International Journal of Scientific Advances
ISSN: 2708-7972
Volume: 3 | Issue: 4 | Jul - Aug 2022 Available Online: www.ijscia.com
DOI: 10.51542/ijscia.v3i4.35

Approximate Analytical Solution of Unstable Ordinary Differential Equation Using Differential Evolution Algorithm

Ashiribo Senapon WUSU[1*], Olusola Aanu OLABANJO[2] and Manuel MAZZARA[3]

[1]Department of Mathematics, Lagos State University, Lagos, Nigeria
[2]Department of Mathematics, Morgan State University, Baltimore, MD 21251, USA
[3]Institute of Software Development and Engineering, Innopolis University Innopolis 420500, Russia

E-mail: ashiribo.wusu@lasu.edu.ng; olola57@morgan.edu; m.mazzara@innopolis.ru

*Corresponding author details: Ashiribo Senapon WUSU; ashiribo.wusu@lasu.edu.ng

ABSTRACT
The application of evolutionary optimization algorithms in problem solving is currently gaining wide popularity. Use of Differential Evolution (DE) algorithm in obtaining analytically approximate solution of unstable second–order initial value Ordinary Differential Equation (ODE) is presented in this work. The methodology involves solving an associated problem of optimization with constrains to get an analytically approximate solution for the ODE under consideration. Three test cases were used to demonstrate the efficiency of our method. In comparison with other methods discussed in the literature, our method gave significant improvement on the accuracy of the obtained results.

Keywords: unstable; ordinary differential equation; initial value problems; optimization differential evolution

INTRODUCTION
Several numerical methods exist for obtaining approximate solutions of different classes of ODEs [8, 12]. For many of these methods however, accumulated errors give impulse in the unstable term, hence, they are usually less efficient when applied to unstable ODEs [7]. To tackle this challenge, one direction of interest lies in applying evolutionary optimization techniques. This approach requires that the ODE be formed as an optimization problem and then solved using some evolutionary algorithms [1, 2, 11]. The author in [9] obtained approximate solutions of first–order initial value problems by combining collocation method together with genetic algorithms. By combining Nelder–Mead method together with genetic algorithm, the authors in [6, 10] solved second–order initial value problems. Neural network was introduced in [2] to obtain approximate solution. Authors in [1] proposed the use of genetic algorithm with continuity to get solution of two–point second–order ODE. In [11, 3], the authors, respectively applied differential evolution algorithm to get approximate solutions of u^''+p(t)u^'+q(t)y=r(t) and u^''=f(t,u); u(a)=η_1; u(b)=η_2. Stiff systems of first–order ODEs were solved in [4] using differential evolution algorithm. Approximate solutions of problem with singularities were obtained using the Nelder–Mead algorithm in [5]. In this research work, the algorithm of differential evolution was implemented to obtain approximate analytical solution of unstable second–order initial value ODE. Differential evolution is one of the commonly used algorithms of the family of evolutionary computing.

Unlike its counterparts, it can conveniently handle nonlinear and non–differentiable multi–dimensional objective functions, while requiring very few control parameters. With these characteristics, it becomes very easy and more practical to use. An overview of the algorithm is described in [13] and details can be found in many standard texts.

PROPOSED METHOD
Consider the unstable second–order ODE

$$u'' = f(t,u,u'), \quad u(t_0)=u_0, u'(t_0)=u'_0 \quad t\in[t_0,b]. \tag{2.1}$$

This work assume the solution of Eq. (2.1) can be expressed as

$$u(t) = \sum_{i=0}^{k} \xi_i t^i + \sum_{j=1}^{2} \alpha_j e^{\omega_j t}, \quad k\in\mathbb{Z}^+ \tag{2.2}$$

where $\xi_i, \alpha_1, \alpha_2, \omega_1, \omega_2$ are real constants whose values are to be determined by our proposed approach. Substituting Eq. (2.2) together with its derivatives into Eq. (2.1) results in

$$\sum_{i=2}^{k} i(i-1)\xi_i t^{i-2} + \sum_{j=1}^{2} \alpha_j \omega_j^2 e^{\omega_j t} = f(t,u,u') \tag{2.3}$$

Now, considering the initial conditions, we have the constraints that

$$\left.\begin{aligned}\left[\sum_{i=0}^{k} \xi_i t^i + \sum_{j=1}^{2} \alpha_j e^{\omega_j t}\right]_{t=t_0} &= u_0,\\ \left[\sum_{i=1}^{k} i\xi_i t^{i-1} + \sum_{j=1}^{2} \alpha_j \beta_j e^{\beta_j t}\right]_{t=t_0} &= u_0'\end{aligned}\right\} \quad (2.4)$$

At each node point t_n, we require that

$$\mathcal{E}_n(t) = \left[\sum_{i=2}^{k} i(i-1)\xi_i t^{i-2} + \sum_{j=1}^{2} \alpha_j \omega_j^2 e^{\omega_j t} - f(t,u,u')\right]_{t=t_n} \approx 0 \quad (2.5)$$

To solve the above problem, we need to find the set $\{\xi_i, \alpha_j, \omega_j | i = 0(1)k, j = 1,2\}$, which minimizes the sum of square of the error at each node point given by

$$\sum_{n=1}^{N} \mathcal{E}_n^2(t) \quad (2.6)$$

where $N = \frac{b-t_0}{h}$ and h is the step-length. We now formulate the problem as an optimization problem in the following way:

$$\begin{aligned}&\text{Minimize:} && \sum_{n=1}^{N} \mathcal{E}_n^2(t)\\ &\text{Subject to:} && \left.\begin{aligned}&\left[\sum_{i=0}^{k} \xi_i t^i + \sum_{j=1}^{2} \omega_j e^{\beta_j t}\right]_{t=t_0} = u_0,\\ &\left[\sum_{i=1}^{k} i\xi_i t^{i-1} + \sum_{j=1}^{2} \alpha_j \beta_j e^{\beta_j t}\right]_{t=t_0} = u_0'\end{aligned}\right\}\end{aligned} \quad (2.7)$$

We shall now use the DE algorithm to obtain real constants $\{\xi_i, \alpha_j, \omega_j | i = 0(1)k, j = 1,2\}$ which optimizes Eq. (minimizer). Our proposed solution shall be referred to as: "Differential Evolution for Unstable ODEs (DEUODEs)".

TEST CASES

Here, we implement our scheme on three test cases. To demonstrate the accuracy and efficiency of our proposed scheme, we compare our results with those produced by the well–known classical Runge-Kutta Nystrom scheme.

For each of the considered cases, comparison of the maximum absolute errors together with the execution–time are presented. The default values used in the implementation of the DE algorithm on the test cases are given in Table 1.

TABLE 1: DE parameter values used in the implementation

Parameter name	Values
Cross Probability	0.5
Initial Points	Automatic
Penalty Function	Automatic
Post Process	True
Random Seed	0
Scaling Factor	0.7
Search Points	All
Tolerance	0.000001

A "10th Generation, Core i7 Intel" processor computer was used for the computations carried out in this section.

Problem 1

Consider the unstable ODE

$$u''(t) - 10u'(t) - 11u(t) = 0. \quad (3.1)$$

Eq. (3.1) has the theoretical solution

$$u(t) = C_1 \exp(11t) + C_2 \exp(-t). \quad (3.2)$$

In this case, the accumulated errors give impulse in the unstable term, $\exp(11t)$, hence it becomes tedious to find a numerical solution that will be an approximation of $y(t) = \exp(-t)$.

To overcome this challenge, we choose $k = 0$ in Eq. (2.2) and solve Eq. (3.1) together with the initial conditions: $u(0) = 1, u'(0) = -1$. Using a steplength of $h = 0.01$, we use the DE algorithm to obtain values of the associated real constants as given in Table 2.

TABLE 2: Estimated method coefficient values for Problem 1

Constants	**Values**
ξ_0	$-\frac{58596483280504527}{1408790309017455389363402685223409483351015 8}$
α_1	$\frac{335412}{5494624487264310074760429017 93}$
α_2	$\frac{1439621835603878864855901 72154}{1439621835603878864855902 59435}$
ω_1	$-\frac{7077876696229006436287339 52243}{7831739379532137122477973 60578}$
ω_2	$-\frac{2526056395966082493073880999103}{2526056395966082493073880861180}$

The analytical approximate solution is given as Eq. (3.3).

$$u(t) = -\frac{58596483280504527}{14087903090174553893634026852234094833510158} + \frac{335412}{549462448726431007476042901793}\exp\left(-\frac{707787669622900643628733952243}{783173937953213712247797360578}t\right) + \frac{143962183560387886485590172154}{143962183560387886485590259435}\exp\left(-\frac{2526056395966082493073880999103}{2526056395966082493073880861180}t\right) \quad (3.3)$$

Table 3 shows the absolute maximum error and execution time (seconds) of our technique compared with the classical Runge–Kutta Nystrom method for varying steplength.

TABLE 3: Absolute maximum error and execution-time in seconds for Problem 3.1 with step-size $h = 2^{-i}, i = 3(1)9$

	Absolute Maximum Error		Execution-Time (Seconds)	
i	Runge-Kutta Nystrom Method	DEUODEs	Runge-Kutta Nystrom Method	DEUODEs
3	5.703708E-01	1.110223E-16	4.687500E-03	0.000000
4	5.384612E-02	1.110223E-16	7.812500E-03	0.000000
5	3.966185E-03	1.110223E-16	1.562500E-02	0.000000
6	2.671148E-04	1.110223E-16	2.968750E-02	1.562500E-03
7	1.730622E-05	1.110223E-16	5.937500E-02	1.562500E-03
8	1.100942E-06	1.110223E-16	1.234375E-01	3.125000E-03
9	6.941755E-08	1.110223E-16	2.515625E-01	6.250000E-03

Problem 2

The second case is given as

$$u''(t) = 100u(t). \tag{3.4}$$

Eq. (3.4) has the theoretical solution

$$u(t) = C_1\exp(10t) + C_2\exp(-10t). \tag{3.5}$$

However, Eq. (3.4) with the initial conditions: $u(0) = 1, u'(0) = -10$ has its solution as

$$u(t) = \exp(-10t). \tag{3.6}$$

Again, the accumulated errors gave impulse in the unstable term, $\exp(10t)$ in Eq. (3.5). Applying the DE algorithm again but choosing $k = 1$ in Eq. (2.2), values of the associated real constants are given in Table 4.

TABLE 4: Estimated method coefficient values for Problem 2

Constants	Values
ξ_0	$-\frac{1866130124028508}{27968489541978442147822524598346085350438563}$
ξ_1	$\frac{5490012867360566}{88055038859759348231995436075796820268469007}$
α_1	$\frac{235}{2033336901794996078850609459979}$
α_2	$\frac{593642189642882704140492449808}{593642189642882704140492449837}$
ω_1	$-\frac{1864702798268403069139083489522}{778691675612344064906052135217}$
ω_2	$-\frac{3981513280893406487103789665829}{398151328089340648710378966572}$

The analytically approximate solution is given as Eq. (3.7).

$$\begin{aligned} u(t) &= -\frac{1866130124028508}{27968489541978442147822524598346085350438563} + \\ &= \frac{5490012867360566}{88055038859759348231995436075796820268469007} t + \\ &\frac{235}{2033336901794996078850609459979} \exp\left(-\frac{1864702798268403069139083489522}{778691675612344064906052135217} t\right) + \\ &\frac{593642189642882704140492449808}{593642189642882704140492449837} \exp\left(-\frac{3981513280893406487103789665829}{398151328089340648710378966572} t\right) \end{aligned} \tag{3.7}$$

The absolute maximum error and execution time (seconds) of the classical Runge–Kutta Nystrom and our technique in comparison for different step–lengths is shown in Table 5.

TABLE 5: Absolute maximum error and execution-time in seconds for Problem 3.2 with step-size $h = 2^{-i}, i = 3(1)9$

	Absolute Maximum Error		Execution-Time (Seconds)	
i	Runge-Kutta Nystrom Method	DEUODEs	Runge-Kutta Nystrom Method	DEUODEs
3	1.054952E02	5.551115E-17	6.250000E-03	0.000000
4	8.181046E00	1.110223E-16	7.812500E-03	0.000000
5	5.380984E-01	1.110223E-16	1.562500E-02	0.000000
6	3.405028E-02	1.110223E-16	3.125000E-02	3.125000E-03
7	2.134674E-03	1.110223E-16	6.406250E-02	1.562500E-03
8	1.335191E-04	1.110223E-16	1.515625E-01	3.125000E-03
9	8.34654E-06	1.110223E-16	3.093750E-01	2.187500E-02

Problem 3

The third case considered is given as

$$u''(t) = -u'(t) + 2u(t). \tag{3.8}$$

The theoretical solution of Eq. (problem3) is given as

$$u(t) = C_1 \exp(-2t) + C_2 \exp(t). \tag{3.9}$$

Here, Eq. (3.8) has the initial conditions: $u(0) = 1, u'(0) = 1$ and the exact solution also given as

$$u(t) = \exp(t). \tag{3.10}$$

Similarly, the unstable term, $\exp(-2t)$ in Eq. (3.9) has impulse of accumulated errors. Here, we apply the DE algorithm again but choose $k = 0$ in Eq. (2.2), values of the associated real constants are given in Table 6.

TABLE 6: Estimated method coefficient values for Problem 3

Constants	Values
ξ_0	$\frac{13976606731205971}{45451913496634381658353114106888650226127076}$
α_1	$-\frac{8}{168513967809568476850041066 73}$
α_2	$\frac{20868844604101529107556979602 09}{20868844604101529107556979598 60}$
ω_1	$\frac{2943251971032013463444856275 39}{12577573802429244655846772436 92}$
ω_2	$\frac{884175781706239643559737124 78}{884175781706239643559737124 83}$

The analytically approximate solution is hence given as Eq. (3.11).

$$u(t) = \frac{13976606731205971}{45451913496634381658353114106888650226127076} + -\frac{8}{16851396780956847685004106673}\exp\left(\frac{294325197103201346344485627539}{1257757380242924465584677243692}t\right) + \frac{2086884460410152910755697960209}{2086884460410152910755697959860}\exp\left(\frac{88417578170623964355973712478}{88417578170623964355973712483}t\right) \tag{3.11}$$

Again, Table 7 shows the absolute maximum error and execution time (seconds) of our technique compared with the classical Runge–Kutta different step–length.

TABLE 7: Absolute maximum error and execution-time in seconds for Problem 3.3 with step-size $h = 2^{-i}, i = 3(1)9$

	Absolute Maximum Error		Execution-Time (Seconds)	
i	Runge-Kutta Nystrom Method	DEUODEs	Runge-Kutta Nystrom Method	DEUODEs
3	1.245609E-05	0.000000	3.125000E-03	0.000000
4	7.977266E-07	0.000000	7.812500E-03	0.000000
5	5.044025E-08	4.440892E-16	1.406250E-02	1.562500E-03
6	3.170410E-09	4.440892E-16	2.968750E-02	0.000000
7	1.987064E-10	4.440892E-16	6.093750E-02	0.000000
8	1.243761E-11	4.440892E-16	1.171875E-01	4.687500E-03
9	7.780443E-13	8.881784E-16	2.390625E-01	7.812500E-03

CONCLUSION

We conclude here that we have been able to obtain analytically approximate solutions of unstable ODEs using differential evolution algorithm. Compared to the Runge–Kutta Nystrom method, the accuracy and efficiency of our approach is clearly demonstrated with the three test cases considered. In future works, application of other evolutionary techniques can be considered.

REFERENCES

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

[2] A. Junaid, A. Z. Raja, I. M. Qureshi, Evolutionary computing approach for the solution of initial value problems in ordinary diffential equations, World Academic of Science, Engineering and Tecnology 55 (2009), 578–581.

[3] A. S. Wusu, M. A. Akanbi, Solving oscillatory/periodic ordinary differential equations with differential evolution algorithms, Communications in Optimization Theory 2016 (2016) 1–8.

[4] A. S. Wusu, O. A. Olabanjo, B. S. Aribisala, Application of differential evolution inthe solution of stiff system of ordinary differential equations, Transactions on Machine Learning and Artificial Intelligence 8 (2020) 1–8. https://doi.org/10.14738/tmlai.81.7510.

[5] A. S. Wusu, O. A. Olabanjo, Nelder-mead algorithm in solving ordinary differential equations whose solutions possess singularities, Transactions on Networks and Communications 9 (2021) 11–17. https://doi.org/10.14738/tnc.91.9772.

[6] D. M. George, On the appliaction of genetic algorithms to differential equations, Romanian Journal of Economic Forecasting 3 (2) (2006) 5–9.

[7] E. Balagusuramy, Numerical Methods, McGraw Hill, New Delhi, 1999.

[8] J. Lambert, Computational Methods in ODEs, John Wiley & Sons, New York, 1973.

[9] N. E. Mastorakis, 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 Eolutionary Computing. Lisbon, Portugal., Springer Berlin Heidelberg, June 16–18, (2005), pp. 36–42.

[10] N. E. Mastorakis, 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), pp. 1–6.

[11] O. F. Bakre, A. S. Wusu, M. A. Akanbi, Solving ordinary differential equations with evolutionary algorithms, Open Journal of Optimization 4 (2015) 69–73.

[12] S. Fatunla, Non-linear multistep methods for initial value problems, An International Journal of Computers and Mathematics with Applications, 8 (3) (1982) 231–239.

[13] Wikipedia contributors, Differential evolution, Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Differential_evolution&oldid=1094959469 (accessed August 11, 2022).