SOLUTIONS OF GENERAL SECOND ORDER ODES USING DIRECT BLOCK METHOD OF RUNGE-KUTTA TYPE

(Penyelesaian bagi Persamaan Pembezaan Biasa Umum Peringkat Kedua dengan Kaedah Blok Langsung Jenis Runge-Kutta)

NUR ZAHIDAH MUKHTAR, ZANARIAH ABDUL MAJID, FUDZIAH ISMAIL & MOHAMED SULEIMAN

ABSTRACT

This paper presents a three point block variable step size method of Runge-Kutta type for solving general second order ordinary differential equations (ODEs). The block method is formulated using Lagrange interpolation polynomial. Most of the mathematical problems which involve higher order ODEs could be reduced to system of first order equations. The proposed method obtains the numerical solutions directly without reducing to first order systems of ODEs. The method is used to compute the solutions at three points simultaneously by integrating the coefficients over the closest point in the block. The stability region of the block method is also studied. The numerical results obtained shows that the proposed method is more efficient compared to existing block methods in terms of total steps and execution time.

Keywords: Block method; variable step size; ordinary differential equations

1. Introduction

This study is intended to solve second order non-stiff initial value problems (IVPs) of ordinary differential equations in the following form

\[ y'' = f(x, y, y'), \quad y(a) = y_0, \quad y'(a) = y'_0, \quad x \in [a, b] \]

(1)

The second order ODEs can be formed from many real problems in a wide variety of applications especially in science and engineering such as fluid dynamic, electrical circuit and other area of applications. The approach here is to solve Eq. (1) using block implicit one-step method. This block method of Runge-Kutta type has been studied by Rosser (1967), Shampine and Watts (1969) and Majid et al. (2003). Rosser (1967) introduced a block with a new approximation values simultaneously based on Newton-Cotes type while Majid et al.
Nur Zahidah Mukhtar, Zanariah Abdul Majid, Fudziah Ismail & Mohamed Suleiman

(2003) presented a two point implicit block one-step method based on integration formula using Newton backward divided difference formula.

In the literature, there were a few researchers suggested to solve higher order ODEs directly such as Chakravarti and Worland (1971), Fatunla (1991), Suleiman (1989), and Omar and Suleiman (2005). The higher order ODEs can be reduced to a system of first order equations and then solved using first order ODEs. Consequently, this approach will enlarge the system of first order ODEs.

The idea in this paper is to derive the formulae of the three point block method based on the Lagrange interpolation polynomial and estimate the solutions of second order ODEs at three points simultaneously. Majid et al. (2006) has presented a three point implicit block method for solving system of first order ODEs. The numerical results in Majid et al. (2003, 2006) were compared to Rosser (1967). Both methods were implemented to solve the system of first order ODEs. Thus, this method inspired our research to solve the system of second order ODEs directly and compared the performance of the proposed method with Majid et al. (2006).

2. Formulation of the Method

In Figure 1, the proposed method will compute the three approximation values of \( y_{n+1}, \ y_{n+2} \) and \( y_{n+3} \) simultaneously. The interval \([a,b]\) is divided into a series of blocks with each block containing three points and the computed block has the step size \(3h\). The solution at the point \( x_n \) is used to start the \( k \) th block while the solution at the point \( x_{n+3} \) is the last point in the \( k \) th block will be used to start the \((k+1)\) th block and the process continues for the next block.

\[ \int_{x_n}^{x_{n+1}} y''(x) \, dx = \int_{x_n}^{x_{n+1}} f(x, y, y') \, dx \]

\[ \int_{x_n}^{x_{n+1}} y'(x) \, dx = \int_{x_n}^{x_{n+1}} \int_{x_n}^{x} f(x, y, y') \, dx \, dx \]

Taking \( x_{n+1} = x_n + h \), gives

Figure 1: Three point block method

The predictor formula involved the Euler method, while the corrector formula of the block method in Figure 1 was derived using Lagrange interpolation polynomial. The first point, \( y_{n+1} \) and \( y'_{n+1} \) can be obtaining by integrating Eq. (1) once and twice over the interval \([x_n, x_{n+1}]\),
Solutions of general second order ODEs using direct block method of Runge-Kutta type

\[ y'(x_{n+1}) - y'(x_n) = \int_{x_n}^{x_{n+1}} f(x, y, y') \, dx \]
\[ y(x_{n+1}) - y(x_n) - h y'(x_n) = \int_{x_n}^{x_{n+1}} (x_{n+1} - x) f(x, y, y') \, dx \]

(3)

The Lagrange interpolation polynomial will replace the \( f(x, y, y') \) in Eq. (3). The interpolation points in the block will involve \((x_n, f_n), (x_{n+1}, f_{n+1}), (x_{n+2}, f_{n+2})\) and \((x_{n+3}, f_{n+3})\).

Taking \( s = \frac{x - x_{n+3}}{h} \) and \( dx = h \, ds \); then replace into Eq. (3) and changing the limit of integration from -1 to 0. The formulae of \( y_{n+1} \) and \( y'_{n+1} \) will be obtained by using MATHEMATICA,

\[ y_{n+1}' = y_n' + \frac{h}{24} (9 f_n + 19 f_{n+1} - 5 f_{n+2} + f_{n+3}), \]
\[ y_{n+1} = y_n + h y_n' + \frac{h^2}{360} (97 f_n + 114 f_{n+1} - 39 f_{n+2} + 8 f_{n+3}). \]

(4)

To approximate the value of \( y_{n+2} \) and \( y'_{n+2} \), let \( x_{n+2} = x_n + 2h \) and integrate Eq. (1) once and twice over the interval \([x_{n+1}, x_{n+2}]\).

\[ y_{n+2}' = y_{n+1}' + \frac{h}{24} (-f_n + 13 f_{n+1} + 13 f_{n+2} - f_{n+3}), \]
\[ y_{n+2} = y_{n+1} + h y_{n+1}' + \frac{h^2}{360} (-8 f_n + 129 f_{n+1} + 66 f_{n+2} - 7 f_{n+3}). \]

(5)

Taking \( x_{n+3} = x_n + 3h \), the value of \( y_{n+3} \) and \( y'_{n+3} \) can be obtained by integrating once and twice of Eq. (1) over the interval \([x_{n+2}, x_{n+3}]\). Then, we obtain the following corrector formulae,

\[ y_{n+3}' = y_{n+2}' + \frac{h}{24} (f_n - 5 f_{n+1} + 19 f_{n+2} + 9 f_{n+3}) \]
\[ y_{n+3} = y_{n+2} + h y_{n+2}' + \frac{h^2}{360} (7 f_n - 36 f_{n+1} + 171 f_{n+2} + 38 f_{n+3}) \]

(6)

In order to determine the order and error constant of the method, we had referred to Lambert (1981) and Fatunla (1991). Thus, the formula in Eq. (4), (5) and (6) are order of \( p = 4 \) and the error constant is \( C_{p+2} = C_6 \)

\[ \begin{bmatrix} 19/160 & -7/480 & 11/720 & 11/1440 & -19/720 & -17/1440 & \end{bmatrix} \neq 0. \]

3. Implementation of the Method

The approximation values of \( y_{n+1}, y'_{n+1}, y_{n+2}, y'_{n+2}, y_{n+3} \) and \( y'_{n+3} \) in Eq. (4), (5) and (6) will be approximated using the predictor-corrector schemes. If \( r \) corrections are needed, then the
sequence of computations at any mesh point is \((PE)(C_0E)\ldots(C_rE)\) where \(P\), \(E\) and \(C\) indicate the predictor, evaluation of the function \(f\) and the corrector respectively.

The predictor equations:

\[
P: \quad \begin{align*}
y'_{n+i,0}^p &= y'_{n+(i-1),0} + hf_{n+(i-1),0} \\
y''_{n+i,0}^p &= y''_{n+(i-1),0} + h^2 f_{n+(i-1),0} \\
E: \quad f^p_{n+i,0} &= f(x_{n+i}, y_{n+i,0}^p, y''_{n+i,0}^p) \quad i = 1, 2, 3. \quad (7)
\end{align*}
\]

The corrector equations: 

For \(r = 0\):

\[
C: \quad \begin{align*}
y'_{n+i+1}^c &= y'_{n+(i-1),1} + h \sum_{j=0}^3 \alpha_{j,i} f^p_{n+j,0} \\
y''_{n+i+1}^c &= y''_{n+(i-1),1} + h^2 \sum_{j=0}^3 \beta_{j,i} f^p_{n+j,0} \\
E: \quad f^c_{n+i+1} &= f(x_{n+i+1}, y_{n+i+1,1}^c, y''_{n+i+1}^c) \quad i = 1, 2, 3. \quad (8)
\end{align*}
\]

For \(r = 1, 2, 3\ldots\):

\[
C: \quad \begin{align*}
y'_{n+i+r}^c &= y'_{n+(i-1), r+1} + h \sum_{j=0}^3 \alpha_{j,i} f^c_{n+j,r} \\
y''_{n+i+r}^c &= y''_{n+(i-1), r+1} + h^2 \sum_{j=0}^3 \beta_{j,i} f^c_{n+j,r} \\
E: \quad f^c_{n+i+r} &= f(x_{n+i+r}, y_{n+i+r}, y''_{n+i+r}) \quad i = 1, 2, 3. \quad (9)
\end{align*}
\]

In the code, the convergence test will iterate the corrector to converge as follows:

\[
|y_{n+3,s+1} - y_{n+3,s}| < 0.1 \times \text{TOL},
\]

where \(s\) is the number of iterations. After the successful convergence test, the error for the block will be performed. Then, the maximum error is defined as follow;

\[
\text{MAXE} = \max_{1 \leq i \leq TS} \left( \max_{1 \leq j \leq N} (E_j) \right),
\]

where \(TS\) is the number of successful steps, \((E_j)\) is the error and \(N\) is the number of equation.

The selection of the next step size will be similar as in Majid et al. (2003). The choices for the next step size will be constant or double. The step size will be restricted to half if the step size failure.
4. Stability Region

The main concern in this section is to discuss the stability of the proposed method derived in the previous section on a linear second order problem when the method is applied to the test equation

\[ y'' = \theta y' + \lambda y. \]  

(10)

The formulae of the three point implicit block method were given by Eq. (4), (5) and (6). Hence for \( r = 0 \), the test equation in Eq. (10) is substituting into Eq. (7) and (8). Then, transforming into a matrix form and we obtain the stability polynomial as follows:

\[ r = 0: \]

\[
\begin{align*}
&\left(1 + \frac{77H_1^2}{24} + \frac{383H_1^3}{72} + \frac{337H_1^4}{9} + \frac{25H_1^5}{6} + \frac{H_1^6}{4} + \frac{49H_2}{180} + \frac{11H_1H_2}{30} - \\
&289H_1^2H_2 + \frac{13H_1^3H_2}{10} + \frac{3H_1^5H_2}{40} + \frac{467H_2^2}{1080} - \frac{863H_1H_2^2}{2160} - \frac{517H_1^2H_2^2}{864} - \\
&193H_1^3H_2^2 + \frac{H_1^4H_2^2}{8} + \frac{397H_1^3H_2^3}{8640} - \frac{3989H_1H_2^3}{720} + \frac{49H_1^2H_2^3}{160} + \frac{9H_1^3H_2^3}{160} + \\
&1607H_1^4H_2^4 + \frac{61H_1^3H_2^4}{8640} - \frac{3H_1^5H_2^4}{320} + \frac{7H_1^5H_2^5}{960} + \frac{7H_1^4H_2^5}{160} - \frac{7H_2^6}{640} \right) t^4 + \left(2 - \frac{41H_1}{12} \right)
\end{align*}
\]

\[
\begin{align*}
&\frac{397H_1^2}{72} + \frac{335H_1^3}{72} + \frac{199H_1^4}{72} + \frac{5H_1^5}{6} + \frac{H_1^6}{4} \frac{859H_2}{90} - \frac{1211H_1H_2}{80} - \frac{1967H_2^2}{144} - \\
&\frac{937H_1^2H_2}{120} + \frac{13H_1^3H_2}{40} + \frac{27H_1^5H_2}{2160} - \frac{18949H_2^2}{2160} - \frac{23999H_1H_2^2}{2160} - \frac{16789H_2^3}{2160} - \\
&\frac{2183H_1^3H_2^2}{720} + \frac{31H_1^4H_2^2}{40} + \frac{12563H_1^2H_2^3}{1080} + \frac{3539H_1H_2^3}{144} - \frac{247H_1^2H_2^3}{80} - \frac{39H_1^3H_2^3}{120} + \\
&\frac{717H_1^4H_2^4}{3456} + \frac{277H_1^3H_2^4}{320} + \frac{57H_1^5H_2^4}{240} - \frac{13H_1^5H_2^5}{640} + \frac{23H_1^4H_2^5}{320} - \frac{H_1^6}{24} + \\
&\frac{7H_2^6}{36} + \frac{H_1^3}{36} + \frac{49H_2}{180} + \frac{3H_1H_2^2}{40} + \frac{23H_1^2H_2}{40} + \frac{H_1^3H_2}{2160} - \frac{43H_1^2H_2}{1080} - \\
&\frac{71H_1^2H_2^2}{4320} - \frac{H_3^3}{135} + \frac{H_1H_2^3}{216} - \frac{H_2^4}{2160} \right) t^6 = 0,
\end{align*}
\]

(11)

where \( H_1 = h\theta \) and \( H_2 = h^2\lambda \).

For the case \( r = 1 \), we substitute \( C \) and \( E \) into the right hand side of \( C \) in Eq. (9) and we obtain the stability polynomial and plot the stability region as in Figure 3. Figures 2 and 3 show the stability region of the three point implicit block one step method when \( r = 0 \) and \( r = 1 \).
The stability region is plotted using MATHEMATICA and the shaded region inside the boundary in Figures 2 and 3 demonstrate the stability region for the proposed method.

5. Numerical Results

The tables below show the numerical results for the three given problems when solved using the proposed method compared to the problems reduce to system of first order ODEs using the method in Majid et al. (2006).

The following notations are used in Tables 1-3:

<table>
<thead>
<tr>
<th>TOL</th>
<th>Tolerance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTD</td>
<td>Method employed</td>
</tr>
<tr>
<td>TS</td>
<td>Total steps taken</td>
</tr>
</tbody>
</table>
Solutions of general second order ODEs using direct block method of Runge-Kutta type

<table>
<thead>
<tr>
<th>FS</th>
<th>Total failure steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXE</td>
<td>Magnitude of the maximum error of the computed solution</td>
</tr>
<tr>
<td>AVERR</td>
<td>The average error</td>
</tr>
<tr>
<td>FCN</td>
<td>Total function calls</td>
</tr>
<tr>
<td>TIME</td>
<td>Execution time taken in microseconds</td>
</tr>
<tr>
<td>3PRED</td>
<td>Implementation of the three point implicit block one-step method in Majid et al. (2006) i.e. reducing to system of first order ODEs</td>
</tr>
<tr>
<td>3PDIR</td>
<td>Implementation of the direct three point block method proposed in this research</td>
</tr>
</tbody>
</table>

Tested problems:

Problem 1: \[2y'' - (y')^2 + 4y^2 = 0\]
\[y\left(\frac{\pi}{6}\right) = \frac{1}{4}, \quad y'\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2}, \quad x \in \left[\frac{\pi}{6}, 2\right]\]

Exact solution: \[y(x) = (\sin x)^2\]
Source: Jator & Li (2009)

Problem 2: \[y'' = -y_2 + \sin \pi x, \quad y_1(0) = 0, \quad y_1'(0) = -1\]
\[y_2'' = y_1 + 1 - \pi^2 \sin \pi x, \quad y_2(0) = 1, \quad y_2'(0) = 1 + \pi \quad [0, 10]\]

Exact solution: \[y_1(x) = 1 - e^x, \quad y_2(x) = e^x + \sin \pi x\]
Source: Bronson (1973)

Problem 3: \[x^2y'' + xy' + (x^2 - 0.25)y = 0\]
\[y(1) = \sqrt{\frac{2}{\pi}} \sin 1, \quad y'(1) = \frac{(2 \cos 1 - \sin 1)}{\sqrt{2\pi}}, \quad x \in [1, 8]\]

Exact solution: \[y(x) = \sqrt{\frac{2}{\pi x}} \sin x\]
Source: Jator & Li (2009)

Table 1: Comparison between 3PRED and 3PDIR for solving Problem 1

<table>
<thead>
<tr>
<th>TOL</th>
<th>MTD</th>
<th>TS</th>
<th>FS</th>
<th>MAXE</th>
<th>AVERR</th>
<th>FCN</th>
<th>TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>3PRED</td>
<td>2</td>
<td>0</td>
<td>7.69304e-003</td>
<td>2.61683e-004</td>
<td>37</td>
<td>197</td>
</tr>
<tr>
<td>3PDIR</td>
<td>2</td>
<td>0</td>
<td></td>
<td>9.97275e-004</td>
<td>5.83536e-004</td>
<td>28</td>
<td>180</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>3PRED</td>
<td>8</td>
<td>1</td>
<td>1.06150e-005</td>
<td>2.91476e-006</td>
<td>148</td>
<td>325</td>
</tr>
<tr>
<td>3PDIR</td>
<td>4</td>
<td>0</td>
<td></td>
<td>1.90443e-005</td>
<td>6.18695e-006</td>
<td>61</td>
<td>241</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>3PRED</td>
<td>19</td>
<td>1</td>
<td>2.34728e-007</td>
<td>6.20481e-008</td>
<td>340</td>
<td>547</td>
</tr>
<tr>
<td>3PDIR</td>
<td>9</td>
<td>0</td>
<td></td>
<td>2.32153e-007</td>
<td>1.34058e-007</td>
<td>136</td>
<td>291</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>3PRED</td>
<td>49</td>
<td>1</td>
<td>4.72364e-009</td>
<td>1.22088e-009</td>
<td>841</td>
<td>1132</td>
</tr>
<tr>
<td>3PDIR</td>
<td>23</td>
<td>0</td>
<td></td>
<td>1.07486e-008</td>
<td>2.94740e-009</td>
<td>343</td>
<td>432</td>
</tr>
</tbody>
</table>
Table 2: Comparison between 3PRED and 3PDIR for solving Problem 2

<table>
<thead>
<tr>
<th>TOL</th>
<th>MTD</th>
<th>TS</th>
<th>FS</th>
<th>MAXE</th>
<th>AVERR</th>
<th>FCN</th>
<th>TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>3PRED</td>
<td>21</td>
<td>1</td>
<td>5.69787e-003</td>
<td>8.64637e-004</td>
<td>325</td>
<td>1470</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>10</td>
<td>0</td>
<td>2.80809e-002</td>
<td>1.69345e-002</td>
<td>121</td>
<td>540</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>3PRED</td>
<td>74</td>
<td>1</td>
<td>4.20700e-006</td>
<td>1.02946e-006</td>
<td>1015</td>
<td>3650</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>25</td>
<td>0</td>
<td>8.5020e-004</td>
<td>2.95871e-004</td>
<td>298</td>
<td>931</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>3PRED</td>
<td>219</td>
<td>2</td>
<td>4.16237e-007</td>
<td>4.21474e-008</td>
<td>3289</td>
<td>9170</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>82</td>
<td>1</td>
<td>2.74912e-006</td>
<td>8.18396e-007</td>
<td>964</td>
<td>3654</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>3PRED</td>
<td>665</td>
<td>2</td>
<td>3.38589e-010</td>
<td>3.38589e-010</td>
<td>9352</td>
<td>32901</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>157</td>
<td>2</td>
<td>5.73548e-008</td>
<td>2.79436e-008</td>
<td>1810</td>
<td>5422</td>
</tr>
</tbody>
</table>

Table 3: Comparison between 3PRED and 3PDIR for solving Problem 3

<table>
<thead>
<tr>
<th>TOL</th>
<th>MTD</th>
<th>TS</th>
<th>FS</th>
<th>MAXE</th>
<th>AVERR</th>
<th>FCN</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-2}</td>
<td>3PRED</td>
<td>7</td>
<td>0</td>
<td>8.14259e-004</td>
<td>3.07213e-004</td>
<td>124</td>
<td>329</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>6</td>
<td>0</td>
<td>8.58857e-004</td>
<td>3.08802e-004</td>
<td>76</td>
<td>235</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>3PRED</td>
<td>17</td>
<td>0</td>
<td>1.12729e-005</td>
<td>4.87629e-006</td>
<td>307</td>
<td>548</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>10</td>
<td>0</td>
<td>2.56193e-005</td>
<td>9.57997e-006</td>
<td>145</td>
<td>303</td>
</tr>
<tr>
<td>10^{-6}</td>
<td>3PRED</td>
<td>43</td>
<td>0</td>
<td>2.51517e-007</td>
<td>1.17662e-007</td>
<td>772</td>
<td>1180</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>24</td>
<td>0</td>
<td>9.25077e-007</td>
<td>3.17166e-007</td>
<td>334</td>
<td>443</td>
</tr>
<tr>
<td>10^{-8}</td>
<td>3PRED</td>
<td>179</td>
<td>1</td>
<td>9.70038e-010</td>
<td>4.37159e-010</td>
<td>2683</td>
<td>4897</td>
</tr>
<tr>
<td></td>
<td>3PDIR</td>
<td>58</td>
<td>0</td>
<td>2.67589e-009</td>
<td>9.97129e-009</td>
<td>820</td>
<td>741</td>
</tr>
</tbody>
</table>

Figure 4: Comparison between 3PRED and 3PDIR methods for solving Problem 1 in terms of – log (tolerance) versus number of total steps and time in microsecond
5. Discussions and Conclusions

From tables 1-3 and figures 4-6, the total number of steps taken by 3PDIR is comparable or less than 3PRED. It is also observed that 3PDIR required less number of function call compared to 3PRED at all given tolerances. In Problem 1 and 3, the maximum error of 3PDIR is comparable or one decimal place better than 3PRED method but vice versa at TOL=10^{-8}. The accuracy of 3PDIR in Problem 2 is one and two decimal places larger than 3PRED but it is still within the given tolerances. In terms of execution time, 3PDIR is faster than 3PRED when the problems were solved directly. Hence, we have shown the efficiency of the proposed method is suitable for solving general second order ODEs directly.
Acknowledgments

The authors gratefully acknowledge that this research was partially supported by Universiti Putra Malaysia under the Graduate Research Fellowship (GRF).

References


Institute for Mathematical Research
Universiti Putra Malaysia
43400 Serdang
Selangor DE, MALAYSIA
E-mail: bintimukhtar1408@yahoo.com, zanariah@math.upm.edu.my*, fudziah@math.upm.edu.my, mohamed@math.upm.edu.my

Department of Mathematics
Faculty of Science
Universiti Putra Malaysia
43400 Serdang
Selangor DE, MALAYSIA
E-mail: zanariah@math.upm.edu.my*, fudziah@math.upm.edu.my, mohamed@math.upm.edu.my

*Corresponding author