Explicit and Implicit 3-point Block Methods for
Solving Special Second Order
Ordinary Differential Equations Directly

Fudziah Ismail
Department of Mathematics, Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia
fudziah_i@yahoo.com.my

Yap Lee Ken
Institute for Mathematical Research, Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia

Mohamad Othman
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
43400 Serdang, Selangor, Malaysia

Abstract
This paper focused mainly on deriving explicit and implicit 3-point block methods of constant step size using linear difference operator for solving special second order ordinary differential equations (ODEs). The methods compute the solutions of the ODEs at three points simultaneously. Regions of stability for both the explicit and implicit block methods are presented. A standard set of problems are solved using the new methods and the numerical results are compared when the same set of problems are solved using existing methods. The results suggest a significant
improvement in efficiency of the new methods in terms of number of steps and accuracy.

Mathematics Subject Classification: 65L06; 65L05

Keywords: initial value problems, linear difference operator, block method

1. Introduction

Since the advent of computers, the numerical solution of Initial Value Problems (IVP) for Ordinary Differential Equations (ODEs) has been the subject of research by numerical analysts, especially methods for the numerical solutions of the special second-order ODEs of the form

\[ y'' = f(x, y), \quad y(x_0) = y_0, \quad y'(x_0) = y'_0. \]  

(1)

in which \( f \) does not depend on \( y' \). In general the special second order equation (1) can be reduced to an equivalent first-order system of twice the dimension and solved using the standard Runge-Kutta method or multistep method. However, it is more efficient if the equation can be solved directly as suggested by Chakravati and Worland [1], Dahlquist [2], Jeltsch [6], Sharp and Fine [10], Dormand et. al [3] and El-Mikkawy and El-Desouky [4]. These methods compute the numerical solution at one point at a time. More efficient codes can be developed if the solutions to the equations can be computed at more than one point simultaneously. This type of method is called block method.

The block method for numerical solutions of first order ODEs has been proposed by Lee [7], while Omar et al. [9] and Majid and Suleiman [8] developed block method for general second order ODEs.

In a block method, a set of new values that are obtained by each application of the formula is referred to as “block”. In \( r \)-point block method, each application of the formulas generates a block of \( r \) new equally spaced solution values simultaneously. The computation which proceeds in blocks is based on the computed values at earlier blocks. The computed values at the previous k-block are used to compute the current block containing \( r \) points and the method is called \( r \)-point \( k \)-block method. Fatunla [5] has developed 2-point block method for the special second order ODEs. In this paper, we are going to derive 3-point block method for the solution of the special second order ODEs.
2. Derivation of 3-Point 1-Block Method

According to Fatunla [5], the \( r \)-point \( k \)-block method for (1) is given by the matrix finite difference equation

\[
Y_m = \sum_{i=1}^{k} A^{(i)} Y_{m-i} + h^2 \sum_{i=0}^{k} B^{(i)} F_{m-i}
\]

where \( Y_m = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_{n+r} \end{bmatrix} \), \( F_m = \begin{bmatrix} f_{n+1} \\ f_{n+2} \\ \vdots \\ f_{n+r} \end{bmatrix} \), (for \( n = mr, m = 0,1 \ldots \)), \( A^{(i)}, B^{(i)} \) are \( r \) by \( r \) matrices.

The block scheme is explicit if the coefficient matrix \( B^{(0)} \) is a null matrix.

The linear difference operator \( L \) associated with the linear multistep method is defined by

\[
L[Z(x),h] = \sum_{j=0}^{\infty} \left[ \alpha_j Z(x + jh) - h^2 \beta_j Z''(x + jh) \right]
\]

where \( Z(x) \) is an arbitrary function and is assumed to be differentiable. Expand \( Z(x + jh) \) and \( Z''(x + jh) \) about \( x \) and collecting terms to obtain

\[
L[Z(x),h] = C_0 Z(x) + C_1 hZ'(x) + \cdots + C_q h^q Z^{(q)}(x) + O(h^{q+1})
\]

whose coefficients \( C_q, q = 0,1,\ldots \) are constants and given as:

\[
C_0 = \sum_{j=0}^{k} \alpha_j \\
C_1 = \sum_{j=1}^{k} j \alpha_j \\
\vdots \\
C_q = \frac{1}{q!} \left[ \sum_{j=1}^{k} j^q \alpha_j - q(q-1) \sum_{j=1}^{k} j^{q-2} \beta_j \right]
\]

The associated linear multistep method and the linear difference operator (3) are said to be of order \( p \) if \( C_0 = C_1 = \ldots = C_{p+1} = 0 \) and \( C_{p+2} \neq 0 \).

Derivation of Explicit 3-Point 1-Block Method:
The explicit 3-point 1-block method \( (r = 3, k = 1) \) for second order IVP can be derived from (3) by ensuring that the coefficient matrix \( B^{(0)} \) is null. It can be described by matrix finite difference equation (2)\
\[
A^{(0)} Y_m = A^{(1)} Y_{m-1} + h^2 B^{(1)} F_{m-1}
\]
(5)
with the matrix coefficients specified as\
\[
\begin{bmatrix}
    y_{n+1} \\
    y_{n+2} \\
    y_{n+3}
\end{bmatrix}
\]
\[
\begin{bmatrix}
    y_{n-2} \\
    y_{n-1} \\
    y_n
\end{bmatrix}
\]
\[
\begin{bmatrix}
    f_{n-2} \\
    f_{n-1} \\
    f_n
\end{bmatrix}
\]
\[
A^{(0)} = \begin{bmatrix}
    1 & 0 & 0 \\
    0 & 1 & 0 \\
    0 & 0 & 1
\end{bmatrix},
A^{(1)} = \begin{bmatrix}
    b_{11} & b_{12} & b_{13} \\
    b_{21} & b_{22} & b_{23} \\
    b_{31} & b_{32} & b_{33}
\end{bmatrix},
B^{(1)} = \begin{bmatrix}
    c_{11} & c_{12} & c_{13} \\
    c_{21} & c_{22} & c_{23} \\
    c_{31} & c_{32} & c_{33}
\end{bmatrix}
\]
We consider the linear difference operator (3) and using (4) three sets of equations are obtained. The following three sets of equations have to be satisfied so that the method is at least of order three.

1\textsuperscript{st} set of equations:
\[
\begin{align*}
b_{11} &= 0, & b_{12} &= -1, & b_{13} &= 2, \\
c_{11} + c_{12} + c_{13} &= 1, & c_{12} + 2c_{13} &= 2, & c_{12} + 4c_{13} &= \frac{25}{6}.
\end{align*}
\]

2\textsuperscript{nd} set of equations:
\[
\begin{align*}
b_{21} &= 0, & b_{22} &= -2, & b_{23} &= 3, \\
c_{21} + c_{22} + c_{23} &= 3, & c_{22} + 2c_{23} &= 7, & c_{22} + 4c_{23} &= \frac{35}{2}.
\end{align*}
\]

3\textsuperscript{rd} set of equations:
\[
\begin{align*}
b_{31} &= 0, & b_{32} &= -3, & b_{33} &= 4, \\
c_{31} + c_{32} + c_{33} &= 6, & c_{32} + 2c_{33} &= 16, & c_{32} + 4c_{33} &= 47.
\end{align*}
\]
Solving the equations, giving
Explicit and implicit 3-point block methods

\[ A^{(0)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A^{(1)} = \begin{bmatrix} 0 & -1 & 2 \\ 0 & -2 & 3 \\ 0 & -3 & 4 \end{bmatrix}, \quad B^{(i)} = \begin{bmatrix} 1 \\ 12 \\ 5 \\ 11 \\ 2 \end{bmatrix}, \quad B^{(i)} = \begin{bmatrix} 1 \\ -6 \\ -7 \\ -2 \\ -15 \end{bmatrix}, \quad B^{(i)} = \begin{bmatrix} 13 \\ 12 \\ 21 \\ 4 \\ 31 \\ 2 \end{bmatrix} \]

Hence, the explicit 3-point 1-block method can be written as

\[ \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{bmatrix} = \begin{bmatrix} 0 & -1 & 2 \\ 0 & -2 & 3 \\ 0 & -3 & 4 \end{bmatrix} \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_{n} \end{bmatrix} + h^2 \begin{bmatrix} 1 \\ -6 \\ -7 \\ -2 \\ -15 \end{bmatrix} \begin{bmatrix} f_{n-2} \\ f_{n-1} \\ f_{n} \end{bmatrix} \]

Derivation of Implicit 3-Point 1-Block Method:

The implicit 3-point 1-block method \((r = 3, k = 1)\) for second order ODEs can be described by the matrix difference equation

\[ A^{(0)} Y_m = A^{(1)} Y_{m-1} + h^2 B^{(0)} F_m + B^{(1)} F_{m-1} \]

with the matrix coefficients specified as

\[ Y_m = \begin{bmatrix} y_{n+1} \\ y_{n+2} \\ y_{n+3} \end{bmatrix}, \quad Y_{m-1} = \begin{bmatrix} y_{n-2} \\ y_{n-1} \\ y_{n} \end{bmatrix}, \quad F_m = \begin{bmatrix} f_{n+1} \\ f_{n+2} \\ f_{n+3} \end{bmatrix}, \quad F_{m-1} = \begin{bmatrix} f_{n-2} \\ f_{n-1} \\ f_{n} \end{bmatrix}, \quad B^{(0)} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}, \quad B^{(1)} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}, \quad B^{(i)} = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{bmatrix} \]

Based on (4), we obtained the following three sets of equations, for the method to be of order at least six.

1st set of equations:
\begin{align*}
b_{11} &= 0, \quad b_{12} = -1, \quad b_{13} = 2, \\
c_{11} + c_{12} + c_{13} + d_{11} + d_{12} + d_{13} &= 1, \\
d_{12} + 2d_{13} + 3c_{11} + 4c_{12} + 5c_{13} &= 2, \\
d_{12} + 4d_{13} + 9c_{11} + 16c_{12} + 25c_{13} &= \frac{25}{6}, \\
d_{12} + 8d_{13} + 27c_{11} + 64c_{12} + 125c_{13} &= 9, \\
d_{12} + 16d_{13} + 81c_{11} + 256c_{12} + 625c_{13} &= \frac{301}{15}, \\
d_{12} + 32d_{13} + 243c_{11} + 1024c_{12} + 3125c_{13} &= 46. \\
\end{align*}

2\textsuperscript{nd} set of equations:

\begin{align*}
b_{21} &= 0, \quad b_{22} = -2, \quad b_{23} = 3, \\
c_{21} + c_{22} + c_{23} + d_{21} + d_{22} + d_{23} &= 3, \\
d_{22} + 2d_{23} + 3c_{21} + 4c_{22} + 5c_{23} &= 7, \\
d_{22} + 4d_{23} + 9c_{21} + 16c_{22} + 25c_{23} &= \frac{35}{2}, \\
d_{22} + 8d_{23} + 27c_{21} + 64c_{22} + 125c_{23} &= \frac{93}{2}, \\
d_{22} + 16d_{23} + 81c_{21} + 256c_{22} + 625c_{23} &= \frac{651}{5}, \\
d_{22} + 32d_{23} + 243c_{21} + 1024c_{22} + 3125c_{23} &= 381. \\
\end{align*}

3\textsuperscript{rd} set of equations:

\begin{align*}
b_{31} &= 0, \quad b_{32} = -3, \quad b_{33} = 4, \\
d_{31} + d_{32} + d_{33} + c_{31} + c_{32} + c_{33} &= 6, \\
d_{32} + 2d_{33} + 3c_{31} + 4c_{32} + 5c_{33} &= 16, \\
d_{32} + 4d_{33} + 9c_{31} + 16d_{32} + 25d_{33} &= 47, \\
d_{32} + 8d_{33} + 27c_{31} + 64c_{32} + 125c_{33} &= 150, \\
d_{32} + 16d_{33} + 81c_{31} + 256c_{32} + 625c_{33} &= \frac{2562}{5}, \\
d_{32} + 32d_{33} + 243c_{31} + 1024c_{32} + 3125c_{33} &= 1848. \\
\end{align*}

Solving the three sets of equations, we obtain the method which can be written as
Explicit and implicit 3-point block methods

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\dot{y}_{n+1} \\
\dot{y}_{n+2} \\
\dot{y}_{n+3} \\
\end{bmatrix} =
\begin{bmatrix}
0 & -1 & 2 \\
0 & -2 & 3 \\
0 & -3 & 4 \\
\end{bmatrix}
\begin{bmatrix}
y_{n-2} \\
y_{n-1} \\
y_{n} \\
\end{bmatrix}
\]

\[+ h^2 \begin{bmatrix}
1 & -1 & 0 & 0 & 0 \\
10 & -240 & 0 & 0 & 0 \\
121 & 11 & -1 & 0 & 0 \\
120 & 120 & 240 & 0 & 0 \\
29 & 127 & 1 & 0 & 0 \\
15 & 120 & 15 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
\dot{f}_{n+1} \\
\dot{f}_{n+2} \\
\dot{f}_{n+3} \\
\end{bmatrix} +
\begin{bmatrix}
1 & 1 & 97 \\
240 & 10 & 120 \\
1 & 47 & 103 \\
120 & 240 & 60 \\
1 & 4 & 161 \\
120 & 15 & 60 \\
\end{bmatrix}
\begin{bmatrix}
f_{n-2} \\
f_{n-1} \\
f_{n} \\
\end{bmatrix}
\]

3. Stability of 3-Point 1-Block Method

One of the practical criteria for a good method to be useful is that it must have region of absolute stability. The most convenient method for finding region of absolute stability is the boundary locus technique. The region \( R \) of the complex \( h - \) plane is defined by the requirement that for all \( h \in R \), where \( h = h^2 \lambda \), all the roots of the stability polynomial \( \pi(r, h) \) have modulus less than one.

When the explicit 3-point 1-block method is applied to the test equation \( y'' = \lambda y \) as follows:

\[
A^{(0)} Y_m = (A^{(1)} + B^{(1)} h^2 \lambda) Y_{m-1}
\]

\[
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
\dot{y}_{n+1} \\
\dot{y}_{n+2} \\
\dot{y}_{n+3} \\
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{12} h^2 \lambda & -1 & \frac{1}{6} h^2 \lambda & 2 + \frac{13}{12} h^2 \lambda \\
\frac{5}{4} h^2 \lambda & -2 & \frac{7}{2} h^2 \lambda & 3 + \frac{21}{4} h^2 \lambda \\
\frac{11}{2} h^2 \lambda & -3 & -15 h^2 \lambda & 4 + \frac{31}{2} h^2 \lambda \\
\end{bmatrix}
\begin{bmatrix}
y_{n-2} \\
y_{n-1} \\
y_{n} \\
\end{bmatrix}
\]

We have the stability polynomial of the method by taking, \( \det[A^{(0)} - (A^{(1)} + B^{(1)} h)] = 0 \), where \( h = h^2 \lambda \). The polynomial can be written as

\[
l^3 + l^2 \left(-\frac{145}{12} \frac{h}{h - 2}\right) + l \left(\frac{79}{4} \frac{h^2}{h} + \frac{37}{6} \frac{h}{h + 1}\right) + \left(-\frac{h^3}{4} + \frac{5}{3} \frac{h^2}{h} - \frac{37}{12} \frac{h}{h}\right) = 0
\]

(7)
Solving (7) for \( \bar{h} = h^2 \lambda \) which gives \( |t| \leq 1 \) we obtained the stability region of the method. This is done by using MATHEMATICA Programming. The stability region is shown in Figure 1 where the region of absolute stability lies inside the boundary.

Similarly, the stability region of the implicit 3-point 1-block method is obtained by tracing the boundary for which the modulus of roots of the stability polynomial of the method does not exceed one. Substituting the test equation into (6) gives

\[
\left( A^{(0)} - h^2 \lambda B^{(0)} \right) Y_m = \left( A^{(1)} + h^2 \lambda B^{(1)} \right) Y_{m-1}
\]

The stability polynomial is obtained by taking

\[
\text{det} \left[ t \left( A^{(0)} - B^{(0)} \bar{h} \right) - \left( A^{(1)} + B^{(1)} \bar{h} \right) \right] = 0
\]

which gives the polynomial

\[
t^3 \left( -\frac{59}{43200} \bar{h}^3 + \frac{11}{360} \bar{h}^2 - \frac{31}{120} \bar{h} + 1 \right) + t^2 \left( -\frac{11957}{21600} \bar{h}^3 - \frac{2153}{480} \bar{h}^2 - \frac{679}{80} \bar{h} - 2 \right) + t \left( \frac{89}{4320} \bar{h}^3 + \frac{1}{15} \bar{h}^2 - \frac{1}{4} \bar{h} + 1 \right) - \frac{1}{21600} \bar{h}^3 + \frac{1}{1440} \bar{h}^2 - \frac{1}{240} \bar{h} = 0
\]

The stability polynomial is solved for \( \bar{h} = h^2 \lambda \) which gives \( |t| \leq 1 \) and the stability region is obtained by tracing the values of \( \bar{h} \). The stability region is shown in Figure 2. Where the stability region lies inside the boundary.
Explicit and implicit 3-point block methods

Stability Region of Explicit 3P1B

Figure 1

Stability Region of Implicit 3P1B

Figure 2
4. Test Problems

To validate the new method, the following problems are solved and the numerical results are compared with existing methods.

Problem 1:
\[ y_1'' = -\frac{y_1}{r}, \quad y_1(0) = 1, \quad y_1'(0) = 0 \]
\[ y_2'' = -\frac{y_2}{r}, \quad y_2(0) = 0, \quad y_2'(0) = 1 \]
with \( r = \sqrt{y_1^2 + y_2^2}, \quad x \in [0, 15\pi] \)

Exact Solution: \( y_1(x) = \cos(x) \quad y_2(x) = \sin(x) \)

Problem 2:
\[ y_1'' = -4x^2y_1 - \frac{2y_2}{r}, \quad y_1(x_0) = 0, \quad y_1'(x_0) = -\sqrt{2\pi} \]
\[ y_2'' = -4x^2y_2 + \frac{2y_1}{r}, \quad y_2(x_0) = 1, \quad y_2'(x_0) = 0 \]
with \( r^2 = y_1^2 + y_2^2, \quad x \in \left[ \sqrt{\frac{\pi}{2}}, 10 \right] \)

Exact solution: \( y_1(x) = \cos(x^2) \quad y_2(x) = \sin(x^2) \)

Problem 3:
\[ z'' + z = 0.001e^z, \quad z(0) = 1, \quad z'(0) = 0.9995i \quad z \in C, \]
\[ z(x) = u(x) + iv(x), \quad u, v \in \mathbb{R}, \]
\[ u(x) = \cos(x) + 0.0005x \sin(x) \]
\[ v(x) = \sin(x) - 0.0005x \cos(x) \]

We choose to solve the equivalent real problems
\[ u'' + u = 0.001 \cos(x) \quad u(0) = 1, \quad u'(0) = 0 \]
\[ v'' + v = 0.001 \sin(x) \quad v(0) = 0, \quad v'(0) = 0.9995 \quad x \in [0, 40\pi] \]
5. Numerical Results

The tables below show the numerical results when Problems 1 to 3 are solved numerically using Fatunla [5], 2-point 1-block (2P1B) method, 2-point third order block method by Omar et. al [9] and the 3-point 1-block (3P1B) method derived in the previous section. To obtain the starting values, we use the well-known fourth-order Runge-Kutta method whereby the second order equations are reduced to system of first order equation. The following notations are used in the tables:

- $h$: Step size used.
- METHOD: Method employed.
- TSTEP: Total number of steps taken to obtain the solution.
- MAXERR: Maximum error.
- E2P1B: Explicit 2-point 1-block method in Fatunla [5].
- E3P1B: Explicit 3-point 1-block method.
- E2PO: Explicit 2-point block method of order three in Omar et. al [9].
- I2P1B: Implicit 2-point 1-block method in Fatunla [5].
- I3P1B: Implicit 3-point 1-block method.
- I2PO: Implicit 2-point block method of order three in Omar et. al [9].

The maximum error is defined as the absolute value of the true solution minus the computed solution.

$$\text{MAXERR} = \max_{1 \leq i \leq TSTEP} \left| y(x_i) - y_i \right|$$
Table 1: Performance Comparison between E2P1B and E3P1B for Solving Problem 1.

<table>
<thead>
<tr>
<th>$h$</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>E2PO</td>
<td>2358</td>
<td>6.80011e(-3)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>2357</td>
<td>1.64890e(-3)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>1573</td>
<td>6.79302e(-4)</td>
</tr>
<tr>
<td>0.005</td>
<td>E2PO</td>
<td>4714</td>
<td>3.32365e(-3)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>4713</td>
<td>4.35356e(-4)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>3143</td>
<td>8.49136e(-5)</td>
</tr>
<tr>
<td>0.001</td>
<td>E2PO</td>
<td>23563</td>
<td>6.65714e(-1)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>23563</td>
<td>1.81544e(-1)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>15710</td>
<td>6.79035e(-7)</td>
</tr>
<tr>
<td>0.0005</td>
<td>E2PO</td>
<td>47125</td>
<td>3.33056e(-1)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>47125</td>
<td>4.56201e(-0)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>31418</td>
<td>8.43528e(-8)</td>
</tr>
</tbody>
</table>

Table 2: Performance Comparison between E2P1B and E3P1B for Solving Problem 2.

<table>
<thead>
<tr>
<th>$h$</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>E2PO</td>
<td>439</td>
<td>1.78118e(-1)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>438</td>
<td>5.84193e(-1)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>293</td>
<td>2.17171e(-1)</td>
</tr>
<tr>
<td>0.005</td>
<td>E2PO</td>
<td>876</td>
<td>2.47420e(-2)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>876</td>
<td>1.43272e(-1)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>585</td>
<td>3.15329e(-2)</td>
</tr>
<tr>
<td>0.001</td>
<td>E2PO</td>
<td>4375</td>
<td>1.09983e(-3)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>4374</td>
<td>5.71844e(-3)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>2917</td>
<td>2.60846e(-4)</td>
</tr>
<tr>
<td>0.0005</td>
<td>E2PO</td>
<td>8748</td>
<td>5.49964e(-4)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>8748</td>
<td>1.42972e(-3)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>5833</td>
<td>3.26213e(-5)</td>
</tr>
</tbody>
</table>
Table 3: Performance Comparison between E2P1B and E3P1B for Solving Problem 3.

<table>
<thead>
<tr>
<th>h</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>E2PO</td>
<td>6285</td>
<td>5.00010(-3)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>6284</td>
<td>3.65778(-3)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>4191</td>
<td>1.14794(-4)</td>
</tr>
<tr>
<td>0.005</td>
<td>E2PO</td>
<td>12568</td>
<td>2.49800(-3)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>12567</td>
<td>9.14567(-4)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>8379</td>
<td>1.43600(-5)</td>
</tr>
<tr>
<td>0.001</td>
<td>E2PO</td>
<td>62833</td>
<td>4.99503(-4)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>62833</td>
<td>3.65869(-5)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>41890</td>
<td>1.14912(-7)</td>
</tr>
<tr>
<td>0.0005</td>
<td>E2PO</td>
<td>125665</td>
<td>2.49750(-4)</td>
</tr>
<tr>
<td></td>
<td>E2P1B</td>
<td>125665</td>
<td>9.14662(-6)</td>
</tr>
<tr>
<td></td>
<td>E3P1B</td>
<td>83778</td>
<td>1.42808(-8)</td>
</tr>
</tbody>
</table>

Table 4: Performance Comparison between I2P1B and I3P1B for Solving Problem 1.

<table>
<thead>
<tr>
<th>h</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>I2PO</td>
<td>2358</td>
<td>1.61702(-3)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>2357</td>
<td>3.23212(-8)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>1573</td>
<td>2.14918(-8)</td>
</tr>
<tr>
<td>0.005</td>
<td>I2PO</td>
<td>4714</td>
<td>8.15183(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>4713</td>
<td>2.15792(-9)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>3143</td>
<td>1.34949(-9)</td>
</tr>
<tr>
<td>0.001</td>
<td>I2PO</td>
<td>23563</td>
<td>1.65754(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>23563</td>
<td>1.22006(-10)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>15710</td>
<td>8.64701(-11)</td>
</tr>
<tr>
<td>0.0005</td>
<td>I2PO</td>
<td>47125</td>
<td>8.30992(-5)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>47125</td>
<td>6.19152(-10)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>31418</td>
<td>3.95277(-10)</td>
</tr>
</tbody>
</table>
Table 5: Performance Comparison between I2P1B and I3P1B for Solving Problem 2.

<table>
<thead>
<tr>
<th>$h$</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>I2PO</td>
<td>439</td>
<td>2.70602(-3)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>438</td>
<td>2.21752(-3)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>293</td>
<td>2.61036(-3)</td>
</tr>
<tr>
<td>0.005</td>
<td>I2PO</td>
<td>876</td>
<td>1.36406(-3)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>876</td>
<td>1.39189(-4)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>585</td>
<td>8.03535(-5)</td>
</tr>
<tr>
<td>0.001</td>
<td>I2PO</td>
<td>4375</td>
<td>2.74575(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>4374</td>
<td>2.22967(-7)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>2917</td>
<td>1.22110(-7)</td>
</tr>
<tr>
<td>0.0005</td>
<td>I2PO</td>
<td>8748</td>
<td>1.37397(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>8748</td>
<td>1.41201(-8)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>5833</td>
<td>2.34300(-8)</td>
</tr>
</tbody>
</table>

Table 6: Performance Comparison between I2P1B and I3P1B for Solving Problem 3.

<table>
<thead>
<tr>
<th>$h$</th>
<th>METHOD</th>
<th>TSTEP</th>
<th>MAXERR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>I2PO</td>
<td>6285</td>
<td>1.24874(-3)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>6284</td>
<td>5.39496(-8)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>4191</td>
<td>4.10154(-9)</td>
</tr>
<tr>
<td>0.005</td>
<td>I2PO</td>
<td>12568</td>
<td>6.24373(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>12567</td>
<td>3.40287(-9)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>8379</td>
<td>2.23512(-9)</td>
</tr>
<tr>
<td>0.001</td>
<td>I2PO</td>
<td>62833</td>
<td>1.24875(-4)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>62833</td>
<td>1.44249(-10)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>41890</td>
<td>5.04009(-11)</td>
</tr>
<tr>
<td>0.0005</td>
<td>I2PO</td>
<td>125665</td>
<td>6.24376(-5)</td>
</tr>
<tr>
<td></td>
<td>I2P1B</td>
<td>125665</td>
<td>2.32200(-10)</td>
</tr>
<tr>
<td></td>
<td>I3P1B</td>
<td>83778</td>
<td>2.31953(-10)</td>
</tr>
</tbody>
</table>
6. Discussions and conclusion

The two measured parameters in the previous section, namely the total number of steps and accuracy will be used to illustrate the effectiveness of the method.

The results demonstrate the advantage of using both the explicit and implicit 3-point 1-block method over the explicit and implicit 2-point 1-block methods by Fatunla [5] and Omar et.al [9] in terms of total number of steps taken. In all the problems, it is observed that the number of steps taken by the 3-point 1 block methods are always less than the number of steps taken by the existing methods. These results are expected since the new methods calculate the values of $y$ at three points simultaneously compared to two points at a time for the existing methods. From the tables the new methods reduced the number of steps to almost two third compared to the existing methods.

Comparing the explicit and the implicit methods, it is observed that the implicit methods are more accurate than the explicit counterparts this is true for both the new and the existing ones.

From the magnitude of the maximum error we can conclude that the new explicit method is more accurate compared to the existing explicit methods. From the numerical results we can say that the new implicit method is more accurate than the I2P1B method and just as accurate as I2PO method. Thus, it can be concluded that the numerical results suggest a significant improvement in the efficiency of the new methods.

References


Received: June 8, 2008