MATEMATIKA, 2015, Volume 31, Number 1, 47–57

Behaviour of the Extrapolated Implicit Midpoint and Implicit Trapezoidal Rules With and Without Compensated Summation

[1]Amira Ismail and [2]Annie Gorgey
[1,2]Department of Mathematics
Universiti Pendidikan Sultan Idris
35900 Tanjung Malim, Perak, Malaysia
e-mail: [1]amira160790@gmail.com, [2]annie_gorgey@fsmt.upsi.edu.my

Abstract Extrapolation involves taking a certain linear combination of the numerical solutions of a base method applied with different stepsizes to obtain greater accuracy. This linear combination is done so as to eliminate the leading error term. The technique of extrapolation in accelerating convergence has been successfully in numerical solution of ordinary differential equations. In this study, symmetric Runge-Kutta methods for solving linear and nonlinear stiff problem are considered. Symmetric methods admit asymptotic error expansion in even powers of the stepsize and are therefore of special interest because successive extrapolations can increase the order by two at time. Although extrapolation can give greater accuracy, due to the stepsize chosen, the numerical approximations are often destroy due to the accumulated round off errors. Therefore, it is important to control the rounding errors especially when applying extrapolation. One way to minimize round off errors is by applying compensated summation. In this paper, the numerical results are given for the symmetric Runge-Kutta methods which are the implicit midpoint rule and the implicit trapezoidal rule applied with and without compensated summation. The result shows that symmetric methods with higher level extrapolation using compensated summation give much smaller errors. On the other hand, symmetric methods without compensated summation when applied with extrapolation, the errors are affected badly by rounding errors.

Keywords Runge-Kutta methods; symmetric methods; compensated summation; extrapolation.

2010 Mathematics Subject Classification 65L04, 65L05

1 Introduction

Consider a system of an ordinary differential equation with the initial value,

$$y' = f(x, y), \quad y(x_0) = y_0, \tag{1}$$

Any Runge-Kutta (RK) methods can be used to solve problem (1). Runge-Kutta method is defined by formulas (2a) and (2b)

$$Y_i = y_{n-1} + h \sum_{j=1}^{s} a_{ij} f\left(x_{n-1} + c_j h, Y_j\right), i = 1, 2, \ldots, s \tag{2a}$$

$$y_n = y_{n-1} + h \sum_{j=1}^{s} b_j f(x_{n-1} + c_j h, Y_j), j = 1, 2, \ldots, s, \tag{2b}$$

where Y_i represent the internal stage values and y_n represent the update of y at the n^{th} step. Examples of order-2 RK methods are the implicit midpoint rule (IMR) and implicit trapezoidal rule (ITR) as given in Table 1.

Table 1: Order-2 of IMR and ITR

IMR:	ITR:
$Y = y_{n-1} + \frac{h}{2} f\left(x_{n-1} + \frac{h}{2}, Y\right)$,	$Y_1 = y_{n-1}$,
$y_n = y_{n-1} + hf(x_{n-1} + \frac{h}{2}, Y)$.	$Y_2 = y_{n-1} + \frac{h}{2} hf\big(x_{n-1}, Y_1\big) + \frac{h}{2} f\big(x_{n-1} + h, Y_2\big)$,
	$y_n = Y_2$.

Both IMR and ITR are also symmetric, [1]. The symmetric methods are special type of RK methods because when they are applied with extrapolation technique this makes the order of the method increase by two at a time, [2–4].

2 Extrapolation Technique

Extrapolation which is founded by L. F. Richardson, [5] is a technique to increase the stability and efficiency of a method. This technique is also known as Richardson extrapolation. Extrapolation can be applied in two ways. Active extrapolation occurs when the value of extrapolation is used to propagate the next computation. If the extrapolated values are not used in any subsequent computations then it is called passive extrapolation. Since extrapolation can increase accuracy and efficiency, many researchers are still finding the best ways to apply extrapolation. Gorgey [6] showed that passive extrapolation of the 2-stage Gauss method is more efficient than the active extrapolation for linear problems. In terms of efficiencies, Faragó and Zlatev, [7] found out that computing time spent with the Richardson extrapolation for both active and passive is more than ten times smaller than the corresponding computing time for the backward Euler formula. Hence they concluded that extrapolation is a powerful tool for increasing the accuracy and efficient with regard to the computational cost especially when the accuracy requirement is not extremely low. Besides that, Faragó and Zlatev, [7] also studied the convergence of diagonally implicit RK methods combined with Richardson extrapolation.The extrapolation results in a convergent numerical method if the initial value problem satisfied the Lipschitz condition. In addition to that, Zlatev and Dimov, [8] studied on the absolute stability properties of the Richardson extrapolation by the explicit RK methods of order-1 to order-4. They concluded that the passive extrapolation may fail when the method is not stable for large stepsizes in solving certain problems but active extrapolation works fine for larger stepsizes although the method is not stable.

The extrapolation formula is defined as

$$T_{ij} = T_{i,j-1} + \frac{T_{i,j-1} - T_{i-1,j-1}}{\left(\frac{m_i}{m_{i-j+1}}\right) - 1}, \tag{3}$$

where $i = j = 2, \ldots, n$. For this paper, stepsize sequence is chosen to be $m = 1, 2, 4, 8$. Higher level extrapolation is also possible. The higher the levels of extrapolations, the more accurate the method will be since on every level of extrapolations, the order p of the symmetric method will increase by 2 at a time. This can be summarize in Table 2.

Table 2: Extrapolation Table

$\frac{H}{m_1}$	$T_{1,1}$					
$\frac{H}{m_2}$	$T_{2,1}$	$T_{2,2}$				
$\frac{H}{m_3}$	$T_{3,1}$	$T_{3,2}$	$T_{3,3}$			
$\frac{H}{m_4}$	$T_{4,1}$	$T_{4,2}$	$T_{4,3}$	$T_{4,4}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$\frac{H}{m_n}$	$T_{n,1}$	$T_{n,2}$	$T_{n,3}$	$T_{n,4}$	$\ldots$	$T_{n,n}$
order	p	$p+2$	$p+4$	$p+6$	$\ldots$	$p+2(n+1)$
level		1^{st}	2^{nd}	3^{rd}	$\ldots$	n^{th}

where in Table 2 the meaning of the symbols are as follows:

$$\begin{aligned}
p\ &= \text{order of the methods,}\\
n\ &= \text{positive integer,}\\
H\ &= \text{the stepsize at length } n \text{ where } h_i = \tfrac{H}{m_i}, \quad i = 1, 2, \ldots, n,\\
m_i\ &= \text{increasing sequence,} \quad i = 1, 2, \ldots, n,\\
T_{i,1} &= \text{approximations using stepsize } h_i = \tfrac{H}{m_i}.
\end{aligned}$$

In this paper, up to third level extrapolation is considered.

3 Compensated Summation

Iterated methods or usually called numerical methods such as RK methods, general linear methods and multistep methods are always applied in solving linear and nonlinear problems. Higher order methods such as Radau IIA of order-5 or 3-stage Gauss method of order-6 are always preferable since these methods are higher order and therefore will give greater accuracy than lower order methods. However, instead of using higher order methods, one way to get greater accuracy is by applying extrapolation. Lower order methods require smaller stepsize than higher order methods. If the stepsize is chosen to be very small then this can lead to round-off error where eventually will destroy the solutions. Round-off error is an error created due to approximate representation of number, [1]. One way to solve this problem is by not using a small stepsize. However this cannot be applied to certain problems when relatively smaller stepsize is required. Therefore the other way to solve this rounding error is by applying a technique known as compensated summation. Compensated summation is a technique used to minimize the round-off error. Compensated summation when applied with any numerical methods can improve the figure of accuracy up to certain smaller stepsizes. Higham [9], gave a good study of compensated summation. He continued the work by Stetter, [10] who found sufficient conditions for the global error to be linear in the tolerance. He also studied on standard error control method of explicit RK methods. He showed that by ignoring higher-order terms, the global error in the numerical solution behaves like a known rational power of the error tolerance.

Compensated summation is important to get better result especially when extrapolation is applied. This is because of the reduction of the stepsizes for chosen stepsize sequences. Most often, instability is not caused by the accumulation of millions of rounding errors, but by the insidious growth of just a few rounding errors, [9]. Hence, compensated summation is a way to minimize the accumulation of error. The compensated summation captures the round-off error at each step and the quantity is where round-off error is stored for y-values. Compensated summation works when the quantity to be summed is small compared with the total being added to estimates the value of which is the small quantity that is added in any particular step. These values then add to the value. In each step, this small value is captured and stored.

The two Matlab codes given below show the implementation of the compensated summation by the IMR for the PR problem (Problem 1) where $x_0 = x$ and $y_0 = y$. Table 3 show the algorithm for a simple version of IMR without compensated summation while Table 4 show the algorithm for the modified version of IMR with compensated summation.

The implementation of the IMR as given in Table 1 is given as suggested by Hairer and Wanner in, [11]. Since IMR is an implicit method, full Newton method (DY, see Table 3) has been considered to solve the nonlinear terms involving Y and y. It is advisable not to evaluate the function f many times. In order to avoid re-evaluating f many times, consider rewriting the update y_n for the IMR given in Table 1 as follows:

$$\begin{aligned} Y &= y_{n-1} + \frac{h}{2} f\left(x_{n-1} + \tfrac{h}{2}, Y\right), \\ 2Y &= 2y_{n-1} + hf\left(x_{n-1} + \tfrac{h}{2}, Y\right), \\ hf\left(x_{n-1} + \tfrac{h}{2}, Y\right) &= 2Y - 2y_{n-1}. \end{aligned}$$

Therefore, y_n as defined in Table 3 is given by

$$\begin{aligned} y_n &= y_{n-1} + 2Y - 2y_{n-1}, \\ y_n &= 2Y - y_{n-1}. \end{aligned}$$

Full Newton requires evaluating the Jacobian at each iteration whereas simplified Newton, requires evaluating Jacobian only on the first approximation and may be re-evaluated later. Evaluating the Jacobian at each iteration is wasteful and expensive because of the need to update and LU-factorize of the Jacobian matrix. However, this research is not focusing on the cheaper implementation cost. The purpose of this research is to show that the IMR using compensated summation gives improve result than without compensated summation. To study more about the implementation using simplified Newton, one can refer to Gorgey, [6].

Algorithm on the implementation of the IMR written in Matlab2014 is given in Table 3.

Algorithm for a simple version of IMR

In Table 4, the only modification done is by introducing the terms s, sx and z where $z = Y - y$. It is suggested in, [11] as well as in, [1] that the influence of round-off errors can

Table 3: Algorithm for a Simple Version of IMR without Compensated Summation

```
n = 10;
tol = 1.e-10;
m = length(y);
I = eye(m);
Y = y;

for i = 1:n
    term1 = 0.5*h*f(x+0.5*h,Y);
    term2 = 0.5*h*J(x+0.5*h,Y);
    G = Y - y - term1;
    DY = (I -term2 )\(-G);

    if norm(DY,inf) < tol*max(1,norm(Y,inf))
        break;
    end
    Y =  Y + DY;
end

if i >= n
    disp('nonconverging')
end

xn = x + h;
yn = 2*Y - y;
```

be reduced by using smaller quantities z. The aim of compensated summation is to capture the round-off error in each individual step. The s quantity is where the round-off error is stored for the y-values and s_x is the round-off error stored for the x-values. Instead of solving for $Y = y + \frac{h}{2} f(x_{n-1} + \frac{h}{2}, Y)$ as for the IMR given in Table 1, use $z = \frac{h}{2} f(x_{n-1} + \frac{h}{2}, y + z)$ and solve by Newton. To apply the compensated summation, let $s = 0, s_x = 0$.

Algorithm modified version of IMR

For example, consider applying IMR and ITR to the PR problem (Problem 1) with $q = -10^2$ and $h = 0.001$.

Figure 1 shows the effect of round-off error by the IMR and ITR in solving Problem 1. Both IMR and ITR give better results with compensated summation than without compensated summation. When $h = 0.00025$, IMR without compensated summation cannot give a straight line because of the accumulation of the round-off error (see Figure 1). However, it is shown that when IMR is applied with compensated summation the graph gives much improved result. A similar result is shown for ITR.

This paper is mainly about compensated summation applied to extrapolation of the second order implicit RK methods which are the IMR and ITR methods. The objective of

Table 4: Algorithm for Modified Version of IMR with Compensated Summation

```
z = 0*y;

for i = 1:n
    term1 = 0.5*h*f(x+0.5*h,z+y)
    term2 = 0.5*h*J(x+0.5*h,z+y)
    G  = z - term1;
    DG = Im - term2;
    DY = DG\(-G);
    if norm(DY,inf) < tol*max(1,norm(z,inf))
        break;
    end
    z =  z + DY;
end

if i >= n
    disp('nonconverging')
end

increment =  2*z + s;
yn = y + increment;
sn = increment - (yn - y);
incx = h + sx;
xn = x + incx;
sxn = incx - (xn - x);
```

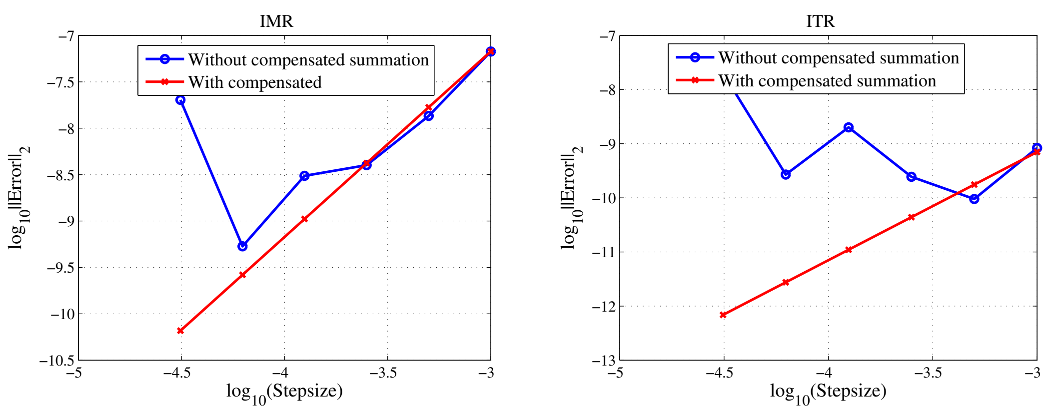

Figure 1: The Effect of Round-off Errors by the IMR and ITR, Respectively With and Without Compensated Summation in Solving Problem 1

this research is to show that although extrapolation requires reduction of stepsizes according to the levels, the accuracy of the errors can still be improved with compensated summation.

4 Numerical Results

Numerical results are given for linear problem (Problem 1) and nonlinear problem (Problem 2). These problems are chosen as the test problems since the exact solutions are known. On each problem, the results are tested in terms of accuracy (order). The order of a method is given by the gradient of the graph of error versus stepsize. The accuracy graph shows the comparison of extrapolated IMR and ITR with and without compensated summation.

Problem 1: Prothero Robinson Problem

The Prothero Robinson problem is given by

$$y'(x) = q(y(x) - g(x)) + g'(x), y(x_0) = 1,$$

where $g(x) = \sin(x)$ and $q = -10^3$ and the $y(x) = \sin(x)$.

The numerical results for Problem 1 are given in Figure 2 to Figure 4. Figure 2 shows the accuracy of the first level extrapolated IMR and ITR respectively while Figure 3 shows the accuracy of the second level extrapolated IMR and ITR respectively and lastly Figure 4 shows the accuracy of the third level extrapolated IMR and ITR respectively. Result shows that first level extrapolation of IMR and ITR (order 4) with compensated summation is more accurate. Second level extrapolated IMR and ITR (order 6) also gives better accuracy with compensated summation. Although the results with compensated summation is much better than the results without compensated summation as seen in all figures, there are up to certain extend where the round-off errors cannot be fixed even with compensated summation. This is due to the accumulation of the round-off errors.

Problem 2: Kaps Problem

The Kaps problem is given by

$$\begin{aligned} y_1' &= (q-2)y_1 - qy_2^2, \quad & y_1(x) &= e^{-2x}, \\ y_2' &= y_1 - y_2 - y_2^2, \quad & y_2(x) &= e^{-x}, \end{aligned}$$

with $y_1(0) = 1$ and $y_2(0) = 1$ and $q = -10^3$.

The numerical results for Problem 2 are given in Figure 5 to Figure 7. Figure 5 shows the accuracy of the first level extrapolated IMR and ITR respectively in solving Kaps problem (Problem 2) also with $q = -10^3$. Figure 6 shows the accuracy of the second level extrapolated IMR and ITR respectively while Figure 7 shows the accuracy of the third level extrapolated IMR and ITR respectively in solving Kaps problem (Problem 2) also with $q = -10^3$. In all these figures, it is showed that extrapolation with compensated summation is indeed more accurate than extrapolation without compensated summation.

Therefore based on these numerical results, it is proven that compensated summation works for linear and nonlinear problems up to third level extrapolations. Fourth level

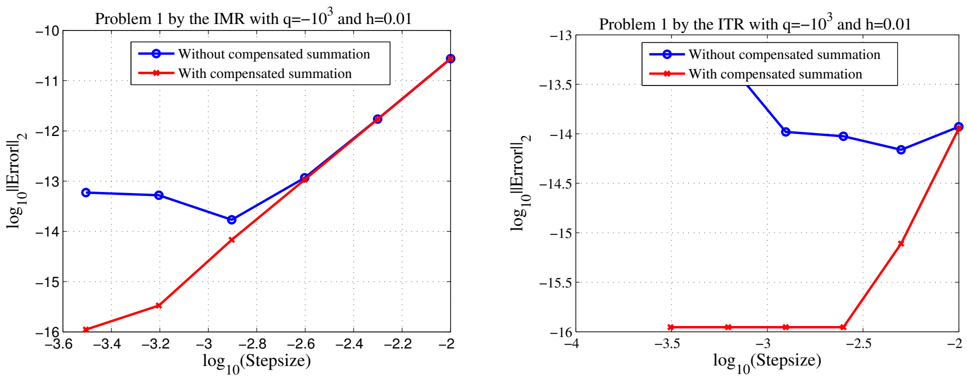

Figure 2: The Accuracy of the First Level Extrapolated IMR and ITR

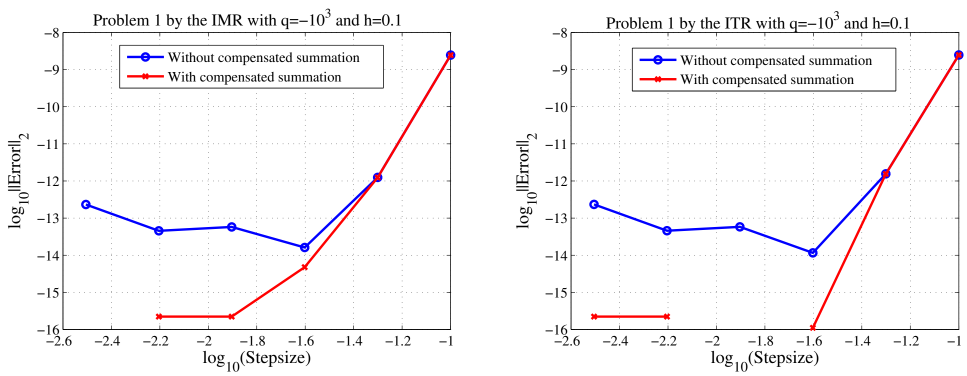

Figure 3: The Accuracy of the Second Level Extrapolated IMR and ITR

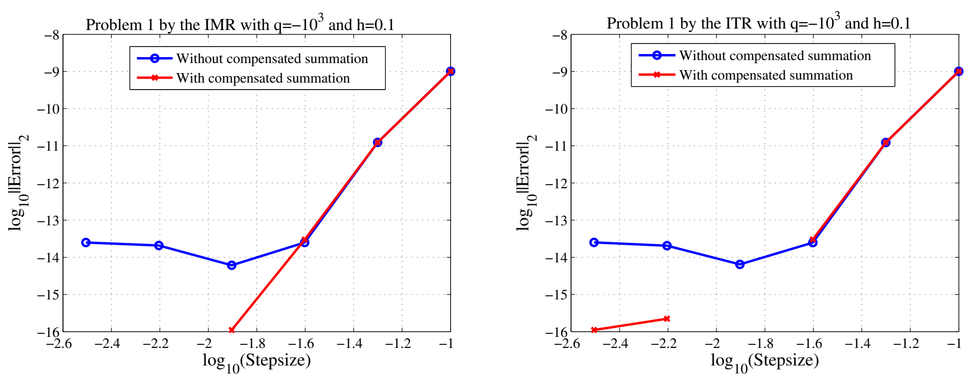

Figure 4: The Accuracy of the Third Level Extrapolated IMR and ITR

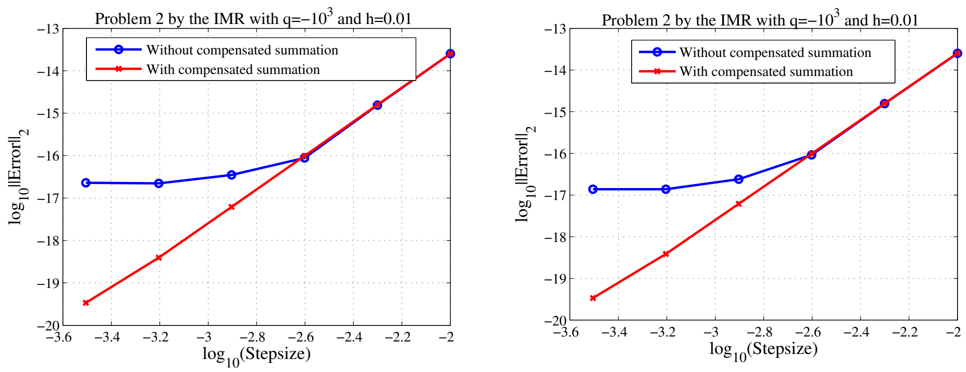

Figure 5: The Accuracy of the First Level Extrapolated IMR and ITR

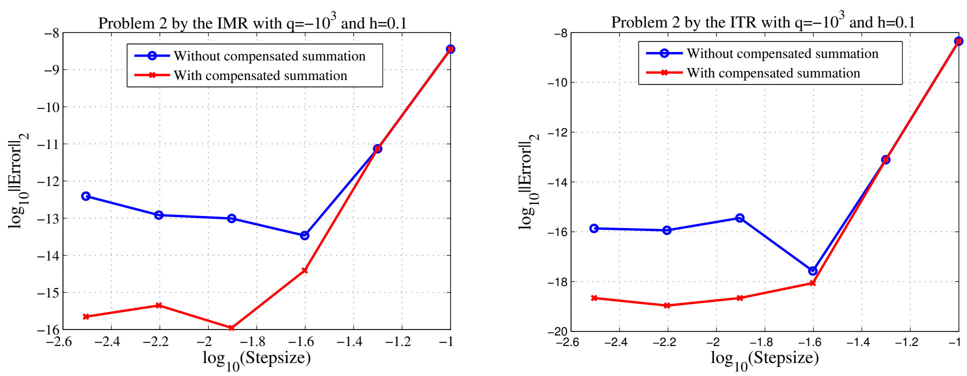

Figure 6: The Accuracy of the Second Level Extrapolated IMR and ITR

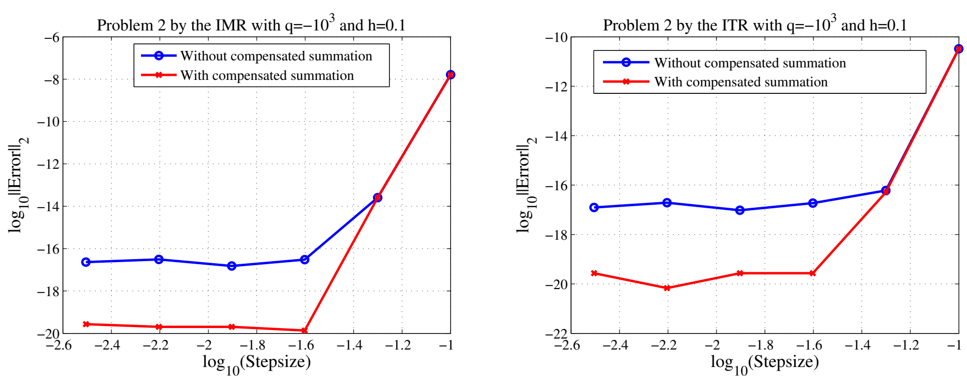

Figure 7: The Accuracy of the Third Level Extrapolated IMR and ITR

extrapolation is not done in this research because although the third level extrapolation by the IMR and ITR are improved using compensated summation as shown in Figure 7, in order to carry out fourth level extrapolation a very small stepsize need to be used and this will end up with the accumulations of the round of errors.

5 Conclusion

Based on the numerical results for both problems, extrapolations of IMR and ITR with compensated summation give better accuracy than without compensated summation. The results also showed that if compensated summation is not applied to any levels of extrapolation then the results obtained is worse. Therefore, whenever small stepsize is chosen then compensated summation technique is the best way to minimize the round-off errors.

Acknowledgments

This research was supported by the research grant RAGS (Vote No: 2013-0004-107-72).

References

[1] J . C. Butcher. *Runge-Kutta Methods for Ordinary Differential Equations.* New Zealand: John Wiley and Sons. Ltd. 2008.

[2] Chan, R. P. K. Generalized symmetric runge-kutta methods. *Computing.* 1993. 50(1): 31–49.

[3] Chan, R. P. K. and Gorgey, A. Active and passive symmetrization of runge-kutta gauss methods. *Applied Numerical Mathematics.* 2013. 67: 64–77.

[4] Kulikov, G. Y. Proportional extrapolation and symmetric one-step methods. *Russian Journal of Numerical Analysis and Mathematical Modelling.* 2011. 26: 49–73.

[5] Richardson, L. F. The approximate arithmetical solution by finite differences of physical problems involving differential equation, with an application to the stresses in a masonry dam. *Philosophy Transactions of the Royal Society London.* 1911. 210: 307–357.

[6] A. Gorgey. *Extrapolation of symmetrized Runge-Kutta methods.* University of Auckland: Ph.D. Thesis. 2012.

[7] Faragó, I. A. H. and Zlatev, Z. The convergence of diagonally implicit rungekutta methods combined with richardson extrapolation. *Computers and Mathematics with Applications.* 2013. 65: 395–401.

[8] Zlatev, Z. and Dimov, I. Studying absolute stability properties of the richardson extrapolation combined with explicit rungekutta methods. *Computers and Mathematics with Applications.* 2014. 67: 22942307.

[9] Higham, D. J. Studying absolute stability properties of the richardson extrapolation combined with explicit rungekutta methods. *Computers and Mathematics with Applications.* 1993. 45: 227–236.

[10] Stetter, H. J. Tolerance proportionality in ode-codes. *Proceeding of Second Conference on Numerical Treatment of Ordinary Differential Equations*. 1980. 32: 109–123.

[11] E. Hairer and G. Wanner. *Solving ordinary differential equations, II. Stiff and differential-algebraic problems*. Berlin: pringer Series in Computational Mathematics. 1991.