Comparative Study of Matlab ODE Solvers for the Korakianitis and Shi Model

Emagbetere E.1*, Oluwole O.2, Salau T.A.O.2

1Mechanical Engineering Department, Federal University of Petroleum Resources, Effurun, Delta State, Nigeria
2Mechanical Engineering Department, University of Ibadan, Ibadan, Nigeria

Email: emagbetere.eyere@fupre.edu.ng

Keywords: Ordinary differential equation, numerical solvers, mathematical model, MATLAB ODE suite

Abstract. Changing parameters of the Korakianitis and Shi heart valve model over a cardiac cycle has led to the investigation of appropriate numerical technique(s) for good speed and accuracy. Two sets of parameters were selected for the numerical test. For the seven MATLAB ODE solvers, the computed results, computational cost and execution time were observed for varied error tolerance and initial time steps. The results were evaluated with descriptive statistics; the Pearson correlation and ANOVA at \( \alpha_{0.05} \). The dependence of the computed result, accuracy of the method, computational cost and execution time of all the solvers, on relative tolerance and initial time steps were ascertained. Our findings provide important information that can be useful for selecting a MATLAB ODE solver suitable for differential equation with time varying parameters and changing stiffness properties.

1.0 Introduction

The Korakianitis and Shi (KS) heart valve model is a zero-dimensional, second order, nonlinear, Ordinary Differential Equation (ODE) that describes the dynamics of the human heart valve. The equation is embedded in the overall model of other parts of the entire cardiovascular system [1]. The equation may change from being non-stiff to stiff and vice versa, as its parameter changes within a heart cycle. An equation is said to be stiff if certain numerical schemes that are not absolutely stable cannot be used to seek an approximate solution to the problem [2]. Consequently, the conventional fourth order explicit Runge-Kutta method used to find a solution to the KS heart valve model in previous studies [1, 3-4] may not give accurate results for large step size, while smaller steps size may increase the round-off errors and increase computation time.

On cannot exhaust the list of numerical techniques and their suitability for solving ODE as older ones are being modified and newer ones are developed. Butcher [5] reported a good number of these solvers developed over the 20th century. Evaluations of ODE solvers have a long history [6-9] and earlier intentions were either to ascertain their effectiveness [6-7] or to categorize them as either stiff or non-stiff solvers [10-13]. Some researchers, however, reported the performance of the solving environment for some of these ODE solvers [14-16]. Very recently, focus has been to investigate ODE solvers for specific problem sets [17-20].

The MATLAB software is widely used in engineering studies. Its ODE suite has seven powerful solvers, three of which are explicit schemes while the other four are implicit schemes. The ode45, ode23 and ode113 are the three explicit schemes while ode23t, ode23s, ode15s and ode23tb are the four implicit solvers. Interested readers are referred to the literatures [21-23] for mathematical details of the different solvers in the MATLAB ODE suite.

Runge-Kutta methods are widely discussed in the literature [24-27]. The ode23 and ode45 solvers implement “Runge-Kutta formula of order 2 and 3” [21] and the “Dormand-Prince (4,5) pair” [28], respectively. The Dormand-prince (4,5) pair formula, which is an improved version of Fehlberg (4,5) pair [29], gives better results when compared with other Runge-Kutta schemes [30-31]. The ode113 solver implements the Adams’ formulas of variable order in a Predict, Evaluate,
and Correct, Evaluate (PECE) method [21]. Adams’ formulas have successfully been used to solve many differential equations [32]. Spijker investigated their stability and convergence property [2] while their effectiveness for treating stiff problems was reported by Lavernano and Mazzia [33].

The ode23t solver implements the trapezoidal rule. The trapezoidal rule is a one-step low order but absolutely stable technique [26]. It is the simplest implicit technique for solving ODE [34] but has a low efficiency [22]. The ode23s solver is a more efficient one step solver in MATLAB ODE suite. It implements the modified Rosenbrock techniques. The general form of Rosenbrock techniques is discussed in the literature [35]. The modified versions [9,36] have improved stability property. A very reliable solver for stiff problems is the Backward Differential Formulas (BDFs). Their development and implementation are discussed [2]. The numerical differential formulas (NDFs) are a modification of the BDFs [23]. Being modified versions of BDFs, NDFs give better results but cannot be implemented beyond order six as they are unstable at higher order [37]. The MATLAB ode15s solver implements the NDF and BDF alternatively, while the ode23tb implements the trapezoidal rule as a first stage and then the BDF formula as a second.

Omale et al. reported the solution to six sets of standards equation with the MATLAB ODE solvers [38]. Abelman and Patidar [39] compared the numerical results of three MATLAB ODE solvers (ode45, ode15s and ode23s) with that of the nonstandard finite difference method (NSFDM). They showed that the NSFDM can be used to obtain good result without step size restrictions. Yatim et al [40] compared results they obtained using a newly developed backward Differential formula with those obtained using ode15s and ode23s for three sets of problems. Ocherova [41] modified the ode23 solver and showed that the relative tolerance has effect on the solution, evaluation time, computation cost and accuracy of the solver. Dallas and Machairas [20] investigated the MATLAB solvers and their parameters for a class of nonlinear hybrid system. He attempted to determine which of the solver was best in terms of speed and accuracy for such type of problems.

The aim of this paper is to evaluate the effect of tolerated error and initial time step on the accuracy, execution time and computational cost of the seven MATLAB ODE solvers for an equation with time varying parameters that may become stiff at certain parameter levels. The other sections of this paper are organized as follows: Section two contains a brief description of the KS model and the selected parameters and variables for simulation. Section three of the paper presents the analysis and discussion of results. The paper ends with conclusion and remarks based on the findings.

### 2.0 Methodology

#### 2.1 The KS Heart Valve Model

The KS heart valve model [3] that describes the dynamics of the heart valve is shown in equation (1) below:

\[
\theta'' = (P_1 - P_2)K_p \cos(\theta) - K_f \theta' + K_b Q \cos(\theta) - K_v Q \sin(\theta)
\]  

\(\theta\) represents the angular displacement of a heart valve. \(P_1\) is the pressure of the chamber or vessel from which blood is flowing through the valve while \(P_2\) is the heart chamber or blood vessel which blood flows into from the valve. Consequently, \((P_1 - P_2)\) is the pressure gradient across the valve. \(Q\) is the flow rate through the valve. \(K_p, K_f, K_b\) and \(K_v\) are time invariant parameters representing the pressure, friction, blood flow and vortex constant respectively [3]. Their values for the aortic and mitral valve respectively are given in Table 1:


| Parameter | Value | Unit          |
|-----------|-------|--------------|
| $K_{p,ao}$ | 5500  | rad/(s^2 mmHg) |
| $K_{f,ao}$ | 50    | 1/s          |
| $K_{b,ao}$ | 2     | rad/(s m)    |
| $K_{v,ao}$ | 7     | rad/(s m)    |
| $K_{p,mi}$ | 5500  | rad/(s^2 mmHg) |
| $K_{f,mi}$ | 50    | 1/s          |
| $K_{b,mi}$ | 2     | rad/(s m)    |
| $K_{v,mi}$ | 3.5   | rad/(s m)    |

Source: [3]

Equation (2) is a simplified form of equation (1) above

$$\theta'' = A \cos(\theta) - B \sin(\theta) - C \theta'$$  \hspace{1cm} (2)

This can be represented with a system of two 1st order ordinary differential equations as follows:

$$\theta_1' = \theta_2$$  \hspace{1cm} (2a)

$$\theta_2' = A \cos(\theta_1) - B \sin(\theta_1) - C(\theta_2)$$  \hspace{1cm} (2b)

### 2.2 Model Parameters for Simulation

Prior simulation was used to observe the dynamical behavior of the model on a time series plot. After then, two sets of parameter values of $A$, $B$ and $C$ that yielded differing stability property were selected for this study. They are presented as follows:

**Case 1:** $A=118771.6264$; $B=1250.4881$; $C=50$;  
**Case 2:** $A=380.5941$; $B=19.3363$; $C=50$;

### 2.3 Numerical Experiments

For the numerical experiments, the valve was assumed to be opening from its rest position, so initial values of $(0, 0)$ was used for $\theta_1$ and $\theta_2$. An average time over which the semilunar valves open is 0.3 Seconds [42-43]. Thus, selected iteration time for the simulation is 0.3 Seconds. A MATLAB function file for equations 2a and 2b was written for Simulations purpose. While varying the relative tolerance and initial time steps of the schemes, the following were observed for the seven (7) MATLAB ODE solvers:

1. The calculated results for angular displacements.
2. The computational cost which is equivalent to the number of times the integrand function was evaluated.
3. The total number of failed and number of successful steps.
4. The execution time which is equivalent to the total time taken to generate the result with a particular solver.

The selected relative tolerance (Rel. Tol.) used for the simulations are: 1E-2, 1E-4, 1E-6, 1E-8, 1E-10, 1E-12, and 1E-14

The selected initial time steps used for the simulations are: 0.001, 0.005, 0.01, 0.05, and 0.1.

The results were then evaluated with descriptive statistics, Pearson correlation and the two-way ANOVA in Microsoft excel version 2010.
3.0 Results

Fig. 1 shows the time series plot of angular displacement for the KS model over an iteration time of 0.3 Seconds for both cases of parameter set. As observed, the solution is a decaying sinusoid thatsettles to about 89.3978 degrees for case 1 while the solution decays exponentially and settles to about 73.3152 degrees for case 2. The change in dynamics of the system is due to the change of stability properties that occurs as parameter changes. As observed in Table 2, the calculated values of angular displacement for a very low relative tolerance was unaffected by the type of MATLAB solver as the value of $P=0.3846$ is greater than 0.05. The values of the angular displacements for the two different sets of parameter (Case 1 and Case 2) vary significantly as $P=1.71E-15$ is far less than 0.05.

![Image](image_url)

**Figure 1.** Time series of angular displacement in degrees for case 1 and case 2. Generated with initial time step of 0.003 and relative tolerance of 10E-12.

**Table 2.** ANOVA table for values of calculated angular displacements for varied parameters (Columns 1 and 2) and different solvers (Rows 1-7).

| Source of Variation       | SS     | df | MS      | F       | P-value | F crit  |
|---------------------------|--------|----|---------|---------|---------|---------|
| Rows (Solvers)            | 0.009673 | 6  | 0.001612 | 1.283852| 0.384676| 4.283866|
| Columns (Parameter set)   | 427.3755 | 1  | 427.3755 | 340330.2| 1.71E-15| 5.987378|
| Error                     | 0.007535 | 6  | 0.001256 |         |         |         |
| Total                     | 427.3928 | 13 |         |         |         |         |

The average angular displacements, standard deviation and their correlation with relative tolerance, for the different solvers, are shown in Table 3. For all the solvers, there is a correlation between the calculated angular displacements and the relative tolerance. However, the correlation coefficient changes from negative to positive or vice versa for the two cases of selected parameters, for all solvers except ode113. The values of angular displacement for ode45 and ode23tb solvers were least affected by the tolerated errors, as they have the least standard deviations. Table 4 is the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. As shown, the P-values for rows and columns were 0.4592 and 0.4068 respectively, both being greater than 0.05. This is an indication that both the type of solver and the relative tolerance do not significantly affect the result of angular displacements obtained for the selected initial time step.
Table 3. Average angular displacements in degrees for the different solvers, their standard deviation and correlation coefficient for varied relative tolerance of \([1E-2, 1E-4, 1E-6, 1E-8, 1E-10, 1E-12 \text{ and } 1e-14]\). (Initial time step =0.003).

| Solver   | Case 1          | Case 2          |
|----------|-----------------|-----------------|
|          | Mean            | Standard Deviation | Correlation coefficient | Mean            | Standard Deviation | Correlation coefficient |
| ode45    | 89.3993         | 0.00503          | 0.95856               | 78.3174         | 0.00508           | -0.10679               |
| ode23    | 89.3838         | 0.03503          | 0.15726               | 78.3148         | 0.01613           | -0.95528               |
| ode113   | 89.3979         | 0.00003          | -0.51742              | 78.4601         | 0.34913           | -0.14428               |
| ode15s   | 89.3786         | 0.03522          | -0.18652              | 78.3210         | 0.01468           | 0.99899                |
| ode23s   | 89.3905         | 0.01702          | -0.99896              | 78.3235         | 0.01690           | 0.98724                |
| ode23t   | 89.3894         | 0.01970          | -0.99893              | 78.3193         | 0.00720           | 0.92950                |
| ode23tb  | 89.3924         | 0.01261          | -0.99873              | 78.3193         | 0.00720           | 0.92950                |

Table 4. ANOVA table for calculated angular displacements for different solvers (Rows) and varied Relative tolerance (Columns) for the parameter set of case 1.

| Source of Variation | SS     | Df  | MS       | F        | P-value | F crit  |
|---------------------|--------|-----|----------|----------|---------|---------|
| Rows (Solvers)      | 0.1181 | 6   | 0.01969  | 0.97010  | 0.45925 | 2.3637  |
| Columns (Rel. Tol.)| 0.1286 | 6   | 0.02143  | 1.05556  | 0.40680 | 2.3637  |
| Error               | 0.7310 | 36  | 0.02030  |          |         |         |
| Total               | 0.9778 | 48  |          |          |         |         |

The average angular displacements, standard deviation and their correlation with initial time step, of the different solvers, are shown in Table 5. For all the solvers, there was a strong correlation between the calculated angular displacements and the initial time step. The correlation of ode23t does not vary much with the model parameters. All other schemes, but ode113, had correlation coefficients which changed from negative to positive or vice versa for the two cases of selected parameters. The deviation of angular displacements for all solvers are very small indicating that the calculated result of a solver was not very much affected by the initial time step. Table 6 shows the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. As shown, the P-values for rows and columns were 3.24E-42 and 0.4759 respectively. The calculated angular displacement was affected by the type of solvers for the selected relative tolerance, while the result of angular displacement is unaffected by the initial time step (P<<0.05 for columns).

Table 5. Average angular displacements in degrees for the different solvers, their standard deviation and correlation coefficient for varied Initial time step of \([0.001, 0.005, 0.01, 0.05, 0.1]\) (Rel. Tol =1E-6).

| Solver   | Case 1          | Case 2          |
|----------|-----------------|-----------------|
|          | Mean            | Standard Deviation | Correlation coefficient | Mean            | Standard Deviation | Correlation coefficient |
| ode45    | 89.3979         | 0.00000000004    | -0.6253               | 78.31528       | 0.00000000112    | 0.42770               |
| ode23    | 89.3979         | 0.00000006418    | 0.8659                | 78.31532       | 0.00000002242    | -0.45593              |
| ode113   | 89.3979         | 0.0000000000     | -0.4268               | 78.31528       | 0.00000000000    | -0.90476              |
| ode15s   | 89.3980         | 0.00000023517    | 0.1986                | 78.31532       | 0.00000002821    | -0.51892              |
| ode23s   | 89.3978         | 0.0000005591     | -0.2301               | 78.31575       | 0.00000009937    | 0.85460               |
| ode23t   | 89.3978         | 0.0000003642     | 0.2442                | 78.31566       | 0.00000000002    | 0.24470               |
| ode23tb  | 89.3978         | 0.0000002447     | -0.4973               | 78.31564       | 0.0000083396     | -0.64862              |
The number of times the integrand is evaluated defines the computational cost [37]. As expected, computation costs were higher for parameters of case 1 than that of case 2 due to higher degree of non-linearity. As observed in Figure 2, the computational costs are higher for low relative tolerance for all the solvers. Table 7 shows the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. The $P$-values for rows and columns were 0.002213 and 0.000535 respectively, indicating that the computational cost was affected by the type of solvers and the relative tolerance selected. (P<<0.05 for both rows and columns).

![Figure 2. The Log of relative tolerance versus Log of computation cost for the different techniques for case 1 parameter set (Initial time step = 0.003).](image)

| Source of Variation | SS   | df | MS    | $F$   | $P$-value | $F$ crit |
|---------------------|------|----|-------|-------|-----------|----------|
| Rows (Solvers)      | 1.28E-06 | 6  | 2.13E-07 | 16696.1 | 3.24E-42 | 2.5081   |
| Columns (Initial time step) | 4.63E-11 | 4  | 1.16E-11 | 0.90643 | 0.4759 | 2.7762   |
| Error               | 3.07E-10 | 24 | 1.28E-11 |       |           |          |
| Total               | 1.28E-06 | 34 |       |       |           |          |

Table 7. ANOVA Table for computation cost for the different techniques (Rows) and relative tolerance (columns) for case 1 parameter set.
Fig. 3 shows the log values of computational costs for varied initial time steps. The ode23s is the most expensive technique, followed by the ode23tb and ode23s respectively. The implicit solvers are more expensive than the explicit ones because of the evaluation of Jacobian matrices and derivation of partial derivatives in its procedure for improved efficiency [23]. The multistep techniques are a function of already computed values of preceding steps and may need to evaluate the differential function a minimum number of times. Consequently, ode113 and ode15s are the least expensive. Although ode45 and ode23 are both Runge-Kutta scheme, ode45 is less expensive because it is a higher order Runge-Kutta scheme over the ode23.

Table 8 shows the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. The P-values for rows and columns were 1.83E-07 and 0.527274 respectively, indicating that the computational cost was affected by the type of solvers (P<<0.05) but unaffected by the initial time step (P>0.05 for columns).

![Log Initial time step vs. Log of computation cost](image)

**Table 8.** ANOVA Table for computational cost as it varies with initial time step case 1 parameter set.

| Source of Variation      | SS    | df | MS     | F      | P-value | F crit |
|--------------------------|-------|----|--------|--------|---------|--------|
| Rows (Solvers)           | 1.66E+09 | 6  | 2.77E+08 | 16.5296 | 1.83E-07 | 2.5081 |
| Columns (Initial time step.) | 54630733 | 4  | 13657683 | 0.81643 | 0.527274 | 2.7762 |
| Error                    | 4.01E+08  | 24 | 16728472 |        |         |        |
| Total                    | 2.12E+09  | 34 |        |        |         |        |

Fig. 4 shows the execution time for the different solvers for different relative tolerances. The execution time is expected to be related to the computation cost of a solver. So as expected, the implicit schemes had higher execution time, with ode23s being the highest followed by ode23tb and ode23t respectively. The explicit multi-steps technique (ode15s) had the least execution time, followed by the Runge-Kutta techniques (ode45 and ode23).
Table 9 shows the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. The *P*-values for rows and columns were 0.001548 and 0.00038 respectively, indicating that both the solver and relative tolerance affected the execution time (*P*>0.05 for columns).

![Figure 4](image)

**Figure 4.** The computation time for the different techniques as the tolerated error was varied for case 1 parameter set (initial time step =0.003).

**Table 9.** ANOVA Table for computation time as it varies with each solver (Columns) and Relative Tolerance (Rows).

| Source of Variation   | SS    | df | MS     | F      | P-value | F crit |
|-----------------------|-------|----|--------|--------|---------|--------|
| Rows (Solvers)        | 2106.95 | 6  | 351.158 | 4.560346 | 0.001548 | 2.3637 |
| Columns (Rel. Tol.)   | 2557.85 | 6  | 426.308 | 5.536285 | 0.00038  | 2.3637 |
| Error                 | 2772.09 | 36 | 77.0026 |        |         |        |
| Total                 | 7436.90 | 48 |        |        |         |        |

Fig. 5 shows the execution time for varied initial time steps. Table 8 shows the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. The *P*-values for rows and columns were 9.5E-29 and 0.025105 respectively, indicating that the execution time was affected by the type of solvers and the preset initial time steps (*P*<0.05 for both cases).
Figure 5. The execution time in Seconds for the different techniques as the initial time step was varied for case 1. (Rel. Tol. =1E-6).

Table 10. ANOVA Table for execution time as it varies with each solver (Columns) and Initial time step (Rows).

| Source of Variation       | SS     | Df  | MS    | F      | P-value | F crit |
|---------------------------|--------|-----|-------|--------|---------|--------|
| Rows (Solvers)            | 6.39521| 6   | 1.06586| 1255.737| 9.5E-29 | 2.50818|
| Columns (Initial time step)| 0.01146| 4   | 0.00286| 3.37564| 0.025105| 2.77628|
| Error                     | 0.02037| 24  | 0.00084|        |         |        |
| Total                     | 6.42704| 34  |       |        |         |        |

The percentage number of unsuccessful steps in Table 11 shows the percentage of times a step was repeated due to re-computation of the initial time step. It is related to the local error generated by the respective method embedded in the solver. As observed, the percentage failed steps has positive correlation with the selected relative tolerance. It implies that the smaller the relative tolerance, the lower the local computational error. The ode23tb is the most accurate solver regardless of relative tolerance and parameter combination, having the least number of failed steps.

Table 12 is the ANOVA table for case 1 parameter set. It can be deduced from the table that both the type of solver and tolerated error have effect on the accuracy of the techniques embedded in the solver.
Table 11. The percentage number of unsuccessful steps as the tolerated error was varied (initial time step = 0.003).

| Solver  | Case 1 Parameter set | Case 2 Parameter set |
|---------|----------------------|----------------------|
|         | Mean | Standard Deviation | Correlation Coefficient | Mean | Standard Deviation | Correlation Coefficient |
| ode45   | 6.4701 | 6.8066 | 0.5084 | 3.5533 | 2.2003 | 0.8344 |
| ode23   | 4.2111 | 6.3268 | 0.9291 | 3.5775 | 2.7917 | 0.6551 |
| ode113  | 3.3460 | 2.9321 | 0.8432 | 5.9941 | 2.3794 | 0.7772 |
| ode15s  | 7.3825 | 6.3452 | 0.8846 | 6.9253 | 2.1705 | 0.2485 |
| ode23s  | 4.5723 | 6.6230 | 0.9617 | 1.2878 | 2.0977 | 0.9810 |
| ode23t  | 2.5306 | 4.6039 | 0.9863 | 3.4555 | 5.8658 | 0.9824 |
| ode23tb | 2.8115 | 5.0560 | 0.9847 | 2.9810 | 5.6257 | 0.9963 |

Table 12. ANOVA Table for the accuracy of the solvers versus relative tolerance case 1 parameter set.

| Source of Variation | SS     | Df | MS     | F       | P-value | F crit |
|---------------------|--------|----|--------|---------|---------|--------|
| Rows (Solver)       | 142.3489 | 6  | 23.72482 | 3.252617 | 0.0117  | 2.363751 |
| Columns (Rel. Tol.)| 1544.164 | 6  | 257.3606 | 35.28353 | 1.18E-13 | 2.363751 |
| Error               | 262.5866 | 36 | 7.294072 |         |         |        |
| Total               | 1949.099 | 48 |        |         |         |        |

Fig. 6 shows the percentage of unsuccessful steps of the different solvers for varied initial time steps. The percentage of failed number of steps is highest for the ode15s scheme, followed by the ode45 and ode113 schemes regardless of the initial time steps and parameter set. The implicit solvers had the least number of failed steps, indicating that they are more accurate schemes.

Table 13 is the ANOVA table for case 1 parameters. Similar results were obtained for case 2 parameters. The P-values for rows and columns were 4.85E-14 and 4.22E-06 respectively, indicating that the execution time depends on both the solver and the initial time step (P>0.05 for columns).

**Figure 6.** The percentage number of unsuccessful steps as the initial time step was varied for case 1 parameter set (Rel. Tol. =1E-12).
Table 13. ANOVA Table for accuracy of solver versus initial time step for case 1 parameters set.

| Source of Variation       | SS       | Df | MS     | F        | P-value  | F crit |
|---------------------------|----------|----|--------|----------|----------|--------|
| Rows (Solvers)            | 195.0555 | 6  | 32.509 | 37.4165  | 4.85E-14 | 2.3637 |
| Columns (Initial time steps) | 47.75064 | 6  | 7.9584 | 9.15977  | 4.22E-06 | 2.3637 |
| Error                     | 31.27847 | 36 | 0.8688 |          |          |        |
| Total                     | 274.0846 | 48 |        |          |          |        |

4.0 Discussions and Concluding Remarks

The aim of this work was to evaluate the performance of the seven MATLAB ODE solvers for the Korakianitis and Shi model with changing parameters. The choice of error tolerance is fundamental to the accuracy and cost of results regardless of the type of solver, whereas, the effect of the initial time step is insignificant. However, the value of initial time step can affect the number of unsuccessful steps, and hence the execution time. If computation time is a priority, the multistep techniques (ode113 and ode15s) are best suitable as they are the least expensive. If a high level of accuracy is desired, the implicit solvers, ode23tb followed by ode23s and ode23t respectively are the best alternatives. However, the high accuracy of the implicit solvers will be a tradeoff for computation time as they are relatively costly. For a non-stiff equation, the ode45 solver is a good choice for first alternative as it presents a balance between accuracy and computational cost.

Nomenclature

| Symbol | Description                          |
|--------|--------------------------------------|
| $K_p$  | Coefficient of blood pressure        |
| $K_f$  | Coefficient of tissue friction       |
| $K_b$  | Coefficient of blood motion          |
| $K_v$  | Coefficient of flow vortex           |
| $mi$   | Mitral Valve                         |
| $ao$   | Aortic Valve                         |
| ode45  | MATLAB solver: Ringe-Kutta (4,5) formula (Dormand-Prince pair) |
| ode23  | MATLAB solver: Runge-kutta (2,3) pair |
| ode113 | MATLAB solver: Adam's formula        |
| ode15s | MATLAB solver: NDF and BDF formula   |
| ode23s | MATLAB solver: Rosenbrock formula    |
| ode23t | MATLAB solver: Trapezoidal rule      |
| ode23tb| MATLAB solver: Trapezoidal rule and BDF formula |
| $\Theta$ | Angular displacement of valve leaflet |

Conflict of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

The authors wish to acknowledge the Tertiary Education Trust Fund (TETFund) for the studentship they provided to Emagbetere Eyere, while undertaking his Ph.D. program.
References

[1] T. Koriatinits, Y. Shi, Concentrated parameter model for the human cardiovascular system including heart valve dynamics and atrioventricular interaction, Med. Eng. Phys. 28(7) (2006) 623–628.

[2] M.N. Spijker, Stiffness in numerical initial value problems, Journal of Computational and Applied Mathematics. 72(2) (1996) 393–406.

[3] T. Korakianitis, Y. Shi, Numerical simulation of cardiovascular dynamics with healthy and diseased heart valve, Journal of Biomechanics. 39 (2006) 1964 - 1982.

[4] Y. Shi, Lumped-parameter modelling of cardiovascular system dynamics under different healthy and diseased conditions. Sheffield, UK: Thesis submitted to Department of Cardiovascular Science, Faculty of Medicine, Dentistry and Health, University of Sheffield, 2013.

[5] J.C. Butcher, Numerical methods for ordinary differential equations in the 20th century, Journal of Computational and Applied Mathematics. 125(1) (2000) 1–29.

[6] T.E. Hull, W.H. Enright, A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, SIAM Journal on Numerical Analysis. 9(4) (1972) 603-637.

[7] F.T. Krogh, On testing a subroutine for the numerical integration of ordinary differential equations, Journal of the ACM (JACM). 20(4) (1973) 545-562

[8] G.D. Byrne, A.C. Hindmarsh, Stiff ODE solvers: a review of current and coming attractions, Journal of Computational Physics. 70(1) (1987) 1-62.

[9] T.D. Bui, A.K. Oppenheim, D.T. Pratt, Recent advances in methods for numerical solution of O.D.E. initial value problems, Journal of Computational and Applied Mathematics. 11 (1984) 283-296

[10] H.W. Enright, T.E. Hull, B. Lindberg, Comparing numerical methods for stiff systems of ODE's, BIT Numerical Mathematics. 15(1) 1975 10-48

[11] L. F. Shampine, Stiff and nonstiff differential equation solvers, II: Detecting stiffness with Runge-Kutta methods, ACM Transactions on Mathematical Software (TOMS). 3(1) (1977) 44-53.

[12] L.F. Shampine, Evaluation of a test set for stiff ODE solvers, ACM Transactions on Mathematical Software (TOMS). 7(4) (1981) 409-420.

[13] L. Petzold, Automatic selection of methods for solving stiff and non-stiff systems of ordinary differential equations, SIAM Journal on Scientific and Statistical Computing, 4(1) (1983) 136-148.

[14] W.H. Enright, J.D. Pryce, Two FORTRAN packages for assessing initial value methods, ACM Transactions on Mathematical Software (TOMS). 13(1) (1987) 1-27.

[15] L.F. Shampine, R.M. Corless, Initial value problems for ODEs in problem solving environments, Journal of Computational and Applied Mathematics. 125 (2000) 31-40.

[16] D. Petcu, Software issues in solving initial value problems for ordinary differential equations, Creative Math. 13 (2004) 97-110.

[17] A. Sandu et al., Benchmarking stiff ode solvers for atmospheric chemistry problems-I. Implicit vs Explicit, Atmospheric Environment. 31(19) (1997) 3151-3166

[18] L.A. Nejad, A comparison of stiff ODE solvers for astrochemical kinetics problems, Astrophysics and Space Science. 299(1) (2005) 1-29.
[19] S. Abelman, K.C. Patidar, Comparison of some recent numerical methods for initial-value problems for stiff ordinary differential equations, Computers & Mathematics with Applications. 55(4) (2008) 733-744.

[20] S. Dallas, K. Machairas, A Comparison of ODE solvers for dynamical systems with impacts, Journal of Computational and Nonlinear Dynamics, Special Issue on Dynamics of Systems with Impacts. (2017). doi:10.1115/1.4037074.

[21] A. Ryuichi, N. Michihiro, V. Remi, Behind and beyond the MATLAB ODE Suite. CRM-2651, 2000.

[22] M.A. Natick, Using MATLAB version 5.1, The MathWorks, 1997.

[23] J.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, SIAM J. Sci. Comput. 18(1) (1997) 1-22.

[24] F. Scheid, Schaum’s outline of theory and problems of numerical analysis, Second Edition, McGraw-Hill Companies Inc., 1988, pp. 184-240.

[25] J.D. Lambert, Numerical methods for ordinary differential equations: the initial value problem, Chichester, John Wiley & Sons, Inc., 1991.

[26] I. Farago, Numerical methods for ordinary differential equations, Eotvos Lorand University, 2013.

[27] E. Kreyszig, Advance engineering mathematics, John Wiley & Sons, Inc., 2006.

[28] J.R. Dormand, P.J. Prince, A family of embedded Runge-Kutta formulae, Journal of Computational Mathematics. 6(1) (1980) 19-26.

[29] E. Fehlberg, Klassische Runge-Kutta-formula vierter und niedrigerer ordnung mit schrittweiten-kontrolle und ihre anwendung auf waermeleitungsprobleme, Computing. 6(1) (1970) 61-71.

[30] L.F. Shampine, Some practical Runge-Kutta formulas, Math. Comp. 46(173) (1986) 135-150.

[31] P. Bogacki, L.F. Shampine, An efficient Runge-Kutta (4,5) pair, Computer Maths. Applied. 32(6) (1996) 15-28.

[32] L. Brugnano, D. Trigiante, Solving differential problems by multistep, initial and boundary value methods, Gordon and Breach, Amsterdam, 1998.

[33] F. Lavernano, F. Mazzia, Solving ordinary differential equations by generalised Adams method: properties and implementation techniques, Applied Numerical Mathematics. 28(2-4) (1998) 107-126.

[34] Y.Y. Lui, Numerical methods for differential equations, Hongkong: Lecture Note, Department of Mathematics, City University of Hongkong, Kowloon, 2016.

[35] E. Hairer, G. Wanner, Solving differential equations II, stiff and differential algebraic problems, Springer-Verlag, Berlin, 1991, pp. 5-8.

[36] J.D. Verwer, A second-order Rosenbrock method applied to photochemical dispersion problems, SIAM Journal of Scientific Computation. 20 (1999) 1456-1480.

[37] E. Hairer, G. Wanner, On the instability of the BDF formulas, SIAM Journal of Numerical Anal. 20(6) (1983) 1206-1209.

[38] D. Omale, P.B. Ojih, M.O. Ogwo, Mathematical analysis of stiff and non-stiff initial value problems of ordinary differential equation using MATLAB, International Journal of Scientific & Engineering Research. 5(9) (2014) 49-59.
[39] S.A. Belmana, K.C. Patidarb, Comparison of some recent numerical methods for initial-value problems for stiff ordinary differential equations, Computers and Mathematics with Applications. 55 (2007) 733–744.

[40] S.A.M. Yatim et al., A quantitative comparison of numerical method for solving stiff ordinary differential equations, Mathematical Problems in Engineering. (2011) 1-12

[41] E. Cocherova, Modification of ordinary differential equations MATLAB solver, Radioengineering. 12(4) (2003) 63-66.

[42] J. Hall, A. Guyton, Textbook of medical physiology, Eleventh Edition, Philadelphia, Pennsylvania: Elsevier Saunders, 2006.

[43] R.M. Bernie, M.N. Levy, Cardiovascular physiology, St Louis, MO, Mosby, 2001.