World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:2, No:2, 2008

Embedded Singly Diagonally Implicit Runge-Kutta –Nystrom Method Order 5(4) for the Integration of Special Second Order ODEs

Fudziah Ismail

***Abstract*—**In this paper a new embedded Singly Diagonally Implicit Runge-Kutta Nystrom fourth order in fifth order method for solving special second order initial value problems is derived. A standard set of test problems are tested upon and comparisons on the numerical results are made when the same set of test problems are reduced to first order systems and solved using the existing embedded diagonally implicit Runge-Kutta method. The results suggests the superiority of the new method.

***Keywords*—**Runge-Kutta Nystrom, Special second order problems.

I. Introduction

SEVERAL methods have been proposed for the numerical solutions of the special second-order ordinary differential equations (ODEs) of the form:

$$y'' = f(x,y),\ y(x_0) = y_0,\ y'(x_0) = y'_0 \qquad (1)$$

In which f does not depend on y'. In general the second order equation (1) can be reduced to an equivalent first-order system of twice the dimension and solved using the standard Runge-Kutta (RK) method. However, it is more efficient if the equation can be solved directly using Runge-Kutta Nystrom (RKN) method, such work can be seen in Sharp and Fine [1], Dormand, El-Mikkawy and Prince [2] and El-Mikkawy and El- Desouky [3]. Generally efficient Runge-Kutta and Runge-Kutta Nystrom codes involved the embedded pairs of orders $q(p)$ where the method of order $q = p+1$ is used to obtain the numerical solutions of the problems and the method of order p is used to obtain the local truncation error, hence, the next step of the integration can be calculated. In this paper we are going to derive embedded pairs which are diagonally implicit and all the diagonal element are equal, such method is very efficient in solving stiff differential equations, since the iteration matrix $\left(I - h^2\gamma\, J\right)$ (where J is the Jacobian of the system of equations) of the Newton iteration can be used in all stages.

The Runge-Kutta Nystrom pair generates the approximations $y_{n+1}, \bar{y}_{n+1}, y'_{n+1}$, $\bar{y'}_{n+1}$ to $y(x_{n+1})$ and $y'(x_{n+1})$, according to the following:

$$y_{n+1} = y_n + hy'_n + h^2\sum_{i=1}^{s} b_i k_i \ \ , y'_{n+1} = y'_n + h\sum_{i=1}^{s} b'_i k_i \qquad (2a)$$

$$\bar{y}_{n+1} = y_n + hy'_n + h^2\sum_{i=1}^{s-1} \bar{b}_i k_i\,,\ \bar{y'}_{n+1} = y'_n + h\sum_{i=}^{s-1} \bar{b'}_i k_i \qquad (2b)$$

The first two formulae are order q and the second two are order p and

$$k_i = f(x_n + c_i h,\ y_n + c_i hy'_n + h^2\sum_{j=1}^{i} a_{ij}k_j)\,, \qquad (2c)$$
$$i = 1,2,\cdots,s.$$

The method can also be written in Butcher Tableau

C	A
	b
	b'
	$\bar{b}$
	$\bar{b'}$

($C = [c_1, c_2, \cdots, c_s]^T$, $A = [a_{ij}]$, $b = [b_1, b_2, \cdots, b_s]$, $b' = [b'_1, b'_2, \cdots, b'_s]$, $\bar{b} = [\bar{b}_1, \bar{b}_2, \cdots, \bar{b}_{s-1}]$, $\bar{b'} = [\bar{b'}_1, \bar{b'}_2, \cdots, \bar{b'}_{s-1}]$. We refer to (2) as the Runge-Kutta Nystrom pair, where the approximations of order $p+1$ are being advanced from step to step and the approximations of order p is used for the local truncation error so that the next stepsize can be obtained based on the local truncation error. Hence, the code developed here is the variable stepsize code.

II. Derivation of the Method

According to Papageorgiou, Famwlis and Tsitouras [4] the coefficients of a fifth order Runge-Kutta-Nystrom method must satisfy the following order conditions after using two basic simplifying assumptions.

Fudziah Ismail is with the Department of Mathematics, Universiti Putra Malaysia, Serdang 43400, Selangor, Malaysia (phone: 603-89466811; fax: 603-89437958; e-mail: fudziah_i@yahoo.com.my).

TABLE I
ORDER CONDITIONS FOR y'

Order conditions:	1	$\sum_i b_i' = 1$	(3)
Order conditions:	2	$\sum_i b_i' c_i = \frac{1}{2}$	(4)
Order conditions:	3	$\sum_i b_i' c_i^2 = \frac{1}{3}$	(5)
Order conditions:	4	$\sum_i b_i' c_i^3 = \frac{1}{4}$	(6)
		$\sum_{ij} b_i' a_{ij} c_j = \frac{1}{24}$	(7)
Order conditions:	5	$\sum_i b_i' c_i^4 = \frac{1}{5}$	(8)
		$\sum b_i' a_{ij} c_j^2 = \frac{1}{60}$	(9)
		$\sum_{ij} b_i' c_i a_{ij} c_j = \frac{1}{60}$	(10)

TABLE II
ORDER CONDITIONS FOR $\overline{y'}$

Order conditions:	1	$\sum_i \overline{b_i'} = 1$	(3)*
Order conditions:	2	$\sum_i \overline{b_i'} c_i = \frac{1}{2}$	(4)*
Order conditions:	3	$\sum_i \overline{b_i'} c_i^2 = \frac{1}{3}$	(5)*
Order conditions:	4	$\sum_i \overline{b_i'} c_i^3 = \frac{1}{4}$	(6)*
		$\sum_{ij} \overline{b_i'} a_{ij} c_j = \frac{1}{24}$	(7)*

The simplifying assumptions are

$$\sum_{ij} a_{ij} = \frac{1}{2} c_i^2 \tag{11}$$

$$b_i = b'(1 - c_i) \tag{12}$$

From which the values of a_{i1} and b_i for $i = 1,2,\cdots,s$ can be obtained respectively. The proof of using the simplifying assumptions can be found in Hairer and Wanner [5].

For the fifth order method all eight order equations for y' in Table I have to be satisfied and for the fourth order method all the $\overline{b_i'}$ have to satisfy all the equations in Table II and the values of $\overline{b_i}$ can be obtained from the following equation:

$$\overline{b_i} = \overline{b_i'}(1 - c_i) \tag{13}$$

Here we are going to derive embedded singly diagonally implicit RKN method of order 4, 4-stage in order 5, 5-stage or can be written as SDIRKN 5(4) method and γ is the diagonal element.

Algorithm to find the coefficients of SDIRKN 5(4) method.

Step 1: Let $\gamma = 0.25$

Step 2: $c_1 = \sqrt{2\gamma}$ (from equation 11 for i=1).

Step 3: Give the values of c_2, c_3, c_4, c_5, with all the values of c's obtain b_1', b_2', b_3', b_4' and b_5' from equations (3),(4),(5),(6) and (8).

Step 4: Use (12) to get the values of b_1, b_2, b_3, b_4 and b_5.

Step 5: Use (3)*, (4)*,(5)* and (6)* and solve for $\overline{b_1'}, \overline{b_2'}, \overline{b_3'}$ and $\overline{b_4'}$.

Step 6: Use (13) to get the values of $\overline{b_1}, \overline{b_2}, \overline{b_3}$ and $\overline{b_4}$.

Step 7: Give the values of a_{52} and a_{54} and use (7), (7)*, (9) and (10) to obtain the values of a_{ij} where $i \neq 1$.

Step 8: Use (11) to get the values of a_{i1}.

By using MAPLE, the following are the parameters of the SDIRKN 5(4) method with free parameters are chosen to be $\gamma = 0.25$, $c_2 = 0.2$, $c_3 = 0.4$, $c_4 = 0.6$, $c_5 = 0.9$, $a_{52} = -0.1$ and $a_{53} = 0.15$

$c_1 = 0.7071067811865475$
$c_2 = 0.2$
$c_3 = 0.4$
$c_4 = 0.6$
$c_5 = 0.9$
$a_{11} = a_{22} = a_{33} = a_{44} = a_{55} = 0.25$
$a_{21} = -0.23$
$a_{31} = -0.3925002502501825$
$a_{32} = 0.2225002502501825$
$a_{41} = -0.008891426702213870$
$a_{42} = 0.2120976370788504$
$a_{43} = -0.2732062103766366$

$a_{51} = -1.672156796751771$
$a_{52} = -0.1$
$a_{53} = 0.15$
$a_{54} = 1.777156796751771$
$b_1' = -0.8909522811353591$
$b_2' = 0.6247556718194198$
$b_3' = -0.7000548195198433$
$b_4' = 1.615190309345717$
$b_5' = 0.3510611194900651$
$b_1 = -0.2609538814309234$
$b_2 = 0.4998045374555358$
$b_3 = -0.4200328917119060$
$b_4 = 0.6460761237382868$
$b_5 = 0.03510611194900651$
$\overline{b_1'} = 1.318915246389200$
$\overline{b_2'} = 0.3743745692181844$
$\overline{b_3'} = 0.4575746950566785$
$\overline{b_4'} = -1.150864510664063$
$\overline{b_1} = 0.3863013318570706$
$\overline{b_2} = 0.2994996553745475$
$\overline{b_3} = 0.2745448170340071$
$\overline{b_4} = -0.4603458042656252$

The above coefficients are substituted into the error equations of the sixth order method both for y and y' see Dormand [6] and we obtained the error norm of the method which is:

$\left\|\tau^{(6)}\right\|_2 = 8.29696x10^{-3}$ and $\left\|\tau'^{(6)}\right\|_2 = 9.94999x10^{-3}$.

Where $\left\|\tau^{(6)}\right\|_2 = \sqrt{\sum_{j=1}^{n6}\left(\tau_j^{(6)}\right)^2}$ and

$\left\|\tau'^{(6)}\right\|_2 = \sqrt{\sum_{j=1}^{n_6}\left(\tau'^{(6)}_j\right)^2}$, τ and τ' are error equations associated with the method. The error norms are smaller compared to other methods such as method in [4] which has error norms 10^{-2}.

III. Test Problems

Below are some of the problems tested:

Problem 1: A nonlinear problem.

$$y'' + 100y = \sin(y)$$
$$y(0) = 0, \quad y'(0) = 1, \quad 0 \le x \le 20\pi$$

There is no true solution but the value at 20π is 0.000392823991.
Source: Chawla and Rao [7]

The first order system: The new variables are $y_1 = y$ and $y_2 = y'$

$$y_1' = y_2$$
$$y_2' + 100y_1 = \sin(y_1)$$
$$y_1(0) = 0, \quad y_2(0) = 1 \qquad 0 \le x \le 20\pi$$

Problem 2:

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

Solution: $y(x) = \sin(x) + \cos(x) + x$
Source: Allen and Wing [8]

The first order system :

$$y_1' = y_2$$
$$y_2' = -y_1 + x \quad , \quad y_1(0) = 1, \quad y_2(0) = 2 \quad ,$$
$$0 \le x \le 20\pi$$

Solutions:
$y_1(x) = \sin(x) + \cos(x) + x,$
$y_2(x) = \cos(x) - \sin(x) + 1.$

Problem 3:

$$y_1'' = -4x^2 y_1 - \frac{2y_2}{\sqrt{y_1^2 + y_2^2}} \ ,$$
$$y_1(x_0) = 0, \quad y_1'(x_0) = -\sqrt{2\pi}$$
$$y_2'' = -4x^2 y_2 + \frac{2y_1}{\sqrt{y_1^2 + y_2^2}} \ ,$$
$$y_2(x_0) = 1, \quad y_2'(x_0) = 0$$
$$\sqrt{\frac{\pi}{2}} \le x \le 5\pi$$

Solution: $y_1(x) = \cos(x^2), \quad y_2(x) = \sin(x^2)$

Source: Sharp and Fine [1]

The first order system:

$$y_1' = y_3, \ y_2' = y_4, \ y_3' = -4x^2 y_1 - \frac{2y_2}{\sqrt{y_1^2 + y_2^2}} \ ,$$
$$y_4' = -4x^2 y_2 + \frac{2y_1}{\sqrt{y_1^2 + y_2^2}}$$
$$y_1(x_0) = 0, \quad y_2(x_0) = 1, \quad y_3(x_0) = -\sqrt{2\pi},$$
$$y_4(x_0) = 0$$

Solution: $y_1(x) = \cos(x^2), \quad y_2(x) = \sin(x^2),$
$y_3(x) = -2x\sin(x^2), \quad y_4(x) = 2x\cos(x^2).$

Problem 4: (The two-body gravitational problem)

$$y_1'' = \frac{-y_1}{\left(\sqrt{y_1^2 + y_2^2}\right)^3}, \quad y_1(0) = 1,\ y_1'(0) = 0$$

$$y_2'' = \frac{-y_2}{\left(\sqrt{y_1^2 + y_2^2}\right)^3}, \quad y_2(0) = 0, \quad y_2'(0) = 1$$

$0 \le x \le 16\pi$,

Solution: $y_1(x) = \cos(x), \quad y_2(x) = \sin(x)$.

Source: Dorman *et. al.* [2]

The first order system:

$$y_1' = y_3,\ y_2' = y_4,\ y_3' = \frac{-y_1}{(\sqrt{y_1^2 + y_2^2})^3}, \quad y_4' = \frac{-y_2}{(\sqrt{y_1^2 + y_2^2})^3}$$

$$y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad y_4(0) = 1$$

Solutions: $y_1(x) = \cos(x), \quad y_2(x) = \sin(x)$,

$y_3(x) = -\sin(x), \quad y_4(x) = \cos(x)$.

IV. Implementation and Numerical Results

The set of tested problems in section III is solved using the new method and the results are compared with the numerical results when the same set of test problems are reduced to first order system twice the dimension and solve using method by Butcher and Chen [9].

For all the problems, they are considered as nonstiff and solve using simple iterations where every *k's* are iterated three times once stiffness is detected through $h_{acc} > h_{iter}$, the whole system is considered stiff and solve using Newton iterations. h_{acc} is the stepsize which is expected to meet the specified accuracy .

h_{iter} is the stepsize which will make the iteration converge. Coefficient matrix for the Newton iteration is $\left(I - h^2\gamma J\right)$. Local truncation error : LTE = $y_n - \overline{y_n} = h^2 \sum_i^s (b_i k_i - \bar{b}_i k_i)$.

In this paper we just control the stepsize for *y* because we feel that the formula for y contained the value of y' , thus controlling y means we are also indirectly controlling the value of y'.

For the new method (F1) the next stepsize is

$$h_{new} = 0.5\left[\frac{tol}{2\left|y_n - \overline{y_n}\right|}\right]^{\frac{1}{p+1}} h_{old}$$

and for the second method (B1)

$$h_{new} = 0.2\left[\frac{tol}{2\left|y_n - \overline{y_n}\right|}\right]^{\frac{1}{p+1}} h_{old}$$

Where *tol* is the chosen tolerance, h_{old} is the current stepsize, p is the order of the method and $\left|y_n - \overline{y_n}\right|$ is the local truncation error. The reason why we used safety factor 0.2 here is that to keep the global error small enough so that it is comparable with method F1.

Some of the notations used:

2.345789 (10^{-2}) means 2.345789 x 10^{-2}

TOL ~ The tolerance chosen

MTD ~ The method used.

F1 - SDIRKN 5(4) which has been derived in this paper.

B1 - SDIRK 3rd order 4-stage embedded in 4th order 5-stage method by Butcher and Chen [9].

FCN ~ Number of functions evaluated.

STEP ~ Number of steps.

JAC ~ The number of Jacobian evaluation.

FS ~ The number of failed steps.

GE ~ Maximum global error (max $\|y_n - y(x_n)\|$), that is the computed solution minus the true solution.

TABLE III
Numerical Results for Problem 1

TOL	MTD	FCN	STEP	JAC	FS	GE
10^{-2}	F1	976	882	1	0	0.115425
	B1	10366	797	1	0	5.808069(-4)
10^{-4}	F1	26707	2413	1	16	4.023551(-4)
	B1	26880	2067	1	1	2.448647(-2)
10^{-6}	F1	85927	7807	0	5	9.651620(-7)
	B1	67691	5206	1	2	2.502111(-2)
10^{-8}	F1	216716	19700	0	2	1.265587(-8)
	B1	170070	13081	1	3	2.503427(-2)

TABLE IV
NUMERICAL RESULTS FOR PROBLEM 2

TOL	MTD	FCN	STEP	JAC	FSTEP	GE
10^{-2}	F1	1740	158	0	0	1.434827(-2)
	B1	2537	195	0	0	3.715717(-2)
10^{-4}	F1	4655	423	0	0	6.379218(-6)
	B1	6385	491	0	0	2.361495(-3)
10^{-6}	F1	11783	1071	0	0	3.575833(-8)
	B1	16044	1234	0	0	1.487438(-4)
10^{-8}	F1	29614	2692	0	0	3.005017(-10)
	B1	40289	3099	0	0	9.376044(-6)

TABLE V
NUMERICAL RESULTS FOR PROBLEM 3

TOL	MTD	FCN	STEP	JAC	FS	GE
10^{-2}	F1	1916	174	0	0	0.183162
	B1	2386	183	1	0	2.470289(-2)
10^{-4}	F1	4853	441	0	0	1.589209(-3)
	B1	5917	455	0	0	1.582103(-1)
10^{-6}	F1	12201	1109	0	0	1.542600(-5)
	B1	14952	1150	0	0	9.933131(-3)
10^{-8}	F1	30648	2786	0	0	1.533881(-7)
	B1	37585	2891	0	0	6.256793(-4)

TABLE VI
NUMERICAL RESULTS FOR PROBLEM 4

TOL	MTD	FCN	STEP	JAC	FS	GE
10^{-2}	F1	9018	819	1	1	6.219093(-3)
	B1	20928	3201	1	1	6.784609(-1)
10^{-4}	F1	23590	2144	0	1	4.094247(-5)
	B1	52786	8120	0	1	4.258569(-2)
10^{-6}	F1	59505	5409	0	1	3.777785(-7)
	B1	132736	20420	0	1	2.663875(-3)
10^{-8}	F1	149631	13601	0	2	3.654645(-9)
	B1	333534	51312	0	1	1.675455(-4)

V. CONCLUSION

From the tables we observed that the new embedded SDIRKN 5(4) method produced better results in terms of function evaluations, number of steps and maximal global error. Problem 1 is considered stiff for all the tolerances whereas problem 4 is only stiff for lower tolerance. The numerical results suggest that, the new method is more efficient than the existing technique whereby the problem is reduce to first order system of ODEs.

REFERENCES

[1] Sharp and Fine, Some Nystrom pairs for the general second order initial-value Problem, *Journal of Computational and Applied Mathematics* 42: 279-291.(1992).

[2] J. R. Dormand , M. E. A. El-Mikkawy and P. J. Prince, Families of Runge-Kutta Nystrom Formula, *IMA Journal of Numerical Analysis*, 7: 235-250. (1987).

[3] M. E. A. El-Mikkawy and R. El- Desouky, A new optimized non-FSAL embedded Runge-Kutta-Nystrom algorithm of orders 6 and 4 in six stages, *Applied Mathematics and Computation* 145 : 33-43, (2003).

[4] G. Papageorgiou., I. Th. Famwlis and Ch. Tsitouras, A P-stable singly diagonally implicit Runge-Kutta-Nystrom method. *Numerical Algorithm* 17: 345-353.(1998).

[5] Hairer, E. and Wanner, G. (1987), Solving Ordinary Differential Equations I, Springer-Verlag Berlin.

[6] J. R. Dormand. *Numerical Methods for Differential Equations A Computational Approach*, CRC Press London. 1996, p 57.

[7] M.M. Chawla and P. S. Rao, High-accuracy P-stable methods for $y'' = f(t, y)$, *IMA J. Numer.Anal*.5:215-220 (1985).

[8] R. C. Allen, Jr. and G. M. Wing, An invariant imbedding algorithm for the solution of inhomogeneous linear two-point boundary value problems, *J. Computer Physics* 14: 40-58.(1974).

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