http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 11 (9): 1141-1153, 2012
ISSN 1812-5638 / DOI: 10.3923/itj.2012.1141.1153

Local Truncation Error for the Parallel Runge-Kutta-Fifth Order Methods

[1]R. Ponalagusamy and [2]K. Ponnammal
[1]Department of Mathematics, National Institute of Technology, Tiruchirappalli-620 015, India
[2]Department of Mathematics, Periyar E.V.R. College, Tiruchirappalli-620 023, India

Abstract: We derive for the first time, the new mathematical expression for Local Truncation Error (LTE) by using the order trees for the parallel fifth order Runge-Kutta formulae while solving the initial value problem $y' = f(x\ y)$, $y(x_0) = y_0$. Based on the concept of tree theory, a modest effort has been made to obtain the error coefficients and derivatives of the LTE using the order trees and elementary differentials of the explicit parallel Runge-Kutta fifth order methods. It is shown that the new parallel scheme (PPRKF) compares well with the analytic solution and is often more accurate than other parallel methods (PRKF 1 and PRKF 2). With the help of numerical examples, the absolute error and estimated LTE have been computed.

Key words: Parallel Runge-Kutta fifth order methods, local truncation error, order conditions, order trees, elementary differentials

INTRODUCTION

It is well known that ordinary differential equations arise in many fields of scientific endeavour ranging from mathematical modeling of electron transport (Efurumibe *et al*., 2012), the problem of unsteady stagnation point flow and heat transfer (Nik Long *et al*., 2011), the flow and heat transfer characteristics of a visco-elastic fluid (Krishnambal and Anuradha, 2006), the steady and unsteady laminar flow of an electrically conducting viscous incompressible fluid between two parallel porous plates (Ganesh and Krishnambal, 2006, 2007), in the simulation of cellular neural networks (Ponalagusamy and Senthilkumar, 2008), in the Potato-Osma dehydration process (Mazaheri *et al*., 2005) and invariably arise if there is motion or growth (Burrage, 1995) and many such system of differential equations (Ponalagusamy, 2008), cannot be solved by analytical methods in a closed form solution and hence we go in for numerical methods.

Much research work has gone into the derivation and analysis of numerical methods of Initial Value Problems (IVPs) because of practical importance and the main class of method commonly employed in practice is the Runge-Kutta method (Butcher, 1987). Runge-Kutta-Butcher (RKB) methods are being applied to determine numerical solutions for the IVPs that arise in the fields of science and Engineering. Runge-Kutta method was originally derived from Runge in the year 1895 and extended by Heun and Kutta about the year 1901. The computational techniques and the subject of research of both the implicit and explicit Runge-Kutta methods were discussed elaborately by Butcher (1964, 1987), Hairer and Wanner (1974), Hairer (1981) and Hairer *et al*. (1993) and many others. Further approaches in Runge-Kutta methods are found by Khiyal and Rashid (2005) and Bazuaye (2006). It is of interest to note that increasing availability of parallel computers has recently spurred a substantial amount of research concerned with the possibilities for exploiting parallelism in numerical solution of Initial Value Problems (IVPs) for ordinary differential equations. The preliminary surveys of parallel methods for IVPs and the need for parallel computation in IVPs were provided by Gear (1986), Jackson (1991) and Jackson and Norsett (1995). Burrage (1995) extended the surveys of parallel methods for IVPs using Runge-Kutta-Butcher methods. Evans and Sanugi (1989) and Katti and Srivastava (2003) have proposed parallel algorithms for Runge-Kutta formulas. The construction of variable step-size approach for class of general linear methods for stiff and nonstiff systems in sequential and parallel environment was given by Bazuaye and Ataha (2006). The stability of the Runge-Kutta methods was analysed and elaborated by Butcher (1987). Ponalagusamy and Ponnammal (2008) described the stability regions of the existing and the new parallel Runge-Kutta-fifth order methods.

In regard of the error analysis of the Runge-Kutta methods, Lagrange has given the error bounds for the Taylor polynomials in the year 1797 and Cauchy derived bounds for the Euler polygons in the year 1824. Kutta and Runge derived error coefficients in the year 1901 and 1905, respectively for the fourth order classical Runge-Kutta

Corresponding Author: R. Ponalagusamy, Department of Mathematics, National Institute of Technology, Tiruchirappalli-620 015, India Tel: +91 431 2503664 Fax: + (91) (431) 2500133

method. Bieberbach used the Taylor series to find the error bound in 1951. Evans and Yaakub (1996), provided error estimates and error control for fifth order weighted Arithmetic mean Runge-Kutta formula by using Taylor expansion for the initial value problem $y' = f(y)$, $y(x_0) = y_0$.

The local truncation error coefficients and formulas for Runge-Kutta methods by using the order trees and elementary differentials were discussed elaborately by Hairer *et al.* (1993). The order trees and elementary differentials related to the derivatives in the Taylor series expansion were elaborated by Butcher (1987) and Hairer *et al.* (1993). According to Butcher (2010a, b), trees play a central role in the theory of Runge-Kutta methods and they also have applications to more general methods, involving multiple values and multiple stages.

In this study, we derive the Local Truncation Error (LTE) for the parallel Runge-Kutta fifth order methods for the initial value problem $y' = f(x, y)$, $y(x_0) = y_0$, by using the order trees and elementary differentials. The LTE is obtained for the examples and compared with the absolute error at each step.

ORDINARY DIFFERENTIAL EQUATIONS AND RUNGE-KUTTA METHODS

Ordinary differential equations can be represented in one of two ways. The first is known as non-autonomous form. The Ordinary Differential Equation (ODE) is written as:

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

The variable x is called the independent variable and y (x) is the solution to the differential equation. It should be noted that y (x) can be a vector-valued function, going from $R \to R^m$, where m is the dimension of the differential equation.

In the second form, y' (x) does not depend directly on x, except as a parameter of y (x). A big advantage is obtained by transforming Eq. 1 to an autonomous form by appending x to the dependent variables as:

$$\begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} 1 \\ f(x, y) \end{bmatrix} \tag{2}$$

Any non-autonomous system may be written in autonomous form by adding the equation $x' = 1$ to the system. If we add the initial condition $y_0 = y(x_0)$ to the system of equations we get the Initial Value Problem (IVP):

$$y'(x) = f(x, y(x)),\ y_0 = y(x_0) \tag{3}$$

Definition 1: Let s be an integer (the number of stages) and $a_{21}, a_{31}, a_{32}, \ldots, a_{s1}, a_{s2}, \ldots, a_{s,s-1}, b_1, b_2, b_3, \ldots, b_s, c_2, \ldots, c_s$ be real coefficients. Then the method:

$$\begin{aligned} k_1 &= f(x_0, y_0) \\ k_2 &= f(x_0+c_2h,\ y_0+ha_{21}k_1) \\ k_3 &= f(x_0+c_3h,\ y_0+h(a_{31}k_1+a_{32}k_2)) \\ &\ldots \\ &\ldots \\ &\ldots \\ k_s &= f(x_0+c_sh,\ y_0+h(a_{s1}k_1+\ldots+a_{s,s-1}k_{s-1})) \\ y_1 &= y_0+h(b_1k_1+\ldots+b_sk_s) \end{aligned} \tag{4}$$

where, c_i satisfy the conditions:

$$c_2 = a_{21},\ c_3 = a_{31}+a_{32},\ \ldots,\ c_s = a_{s1}+\ldots+a_{s,s-1} \tag{5}$$

or:

$$c_i = \sum_{j=1}^{i-1} a_{ij} \tag{6}$$

is called an s-stage explicit Runge-Kutta method for Eq. 1.

DISCUSSION OF ORDER CONDITIONS OF EXPLICIT RUNGE-KUTTA METHODS

To determine the coefficients of Runge-Kutta methods, it is necessary to derive the order conditions. For the order conditions we must compute the derivatives of $y_1 = y_1(h)$ for $h = 0$ and compare them with those of the true solution for order 1, 2, 3 and 4 etc., depending on the order of the method. Butcher (1987) and Hairer *et al.* (1993) used the rooted trees and elementary differentials to determine the order conditions instead of expanding into Taylor series as deriving the order conditions is more complicated in higher order cases.

The main difficulty in the derivation of the order conditions using rooted trees is to understand the correspondence of the formulas to certain rooted labeled trees. The components of vectors are denoted by superscript indices as in tensor notation (Hairer *et al.*, 1993). Then Eq. 2 can be written as:

$$(y^J)' = f^J(y^1, \ldots, y^n),\ J = 1, \ldots n \tag{7}$$

We next rewrite the Runge-Kutta method Eq. 4 for the autonomous differential Eq. 7. In order to get a better symmetry in all formulas of Eq. 4, we replace k_i by the argument g_i such that $k_i = f(g_i)$.

Then Eq. 4 becomes:

$$g_i^J = y_0^J + \sum_{j=1}^{i-1} a_{ij} hf(g_j^1,...,g_j^n) \quad i = 1,2,...,s$$
$$y_1^J = y_0^J + \sum_{j=1}^{s-1} b_j hf^J(g_j^1,...,g_j^n) \tag{8}$$

If the system Eq. 7 originates from Eq. 2, then, for J = 1:

$$g_i^1 = y_0^1 + \sum_{j=1}^{i-1} a_{ij} h = x_0 + c_i h$$

by Eq. 5. We see that if Eq. 5 is satisfied, then for the derivation of order conditions only the autonomous Eq. 7 has to be considered.

As indicated in the beginning of this section, we have to compare the Taylor series of y_1^J with that of the exact solution. Therefore, we compute the derivatives of y_1^J and g_i^J with respect to h for h = 0.

Trees and elementary differentials: The continuation of the process of Taylor series gives rise to very complicated formulas. It is therefore, advantageous to use a graphical representation (Butcher, 1987). For a convenient development of the order of a method, the basic tree theory is introduced. A tree is a rooted graph which contains no circuits. The symbol τ is used to represent the tree with only one vertex. All rooted trees can be represented using τ and the operation $[t_1, \ldots, t_m]$.

Definition 2: Let A be an ordered chain of indices A = {j<k<l<m<...} and denote by A_q the subset consisting of the first q indices. A (rooted) labeled tree of order q (q ≥ 1) is a mapping (the son-father mapping):

$$t: A_q \backslash \{j\} \rightarrow A_q$$

such that $t(z)<z$ for all $z \in A_q \backslash \{j\}$. The set of all labeled trees of order q is denoted by LTq (Hairer *et al*., 1993).

Definition 3: The order of the tree $t \in LT_q$ is defined by:

$$q = r(t) = \begin{cases} 1, \text{if } t = \tau \\ 1 + r(t_1) + \cdots + r(t_m), \text{if } t = [t_1, \ldots, t_m] \end{cases}$$

That is, the order of a tree is the number of vertices the tree has. The number of trees up to order 10 can be seen in Table 1. The height of a tree is k-1, where k is the number of vertices in the longest path beginning with the root (Butcher, 1987).

Definition 4: The density of the tree $t \in LT_q$ is defined by:

$$\gamma(t) = \begin{cases} 1, \text{if } t = \tau \\ r(t)\gamma(t_1)\gamma(t_2)\cdots\gamma(t_m), \text{if } t = [t_1, \ldots, t_m] \end{cases}$$

A simple way of finding the density of a tree is to attach to each vertex a number that is equal to the number of vertices above it plus one. The density is then equal to the product of the numbers attached to the vertices.

Definition 5: For a labeled tree $t \in LT_q$ we call:

$$F^J(t)(y) = \sum_{K,L,\ldots} f^J_{K,\ldots}(y) f^K_{\ldots}(y) f^L_{\ldots}(y) \ldots$$

the corresponding elementary differential. The various terms in the elementary differential have a structure related to rooted trees. The summation is over q-1 indices K, L, ... (which correspond to $A_q \backslash \{j\}$) and the summand is a product of q f ' s, where the upper index runs through all vertices of t and the lower indices are the corresponding sons. We also denote by F (t) (y) the vector (F^1 (t) (y), ..., F^n (t) (y)) (Hairer *et al*., 1993).

Definition 6: The set of all trees of order q is denoted by T_q. Also α (t), for $t \in T_q$, is the number of elements in the equivalence class t; i.e., the number of possible different monotonic labellings of t.

Definition 7: Let t be a labeled tree with root j. We denote the elementary weight T_j (t) for stage j by:

$$\Phi_j([t_1, t_2, \ldots, t_m]) = \sum_{K,l,\ldots} a_{jk} a_{\ldots} \ldots \tag{9}$$

the sum over the q-1 remaining indices k, l,... as in Definition 5. The summand is a product of q-1 a's, where all fathers stand two by two with their sons as indices.

The Taylor expansion of the exact solution: The general result for the qth derivative of the true solution is as follows:

Theorem 1: The exact solution of Eq. 7 satisfies:

$$(y)^{(q)}(x_0) = \sum_{t \in LT_q} F(t)(y_0) = \sum_{t \in T_q} \alpha(t) F(t)(y_0) \tag{10}$$

Table 1: Order trees, order conditions and elementary differentials of order 6 (q = 6)

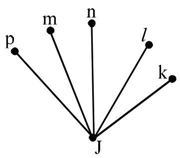
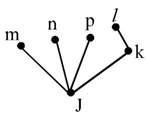
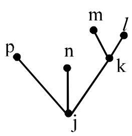
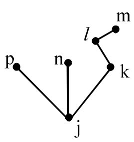
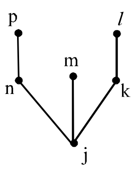
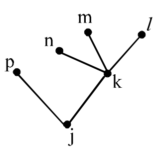
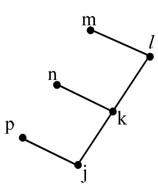
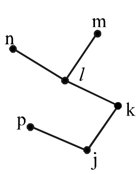
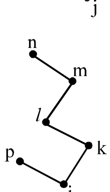
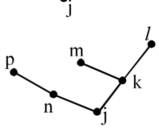
T	Graph	ϒ(t)	α(t)	F(t)(y)	$\Phi_j(t)$	Order conditions $\Sigma b_j\Phi_j(t) = 1/\Upsilon(t)$
t_{61}		6	1	$\sum f_{KLMNP} f^K f^L f^M f^N f^P$	$\sum a_{jk} a_{jl} a_{jm} a_{jn} a_{jp}$	$\sum b_j c_j^5 = \frac{1}{6}$
t_{62}		12	10	$\sum f^J_{KMNP} f^K_L f^L f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{jm} a_{jn} a_{jp}$	$\sum b_j c_j^3 a_{jk} c_k = \frac{1}{12}$
t_{63}		18	10	$\sum f^J_{KNP} f^K_{ML} f^L f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{km} a_{jn} a_{jp}$	$\sum b_j c_j^2 a_{jk} c_k^2 = \frac{1}{18}$
t_{64}		36	10	$\sum f^J_{KNP} f^K_L f^L_M f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{jn} a_{jp}$	$\sum b_j c_j^2 a_{jk} \alpha_{kl} c_l = \frac{1}{36}$
t_{65}		24	15	$\sum f^J_{KMN} f^K_L f^L f^M f^N_P f^M f^P$	$\sum a_{jk} a_{kl} a_{Jn} a_{np} a_{jm}$	$\sum b_j c_j a_{jk} c_k a_{jn} c_n = \frac{1}{24}$
t_{66}		24	5	$\sum f^J_{KP} f^K_{LMN} f^L f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{km} a_{kn} a_{jp}$	$\sum b_j a_{jk} c_k^3 c_n = \frac{1}{24}$
t_{67}		48	15	$\sum f^J_{KP} f^K_{NL} f^L_M f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{kn} a_{jp}$	$\sum b_j c_j a_{jk} c_k a_{kl} c_l = \frac{1}{48}$
t_{68}		72	5	$\sum f^J_{KP} f^K_L f^L_{MN} f^M f^N f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{ln} a_{jp}$	$\sum b_j c_j a_{jk} a_{kl} c_l^2 = \frac{1}{72}$
$t_{6,9}$		144	5	$\sum f^J_{KP} f^K_L f^L_M f^M_N f^N f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{mn} a_{jp}$	$\sum b_j c_j a_{jk} a_{kl} a_{lm} c_m = \frac{1}{144}$
$t_{,10}$		36	10	$\sum f^L_{KN} f^K_L f^L_M f^M f^N_P f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{jn} a_{jp}$	$\sum b_j a_{jn} c_n a_{jk} a_{kl} c_l = \frac{1}{72}$

Table 1: Continue

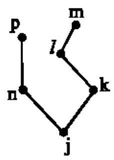
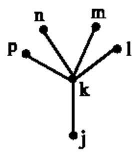
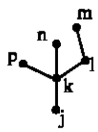
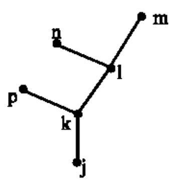
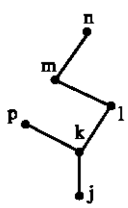
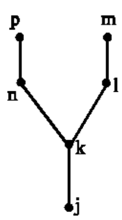
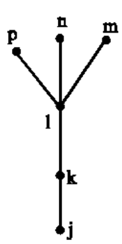
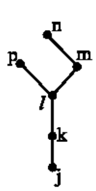
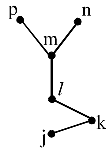
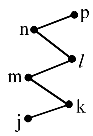
T	Graph	$\Upsilon(t)$	$\alpha(t)$	$F(t)(y)$	$\Phi_j(t)$	Order conditions $\Sigma b_j\Phi_j(t) = 1/\Upsilon(t)$
$t_{6,11}$		72	10	$\sum f^{L}_{KN}\, f^{K}_{ML}\, f^{N}_{P}\, f^{L} f^{M} f^{P}$	$\sum a_{jk} a_{kl} a_{km} a_{jn} a_{np}$	$\sum b_j a_{jn} c_n a_{jk} c_k^2 = \frac{1}{36}$
$t_{6,12}$		30	1	$\sum f^{J}_{K}\, f^{K}_{LMNP}\, f^{L} f^{M} f^{N} f^{P}$	$\sum a_{jk} a_{kl} a_{km} a_{kn} a_{kp}$	$\sum b_j a_{jk} c_k^4 = \frac{1}{30}$
$t_{6,13}$		60	6	$\sum f^{J}_{K}\, f^{K}_{L}\, f^{L}_{M}\, f^{M} f^{N} f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{kn} a_{kp}$	$\sum b_j a_{jk} c_k^2 a_{kl} c_l = \frac{1}{60}$
$t_{6,14}$		90	4	$\sum f^{J}_{K}\, f^{K}_{LP}\, f^{L}_{MN}\, f^{M} f^{N} f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{ln} a_{kp}$	$\sum b_j a_{jk} c_k a_{kl} c_l^2 = \frac{1}{90}$
$t_{6,15}$		180	4	$\sum f^{J}_{K}\, f^{K}_{LP}\, f^{L}_{M}\, f^{M}_{N}\, f^{N} f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{mn} a_{kp}$	$\sum b_j a_{jk} c_k a_{kl} a_{lm} c_m = \frac{1}{180}$
$t_{6,16}$		120	3	$\sum f^{J}_{K}\, f^{K}_{LN}\, f^{L}_{M}\, f^{M} f^{N}_{P}\, f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{kn} a_{kp}$	$\sum b_j a_{jk} a_{kn} c_n a_{kl} c_l = \frac{1}{120}$
$t_{6,17}$		120	1	$\sum f^{J}_{K}\, f^{K}_{L}\, f^{L}_{MNP}\, f^{M} f^{N}_{P}\, f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{ln} a_{lp}$	$\sum b_j a_{jk} a_{kl} c_l^3 = \frac{1}{120}$
$t_{6,18}$		240	3	$\sum f^{J}_{K}\, f^{K}_{L}\, f^{L}_{MP}\, f^{M}_{N}\, f^{N} f^{P}$	$\sum a_{jk} a_{kl} a_{lm} a_{mn} a_{lp}$	$\sum b_j a_{jk} a_{kl} c_l a_{lm} c_m = \frac{1}{240}$

Table 1: Continue

T	Graph	$\Upsilon(t)$	$\alpha(t)$	$F^J(t)(y)$	$\Phi_j(t)$	Order conditions $\Sigma b_j\Phi_j(t) = 1/\Upsilon(t)$
$t_{6,19}$		360	1	$\sum f^J_K\, f^K_L\, f^L_M\, f^M_{NP}\, f^N f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{mn} a_{mp}$	$\sum b_j a_{jk} a_{kl} a_{lm} c^2_m = \frac{1}{360}$
$t_{6,20}$		720	1	$\sum f^J_K\, f^K_L\, f^L_M\, f^M_N\, f^N_P\, f^P$	$\sum a_{jk} a_{kl} a_{lm} a_{mn} a_{np}$	$\sum b_j a_{jk} a_{kl} a_{lm} a_{mn} a_{np} = \frac{1}{720}$

For the proof, refer to Hairer *et al*. (1993).

Theorem 2: Hairer *et al.* (1993): The derivatives of g_i satisfy:

$$g_i^{(q)}|_{h=0} = \sum_{t\in LT_q} \gamma(t) \sum_j a_{ij}\, \Phi_j(t) F(t)(y_0) \tag{11}$$

The Runge-Kutta solution y_1 of Eq. 8 satisfies:

$$y_1^{(q)}|_{h=0} = \sum_{t\in LT_q} \alpha(t)\gamma(t) \sum_j b_j\, \Phi_j(t) F(t)(y_0) \tag{12}$$

Comparing Theorem 1 and Theorem 2 to match the Taylor series, we obtain the following Theorem.

Theorem 3: A Runge-Kutta method Eq. 4 is of order p iff:

$$\sum_{j=1}^{s} b_j\, \Phi_j(t) = \frac{1}{\gamma(t)} \tag{13}$$

for all trees of order ≤p (Hairer *et al*., 1993).

Therefore, For any Runge-Kutta method of order p, if we construct the order trees, elementary differentials, elementary weight function and other functions related to order tree structure such as α (t), γ (t) and r (t) as in Definitions 2-7, then Eq. 13 in Theorem 3 represent the order conditions for any Runge-Kutta method of order p along with Eq. 6 by not actually expanding into Taylor series. Finding the LTE for parallel fifth order Runge-Kutta method using the tree functions is explained in section 4 which can be extensible to higher order methods.

The local truncation error (LTE): For higher order methods, obtaining rigorous error bounds is unpractical and therefore it is necessary to consider the first nonzero term in the Taylor expansion of the error. For autonomous equations Eq. 3, the error term is best obtained by subtracting the Taylor series and using Eq. 10 in Theorem 1 and Eq. 12 in Theorem 2.

Theorem 4: If the Runge-Kutta method is of order p and if f is (p+1)-times continuously differentiable, then the LTE can be obtained from:

$$y^J(x_0+h) - y_1^j = \frac{h^{p+1}}{(p+1)} \sum_{t\in T_{p+1}} \alpha(t)\, e(t) F^J(t)(y_0) + O(h^{p+2}) \tag{14}$$

Where:

$$e(t) = 1 - \gamma(t) \sum_{j=1}^{s} b_j\, \Phi_j(t) \tag{15}$$

and γ(t), α(t) and $\Phi_j(t)$ are given in Definitions 4, 6 and 7. The expressions e (t) are called the error coefficients (Hairer *et al*., 1993).

THE PARALLEL RUNGE-KUTTA FIFTH ORDER METHODS

Parallel machines are computers with more than one processor and this facility might help us to speed up the computations in ordinary differential equations. In this section, we present formulas for six stage fifth order parallel versions of Runge-Kutta methods.

Definition 8: s-stage p-parallel q-processor Runge-Kutta method: In a Runge-Kutta method, if the Runge-Kutta matrix, A, consists of p×p blocks with each block of dimension atmost q and A strictly block lower triangular, then such a method is called an s-stage, p-parallel, q-processor explicit method and the parallelism arises from the fact that up to q processors can be used to compute the stages within a block concurrently (Jackson and Norsett, 1995).

Parallel Runge-Kutta-fifth order method 1 (PRKF 1): The following is the existing 6-stage 5th order 5-parallel

2-processor parallel Runge-Kutta-fifth order algorithm (Jackson and Norsett, 1995) (selecting a_{43} = 0 so that k_3 and k_4 can be evaluated simultaneously):

$$\begin{aligned}
K_1 &= hf(x_n, y_n)\\
K_2 &= hf(x_n + \frac{2h}{5}, y_n + \frac{2h_1}{5})\\
K_3 &= hf(x_n + \frac{h}{4},\ y_n + \frac{11}{64}k_1 + \frac{5}{64}k_2)\\
K_4 &= hf(x_n + \frac{h}{2},\ y_n + \frac{3}{16}k_1 + \frac{5}{16}k_2)\\
K_5 &= hf(x_n + \frac{3h}{4},\ y_n + \frac{9}{32}k_1 - \frac{27}{32}k_2 + \frac{3}{4}k_3 + \frac{9}{16}k_4)\\
K_6 &= hf(x_n + h,\ y_n - \frac{9}{28}k_1 + \frac{35}{28}k_2 - \frac{12}{7}k_4 + \frac{8}{7}k_5)\\
y_{n+1} &= y_n + \frac{1}{90}(7k_1 + 35k_3 + 12k_4 + 32k_5 + 7k_6)
\end{aligned} \tag{16}$$

Parallel Runge-Kutta-fifth order method 2 (PRKF 2): The following is the existing 6-stage 5th order 5-parallel 2-processor parallel Runge-Kutta-Fifth order algorithm (Jackson and Norsett, 1995) (selecting a_{65} = 0 so that k_5 and k_6 can be evaluated simultaneously):

$$\begin{aligned}
K_1 &= hf(x_n, y_n)\\
K_2 &= hf(x_n + \frac{h}{3}, y_n + \frac{k_1}{3})\\
K_3 &= hf(x_n + \frac{2h}{5},\ y_n + \frac{4}{25}k_1 + \frac{6}{25}k_2)\\
K_4 &= hf(x_n + \frac{h}{2},\ y_n + \frac{1}{4}k_1 - 3k_2 + \frac{15}{4}k_3)\\
K_5 &= hf(x_n + \frac{2h}{3},\ y_n + \frac{6}{81}k_1 - \frac{90}{81}k_2 - \frac{50}{81}k_3 + \frac{8}{81}k_4)\\
K_6 &= hf(x_n + \frac{4h}{5},\ y_n - \frac{6}{75}k_1 + \frac{36}{75}k_2 + \frac{10}{75}k_3 + \frac{8}{75}k_4)\\
y_{n+1} &= y_n + \frac{1}{192}(23k_1 + 125k_3 - 81k_5 + 125k_6)
\end{aligned} \tag{17}$$

Proposed parallel Runge-Kutta-fifth order method (PPRKF): The following is the new 6-stage 5th order 5-parallel 2-processor parallel Runge-Kutta-fifth order algorithm (Ponalagusamy and Ponnammal, 2008) (selecting a_{54} = 0 so that k_4 and k_5 can be evaluated simultaneously):

$$\begin{aligned}
K_1 &= hf(x_n, y_n)\\
K_2 &= hf(x_n + \frac{h}{5}, y_n + \frac{k_1}{5})\\
K_3 &= hf(x_n + \frac{2h}{5},\ y_n + \frac{39}{160}k_1 + \frac{5}{32}k_2)\\
K_4 &= hf(x_n + \frac{h}{2},\ y_n + \frac{1}{24}k_1 - \frac{5}{24}k_2 + \frac{2}{3}k_3)\\
K_5 &= hf(x_n + \frac{3h}{16},\ y_n + \frac{1}{8}k_1 - \frac{3}{16}k_2 + \frac{1}{4}k_3)\\
K_6 &= hf(x_n + h, y_n - \frac{9}{14}k_1 + \frac{15}{14}k_2 + \frac{8}{7}k_3 - \frac{12}{7}k_4 + \frac{8}{7}k_5)\\
y_{n+1} &= y_n + \left(-\frac{17}{306}k_1 - \frac{250}{153}k_3 + \frac{442}{255}k_4 + \frac{8192}{9945}k_5 + \frac{31}{234}k_6\right)
\end{aligned} \tag{18}$$

We have made use of the order trees and order conditions to find out the LTE for the parallel Runge-Kutta-fifth order algorithms. We have computed $\alpha(t)$, $\gamma(t)$ and $\Phi_j(t)$ which are depicted in Table 1. Table 1 also presents the order conditions related to trees $t \in LT_q$ for deriving LTE for the parallel fifth order Runge-Kutta methods presented in this study. Each subscript j, k,... is summed from 1 to s, the stage of the method. The expressions e(t) are called the error coefficients defined as in Eq. 15 and these coefficients are calculated using γ (t) and (a_{ij}) matrix and b_j's from the following Runge-Kutta-fifth order methods and presented in Table 2.

Description for obtaining the error coefficients: To obtain the LTE of the parallel fifth order methods we need the 20 order trees of methods of order 6 which are listed here in Table 1 are used. We are making use of the

Table 2: The error coefficients for the parallel Runge-Kutta-fifth order methods given by Eq. 16-18

Error coefficient e(t)	PPRKF	PRKF 1	PRKF 2
e_{61}	$\frac{-33}{1600}$	0	$\frac{1}{75}$
e_{62}	$\frac{125221}{309400}$	$\frac{27}{100}$	$\frac{1}{75}$
e_{63}	$\frac{123477}{61880}$	$\frac{101}{200}$	$\frac{1}{25}$
e_{64}	$\frac{3919457}{6188000}$	$\frac{49}{250}$	$\frac{1}{150}$
e_{65}	$\frac{-2371833}{43316000}$	$\frac{99}{250}$	$\frac{1}{75}$
e_{66}	$\frac{63193833}{123760000}$	$\frac{643}{5000}$	$\frac{-109}{225}$
e_{67}	$\frac{11999}{7140}$	$\frac{91}{200}$	$-\frac{1}{25}$
e_{68}	$\frac{29652313129}{23390640000}$	$\frac{66172}{118125}$	$\frac{1}{150}$
e_{69}	$\frac{147593}{77350}$	$\frac{53}{125}$	0
$e_{6,10}$	$\frac{153}{91}$	1	$-\frac{1}{5}$
$e_{6,11}$	$\frac{1000511}{2165800}$	$\frac{173}{200}$	$\frac{1}{25}$
$e_{6,12}$	$\frac{1633617}{1732640}$	$\frac{31}{40}$	$-\frac{3}{5}$
$e_{6,13}$	$\frac{6855256693}{23761920000}$	$\frac{4}{625}$	$-\frac{1}{10}$
$e_{6,14}$	$\frac{1945639}{1697280}$	$\frac{29}{40}$	$\frac{9}{20}$
$e_{6,15}$	$\frac{188771}{116025}$	$\frac{381}{1000}$	$\frac{1}{5}$
$e_{6,16}$	$\frac{78291}{49504}$	$\frac{23}{50}$	0
$e_{6,17}$	$\frac{16143}{11648}$	$-\frac{1}{8}$	$-\frac{1}{2}$
$e_{6,18}$	$\frac{122}{91}$	$\frac{1}{8}$	$\frac{1}{5}$
$e_{6,19}$	$\frac{457}{364}$	$-\frac{1}{5}$	0
$e_{6,20}$	1	1	1

elementary weight function $\Phi_j(t)$ and density function γ (t) in the Table 1 and by the formula given in Eq. 14 and 15.

From Eq. 15:

$$e(t) = 1-\gamma(t)\sum_{j=1}^{s} b_j\,\Phi_j(t)$$

The b_j's and a_{ij}'s are defined in Eq. 16-18 for the parallel Runge-Kutta fifth order methods, respectively.

For example, to find the error coefficient e_{61} for the parallel Runge-Kutta fifth order method 1 given in Eq. 16:

$$e_{61}=1-\gamma(t)\sum_{j=1}^{6} b_j\,\Phi_j(t)$$

where, $\Phi_j(t) = \Sigma\, a_{jk}a_{jl}a_{jm}a_{jn}a_{jp}$.

Here, the indices j takes value from 1 to 6 and k, l, m, n, p vary from 1 to 5.

By making use of the coefficients:

$$a_{21}=\frac{2}{5}, a_{31}=\frac{11}{64}, a_{32}=\frac{5}{64}, a_{41}=\frac{3}{16}, a_{42}=\frac{5}{16}, a_{43}=0, a_{51}=\frac{9}{32}, a_{52}=\frac{27}{32},$$
$$a_{53}=\frac{3}{4}, a_{54}=\frac{9}{16}, a_{61}=\frac{9}{28}, a_{62}=\frac{35}{28}, a_{63}=0, a_{64}=-\frac{12}{7}, a_{65}=\frac{8}{7}, b_1=\frac{7}{90},$$
$$b_2=0, b_3=\frac{32}{90}, b_4=\frac{12}{90}, b_5=\frac{32}{90}, b_6=\frac{7}{90}$$

We obtain, $e_{61}(t) = 0$.Similarly, The error coefficients of the parallel Runge-Kutta fifth order methods are obtained and given in Table 3.

Description for obtaining the derivative terms: We have obtained the derivative terms using the formula $F^J(t)$ given in Table 1. For example, to find the first derivative term, i.e., for $t\epsilon T_{61}$, by making use of $f^1 = 1$, $f^2 = f$:

$$F^J(t)(y) = \Sigma f^J_{KLMNP} = f_{xxxxx}+4\,f_{xxxxy}+f_{yxxx}\,f+10f_{xxxyy}f^2+10\,f_{xxyyy}f^3+5\,f_{xyyyy}f^4+f_{yyyyy}f^5$$

upon simplification. Similarly, all the other derivative terms are obtained which are upon simplification given as follows:

- For $t\epsilon t_{61}$: $f_{xxxxx}+4\,f_{xxxxy}+f_{yxxx}\,f+10f_{xxxyy}f^2+10\,f_{xxyyy}f^3+5\,f_{xyyyy}f^4+f_{yyyyy}f^5$
- For $t\epsilon t_{62}$: $f_{yxxx}\,f_x+3\,f_{xxyy}\,f_x\,f+f_{xyyy}f_xf+f_{xxxy}\,f_yf+2\,f_{xyyy}\,f_x\,f^2+3\,f_{xxyy}\,f_y\,f^2+3\,f_{yyyx}f_yf^3+f_{yyyy}f_yf^4$
- For $t\epsilon t_{63}$: $f_{xxy}f_{xx}+2\,f_{xyy}f_{xx}f+2\,f_{xxy}f_{xy}f+f_{yyy}\,f_{xx}f+4\,f_{xyy}f_{xy}f^2+f_{xxy}\,f_{yy}f^2+2\,f_{xyy}f_{yy}f^3+2\,f_{yyy}f_{xy}f^3+f_{yyy}\,f_{yy}f^4$
- For $t\epsilon t_{64}$: $f_{xxy}f_xf_y+2\,f_{xyy}f_xf_yf+f_{xxy}f_y^2f+2\,f_{xyy}f_y^2f^2+f_{yyy}f_yf_xf^2+f_{yyy}f_y^2f^3$
- For $t\epsilon t_{65}$: $f_{xyy}f_x^2+2\,f_{xyy}\,f_x\,f_y\,f+f_{yyy}f_x^2f+2\,f_{yyy}\,f_x\,f_y\,f^2+f_{xyy}f_y^2f^2+f_{yyy}f_y^2f^3$
- For $t\epsilon t_{66}$: $f_{xy}\,f_{xxx}+f_{yy}f_{xxx}f+3\,f_{xy}f_{xxy}f+2\,f_{yy}f_{xxy}f^2+3f_{xy}\,f_{xyy}f^2+f_{yy}\,f_{xxy}f^2+3\,f_{yy}f_{xyy}f^3+f_{xy}f_{yyy}f^3+f_{yy}f_{yyy}f^4$
- For $t\epsilon t_{67}$: $f_{xy}^2f_x+2\,f_{yy}\,f_{xy}f_xf+f_{xy}^2f_yf+f_{xy}f_{yy}f_y^2f^2+f_{yy}^2\,f_xf^2+f_{yy}f_{xy}\,f_yf^2+f_{yy}^2f_yf^3$
- For $t\epsilon t_{68}$: $f_{yy}f_yf_{xx}f+2\,f_{xy}^2\,f_yf+f_{xy}f_yf_{yy}f^2+2\,f_{yy}\,f_{xy}\,f_yf^2+f_{yy}^2f_yf^3$
- For $t\epsilon t_{69}$: $f_{xy}f_y^2f_x+f_{xy}f_y^3\,f+f_{yy}f_y^2f_x\,f+f_{yy}f_y^3f^2$
- For $t\epsilon t_{6,10}$: $f_{xx}f_{yy}f_x+f_{xx}\,f_{yy}f_yf+2\,f_{yy}f_{xy}f_xf+f_{yy}^2f_xf^2+2f_{yy}f_{xy}f_yf^2+f_{yy}^2f_yf^3$
- For $t\epsilon t_{6,11}$: $f_{yy}f_yf_x^2+2\,f_{yy}f_y^2f\,f_x+f_{yy}f_y^3f^2$
- For $t\epsilon t_{6,12}$: $f_yf_{xxxx}+4\,f_y\,f_{xxxy}f+4\,f_yf_{xxyy}f^2+f_xf_{xxyy}f^2+f_y\,f_{xxyy}f^2+3\,f_yf_{xyyy}f^3+f_x\,f_{xyyy}\,f^3+f_yf_{yyyy}f^4$
- For $t\epsilon t_{6,13}$: $f_y^2f_x+3\,f_y^2\,f_xf+f_y^3f+2\,f_y^3f^2+f_y^3f^3$
- For $t\epsilon t_{6,14}$: $f_yf_{yx}f_{xx}+f_yf_{yy}f_{xx}f+2\,f_y\,f_{yx}^2f+f_yf_{yx}f_{yy}f^2+2\,f_yf_{yy}f_{xy}f^2+f_yf_{yy}^2f^3$
- For $t\epsilon t_{6,15}$: $f_y^3f_{xx}f^2+f_yf_{xy}f_x^2f+f_y^2f_{xy}f_x+f_yf_{xy}f^2f+f_y^2f_{yy}f_xf+f_y^3f_{yy}f^2$
- For $t\epsilon t_{6,16}$: $f_yf_{yy}f_x^2+f_y^2f_{yy}f_xf^2+f_y^3f_{xy}f^2+f_y^2f_{yy}f_xf+f_y^3f_{yy}f^2$
- For $t\epsilon t_{6,17}$: $f_y^2f_{xxx}+3\,f_y^2f_{xxy}f+3\,f_y^2f_{yyx}f^2+f_y^2f_{yyy}f^3$
- For $t\epsilon t_{6,18}$: $f_y^2\,f_{xy}f_x+f_y^2f_{xy}f_yf+f_y^2\,f_{yy}f_xf+f_y^3f_{yy}f^2$

Table 3: Numerical solutions of Eq. 29 using 16-18 and the exact solution of Eq. 30

t	y (t) PRKF 1	y (t) PRKF 2	y (t) PPRKF	y (t) exact solution
0.10	1.111200676166252	1.1156189945744146	1.1115122547098906	1.1114633762807828
0.20	1.252305861366426	1.2642992832206825	1.2531607259800628	1.2530169332674046
0.30	1.4381634992684074	1.463715868778002	1.4400011995155453	1.4396707765716792
0.40	1.6931091629940396	1.7444486876984913	1.6968269156284592	1.696110908922183
0.50	2.0609571381417444	2.1659480419818555	2.0685859656164265	2.066999712085663
0.60	2.63096564782441	2.864407885295942	2.647849838494417	2.6439996941524915
0.70	3.620264308146107	4.245626037377813	3.6641998062742247	3.6529027867046437
0.80	5.736376893871067	---	5.897953017675903	5.848616804732598
0.90	13.343869372107086	---	14.973303510123667	14.304864332834065

- For $t\epsilon t_{6,19}$: $f_y^3f_{xx}+2\,f_y^3f_{yx}f+f_y^3f_{yy}f^2$
- For $t\epsilon t_{6,20}$: $2f_y^4f_x+f_y^5\,f$

Thus by making use of the derivative terms in this section and error coefficients given in Table 2 and also by making use of Eq. 14, we can find the LTE of all the Runge-Kutta fifth order algorithms given earlier which are as follows:

- The estimated LTE obtained for Parallel Runge-Kutta-Fifth order method 2 (PRKF 1) is given as:

$$
\begin{aligned}
&\frac{h^6}{680400000}\big(9231705f^2f_{xy}f_{xyy}+2938572ff_{xxxy}f_y+8138340f^2f_{xxyy}f_y-472500f^3f_{xy}f_y\\
&+13934025ff_{xy}^2f_y+7944804f^3f_{xyxy}f_y+360045f^2f_{xxx}f_y^2+16237935f^2f_{xyy}f_y^2\\
&+4110750ff_y^3+8221500f^2f_y^3+4110750f^3f_y^3+4701375ff_{xy}f_y^3+2197125f^2f_{xy}f+\\
&945000ff_y^5+189(3215f_{xxx}f_{xy}+512f_{xxx}f_y+607632ff_{xxx}f_{yy}+5527305f^3f_{xyy}f_{yy}+\\
&28263745f^2f_{xy}f_yf_{yy}+6449625f^2f_{xy}f_y^2f_{yy}+14789250f^2f_y^2f_{yy}+15485585f^3f_yf_{yy}^2\\
&+945ff_{xxy}(5849f_{xy}+6193f_y^2+3889ff_{yy})+4312035f^3f_{xy}f_{yyy}+10745595f^3f_y^2f_{yyy}\\
&+2459835f^4f_{yy}f_{yyy}+4725f_x^2(1188f_{xyy}-5f_y(20ff_{xy}-439f_{yy})+1188ff_{yyy})\\
&+20f_{xx}(92610ff_{xxy}+185220ff_{xyy}+f_y(945(92f_{xy}-5(2+5f^2)f_y^2)+451798ff_{yy})\\
&+92610ff_{yyy})+f_x(2551500f_{xxxy}+756f(10125+128f)f_{xxyy}+756f(3375\\
&+2f(3375+64f))f_{xyyy}+23625(273f_{xy}^2+202f_{xxy}f_y)+5293760f_{xx}f_{yy}\\
&+5(4154220ff_{xyy}f_y+378000f_y^4+2348677ff_{yy}(2f_{xy}+f_{yy})\\
&+14175f_y^2(58+174f+65f_{xy}+f(96+31f)f_{yy})+3199770f^2f_yf_{yyy}))\\
&+2648268f^4f_yf_{yyyy}+16348500f_y^2f_{yy}ff_x
\end{aligned}
\tag{19}
$$

- The estimated LTE obtained for Parallel Runge-Kutta-Fifth order method 2 (PRKF 2) is given as follows:

$$
\begin{aligned}
&\frac{h^6}{324000}\big(6f_{xxxxx}+24f_{xxxyy}+60f^2f_{xxxyy}+30f_{xx}-3210f_{xxy}f_{xy}-270f_xf_{xy}^2+90f_x^2f_{xyy}\\
&+60ff_{xx}f_{xyy}-3150f^2f_{xy}f_{xyy}+60ff_xf_{xyyy}+120f^2f_xf_{xyyy}-45f^3f_xf_{xyyy}+30f^4f_{xyyy}\\
&+180f_xf_{xxy}f_y-45f^2f_{xxyy}f_y-900f^3f_{xy}f_y-900ff_x^2f_{xy}f_y-270f_{xy}^2f_y540ff_xf_{xyy}f_y+\\
&45f^3f_{xyyy}f_y+1215f_xf_y^2+3645ff_xf_y^2+90f_{xxx}f_y^2+450ff_{xxy}f_y^2-1080f_xf_{xyy}f_y^2+\\
&720f^2f_{xyy}f_y^2+1215ff_y^3+3430f^2f_y^3+1215f^3f_y^3-900f^2f_{xx}f_y^3-180ff_{xy}f_y^3-\\
&810ff_{xy}f_y^3-810f^2f_{xy}f_y^3+900f_xf_y^4+450ff_y^5+6f_{xxxy}(f+10f_x-20ff_y)+\\
&5(-9(-4+f)ff_xf_{xxyy}+12f^3f_{xxyyy}-218f_{xxx}f_{xy}-f_{xxxx}f_y)+30f_xf_{xx}f_{yy}-\\
&1090ff_{xxx}f_{yy}-3240f^2f_{xxy}f_{yy}-480ff_xf_{xy}f_{yy}-3210f^3f_{xyy}f_{yy}-630f_x^2f_yf_{yy}\\
&+30f_{xx}f_yf_{yy}-210f^2f_{xy}f_yf_{yy}^2+1890ff_xf_y^2f_{yy}-810f^2f_xf_y^2f_{yy}-270f^2f_{xy}f_y^2f_{yy}\\
&-1710f^2f_y^3f_{yy}-240f^2f_xf_{yy}^2-240f^3f_yf_{yy}^2+90ff_x^2f_{yyy}+30ff_{xx}f_{yyy}\\
&-1030f^3f_{xy}f_{yyy}+360f^2f_xf_yff_{yyy}+360f^3f_y^2f_{yyy}-1060f^4f_{yy}f_{yyy}\\
&+15f^4f_yf_{yyy}+6f^5f_{yyyy}+360f_y^2f_{yy}ff_x\big)
\end{aligned}
\tag{20}
$$

- The estimated LTE obtained for Proposed Parallel Runge-Kutta-Fifth order method (PPRKF) is given as follows:

$$
\begin{aligned}
&\frac{h^6}{720}(2f_xf_y^4+ff_y^5+\frac{1945639f_y^2((1+3f)f_x+f(1+f)^2f_y}{282880}+\frac{87}{7}f_y^2(f_x+ff_y)(f_{xy}+ff_{yy})\\
&+\frac{457}{364}f_y^3(f_{xx}+f(2f_{xy}+ff_{yy})-\frac{29652313129(f_x+ff_y)f_{yy}(f_{xx}+f(2f_{xy}+ff_{yy}))}{2339064000}\\
&+\frac{78291f_y(f_{xy}+ff_{yy})(f_{xx}+f(2f_{xy}+ff_{yy}))}{12376}+\frac{147593ff_y(f_{xx}ff_{yy})}{15470}\\
&+\frac{(f_{xy}+ff_{yy})(2f_{xy}+ff_{yy}}{15470}+\frac{4900851f_y(f_x^2f_{yy}+ff_y(ff_{xy}f_y+((1+f)f_x+ff_y)f_{yy}))}{1732640}\\
&-\frac{11999}{476}(f_x(f_{xy}+ff_{yy})^2+ff_y(f_{xy}^2+ff_{xy}(1+f_y)f_{yy}+f^2f_{yy}^2))\\
&+\frac{16143f_y(f_{xy}(f^3+f_x(f_{xy}+f))}{2912}+\frac{f_{yy}(f_xf_{yy}+ff_y(f_{xx}+f_{yy})))}{2912}\\
&+\frac{7116499(f_x+ff_y)^2(f_{xyy}+f_{yyy})}{8663200}+\frac{188771f_y^2(f_{xxx}+3f(f_{xxy}+ff_{xyy})+f^3f_{yyy})}{116025}\\
&+\frac{63193833(f_{xy}+ff_{yy})(f_{xxx}+3f(f_{xxy}+ff_{xxy})+f^3f_{yyy})}{24752000}\\
&+\frac{123477f_y(f_x+ff_y)(f_{xxy}+f(2f_{xyy}+ff_{yyy})}{6188}+\\
&+\frac{125221(f_x(f_{xxy}+3ff_{xxyy})+f(1+2f)f_{xyyy}+ff_y(f_{xxxy}+3f(f_{xxxy}+ff_{xyyy})+f^3f_{yyy})}{30940}\\
&+\frac{3919457(f_{xx}(f_{xxy}+f(2f_{xyy}+f_{yyy}))+f(2f_{xy}+ff_{yy})(f_{xxy}+f(2_{xyy}+ff_{yyy})))}{618800}\\
&+\frac{6855256693(f^2f_x(f_{xxyy}+ff_{xyyy})+f_y(f_{xxxx}+f(4f_{xxxy}+5ff_{xxyy}+3f^2f_{xyyy}+f^3f_{yyyy})))}{23761920000}\\
&+\frac{33(f_{xxxx}+4f_{xxxxy})+f(f_{xxxy}+5f(2f_{xxxyy}+2ff_{xxyyy}+f^2f_{xyyyy})+f^4f_{yyyyy}))}{1600}\\
&+\frac{1000511f_yf_{yy}(f_x^2+f_y(f^2f_y+2ff_x))}{216580})
\end{aligned}
\tag{21}
$$

LTE bounds: If we assume that the following bounds for f and its partial derivatives for $x\epsilon[x_0,\ x_n]$ and $y\epsilon[-\infty,\infty]$ according to Lotkin, 1951, then we have:

$$
\begin{aligned}
&|F(x,y)<Q|\\
&\left|\frac{\partial^{i+j}f(x,y)}{\partial x^i\,\partial y^i}\right|<\frac{P^{i+j}}{Q^{j-1}}\quad i+j\le p
\end{aligned}
\tag{22}
$$

where, P and Q are positive constants and p is the order of the method. Here p = 5.

Therefore, the derivatives in Eq. 19 using Eq. 22 are given by:

$$\left|f^2f_{xy}f_{xyy}\right|<Q^2\times\frac{P^2}{Q^0}\times\frac{P^3}{Q}=P^5Q$$

$$\left|ff_{xxxy}f_y\right|<Q\times\frac{P^4}{Q^0}\times\frac{P}{Q^0}=P^5Q$$

$$\left|f^2ff_{xxxy}f_y\right|<Q^2\times\frac{P^4}{Q^0}\times\frac{P}{Q^0}=P^5Q$$

and so on.

Therefore, we obtain from Eq. 19:

$$LTE_{PRKF-1}\le\frac{3831543131}{6429780000}\,\frac{h^6P^5Q}{00000}\tag{23}$$

Substituting Eq. 22 into 20, we get:

$$LTE_{PRKF-2} \leq \frac{1751 \quad h^6P^5Q}{18225000} \tag{24}$$

Similarly, Eq. 21 gives as:

$$LTE_{PPRKF} \leq \frac{22969445 \quad h^6P^5Q}{6736504320\ 00000000} \tag{25}$$

These LTE bounds are useful in selecting suitable h values according to the error at each step. We may select an error tolerance TOL = 0.000001. The step-size selection h can be made using PRKF 1 and it is given by:

$$\frac{383154131 \quad h^6P^5Q}{6429780000 \quad 00000} < TOL$$
$$\text{or} \quad h < \left(\frac{167811.761\ TOL}{P^5Q}\right)^{\frac{1}{6}} \tag{26}$$

Similarly, the step-size selection for PRKF 2 is given by:

$$\frac{1751\, h^6P^5Q}{18225000} < TOL \text{ or } h < \left(\frac{10425.471\ Tol}{P^5Q}\right)^{\frac{1}{6}} \tag{27}$$

and the step-size selection for PPRKF is given by:

$$\frac{2296445 \quad h^6P^5Q}{6736504320 \quad 00000000} < TOL$$
$$\text{or} \quad h < \left(\frac{171254.521\ TOL}{P^5Q}\right)^{\frac{1}{6}} \tag{28}$$

It is observed from Eq. 26-28 that the PPRKF allows us to select the large step-size as compared to other two existing methods (PRKF 1 and PRKF 2) when the error tolerance (TOL) is fixed.

NUMERICAL EXAMPLES AND DISCUSSION

Example 1: We consider the differential equation:

$$y' = y^2+t^2,\ y\ (0) = 1 \tag{29}$$

The exact solution of this equation is given as follows:

$$y(t) = -\frac{\left(\text{BesselJ}\left[\frac{3}{4},\frac{t^2}{2}\right]\text{Gamma}\left[\frac{1}{4}\right] + 2\text{BesselJ}\left[\frac{3}{4},\frac{t^2}{2}\right]\text{Gamma}\left[\frac{3}{4}\right]\right)}{\text{BesselJ}\left[\frac{1}{4},\frac{t^2}{2}\right]\text{Gamma}\left[\frac{1}{4}\right] - 2\text{BesselJ}\left[\frac{1}{4},\frac{t^2}{2}\right]\text{Gamma}\left[\frac{3}{4}\right]} \tag{30}$$

Using the methods of PRKF 1, PRKF 2 and PPRKF, the discrete and exact solutions of the numerical example-1 have been computed for different time intervals and depicted in Table 3. The values of y (t) is calculated for time t ranging from 0.10 and 0.90 using a standard step-size h = 0.02.

It is pertinent to point out here that the obtained discrete solution for the present example-1 using the PPRKF guarantees more accurate values compared to the methods of PRKF 1 and PRKF 2. For the higher value of t, the methods of PRKF 1 and PRKF 2 give rise larger values of y (t) and these numerical values are very much deviated from their corresponding exact solutions. The absolute error which is the difference between the numerical solutions given by Eq. 16, 17 and 19 and the exact solution given by Eq. 30 and estimated LTE using Eq. 19, 20 and 21 for the methods considered in the present study are computed and shown in Fig. 1. It is observed from Fig. 1 that both the estimated LTE and absolute error are less in present numerical technique PPRKF and these errors are very high in the methods of PRKF 1 and PRKF 2. The absolute error of the method PRKF 2 is not plotted for the t values 0.8-0.9 as it is considerably high. Similarly, the estimated errors of the method PRKF 1 is not taken into account for the t values from 0.5-0.9 as it shows very high values. The estimated and absolute errors of PPRKF shows values nearer to exact solution and is working well.

Example 2: We consider the differential equation:

$$y' = -2\,t\,y^2,\ y\ (0) = 1 \tag{31}$$

The exact solution of this equation is given as:

$$y(t) = \frac{1}{1+t^2} \tag{32}$$

Using the methods of PRKF 1, PRKF 2 and PPRKF, the discrete and exact solutions of the numerical example-2 have been computed for different time intervals and depicted in Table 4. The values of y (t) is calculated for time t ranging from 0.10 and 1.0 using a standard step-size h = 0.02.The absolute error and estimated LTE are showed in Fig. 2.

It is important to note it out here that the obtained discrete solution for the present example-2 using the PPRKF guarantees more accurate values compared to the methods of PRKF 1 and PRKF 2. The absolute error which is the difference between the numerical solutions given by Eq. 16, 17 and 19 and the exact solution given by Eq. 32 and the estimated LTE using Eq.19, 20 and 21 for the

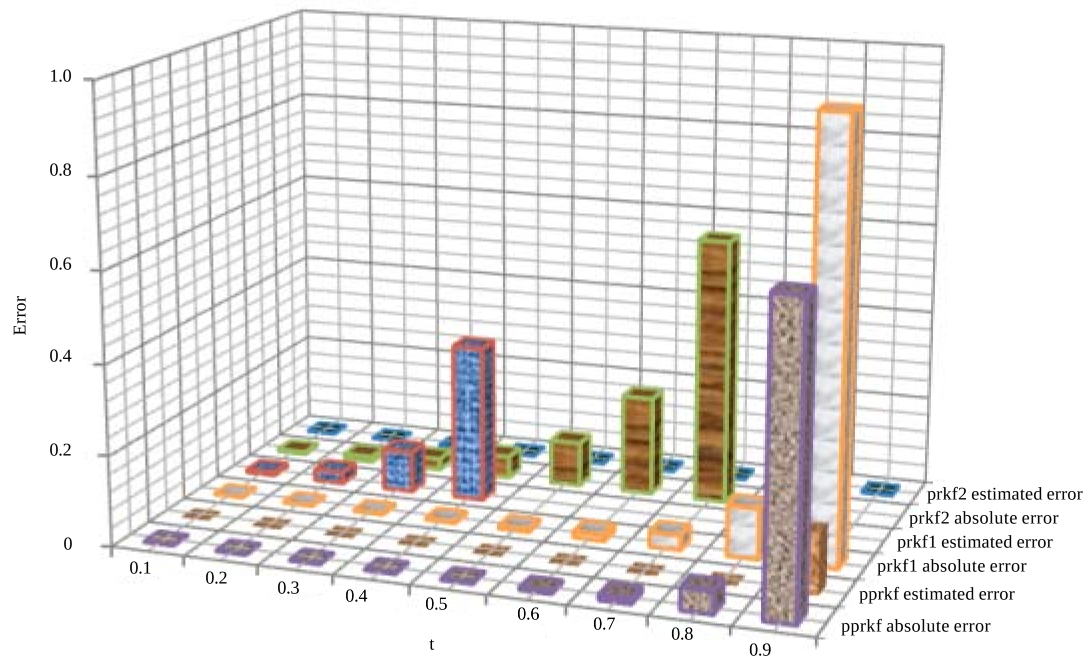


Fig. 1: Comparison of absolute and estimated errors of Parallel Runge-Kutta fifth order methods of example-1

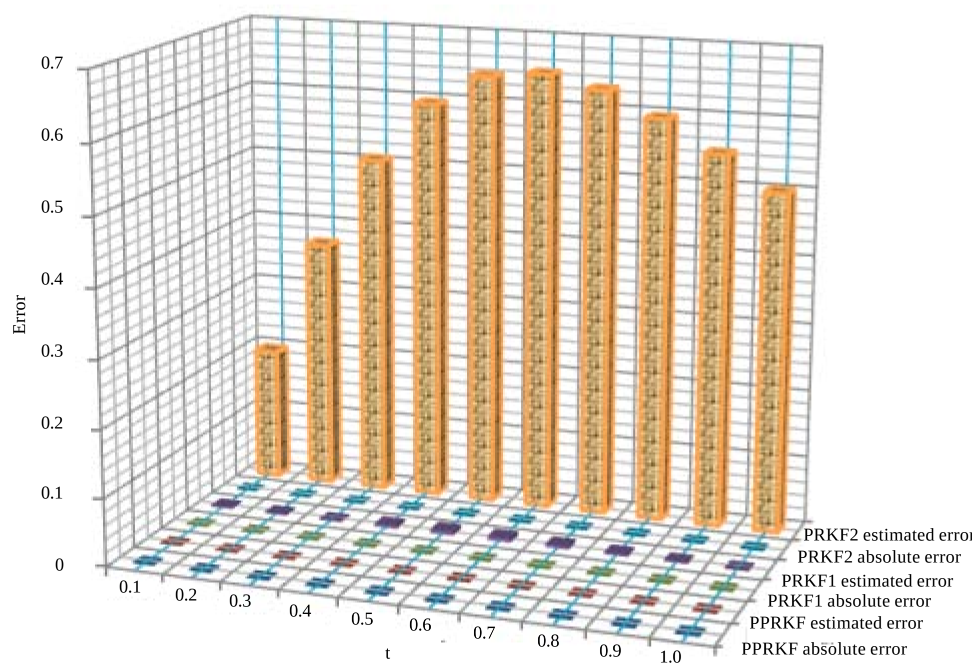


Fig. 2: Comparison of absolute and estimated errors of Parallel Runge-Kutta fifth order methods of example-2

Table 4: Numerical solutions of Eq. 31 using Eq. 16-18 and the exact solution of Eq. 32

t	y (t) PRKF 1	y(t) PRKF 2	y(t) PPRKF	Exact solution
0.10	0.990096	0.990096	0.990102	0.990099
0.20	0.961516	0.961516	0.96155	0.961538
0.30	0.917368	0.917368	0.917453	0.917431
0.40	0.861944	0.861944	0.862098	0.862069
0.50	0.799802	0.799802	0.800031	0.8
0.60	0.735023	0.735023	0.735322	0.735294
0.70	0.670805	0.670805	0.671161	0.671141
0.80	0.60937	0.60937	0.609767	0.609756
0.90	0.552065	0.552065	0.552488	0.552486
1.0	0.499559	0.499559	0.499993	0.5

methods considered in the present study are computed and shown in Fig. 2. It is observed from Fig. 2 that both the estimated LTE and absolute error are less in present numerical technique PPRKF in compared with the methods of the methods of PRKF 1 and PRKF 2. In particular, the estimated error in PRKF 2 is very high when compared to the other methods observed by Fig. 2.

CONCLUSION

In this study, for the first time, we obtained the LTE of the parallel fifth order explicit Runge-Kutta methods using the order trees and elementary differentials and also by not using the Taylor expansion comparisons which is useful in the examination of these methods. We have taken two examples in section 5 and applied the parallel methods to solve the examples. We also compared the numerical solutions with the exact solutions and observed that the method PPRKF is well suited in those examples. The LTE expressions and the LTE bounds of the parallel methods are useful in deciding the step size in various applications to have error control. Further, higher order parallel methods and their corresponding LTE can be developed to use in engineering applications.

REFERENCES

Bazuaye, F.E., 2006. An improved linearized semi-implicit Runge-Kutta methods. Trends Applied Sci. Res., 1: 444-455.

Bazuaye, F.E. and J. Ataha, 2006. Approach to the construction of variable stepsize for a class of general linear methods. Trends Applied Sci. Res., 1: 477-486.

Burrage, K., 1995. Parallel and Sequential Methods for Ordinary Differential Equations. Numerical Mathematics and Scientific Computation Series. Oxford University Press, New York, ISBN: 0-19-853432-9.

Butcher, J.C., 1964. On Runge-Kutta processes of high order. J. Aust. Math. Soc., 4: 179-194.

Butcher, J.C., 1987. The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods. John Wiley and Sons, New York, USA., ISBN-13: 9780471910466, Pages: 512.

Butcher, J.C., 2010a. Trees and numerical methods for ordinary differential equations. Numer. Algorithms, 53: 153-170.

Butcher, J.C., 2010b. Trees, B-series and exponential integrators. IMA J. Numer. Anal., 30: 131-140.

Efurumibe, E.L., A.D. Asiegbu and M.U. Onuu, 2012. Mathematical modelling of electron transport through the anode (TiO_2) of a standard dye-sensitized solar cell. Asian J. Appl. Sci., 5 : 33-42.

Evans, D.J. and A.R. Yaakub, 1996. A new fifth order weighted Runge Kutta formula. Int. J. Comput. Math., 59: 227-243.

Evans, D.J. and B.B. Sanugi, 1989. A Parallel Runge-Kutta integration method. Parallel Comput., 11: 245-251.

Ganesh, S. and S. Krishnambal, 2006. Magnetohydrodynamic flow of viscous fluid between two parallel porous plates. J. Applied Sci., 6: 2420-2425.

Ganesh, S. and S. Krishnambal, 2007. Unsteady magnetohydrodynamic stokes flow of viscous fluid between two parallel porous plates. J. Applied Sci., 7: 374-379.

Gear, C.W., 1986. Potential for parallelism in ordinary differential equations. Technical Report No. UIUCDCS-R-86-1246, Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL., USA., pp: 19.

Hairer, E. and G. Wanner, 1974. On the Butcher group and general multi-value methods. Computing, 13: 1-15.

Hairer, E., 1981. Order conditions for numerical methods for partitioned ordinary differential equations. Numer. Math., 36: 431-445.

Hairer, E., S.P. Norsett and G. Wanner, 1993. Solving Ordinary Differential Equations I: Nonstiff Problems. 2nd Edn., Springer-Verlag, Berlin, Heidelberg, ISBN: 978-3-540-56670-0.

Jackson, K.R., 1991. A survey of parallel numerical methods for initial value problems for ordinary differential equations. IEEE Trans. Magn., 27: 3792-3797.

Jackson, K.R. and S.P. Norsett, 1995. The potential for parallelism in Runge-Kutta methods. Part 1: RK formulas in standard form. Siam J. Numer. Anal., 32: 49-82.

Katti, C.P. and D.K. Srivastava, 2003. On a parallel mesh-chopping algorithm for a class of initial value problems using fourth-order explicit Runge-Kutta method. Applied Math. Comput., 143: 565-570.

Khiyal, M.S.H. and K. Rashid, 2005. Implementation of s-stage implicit Runge-Kutta method of order 2s for second order initial value problems. J. Applied Sci., 5: 411-427.

Krishnambal, S. and P. Anuradha, 2006. Effect of radiation on the flow of a visco-elastic fluid and heat transfer in a porous medium over a stretching sheet. J. Applied Sci., 6: 2901-2906.

Lotkin, M., 1951. On the accuracy of Runge-Kutta's method. Math. Tables Other Aids Comput., 5: 128-132.

Mazaheri, A., R. Sotudeh-Gharebagh and Z. Emam-Djomeh, 2005. Simulation of the potato osmo-dehydration process using equation oriented approach. J. Biological Sci., 5: 134-140.

Nik Long, N.M.A., M. Suali, A. Ishak, N. Bachok and N.M. Arifin, 2011. Unsteady stagnation point flow and heat transfer over a stretching/shrinking sheet. J. Applied Sci., 11: 3520-3524.

Ponalagusamy, R. and K. Ponnammal, 2008. Investigations on robot arm model using a new parallel RK-fifth order algorithm. Int. J. Comput. Math. Sci. Appl., 2: 155-164.

Ponalagusamy, R. and S. Senthilkumar, 2008. A comparison of RK-fourth orders of variety of means on multilayer raster CNN simulation. Trends Applied Sci. Res., 3: 242-252.

Ponalagusamy, R., 2008. A novel and efficient computational algorithm of STWS for generalized linear non-singular/singular time varying systems. J. Software Eng., 2: 1-9.