INTERNATIONAL JOURNAL OF SCIENCE FOR GLOBAL SUSTAINABILITY

Derivation of Five-Stage Implicit Runge-Kutta Method of Order 10 via Gauss-Legendre Quadrature for Ordinary Differential Equations

Orapine, Hycienth O.¹, Oladele J.O.², Baidu Ali A.¹, Nyamtswam Elija Vafa¹*
¹Department of Mathematics, Nigerian Army University Biu, Borno State, Nigeria.
²Air Force Institute of Technology, Kaduna-Nigeria.

*Corresponding Author’s E-mail: domnyam@gmail.com

Received on: April, 2023 Revised and Accepted on: May, 2023 Published on: July, 2023

ABSTRACT

This paper presents developing and implementing a five-stage implicit Runge-Kutta method of order ten via Gauss-Legendre quadrature for stiff or oscillatory first-order initial value problems (IVPs) of ordinary differential equations (ODEs). Using continuous collocation and interpolation techniques, the implicit Runge-Kutta method was developed by combining Legendre polynomials and exponential functions as the basis function. The properties of the method were investigated, and it was shown that it is consistent and A-stable. The new method was evaluated on two sampled problems involving stiffness and oscillation. The numerical results demonstrate that the new implicit Runge-Kutta is computationally efficient and outperforms previous methods of similar derivations.

Keywords: Combined basis; Gaussian points; Gauss-Legendre Quadrature; ODEs; Runge-Kutta method.

1. Introduction

Numerical analysis is an essential branch of mathematics that plays a crucial role in the solution of complex problems in the sciences and engineering. In numerical analysis, the development of accurate and efficient numerical methods for approximating the solutions of ordinary differential equations (ODEs) is a crucial research topic (Hairer et al., 2006). In recent years, the derivation of high-order implicit Runge-Kutta methods has attracted considerable interest due to their superior stability and accuracy in comparison to explicit methods (Bendtsen, 1997). Not limited to, Butcher and Jackiewicz (1997 & 1998), Press et al. (2007), Yakubu (2010), Agam and Yahaya (2014), Agam (2016), Zhang et al., (2021) are additional scholarly works in this area of study.

In this vein, this paper presents a novel five-stage implicit Runge-Kutta method of order 10 derived via the Gauss-Legendre quadrature for ODEs that may exhibit stiffness or oscillatory behavior given by:

\[ y' = f(t, y(t)), y(t_0) = y_0, \forall a \leq t \leq b \]  

where \( t_0 \) is the initial point, \( y_0 \) is the solution at \( t_0 \) and \( f \) is assumed to be continuous and satisfy the Lipchitz theorem for the existence and uniqueness of the solution.

The method approximates the solution of the ODE (1.0) at discrete time increments using a five-stage approach based on the Gauss-Legendre quadrature formulas. This method is an important contribution to the field of numerical analysis, as it provides a highly accurate and efficient numerical scheme for solving ODEs (Kulikov et al., 2011). The resulting method is of order 10, ensuring outstanding accuracy, and its implicit nature ensures its robustness and stability. This implicit Runge-Kutta algorithm has the following form

\[ y_{n+1} = y_n + h \sum_{i=1}^{5} w_i K_i, \quad (S = \text{Stage}), \]  

where \( w_i \)s are weights of the Gauss quadrature \((w_i, q_i)\) and,
\[ K_i = f(t_n + q_i h, y_i), \]
\[ y_i = y_n + \sum_{i=1}^{S} a_{ij} K_i \quad j = 1, 2, ..., S. \]  

(1.2)

Deriving the method requires sophisticated numerical analysis techniques, such as continuous collocation and interpolation, as well as the properties of Gauss-Legendre quadrature formulas and implicit Runge-Kutta methods (Laurie, 2001; Press et al., 2007; Kulikov, 2009; Sahu & Saha Ray, 2016, Mohamed et al., 2020). Gauss-Legendre methods are founded on the quadrature formula with the highest possible order, \( p = 2S \). The coefficients \( a_{ij} \) are obtained from the Gauss points \( q_i \) which represent a linear system (i.e. summing \( a_{ij} \) gives \( q_i \)). The methods are symmetric and symplectic in nature, thus well suited for the long-term integration of Hamiltonian systems (Hairer & Wanner, 2015, Zhong et al., 2019).

This paper extends the works of Agam and Yahaya (2014) and Agam (2016) by constructing a five-stage implicit Runge-Kutta method of order 10 via the Gauss-Legendre quadrature for ODEs, as well as its numerical implementation and applications in several scientific disciplines. Researchers and practitioners in disciplines such as Physics, Chemistry, Engineering, and Mathematics who require accurate and efficient numerical methods for solving ODEs may find this paper's contributions significant.

2. Methodology

In this paper, we combined Legendre polynomials and exponential functions as an approximate solution to the equation (1.0) of the form:

\[ y(t) = \sum_{k=0}^{r} \varphi_k L_k(t) + \sum_{k=r+1}^{S} \sum_{j=0}^{k} \varphi_k \frac{\beta (j-1)!}{j!}, \]  

(1.3)

Equation (1.3) is called the basis function and is continuously differentiable. Where \( \beta \in \mathbb{R} \) is an arbitrary constant and \( \varphi_k \in \mathbb{R} \), \( k = 0, 1, ..., S \) are coefficients to be determined. \( L_k(t) \) are the Legendre polynomials, generated by the Rodrigues formula

\[ L_k(t) = \frac{1}{2^k k!} \frac{d^k}{dt^k} (t^2 - 1)^k. \]  

(1.4)

The first six Legendre polynomials are

\[ L(t) = 1, \quad L_1(t) = t, \quad L_2(t) = \frac{1}{2} (3t^2 - 1), \quad L_3(t) = \frac{1}{2} (5t^3 - 3t), \quad L_4(t) = \frac{1}{8} (35t^4 - 30t^2 + 3), \quad and \quad L_5(t) = \frac{1}{8} (63t^5 - 70t^3 + 15t). \]

Furthermore, \( S \) is the stage of Runge Kutta to be derived or the number of Gaussian points \( q_i \) which are the transformed zeros of Legendre polynomials defined by

\[ q_i = T(t_i) = \frac{1}{2} (1 \pm t_i), \]

\( t_i \) are perturbed zeros of the Legendre polynomials.

Using the degree five Legendre polynomial \( L_5(t) \), we obtained following the Gaussian points;

\[ q_1 = \frac{1}{2}, \quad q_2 = \frac{1}{2} + \frac{99\sqrt{739}}{10000}, \quad q_3 = \frac{1}{2} + \frac{99\sqrt{739}}{10000}, \quad q_4 = \frac{1}{2} - \frac{\sqrt{739}}{60}, \quad q_5 = \frac{1}{2} + \frac{\sqrt{739}}{60}. \]  

(1.5)

Now, obtaining the first derivative of (1.3) and substituting in (1.0) gives,

\[ f(t, y) = \sum_{k=1}^{r} \varphi_k L_k(t) + \sum_{k=r+1}^{S} \sum_{j=0}^{k} \varphi_k \frac{\beta (j-1)!}{j!}, \]  

(1.6)

Interpolating equation (1.3) at \( t = t_n \) and collocating equation (1.6) at \( t = t_{n+q_i} \), \( i = 1, 2, ..., S = 5 \); the system of nonlinear equations is obtained

\[ B \cdot \Phi = U, \]  

(1.7)

where

\[ U = \begin{pmatrix} \varphi_0 & \varphi_1 & \varphi_2 & \varphi_3 & \varphi_4 & \varphi_5 \end{pmatrix}^T, \]

\[ \Phi = \begin{pmatrix} \varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5 \end{pmatrix}^T, \]  

and
\[
B = \begin{pmatrix}
1 & \frac{1}{2}(3r_n^2 - 1) & \frac{1}{2}(Sr_n^3 - 3r_n) & \sum_{j=1}^{r_n} \beta_j t_j^n & \sum_{j=1}^{r_n} \beta_j t_j^{n+1} \\
0 & 1 & 3r_n + 1 & \frac{1}{2}(15r_n^2 + 3) & \sum_{j=1}^{r_n} \beta_j t_{n+1}^{r_n+1} \\
0 & 1 & 3r_n + 2 & \frac{1}{2}(15r_n^2 + 3) & \sum_{j=1}^{r_n} \beta_j t_{n+1}^{r_n+2} \\
0 & 1 & 3r_n + 3 & \frac{1}{2}(15r_n^2 + 3) & \sum_{j=1}^{r_n} \beta_j t_{n+1}^{r_n+3} \\
0 & 1 & 3r_n + 4 & \frac{1}{2}(15r_n^2 + 3) & \sum_{j=1}^{r_n} \beta_j t_{n+1}^{r_n+4} \\
0 & 1 & 3r_n + 5 & \frac{1}{2}(15r_n^2 + 3) & \sum_{j=1}^{r_n} \beta_j t_{n+1}^{r_n+5}
\end{pmatrix}
\]

Solving equation (1.7) in maple soft, using the matrix inversion method, the value of the unknown column vector \( \Phi \) is obtained. Vector \( \Phi \) is substituted in equation (1.0) to obtain a continuous scheme which is evaluated at the \( t = t_n + q_i \), \( i = 1, \ldots, 5 \), \( (q_i \) is the Gaussian points in equation (1.5)), and also evaluated at \( t = t_n + 1 \) to achieve the following block discrete schemes respectively:

\[
y_{n+q_1} = y_n + \frac{4563950663}{16} h_{f_n+q_1} + \frac{45156250000\sqrt{739}}{874738808389} + \frac{310937500000000}{2597974476091533} h_{f_n+q_2} + \left( \frac{45156250000\sqrt{739}}{874738808389} + \frac{310937500000000}{2597974476091533} \right) h_{f_n+q_3} + \left( \frac{709703235\sqrt{739}}{353429850844} + \frac{5236016175}{88357462711} \right) h_{f_n+q_4} + \left( \frac{709703235\sqrt{739}}{353429850844} + \frac{5236016175}{88357462711} \right) h_{f_n+q_5}
\]  

(1.8)

\[
y_{n+q_2} = y_n + \left( \frac{-38339103\sqrt{739}}{6250000000} + \frac{4563950663}{32115191526} \right) h_{f_n+q_1} + \left( \frac{9557056475401\sqrt{739}}{34989555233556000} + \frac{310937500000000}{2597974476091533} \right) h_{f_n+q_2} + \left( \frac{-14074198220719489\sqrt{739}}{34989555233556000} + \frac{310937500000000}{2597974476091533} \right) h_{f_n+q_3} + \left( \frac{5601362553163918341\sqrt{739}}{220893656777500000000} + \frac{5236016175}{88357462711} \right) h_{f_n+q_4} + \left( \frac{5040458465159165409\sqrt{739}}{220893656777500000000} + \frac{5236016175}{88357462711} \right) h_{f_n+q_5}
\]  

(1.9)
\[
y_{n+q_3} = y_n + \left( \frac{38339103\sqrt{739}}{625000000} + \frac{4563950663}{32115191526} \right) h f_{n+q_1} \\
+ \left( \frac{14074198220719498\sqrt{739}}{349895552335560000} + \frac{3109375000000000}{2597974476091533} \right) h f_{n+q_2} \\
+ \left( - \frac{9557056475401}{349895552335560000} + \frac{3109375000000000}{2597974476091533} \right) h f_{n+q_3} \\
+ \left( \frac{5040458465159165409\sqrt{739}}{2208936567775000000000} + \frac{5236016175}{88357462711} \right) h f_{n+q_4} \\
+ \left( - \frac{560136255316918341\sqrt{739}}{2208936567775000000000} + \frac{5236016175}{88357462711} \right) h f_{n+q_5} 
\]

(1.10)

\[
y_{n+q_4} = y_n + \left( - \frac{38209\sqrt{739}}{7938810} + \frac{4563950663}{32115191526} \right) h f_{n+q_1} \\
+ \left( - \frac{359369071903750\sqrt{739}}{70145310854471391} + \frac{3109375000000000}{2597974476091533} \right) h f_{n+q_2} \\
+ \left( - \frac{323282178906250\sqrt{739}}{70145310854471391} + \frac{3109375000000000}{2597974476091533} \right) h f_{n+q_3} \\
+ \left( - \frac{470139\sqrt{739}}{1413719403376} + \frac{5236016175}{88357462711} \right) h f_{n+q_4} \\
+ \left( \frac{44986764863\sqrt{739}}{21205791050640} + \frac{5236016175}{88357462711} \right) h f_{n+q_5} 
\]

(1.11)

\[
y_{n+q_5} = y_n + \left( \frac{38209\sqrt{739}}{7938810} + \frac{4563950663}{32115191526} \right) h f_{n+q_1} \\
+ \left( \frac{359369071903750\sqrt{739}}{70145310854471391} + \frac{3109375000000000}{2597974476091533} \right) h f_{n+q_2} \\
+ \left( \frac{323282178906250\sqrt{739}}{70145310854471391} + \frac{3109375000000000}{25974476091533} \right) h f_{n+q_3} \\
+ \left( \frac{44986764863\sqrt{739}}{21205791050640} + \frac{5236016175}{88357462711} \right) h f_{n+q_4} \\
+ \left( \frac{470139\sqrt{739}}{1413719403376} + \frac{5236016175}{88357462711} \right) h f_{n+q_5} 
\]

(1.12)

\[
y_{n+1} = y_n + \frac{4563950663}{16057595763} h f_{n+q_1} + \frac{621875000000000}{2597974476091533} h f_{n+q_2} + \frac{10472032350}{88357462711} h f_{n+q_3} + \frac{10472032350}{88357462711} h f_{n+q_4} + \frac{10472032350}{88357462711} h f_{n+q_5} 
\]

(1.13)

The block discrete schemes (1.8)-(1.12) must satisfy the first order IVP (1.0) to be converted into Runge-Kutta functions, and we let \( K_1 = y'_{n+q_1} = f(t_{n+q_1}, y_{n+q_1}) = f_{n+q_1}, K_2 = y'_{n+q_2} = f_{n+q_2}, \ldots, K_5 = y'_{n+q_5} = f_{n+q_5} \). Thus it follows that,
\[ K_1 = f \left[ t_n + \frac{h}{2}, \ y_n + \frac{4563950663}{3211519152} h K_1 + \left( \frac{45156250000\sqrt{739}}{874738880839} + \frac{31093750000000}{2597974476091533} \right) h K_2 \right. \]

\[ + \left( -\frac{45156250000\sqrt{739}}{874738880839} + \frac{31093750000000}{2597974476091533} \right) h K_3 \]

\[ + \left( \frac{709703235\sqrt{739}}{353429850844} + \frac{5236016175}{88357462711} \right) h K_4 \]

\[ + \left( -\frac{709703235\sqrt{739}}{353429850844} + \frac{5236016175}{88357462711} \right) h K_5 \] (1.14)

\[ K_2 = f \left[ t_n + \left( \frac{1}{2} - \frac{99\sqrt{739}}{10000} \right) h y_n + \left( -\frac{38339103\sqrt{739}}{6250000000} + \frac{4563950663}{3211519152} \right) h K_1 \right. \]

\[ + \left( \frac{9557056475401\sqrt{739}}{349895552335560000} + \frac{31093750000000}{2597974476091533} \right) h K_2 \]

\[ + \left( -\frac{14074198220719489\sqrt{739}}{349895552335560000} + \frac{31093750000000}{2597974476091533} \right) h K_3 \]

\[ + \left( \frac{560136253163918341\sqrt{739}}{220893656775000000000} + \frac{5236016175}{88357462711} \right) h K_4 \]

\[ + \left( -\frac{5040458465159165409\sqrt{739}}{220893656775000000000} + \frac{5236016175}{88357462711} \right) h K_5 \] (1.15)

\[ K_3 = f \left[ t_n + \left( \frac{1}{2} + \frac{99\sqrt{739}}{10000} \right) h y_n + \left( 38339103\sqrt{739} \right) + \frac{4563950663}{6250000000} \right) h K_1 \right. \]

\[ + \left( \frac{14074198220719489\sqrt{739}}{349895552335560000} + \frac{31093750000000}{2597974476091533} \right) h K_2 \]

\[ + \left( -\frac{9557056475401\sqrt{739}}{349895552335560000} + \frac{31093750000000}{25974476091533} \right) h K_3 \]

\[ + \left( \frac{5040458465159165409\sqrt{739}}{220893656775000000000} + \frac{5236016175}{88357462711} \right) h K_4 \]

\[ + \left( -\frac{5040458465159165409\sqrt{739}}{220893656775000000000} + \frac{5236016175}{88357462711} \right) h K_5 \] (1.16)

\[ K_4 = f \left[ t_n + \left( \frac{1}{2} - \frac{\sqrt{739}}{60} \right) h y_n + \left( -\frac{38209\sqrt{739}}{7938810} + \frac{4563950663}{3211519152} \right) h K_1 \right. \]

\[ + \left( \frac{359369071093750\sqrt{739}}{70145310854471391} + \frac{31093750000000}{2597974476091533} \right) h K_2 \]

\[ + \left( -\frac{323282178906250\sqrt{739}}{70145310854471391} + \frac{31093750000000}{2597974476091533} \right) h K_3 \]

\[ + \left( -\frac{470139\sqrt{739}}{1413719403376} + \frac{5236016175}{88357462711} \right) h K_4 \]

\[ + \left( -\frac{44986764863\sqrt{739}}{21205791050640} + \frac{5236016175}{88357462711} \right) h K_5 \] (1.17)
We combine these schemes (1.1)-(1.18) into one scheme by a linear combination of a scalar
\[ w = (w_1, w_2, w_3, w_4, w_5) \] to give the Runge-Kutta type scheme as
\[ y_{n+1} = y_n + h \sum_{i=1}^{5} w_i K_i \] (1.19)
where \( w_i, i = 1, 2, ..., 5 \) are the weights of the Gauss Legendre Quadrature defined from equation (1.13) as
\[ w_1 = \frac{4563950663}{1605795763}, \quad w_2 = w_3 = \frac{62187500000000}{2597974476091533}, \quad w_4 = w_5 = \frac{10472032350}{88357462711}, \]
and
\[ \sum_{i=1}^{5} w_i = 1 \] (1.20)
Hence, our five-stage Runge-Kutta method is derived as
\[ y_{n+1} = y_n + h \sum_{i=1}^{5} w_i K_i \]
\[ y_{n+1} = y_n + \frac{4563950663}{1605795763} h K_1 + \frac{62187500000000}{2597974476091533} h (K_2 + K_3) + \frac{10472032350}{88357462711} h (K_4 + K_5) \] (1.21)
where \( K_i, i = 1, 2, ..., 5 \) are given by equations (1.14)-(1.18), which is summarised in the Butcher tableau in Appendix I.

3. Analysis of the Runge-Kutta Method
3.1 Consistency: The consistency of the Runge-Kutta method is obvious since
\[ \sum_{i=1}^{5} w_i = 1, \quad \sum_{i=1}^{5} a_{ij} = q_i, \quad j = 1, 2, ..., 5. \]
as can be obtained from the Butchers tableau; Table 1 of Appendix I.

3.2. Stability: The stability investigation of the method is carried out by using the linear test equation
\[ y' = \varphi y, \quad \varphi \in \mathbb{C} \]
Putting \( r = \varphi h, \quad h \in (0,1) \) the stability function is
\[ R(r) = I + r w^T (I - r A)^{-1} e \]
where \( I \) is the identity matrix, \( w = (w_1, w_2, w_3, w_4, w_5) \) the weights, \( e = (1,1,1,1,1)^T \), and \( A \) is the Runge Kutta matrix of coefficients in the Butcher’s table 1 or equations (1.11)-(1.17). Since this method is based on Gauss-Legendre quadrature and related methods, A-stability is achievable satisfying the domain;
\[ R(r) = \{ r; R(r) \leq 0 \text{ and } Re(r) \leq 1 \} \]
The characteristic polynomial is defined as;
\[ P(r) = \det (R(r) - I \varphi) \]
and can be plotted in MATLAB to determine the region of A-stability.

3.3. Order and Error Estimation of the Scheme
Since the method is based on the Gauss-Legendre quadrature formula \((w_i, q_i)\) where \( q_i \)'s the Gaussian points are obtained from fifth-order Gauss-Legendre polynomial and all Gauss quadrature methods have the order \( p = 2S \) \((S = \text{stages})\) (Hairer & Wanner,
2015). Thus the order of the method is \( p = 2S = 10 \).

Furthermore, The Runge-Kutta method can be expanded into Taylor’s series as:

\[
y_{n+1} = y_n + h \sum_{i=1}^{5} w_i K_i
\]

\[
y_{n+1} = y_n + h \sum_{i=1}^{5} w_i f(t_n + q_i, y_n + q_i)
\]

\[
y_n + h \sum_{i=1}^{5} w_i y'_{n+q_i}
\]

\[
y_n + w_i \left( h y_n' + q_i h^2 y_n'' + \frac{q_i^2 h^3}{2!} y_n''' + \cdots + \frac{q_i^{p-1} h^p}{(p-1)!} y_n^{(p)} + \frac{q_i^p h^{p+1}}{p!} y_n^{(p+1)} \right)
\]

Thus,

\[
y(t_n + h) - y_{n+1} = C_1 h y_n' + C_2 h^2 y_n'' + \cdots + C_p h^p y_n^{(p)} + C_{p+1} h^{p+1} y_n^{(p+1)}
\]

Here \( y(t_n + h) \) is the exact solution of the problem (1.0) and the Runge-Kutta solution agrees with Taylor’s series expansion up to a term in \( h^{10} \). The truncation error at \( c_{11} \) is \( O(h^{11}) \) and the Error constant term is \( -\frac{20633543467499902945281043}{465589555200000000000000000000000000} \).

The local or global error bound is estimated to be (Agam & Yahaya, 2014):

\[
0(h^{p+1}) = \frac{2^{p+1}}{2^{p+1} - 1} \left( y_h^{(\frac{h}{2})} - y(h) \right),
\]

where \( y_h^{(\frac{h}{2})} \) and \( y(h) \) are the solutions of the method with step length \( \frac{h}{2} \) and \( h \) respectively.

### 4.0 Numerical Experiment

We used two examples to compare our computational solutions with exact and similar existing methods of lower order to show the efficiency of the new methods so derived.

**Example 4.1.** We consider the oscillatory problem arising from the electrical RL circuit system modeled Kirchhoff’s law as \( L \frac{dl}{dt} + RI = E(t) \) where \( L \) is the inductance constant, \( I \) is current, \( R \) is the resistant constant and \( E(t) \) is the periodic electromotive force, require to determine the current \( I \) at time \( t = 0.1(0.1)0.5 \text{sec} \), with \( L = 0.05 \), \( R = 1 \), \( E(t) = \sin(t) \). i.e.

\[
\frac{dl}{dt} = -20I + 20 \sin(t), I(0) = \frac{1}{10}, h = 0.1
\]

with an analytical solution:

\[
I(t) = -\frac{20}{401} \cos(t) + \frac{400}{401} \sin(t) + \frac{601}{4010} e^{-20t}.
\]

**Example 4.2:** We consider the ODE:

\[
y' = y^2 - 8y + 15, \ y(0) = 0,
\]

with exact solution \( y(t) = \frac{15(e^{2t-1})}{5e^{2t}-3} \).

This equation is a particular case of a general differential equation \( y' = K(q - y)(p - y) \), \( y(t_0) = y_0 \) derived from chemical reaction Law which states that, at a constant temperature, the chemical reaction is proportional to the product of concentrations of the reacting substances. In this problem the proportional constant \( K = 1 \), and arbitrary constants \( q = 5 \) and \( p = 3 \) while \( t_0 = 0, y_0 = 0 \) (Agam, 2016).

The following tables show the comparison of the results of the two problems and Figure 1 shows the comparison of absolute error of different existing methods with the new method.
Table 4.1: Comparison of Numerical Solutions to Example 4.1

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.0702417303634</td>
<td>0.0702417300835</td>
<td>1.82 E-06</td>
<td>2.9919 $E - 5$</td>
<td>1.75 E-09</td>
<td>2.799E-10</td>
</tr>
<tr>
<td>0.2</td>
<td>0.152037832066</td>
<td>0.152037831990</td>
<td>3.80 E-06</td>
<td>8.0931 $E - 6$</td>
<td>2.32 E-09</td>
<td>7.6E-11</td>
</tr>
<tr>
<td>0.3</td>
<td>0.247507047102</td>
<td>0.247507047087</td>
<td>1.65 E-06</td>
<td>1.6429 $E - 6$</td>
<td>3.27 E-09</td>
<td>1.5E-11</td>
</tr>
<tr>
<td>0.4</td>
<td>0.342559297624</td>
<td>0.342559297622</td>
<td>2.68 E-07</td>
<td>2.9800 $E - 7$</td>
<td>3.37 E-09</td>
<td>2.0E-12</td>
</tr>
<tr>
<td>0.5</td>
<td>0.434467064189</td>
<td>0.434467064188</td>
<td>1.46 E-07</td>
<td>5.2647 $E - 8$</td>
<td>3.39E-10</td>
<td>1.0E-12</td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of Numerical Solutions to Example 4.2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>1.06888530145</td>
<td>1.06888530006</td>
<td>5.18 E-06</td>
<td>9.93 E-08</td>
<td>6.987E-8</td>
<td>1.30 E-08</td>
<td>1.39E-9</td>
</tr>
<tr>
<td>0.2</td>
<td>1.65444408170</td>
<td>1.65444408071</td>
<td>3.66 E-06</td>
<td>7.72 E-08</td>
<td>5.382E-8</td>
<td>1.02 E-08</td>
<td>9.9E-10</td>
</tr>
<tr>
<td>0.3</td>
<td>2.01809873180</td>
<td>2.01809873112</td>
<td>2.49 E-06</td>
<td>5.55 E-08</td>
<td>3.856E-8</td>
<td>7.31 E-09</td>
<td>6.8E-10</td>
</tr>
<tr>
<td>0.4</td>
<td>2.26178419815</td>
<td>2.26178419765</td>
<td>1.17 E-06</td>
<td>4.03 E-08</td>
<td>2.794E-8</td>
<td>5.28 E-09</td>
<td>5.0E-10</td>
</tr>
<tr>
<td>0.5</td>
<td>2.43350314209</td>
<td>2.43350314174</td>
<td>1.26 E-06</td>
<td>2.98 E-08</td>
<td>2.070E-8</td>
<td>3.89 E-09</td>
<td>3.5E-10</td>
</tr>
</tbody>
</table>

In Tables 4.1 and 4.2, the new implicit Runge-Kutta method has been shown to outperform the existing method of similar derivations. In addition, as shown in Figure 1, the absolute error is plotted against the number of $n$ subintervals.
5.0 CONCLUSION
This study was able to demonstrate that selecting a higher degree Legendre polynomial to develop an implicit Gauss-Legendre Runge-Kutta type method yields a superior result and a solution that is extremely convergent for first-order IVP ODEs. We constructed a five-stage Runge-Kutta method of order 10 via the Gauss-Legendre quadrature, which has a maximum order of 2S. The new implicit method of order 10 is tested using experimental problems of both stiff and oscillatory types and is found to be highly efficient, A-stable, and produce fewer errors than the existing Gauss Legendre methods and other methods that share comparable derivations. Thus, the new method converges more effectively.

REFERENCES

Lie, J. and Norsett, S.P (1989) Superconvergence
for multistep collocation, math compt. (52) P 65-79.
## Appendix I

### Table 1: Butcher tableau summary of the five-stage Implicit Runge-Kutta method via Gauss-Legendre quadrature.

<table>
<thead>
<tr>
<th>$q$</th>
<th>$A = (a_{ij}), ij = 1, 2, ..., 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\begin{pmatrix} 4563950663 &amp; (310937500000000) \ 32115191526 &amp; (2597974476091533) \ + &amp; 8747388808389 \ \end{pmatrix}$</td>
</tr>
<tr>
<td>$2\sqrt{739}$</td>
<td>$\begin{pmatrix} 32115191526 &amp; (310937500000000) \ 32115191526 &amp; (2597974476091533) \ + &amp; 8747388808389 \ \end{pmatrix}$</td>
</tr>
<tr>
<td>$\frac{1 + \sqrt{739}}{2}$</td>
<td>$\begin{pmatrix} 32115191526 &amp; (310937500000000) \ 32115191526 &amp; (2597974476091533) \ + &amp; 8747388808389 \ \end{pmatrix}$</td>
</tr>
<tr>
<td>$\frac{1 + \sqrt{739}}{2}$</td>
<td>$\begin{pmatrix} 32115191526 &amp; (310937500000000) \ 32115191526 &amp; (2597974476091533) \ + &amp; 8747388808389 \ \end{pmatrix}$</td>
</tr>
<tr>
<td>$w$</td>
<td>$\begin{pmatrix} 4563950663 &amp; (621875000000000) \ 16057595763 &amp; (2597974476091533) \ + &amp; 88357462711 \ \end{pmatrix}$</td>
</tr>
</tbody>
</table>