Acta Univ. Palacki. Olomuc., Fac. rer. nat.,
Mathematica **53**, 2 (2014) 53–69

Zero Dissipative DIRKN Pairs of Order 5(4) for Solving Special Second Order IVPs

S. O. IMONI [1*], M. N. O. IKHILE [2]

[1] *Department of Mathematics, Federal University Lokoja, Kogi State, Nigeria*
e-mail: sundayimoni13@gmail.com; imoni4u@yahoo.com

[2] *Department of Mathematics, University of Benin, Benin city, Nigeria*
e-mail: mnoikhilo@yahoo.com

(Received September 9, 2013)

Abstract

For initial value problem (IVPs) in ordinary second order differential equations of the special form $y'' = f(x, y)$ possessing oscillating solutions, diagonally implicit Runge–Kutta–Nystrom (DIRKN) formula-pairs of orders 5(4) in 5-stages are derived in this paper. The method is zero dissipative, thus it possesses a non-empty interval of periodicity. Some numerical results are presented to show the applicability of the new method compared with existing Runge–Kutta (RK) method applied to the problem reduced to first-order system.

Key words: Initial value problems, Runge–Kutta–Nystrom pairs, zero dissipative.

2010 Mathematics Subject Classification: 65L05

1 Introduction

We are concerned with IVPs of second order ordinary Differential Equations (ODEs) in which the first derivative does not appear explicitly,

$$y'' = f(x, y), \ y(x_0) = y_0, \ y'(x_0) = y'_0 \tag{1.1}$$

*Corresponding author

having a periodic or an oscillating solution, where $f\colon \mathbb{R}\times\mathbb{R}^n \to \mathbb{R}^n$. The solution of such systems are known to be highly oscillatory in nature and are frequently encountered in many fields of physics, seismology, mechanics and in other engineering applications and are usually considered as a difficult integration problem. One way of solving (1.1) is to transform it to a first order system and then use a Runge–Kutta (RK) pairs. The second is to solve directly using Runge–Kutta–Nystrom (RKN) pairs as can be seen in [3], [4], [6], [7], [9] and [14].

For an efficient implementation, we need a step size control mechanism to estimate the local error at relatively small extra cost. This is usually done by identifying an embedded pairs. The idea is to construct two RKN formulae containing besides the numerical approximation, y_{n+1}, y'_{n+1}, a second approximation $\hat{y}_{n+1}$, $\hat{y}'_{n+1}$ to $y\left(x_{n+1}\right)$ and $y'\left(x_{n+1}\right)$ respectively according to

$$\begin{aligned} y_{n+1} &= y_n + hy'_n + h^2 \textstyle\sum_{j=1}^{s} b_j f_j, \quad & y'_{n+1} &= y'_n + h \textstyle\sum_{j=1}^{s} b_j f_j \\ \hat{y}_{n+1} &= y_n + hy'_n + h^2 \textstyle\sum_{j=1}^{s} \widehat{b}_j f_j, \quad & \hat{y}'_{n+1} &= y'_n + h \textstyle\sum_{j=1}^{s} \hat{b}'_j f_j \end{aligned} \tag{1.2}$$

where

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

such that both use the same function values. The first two formulae are of order$p$, the second two are of order q $(q < p)$ and s denotes the number of function evaluation that is needed for each step, $h = x_{n+1} - x_n$ is the step size with the row sum assumption

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

The constant coefficients a_{jk}, b_j, b'_j, $\widehat{b}_j$, $\widehat{b}'_j$ determine a particular method explicitly. The differences $y_{n+1} - \hat{y}_{n+1}$ and $y'_{n+1} - \hat{y}'_{n+1}$respectively are estimates of the local truncation error used for step size control.

It is often advantageous, see [3], [5], [11] and [14] to apply a direct method for this type of differential equation rather than reducing it to its first-order form and solved using the RK pairs. This approach minimizes the computation cost of realizing the numerical solution of (1.1). Also, an embedded method provides cheap error estimation. We refer to (1.2) as an embedded RKN pairs, where we have assumed that order $p = q + 1$ approximations are used to obtain the numerical solution of the problem (1.1) and the embedded method of order q is used to obtain the local truncation error, which is used for the step size control algorithm. The method can be represented in Butcher Tableau as

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

$c = [c_1, c_2, \ldots, c_s]^T$, $A = [a_{ij}]$, $b = [b_1, b_2, \ldots, b_s]$ $b' = \left[b_1^{'}, b_2^{'}, \ldots, b_s^{'}\right]$, $\hat{b} = \left[\widehat{b}_1, \widehat{b}_2, \ldots, \widehat{b}_s\right]$, $\hat{b}' = \left[\widehat{b}_1', \widehat{b}_2', \ldots, \widehat{b}_s'\right]$.

Also, if it is initially recognized that the solution of (1.1) is of periodic nature, then it important to consider some special numerical properties such as phase-lag, dissipation, interval of periodicity, P-stability, exponential fitting, adaptive properties e.t.c., See [17] for more details. In this paper, we present the construction of embedded pairs of RKN methods of orders 5(4) that integrate exactly system (1.1) and satisfied the zero dissipative condition. This is actually a type of truncation error besides the truncation error due to the algebraic order, which is described as the distance from a standard cyclic solution [17]. This property assured us that the numerical method, neither damp nor amplify the oscillation of the problem (1.1).

The work is organized as follows: In section 2, some basic elements of RKN methods and embedded pairs are presented. Section 3 describes the stability of RKN method. In section 4 we describe the derivation of the new embedded DIRKN methods. In section 5 some numerical experiments show the properties of the developed algorithms. Finally, in Section 6 some conclusions are drawn.

2 Local error estimation

From (1.2), a local error estimation in the integration point $\mathrm{x}_{n+1} = \mathrm{x}_n + \mathrm{h}$ is determined by the expressions,

$$\Delta_{n+1} = y_{n+1} - \hat{y}_{n+1} \quad \text{and} \quad \Delta_{n+1}' = y_{n+1}' - \hat{y}_{n+1}^{'} \tag{2.1}$$

which is used for step size algorithm.

$$EST_{n+1}(h) = \max\left\{\|\Delta_{n+1}\|_\infty, \left\|\Delta_{n+1}'\right\|_\infty\right\} \tag{2.2}$$

represents the local error estimation to control the step-size h for RKN process by the well-known standard formula, see [3], [4]

$$h_{n+1} = 0.9 h_n \left(\frac{ToL}{EST_{n+1}(h_n)}\right)^{\frac{1}{(p+1)}} \tag{2.3}$$

where TOL is the maximum allowable error or tolerance, h_{n+1} is the current step size, h_n is the size of the previous step. If $EST_{n+1}(h) < ToL$, then the step is accepted. If $EST_{n+1}(h) \geq ToL$ then the step is rejected and we calculate h_{n+1} given by (2.3). When this difficulty repeatedly occurs, the factor 0.9 should be decreased.

3 Stability of RKN method

Linear stability analysis of numerical methods to solve problem (1.1) is based on the test equation.

$$y'' = -\omega^2 y, \quad \omega \in \mathbb{R} \tag{3.1}$$

Application of the RKN method (1.2) to problem (1.3) yields the following

$$\begin{aligned} y_{n+1} &= y_n + hy'_n - zb^T Y_n \\ hy'_{n+1} &= hy'_n - zb'^T Y_n \end{aligned}$$

where $z = -\omega^2 h^2$, $Y_n = (y_n e + hy'_n c)\, N^{-1}$, $N = I + zA$, $e = (1, \dots, 1)^T$, $c = (c_1, c_2, \dots, c_s)^T$.

Eliminating the auxiliary vector Y_n yields

$$\begin{bmatrix} y_{n+1} \\ hy'_{n+1} \end{bmatrix} = R(z) \begin{pmatrix} y_n \\ hy'_n \end{pmatrix},$$

$$\begin{aligned} R(z) &= \begin{bmatrix} 1 + zb^T (I - zA)^{-1} e & 1 + zb^T (I - zA)^{-1} c \\ zb'^T (I - zA)^{-1} e & 1 + zb'^T (I - zA)^{-1} c \end{bmatrix} \\ b &= [b_1, b_2, \dots, b_s]^T, \ b' = [b'_1, b'_2, \dots, b'_s]^T, \ A = \{a_{ij}\}_{i,j=1}^{s} \end{aligned} \tag{3.2}$$

The matrix $R(z)$ which determines the stability of the method is called the amplification matrix.

Following [19], we introduce the following functions

$$s(z) = \mathrm{trace}(R(z)), \quad p(z) = \det(R(z)) \tag{3.3}$$

The eigenvalues of the amplification matrix $R(z)$ are the roots of the characteristic equation

$$\zeta^2 - s(z)\zeta + p(z) = 0 \tag{3.4}$$

We give the following definitions which are originally from [19].

Definition 3.1 An RKN method has a zero dissipation if det(R(z)) = 1.

This is also essential for a method to possess a non-empty interval of periodicity.

Definition 3.2 An RKN method has periodicity interval $I_Z = (0, z_0)$ if the roots of its characteristic equation ζ_1, ζ_2 are on the unit circle and $\zeta_1 \neq \zeta_2, \forall z \in (0, z_0)$, z_0 is called the stability boundary. Consequently, the method is called P-stable if $I_Z = (0, 8)$.

An essential property for periodic motion is the situation where the eigenvalues ζ_1, ζ_2 are on the unit circle. Obviously, this periodicity condition requires zero dissipation.

Based on [18], let us write tr(R(z)) and det(R(z)) as introduced in (3.3) in the form

$$\mathrm{tr}(R(z)) = 2 + \sum_{i=1}^{s} (-1)^i \sigma_i z^{2i} \tag{3.5}$$

$$\det(R(z)) = 1 + \sum_{i=1}^{s} (-1)^i \pi_i z^{2i} \tag{3.6}$$

In this paper, we are interested in deriving an embedded pairs which are DIRKN method that is suitable for the integration of periodic IVPs (1.1).The embedded formula for the DIRKN method would have to be chosen so that its stability matrix has bounded eigenvalues, see [15].

Definition 3.3 (Order, [8]) A Nystrom method is of order p if, for sufficiently smooth problems (1.1),

$$y\left(x_n+h\right)-y_{n+1}=O\left(h^{p+1}\right) \quad \text{and} \quad y'\left(x_n+h\right)-y'_{n+1}=O\left(h^{p+1}\right)$$

where y_{n+1} and y'_{n+1} are the numerical solutions given by the method under the conditions that $y_n=y\left(x_n\right)$ and $y'_n=y'\left(x_n\right)$.

The order conditions for conventional RKN method has been thoroughly studied by [8]. These are also given by [5]. We list all the order conditions related to our method up to order five in Section 4.

4 Derivation of the Method

For the 5-stage, fourth- and fifth-order solution formulae with zero dissipative condition, the following order conditions are needed to be satisfied after we make use of three basic simplifying assumptions to be stated later.

$$\text{Order one:} \quad \sum_{i=1}^{s}\hat{b}'_i=1 \tag{4.1}$$

$$\text{Order two:} \quad \sum_{i=1}^{s}\hat{b}'_i c_i=\frac{1}{2} \tag{4.2}$$

$$\text{Order three:} \quad \sum_{i=1}^{s}\hat{b}'_i c_i^2=\frac{1}{3} \tag{4.3}$$

$$\text{Order four:} \quad \sum_{i=1}^{s}\hat{b}'_i c_i^3=\frac{1}{4} \tag{4.4}$$

$$\sum_{i=1}^{s}\hat{b}'_i a_{ij} c_j=\frac{1}{24} \tag{4.5}$$

$$\text{Order five:} \quad \sum_{i=1}^{s}\hat{b}'_i c_i^4=\frac{1}{5} \tag{4.6}$$

$$\sum_{i,j}^{s}\hat{b}'_i c_i a_{ij} c_j=\frac{1}{30} \tag{4.7}$$

$$\sum_{i,j}^{s}\hat{b}'_i a_{ij} c_j^2=\frac{1}{60} \tag{4.8}$$

for $\hat{y}'$ and

$$\text{Order two:} \quad \sum_{i=1}^{s} b' = \frac{1}{2} \tag{4.9}$$

$$\text{Order three:} \quad \sum_{i=1}^{s} b'_1 c_i = \frac{1}{6} \tag{4.10}$$

$$\text{Order four:} \quad \sum_{i=1}^{s} b' c_i^2 = \frac{1}{12} \tag{4.11}$$

$$\text{Order five:} \quad \sum_{i=1}^{s} b' c_i^3 = \frac{1}{20} \tag{4.12}$$

$$\sum_{i=1}^{s} b_i^{'} a_{ij} c_j = \frac{1}{120} \tag{4.13}$$

for y'.

Simplifying assumptions:

$$\frac{1}{2} c_j^2 = \sum_{k=1}^{s} a_{jk}, j = 1, \cdots 5 \tag{4.14}$$

$$\hat{b}_j = \hat{b}'_j (1 - c_j) \tag{4.15}$$

$$b_j = b'_j (1 - c_j) \tag{4.16}$$

See for example [8], [10], [12] and [16].

The tableau of the 5-stage, 5(4) RKN method is of the form.

There are eighteen non-linear equations to solve in twenty-six unknowns. Thus we have eight parameter families for 5(4) embedded DIRKN pairs.

4.1 Construction of the algorithm

We choose c_1, c_2, c_3, c_4, c_5, a_{43}, a_{52} and a_{53} to be the free parameters and the process in the derivation of the method is the following:

(1) Compute the det(R(z)) in terms of the RKN parameters. For an arbitrary 5-stage DIRKN method, the $\det(R(z))$ is given by

$$p(z) = \frac{1 - \pi_1 z + \pi_2 z^2 - \pi_3 z^3 + \pi_4 z^4 - \pi_5 z^5}{(1 - \lambda z)^5} \tag{4.17}$$

where λ is the diagonal element of matrix A,
$\pi_1 = b_1 + b_2 + b_3 + b_4 + b_5 - b'_1 + c_1 b'_1 - b'_2 + c_2 b'_2 - b'_3 + c_3 b'_3 - b'_4 + c_4 b'_4 - b'_5 + c_5 b'_5$
$\pi_2 = \lambda b_1 + \lambda b_2 + a_{21} b_2 + \lambda b_3 + a_{31} b_3 + a_{32} b_3 + \lambda b_4 a_{41} b_4 + a_{42} b_4 + a_{43} b_4 + \lambda b_5 + a_{51} b_5 + a_{52} b_5 + a_{53} b_5 + a_{54} b_5 - \lambda b'_1 - \lambda b'_2 - a_{21} b'_2 - \lambda b'_3 - a_{31} b'_3 - a_{32} b'_3 - \lambda b'_4 - a_{41} b'_4 - a_{42} b'_4 - a_{43} b'_4 - \lambda b'_5 - a_{51} b'_5 - a_{52} a_{53} b'_5 - a_{54} b'_5 + \lambda c_5 b'_5 + c_1 (\lambda b'_1 + a_{21} b'_2 + a_{31} b'_3 + a_{41} b'_4 + a_{51} b'_5) + c_2 (\lambda b'_2 + a_{32} b'_3 + a_{42} b'_4 + a_{52} b'_5) + c_3 (\lambda b'_3 + a_{43} b'_4 + a_{53} b'_5) + c_4 (\lambda b'_4 + a_{54} b'_5)$

$\pi_3 = 2\lambda b_1 + \lambda(\lambda+2a_{21})b_2 + 2\lambda b_3 + 2\lambda a_{31}b_3 + 2\lambda a_{32}b_3 + a_{21}a_{32}b_3 + 2\lambda b_4 + 2\lambda a_{41}b_4 + 2\lambda a_{42}b_4 + a_{21}a_{42}b_4 + 2\lambda a_{43}b_4 + a_{31}a_{43}b_4 + a_{32}a_{43}b_4 + 2\lambda b_5 + 2\lambda a_{51}b_5 + 2\lambda a_{52}b_5 + a_{21}a_{52}b_5 + 2\lambda a_{53}b_5 + a_{31}a_{53}b_5 + a_{32}a_{53}b_5 + 2\lambda a_{54}b_5 + a_{41}a_{54}b_5 + a_{42}a_{54}b_5 + a_{43}a_{54}b_5 - 2\lambda b'_1 + 2\lambda c_1 b'_1 - 2\lambda b'_2 - 2\lambda a_{21}a_{31}c_1 b'_3 + a_{21}a_{32}c_1 b'_3 + 2\lambda a_{32}c_2 b'_3 + 2\lambda c3 b'_3 - 2\lambda b'_4 - 2\lambda a_{41}b'_4 - 2\lambda a_{42}b'_4 - a_{21}a_{41}b'_4 - 2\lambda a_{43}b'_4 - a_{31}a_{43}b'_4 - a_{32}a_{43}b'_4 + 2\lambda a_{41}c_1 b'_4 + a_{21}a_{42}c_1 b'_4 + a_{31}a_{43}c_1 b'_4 + 2\lambda a_{42}c_2 b'_4 + a_{32}a_{43}c_2 b'_4 + 2\lambda a_{43}c_3 b'_4 + 2\lambda c_4 b'_4 - 2\lambda b'_5 - 2\lambda a_{51}b'_5 - 2\lambda a_{52}b'_5 - a_{21}a_{52}b'_5 - 2\lambda a_{53}b'_5 - a_{31}a_{53}b'_5 - a_{32}a_{53}b'_5 - 2\lambda a_{54}b'_5 - a_{41}a_{54}b'_5 - a_{42}a_{54}b'_5 - a_{43}a_{54}b'_5 + 2\lambda a_{51}c_1 b'_5 + a_{21}a_{52}c_1 b'_5 + a_{31}a_{53}c_1 b'_5 + a_{41}a_{54}c_1 b'_5 + 2\lambda a_{52}c_2 b'_5 + a_{32}a_{53}c_2 b'_5 + a_{42}a_{54}c_2 b'_5 + 2\lambda a_{53}c_3 b'_5 + a_{43}a_{54}c_3 b'_5 + 2\lambda a_{54}c_4 b'_5 + 2\lambda c_5 b'_5$,

$\pi_4 = 3\lambda b_1 + 2\lambda(\lambda + 3a_{21})b_2 + 3\lambda b_3 + 6\lambda a_{31}b_3 + 6\lambda a_{32}b_3 + 3\lambda a_{21}a_{32}b_3 + 3\lambda b_4 + 6\lambda a_{41}b_4 + 6\lambda a_{42}b_4 + 3\lambda a_{21}a_{42}b_4 + 6\lambda a_{43}b_4 + 3\lambda a_{31}a_{43}b_4 + 3\lambda a_{32}a_{43}b_4 + a_{21}a_{32}a_{43}b_4 + 3\lambda b_5 + 6\lambda a_{51}b_5 + 6\lambda a_{52}b_5 + 3\lambda a_{21}a_{52}b_5 + 6\lambda a_{53}b_5 + 3\lambda a_{31}a_{53}b_5 + 3\lambda a_{32}a_{53}b_5 + a_{21}a_{32}a_{53}b_5 + 6\lambda a_{54}b_5 + 3\lambda a_{41}a_{54}b_5 + 3\lambda a_{42}a_{54}b_5 + a_{21}a_{42}a_{54}b_5 + 3\lambda a_{43}a_{54}b_5 + a_{31}a_{43}a_{54}b_5 + a_{32}a_{43}a_{54}b_5 - 3\lambda b'_1 + 3\lambda c_1 b'_1 - 3\lambda b'_1 - 6\lambda a_{21}b'_1 + 6\lambda a_{21}c_1 b_2 + 3\lambda c_2 b'_2 - 3\lambda b'_3 - 6\lambda a_{31}b'_3 - 6\lambda a_{32}b'_3 - 3\lambda a_{21}a_{32}b'_3 + 6\lambda a_{31}c_1 b'_3 + 3\lambda a_{21}a_{32}c_1 b'_3 + 6\lambda a_{32}c_2 b'_3 + 3\lambda c_3 b'_3 - 3\lambda b'_4 - 6\lambda a_{41}b'_4 - 6\lambda a_{42}b'_4 - 3\lambda a_{21}a_{42}b'_4 - 6\lambda a_{43}b'_4 - 3\lambda a_{31}a_{43}b'_4 - 3\lambda a_{32}a_{43}b'_4 - a_{21}a_{32}a_{43}b'_4 + 6\lambda a_{41}c_1 b'_4 + 3\lambda a_{21}a_{42}c_1 b'_4 + 3\lambda a_{31}a_{43}c_1 b'_4 + a_{21}a_{32}a_{43}c_1 b'_4 + 6\lambda a_{42}c_2 b'_4 + 3\lambda a_{32}a_{43}c_2 b'_4 + 6\lambda a_{43}c_3 b'_4 + 3\lambda c_4 b'_4 - 3\lambda b'_5 - 6a_{51}b'_5 - 6a_{52}b'_5 - 3\lambda a_{21}a_{52}b'_5 - 6\lambda a_{53}b'_5 - 3\lambda a_{31}a_{53}b'_5 - 3\lambda a_{32}a_{53}b'_5 - a_{21}a_{32}a_{53}b'_5 - 6\lambda a_{54}b'_5 - 3\lambda a_{41}a_{54}b'_5 - 3\lambda a_{42}a_{54}b'_5 - a_{21}a_{42}a_{54}b'_5 - 3\lambda a_{43}a_{54}b'_5 - a_{31}a_{43}a_{54}b'_5 - a_{32}a_{43}a_{54}b'_5 + 6\lambda a_{51}c_1 b'_5 + 3\lambda a_{21}a_{52}c_1 b'_5 + 3\lambda a_{31}a_{53}c_1 b'_5 + a_{21}a_{32}a_{53}c_1 b'_5 + 3\lambda a_{41}a_{54}c_1 b'_5 + a_{21}a_{42}a_{54}c_1 b'_5 + a_{31}a_{43}a_{54}c_1 b'_5 + 6\lambda a_{52}c_2 b'_5 + 3\lambda a_{32}a_{53}c_2 b'_5 + 3\lambda a_{42}a_{54}c_2 b'_5 + a_{32}a_{43}a_{54}c_2 b'_5 + 6\lambda a_{53}c_3 b'_5 + 3\lambda a_{43}a_{54}c_3 b'_5 + 6\lambda a_{54}c_4 b'_5 + 3\lambda c_5 b'_5$,

$\pi_5 = 4\lambda b_1 + 3\lambda(\lambda+4a_{21})b_2 + 4\lambda b_3 + 12\lambda a_{31}b_3 + 12\lambda a_{32}b_3 + 12\lambda a_{21}a_{32}b_3 + 4\lambda b_4 + 4\lambda a_{41}b_4 + 12\lambda a_{42}b_4 + 12\lambda a_{21}a_{42}b_4 + 12\lambda a_{43}b_4 + 12\lambda a_{31}a_{43}b_4 + 12\lambda a_{32}a_{43}b_4 + 4\lambda a_{21}a_{32}a_{43}b_4 + 4\lambda b_5 + 12\lambda a_{51}b_5 + 12\lambda a_{52}b_5 + 12a_{21}a_{52}b_5 + 12\lambda a_{53}b_5 + 12\lambda\cdot a_{31}a_{53}b_5 + 12\lambda a_{32}a_{53}b_5 + 4\lambda a_{21}a_{32}a_{53}b_4 + 2\lambda a_{54}b_5 + 12\lambda a_{41}a_{54}b_5 + 12\lambda a_{42}a_{54}b_5 + 4\lambda a_{21}a_{42}a_{54}b_5 + 12a_{43}a_{54}b_5 + 4\lambda a_{31}a_{43}a_{54}b_5 + 4\lambda a_{32}a_{43}a_{54}b_5 + a_{21}a_{32}a_{43}a_{54}b_5 - 4\lambda b'_1 + 4\lambda c_1 b'_1 - 4\lambda b'_2 - 12\lambda a_{21}b'_2 + 12\lambda a_{21}c_1 b'_2 + 4\lambda c_2 b'_2 - 4\lambda b'_3 - 12\lambda a_{31}b'_3 - 12\lambda a_{32}b'_3 - 12\lambda a_{21}a_{32}b'_3 + 12\lambda a_{31}c_1 b'_3 + 12a_{21}a_{32}c_1 b'_3 + 12\lambda a_{32}c_2 b'_3 + 4\lambda c_3 b'_3 - 4\lambda b'_4 - 12\lambda a_{41}b'_4 - 12\lambda a_{42}b'_4 - 12\lambda a_{21}a_{42}b'_4 - 12\lambda a_{43}b'_4 - 12\lambda a_{31}a_{43}b'_4 - 12\lambda a_{32}a_{43}b'_4 - 4\lambda a_{21}a_{32}a_{43}b'_4 + 12\lambda a_{41}c_1 b'_4 + 12\lambda a_{21}a_{42}c_1 b'_4 + 12\lambda a_{31}a_{43}c_1 b'_4 + 4\lambda + a_{21}a_{32}a_{43}c_1 b'_4 + 12\lambda a_{42}c_2 b'_4 + 12\lambda a_{32}a_{43}c_2 b'_4 + 4\lambda a_{43}c_3 b'_4 + 4\lambda c_4 b'_4 - 4\lambda b'_5 - 12\lambda a_{51}b'_5 - 12\lambda a_{52}b'_5 - 12\lambda a_{21}a_{52}b'_5 - 12\lambda a_{53}b'_5 - 12\lambda a_{31}a_{53}b'_5 - 12\lambda a_{32}a_{53}b'_5 - 4\lambda a_{21}a_{32}a_{53}b'_5 - 12\lambda a_{54}b'_5 - 12a_{41}a_{54}b'_5 - 12a_{42}a_{54}b'_5 - 4\lambda a_{21}a_{42}a_{54}b'_5 - 12\lambda a_{43}a_{54}b'_5 - 4\lambda a_{31}a_{43}a_{54}b'_5 - 4\lambda a_{32}\cdot a_{43}a_{54}b'_5 - a_{21}a_{32}a_{43}a_{54}b'_5 + 12\lambda a_{51}c_1 b'_5 + 12\lambda a_{21}a_{52}c_1 b'_5 + 12\lambda a_{31}a_{53}c_1 b'_5 + 4\lambda a_{21}\cdot a_{32}a_{53}c_1 b'_5 + 12\lambda a_{41}a_{54}c_1 b'_5 + 4\lambda a_{21}a_{42}a_{54}c_1 b'_5 + 4\lambda a_{31}a_{43}a_{54}c_1 b'_5 + 12\lambda a_{31}a_{53}c_1 b'_5 + 4\lambda a_{21}a_{32}a_{54}c_1 b'_5 + 12\lambda a_{41}a_{54}c_1 b'_5 + 4\lambda a_{21}a_{31}a_{42}a_{54}c_1 b'_5 + 4\lambda a_{31}a_{43}a_{54}c_1 b'_5 + a_{21}a_{32}\cdot a_{43}a_{54}c_1 b'_5 + 12\lambda a_{52}c_2 b'_5 + 12\lambda a_{32}a_{53}c_2 b'_5 + 12\lambda a_{42}a_{54}c_2 b'_5 + 4\lambda a_{32}a_{43}a_{54}c_2 b'_5 + 12\lambda a_{53}c_3 b'_5 + 12\lambda a_{43}a_{54}c_3 b'_5 + 12\lambda a_{54}c_4 b'_5 + 4\lambda c_5 b'_5$

(2) Then solve equations (4.1)–(4.4) and (4.6) for $\hat{b}'_1$, $\hat{b}'_2$, $\hat{b}'_3$, $\hat{b}'_4$ and $\hat{b}'_5$.

(3) Use the simplifying assumption (4.15) to get $\hat{b}_1$, $\hat{b}_2$, $\hat{b}_3$, $\hat{b}_4$ and $\hat{b}_5$.

(4) Using (4.5), (4.7)–(4.8) and the simplifying assumption (4.14), we solve for λ, a_{21}, a_{31}, a_{32}, a_{41}, a_{42}, a_{51} and a_{54}.

An embedded method of order four requires the evaluation of the coefficients b'_1, b'_2, b'_3, b'_4 and b'_5 satisfying the five equations (4.9)–(4.13). We then get the coefficient vector b using the simplifying assumption (4.16). Unfortunately, this is not possible since the lower order method coincides with the fifth order method. So, we have to decrease the number of stages.

(5) Use (4.9)–(4.13) to solve for b'_1, b'_2, b'_3, b'_4.

(6) Finally, we get expression for b_1, b_2, b_3, b_4 considering the simplifying assumption (4.16).

The above steps with the help of a symbolic manipulation package, we derive expressions for the coefficients explicitly in terms of the free parameters as:

$$\hat{b}'_1 = \frac{\begin{array}{c}12 - 15c_4 - 15c_5 + 20c_4c_5 - 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) + \\ 5c_2(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5)))\end{array}}{60(c_1 - c_2)(c_1 - c_3)(c_1 - c_4)(c_1 - c_5)},$$

$$\hat{b}'_2 = \frac{\begin{array}{c}-12 + 15c_4 + 15c_5 - 20c_4c_5 + 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) - \\ 5c_1(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5)))\end{array}}{60(c_1 - c_2)(c_2 - c_3)(c_2 - c_4)(c_2 - c_5)},$$

$$\hat{b}'_3 = \frac{\begin{array}{c}12 - 15c_4 - 15c_5 + 20c_4c_5 - 5c_2(3 - 4c_5 + c_4(-4 + 6c_5)) + \\ 5c_1(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_2(2 - 3c_5 + c_4(-3 + 6c_5)))\end{array}}{60(c_1 - c_3)(c_2 - c_3)(c_3 - c_4)(c_3 - c_5)},$$

$$\hat{b}'_4 = \frac{\begin{array}{c}-12 + 15c3 + 15c_5 - 20c_3c_5 + 5c_2(3 - 4c_5 + c_3(-4 + 6c_5)) - \\ 5c_1(-3 + c_3(4 - 6c_5) + 4c_5 + 2c_2(2 - 3c_5 + c_3(-3 + 6c_5)))\end{array}}{60(c_1 - c_4)(-c_2 + c_4)(-c_3 + c_4)(c_4 - c_5)},$$

$$\hat{b}'_5 = \frac{\begin{array}{c}12 - 15c_3 - 15c_4 + 20c_3c_4 - 5c_2(3 - 4c4 + c_3(-4 + 6c_4)) + \\ 5c_1(-3 + c_3(4 - 6c_4) + 4c_4 + 2c_2(2 - 3c_4 + c_3(-3 + 6c_4)))\end{array}}{60(c_1 - c_5)(c_2 - c_5)(c_3 - c_5)(c_4 - c_5)},$$

$$\hat{b}_1 = \frac{\begin{array}{c}(1 - c_1)(12 - 15c_4 - 15c_5 + 20c_4c_5 - 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) + \\ 5c_2(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5))))\end{array}}{60(c_1 - c_2)(c_1 - c_3)(c_1 - c_4)(c_1 - c_5)},$$

$$\hat{b}_2 = \frac{\begin{array}{c}(1 - c_2)(-12 + 15c_4 + 15c_5 - 20c_4c_5 + 5c_3(3 - 4c_5 + c_4(-4 + 6c_5)) - \\ 5c_1(-3 + c_4(4 - 6c_5) + 4c_5 + 2c_3(2 - 3c_5 + c_4(-3 + 6c_5))))\end{array}}{60(c_1 - c_2)(c_2 - c_3)(c_2 - c_4)(c_2 - c_5)},$$

$$\hat{b}_3 = \frac{\begin{array}{c}(1-c_3)(12-15c_4-15c_5+20c_4c_5-5c_2(3-4c_5+c_4(-4+6c_5))+\\ 5c_1(-3+c_4(4-6c_5)+4c_5+2c_2(2-3c_5+c_4(-3+6c_5))))\end{array}}{60(c_1-c_3)(c_2-c_3)(c_3-c_4)(c_3-c_5)},$$

$$\hat{b}_4 = \frac{\begin{array}{c}(1-c_4)(-12+15c_3+15c_5-20c3c_5+5c_2(3-4c_5+c_3(-4+6c_5))-\\ 5c_1(-3+c_3(4-6c_5)+4c_5+2c_2(2-3c_5+c_3(-3+6c_5))))\end{array}}{60(c_1-c_4)(-c_2+c_4)(-c_3+c_4)(c_4-c_5)},$$

$$\hat{b}_5 = \frac{\begin{array}{c}(1-c_5)(12-15c_3-15c_4+20c_3c_4-5c_2(3-4c4+c_3(-4+6c_4))+\\ 5c_1(-3+c_3(4-6c_4)+4c_4+2c_2(2-3c_4+c_3(-3+6c_4))))\end{array}}{60(c_1-c_5)(c_2-c_5)(c_3-c_5)(c_4-c_5)},$$

$$b'_1 = -\frac{-3+4c_2+4c_3-6c_2c_3+4c_4-6c_2c_4-6c_3c_4+12c_2c_3c_4}{12(c_1-c_2)(c_1-c_3)(c_1-c_4)}$$

$$b'_2 = -\frac{3-4c_1-4c_3+6c_1c_3-4c_4+6c_1c_4+6c_3c_4-12c_1c_3c_4}{12(c_1-c_2)(c_2-c_3)(c_2-c_4)}$$

$$b'_3 = -\frac{-3+4c_1+4c_2-6c_1c_2+4c_4-6c_1c_4-6c_2c_4+12c_1c_2c_4}{12(c_1-c_3)(c_2-c_3)(c_3-c_4)}$$

$$b'_4 = -\frac{3-4c_1-4c_2+6c_1c_2-4c_3+6c1c_3+6c_2c_3-12c_1c2c_3}{12(c_1-c_4)(c_2-c_4)(c_3-c_4)}$$

$$b_1 = \frac{(-1+c_1)(-3+c_3(4-6c_4)+4c_4+2c_2(2-3c_4+c_3(-3+6c_4)))}{12(c_1-c_2)(c_1-c_3)(c_1-c_4)}$$

$$b_2 = -\frac{(-1+c_2)(-3+c_3(4-6c_4)+4c_4+2c_1(2-3c_4+c_3(-3+6c_4)))}{12(c_1-c_2)(c_2-c_3)(c_2-c_4)}$$

$$b_3 = \frac{(-1+c_3)(-3+c_2(4-6c_4)+4c_4+2c_1(2-3c_4+c2(-3+6c_4)))}{12(c_1-c_3)(c_2-c_3)(c_3-c_4)}$$

$$b_4 = -\frac{(-1+c_4)(-3+c_2(4-6c_3)+4c_3+2c_1(2-3c_3+c_2(-3+6c_3)))}{12(c_1-c_4)(-c_2+c_4)(-c_3+c_4)}$$

$$\lambda = \frac{1}{2}c_1^2, \quad a_{21} = \frac{1}{2}(c_2+c_1)(c_2-c_1), \quad a_{31} = \frac{1}{2}(c_3+c_1)(c_3-c_1) - a_{32}$$

$$a_{41}=\frac{1}{2}(c_4+c_1)(c_4-c_1)-a_{42}-a_{43},\ a_{51}=\frac{1}{2}(c_5+c_1)(c_5-c_1)-a_{52}-a_{53}-a_{54}$$

where

$$a_{32}=\frac{P}{120b_3(c_1-c_2)(c_2-c_4)(c_3-c_4)},\ a_{42}=\frac{Q}{120b_4(c_1-c_2)(c_2-c_4)(c_3-c_4)}$$

$$a_{54}=-\frac{R}{120(b_5c_1c_2-b_5c_1c_4-b_5c_2c_4+b_5c_4^2)}$$

$P=(60b_2c_1c_2^4-60b_4c_1^2c43-60c_2^2(b_2c_1(c_1^2+2c_1c_4-c_4^2)+2a_{52}b_5(c_4-c_5))+60b_2c_1c_2^3(c_1-c_4-c_5)+c_2(-4-120a_{43}b_4c_1c_4+120a_{52}b_5c_1c_4+120b_2c_1^3c_4-60b_4c_1^3c_4+120a_{43}b_4c_3c_4+120a_{52}b_5c_4^2+60b_2c_1^2c_4^2+60b_4c_1^2c_4^2+60b_4c_1c_4^3+60b_1c_1^3(c_1-c_5)-60b_3c_1(c_1^2-c_1c^3-c_3^2)(c_3-c_5)+5c_5+120a_{43}b_4c_1c_5-120a_{52}b_5c_1c_5+60b_4c_1^3c_5-120a_{43}b_4c_3c_5-120a_{52}b_5c_4c_5-60b_4c_1^2c_4c_5-60b_4c_1c_4^2c_5)+5c_4^2(-1-24a_{52}b_5c_1+12b_1c_1^3-12b_2c_1^3-12b_3c_1^3+12b_4c_1^3-12b_5c_1^3-24a_{53}b_5(c_1-c_3)+12b_3c_1^2c_3+12b_3c_1c_3^2+12b_4c_1^2c_5+12b_5c_1^2c_5+12b_5c_1c_5^2)+c_5(-5c_1(-1+24a_{43}b_4c_3+24a_{53}b_5c_3+2(-1+60a_{43}b_4c_3^2+60a_{53}b_5c_3^2)-60c_1^3(b^3c_3+b_5c_5)+60c_1^2(b_3c_3^2+b_5c_5))-c_4(60b_1c_1^4-60c_1^3(2b_2c_3-(b_4-2b_5)c_5)+120c_1^2(b_3c_3^2+b_5c_5^2)-5c_1(-1+24a_{43}b_4c_3-12b_3c_3^2+24a_{52}b_5c_5-12b_5c_5^2+24a_{53}b_5(c_3+c_5))+6(-1+20a_{43}b_4c_3^2+20a_{53}b_5c_3(c_3+c_5))))$

$Q=(60b_2c_1c_2^4-60(2a_{43}b_4+2a_{53}b_5+b_3c_1^2)c_3^3+4c_4-60b_1c_1^4c_4+120a_{43}b_4c_1c_4^2+60b_4c_1^3c_4^2-60b_4c_1^2c_4^2-60b_4c_1c_4^4-60c_2^2(b_2c_1(c_1^2-c_2c_3+c_1(c_3+c_4))+2a_{52}b_5(c_3-c_5)+60b_2c_1c_2^3(c_1-c_4-c_5)-2c_5+5c_1c_5+120a_{52}b_5c_1c_4c_5+120a_{53}b_5c_1c^4c_5-60b_4c_1^3c_4c_5+60b_5c_1^3c_4c_5+60b_4c_1^2c_4^2c_5-60b5c_1^3c_5^2-60b_5c_1^2c_4c_5^2+60b_5c_1^2c_5^3-60b_5c_1c_4c_5^3+c_2(-4+60b_2c_1^3c_3-60b_3c_1^3c_3+60b_3c_1^2c_3^2+60b_3c_1c_3^3-120a_{43}b_4c_1c_4+60b_2c_1^3c_4-60b4c_1^3c_4+120a_{43}b_4c_3c_4+60b2c_1^2c_3c_4+60b_4c_1^2c_4^2+60b_4c_1c_4^3+60b_1c_1^3(c_1-c_5)+120a_{52}b_5(c_1+c_4)(c_3-c_5)+5c_5+120a_{43}b_4c_1c_5+60b_3c_1^3c_5+60b_4c_1^3c_5-120a_{43}b_4c_3c_5-60b_3c_1^2c_3c_5-60b_3c_1c_3^2c_5-60b_4c_1^2c_4c_5-60b_4c_1c_42c_5)+60c_3^2(b_3c_1^2(c_1+c_5)+2a_{43}b_4(c_1+c_4+c_5)+2a_{53}b_5(c_1+c_4+c_5)-c_3(-2+120a_{43}b_4c_4^2-60b_5c_1^2(c_4-c_5)-60c_1^3(b_1c_4-b_2c_4-b_5c_4-b_3c_5+b_5c_5)+5c_4(1+24a_{53}b_5c_5)+5c_1(1+24a_{52}b_5c_4+24a_{53}b_5c_4-12b_4c_4^3+24a_{53}b_5c_5-12b_5c_4c_5^2+24a_{43}b_4(c_4+c_5))))$

$R=(-2+5c_1+5c_2+120a_{43}b_4c_1c_2+120a_{53}b_5c_1c_2-60b_1c_1^3c_2+60b_3c_1^3c_2+60b_4c_1^3c_2+60b_5c_1^3c_2-60b_2c_1c_2^3-120a_{43}b_4c_1c_3-120a_{53}c_1c_3-60b_3c_1^3c_3-120a_{43}b_4c_2c_3-120a_{53}b_5c_2c_3-60b_3c_1^2c_2c_3+120a_{43}b_4c_3^2+120a_{53}b_5c_3^2+60b_3c_1^2c_2^3-60b_3c_1c_2c_3^2-60b_4c_1^3c_4-60b_4c_1^2c_2c_4+60b_4c_1^2c_4^2-60b_4c_1c_2c_4^2-60b_5c_1^3c_5-60b_5c_1^2c_2c_5+60b_5c_1^2c_5^2-60b_5c_1c_2c_5^2$

These led us to the derivation of the $\det(R(z))$ in (4.17) in terms of the free parameters. In order to achieve a 5-stage, fourth-and fifth-order embedded RKN method that is zero dissipative, Definition 3.1 has to be satisfied. Match the coefficients of z, z^2, z^3, z^4 and z^5 of the numerator and the denominator of (4.17). The free parameters can be used to get an optimal method by minimizing the norms of the truncation error constant according to [12] given by

$$\left\|\tau^{(6)}\right\|_2=\sqrt{\sum_{j=1}^{N_6}\left(\tau_j^{(6)}\right)^2},\qquad \left\|\tau'^{(6)}\right\|_2=\sqrt{\sum_{j=1}^{N_6'}\left(\tau_j'^{(6)}\right)^2} \tag{4.18}$$

where

$$\tau_1^{(6)} = \tau_2^{(6)} = \tau_3^{(6)} = \sum_i b_i c_i^4 - \frac{1}{30},$$

$$\tau_4^{(6)} = \sum_{i,j} b_i c_i a_{ij} c_j - \frac{1}{180}$$

$$\tau_5^{(6)} = \tau_6^{(6)} = \sum_{i,j} b_i a_{ij} c_j^2 - \frac{1}{360}$$

$$\tau_1'^{(6)} = \tau_2'^{(6)} = \tau_3'^{(6)} = \sum_i b_i c_i^4 - \frac{1}{6},$$

$$\tau_4'^{(6)} = \tau_5'^{(6)} = \sum_{i,j} b_i c_i^2 a_{ij} c_j - \frac{1}{36},$$

$$\tau_6'^{(6)} = \sum_{i,j} b_i c_i a_{ij} c_j^2 - \frac{1}{72}$$

$$\tau_7'^{(6)} = \tau_8'^{(6)} = \tau_9'^{(6)} = \sum_{i,j} b_i a_{ij} c_j^3 - \frac{1}{24},$$

$$\tau_{10}'^{(6)} = \sum_{i,j,k} b_i a_{ij} a_{jk} c_k - \frac{1}{720}$$

Then substituted the expressions derived above in terms of the free parameters into the principal truncation error coefficients (4.18). The MATHEMATICA package is employed to minimize (4.18) subject to the constraints $\det(R(z)) = 1$ and the bounds $0 \le c_i \le 1$, $i = 1, 2, 3, 4, 5$ to determine the values for the free parameters. The resulting zero dissipative DIRKN pairs of orders 5(4), expressed in Butcher tableau form is;

$$\begin{array}{c|ccccc}
\frac{707}{1000} & \frac{499849}{2000000} & & & & \\
\frac{1}{5} & -\frac{459849}{2000000} & \frac{499849}{2000000} & & & \\
\frac{2}{5} & -\frac{31620096730133}{78774963858500} & \frac{372444553886961}{157549927717000} & \frac{499849}{2000000} & & \\
\frac{3}{5} & -\frac{8552093414343}{545506033572500} & \frac{122017348397423}{54550633572500} & -\frac{273}{1000} & \frac{499849}{2000000} & \\
\frac{9}{10} & -\frac{21219733103899}{11845750157800} & -\frac{1}{10} & \frac{3}{20} & \frac{2252276562125}{11845750157800} & \frac{499849}{2000000} \\
\hline
b & \frac{6446000}{16654443} & \frac{2530}{845} & \frac{422}{1535} & -\frac{1481}{3210} & 0 \\
b' & \frac{22000000}{16654443} & \frac{253}{676} & \frac{422}{921} & -\frac{1481}{1284} & 0 \\
\hline
\hat{b} & -\frac{2519800000}{9642922447} & \frac{26608}{53235} & -\frac{6449}{15350} & \frac{3113}{4815} & \frac{10669}{303975} \\
\hat{b}' & \frac{8600000000}{9642922497} & \frac{6652}{10647} & \frac{6449}{9210} & \frac{3113}{1926} & \frac{21338}{60795}
\end{array}$$

The above coefficients are substituted into the error equations of the sixth order method both for y and y' respectively to obtain the error constants of the method as

$$\left\|\tau^{(6)}\right\|_2 = 1.5704582765 \cdot 10^{-3}, \quad \left\|\tau'^{(6)}\right\| = 2.3320316216 \cdot 10^{-2}$$

Examine the stability interval of the method by plotting the stability region. Using (3.2) the amplification matrix is obtained as

$$R(z) = \begin{pmatrix} 1+\frac{1}{2}z+\frac{43}{1000}z^2+\frac{9}{2000}z^3+\frac{3}{250}z^4+\frac{3}{250}z^5 & 1+\frac{83}{500}z+\frac{1}{125}z^2+\frac{111}{5000}z^3+\frac{131}{5000}z^4+\frac{23}{1250}z^5 \\ z+\frac{43}{1250}z^2+\frac{1}{125}z^3+\frac{1}{200}z^4+\frac{99}{5000}z^5 & 1+\frac{1}{2}z+\frac{23}{625}z^2+\frac{1}{400}z^3+\frac{73}{2000}z^4+\frac{21}{500}z^5 \end{pmatrix}$$

A boundary locus plot of $R(z)$ gives the stability interval of approximately $(0, 6.52)$. Figure 4.1 depicts the stability region of the higher order method used to advance the integration.

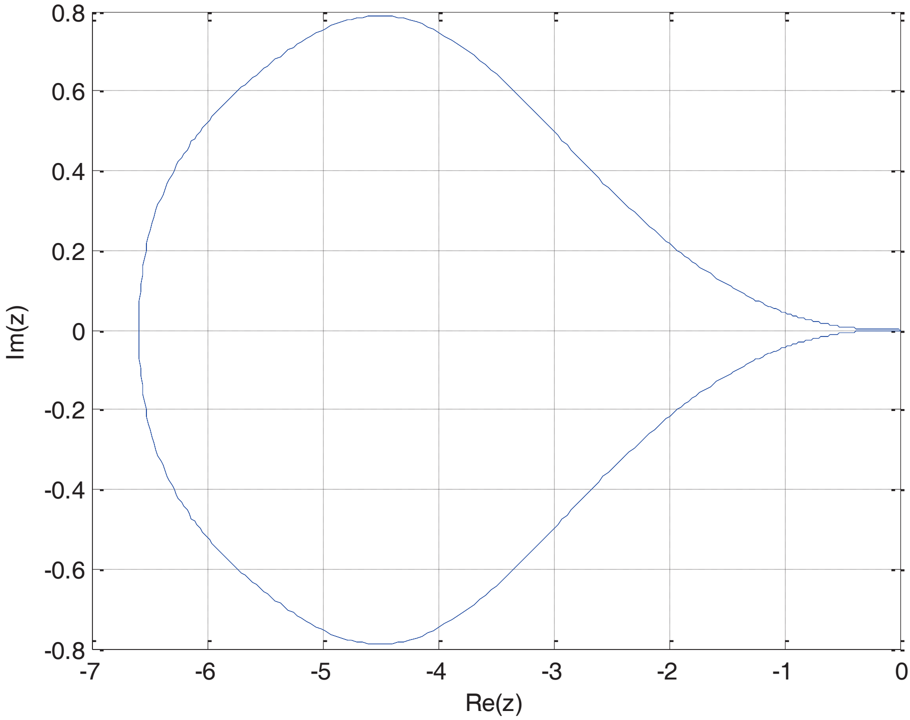

Figure 4.1 Stability Region for the Fifth Order Zero Dissipative DIRKN Pairs of Order 5(4)

5 Numerical Examples

To illustrate the applicability of the method derived in this paper, we consider some test problems which appear in many papers on numerical methods for oscillatory problems. We compare our method with the method of [2], when the same test problems are reduced to first order system twice the dimension of (1.1). We apply the methods with variable step size and use the technique of simplified Newton iterations to handle the nonlinear systems.

The following notations are used in Tables 1–3:
New method: The DIRKN 5(4) pairs obtained in this paper.
Method1: The method obtained by [2].
Tol: The tolerance used.
MTD: Method used.
FCN: The number of functions evaluated.
STEP: The number of steps.
JAE: The number of Jacobian evaluation.
FS: The number of failed steps.
ACPT: The number of accepted steps.
EMAX: $\max \|y_n - y(x_n)\|$, that is the computed solution minus the exact solution.

Problem 5.1 Consider the equation

$$y'' = -y + x,\ y(0) = 1,\ y'(0) = 2,\ 0 \le x \le 16\pi.$$

Exact solution: $y(x) = \sin(x) + \cos(x) + x$. Source: [1]

Problem 5.2 Our second test problem was an inhomogeneous problem

$$y'' = -100y + 99\sin(x),\ y(0) = 1,\ y'(0) = 11$$

Exact solution: $y(x) = \cos(10x) + \sin(10x) + \sin(x)$
We integrate the problem in the interval $x \in [010\pi]$. Source: [17]

Problem 5.3 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

$$\begin{aligned} y(x) = {} & 0.200179477536\cos(1.01x) + 0.00246946143\cos(3.03x) \\ & +0.304014 \cdot 10^{-6}\cos(5.05x) + 0.374 \cdot 10^{-9}\cos(7.07x) \end{aligned}$$

Source: [13]

Table 1 Numerical Results for Problem 5.1

MTD	Tol	FCN	STEP	JAE	FS	ACPT	EMAX
New method Method1	10^{-2}	976 10366	882 797	1 1	0 0	882 797	$1.1542519{\cdot}10^{-01}$ $5.8080693{\cdot}10^{-04}$
New method Method1	10^{-4}	26707 26880	2413 2067	1 1	16 1	2397 2066	$4.0235515{\cdot}10^{-04}$ $2.4486472{\cdot}10^{-02}$
New method Method1	10^{-6}	85927 67691	7807 5206	0 1	0 1	7807 5205	$9.6516201{\cdot}10^{-07}$ $2.5021113{\cdot}10^{-02}$
New method Method1	10^{-8}	216716 170070	19700 13081	0 1	2 3	19698 13078	$1.2655876{\cdot}10^{-08}$ $2.5034278{\cdot}10^{-02}$
New method Method1	10^{-10}	2675520 1923413	21933 28914	1 1	2 3	21931 28911	$2.9308203{\cdot}10^{-09}$ $7.2879331{\cdot}10^{-03}$

Table 2 Numerical Results for Problem 5.2

MTD	Tol	FCN	STEP	JAE	FS	ACPT	EMAX
New method Method1	10^{-2}	1740 2537	158 195	1 1	0 0	158 195	$1.5221335 \cdot 10^{-02}$ $3.7157173 \cdot 10^{-02}$
New method Method1	10^{-4}	4655 6385	423 491	1 1	0 0	423 491	$6.3053737 \cdot 10^{-06}$ $2.3614952 \cdot 10^{-03}$
New method Method1	10^{-6}	11783 16044	1071 1234	1 1	0 0	1071 1234	$2.5758336 \cdot 10^{-08}$ $1.4874384 \cdot 10^{-04}$
New method Method1	10^{-8}	29614 40289	2692 3009	1 1	0 0	2692 3009	$4.0352437 \cdot 10^{-10}$ $9.3760445x10^{-06}$
New method Method1	10^{-10}	35229 43286	2977 3705	1 1	0 0	2977 3705	$3.2937681 \cdot 10^{-12}$ $2.4678332x10^{-08}$

Table 3 Numerical Results for Problem 5.3

MTD	Tol	FCN	STEP	JAE	FS	ACPT	EMAX
New method Method1	10^{-2}	9018 20928	819 3201	1 1	1 1	818 3201	$2.0586713 \cdot 10^{-03}$ $6.7846093 \cdot 10^{-01}$
New method Method1	10^{-4}	23590 52786	2144 8120	1 1	1 1	2143 8119	$4.1773814 \cdot 10^{-05}$ $4.2585692 \cdot 10^{-02}$
New method Method1	10^{-6}	59505 13273	5409 20420	1 1	1 1	5408 20419	$3.0718335 \cdot 10^{-07}$ $2.6638752 \cdot 10^{-03}$
New method Method1	10^{-8}	149631 333534	13601 51312	1 1	2 1	13598 51310	$3.6508864 \cdot 10^{-09}$ $1.6754552 \cdot 10^{-04}$
New method Method1	10^{-10}	186350 429883	19270 58239	1 1	2 1	19298 58238	$3.3133562 \cdot 10^{-10}$ $1.6754557 \cdot 10^{-05}$

The results of our experiment show the applicability of the zero dissipative embedded DIRKN method which we have derived. From the numerical results in Tables 1–3, we observed that the "New method" produces better results compared to Method1 in terms of number of steps, number of functions evaluations which are due to the fact that when Method1 is used to solve second order equation, it has to be reduced to first-order system. Also, in terms of global error, the "New method" produces smaller errors compared to Method1; however, the error produced by Method1 is within the chosen tolerance.

Figures 4.2–4.4 also presented the efficiency of the method developed by plotting the graphs of the maximum global error (log 10) against the computational cost measured by the number of function evaluations required by each method. The New method is more efficient than the Method I for the tested problems. This validates the fact that the zero dissipative is an important property when solving IVPs in which the solution is in oscillating form.

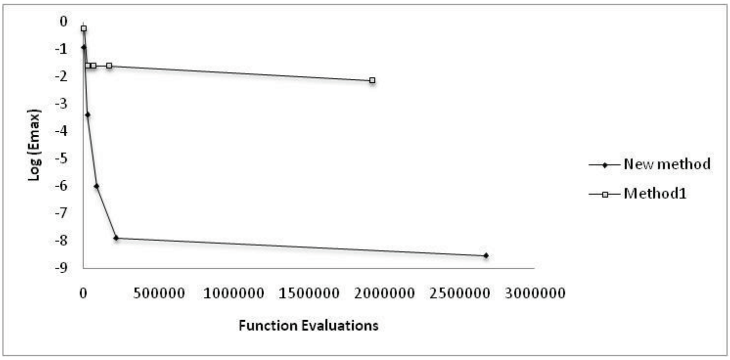

Figure 4.2 Error Plots for Problem 5.1

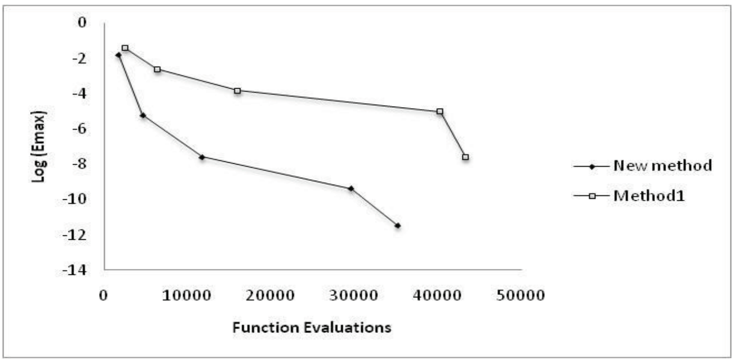

Figure 4.3 Error Plots for problem 5.2

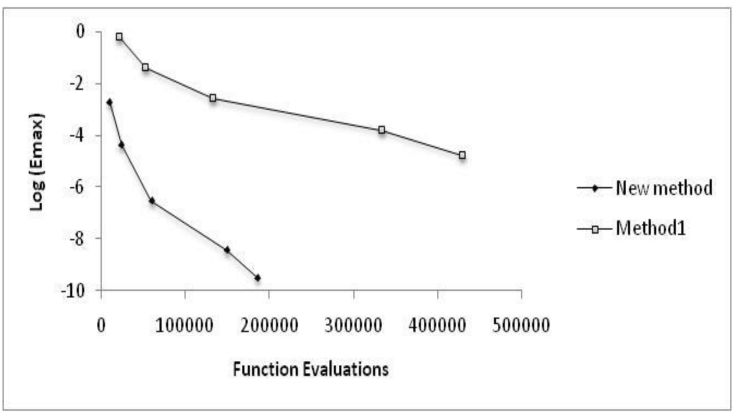

Figure 4.4 Error Plots for problem 5.3

6 Conclusion

This paper derives zero dissipative DIRKN pairs of order 5(4) at a cost of five stages per step for solving the IVPs (1.1). The method controls the local error in the solution and the derivative. It has an appropriate region of stability and computationally less expensive due to its diagonally implicit structure. In view of the numerical results obtained, we conclude that the New method shows some promise compared to Method1. Thus the New method can be used to solve the IVPs (1.1) directly without having to reduce the problems to first order system hence computational time is minimized.

References

[1] Allen, R. C., Wing, G. M.: *An invariant imbedding algorithm for the solution of inhomogeneous linear two-point boundary value-problems.* J. Computer Physics **14** (1974), 40–45.

[2] Butcher, J. C., Chen, D. J.: *A new Type of Singly-implicit Runge–Kutta Method.* Applied Numerical Mathematics **34** (2000), 179–188.

[3] Dormand, J. R., El-Mikkawy, M. E. A., Prince, P. J.: *Families of Runge–Kutta Nystrom Formulas.* IMA J. Num. Analysis **7** (1987), 235–250.

[4] El-Mikkawy, M., El-Dousky, R. A.: *New optimized non-FSAL Embedded Runge–Kutta Nystrom algorithms of orders 6 and 4 in six stages.* Applied Mathematics and computer **145** (2003), 33–43.

[5] Fehlberg, E.: Classical eight and lower-order Runge–Kutta–Nystrom formulae with step size control for the special second-order differential equations. Technical Report, **R-381**, *NASA*, 1972.

[6] Fillipi, S., Graf, J.: *New Runge–Kutta–Nystrom Formulae pairs of order* 8(7), 9(8), 10(9) *and* 11(10) *for differential equation of the form* $y'' = f(x, y)$. J. computing Appl. Math. **14** (1986), 362–370.

[7] Ferracina, L., Spijker, M. N.: *Strong stability of singly-diagonally implicit Runge–Kutta methods.* Applied Numerical Mathematics **58** (2008), 1675–1686.

[8] Hairer, E., Norsett, S. P., Wanner, G.: Solving Ordinary Differential Equations I, Non Stiff problems. *springer-Verlag*, Berlin, 1993.

[9] Imoni, S. O., Otunta, F. O., Ramamohan, T. R.: *Embedded implicit Runge–Kutta–Nystrom method for solving second order differential equations.* International Journal of Computer Mathematics **83**, 11 (2006), 777–784.

[10] Imoni, S. O., Otunta, F. O., Ikhile, M. N. O.: *Embedded diagonally implicit Runge–Kutta–Nystrom method of order two and three for special second order differential equations.* The Journal of the Mathematical Association of Nigeria (ABACUS) **34**, 213 (2007), 363–373.

[11] Kanagarajam, K., Sambath, M.: *Runge–Kutta–Nystrom method of order three for solving Fuzzy Differential Equations.* Computational methods in Applied Mathematics **10**, 2 (2010), 195–203.

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

[13] Qinghong, Li, Yongzhong, S.: *Explicit one-step p-stable methods for second order periodic initial value problems.* Numerical Mathematics, Journal of Chinese Universities (English series) **15**, 3 (2006), 237–247.

[14] Sharp, P. W., Fine, J. M.: *A contrast of direct and transformed Nystrom pairs.* J. Comput. Appl. Math **42** (1992), 293–308.

[15] Sharp, P. W., Fine, J. M., Burrage, K.: *Two-stage and three-stage diagonally implicit Runge–Kutta–Nystrom methods of orders three and four.* IMA Journal Numerical Analysis **10** (1990), 489–504.

[16] Sommeijer, B. P.: *A note on a diagonally implicit Runge–Kutta–Nystrom Method.* J. Comput. Appl. Math. **19** (1987), 395–399.

[17] Simos, T. E., Famelis, I. T., Tsitouras, Ch.: *Zero dissipative, explicit Numerov-type methods for second order IVPs with oscillating solutions.* Numerical Algorithms **34** (2003), 27–40.

[18] Simos, T. E., Dimas, E., Sideridis, A. B.: *A Runge–Kutta–Nystrom method for the numerical integration of special second-order periodic initial-value problems.* J. Computational and Applied Maths. **51** (1994), 317–326.

[19] Van der Houwen, P. J., Sommeijer, B. P.: *Diagonally Implicit Runge–Kutta–Nystrom methods for oscillatory problems.* SIAM J. Numerical Analysis **26**, 2 (1989), 414–429.