Journal of the Nigerian Association of Mathematical Physics
Volume28, No. 1, (*November, 2014*), pp 141 – 150

A Parallel 5-Stage Runge-Kutta-Nystrom Methods for Solving Special Second Order Initial Value Problems

[1]Imoni, S.O., [2]Ikhile, M.N.O.

[1]Department of Mathematics, Federal University Lokoja, Kogi State, Nigeria.
[2]Department of Mathematics, University of Benin, Benin City, Nigeria.

Abstract

This paper investigates the construction of diagonally implicit Runge-Kutta-Nystrom (RKN) methods for the special second-order ordinary differential equations (ODEs) possessing oscillatory solutions for use on parallel computers. The parameters in the matrix coefficients are obtained as to ensure that the resultant scheme has an appropriate region of stability, thus suitable for oscillatory problems. Numerical comparison with existing method of solving this type of ODEs shows its advantage.

Keywords: Numerical analysis, parallel methods, second order IVPs, RKN methods
AMS classification: 65L05, 65L20

1.0 Introduction

A subject of great interest in numerous scientific areas is the integration of second-order initial -value problem.

$$y'' = f(x, y),\ y(x_0) = y_0\ \ y'(x_0) = y_0' \tag{1.1}$$

where $f : \square \times \square^n \to \square^n$ in which f does not depend on y' explicitly. The solutions of such systems are known to be oscillatory in nature and are frequently encountered in many fields of physics, mechanics and in other engineering applications.

A possible approach is to convert (1.1) into a first order system of ODEs and apply Runge-Kutta (RK) method. However, since (1.1) does not have y' term, great efficiency can be obtained by constructing special class of method. One popular class of method is the RKN method which takes the form [1, 2, 3, 4]

$$y_{n+1} = y_n + hy_n' + h^2 \sum_{j=1}^{s} b_j f_j \tag{1.2a}$$

$$y_{n+1}' = y_n' + h \sum_{j=1}^{s} b_j' f_j$$

where

$$f_j = f\left(x_n + c_j h, y_n + c_j h y_n' + h^2 \sum_{k=1}^{s} a_{jk} f_k \right) \tag{1.2b}$$

The coefficients c_j,a_{jk},b_j, b_j' determines the method and they satisfy the following equation

$$\tfrac{1}{2} c_j^2 = \sum_{k=1}^{s} a_{jk} \tag{1.3}$$

The method can be compactly represented by means of the Butcher tableau

Corresponding author:***Imoni, S.O.***E-mail: sunday.imoni@fulokoja.edu.ng, Tel.: +2348060887418

$$\begin{array}{c|c} c & A \\ \hline & b \\ b' & \end{array}$$

RKN methods can be divided into two broad classes Explicit $\left(a_{jk}=0,\ k\geq j\right)$ and implicit ($a_{jk}\neq 0, k\geq j$). The later contains the class of diagonally implicit Runge-Kutta-Nystrom (DIRKN) methods for which $\left(a_{jk}=0,\ k>j\right)$ and the a_{jj} are equal.

In the literature, various DIRKN methods have been developed for the integration of the systems of ODEs (1.1) on one-processor computers. We mention for example, the two-stage and three-stage DIRKN methods of orders three and four of Sharp *et. al* [3], the two-stage DIRKN methods of order four of Sommeijer [5], DIRKN methods for oscillatory problems by Van der Houwen and Sommeijer [6], the RKN methods of orders three for solving fuzzy differential equations of Kanagarajam and Sambath [7].

Parallel initial value problems (IVPs) solvers arise from the need to solve many substantial problems faster than is currently possible. The computational time on a conventional sequential machine is so large that it affects the productivity of scientist and engineers working on the design of complex systems.

In this paper, parallel diagonally implicit Runge-Kutta-Nystrom (PDIRKN) method is developed for the integration of special second order IVP (1.1). The choice of RKN has been motivated by the fact that, it results more efficient and require less storage than RK method applied to the first order system equivalent to (1.1).

According to Burrage [8], Amodio and Brugnano [9], the development of parallelism in IVP solvers can be classified into three main category:

- ***Parallelism across the method***: This means the possibility of distributing the computational effort of each integration steps among the processors.

- ***Parallelism across the system:*** This technique is via the decomposition of a problem into sub-problems which can then be solved in parallel with the processors communicating as appropriate

- ***Parallelism across the steps***: In which general integration steps are performed concurrently with a given numerical method.

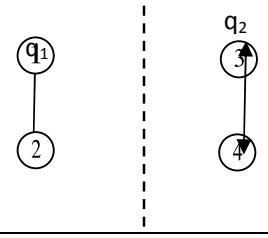
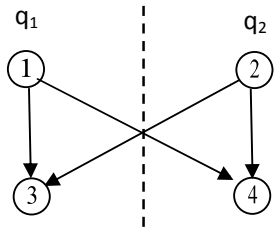
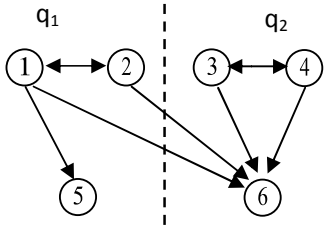
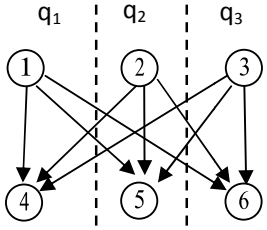
The construction of PDIRKN method for solving the IVPs associated to the special second order IVPs (1.1) in the first category, which is parallelism across the method is investigated is this paper.

2.0 The Concept of Directed Graphs (Digraphs)

The concept of directed graphs or digraphs was first introduced in [10]. From the model of the digraphs, the methods that can be parallelized are easily identified.

A digraph is a graph in which each edge has an orientation or direction assigned to it. It can be pictured like a graph with direction of an arc indicated by an arrow. The digraph models the sparsity pattern of matrix A. Table 1.1 gives four examples of sparsity pattern of matrix A and their digraphs [11,12]. The RKN array $\left\{a_{jk}\right\}_{j,k=1}^{s}$ represents the coefficients of the matrix method A where x and 0 are used to denote non-zero and zero coefficients of A respectively. In plotting the digraph, each arrow in the corresponding "production graph", pointing from vertex "k" to vertex "j" stands for a non-zero a_{jk}. For example, in Table 1.1, for method I, stages one and two can be computed together independent of stages three and four in parallel. Thus, the four stages can be evaluated at the same time as two stages in serial architecture. Here q_1 and q_2 denote the first and the second processor respectively

Table 1.1 RKN Matrices and Digraphs

Method	Runge –Kutta –Nystrom Matrix	Digraph
I.	$\begin{bmatrix} x & x & 0 & 0 \\ x & x & 0 & 0 \\ 0 & 0 & x & x \\ 0 & 0 & x & x \end{bmatrix}$	
II.	$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & 0 & 0 \\ x & x & x & 0 \\ x & x & 0 & x \end{bmatrix}$	
III.	$\begin{bmatrix} x & x & 0 & 0 & 0 & 0 \\ x & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 \\ 0 & 0 & x & x & 0 & 0 \\ x & x & x & x & x & 0 \\ x & x & x & x & 0 & x \end{bmatrix}$	
IV.	$\begin{bmatrix} x & 0 & 0 & 0 & 0 & 0 \\ 0 & x & 0 & 0 & 0 & 0 \\ 0 & 0 & x & 0 & 0 & 0 \\ x & x & x & x & 0 & 0 \\ x & x & x & 0 & x & 0 \\ x & x & x & 0 & 0 & x \end{bmatrix}$	

In method II of Table 1.1, it is possible to evaluate the first two stages which are independent of each other in parallel with two processors. The remaining stages that are mutually independent, though they depend on the preceding stages can again be calculated in parallel. The method is a four-stage two-parallel and two-processor method. Based on [11, 12], we easily extend parallelism to RKN method.

The stability of (1.2) is investigated using the test equation

$$y'' = -\omega^2 y \ , \quad \omega \in \square \tag{2.1}$$

Application of (1.2) to (2.1) yields the following relation.

$$\begin{pmatrix} y_{n+1} \\ hy'_{n+1} \end{pmatrix} = R(z) \begin{pmatrix} y_n \\ hy'_n \end{pmatrix}, \qquad z = -h^2\omega^2$$

where

$$R(z) = \begin{bmatrix} 1 + zb^T (I - zA)^{-1} c & 1 + zb^T (1 - zA)^{-1} c \\ zb'^T (I - zA)^{-1} & 1 + zb'^T (I - zA)^{-1} c \end{bmatrix} \tag{2.2}$$

$A = \{a_{ik}\}_{j,k=1}^{s}$ $e = [1, \cdots, 1]^T$, $b = [b_1, \cdots, b_s]^T$ $b' = [b'_1, \cdots, b'_1]^T$, $c = [c_1, \cdots, c_s]^T$:

The matrix $R(z)$ which determines the stability of the method is called the amplification matrix. Following [6], we introduced the functions

$s(z)$ and $p(z)$ with $s(z) = \text{trace}\,(R(z))$ and $p(z) = \det(R(z))$

The characteristic equation corresponding to (2.2) is of the form

$$\zeta^2 - s(z)\zeta + p(z) = 0 \qquad (2.3)$$

An essential property for computing periodic motion of (1.1) is the situation where the eigenvalues $\zeta_{1,2}$ are on the unit circle.

Definition 2.1 (Periodicity Interval [6]):An RKN methods has periodicity interval $I_z = (z_0,\ 0)$ if the roots of its characteristic equation $\zeta_{1,2}$ are on the unit circle and $\zeta_1 \neq \zeta_2$, $\forall z \in (z_0,\ 0)$, is called the stability boundary.

3.0 Construction of the PDIRKN Method

This section describes the construction of PDIRKN methods for solving the special second order IVPs (1.1). The method introduced are constructed by imposing that the matrix A be block lower triangular with diagonal blocks which themselves are diagonal. In this way, the system (1.2b) is decoupled into m independent systems of order p that can be executed in parallel using q -processors so realizing parallelism in the stage computation.

This paper considers five-stage, two-parallel, and two-processor fifth order method. The method has the sparsity pattern shown in Figure 3.1,

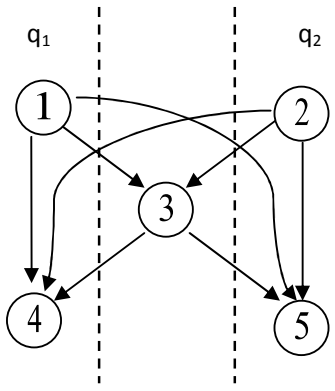
RKN Matrix	Digraph
$\begin{pmatrix} \times & 0 & 0 & 0 & 0 \\ 0 & \times & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & 0 \\ \times & \times & \times & 0 & \times \end{pmatrix}$	

Figure 3.1: 5-Stage PDIRKN Matrix and Digraph

where the symbol $\times$ denotes a non-zero element. Then, stages 1 and 2, and stages 4 and 5 can each be computed concurrently. The following order conditions are needed to be satisfied after making use of two basic simplifying assumptions [2, 13, 14]

$$Order\ one: \sum_{i=1}^{5} b_i^{'} = b_1^{'} + b_2^{'} + b_3^{'} + b_4^{'} + b_5^{'} = 1 \qquad (3.1)$$

$$Order\ two: \sum_{i=1}^{5} b_i^{'} c_i = b_1^{'} c_1 + b_2^{'} c_2 + b_3^{'} c_3 + b_4^{'} c_4 + b_5^{'} c_5 = \tfrac{1}{2} \qquad (3.2)$$

$$Order\ three: \sum_{i=1}^{5} b_i^{'} c_i^2 = b_1^{'} c_1^2 + b_2^{'} c_2^2 + b_3^{'} c_3^2 + b_4^{'} c_4^2 + b_5^{'} c_5^2 = \tfrac{1}{3} \qquad (3.3)$$

$$Order\ four: \sum_{i=1}^{5} b_i^{'} c_i^3 = b_1^{'} c_1^3 + b_2^{'} c_2^3 + b_3^{'} c_3^3 + b_4^{'} c_4^3 + b_5^{'} c_5^3 = \tfrac{1}{4} \qquad (3.4)$$

$$\begin{aligned} \sum_{i,j=1}^{5} b_i^{'} a_{ij} c_j &= b_1^{'}\lambda c_1 + b_2^{'} a_{21} c_1 + b_2^{'}\lambda c_2 + b_3^{'} a_{31} c_1 + b_3^{'} a_{32} c_2 + b_3^{'}\lambda c_3 + \\ & b_4^{'} a_{41} c_1 + b_4^{'} a_{42} c_2 + b_4^{'} a_{43} c_3 + b_4^{'}\lambda c_4 + b_5^{'} a_{51} c_1 + \\ & b_5^{'} a_{52} c_2 + b_5^{'} a_{53} c_3 + b_5^{'} a_{54} c_4 + b_5^{'}\lambda c_5 = \tfrac{1}{24} \end{aligned} \qquad (3.5)$$

$$Order\ \ five: \quad \sum_{i=1}^{5} \hat{b}_i' c_i^4 = b_1' c_1^4 + b_2' c_2^4 + b_3' c_3^4 + b_4' c_4^4 + b_5' c_5^4 = \tfrac{1}{5} \qquad (3.6)$$

$$\sum_{i,j=1}^{s} b_i' c_i a_{ij} c_j = b_1'\lambda c_1^2 + b_2' a_{21} c_1 c_2 + b_2' a_{22} c_2^2 + b_3' a_{31} c_1 c_3 + b_3' a_{32} c_2 c_3 +$$
$$b_3'\lambda c_3^2 + b_4' a_{41} c_1 c_4 + b_4' a_{42} c_2 c_4 + b_4' a_{43} c_3 c_4 + b_4'\lambda c_4^2 + \qquad (3.7)$$
$$b_5' a_{51} c_1 c_5 + b_5' a_{52} c_2 c_5 + b_5' a_{53} c_3 c_5 + b_5' a_{54} c_4 c_5 + b_5'\lambda c_5^2 = \tfrac{1}{30}$$

$$\sum_{i,j=1}^{5} b_i' a_{ij} c_j^2 = b_1'\lambda c_1^2 + b_2' a_{21} c_1^2 + b_2'\lambda c_2^2 + b_3' a_{31} c_1^2 + b_3' a_{32} c_2^2 + b_3'\lambda c_3^2 +$$
$$b_4' a_{41} c_1^2 + b_4' a_{42} c_2^2 + b_4' a_{43} c_3^2 + b_4'\lambda c_4^2 + b_5' a_{51} c_1^2 + b_5' a_{52} c_2^2 + \qquad (3.8)$$
$$b_5' a_{53} c_3^2 + b_5' a_{54} c_4^2 + b_5'\lambda c_5^2 = \tfrac{1}{60}$$
$$for\ y',\ and$$

simplifying assumptions:

$$\tfrac{1}{2} c_j^2 = \sum_{k=1}^{s} a_{jk} \qquad j = 1,...,s \qquad (3.9)$$

$$b_j = b_j' \left(1 - c_j\right),\ j = 1,...,s \qquad (3.10)$$

where the sums on *i* and *j* are from 1 to s respectively.

Consider the PDIRKN method by allowing the RKN matrix to be block lower triangular with diagonal blocks which are themselves diagonal. That is $a_{11} = a_{44}$ and $a_{22} = a_{55}$.

Considering (3.10), we have to satisfy thirteen equations in twenty-one unknowns, thus we have eight free parameters. The Butcher tableau of the 5-stage diagonally implicit PRKN method is

Table 3.1: The 5-Stage PDIRKN Method

c_1	a_{11}	
c_2	0	a_{22}
c_3	a_{31}	a_{32} a_{33}
c_4	a_{41}	a_{42} a_{43} a_{44}
c_5	a_{51} a_{52}	a_{53} 0 a_{55}
		b_1 b_2 b_3 b_4 b_5
b_1'	b_2'	b_3' b_4' b_5'

Let $c_1, c_2, c_3, c_4, c_5, a_{33}, a_{42}, a_{52}$ be our free parameters and the process in the derivation of the method is the following:

1. Solve equation (3.1) – (3.4) and (3.6) to obtain

$$b_1' = \frac{12 - 15c_4 - 15c_5 + 20c_4c_5 - 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) + 5c_2(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5)))}{60(c_1 - c_2)(c_1 - c_3)(c_1 - c_4)(c_1 - c_5)}$$

$$b_2' = \frac{-12 + 15c_4 + 15c_5 - 20c_4c_5 + 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) - 5c_1(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5)))}{60(c_1 - c_2)(c_2 - c_3)(c_2 - c_4)(c_2 - c_5)}$$

$$b_3' = \frac{12 - 15c_4 - 15c_5 + 20c_4c_5 - 5c_2(3 - 4c_5 + c_4(-4 + 6c_5)) + 5c_1(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_2(2 - 3c_5 + c_4(-3 + 6c_5)))}{60(c_1 - c_3)(c_2 - c_3)(c_3 - c_4)(c_3 - c_5)}$$

$$b_4' = \frac{-12+15c_3+15c_5-20c_3c_5+5c_2(3-4c_5+c_3(-4+6c_5))-5c_1(-3+c_3(4-6c_5)+4c_5+2c_3(2-3c_5+c_3(-3+6c_5)))}{60(c_1-c_4)(-c_2+c_4)(-c_3+c_4)(c_4-c_5)}$$

$$b_5' = \frac{12-15c_3-15c_4+20c_3c_4-5c_2(3-4c_4+c_3(-4+6c_4))+5c_1(-3+c_3(4-6c_4)+4c_4+2c_2(2-3c_4+c_3(-3+6c_4)))}{60(c_1-c_5)(c_2-c_5)(c_3-c_5)(c_4-c_5)}$$

2. Using the simplifying assumption (3.9), solve for $a_{11}, a_{22}, a_{31}, a_{41}$ and a_{51}

$$a_{11}=\frac{1}{2}c_1^2,\quad a_{22}=\frac{1}{2}c_2^2,\quad a_{31}=\frac{1}{2}c_3^2-a_{32}-a_{33},\quad a_{41}=\frac{1}{2}c_4^2-\frac{1}{2}c_1^2-a_{42}-a_{43}$$

$$a_{51}=\frac{1}{2}c_5^2-\frac{1}{2}c_2^2-a_{52}-a_{53}$$

3. Then solve for a_{32}, a_{43} and a_{53} using (3.5), (3.7) - (3.8) and obtained

$$a_{32}=\frac{P}{120b_3(c_1-c_2)(c_2-c_3)}$$

$$P = 2-120a_{52}b_5c_2^2-60b_2c_2^4-120a_{42}b_4c_2(c_2-c_3)-5c_3+120a_{52}b_5c_2c_3+60b_2c_2^3c_3+60b_4c_1^2(c_3-c_4)c_4$$
$$+60c_1^3(b_1c+b_4(-c_3+c_4))+60b_5c_2^2c_3c_5-60b_5c_2^2c_5^2+5c_1(-1+12b_2c_2^3+24a_{42}b_4(c_2-c_3)+24a_{52}b_5(c_2-c_3)-$$
$$12b_5c_2^2c_3+12b_3c_3^3+12b_4c_3c_4^2+12b_5c_2^2c_5+12b_5c_3c_5^2)$$

$$a_{43}=\frac{Q}{120b_4(c_1-c_3)(c_2-c_3)(c_4-c_5)},$$

$$Q=(60b_2c_2^5-120b_2c_2^4c_3-120a_{33}b_3c_3^3+60b_2c_2^3(c_3^2+c_1(c_3-c_5))-60c_2^2(2a_{42}b_4(c_3-c_4)+b_5(2a_{52}+$$
$$(c_1-c_3)(c_3-c_5))(c_3-c_5))+(-2+5c_1-60b_4c_1^3c_4+60b_4c_1^2c_4^2)c_5+c_2(-4+120a_{42}b_4c_1c_3+120a_{52}b_5c_1c_3+$$
$$120a_{42}b_4c_3^2+120a_{52}b_5c_3^2+60b_3c_1c_3^3-120a_{42}b_4c_1c_4-60b_4c_1^3c_4-120a_{42}b_4c_3c_4+60b_4c_1^2c_4^2+60b_4c_1c_4^3+$$
$$60b_1c_1^3(c_1-c_5)-120a_{33}b_3(c_1-c_3)(c_3-c_5)+5c_5-120a_{52}b_5c_1c_5+60b_4c_1^3c_5-120a_{52}b_5c_3c_5-60b_3c_1c_3^2c_5-$$
$$60b_4c_1^2c_4c_5-60b_4c_1c_4^2c_5)+5c_3^2(-1-24a_{42}b_4c_1-24a_{52}b_5c_1+12b_1c_1^3-12b_4c_1^3+12b_4c_1^2c_4+12b_4c_1c_4^2+$$
$$12b_5c_1c_5^2+24a_{33}b_3(c_1+c_3))-c_3(-6+60b_1c_1^4-120b_4c_1^3c_4+120b_4c_1^2c_4^2+5c_1(1-24a_{42}b_4c_4+12b_4c_4^3+$$
$$24a_{33}b_3c_5-24a_{52}b_5c_5+12b_5c_5^3)))$$

$$a_{53}=\frac{R}{120b_5(c_1-c_3)(c_2-c_3)(c_4-c_5)},$$

$$R=(-60b_2c_2^5+120b_2c_2^4+120a_{33}b_3c_3^3+c_4(2-5c_1+60b_4c_1^3c_4-60b_4c_1^2c_4^2)-60c_2^3(b_2(c_3^2+c_1(c_3-c_4))+$$
$$b_5(c_1-c_5)(c_4-c_5))-5c_3^2(-1-24a_{42}b_4c_1-24a_{52}b_5c_1+12b_1c_1^3-12b_4c_1^3+12b_4c_1^2c_4+12b_4c_1c_4^2+$$
$$24a_{33}b_3(c_1+c_4)+12b_5c_1c_5^2)+c_2(4-120a_{42}b_4c_1c_3-120a_{52}b_5c_1c_3-120a_{42}b_4c_3^2-120a_{52}b_5c_3^2-60b_3c_1c_3^3$$
$$-60b_1c_1^3(c_1-c_4)+120a_{33}b_3(c_1-c_3)(c_3-c_4)-5c_4+120a_{42}b_4c_1c_4+120a_{42}b_4c_3c_4+60b_3c_1c_3^2c_4+120a_{52}b_5c_1c_5$$
$$+120a_{52}b_5c_3c_5+60b_5c_1c_4c_5^2-60b_5c_1c_5^3+c_3(-6+60b_1c_1^4-120b_4c_1^3c_4+120b_4c_1^2c_4^2+5c_1(1+24a_{53}b_3c_4-2$$
$$4a_{42}b_4c_4+12b_4c_4^3-24a_{52}b_5c_5+12b_5c_5^3))+60c_2^2(2a_{42}b_4(c_3-c_4)+b_5(2a_{52}(c_3-c_5)+(c_1-c_5)$$
$$(c_3^2-2c_3c_5+c_4c_5))))$$

4. Finally, are the expressions for b_1, b_2, b_3, b_4 and b_5 considering the simplifying assumption (3.10)

$$b_1 = \frac{12-15c_4-15c_5+20c_4c_5-5c_3(3-4c_5+c_4(-4+6c_5))+5c_2(-3+c_4(4-6c_5)+4c_5+2c_3(2-3c_5+c_4(-3+6c_5)))(1-c_1)}{60(c_1-c_2)(c_1-c_3)(c_1-c_4)(c_1-c_5)}$$

$$b_2 = \frac{-12+15c_4+15c_5-20c_4c_5+5c_3(3-4c_5+c_4(-4+6c_5))-5c_1(-3+c_4(4-6c_5)+4c_5+2c_3(2-3c_5+c_4(-3+6c_5)))(1-c_2)}{60(c_1-c_2)(c_2-c_3)(c_2-c_4)(c_2-c_5)}$$

$$b_3' = \frac{12-15c_4-15c_5+20c_4c_5-5c_2(3-4c_5+c_4(-4+6c_5))+5c_1(-3+c_4(4-6c_5)+4c_5+2c_2(2-3c_5+c_4(-3+6c_5)))(1-c_3)}{60(c_1-c_3)(c_2-c_3)(c_3-c_4)(c_3-c_5)}$$

$$b_4' = \frac{-12+15c_3+15c_5-20c_3c_5+5c_2(3-4c_5+c_3(-4+6c_5))-5c_1(-3+c_3(4-6c_5)+4c_5+2c_3(2-3c_5+c_3(-3+6c_5)))(1-c_4)}{60(c_1-c_4)(-c_2+c_4)(-c_3+c_4)(c_4-c_5)}$$

$$b_5' = \frac{12-15c_3-15c_4+20c_3c_4-5c_2(3-4c_4+c_3(-4+6c_4))+5c_1(-3+c_3(4-6c_4)+4c_4+2c_2(2-3c_4+c_3(-3+6c_4)))(1-c_5)}{60(c_1-c_5)(c_2-c_5)(c_3-c_5)(c_4-c_5)}$$

The above steps combined with a symbolic manipulation package, help us to derive all the coefficients, explicitly in terms of the free parameters. Since this is achieved, the choice of the free parameters follows.

The free parameters can be used to get an optimal method minimizing the norms of the truncation error constant of the sixth order formula [14] given by

$$\left\|\tau^{(p+1)}\right\|_2 = \sqrt{\sum_{j=1}^{N_{p+1}}\left(\tau_j^{(p+1)}\right)^2}, \quad \left\|\tau'^{(p+1)}\right\|_2 = \sqrt{\sum_{j=1}^{N'_{p+1}}\left(\tau'^{(p+1)}_j\right)^2}$$

where $\tau_j^{(p+1)}$ and $\tau'^{(p+1)}_j$ are the error equations associated with the method. Thus we have the truncation error constant for this method as,

$$\left\|\tau^{(6)}\right\|_2 = \sqrt{\sum_{j=1}^{N_6}\left(\tau_j^{(6)}\right)^2}, \quad \left\|\tau'^{(6)}\right\|_2 = \sqrt{\sum_{j=1}^{N'_6}\left(\tau'^{(6)}_j\right)^2} \qquad (3.11)$$

where

$$\tau_1^{(6)} = \tau_2^{(6)} = \tau_3^{(6)} = \sum_i b_i c_i^4 - \frac{1}{30}, \quad \tau_4^{(6)} = \sum_{i,j} b_i c_i a_{ij} c_j - \frac{1}{180}, \quad \tau_5^{(6)} = \tau_6^{(6)} = \sum_{i,j} b_i a_{ij} c_j^2 - \frac{1}{360}$$

$$\tau_1'^{(6)} = \tau_2'^{(6)} = \tau_3'^{(6)} = \sum_i b_i c_i^4 - \frac{1}{6}, \quad \tau_4'^{(6)} = \tau_5'^{(6)} = \sum_{i,j} b_i c_i^2 a_{ij} c_j - \frac{1}{36}, \quad \tau_6'^{(6)} = \sum_{i,j} b_i c_i a_{ij} c_j^2 - \frac{1}{72},$$

$$\tau_7'^{(6)} = \tau_8'^{(6)} = \tau_9'^{(6)} = \sum_{i,j} b_i a_{ij} c_j^3 - \frac{1}{24}, \quad \tau_{10}'^{(6)} = \sum_{i,j,k} b_i a_{ij} a_{jk} c_k - \frac{1}{720}$$

Substitute the solutions obtained above into (3.11) and minimize using the Mathematical package subject to bounds, $0 \le c_i \le 1,\ i = 1,2,3$, 4,5 to choose the values for the free parameters. The resulting 5-stage order 5 PDIRKN method, is expressed in Butcher tableau form

Table 3.2: The Coefficients of the 5-Stage PDIRKN Method

$$
\begin{array}{cc|ccccc}
\frac{9}{10} & \frac{81}{200} & & & & & \\
 & & \frac{1}{10} & \frac{1}{200} & 0 & & \\
 & & \frac{22}{25} & \frac{74986}{187500} & -\frac{23408}{62500} & \frac{2}{15} & \\
 & & \frac{3}{5} & -\frac{4798}{7605} & \frac{17}{20} & \frac{38042}{60840} & \frac{81}{200} \\
 & & \frac{2}{5} & \frac{264615}{438360} & \frac{21}{40} & -\frac{266588}{438360} & 0 \quad \frac{1}{200} \\
\hline
 & & \frac{281}{1080} & \frac{13}{60} & \frac{1711}{7020} & \frac{625}{19656} & \frac{26}{105} \\
 & b & \frac{13}{600} & \frac{1711}{7800} & \frac{25}{6552} & \frac{52}{525} & \frac{281}{1800} \\
 & b, & & & & &
\end{array}
$$

We examine the stability interval of the method by using (2.2) and obtain the amplification matrix given by

$$R(z)=\begin{bmatrix} 1+z+\frac{63}{200}z^2+\frac{17}{125}z^3+\frac{207}{5000}z^4+\frac{3}{200}z^5 & 1+\frac{1471}{2500}z+\frac{91}{2500}z^2+\frac{41}{500}z^3+\frac{39}{1250}z^4+\frac{8}{625}z^5 \\ \frac{1}{2}z+\frac{541}{2500}z^2+\frac{79}{1250}z^3+\frac{77}{5000}z^4+\frac{3}{500}z^5 & 1+\frac{417}{2500}z+\frac{251}{5000}z^2+\frac{17}{1250}z^3+\frac{4}{625}z^4+\frac{1}{250}z^5 \end{bmatrix}$$

A boundary locus plot of R(z) gives the stability interval of approximately $(-2.4, 0)$. The stability region of the 5-stage PDIRKN method is shown in figure 3.2, where the stability region lies inside the boundary.

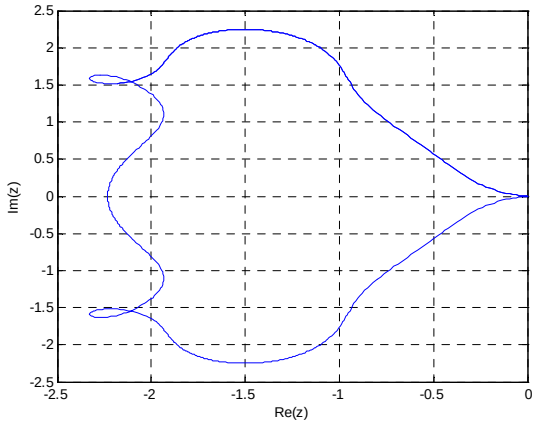


Figure 3.2: The Stability Region for the 5-Stage PDIRKN Method

4.0 Numerical Examples

We present the results obtained by applying the PDIRKN method in Table 3.2 and a reference method to some test problems. The method was implemented sequentially since parallel computers were not readily available. It was implemented on Pentium IV using MATLAB 7.5 software. Also, as there is no other fifth order PDIRKN method, we compare the "New method" derived in this paper with the SDIRKN method of Franco *et. al* [15]. This later method will be referred to as "FGR method".

The following are the two test problems considered and their computed results are shown in Table 4.1 and 4.2 respectively. The notations used are as follows:

New Method: The PDIRKN Method obtained in Table 3.2

FGR Method: SDIRKN method of Franco *et. al* [15]
h: Step size
FCN: the number of functions evaluations
Step: the number of steps

Emax: $\max\| y_n - y(x_n) \|$, that is, the absolute value of the computed solution minus the exact solution.

Problem 4.1
Consider the Duffing equation:

$$y''=-y-y^3+\frac{1}{500}\cos(1.01x),\ y(0)=0.20042678067,\ y'(0)=0$$

The exact solution of the nonlinear problem is given as

$$y(x)=0.200179477536\cos(1.01x)+0.00246946143\cos(3.03x)+0.304014.10^{-6}\cos(5.05x)+$$
$$0.374.10^{-9}\cos(7.07x)$$

Source: Qinghong and Yongzhong [16]

Problem 4.2
Consider the "test-like equation" with double frequencies, which is

$$y(x)+\omega^2 y(x)=12\cos(x),\quad y(0)=1,\ y'(0)=0,\quad x\in[0,10\pi]$$

The exact solution of the problem is

$$y(x)=\frac{1}{2}(\cos(5x)+\cos(x)),\ \text{set}\ \omega=5.$$

Source: Wang [17].

Table 4.1: Numerical Results for Problem 4.1

Method	h	FCN	Steps	Emax
New Method	0.1	18740	3748	$4.3579481799X10^{-01}$
FGR Method		14992	3748	$3.9859488172X10^{-01}$
New Method	0.05	37480	7496	$1.3799580621X10^{-02}$
FGR Method		29984	7496	$3.9840695984X10^{-01}$
New Method	0. 02	93700	18740	$6.8355067142X10^{-03}$
FGR Method		74960	18740	$3.9835333766X10^{-01}$
New Method	0.01	187405	37481	$5.2474479606X10^{-03}$
FGR Method		149924	37481	$3.9835773762X10^{-01}$
New Method	0.002	374810	74962	$4.5960355578X10^{-04}$
FGR Method		299848	74962	$3.9834589432X10^{-02}$

Table 4.2: Numerical Results for Problem 4.2

Method	h	FCN	Steps	Emax
New Method	0.04	3925	785	$8.4060967436X10^{-01}$
FGR Method		3140	785	$1.7037935366X10^{+00}$
New Method	0.02	7850	1570	$3.7531904192X10^{-01}$
FGR Method		6280	1570	$1.6903562761X10^{+00}$
New Method	0. 01	15705	3141	$9.3114750249X10^{-02}$
FGR Method		12564	3141	$1.4810039191X10^{+00}$
New Method	0.004	39265	7853	$7.9878530503X10^{-02}$
FGR Method		31412	7853	$1.9044356947X10^{-01}$
New Method	0.005	284155	56831	$8.5945987496X10^{-03}$
FGR Method		227324	56831	$1.7551432789X10^{-01}$

We decided to use the usual test based on computing of the maximum global error over the whole integration interval, because it gives a more significant measure of efficiency. As it can be observed in Table 4.1 and Table 4.2, the numerical results show that the new method performs compares in performance for the integration of the special second order IVPs (1.1) with oscillatory behaviour in terms of global error compared to FGR method. Hence, the new method is suitable for the integration of the IVPs (1.1).

5.0 Conclusion

This paper has examined the construction of PDIRKN methods for the integration of the special second order IVPs (1.1) on parallel computers.

The method introduced is constructed by imposing that the matrix A be block lower triangular with diagonal blocks which themselves are diagonal. In this way, the system (2.2) is decoupled into m independent systems that can be solved in parallel using q -processors, so realizing parallelism in the stage computation. The method considered is a 5-stage, two-parallel, two-processor fifth order. Also, the derived method has an appropriate region of stability, thus suitable for oscillatory solutions.

Numerical experiments were performed using a sequential computer compared with the method of Franco *et. al* [15]. From the magnitude of the maximum error, we may conclude that the new PDIRKN method is more promising compared to the existing method Franco *et. al* [15]. We mention also that PDIRKN method is cost-optimal.

Reference

[1] Hairer E.,Norsett S.P. and Wanner G. (1993), Solving ODEs I: Nonstiff Problems, Springer-Verlag, Berlin

[2] Dormand J.R., EL-Mikkawy M.E.A and Prince P.J. (1987), Families of RKN Formulae. IMA J. Numerical Analysis, 7, 235-250

[3] Sharp P.W and Fine J.M. and Burrage (1990), Two-stage and Three- stage DIRKN Methods of Orders Three and Four. IMA Journal of Numerical Analysis, 10, 489-504

[4] Imoni S.O. Otunta F.O. and Ramamohan T.R. (2006), Embedded Implicit RKN Method for Solving Second Order Differential Equations. Int. Journal of Comput. Math. Vol. 83, No. 11, 777-784

[5] Sommeijer B.P. (1987), A Note on DIRKN Method. J. Comput. Appl. Math. 19, 395-399

[6] Van de Houwen P.J. and Sommeijer B.P. (1989), DIRKN Methods for Oscillatory Problems. SIAM J. Numeri. Anal.Vol. 26, 414-429

[7] Kanagarajam K. and Sambath M. (2010), RKN Method of Order Three for Solving Fuzy Differential Equations. Computational Methods in Applied Mathematics, Vol. 10, No. 2, 195-203

[8] Burrage K. (1997), Parallel Methods for ODEs. Advances in Computational Mathematics, 7, 1-3

[9] Amodio P.and Brugnano L. (2008), Parallel Solution in Time ODEs: Some Achievements and Perspectives. Applied Numerical Mathematics, doi:10.1016/j.apnum.2008.03.024

[10] Iserles A. and Norsett S.P. (1990), On the theory of Parallel RK Methods IMA Journal of Numerical Analysis, 10, 463-488

[11] Norsett S.P and Simonsen H.H., (1987), Aspects of Parallel RK Method in Numerical Methods for ODEs, Lecture Notes in Mathematics. Proceedings of the I'Aquila Symposium

[12] Ismail F., Siri Z., Othman M. and Suleiman M. (2009), Parallel Execution of DIRK Method for Solving IVPs. European Journal ofScientific Research, Vol. 26, No.4, pp.480-489

[13] Fehlberg E. (1972), Classical Eight- and Lower-order RKN Formula with Stepsize control for Special Second-order Differential Equations. NASA, Tech. Report R-381

[14] Papakostas S.N. and Tsitouras, Ch.(2002), High phase-lag order Runge-Kutta- Nystrom pairs, SIAM J. Sci. Comput. 21, 747-763

[15] Franco J.M., Gomez I. and Randez L. (2001), Four-Stage, Symplectic and P-Stable SDIRKN Methods with Dispersion of High Order. Numerical Algorithm, 26, 347-363

[16] Qinghong Li. And Yongzhong S. (2006), Explicit One-step P-stable Methods for Second Order Periodic IVPs. A Journal of ChineseUniversities (English series), Vol. 15, No. 3, 237-247

[17] Wang Z. (2005), P-Stable Linear Symmetric Multistep Methods forPeriodic IVPs. Computer Physics Communications, 171, 162-174.