. Okunuga, and A.B. Sofoluwe, A class of exponentially fitted second derivative extended backward differentiation formula for solving stiff problems. *FASCICULI MATHEMATICI*, 2013. 51: p. 71–84
- [11] George D.M., On the application of genetic algorithms to differential equations. *Romanian Journal of Economic Forecasting*, 2006. 3(2): p. 5–9
- [12] Jackson L.W. and S.K. Kenue, A Fourth Order Exponentially Fitted Method. *SIAM J. Numer. Anal.*, 1974. 11: p. 965–978
- [13] Jator S.N. and R. K. Sahi, Boundary value technique for initial value problems based on adams–type second derivative methods. *International Journal of Mathematical Education in Science and Technology*, First published on: 07 June 2010 (*iFirst*), 2010. p. 1–8,
- [14] Junaid A., A. Z. Raja, and I. M. Qureshi, Evolutionary computing approach for the solution of initial value problems in ordinary differential equations. *World Academic of Science, Engineering and Technology*, 2009. 55: p. 578–581
- [15] Lambert J.D., *Computational Methods in ODEs*. John Wiley & Sons, New York, 1973.
- [16] Liniger W.S. and R.A. Willoughby, Efficient Integration methods for Stiff System of ODEs. *SIAM J. Numerical Anal.*, 1970. 7: p. 47–65
- [17] Mastorakis N.E., Unstable ordinary differential equations: Solution via genetic algorithms and the method of nelder-mead. In *Proceedings of the 6th WSEAS Int. Conf. on Systems Theory & Scientific Computation. Elounda, Greece.*, August 21–23, 2006. p. 1–6
- [18] Mastorakis N.E., Numerical solution of non-linear ordinary differential equations via collocation method (finite elements) and genetic algorithms. In *Proceedings of the 6th WSEAS Int. Conf. on Evolutionary Computing. Lisbon, Portugal.*, Springer Berlin Heidelberg, June 16–18, 2005. p. 36–42.
- [19] Okunuga S.A., A Class of Multiderivative Composite Formula for Stiff Initial Value Problems. *Advances in Modeling & Analysis*, 1999. 35(2): p. 21–32
- [20] Okunuga S.A., A Fourth Order Composite Two-Step Method for Stiff Problems. *International Journal of Computer Mathematics*, 1999. 72(1): p.39–47
- [21] Omar A.A., A. Zaer, M. Shaher, and S. Nabil, Solving singular two-point boundary value problems using continuous genetic algorithm. *Abstract and Applied Analysis*, 2012. Article ID 205391, vol. 2012: p.1-25
- [22] Wikipedia. Differential evolution — wikipedia, the free encyclopedia, 2019. [Online; accessed 1 December 2019].
- [23] Wusu A.S. and Akanbi M.A., Solving oscillatory/periodic ordinary differential equations with differential evolution algorithms. *Communications in Optimization Theory*, 2016. 2016: p. 1–8