American Journal of Sciences and Engineering Research wwww.iarjournals.com

American Journal of Sciences and Engineering Research

E-ISSN -2348 – 703X, Volume 5, Issue 4, 2022

On the Stability Analysis and Implementation of a New fully Implicit Third-Stage Fourth-Order Runge-Kutta Method

Esekhaigbe Aigbedion Christopher

Department of Mathematics, Aduvie Pre-University College, Jahi District, Nigeria

ABTRACT: *The object of this paper is to analyze the stability and implement a newly derived implicit third-stage fourth-order Runge-Kutta method. Efforts will be made to carry out a comparative analysis with an existing method. The analysis revealed that the method is A- and L-stable. The implementation on initial-value problems revealed that the method performed better than the existing method. The implementation was carried out by MAPLE program and the stability curve plotted using MATLAB codes.*

Keywords: *Stability, Implicit, Runge-Kutta Methods, Linear and non- linear equations, Taylor series, Parameters, Initial-value Problems, Implementation, Curve.*

I. INTRODUCTION

Runge-Kutta methods are numerical (one-step) methods for solving initial-value problems in Ordinary Differential Equations. Initial-value problems are first-order Differential Equations that are applicable to real life problems such as growth and decay problems, falling bodies problems, temperature problems, problems in chemical engineering, control theory e.t.c. We shall be focusing on stiff initial-value problems because over the years, this has shown much concern in modern day research.

For many classes of numerical methods, A-stability and even L-stability imply implicitness. According to Yahaya (2014), "explicit Runge-Kutta methods have rather small regions of absolute stability, even those designed to have extended regions of stability are inadequate unless the system is only very mildly stiff". On the other hand it is rather easier to find A-stable implicit Runge-Kutta methods than to find A-stable implicit linear multistep methods. For example, Butcher (1987, 1988, 1997, 2000, 2003, 2008, 2009, 2010) has shown that Ehle's R-stage implicit Runge-Kutta methods of order$s\ 2R-1\ and\ 2R-2$, having Radau and Lobatto's quadratures respectively are all A-stable; thus there exist A-stable methods of this type of arbitrarily high order. L-stable implicit Runge-Kutta methods are also possible. Several authors like Yakubu (2010), Agam and Yahaya (2014), and Lambert (1992) have considered A- or L-stable implicit Runge-Kutta methods associated with various types of quadratures. Also, Van Der Houwen and Sommeijer (2015) worked on Runge-Kutta projection methods with low dispersion and dissipation errors. However, all such methods suffer a serious practical disadvantage in that the solution of the implicit non-linear equations at each step is considerably harder to achieve in the case of implicit Runge-Kutta than in the case of implicit linear multistep methods. If we considered R-stage fully implicit Runge-Kutta methods applied to an m-dimensional stiff system, then it is clear that the $k_r, r = 1,2, \dots \dots, R$ are also m-vectors. It follows that at each step we have to solve a system of mR simultaneous non-linear equations by some form of Newton iteration - and this will converge only if we

 Received-03-03-2022, Accepted- 30-04-2022

can find a suitably accurate initial iterate. In special situations implicit methods may be competitive; much will depend on the structure of the implicit equations arising from a particular problem.
If the Runge-Kutta method is semi-explicit, then the mR simultaneous equations split into R distinct sets of equations, each set containing m equations. The class of semi-explicit methods developed by Butcher (1963, 1964, 1975) as quoted in the work of Lambert (1977), are not, however, A-stable because they do not satisfy the conditions for A-stability.
It is clear from the above discussion that our troubles with stiff systems are almost over when we find an A- or L-stable method; but the real test is the efficiency with which we can handle the resultant implicitness.

II. METHODS OF DERIVATION

i From the general R – Stage Implicit Runge–Kutta Method, get$K_r, r = 1,2,3 \ldots.. R$, Obtain the Taylor series expansion of the $K_r, r = 1,2,3 \ldots.. R$,
ii $Since\ k_1, k_2\ and\ k_3\ are\ implicit, we\ cannot\ proceed\ by\ successive\ substitution$
$as\ in\ the\ case\ of\ explicit. we\ assume\ that\ the\ solutions\ for\ k_1, k_2, k_3\ may$
$be\ expressed\ in\ the\ form: K_r = \alpha_r + h\beta_r + h^2\theta_r + h^3\emptyset_r + O(h^4), r = 1,2,3$
iii.$Puting\ the\ k_1,\ k_2\ and\ k_3\ from\ (ii)\ into\ k_r\ expanded\ by\ Taylor's\ series\ above$
$from\ (i)$, equate coefficients in power of h to get α_r,β_r,θ_r and $\emptyset_r$, $r = 1,2,3 \ldots.. R$.
iv Also, $Put\ the\ k_1,\ k_2\ and\ k_3\ \ from\ (ii)\ into\ \emptyset(x, y, h) = \sum_{r=1}^{3} c_r k_r = c_1k_1 + c_2k_2 + c_3k_3$ and arrange in power of h.
v Equate $\emptyset(x, y, h) from$ (iv) above and compare with Taylor's series of the form:
$\emptyset_T(x, y, h) = f + \frac{h}{2!}(f_x + ff_y) + \frac{h^2}{3!}(f_{xx} + 2ff_{xy} + f_xf_y + ff_y^2 + f^2f_{yy}) + \frac{h^3}{4!}(f_{xxx} + 3ff_{xxy} + 3f^2f_{xyy} + 3f_xf_{xy} + 5ff_yf_{xy} + 3ff_xf_{yy} + f_{xx}f_y + 4f^2f_yf_{yy} + f_xf_y^2 + ff_y^3 + f^3f_{yyy})$.
vi As a result, a set of linear/ non linear equations will be generated.
vii Solve the set of equations to derive a new fourth-order third-stage fully implicit Runge-Kutta formula.

III. DERIVATION OF THE FOURTH-ORDER THIRD-STAGE FULLY IMPLICIT RUNGE-KUTTA METHOD

From Lambert (1977), the general R – Stage Implicit Runge – Kutta Method is
$y_{n+1} - y_n = h\,\emptyset\,(x_n, y_n, h),$
$\emptyset(x, y, h) = \sum_{r=1}^{R} C_rK_r$
$K_r = f(x + ha_r, y + h\sum_{s=1}^{R} b_{rs}\,k_s), r = 1,2,3 \ldots.. R,$
$a_r = \sum_{s=1}^{R} b_{rs},\ r = 1,2,3 \ldots.. R,$ (3.1)
$For\ a\ third - stage, R = 3, \quad K_r\ becomes:$
$K_r = f(x + ha_r, y + h\,(b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2)$ (3.2)
$Expanding\ (3.2)\ with\ Taylor\ series\ about\ the\ point\ (x, y), we\ obtain:$
$K_r = \sum_{p=0}^{\infty}\frac{1}{p!}(ha_r\frac{\partial}{\partial x} + h\,(b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2)\ \frac{\partial}{\partial y})^P\ f(x, y)$
$K_r = f + h\left[a_rf_x + (b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2 + b_{r3}k_3)f_y\right] + \frac{1}{2!}h^2[a_r^2f_{xx} +$
$2a_r(b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2 + b_{r3}k_3)f_{xy} + (b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2 + b_{r3}k_3)^2f_{yy}]$
$+\frac{1}{3!}h^3[a_r^3f_{xxx} + 3a_r(b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2 + b_{r3}k_3)f_{xxy} + 3a_r(b_{r1}k_1 + b_{r2}k_2 +$
$b_{r2}k_2 + b_{r3}k_3)^2f_{xyy} + (b_{r1}k_1 + b_{r2}k_2 + b_{r2}k_2 + b_{r3}k_3)^3f_{yyy}] + 0(h^2), r = 1,2,3$ (3.3)
$Since\ k_1, k_2\ and\ k_3\ are\ implicit, we\ cannot\ proceed\ by\ successive\ substitution$
$as\ in\ the\ case\ of\ explicit. we\ assume\ that\ the\ solutions\ for\ k_1, k_2, k_3\ may$
$be\ expressed\ in\ the\ form:$
$K_r = \alpha_r + h\beta_r + h^2\theta_r + h^3\emptyset_r + O(h^4), r = 1,2,3$ (3.4)
$Puting\ the\ k_1,\ k_2\ and\ k_3\ into\ k_r, we\ have:$
$K_r = f + ha_rf_x + hbr_1(\alpha_1 + h\beta_1 + h^2\theta_1)f_y + hb_{r2}(\alpha_2 + h\beta_2 + h^2\theta_2)f_y +$
$hb_{r3}(\alpha_3 + h\beta_3 + h^2\theta_3)f_y + \frac{h^2}{2!}a_r^2f_{xx} + h^2arb_{r1}(\alpha_1 + h\beta_1)f_{xy} +$

$h^2 arb_{r2}(\alpha_2+h\beta_2)f_{xy} + h^2 arb_{r3}(\alpha_3+h\beta_3)f_{xy} + \frac{h^2}{2!}b_{r1}^2(\alpha_1 + 2h\alpha_1\beta_1)f_{yy} +$
$h^2 br_1 br_2(\alpha_1\alpha_2+h\alpha_1\beta_2+h\beta_1\alpha_2)f_{yy} + h^2 arb_{r3}(\alpha_1\alpha_3+h\alpha_1\beta_3+h\beta_1\alpha_3)f_{yy} +$
$\frac{h^2}{2!}b_{r2}^2(\alpha_2^2+2h\alpha_2\beta_2)f_{yy} + h^2 br_2 br_3(\alpha_2\alpha_3+h\alpha_2\beta_3+h\beta_2\alpha_3)f_{yy} +$
$\frac{h^2}{2!}b_{r3}^2(\alpha_3^2+2h\alpha_3\beta_3)f_{yy} + \frac{h^3}{3!}a_r^3 f_{xxx} + \frac{h^3}{2!}a_r^2 b_{r1}\alpha_1 f_{xxy} + \frac{h^3}{2!}a_r^2 b_{r2}\alpha_2 f_{xxy} +$
$\frac{h^3}{2!}a_r^2 b_{r3}\alpha_3 f_{xxy} + \frac{h^3}{2!}arb_{r1}^2\alpha_1^2 f_{xyy} + h^3 arb_{r1}b_{r2}\alpha_1\alpha_2 f_{xyy} +$
$h^3 arb_{r1}b_{r3}\alpha_1\alpha_3 f_{xyy} + \frac{h^3}{2!}arb_{r2}^2\alpha_2^2 f_{xyy} + h^3 arb_{r2}b_{r3}\alpha_2\alpha_3 f_{xyy} +$
$\frac{h^3}{2!}arb_{r3}^2\alpha_3^2 f_{xyy} + \frac{h^3}{3!}(b_{r1}^3\alpha_1^3 + 3b_{r1}^2 b_{r2}\alpha_1^2\alpha_2 + 3b_{r1}^2 b_{r3}\alpha_1^2\alpha_3 + 3b_{r1}b_{r2}^2\alpha_1\alpha_2^2 +$
$6b_{r1}b_{r2}b_{r3}\alpha_1\alpha_2\alpha_3 + 3b_{r1}b_{r3}^2\alpha_1\alpha_3^2 + b_{r2}^3\alpha_2^3 + 3b_{r2}^2 b_{r3}\alpha_2^2\alpha_3 +$
$3b_{r2}b_{r3}^2\alpha_2\alpha_3^2 + b_{r3}^3\alpha_3^3)f_{yyy} + O(h^4)$ (3.5)

Equating (3.4) *with* (3.5) *in powers of* h, *we obtain*:

$\alpha_r = f$
$\beta_r = \alpha_r f_x + (b_{r1}\alpha_1 + b_{r2}\alpha_2 + b_{r3}\alpha_3)f_y$
$\theta_r = (b_{r1}\beta_1 + b_{r2}\beta_2 + b_{r3}\beta_3)f_y + \frac{1}{2!}a_r^2 f_{xx} + (a_r b_{r1}\alpha_1 + a_r b_{r2}\alpha_2 + a_r b_{r3}\alpha_3)f_{xy}$
$+\frac{1}{2}(b_{r1}^2\alpha_1^2 + 2b_{r1}b_{r2}\alpha_1\alpha_2 + 2b_{r1}b_{r3}\alpha_1\alpha_3 + b_{r2}^2\alpha_2^2 + 2b_{r2}b_{r3}\alpha_2\alpha_3 + b_{r3}^2\alpha_3^2)f_{yy}$
$\emptyset_r = (b_{r1}\theta_1 + b_{r2}\theta_2 + b_{r3}\theta_3)f_y + (a_r b_{r1}\beta_1 + a_r b_{r2}\beta_2 + a_r b_{r3}\beta_3)f_{xy} +$
$(b_{r1}^2\beta_1 + b_{r1}b_{r2}\alpha_1\beta_2 + b_{r1}b_{r2}\beta_1\alpha_2 + b_{r1}b_{r3}\alpha_1\beta_3 + b_{r1}b_{r3}\beta_1\alpha_3 + b_{r2}^2\alpha_2\beta_2 +$
$b_{r2}b_{r3}\alpha_2\beta_3 + b_{r2}b_{r3}\beta_2\alpha_3 + b_{r3}^2\alpha_3\beta_3)f_{yy} + \frac{1}{6}a_r^3 f_{xxx} + \frac{1}{2}a_r^2(b_{r1}\alpha_1 + b_{r2}\alpha_2 + b_{r3}\alpha_3)$
$f_{xxy} + \frac{1}{2}a_r(b_{r1}^2 a_1^2 + 2b_{r1}b_{r2}\alpha_1\alpha_2 + 2b_{r1}b_{r3}\alpha_1\alpha_3 + b_{r2}^2\alpha_2^2 + 2b_{r2}b_{r3}\alpha_2\alpha_3 + b_{r3}^2 a_3^2)$
$f_{xyy} + \frac{1}{6}(b_{r1}^3 a_1^3 + 3b_{r1}^2 b_{r2}a_1^2\alpha_2 + 3b_{r1}^2 b_{r3}\alpha_1^2\alpha_3 + 3b_{r1}b_{r2}^2\alpha_1\alpha_2^2 +$
$6b_{r1}b_{r2}b_{r3}\alpha_1\alpha_2\alpha_3 + 3b_{r1}b_{r3}^2\alpha_1\alpha_3^2 + b_{r2}^3\alpha_2^3 + 3b_{r2}^2 b_{r3}\alpha_2^2\alpha_3 +$
$3b_{r2}b_{r2}^3\alpha_2\alpha_3^2 + b_{r3}^3\alpha_3^3)f_{yyy} + O(h^4), \quad r = 1, 2, 3$ (3.6)

From (3.4), we know that:

$k_1 = \alpha_1 + h\beta_1 + h^2\theta_1 + h^3\emptyset_1$
$k_2 = \alpha_2 + h\beta_2 + h^2\theta_2 + h^3\emptyset_2$
$k_3 = \alpha_3 + h\beta_3 + h^2\theta_3 + h^3\emptyset_3$ (3.7)

Also from (1), $\emptyset(x, y, h) = \sum_{r=1}^{3} c_r k_r = c_1 k_1 + c_2 k_2 + c_3 k_3$ (3.8)

Putting (3.7) into (3.8), we have:

$\emptyset(x, y, h) = c_1\alpha_1 + c_2\alpha_2 + c_3\alpha_3 + h(c_1\beta_1 + c_2\beta_2 + c_3\beta_3) + h^2(c_1\theta_1 + c_2\theta_2 + c_3\theta_3) + h^3(c_1\emptyset_1 + c_2\emptyset_2 +$
$c_3\emptyset_3) + O(h^4)$ (3.9)

The necessary Taylor's series expansion is:

$\emptyset_T(x, y, h) = f + \frac{h}{2!}(f_x + ff_y) + \frac{h^2}{3!}(f_{xx} + 2ff_{xy} + f_x f_y + ff_y^2 + f^2 f_{yy}) + \frac{h^3}{4!}(f_{xxx} + 3ff_{xxy} + 3f^2 f_{xyy} +$
$3f_x f_{xy} + 5ff_y f_{xy} + 3ff_x f_{yy} + f_{xx}f_y + 4f^2 f_y f_{yy} + f_x f_y^2 + ff_y^3 + f^3 f_{yyy})$ (3.10)

Putting $\alpha_i, \beta_i, \theta_i$ and $\emptyset_i$ $i = 1,2,3$, into (3.7), and the resulting k_1, k_2, k_3 into (3.8) and equating coefficients with (3.10), we have the equations below:

$c_1 + c_2 + c_3 = 1, \quad c_1 a_1 + c_2 a_2 + c_3 a_3 = \frac{1}{2}, \quad c_1 a_1^2 + c_2 a_2^2 + c_3 a_3^2 = \frac{1}{3}, \quad c_1 a_1^3 + c_2 a_2^3 + c_3 a_3^3 = \frac{1}{4}$
$c_1(b_{11}a_1 + b_{12}a_2 + b_{13}a_3) + c_2(b_{21}a_1 + b_{22}a_2 + b_{23}a_3) + c_3(b_{31}a_1 + b_{32}a_2 + b_{33}a_3) = \frac{1}{6}$
$c_1(b_{11}a_1^2 + b_{12}a_1 a_2 + b_{13}a_1 a_3) + c_2(b_{21}a_1 a_2 + b_{22}a_2^2 + b_{23}a_2 a_3) + c_3(b_{31}a_1 a_3 + b_{32}a_2 a_3 + b_{33}a_3^2) = \frac{1}{8}$
$c_1(b_{11}a_1^2 + b_{12}a_2^2 + b_{13}a_3^2) + c_2(b_{21}a_1^2 + b_{22}a_2^2 + b_{23}a_3^2) + c_3(b_{31}a_1^2 + b_{32}a_2^2 + b_{33}a_3^2) = \frac{1}{12}$
$c_1[a_1(b_{11}^2 + b_{12}b_{21} + b_{13}b_{31}) + a_2(b_{11}b_{12} + b_{12}b_{22} + b_{13}b_{32}) + a_3(b_{11}b_{13} + b_{12}b_{23} + b_{13}b_{33})] +$
$c_2[a_1(b_{11}b_{21} + b_{21}b_{22} + b_{23}b_{31}) + a_2(b_{12}b_{21} + b_{22}^2 + b_{23}b_{32}) + a_3(b_{21}b_{13} + b_{22}b_{23} + b_{23}b_{33})] +$
$c_3[a_1(b_{11}b_{31} + b_{21}b_{32} + b_{31}b_{33}) + a_2(b_{12}b_{31} + b_{22}b_{32} + b_{32}b_{33}) + a_3(b_{13}b_{31} + b_{23}b_{32} + b_{33}^2)] = \frac{1}{24}$
(3.11)

$Note: a_1 = b_{11} + b_{12} + b_{13}, \quad a_2 = b_{21} + b_{22} + b_{23}, \quad a_3 = b_{31} + b_{32} + b_{33},$

From the general implicit Runge-Kutta scheme in (3.1), we have the third-stage fourth-order method:

$$y_{n+1} - y_n = h(c_1k_1 + c_2k_2 + c_3k_3)$$
$$k_1 = f(x_n + ha_1, y_n + h(b_{11}k_1 + b_{12}k_2 + b_{13}k_3))$$
$$k_2 = f(x_n + ha_2, y_n + h(b_{21}k_1 + b_{22}k_2 + b_{23}k_3))$$
$$k_3 = f(x_n + ha_3, y_n + h(b_{31}k_1 + b_{32}k_2 + b_{33}k_3)) \quad (3.12)$$

Resolving (3.11) and putting the parameters into (3.12), we have the implicit third-stage fourth-order formula below:

$$y_{n+1} - y_n = \frac{h}{4}(k_1 + 2k_2 + k_3)$$
$$k_1 = f\left(x_n + h\left(\frac{3-\sqrt{6}}{6}\right),\ y_n + h\left(\frac{k_1}{8} + \left(\frac{6-\sqrt{6}}{24}\right)k_2 + \left(\frac{1-\sqrt{6}}{8}\right)k_3\right)\right)$$
$$k_2 = f\left(x_n + \frac{h}{2},\ y_n + h\left(\left(\frac{6+\sqrt{6}}{48}\right)k_1 + \frac{k_2}{4} + \left(\frac{6-\sqrt{6}}{48}\right)k_3\right)\right)$$
$$k_3 = f\left(x_n + h\left(\frac{3+\sqrt{6}}{6}\right), y_n + h\left(\left(\frac{1+\sqrt{6}}{8}\right)k_1 + \left(\frac{6+\sqrt{6}}{24}\right)k_2 + \frac{k_3}{8}\right)\right) \quad (3.13)$$

The Butcher's tableau for our method is:

$$\begin{array}{c|ccc}
\frac{3-\sqrt{6}}{6} & \frac{1}{8} & \frac{6-\sqrt{6}}{24} & \frac{1-\sqrt{6}}{8} \\
\frac{1}{2} & \frac{6+\sqrt{6}}{48} & \frac{1}{4} & \frac{6-\sqrt{6}}{48} \\
\frac{3+\sqrt{6}}{6} & \frac{1+\sqrt{6}}{8} & \frac{6+\sqrt{6}}{24} & \frac{1}{8} \\
\hline
 & \frac{1}{4} & \frac{1}{2} & \frac{1}{4}
\end{array}$$

The Lobatto's third-stage fourth-order implicit formula below:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 4k_2 + k_3)$$
$$k_1 = f\left(x_n, y_n + \frac{h}{6}(k_1 - 2k_2 + k_3)\right)$$
$$k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{12}(2k_1 + 5k_2 - k_3)\right)$$
$$k_3 = f\left(x_n + h, y_n + \frac{h}{6}(k_1 + 4k_2 + k_3)\right)$$

IV. STABILITY FUNCTION OF OUR IMPLICIT METHOD

According to Butcher (2010), the stability function of any implicit RungeKutta method is:

$$y_{n+1} = \det\frac{(I - ZA + Zeb^T)y_n}{\det(1 - ZA)}$$

From our implicit method above,

$$A=\begin{bmatrix} \frac{1}{8} & \frac{6-\sqrt{6}}{24} & \frac{1-\sqrt{6}}{8} \\ \frac{6+\sqrt{6}}{48} & \frac{1}{4} & \frac{6-\sqrt{6}}{48} \\ \frac{1+\sqrt{6}}{8} & \frac{6+\sqrt{6}}{24} & \frac{1}{8} \end{bmatrix}$$

$$e=(1,1,1)^T,\ I=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ b=\begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix},\ b^T=\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

$$(I-zA)=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - z\begin{bmatrix} \frac{1}{8} & \frac{6-\sqrt{6}}{24} & \frac{1-\sqrt{6}}{8} \\ \frac{6+\sqrt{6}}{48} & \frac{1}{4} & \frac{6-\sqrt{6}}{48} \\ \frac{1+\sqrt{6}}{8} & \frac{6+\sqrt{6}}{24} & \frac{1}{8} \end{bmatrix}$$

$$(I-zA)=\begin{bmatrix} (1-\frac{z}{8}) & (-\frac{z}{4}+\frac{\sqrt{6}z}{24}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{8}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{48}) & (1-\frac{z}{4}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{48}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{8}) & (-\frac{z}{4}+\frac{\sqrt{6}z}{24}) & (1-\frac{z}{8}) \end{bmatrix}$$

$$Det(I-zA)=Det\begin{bmatrix} (1-\frac{z}{8}) & (-\frac{z}{4}+\frac{\sqrt{6}z}{24}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{8}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{48}) & (1-\frac{z}{4}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{48}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{8}) & (-\frac{z}{4}-\frac{\sqrt{6}z}{24}) & (1-\frac{z}{8}) \end{bmatrix}$$

$$Det(I-zA)=\left(1-\frac{z}{8}\right)\left[\left(1-\frac{z}{4}\right)\left(1-\frac{z}{8}\right)-\left(\frac{-z}{8}+\frac{\sqrt{6}z}{48}\right)\left(\frac{-z}{4}-\frac{\sqrt{6}z}{24}\right)\right]-\left(\frac{-z}{4}+\frac{\sqrt{6}z}{24}\right)\left[\left(\frac{-z}{8}-\frac{\sqrt{6}z}{48}\right)\left(1-\frac{z}{8}\right)-\left(\frac{-z}{8}+\frac{\sqrt{6}z}{48}\right)\left(\frac{-z}{8}-\frac{\sqrt{6}z}{8}\right)\right]+\left(\frac{-z}{8}+\frac{\sqrt{6}z}{8}\right)\left[\left(\frac{-z}{8}-\frac{\sqrt{6}z}{48}\right)\left(\frac{-z}{4}-\frac{\sqrt{6}z}{24}\right)-\left(1-\frac{z}{4}\right)\left(\frac{-z}{8}-\frac{\sqrt{6}z}{8}\right)\right]$$

$$Det(I-zA)=(1-\frac{z}{2}+\frac{5z^2}{96}-\frac{z^3}{1536})-(\frac{5z^2}{192}-\frac{5z^3}{768}+\frac{5\sqrt{6}z^3}{1536})+(\frac{5z^2}{64}-\frac{25z^3}{1536}+\frac{5\sqrt{6}z^3}{1536})$$

$$=1-\left(\frac{1}{2}\right)z+\left(\frac{5}{48}\right)z^2-\left(\frac{1}{96}\right)z^3$$

$$\left(I-zA+zeb^T\right)=\begin{bmatrix} (1-\frac{z}{8}) & (-\frac{z}{4}+\frac{\sqrt{6}z}{24}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{8}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{48}) & (1-\frac{z}{4}) & (-\frac{z}{8}+\frac{\sqrt{6}z}{48}) \\ (-\frac{z}{8}-\frac{\sqrt{6}z}{8}) & (-\frac{z}{4}-\frac{\sqrt{6}z}{24}) & (1-\frac{z}{8}) \end{bmatrix}+\begin{bmatrix} \frac{z}{4} & \frac{z}{2} & \frac{z}{4} \\ \frac{z}{4} & \frac{z}{2} & \frac{z}{4} \\ \frac{z}{4} & \frac{z}{2} & \frac{z}{4} \end{bmatrix}$$

$$Det(I-zA+zeb^T)=\det\begin{bmatrix} (1+\frac{z}{8}) & (\frac{z}{4}+\frac{\sqrt{6}z}{24}) & (\frac{z}{8}+\frac{\sqrt{6}z}{8}) \\ (\frac{z}{8}-\frac{\sqrt{6}z}{48}) & (1+\frac{z}{4}) & (\frac{z}{8}+\frac{\sqrt{6}z}{48}) \\ (\frac{z}{8}-\frac{\sqrt{6}z}{8}) & (\frac{z}{4}-\frac{\sqrt{6}z}{24}) & (1+\frac{z}{8}) \end{bmatrix}$$

$$Det(I-zA+zeb^T)=\left(1+\frac{z}{8}\right)\left[\left(1+\frac{z}{4}\right)\left(1+\frac{z}{8}\right)-\left(\frac{z}{8}+\frac{\sqrt{6}z}{48}\right)\left(\frac{z}{4}-\frac{\sqrt{6}z}{24}\right)\right]-\left(\frac{z}{4}+\frac{\sqrt{6}z}{24}\right)\left[\left(\frac{z}{8}-\frac{\sqrt{6}z}{48}\right)\left(1+\frac{z}{8}\right)-\left(\frac{z}{8}+\frac{\sqrt{6}z}{48}\right)\left(\frac{z}{8}-\frac{\sqrt{6}z}{8}\right)\right]+\left(\frac{z}{8}+\frac{\sqrt{6}z}{8}\right)\left[\left(\frac{z}{8}-\frac{\sqrt{6}z}{48}\right)\left(\frac{z}{4}-\frac{\sqrt{6}z}{24}\right)-\left(1+\frac{z}{4}\right)\left(\frac{z}{8}-\frac{\sqrt{6}z}{8}\right)\right]$$

$$Det(I-zA+zeb^T)=(1+\frac{z}{2}+\frac{5z^2}{96}+\frac{z^3}{1536})-(\frac{5z^2}{192}+\frac{5z^3}{768}+\frac{5\sqrt{6}z^3}{1536})+(\frac{5z^2}{64}+\frac{25z^3}{1536}+\frac{5\sqrt{6}z^3}{1536})$$

$$Det(I-zA+zeb^T)=1+\frac{z}{2}+\frac{5z^2}{48}+\frac{z^3}{96}$$

Gives birth to our stability function $y_{n+1} = \dfrac{1+\frac{z}{2}+\frac{5z^2}{48}+\frac{z^3}{96}}{1-\frac{z}{2}+\frac{5z^2}{48}-\frac{z^3}{96}}$

The above stability function is a rational polynomial which can be expressed as:

$$y_{n+1} = \left(1+\frac{z}{2}+\frac{5z^2}{48}+\frac{z^3}{96}\right)(1-\frac{z}{2}+\frac{5z^2}{48}-\frac{z^3}{96})^{-1} = (1+z+\frac{z^2}{2}+\frac{z^3}{6})$$

Theorem 4.1: The new fourth order third stage implicit Runge-Kutta method is an A-stable and L- stable method.

Proof:

From our definition on A-stability using the stability function in chapter two of this work,

We obtain that$y_{n+1} = R(z)y_n$, where $z = \lambda h$

And $R(z) = \dfrac{1+\frac{z}{2}+\frac{5z^2}{48}+\frac{z^3}{96}}{1-\frac{z}{2}+\frac{5z^2}{48}-\frac{z^3}{96}}$

We have:

$$y_{n+1} = \left(\frac{1+\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}+\frac{(\lambda h)^3}{96}}{1-\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}-\frac{(\lambda h)^3}{96}}\right)y_n$$

$$y_1 = \left(\frac{1+\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}+\frac{(\lambda h)^3}{96}}{1-\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}-\frac{(\lambda h)^3}{96}}\right)y_o$$

$$y_2 = \left(\frac{1+\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}+\frac{(\lambda h)^3}{96}}{1-\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}-\frac{(\lambda h)^3}{96}}\right)^2 y_o$$

$$y_3 = \left(\frac{1+\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}+\frac{(\lambda h)^3}{96}}{1-\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}-\frac{(\lambda h)^3}{96}}\right)^3 y_o$$

Therefore, for any fixed point n, we have:

$$y_n = \left(\frac{1+\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}+\frac{(\lambda h)^3}{96}}{1-\frac{(\lambda h)}{2}+\frac{5(\lambda h)^2}{48}-\frac{(\lambda h)^3}{96}}\right)^n y_o$$

So for any fixed point $t = t_n = nh,$, we have $\left|R(\lambda nh)^n\right| \to 0$ as $n \to \infty$ for all λh with Re(λ)<0, where $y_n \to 0$ as $n \to \infty$. Hence, $y_{n+1} = R(z).y_n$, and by induction, $y_n = (R(Z))^n.y_0$.

Hence, $R(z) = 1+z+\frac{1}{2}z^2+\frac{1}{6}z^3 = 0$. Using MAPLE to solve $R(z)$, thecomplex roots are:

$-1.596071637983321523 10, -0.70196418100833923844 - 1.80733949445202185360\,I,$
$-0.70196418100833923844 + 1.80733949445202185360\,I$

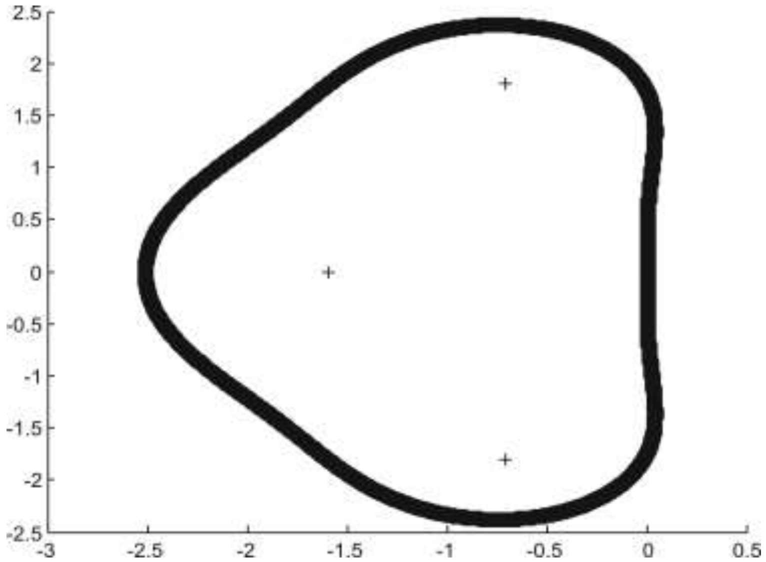
Following our definition in chapter two of this work which gives the conditions for A-stability and L- stability, it follows that $R(z)$ is said to be (i) A-stable if$|R(z)| < 1$ Whenever $Re(z) < 0$ that is z is real and negative (ii) L-stable, if it is A-stable and, in addition, satisfies
$|R(z)| \to 0$ as$Re \to -\infty$

It can be seen that the real parts of the solution of our polynomial are negative and are all less than zero, and for each real root, $|R(z)| < 1$thereby making our method to be A-stable. Also, for L-stability, we can see that the real parts of the complex roots approach $-\infty$ as$|R(z)| \to 0$, for example the absolute value of

$$1 + -1.59607163798332152310 + \frac{(-1.59607163798332152310)^2}{2} + \frac{(-1.59607163798332152310)^3}{6}$$

gives $5.00000000000000000000\ 10^{-20}$ which tends to zero. Hence our method is also L-stable.
For the solution of stiff problems, A-stability is a desirable property but L-stability is a more desirable property when a problem is excessively stiff.
Plotting the complex roots on a graph (the real parts on the x-axis and the imaginary parts on the y-axis) using MATLAB CODE, we have the absolute stability region seen in the diagram below:

Figure 1: Stability Region for our Implicit Method



It can be seen from all our proofs that all our methods (both explicit and implicit are absolutely stable, consistent and convergent. Also, our implicit method is A-stable and L-stable, thereby making the method to handle stiff problems in ordinary differential equations. Also, we found out that our implicit method performed better than Lobatto's implicit formula having the same stage and order as ours as can be seen in the error column of table 4.2.

V. IMPLEMENTATION ON STIFF PROBLEMS

We shall be implementing our method on the stiff problems below:
PROBLEM 1: $y' = -8y + 8x + 1, \quad y(0) = 2, \; 0 \le x \le 1, \quad y(x_n) = x + 2e^{-8x}, \quad h = 0.1$
PROBLEM 2: $y' = -15y, \; y(0) = 1, \; 0 \le x \le 1, y(x_n) = e^{-15x}, h = 0.01$
Table 1 Results for the stiff problems above
Results of problem 1 from the fully implicit third-stage fourth-order method:

XN	YN	TSOL	ERROR
.1D+00	0.99855072463768115940	0.99865792823444318286	0.00010720359676202346
.2D+00	0.60369670237345095568	0.60379303598931081698	0.00009633361585986130
.3D+00	0.48137098222575332792	0.48143590657882500675	0.00006492435307167883
.4D+00	0.48148551375359932124	0.48152440795673243033	0.00003889420313310909
.5D+00	0.53660943371538520230	0.53663127777746836059	0.00002184406208315829
.6D+00	0.61644771659676726480	0.61645949409804005768	0.00001177750127279288
.7D+00	0.70738955383333022042	0.70739572743296586164	0.00000617359963564122
.8D+00	0.80331994447584401207	0.80332311454634786900	0.00000317007050385693
.9D+00	0.90149156925726325180	0.90149317161675335874	0.00000160235949010694
1D+01	1.00067012531848059140	1.00067092525580502370	$7.99937324432300000000 \times 10^{-7}$
1.1+01	1.10030107079525939610	1.10030146615019095320	$3.95354931557100000000 \times 10^{-7}$

Results of problem 1 from Lobatto's fully implicit third-stage fourth-order method:

XN	YN	TSOL	ERROR
.1D+00	0.99820359281437125750	0.99865792823444318286	0.00045433542007192536
.2D+00	0.60338484707232242103	0.60379303598931081698	0.00040818891698839595
.3D+00	0.48116085946361785375	0.48143590657882500675	0.00027504711520715300
.4D+00	0.48135966742378047324	0.48152440795673243033	0.00016474053295195709
.5D+00	0.53653877279511099098	0.53663127777746836059	0.00009250498235736961
.6D+00	0.61640962850079834925	0.61645949409804005768	0.00004986559724170843
.7D+00	0.70736959363808309098	0.70739572743296586164	0.00002613379488277066
.8D+00	0.80330969774165408278	0.80332311454634786900	0.00001341680469378622
.9D+00	0.90148639120134165395	0.90149317161675335874	0.00000678041541170479
1D+01	1.00066754095868637150	1.00067092525580502370	0.00000338429711865220
1.1+01	1.10029979384372142430	1.10030146615019095320	0.00000167230646952890

Results of problem 2 from the fully implicit third-stage fourth-order method:

XN	YN	TSOL	ERROR
.01D+00	0.86070795369259161231	0.86070797642505780723	$2.27324661949200000000 \times 10^{-8}$
.02D+00	0.74081818154968842727	0.74081822068171786607	$3.91320294388000000000 \times 10^{-8}$
.03D+00	0.63762810109989915281	0.63762815162177329314	$5.05218741403300000000 \times 10^{-8}$
.04D+00	0.54881157811458712292	0.54881163609402643263	$5.79794393097100000000 \times 10^{-8}$
.05D+00	0.47236649036180817775	0.47236655274101470714	$6.23792065293900000000 \times 10^{-8}$
.06D+00	0.40656959531226321521	0.40656965974059911188	$6.44283358966700000000 \times 10^{-8}$
.07D+00	0.34993768441484315930	0.34993774911115535467	$6.46963121953700000000 \times 10^{-8}$
.08D+00	0.30119414827262356352	0.30119421191220209664	$6.36395785331200000000 \times 10^{-8}$
.09D+00	0.25924019902391285407	0.25924026064589150757	$6.16219786535000000000 \times 10^{-8}$
.1D+00	0.22313010121673221810	0.22313016014842982893	$5.89316976108300000000 \times 10^{-8}$

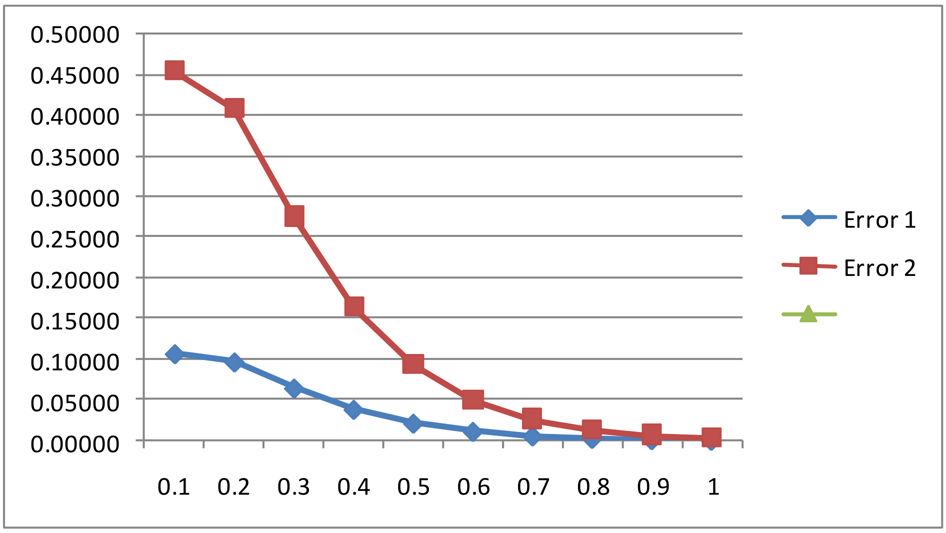
Results of problem 2 from Lobatto's fully implicit third-stage fourth-order method:

XN	YN	TSOL	ERROR
.01D+00	0.86070784837010437479	0.86070797642505780723	$1.28054953432440000000 \times 10^{-7}$

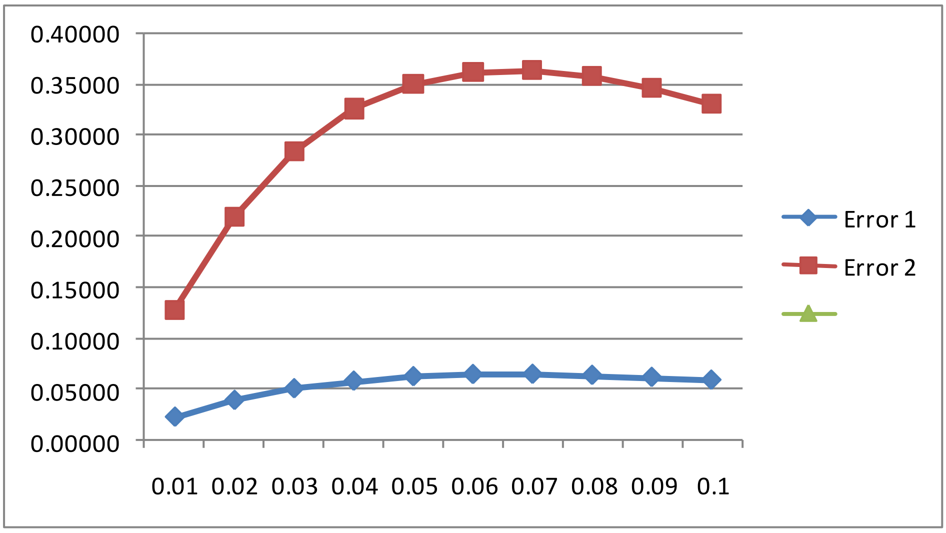
.02D+00	0.74081800024589458406	0.74081822068171786607	$2.20435823282010000000\,10^{-7}$
.03D+00	0.63762786702548738110	0.63762815162177329314	$2.84596285912040000000\,10^{-7}$
.04D+00	0.54881130948832626802	0.54881163609402643263	$3.26605700164610000000\,10^{-7}$
.05D+00	0.47236620135087674985	0.47236655274101470714	$3.51390137957290000000\,10^{-7}$
.06D+00	0.40656929680747261790	0.40656965974059911188	$3.62933126493980000000\,10^{-7}$
.07D+00	0.34993738466850610268	0.34993774911115535467	$3.64442649251990000000\,10^{-7}$
.08D+00	0.30119385342229143798	0.30119421191220209664	$3.58489910658660000000\,10^{-7}$
.09D+00	0.25923991352140106162	0.25924026064589150757	$3.47124490445950000000\,10^{-7}$
.1D+00	0.22312982817865703581	0.22313016014842982893	$3.31969772793120000000\,10^{-7}$

FIGURE 2 ABSOLUTE ERROR GRAPHS FOR THE ABOVE STIFF PROBLEMS

PROBLEM 1



PROBLEM 2



N.B: Error1 is from our three-stage fourth-order fully implicit method
Error2 is from Lobatto's three-stage fourth-order fully implicit method

VI. CONCLUSION

It is clearly seen from the two graphs that our implicit method performs better than the existing Lobatto's method. The error columns in our table of results table 5.1 justify our claim. Also, from our stability analysis it shows that our method is both A- and L-stable.

VII. REFERENCES

1. Agam, S. A and Yahaya, Y.A (2014); "A Highly Efficient Implicit Runge-Kutta Method For First Ordinary Differential Equations" African Journal Of Mathematics And Computer Science Research, ISSN 2006-9731, Vol. 7(5), PP. 55-60.
2. Butcher, J.C (1964); "Implicit Runge-Kutta Processes" http://dx.doi.org/10.11090/s0025-5718-1964-0159424-9.
3. Butcher, J.C (1975); "A Stability Property of Implicit Runge-Kutta methods", Springer link, springer.com/10.1007%2fBfo1(Springer Science + Business media)
4. Butcher, J.C (1988); "Towards Efficient ImplimentationOf Singly-Implicit Method" ACM Trans. MathsSoftw. 14:68-75, http://dx.doi.org/10.1145/42288.42341.
5. Butcher, J.C and Jackiewicz, Z (1997); "Implementation Of Diagonally Implicit Multi-Stage Integration Methods For ODEs" SIAM J. Number. Anal. 34:2119-2141.
6. Butcher, J. C., (1987);" The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General linear methods", John Wiley & Sons.
7. Butcher J.C., (1963);" coefficient for the study of Runge-kutta integration processes", J. Austral mathssoc., 3:185- 201.
8. Butcher J.C., (1963);" On the convergence of numerical solutions of ordinary differential equations", math. Comp. 20 : 1-10.
9. Butcher J.C., (2000). "Numerical methods for ordinary differential equations in the 20th century", journal of computational and applied mathematics 125(2000) 1-29
10. Butcher J.C., (2003); "Numerical methods for ordinary differential equations", wiley, chichester.
11. Butcher J.C., (2008);"Numerical methods for ordinary differential equations (2nd ed.) ", John wiley and sons ltd.
12. Butcher J.C., (2009);" Trees and Numerical methods for ordinary differential equations", Numerical Algorithms (Springer online).
13. Butcher J.C., (2009), "On the fifth and sixth order explicit Runge-Kutta methods. Order conditions and order Barries, Canadian applied Mathematics quarterly volume 17, numbers pg 433-445.
14. Butcher J.C., (2010);" Trees and numerical methods for ordinary differential equations", IMA J. Numer. Algorithms 53: 153 – 170.
15. Butcher J.C (2010) ;" Trees, B- series and exponential integrators" , IMA J. numer. Anal., 30: 131 – 140.
16. Copson, E.T. (1970), 'On a generalization of monotonic sequences', Proc. Edinburgh Math. Soc., 17, 159-164.
17. Gear, C.W., (1971)."The automatic integration of ordinary differential equations' Comm. ACM., 14, 176-179.
18. Lambert, J.D (1977); "Computational Methods In Ordinary differential Equations" .WArrowsmith Ltd, Bristol, Great Britain.
19. Lambert, J.D. (1992), "Numerical methods for ordinary differential systems", New York: Wiley, ISBN 978-0-471-92990-1.
20. Van der Houwen, P. J., Sommeijer, B. P., (2015);"Runge-Kutta projection methods with low dispersion and dissipation errors". Advances in computational methods, 41: 231-251.