# An Approximate Optimization Method for Solving Stiff Ordinary Differential Equations With Combinational Mutation Strategy of Differential Evolution Algorithm

**Werry Febrianti**✉, **Kuntjoro Adji Sidarto**, **Novriana Sumarti**✉

Industrial and Financial Mathematics Research Group, Faculty of Mathematics and Natural Sciences, Ganesha 10 Bandung, Indonesia

werry_febrianti@s.itb.ac.id✉, werry.febrianti@gmail.com✉, sidarto@itb.ac.id, novriana@itb.ac.id✉

**Abstract**

*This paper examines the implementation of simple combination mutation of differential evolution algorithm for solving stiff ordinary differential equations. We use the weighted residual method with a series expansion to approximate the solutions of stiff ordinary differential equations. We solve the problems from an ordinary stiff differential equation for linear and nonlinear problems. Then, we also implement our method for solving stiff systems of ordinary differential equations. We find that our algorithm can approximate the exact solution of a stiff ordinary differential equation with the smallest error for each length of series that we have chosen. Thus, this approximation method, by using the optimization method of simple combination differential evolution, can be a good tool for solving stiff ordinary differential equations.*

**Keywords:** *Stiff ordinary differential equation, Fourier series, Weighted residual method, Simple combination mutation of differential evolution.*

**Received:** 19 October 2022
**Accepted:** 02 December 2022
**Online:** 07 December 2022
**Published:** 20 December 2022

## 1 Introduction

Stiff differential equations are characterized as those whose exact solution has a term of the form $e^{-ct}$, where $c$ is a large positive constant [2, 15, 5]. There are many papers that proposed solutions for the stiff differential equation [12]. Atay [3] has used the Variational Iteration Method for solving Stiff Systems of Ordinary Differential Equations. Then, approximates of first-order Stiff solution based on block methods were shown in Ukpebor [22] using polynomial expansion and Adee [1] using Taylor series expansion with both papers at every step of integration by the Runge-Kutta method. Then, Raymond et. al [20] also used a self-starting five-step eight-order block method with two off-grid for solving stiff ordinary differential equations using interpolation and collocation procedures. Ibrahim et. al [14] employed a class of hybrid block backward differentiation formulas (HBBDFs) methods that possessed stability constructed by reformulating the block backward differentiation formulas (BBDFs) for the numerical solution of stiff ordinary differential equations (ODEs). El-Zahar et. al [9] proposed a new generalized Taylor- like explicit method for stiff ordinary differential equations. Aksah et. al [15] proposed a single diagonally implicit block backward differentiation formula (SDIBBDF) for solving stiff ordinary differential equations. Nasaruddin et. al [17] proposed a six-order fully implicit block backward differentiation formula with two off-step points (BBDF(6)), for the integration of first-order ODEs that exhibit stiffness. Ibrahim et.al [13] constructed an implicit fixed coefficient block backward differentiation formula denoted as A($\alpha$)-BBDF with equal intervals for solving stiff ODEs. So far, we did not find an optimization method for solving Stiff ODEs. Therefore, we propose a new method for solving stiff ODEs using an optimization method.

In this paper, we propose a new numerical method for solving stiff ordinary differential equations, especially in initial value problems. Our method is different from the numerical methods that have been stated before. We use an optimization method for solving the stiff ordinary differential equation. Then, we get solutions for all intervals simultaneously. This is the difference between our method with the numerical method as usual like using the Runge-Kutta and Euler method whether the solutions are getting for each node, not for all of interval and the result is not convergent numerically. Therefore, we propose a new method for the approximate solution of the stiff ordinary differential equation.

In this paper, we use the optimization method like in papers [4, 8, 10, 11, 18] for solving the stiff ordinary differential equation. We use a series expansion and weighted residual method for the approximation method in solving the stiff ordinary differential equation. This method can be used in solving the stiff ordinary differential equation and non-stiff ordinary differential equation. The series expansion and its derivative are implemented into ODEs and its boundary conditions. From here, we define a residual function with a unit weight function as in [4, 10, 11]. In our optimization problem, this function will be minimized using the Combinational Mutation Strategy of the Differential Evolution algorithm. As a preliminary study, we focus on solving linear odes both stiff and non-stiff.

This paper consists of five sections. The first one is an introduction. The second one explains the approximation method for solving stiff ODE and metaheuristic algorithm that is called the Combinational Mutation Strategy of Differential Evolution. The third one shows the results of the approximation solutions and compares them to the exact solutions. The fourth one is a discussion of the results. The last part gives the conclusion of this paper.

## 2 Approximation Method for Solving Stiff Ordinary Differential Equations

In this section, we explain the method for changing stiff ordinary differential equations into an optimization problem. Afterward, we give an overview of the optimization method that we used in this paper. However, before that, we describe the approximation function that we propose to approximate stiff ordinary differential equations.

### 2.1 Weighted Residual Function

Given an ordinary differential equation,

$$Lu = f(x) \tag{1}$$

with boundary conditions (BCs) in the domain $\Omega$. In this equation, $L$ is the linear differential operator upon $u$ and $f$ is the function value. The solution $u(x)$ of Eq. (1) approximately consists of

$$u(x) \approx \hat{u}(x) = \sum_{i=0}^{n} c_i \phi_i(x) \tag{2}$$

satisfying the BCs in the weighted residual function. Afterward, we define $c_i$ as the unknown coefficients yet to be determined for the trial functions $\phi_i(x)$ for $i = 0, 1, 2, \ldots, n$, where $\phi_i(x)$ are linearly independent to each other. The assumed solution is substituted in the governing differential equation (1) resulting in an error or residual. This residue is then minimized to vanish in the domain $\Omega$, resulting in a system of algebraic equations in terms of unknown coefficients $c_i$.

Further, we substitute $\hat{u}(x)$ to $L\hat{u}(x) \neq 0$ that is called an error. Then, we define the residual as the measure of error as stated below

$$R(x) = L\hat{u} - f(x) \tag{3}$$

An arbitrary weight function $w_i(x)$ is then multiplied in Eq. (3) and integrated over $\Omega$, that is considered as weighted residual function (WRF), the result like in Eq. (4) below:

$$\begin{aligned} \text{WRF} &= \int_\Omega w_i(x)[L\hat{u}(x) - f(x)]dx \\ &= \int_\Omega w_i(x)R(x)dx \end{aligned} \tag{4}$$

for $i = 0, 1, 2, \ldots, n$. Appropriating the value of $c_i$ on trial $\hat{u}(x)$ can make WRF zero over the entire domain $\Omega$. It is proper calling that in Eq. (2), $u(x) = \hat{u}(x)$ as $n \to \infty$.

### 2.2 A Series of Trial Solution of ODE

Based on Taylor explanation, the higher exponent can be approximated using sinus-cosines series with long enough series to get the highest accuracy such that using a Fourier series approximation like in reference [4, 10, 11]. But if this Fourier series approximation is used in solving stiff ordinary differential equations, then this Fourier series approximation is not effective to approximate the solution. The reason is that we need a longer term of Fourier series approximation to approximate the exponent with a higher power. Therefore, we propose an approximate function like the series below:

$$y_{\text{trial}}(x) = a_0 e^{cx} + \sum_{i=1}^{\text{nat}} [a_j \cos(j\pi x) + b_j \sin(j\pi x)] \tag{5}$$

Coefficients $c, a_0, a_1, b_1, a_2, b_2, \ldots, a_{\text{nat}}, b_{\text{nat}}$ are searched using the optimization method. Periodic factors in sinus and cosine function are eliminated to make generalize for the optimization result in all of the interval $x$. In this paper, a trial solution is not only used in stiff ordinary differential equations, but also for non-stiff ordinary differential equations.

### 2.3 Weighted Residual Method as Optimization Problem

An ordinary differential equation (ODE) is an equation involving an unknown function and its derivatives. An ODE of order $n$ in the explicit form is

$$f(x; y, y', \ldots, y^{(n)}) = 0 \tag{6}$$

where $y$ is a function of $x$, $y' = du/dx$ is the first derivative with respect to $x$, and $y^{(n)} = (d^{(n)}y)/(d^{(n)}x)$ is the $n$th derivative with respect to $x$. Not all of differential equations can be solved analytically. A residual function is defined by substituting the trial solution (5) and its derivatives to the left-hand side of ODE in Eq. (6):

$$R(x) = f(x; y_{\text{trial}}, y'_{\text{trial}}, \ldots, y^{(n)}_{\text{trial}}) \tag{7}$$

According to [5, 11, 12], the objective function to evaluate ODE's cost function is a weighted residual function as follows:

$$\text{WRF} = \int_\Omega |w(x)| \cdot |R(x)| dx \tag{8}$$

where $w(x)$ and $R(x)$ are denoting weight and residual functions respectively. In this paper, for simplicity purposes, the weight function can be assumed to be unit value, $w(x) = 1$ as in [4, 10, 11].

The approximation solution must also meet the initial and boundary condition of ODE problems. They are given as follows:

$$\begin{aligned} y(x_0) = y_0 &\Rightarrow g_1(x_0) = |y_{\text{trial}}(x_0) - y_0| \\ y'(x_0) = y'_0 &\Rightarrow g_2(x_0) = |y'_{\text{trial}}(x_0) - y'_0| \\ &\vdots \\ y^{(n)}(x_0) = y_0^{(n)} &\Rightarrow g_n(x_0) = |y^{(n)}_{\text{trial}}(x_0) - y_0^{(n)}| \end{aligned} \tag{9}$$

Then, a penalty function is needed to impose the constraints. In this paper, we use the penalty function as suggested in [5, 11, 12] which is

$$\text{PVF} = \sum_{k=1}^{n_{\text{IVs}}+n_{\text{BVs}}} \mu_k g_k \tag{10}$$

in which $g_k$ is the violation of $k$th constraint which is computed from Eq. (10); $n_{\text{IVs}}$ and $n_{\text{BVs}}$ are number of the initial conditions and boundary conditions, $\mu_k$ is penalty multiplier which is chosen to be a large number.

Finally, the optimization model of ODE with boundary values (BVs) and initial values (IVs) in an unconstrained function, will use the fitness function as follows

$$\text{FFV} = \text{WRF} + \text{PFV} \tag{11}$$

The optimum value of FFV is achieved when the value of this function approaches zero. In this condition, more accurate solutions can be obtained. The procedures for solving approximation method are stated below:

1. Define nat of the trial solution of (1);
2. Construct FFV using in (8), (10) and (11);
3. Initialize the parameters for the optimization algorithm to find the coefficients of the trial solution (5). Then, the penalty function is used to handle the constraints;
4. Evaluate the weighted residual function (8) where the integral can be calculated numerically by using the Simpson formula. Simultaneously, calculate the penalty value for violated conditions due to an inappropriate value of the coefficient at the trial solution from (10) to obtain fitness function value (11);
5. Do step 4 until the stopping criteria of the optimization algorithm are reached.

### 2.4 Differential Evolution Algorithm

DE is one of the global optimization methods that was first introduced by Storn and Price in 1995 [19]. The method is initializing population as the first step in the searching method. Then the DE operators (mutation, crossover, and selection respectively) are iteratively evaluated to improve the population to get an optimum result. DE has various schemes describe as DE/x/y/z where x is the vector that has chosen to be mutated, y is the number of different pair vectors that is used in the mutation operation, and z is the type of crossover.

The "DE/rand/1/bin" classical mutation strategy in a G generation of populations has formula:

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) \tag{12}$$

There are three different vectors, where $x_{r_1,G}$ is a base vector, and vectors $x_{r_2,G}, x_{r_3,G}$ are used for its difference. All vectors are randomly chosen from the populations, stated by "rand". The scale factor $F$, a scale factor in mutation, is a constant that is usually taken between 0 and 1. "bin" stands for binomial crossover, and "1" stands for a number of pair of different random vectors in Eq. (12). Moreover, there are the other schemes frequently used forms in mutation strategies, as follows:

DE/best/1/bin:

$$v_{i,G+1} = x_{\text{best},G} + F(x_{r_1,G} - x_{r_2,G}) \tag{13}$$

DE/current-to-best/1/bin:

$$v_{i,G+1} = x_{i,G} + F(x_{\text{best},G} - x_{i,G}) + F(x_{r_1,G} - x_{r_2,G}) \tag{14}$$

DE/current-to-rand/1/bin):

$$v_{i,G+1} = x_{i,G} + F(x_{i,G} - x_{r_3,G}) + F(x_{r_1,G} - x_{r_2,G}) \tag{15}$$

DE/best/2/bin):

$$v_{i,G+1} = x_{\text{best},G} + F(x_{r_1,G} - x_{r_2,G}) + F(x_{r_3,G} - x_{r_4,G}) \tag{16}$$

DE/rand/2/bin):

$$v_{i,G+1} = x_{r_1,G} + F(x_{r_2,G} - x_{r_3,G}) + F(x_{r_4,G} - x_{r_5,G}) \tag{17}$$

Each mutation strategy can be effective and appropriate for a special case, thus in this paper, we will use more than one mutation strategy. The mutation strategies that we used are DE/rand/1, DE/best/1, DE/current-to-best/1, dan DE/current-to-rand/1. Because there is a combination in mutation strategy in DE algorithm, the algorithm we called a combinational mutation strategy of differential evolution algorithm (CmDE).

In the crossover operation, a target vector ($x_{i,G}$) can be potentially directed by a mutant vector ($v_{i,G+1}$) so it becomes a trial vector ($u_{i,G+1}$). This vector has a chance to be accepted as the new target with probability Crossover $Cr$ whose binomial uniform formula as follows with the probability of crossover ($Cr$) is chosen from the range between 0 and 1.

$$u_{i,G+1} = \begin{cases} v^j_{i,G+1} & \text{rand} < Cr \text{ or } j_{\text{rand}} = j \\ x_{i,G} & \text{otherwise} \end{cases} \tag{18}$$

The selection operator will compare the fitness function value of the trial vector to the value of target vector. If fitness function value of a trial vector is lower than a target vector, then a trial vector is added as a new generation of the population, and vice versa.

In this paper, the iteration computation will be terminated when the maximum iteration is reached or:

$$|\text{fitness} - \text{old fitness}| < \epsilon \tag{19}$$

**Algorithm 1** Combinational mutation strategy of differential evolution algorithm

Determining of parameters for DE
Initialization Generate the initial population
Assess the fitness for each individual
**while** Termination condition is not satisfied **do**
  **Mutation**
  Set $F = 0.5$ then find new population for each mutation strategy
  **Crossover**
  for rand $< Cr$ do crossover process for each mutation strategy
  **Evaluate** the boundary constraints for each new individual
  **Particular Selection**
  Find the best solution for each mutation strategy
**end while**
**Global Selection**
Find the best global solution for each mutation strategy
Output Global optimum solution

Table 1: Test problems of an ordinary differential equation.

| Problem | Equation | Source | Domain | Exact Solution |
|---|---|---|---|---|
| ODE1 | $y' + x = 0;\ y(0) = 1;\ y(1) = 1/2$ | [16] | [0,1] | $y(x) = 1 - x^2/2$ |
| ODE2 | $y' + 0.5y - e^{0.8x} = 0;\ y(0) = 2$ | [19] | [0,1] | $y(x) = \frac{40}{13}e^{0.8x} - \frac{14}{13}e^{-0.5x}$ |
| ODE3 | $y' - (1 + 2x)y^{1/2} = 0;\ y(0) = 1$ | [19] | [0,1] | $y(x) = \frac{1}{4}(2 + x + x^2)^2$ |
| ODE4 | $y' + 100y - 99e^{2x} = 0;\ y(0) = 0$ | [6, 20] | [0,1] | $y(x) = \frac{33}{34}(e^{2x} - e^{-100x})$ |
| ODE5 | $y' - 5(y - x)^2e^{5x} - 1 = 0;\ y(0) = 1$ | [1] | [0,1] | $y(x) = x - e^{-5x}$ |
| ODE6 | $y' + y^3/2 = 0;\ y(0) = 1$ | [8, 21] | [0,1] | $y(x) = 1/\sqrt{1 + x}$ |

Table 2: Test problems of a system ordinary differential equations.

| Problem | Equation | Source | Domain | Exact Solution |
|---|---|---|---|---|
| ODE7 | $y_1' - 9y_1 - 24y_2 - 5\cos(x) + \frac{1}{3}sin(x) = 0;\ y_1(0) = \frac{4}{3}$ <br> $y_2' + 24y_1 + 51y_2 + 9\cos x - \frac{1}{3}\sin x = 0;\ y_2(0) = \frac{2}{3}$ | [1, 8] | [0,1] | $y_1(x) = \frac{1}{3}\cos x + 2e^{-3x} - e^{-39x}$ <br> $y_2(x) = -\frac{1}{3}\cos x - e^{-3x} + 2e^{-39x}$ |
| ODE8 | $y_1' - 32y_1 - 66y_2 - \frac{2}{3}x - \frac{2}{3} = 0;\ y_1(0) = \frac{1}{3}$ <br> $y_2' + 66y_1 + 133y_2 + \frac{1}{3}x + \frac{1}{3} = 0;\ y_2(0) = \frac{1}{3}$ | [1, 7] | [0,1] | $y_1(x) = \frac{2}{3}x + \frac{2}{3}e^{-x} - \frac{1}{3}e^{-100x}$ <br> $y_2(x) = -\frac{1}{3}x - \frac{1}{3}e^{-x} + \frac{2}{3}e^{-100x}$ |

where $\epsilon$ is a small number. Stopping criteria (19) have been used in paper [11, 12] and can be made our optimization algorithm stop in the minimum condition that can be reached. To validate how good the performance of the model, the Root of the Mean Squared Error (RMSE) and the Maximum Error (MAXE) are calculated using the numerical solution $y_{\text{trial}}$ and the exact solution $y$, respectively:

$$\text{RMSE} = \sqrt{\frac{1}{n}||\sum_{i=1}^{n}(y_{\text{trial}})_i(t) - y_i(t)||^2} \quad (20)$$

$$\text{MAXE} = \max||(y_{\text{trial}})_i(t) - y_i(t)|| \quad (21)$$

where $n$ is the total number of collocation points. Note that this error is not considered in the process of solving of ODEs that have no exact solutions so that the approximation result can be achieved from FFV.

## 3 Results of the Approximation Solutions

Here, we solve several IV problems that have analytical solutions using the CmDE algorithm and optimization model for solving Stiff ODE, in order to validate the capability and the accuracy of our method. The method is implemented to approximate various types of ODEs, like Stiff and non-Stiff. The algorithm uses population size of 200, and maximum number of iterations in these computations is 300-10000. In the first calculation, we implement our optimization method for solving stiff and non-stiff ODEs with variations of maximum number of iterations with constant nat value. In the second calculation, we various of nat from the Fourier-like Series and Eq. 15 or RMSE as stopping criteria. All computations are running with MATLAB R2018a in HP Pavilion Laptop Model 14-dv0067TX that is equipped with processor Intel Core TM i7 with 8 GB ram and 4.70 GHz running Windows 10. Several problems of ODEs, Stiff and non-Stiff, are given in Table 1.

The results of several problems of ODEs in Table 1 and Table 2 are given in Table 6, Table 4 and Table 8. In Table 3, we run CmDE algorithm with various maximum iteration for the same nat for each trial solutions from single ODE in Table 1. Then, in Table 4 and Table 5, we run CmDE algorithm again for one Stiff ordinary differential equation and system of Stiff ordinary differential equation with various nat and Equation (15) or RMSE as stopping criteria. The graph of each problem shows in Fig. 1 until Fig. 8.

Table 3: Result of Problems with variation of nat, variation of RMSE as stopping criteria and maximum iteration is 60000.

| Problem | nat of Trial Solution | Iteration of the best solution | RMSE as stopping criteria | RMSE | MAXE |
|---|---|---|---|---|---|
| ODE1 | 3 | 51 | 1e-04 | 5.823e-05 | 1.604e-04 |
| | 4 | 296 | 1e-05 | 7.993e-06 | 3.220e-05 |
| | 5 | 1340 | 1e-06 | 9.473e-07 | 3.894e-06 |
| ODE2 | 3 | 256 | 1e-04 | 7.304e-05 | 1.988e-04 |
| | 4 | 890 | 1e-05 | 9.085e-06 | 1.796e-05 |
| | 5 | 4138 | 1e-06 | 9.004e-07 | 3.456e-06 |
| ODE3 | 7 | 423 | 1e-04 | 6.386e-05 | 2.801e-04 |
| | 8 | 904 | 1e-05 | 8.188e-06 | 2.143e-05 |
| | 9 | 885 | 1e-06 | 6.782e-07 | 1.397e-06 |
| ODE4 | 8 | 14233 | 1e-04 | 9.437e-05 | 2.490e-04 |
| | 9 | 2230 | 1e-05 | 9.836e-06 | 2.718e-05 |
| | 10 | 2481 | 1e-06 | 9.738e-07 | 2.223e-06 |
| ODE5 | 6 | 456 | 1e-04 | 6.255e-05 | 1.559e-04 |
| | 7 | 657 | 1e-05 | 8.304e-06 | 2.115e-05 |
| | 8 | 785 | 1e-06 | 7.917e-07 | 2.491e-06 |
| ODE6 | 3 | 245 | 1e-04 | 8.281e-05 | 1.809e-04 |
| | 4 | 902 | 1e-05 | 8.230e-06 | 1.537e-05 |
| | 5 | 3590 | 1e-06 | 8.706e-07 | 2.830e-06 |

Table 4: Result of Problems with variation of nat and maximum iteration is 10000.

| Problem | nat of Trial Solution | Iteration of the best solution | Iteration as stoppin criteria | RMSE | MAXE |
|---|---|---|---|---|---|
| ODE1 | 3 | 801 | >800 | 1.12e-05 | 2.30e-05 |
| | 4 | 802 | >800 | 3.13e-07 | 5.93e-07 |
| | 5 | 803 | >800 | 1.46e-08 | 3.35e-08 |
| ODE2 | 5 | 801 | >800 | 3.36e-05 | 1.33e-04 |
| | 7 | 801 | >800 | 5.43e-07 | 1.85e-06 |
| | 9 | 2001 | >2000 | 1.91e-08 | 8.36e-08 |
| ODE3 | 3 | 801 | >800 | 1.49e-05 | 3.16e-05 |
| | 4 | 810 | >800 | 2.10e-07 | 4.91e-07 |
| | 5 | 1002 | >1000 | 3.17e-09 | 1.07e-08 |
| ODE4 | 4 | 802 | >800 | 8.25e-05 | 1.38e-04 |
| | 6 | 1005 | >1000 | 1.93e-06 | 2.55e-06 |
| | 7 | 2001 | >2000 | 6.89e-08 | 1.25e-07 |
| ODE5 | 3 | 1006 | >1000 | 2.17e-05 | 6.69e-05 |
| | 4 | 3019 | >3000 | 2.40e-06 | 9.24e-06 |
| | 5 | 5001 | >5000 | 1.50e-07 | 4.56e-07 |
| ODE6 | 5 | 1002 | >1000 | 1.15e-03 | 7.87e-03 |
| | 7 | 2006 | >2000 | 9.24e-04 | 6.41e-03 |
| | 10 | 5006 | >5000 | 6.17e-04 | 3.51e-03 |

Table 5: Result of Problems of Stiff System using CmDE with variation of nat and maximum iteration is 10000.

| Problem | nat of Trial Solution | Iteration of best fitness | Time average (s) | Equation | RMSE | MAXE |
|---|---|---|---|---|---|---|
| ODE7 | 3 | 1001 | 248.69 | Equation1 | 4.63e-04 | 7.97e-04 |
| | | | | Equation2 | 4.63e-04 | 4.33e-04 |
| | 4 | 2003 | 425.16 | Equation1 | 1.36e-05 | 3.53e-05 |
| | | | | Equation2 | 1.36e-05 | 1.60e-05 |
| | 5 | 3001 | 675.38 | Equation1 | 1.54e-06 | 3.30e-06 |
| | | | | Equation2 | 1.54e-06 | 1.53e-06 |
| ODE8 | 3 | 802 | 180.86 | Equation1 | 3.31e-05 | 6.42e-05 |
| | | | | Equation2 | 3.31e-05 | 3.17e-05 |
| | 4 | 2001 | 466.73 | Equation1 | 5.12e-07 | 9.33e-07 |
| | | | | Equation2 | 5.12e-07 | 4.60e-07 |

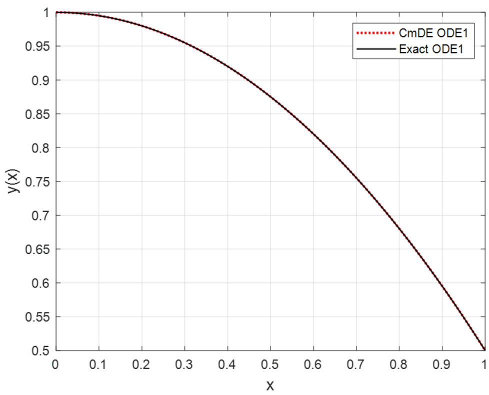


Figure 1: Graph of exact and approximate solution with nat = 5 and RMSE = 9.473e-07 of ODE1

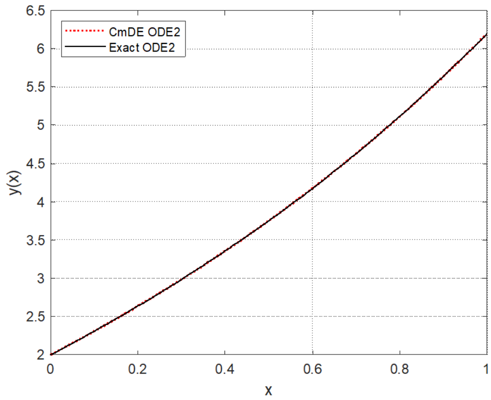


Figure 2: Graph of exact and approximate solution with nat = 5 and RMSE = 9.004e-07 of ODE2

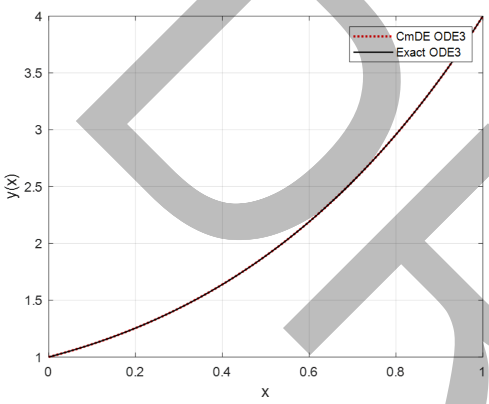


Figure 3: Graph of exact and approximate solution with nat = 9 and RMSE = 6.782e-07 of ODE3

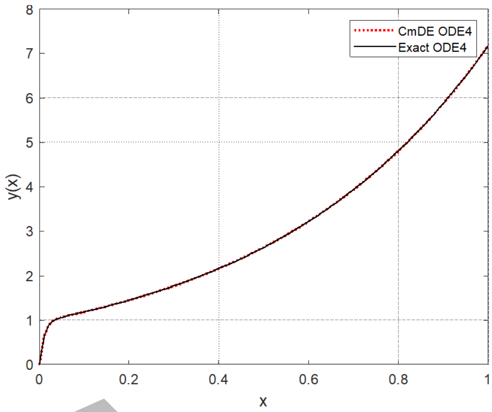


Figure 4: Graph of exact and approximate solution with nat = 7 and RMSE = 9.738e-07 of ODE4

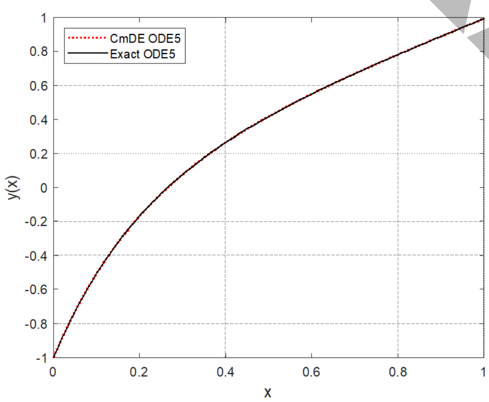


Figure 5: Graph of exact and approximate solution with nat = 8 and RMSE = 7.917e-07 of ODE5

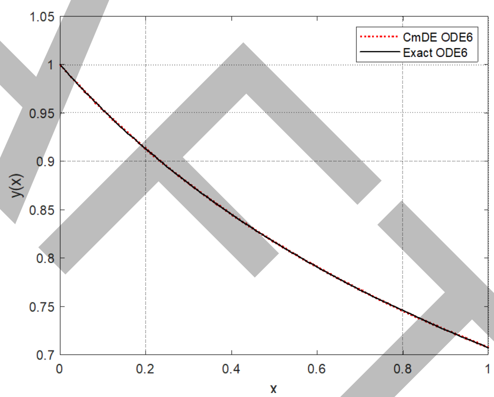


Figure 6: Graph of exact and approximate solution with nat = 5 and RMSE = 8.706e-07 of ODE6

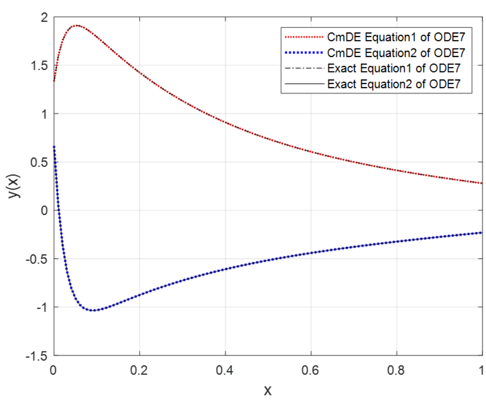


Figure 7: Graph of stiff system of exact and approximate solution with nat = 5 and RMSE1 = RMSE2 = 1.54e-06

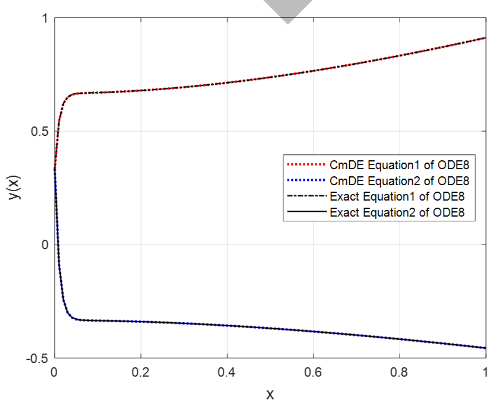


Figure 8: Graph of stiff system of exact and approximate solution with nat = 4 and RMSE1 = RMSE2 = 5.12e-07

## 4 Discussion

In this research, we obtain numerical solutions for examples of ordinary differential equation especially stiff ordinary differential equation like in Table 1. These ODE problems are changed into an optimization problem using the concepts that we have explain in section 2. Then, optimization from these problems or the solutions for this optimization problems are found using CmDE algorithm. The output for this optimization problem is the coefficients of trial solutions from these problems. In order to analyse accuracy of our methods, in comparison with exact solutions, the RMSE and MAXE is calculated for that ODEs.

In searching the coefficient of trial solution using CmDE algorithm, the setting of maximum iteration be one of factor that can affects the accurate value of the trial solution. From Table 2, we have three different maximum iteration for solving optimization problems with CmDE. We show that if the iteration value become bigger, such that the value of RMSE become smaller. The results show that CmDE algorithm can find approximate solution which can near with the exact solution when the maximum iteration become bigger. Furthermore, a trial solution that we propose can be an approximate solution from ODE that we have shown in Table 1.

Adding iteration in a nat value like in Table 3 can achieve condition where the RMSE value stagnant in certain value. This is caused by the long series (nat) only able to approximate the exact solution of the ordinary differential equation in that nat value. Then, if we want to get the better accuracy, then the nat value has to be increased. In Table 3 and 4, we show the comparison of optimization result for different nat or different of a series sum. Then, to see the accuracy that can be achieved from a nat value, we add Eq. (15) as one of stopping criteria. The result of Table 3 and 4 show that we can increase the accuracy by increasing the value of nat. We can see it from the RMSE value for each problem. The graph of each problem shows in Fig. 1 until Fig. 8. The boundary of independent variable $t$ or $x$ can be extended to the bigger space. Trial solution of each ODE in Table 1 can also apply to larger $x$ limit.

This optimization method can be extended for solving differential equation that does not have an exact solution. The accuracy of the solution of differential equations can be seen from the fitness value. As we purpose in Table 3, the fitness value linearly proportion to the RMSE value. Therefore, for differential equation that does not have the exact value, the fitness value can be used as benchmark from accuracy of the approximate solution. Thus, when the fitness value become smaller, then the approximate solution can nearly approximate the solution of that differential equation.

## 5 Conclusion

In solving differential equations, we build the approximate solution in a series as trial solution. This trial solution can be used in stiff ordinary differential equation and non-stiff ordinary differential equation as a base approximated function such that solving ODE problem can be transformed into an optimization problem. The aim is to minimize the weighted residual function, which is the error obtained from the implementation of the series into the differential equations. Boundary and initial conditions are imposed as constraints that are implemented as the penalty in the objective function.

We use Combinational mutation strategy of Differential Evolution (CmDE) algorithm as a tool to minimize the residual function. This CmDE algorithm is successfully giving the most minimum results for weighted residual function in the trial solution of each ODE. Thus, metaheuristic algorithms like CmDE algorithm can be applied to approximate solutions of many differential equation problems. This algorithm will give robust tools in a simple way for approximating the complex linear ordinary differential equations. Therefore, we motivate to build a general approximate solution to approximate the nonlinear ODEs that can apply to Stiff differential equation and non-Stiff differential equation.

**Acknowledgement:** This research was funded by Indonesian Minister of Research and Technology/National Research and Innovation Agency, Indonesia, Grant Number 2/E1/KP.PTNBH/2021.

## References

[1] Adee, S., and Atabo, V. Improved two-point block backward differentiation formulae for solving first order stiff initial value problems of ordinary differential equations. *NIGERIAN ANNALS OF PURE AND APPLIED SCIENCES 3*, 2 (2020), 200–209.

[2] Ahmad, R., Yaacob, N., and Mohd Murid, A.-H. Explicit methods in solving stiff ordinary differential equations. *International Journal of Computer Mathematics 81*, 11 (2004), 1407–1415.

[3] Atay, M. T., and Kilic, O. The semianalytical solutions for stiff systems of ordinary differential equations by using variational iteration method and modified variational iteration method with comparison to exact solutions. *Mathematical Problems in Engineering 2013* (2013).

[4] Babaei, M. A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization. *Applied Soft Computing 13*, 7 (2013), 3354–3365.

[5] Burden, R. L., Faires, J. D., and Burden, A. M. *Numerical analysis*. Cengage learning, 2015.

[6] Cao, H., Kang, L., Chen, Y., and Yu, J. Evolutionary modeling of systems of ordinary differential equations with genetic programming. *Genetic Programming and Evolvable Machines 1*, 4 (2000), 309–337.

[7] Chapra, S. C., Canale, R. P., et al. *Numerical methods for engineers*, vol. 1221. Mcgraw-hill New York, 2011.

[8] Chaquet, J. M., and Carmona, E. J. Solving differential equations with fourier series and evolution strategies. *Applied Soft Computing 12*, 9 (2012), 3051–3062.

[9] El-Zahar, E. R., Tenreiro Machado, J., and Ebaid, A. A new generalized taylor-like explicit method for stiff ordinary differential equations. *Mathematics 7*, 12 (2019), 1154.

[10] Febrianti, W., Sidarto, K. A., and Sumarti, N. Solving some ordinary differential equations numerically using differential evolution algorithm with a simple adaptive mutation scheme. In *AIP Conference Proceedings* (2021), vol. 2329, AIP Publishing LLC, p. 040003.

[11] Febrianti, W., Sidarto, K. A., and Sumarti, N. Solving systems of ordinary differential equations using differential evolution algorithm with the best base vector of mutation scheme. In *AIP Conference Proceedings* (2021), vol. 2423, AIP Publishing LLC, p. 020006.

[12] Gear, C. W. Numerical initial value problems in ordinary differential equations. *Prentice-Hall series in automatic computation* (1971).

[13] Ibrahim, Z. B., Mohd Noor, N., and Othman, K. I. Fixed coefficient a ($\alpha$) stable block backward differentiation formulas for stiff ordinary differential equations. *Symmetry 11*, 7 (2019), 846.

[14] Ibrahim, Z. B., and Nasarudin, A. A. A class of hybrid multistep block methods with a–stability for the numerical solution of stiff ordinary differential equations. *Mathematics 8*, 6 (2020), 914.

[15] Jana Aksah, S., Ibrahim, Z. B., and Mohd Zawawi, I. S. Stability analysis of singly diagonally implicit block backward differentiation formulas for stiff ordinary differential equations. *Mathematics 7*, 2 (2019), 211.

[16] Musa, H., Suleiman, M., and Senu, N. Fully implicit 3-point block extended backward differentiation formula for stiff initial value problems. *Applied Mathematical Sciences 6*, 85 (2012), 4211–4228.

[17] Nasarudin, A. A., Ibrahim, Z. B., and Rosali, H. On the integration of stiff odes using block backward differentiation formulas of order six. *Symmetry 12*, 6 (2020), 952.

[18] Panagant, N., and Bureerat, S. Solving partial differential equations using a new differential evolution algorithm. *Mathematical Problems in Engineering 2014* (2014).

[19] Price, K., Storn, R. M., and Lampinen, J. A. *Differential evolution: a practical approach to global optimization.* Springer Science & Business Media, 2006.

[20] Raymond, D., Donald, J., Michael, A., and Ajileye, G. A self-starting five-step eight-order block method for stiff ordinary differential equations. *Journal of Advances in Mathematics and Computer Science 26*, 4 (2018), 1–9.

[21] Reich, C. Simulation of imprecise ordinary differential equations using evolutionary algorithms. In *Proceedings of the 2000 ACM symposium on Applied computing-Volume 1* (2000), pp. 428–432.

[22] Ukpebor, L., and Omole, E. Three-step optimized block backward differentiation formulae (tobbdf) for solving stiff ordinary differential equations. *African Journal of Mathematics and Computer Science Research 13*, 1 (2020), 51–57.