A Four-Stage Gauss-Lobatto Integral with Accurate Error Estimation Formula for ODES

S.A. Agam, Ph.D.

Department of Mathematics, Nigerian Defence Academy (NDA), Kaduna, Nigeria.

E-mail: agamsylvester@gmail.com

ABSTRACT

We present Gauss-Lobatto integral method with accurate error formula for solution of Ordinary Differential Equations. Legendre polynomial of degree three and its corresponding Lobatto polynomial are used as bases. Collocation approach is used to derive a continuous scheme. The collocation points are the transformed zeros of the Lobatto polynomial onto the positive x-axis. Evaluating the continuous scheme at these points, we obtained discrete schemes which are converted to Runge-Kutta function evaluations for the iteration of the solutions. A corresponding error estimation formula is derived for accurate calculation of error of the solutions. The method is highly efficient and A-stable. Some problems are used to test our formulas.

(Keywords: Gauss-Lobatto polynomials, guadrature points, collocation method, Runge Kutt f-evaluations, error estimation, highly efficient, A-stable)

INTRODUCTION

These are many types of implicit Runge-Kutta methods derived by authors [4] for the integration of initial- value problems of ordinary differential equations. The common implicit method uses function values at predetermined (equidistant) x-values (nodes) which gives result for polynomial of degree n(for even number of nodes) and n+1(for odd number of nodes). However we can get much more accurate integration formulas by using Radau, Lobatto, Gauss-Legendre polynomials [5]. In this paper we use the Legendre polynomial and it's correspond Lobatto polynomial to get the transformed zeros of the Lobatto polynomial in the interval [0,1], see reference [7].

The other problem in numerical integration is to determine the level of accuracy of our results, especially problems without close form or analytic solutions. E. Feldberg [2] proposed and developed error control by using two R-K methods of different orders; this method is not quite efficient since it cannot give accurate error estimate, more so his method is for explicit methods and explicit methods are unsuitable for stiff or oscillation differential equation problems.

In this paper we develop a simple Runge-Kutta method with accurate error estimation formula for both stiff and non-stiff problems. Some varieties of problems are used to test our methods. The new schemes are highly accurate and stable.

METHODOLOGY

We consider the general initial – value problem of first order ordinary differential equation:

$$y^{/} = f(x, y), y(x_0) = y_0 \qquad \textbf{(1)}$$

We seek a continuous scheme of they form:

$$y(x) = \sum_{j=0}^{t-1} Q_j(x)\ y_{n+j}(x) + \sum_{j=0}^{m-1} h\ \beta_j(x) f_{n+j}(x) \qquad \textbf{(2)}$$

Where t is the number of interpolation points, m is the number of collocation points, h is a constant step size. $Q_j(x) and\, \beta_j(x)$ are polynomial functions defined as follows:

$$Q_j(x) = \sum_{j=0}^{t-1} Q_{ij} P_i(x)$$

$$\beta_j(x) = \sum_{j=0}^{m-1} h\ \beta_{ij} P_i(x) \qquad \textbf{(3)}$$

The coefficients $Q_{ij}\ h\beta i_j$ are undetermined elements of $(t+m)$ x $(t+m)$ matrix:

$$C = \begin{pmatrix} Q_{1,1} & Q_{1,2} & Q_{1,3} \ldots\ldots & Q_{1,t-1} & h\,\beta_{1,1} & h\,\beta_{1,2} \ldots\ldots & h\,\beta_{1,m-1} \\ Q_{2,1} & Q_{2,2} & Q_{2,3} \ldots\ldots & Q_{2,t-1} & h\,\beta_{2,1} & h\,\beta_{3,2} \ldots\ldots & h\,\beta_{31,m-1} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots \\ Q_{t+m,1} & Q_{t+m,2} & Q_{t+m,t-1} \ldots\ldots & & h\,\beta_{t+m,1} & \ldots\ldots & h\,\beta_{t+m,m-1} \end{pmatrix} = (c_{ij}).. \quad \textbf{(4)}$$

Substituting Equation (3) into (2) we have:

$$y(x) = \sum_{i=0}^{t+m-1} \left(\sum_{j=0}^{t-1} Q_{ij} y_{n+j} + \sum_{j=1}^{m-1} \beta_{ij} f_{n+j} \right) P_i(x) = \sum_{i-0}^{t+m-1} a_j P_i(x) = a^T P(x) \quad \textbf{(5)}$$

$$Where \ \ a_i = \sum_{j=1}^{t-1} Q_{ij}\, y_{n+j} + \sum_{j=1}^{m-1} \beta_{ij} f_{n+j}$$

$$a = (a_0, a_1, a_2, \ldots\ldots a_{t+m-1})$$

$P(x) = (p_0(x), p_1(x), p_2(x) \ldots \ldots p_{t+m-1}(x))$) are basis functions. From Equation (5) we impose the following interpolation and collocation conditions.

$$\begin{aligned} a_0 p_0(x_i) + a_1 p_1(x_i) + \ldots a_{t+m-1} p_{t+m-1}(x_i) &= y_i \\ a_0 p_0'(x_i) + a_1 p_1'(x_i) + \ldots a_{t+m-1} p_{t+m-1}'(x_i) &= f_i \quad \ldots\ldots\ldots\ldots \end{aligned} \quad \textbf{(6)}$$

Define $V = (y_n, y_{n+1} \ldots \ldots y_{n+q_t}, f_n, f_{n+q_1} \ldots \ldots f_{n+q_m})$ and a D-Matrix:

$$p_0(x_n) \quad p_1(x_n) \; p_2(x_n) \ldots\ldots p_{t+m-1}(x_n)$$

$$D = \begin{pmatrix} p_0(x_{n+q_1}) & p_1(x_{n+q_1}) & \ldots\ldots\ldots & p_{t+m-1}(x_{n+q_1}) \\ p_0(x_{n+q_2}) & p_1(x_{n+q_2}) & \ldots\ldots\ldots & p_{t+m-1}(x_{n+q_2}) \\ \cdots & \cdots & \cdots & \cdots \\ p_0'(x_{n+q_{t-1}} & p_1(x_{n+q_{t-1}}) & \ldots & p_{t+m-1}(x_{n+q_{m-1}}) \\ p_0'(x_n) & p_1'(x_n) & \ldots\ldots & p_{t+q_{m-1}}'(x_{n+q_1}) \\ \cdots & \cdots & \cdots & \cdots \\ p_0'(x_{n+q_{t-1}}) & p_1'(x_{n+q_{t-1}}) & \ldots\ldots & p_{t+m-1}'(x_{n+q_{t-1}}) \end{pmatrix} \quad \textbf{(7)}$$

The matrix D is assumed non-singular.
Then from Equation (6) and (7), we can write:

$$Da = \begin{pmatrix} p_0(x_n), & p_1(x_n) & \cdots & p_{t+m}(x_n) \\ p_0(x_{n+q_1}) & p_1(x_{n+q_1}) & \cdots & p_{t+m-1}(x_{n+q_1}) \\ \cdots & \cdots & \cdots & \cdots \\ p_0(x_{n+q_{t-1}}) & p_1'(x_{n+q_{t-1}}) & \cdots & p_{t+m-1}(x_{n+q_{m-1}}) \\ p_0'(x_n) & p_1'(x_n) & \cdots & p_{t+m-1}'(x_n) \\ \cdots & \cdots & \cdots & \cdots \\ p_0'(x_{n+q_{t-1}}) & p_1'(x_{n+q_{t-1}}) & \cdots & p_{t+m-1}'(x_{n+q_{t-1}}) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_{t+m-1} \end{pmatrix} = V^T \quad \textbf{(8)}$$

<u>Proposition</u>:

V, C, & D defined above, satisfy

(i) C = D^{-1}

(ii) $y(x) = V^T(D^{-1})^{\mathrm{T}}(P(x)$

Proof: See [9]

<u>Specified Method</u>:

We set $p_i(x) = x^i, i = 0,1,\ldots 4,, \ m = 4.$
$\{x_{n+q_1}, x_{n+q_2} \ldots\ x_{n+q_4}\}$ are the collocation points, then Equation (8) reduces to:

$$Da = \begin{pmatrix} 1 & x_n & x_n^2 & x_n^3 & \cdots & x_n^4 \\ 0 & 1 & 2x_{n+q_1} & 3x_{n+q_1} & \cdots & 4x_{n+q_1}^3 \\ 0 & 1 & 2x_{n+q_2} & & \cdots & 4x_{n+q_2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 2x_{n+q_4} & 3x_{n+q_4}^2 & \cdots & 4x_{n+q_4}^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \cdot \\ \\ a_4 \end{pmatrix} = \begin{pmatrix} y_n \\ f_{n+q_1} \\ \cdot \\ \\ f_{n+q_4} \end{pmatrix} \quad \textbf{(9)}$$

Using maple mathematical software, we obtain a continuous scheme of the form:

$$y(x) = V^T (D^{-1})^T p(x). \ldots\ldots\ldots\ldots \quad \textbf{(10)}$$

Now evaluating the continuous scheme at:

$$x_{n+q_1} = 0, \qquad x_{n+q_2} = \frac{1}{2} - \frac{\sqrt{5}}{10}, \qquad x_{n+q_3} = \frac{1}{2} - \frac{\sqrt{5}}{10}, x_{n+q_4} = 1$$

These are the transformed zeros of Lobatto polynomial $p_3'(x) = \frac{15}{2}\, x^2 - \frac{3}{2},$
$[p_3(x)$ is the Legendre polynomial of degree three, the transformation is defined as:

$T(x_i) = \frac{1}{2}[\, a\,(x_i - 1) + \; b(x_i + 1)]$ in [a,b], x_i are the zeros of the Lobatto polynomial].

We obtain discrete schemes as follow:

$$y_{n+q_1} = y_n$$

$$y_{n+q_2} = y_n + \left(\frac{11}{20} + \frac{\sqrt{5}}{10}\right) hf_{n+q_1} + \left(\frac{5}{24} - \frac{5}{120}\right) hf_{n+q_2} + \left(\frac{5}{24} - \frac{13\sqrt{5}}{120}\right) hf_{n+q_3} - \left(\frac{1}{120} - \frac{\sqrt{5}}{120}\right) hf_{n+q_4}$$

$$y_{n+q_3} = y_n + \left(\frac{11}{20} - \frac{\sqrt{5}}{10}\right) hf_{n+q_1} + \left(\frac{5}{24} - \frac{13}{120}\right) hf_{n+q_2} + \left(\frac{5}{24} + \frac{\sqrt{5}}{120}\right) hf_{n+q_3} - \left(\frac{1}{120} - \frac{\sqrt{5}}{120}\right) hf_{n+q_4}$$

$$y_{n+q_4} = y_n + \frac{1}{12} hf_{n+q_1} + \frac{5}{12} hf_{n+q_2} + \frac{5}{12} hf_{n+q_3} + \frac{1}{12} hf_{n+q_4} \qquad \ldots \qquad \textbf{(11)}$$

To convert to Runge–Kutta function evaluations, the discrete schemes (11) must satisfy the differential Equation (1). Thus, we have:

..(2.12)

$$\left.\begin{aligned}
y'_{n+q_1} &= f\left(x_{n+q_1}, y_{n+q_1}\right) = f(x_n, y_n) \\
y'_{n+q_2} &= f\left(x_{n+q_2}, y_{n+q_2}\right) = f(x_n + q_2 h, y_n + \left(\frac{11}{120} + \frac{\sqrt{5}}{120}\right) hf_{n+q_1} + \left(\frac{5}{24} - \frac{\sqrt{5}}{120}\right) hf_{n+q_2} + \left(\frac{5}{24} - \frac{13\sqrt{5}}{120}\right) hf_{n+q_3} - \left(\frac{1}{120} - \frac{\sqrt{5}}{120}\right) hf_{n+q_4} \\
y'_{n+q_3} &= f\left(x_{n+q_3}, y_{n+q_3}\right) = f(x_n + q_3 h, y_n + \left(\frac{11}{120} - \frac{\sqrt{5}}{120}\right) hf_{n+q_1} + \left(\frac{5}{24} + \frac{13\sqrt{5}}{120}\right) hf_{n+q_2} + \left(\frac{5}{24} + \frac{\sqrt{5}}{120}\right) hf_{n+q_3} - \left(\frac{1}{120} + \frac{\sqrt{5}}{120}\right) hf_{n+q_4} \\
y'_{n+q_4} &= f\left(x_{n+q_4}, y_{n+q_4}\right) = f\left(x_n + q_4 h,\; y_n + \frac{1}{12} hf_{n+q_1} + \frac{5}{12} hf_{n+q_2} + \frac{5}{12} hf_{n+q_3} + \frac{1}{12} hf_{n+q_4}\right)
\end{aligned}\right\} \textbf{(12)}$$

The weight of the method is $b = (b_1, b_2, b_3, b_4)$, the values of $bi,\ i = 1$, ...4 are obtained by evaluating the continuous scheme (2.10) at $x_n = 1,$ this we obtain:

$$b = \left(\frac{1}{12}, \ \frac{5}{12}, \ \frac{5}{12}, \frac{1}{12}\right) = (b_1,\ b_2,\ b_3\ , b_4) \qquad . \ . \ . \ \ ... \qquad \textbf{(13)}$$

Now putting $y'_{n+q_i} = f\left(x_{n+q_i}, y_{n+q_i}\right) = k_i, i = 0, 1, \ldots .4.$ in Equation (12), we obtain the Runge – Kutta Solution as:

$$y_{n+1} = y_n + \frac{1}{12} h \{ (k_1 + k_4) + 5\,(k_2 + k_3)\} \qquad \textbf{(14)}$$

Where the $k_i = f_{n+q_i}, i = 1, 2, \ldots 4.$

The method is summarized in the table (Butcher's Tableau).

Table 1: Butchers Tables.

C	A			
0	**0**	**0**	**0**	**0**
$\frac{1}{2} - \frac{\sqrt{5}}{10}$	$\left(\frac{11}{120} + \frac{\sqrt{5}}{120}\right)$	$\left(\frac{5}{24} - \frac{\sqrt{5}}{120}\right)$	$\left(\frac{5}{24} - \frac{13\sqrt{5}}{120}\right)$	$-\left(\frac{1}{120} - \frac{\sqrt{5}}{120}\right)$
$\frac{1}{2} + \frac{\sqrt{5}}{10}$	$\left(\frac{11}{120} - \frac{\sqrt{5}}{120}\right)$	$\left(\frac{5}{24} + \frac{13\sqrt{5}}{120}\right)$	$\left(\frac{5}{24} + \frac{\sqrt{5}}{120}\right)$	$-\left(\frac{1}{120} + \frac{\sqrt{5}}{120}\right)$
1	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$ .. **(15)**
	$\frac{1}{12}$	$\frac{5}{12}$	$\frac{5}{12}$	$\frac{1}{12}$

b

The table can also be rewritten in the form.

Table 2

C	A	U
	B	V

Where $C = (c_1, c_2, \ldots\ c_4)^T$, the abscissae, $A = (a_{ij})\ i_j, = 1,4,$ the coefficients of the method; $U = (1,1,1,1)^T,\ V = (1),\ b = (b_1, b_2,\ b_3, b_4).$

METHOD ANALYSIS

(i) **Consistency**: The Runge-Kutta method (15) is consistent since:

$$\sum_{j=1}^{4} a_{ij} = C_i, \sum_{i}^{4} b_i = 1 \qquad (See\ Butcher's\ Tableau).$$

(ii) The stability of the method is obtained by considering the linear test equation.

$$y' = \lambda y.$$

Putting Z = $\lambda h, h \in (0,1)$

The stability function is R (Z):

R (Z) = I + $Zb^T (I - ZA)^{-1} U$

I, is an identity matrix. The stability region (dom (R (Z)) is the set of points in the complex plane:

$$|R(z)| < 1.$$

The A – stability domain, dom (R (Z)) is:

$$R\ (Z) = \{\ Z: Re\ (Z) < o\ and\ |R\ (Z)\ | < 1\}$$

(iii) **Order and error constant**: The exact solution of (2.01) is defined as $y\ (x_{n+1})$ and the approximate Runge-Kutta solution is y_{n+1}.

Both solutions can be expanded into Taylors Series as:

$$y(x_{n+1}) = y(x_n + h) = y_n + \lambda_1 h y_n' + \lambda_2 h^2 y_4'' + \ \dots \ \lambda_n h^n y_n(n) \dots$$

and,

$$y_{n+1} = y_n + h \sum_{i=1}^{4} b_i k_i = y_n + h \sum_{i=1}^{4} b_i\ f(x_{n+q_i}, y_{n+q_i}) = y_n + h \sum_{i=1}^{4} b_i\ y'_{n+q_i}$$

$$i.e\ \ y_{n+1} = y_n + h \sum_{i=1}^{4} b_i\ y'(x_n + q_i h) = y_n + b_1 q_1 h w_1 y_n' + b_2 q_2^2 h^2 w_2 y_n''$$

$$+ \cdots b_n q_n^n h^n w_n h^n y_n(n) \dots$$

Now we define a linear difference operator $L(y_n(x), h)\ as$ $L(y_n(x), h) = y(x_{n+1}) - y_{n+1}$. Expanding and expressing into one Taylor's series we have:

$$L(y_n(x),h) = y(x_{n+1}) - y_{n+1} = C_0 y_n + C_1 h y_n' + C_2 h^2 y_n'' +..C_p h^p y_n(p) + C_{p+1} h^{p+1} y_n(p+1)\ldots \quad \textbf{(17)}$$

We found that:

$$c_0 = c_1 = c_2 \ldots c_p = 0, \quad c_{p+1} \neq 0, \quad p = 6.$$

Thus the Runge – Kutta method is of order $p = 6$ and error constant $c_{p+1} = c_7 = -\frac{1}{1512000}$ (see [3]).

ERROR ESTIMATION FORMULAR

<u>Theorem</u>: Any Runge-Kutta solution is of the form:

$$y_{n+1} = y_n + h \sum_{k=1}^{n} bk_i, \text{with step size vector } h, \text{has error estimate as } E_r$$

where

$$E_r = \frac{2^{p+1}+1}{2^{p+1}-1} \left(y_{n+1}^{\left(\frac{h}{2}\right)} - y_{n+1}^{(h)} \right) \quad \textbf{(18)}$$

Where $y_{n+1}^{(h)}$ *and* $y_{n+1}^{\left(\frac{h}{2}\right)}$ *are the Runge – Kutta Solutions with step sizes h and* $\frac{h}{2}$ *respectively, p is the order of the method.*

<u>Proof</u>: Let the approximate solutions with step sizes h *and* $\frac{h}{2}$ *respectively be*

$$y_{n+1}^{*} = y_{n+1}^{(h)} + C\, h^{(p+1)} + R_{n1}(x) - - - - \qquad (i)$$

and

$$y_{n+1}^{**} = y_{n+1}^{\left(\frac{h}{2}\right)} + C \left(\frac{h}{2}\right)^{p+1} + R_{n2}(x) \quad - - - - - \qquad (ii)$$

The constant C is independent of the chose of step sizes since any Runge-Kutta solution can be expanded into Taylors series and the terms agree with Taylor's series expansion up to order p, $R_{n1}(x)$, *and* $R_{n2}(x)$ are the reminder terms of each expansion. Now since the solution is unique, as $p \to \infty, y_{n+1}^{*} \to y_{n+1}^{**} = y\,(x_{n+1})$, $R_{n1}(x)$ *and* $R_{n2}(x)$ both approach zeros, they can be neglected for large p. thus (i) and (ii) can be written as:

$$y(x_{n+1}) = y_{n+1}^{(h)} + C\, h^{p+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (iii)$$

and

$$y(x_{n+1}) = y_{n+1}^{\left(\frac{h}{2}\right)} + C\left(\frac{h}{2}\right)^{p+1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (iv), for\ large\ p. \quad \text{(iv)}$$

Subtracting (iv) from (iii) we have:

$$o = \left(y_{n+1}^{(h)} - y_{n+1}^{\left(\frac{h}{2}\right)}\right) + C\left(h^{p+1} - \left(\frac{h}{2}\right)^{p+1}\right)$$

$$i.e \quad Ch^{\,p+1} = \left(\frac{2^p+1}{2^p+1-1}\right)\left(y_{n+1}^{\left(\frac{h}{2}\right)} - y_{n+1}^{(h)}\right)$$

Since $\frac{2^{p+1}}{2^{p+1}-1} \le \frac{2^p+1+1}{2^p+1-1} \quad \forall p,$

Then $Ch^{\,p+1} \le \frac{2^p+1}{2^{p+1}-1}\left(y_{n+1}^{\left(\frac{h}{2}\right)} - y_{n+1}^{(h)}\right).$

Take the least upper bound of $Ch^{\,p+1} = E_r$ as error estimate

We have $E_r = \frac{2^{p+1}+1}{2^{p+1}-1}\left(y_{n+1}^{(h)} - y_{n+1}^{\left(\frac{h}{2}\right)}\right)$. (Error estimation formular).

Numerical Experiments

We use three problems with exact or analytic solutions to test our formulas.

Notations:

$y(x_{n+1}) = \ exact\ or\ analytic\ solutions.$

$y_{n+1} = computational\ solutions\ with\ step\ size\ h.$

$y_{n+1}^{*} = y_{n+1}^{\left(\frac{h}{2}\right)} = computational\ solutions\ with\ step\ size\ \left(\frac{h}{2}\right).$

$E_r = y\,(x_{n+1}) - y_{n+1}(exact\ error)$

E_r^{*} = Calculated error, by method (18).

Example 1: $y' = 3y + sin(x), y(0) = \frac{1}{10} \quad h = .1,$

$$y(x) = -\frac{1}{10}(cosx + \ 3sinx) + \frac{1}{5}e^{3x}$$

Example 2: $y' = -20y + 20e^{-2x}, y(0), h = .01,$

$$y\ (x_i) = \frac{10}{9}\left(e^{-2x_i} - e^{-20x_i}\right)$$

Example 3: $y' = y^2 + x, \ \ y(0) = 1$

Analytic Solution – None

Table 3: Comparison of Solution of Problem 1.

x	$y\ (x)$	y_{n+1}	y^*_{n+1}	E_r	E^*_r
.1	0.140521319993349	0.140521320576694	0.140521320002440	5.83E-10	5.83E-10
.2	0.206816303055460	0.206816304632021	0.206816303080027	1.58E-9	1.58E-9
.3	0.307730911320427	0.307730914515468	0.307730911370216	3.20E-9	3.20E-9
.4	0.455091782454426	0.455091788209119	0.455091782544102	5.75E-9	5.75E-9
.5	0.664751896297314	0.664751906013248	0.664751896448720	9.72E-9	9.72E-9

Table 4: Comparison of Solution of Problem 2.

x	$y\ (x_i)$	y_{n+1}	y^*_{n+1}	E_r	E^*_r
.01	.179408800254	.179408800370	.179408800256	1.16E-10	1.16E-10
.02	.322743770130	.322743770319	.322743770132	1.89E-10	1.89E-10
.03	.436614330545	.436614330777	.436614330548	2.33E-10	2.33E-10
.04	.526430424744	.526430424998	.526430424748	2.54E-10	2.54E-10
.05	.596619974294	.596619974554	.596619974298	2.60E-10	2.60E-10

Table 5: Comparison of Solution of Problem 3.

x	$y\ (x)$	y_{n+1}	y^*_{n+1}	E_r	E^*_r
.1	-	.913794328431573	.913794328451718	-	2.05E-11
.2	-	.851191238284488	.851191238330888	-	4.71E-11
.3	-	.807621623289533	.807621623356392	-	6.79E-11
.4	-	.779807312531194	.779807312610653	-	8.07E-11
.5	-	.765280591342032	.765280591426961	-	8.63E-11

DISCUSSION / CONCLUSION

We used three test problems. The first two have analytic solutions while the last has none. The approximate solutions are highly efficient, the actual errors E_r *and computed errors* E_r^* are equal (see error tables). This implies that our error formula is accurate. The last example has no analytic or exact solutions, but we can deduce the level of accuracy from the calculated results, since our E_r^* is accurate we can deduce that $E_r^* = E_r$.

These methods help us to determine exact solutions of problems arising from mathematical models in science, engineering population etc., with no close–form or analytic solutions. Our method is more general and accurate than Feldberg method [2].

REFERENCES

1. Ehle, B.L. 1968. "High Order A – Stable Methods for Numerical Solution of System of ODEs". *BIT 8*. 276 – 278.

2. Feldberg, E. 1970. "Runge – Kutta Feldberg Method". *Computing*. 6:61 – 71.

3. Lambert, J.D. 1973. *Computational Methods in ODEs*. John Wiley & Sons: New York, NY.

4. Butcher, J.C. 1996. "A History of Runge – Kutta Methods". *Numerical Mathematics Zo*. 247-260.

5. Butcher, J.C. 1999. "John Butcher's Tutorials, implicit Runge – Kutta Methods". *Numerical Methods for Ordinary Differential Equations, Second Edition.* Widely & Sons Ltd.: London, UK.

6. Burrage, K. 1978. "A Special Family of Runge – Kutta Methods for Solving Stiff Differential Equations". *BIT*. 18:22-41.

7. Duk, P.L. 2000. "Computation of Gauss Type Quadrature Formula". J*ourn. of Computational applied Maths.* 61 – 72.

8. Lie, J., and S.P. Norsett. 1989. "Super Convergence for Multistep Collocation". *Math Comp*. (52):65 –79.

9. Yusuf, Y. and P. Onumanyi. 2002. "New Multiple FDMS, through Multistep Collocation for $y'' = f(x,y)$. *Journ. of Math. Assoc. of Nigeria*. 29(2):92 – 100.

SUGGESTED CITATION

Agam, S.A. 2017. "A Four-Stage Gauss-Lobatto Integral with Accurate Error Estimation Formula for ODES". *Pacific Journal of Science and Technology*. 18(2):78-87.

Pacific Journal of Science and Technology