Sains Malaysiana 39(6)(2010): 1049–1054

Embedded Pair of Diagonally Implicit Runge-Kutta Method for Solving Ordinary Differential Equations

(Pasangan Terbenam Kaedah Runge-Kutta Pepenjuru Tersirat untuk Menyelesaikan Persamaan Pembezaan)

FUDZIAH ISMAIL*, RAED ALI AL-KHASAWNEH, MOHAMED SULEIMAN & MALIK ABU HASSAN

ABSTRACT

Improvements over embedded diagonally implicit Runge-Kutta pair of order four in five are presented. Method of higher stage order with a zero first row and the last row of the coefficient matrix is identical to the vector output is given. The stability aspect of it is also looked into and a standard test problems are solved using the method. Numerical results are tabulated and compared with the existing method.

Keywords: Diagonally implicit; Runge-Kutta; stiff equations; stability

ABSTRAK

Penambahbaikan pasangan terbenam kaedah Runge-Kutta pepenjuru tersirat dipersembahkan. Kaedah dengan peringkat tahap yang lebih tinggi dengan baris pertama sifar dan baris terakhir matriks pekali sama dengan vektor output diberikan. Aspek kestabilannya dikaji dan beberapa masalah piawai diselesaikan menggunakan kaedah tersebut. Keputusan berangka diberikan dan dibandingkan dengan kaedah sedia ada.

Kata kunci: Kestabilan; pepenjuru tersirat; persamaan kaku; Runge-Kutta

INTRODUCTION

Many algorithms have been proposed for the numerical solution of stiff initial value problem

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

$$f : \Re \times \Re \to \Re^m. \tag{1}$$

Such algorithm is the Singly Diagonally Implicit Runge-Kutta (SDIRK) method which was introduced to overcome some of the limitations of fully implicit and explicit Runge-Kutta method. Preliminary experiments have shown that these methods are usually more efficient than the standard Singly Implicit Runge-Kutta (SIRK) method and in many cases are competitive with backward differentiation formula.

Many Runge-Kutta (RK) codes for the numerical solution of nonstiff initial value problems in ordinary differential equations (ODEs) are based on embedded pairs of explicit RK formulas. For example the code based on Dormand and Prince (1981) embedded formula of order 5 and 6 was written as 6(5) method. This idea was extended to stiff initial value problems by Norsett and Thompsen (1984), Ismail and Suleiman (1998), Butcher and Chen (2000) and Kvaerno (2004). The codes developed did very well in extensive numerical computations thus we would like to extend the idea to methods which are of higher order and higher stage order.

The family of embedded RK formulas advances the integration from (t_n,y_n) to $t_{n+1} = t_n + h$, computing at each step two approximations y_{n+1} and $\bar{y}_{n+1}$ to $y(t_{n+1})$ of orders q and p respectively, given by

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

$$\bar{y}_{n+1} = y_n + h_n \sum_{j=1}^{s} \bar{b}_j f_j,$$

where

$$f_j = f\left(t_n + c_j h_n, y_n + h_n \sum_{i=1}^{j-1} a_{ji} f_i\right)$$

$$j = 1,\ldots,s\ .$$

An embedded pair of RK formula is given by two formulas of orders p and q where $q \geq p + 1$or can be written as $q(p)$ method which share the same function evaluations. In the usual notation, the procedure advances the numerical solution with higher order approximation y_{n+1} while the lower order solution is used only to estimate the local error and to select the stepsize according to the specified tolerance. Hence embedded method is used so that the stepsize can be controlled at virtually no extra cost at all.

The objective of this research is to derive embedded diagonally implicit Runge-Kutta (DIRK) method of order four in order five which is absolutely stable and can be used to solve stiff system of ordinary differential equations.

DERIVATION OF METHOD

To construct a 5(4) pair, 17 equations for the fifth order formula and 8 equations for the fourth order have to be solved. These nonlinear equations involve *b, A, c* for the higher order and $\bar{b}$, *A*, *c* for the lower order formula, and can be found easily in the literature. such as Butcher (1987).

Here, we assume that the first row of the coefficients matrix is zero, i.e $c_1 = a_{11} = 0$ so that the number of stages to be evaluated is one less than the number of stages and since the last row of the coefficients matrix is identical with the vector output that is $a_{7j} = b_j, j = 1,..,7.$, the value of the first stage in the next step can be obtained from the last stage of the previous step or we call this property as FSAL (First Stage As Last) property and the number of stages used here is seven.

According to Butcher and Chen (2000) if the simplifying assumptions

$$\sum_j a_{ij} c_j = \frac{c_i^2}{2}, \tag{2}$$

$$\sum_j a_{ij} c_j^2 = \frac{c_i^3}{3}, \tag{3}$$

are satisfied then the stage order of the method is three. Using the above simplifying assumptions, the equations needed to be satisfied are

$$\sum_i b_i c_i^k = \frac{1}{(k+1)}, (k = 0,1,2,3,4), \tag{4}$$

$$\sum_j a_{ij} c_j = \frac{c_i^2}{2}, (i = 2,3,\ldots,7), \tag{5}$$

$$\sum_j a_{ij} c_i^2 = \frac{c_i^3}{3}, (i = 3,\ldots,7), \tag{6}$$

$$\sum_{ij} b_i a_{ij} c_j^3 = \frac{1}{20}, \tag{7}$$

$$\sum_i b_i a_{i2} = 0, \tag{8}$$

$$b_2 = 0, \tag{9}$$

$$\sum_i \bar{b}_i c_i^k = \frac{1}{(k+1)}, (k = 0,\ldots,3), \tag{10}$$

$$\bar{b}_2 = 0. \tag{11}$$

From equation (5) , for $k = 2$ and taking all the diagonal elements as γ, giving

$$\gamma c_2 = \frac{c_2^2}{2} \Rightarrow c_2 = 2\gamma.$$

Equation (6) does not hold for $k = 2$, (the method we are going to derive is almost has third-stage order since it does not satisfy (6) for $k = 2$) thus we need to have (8) and (9).

From (5) and (6) for $k = 3$, we have

$$a_{32} c_2^2 + \gamma c_3 = \frac{c_3^2}{2} \quad \text{and}$$

$$a_{32} c_2^2 + \gamma c_3^2 = \frac{c_3^3}{3},$$

Solving the two equations gives,

$$c_3 = 3\gamma + \gamma\sqrt{3} \quad \text{and}$$

$$a_{32} = \frac{3\gamma + 2\sqrt{3\gamma}}{2},$$

$c_7 = 1$ because $a_{7j} = b_j$,

There are 19 equations to be satisfied with 23 unknowns, we have 4 free parameters, setting

$\gamma = 0.28589$, $c_4 = 0.4$, $c_5 = 0.75$, $c_6 = 0.9$,

$c_3 = 3\gamma + \gamma\sqrt{3}$ and $\dfrac{3\gamma + 2\sqrt{3\gamma}}{2}$.

TABLE 1.

0	0						
2γ	γ	γ					
$3\gamma+\gamma\sqrt{3}$	a_{31}	0.924011005	γ				
0.4	a_{41}	-0.049416510	-0.004509476	γ			
0.75	a_{51}	-0.112951603	-0.027793233	0.422539833	γ		
0.9	a_{61}	-0.425378071	-0.107036282	0.395700134	0.503260302	γ	
1	a_{71}	0	-0.019290177	0.535386266	0.234313169	-0.166317293	γ
	a_{71}	0	-0.019290177	0.535386266	0.234313169	-0.166317293	γ
	-0.094388662	0	-0.039782614	0.745608552	-0.505129807	0.704915206	γ

Solving the set of equations using NAG Library Routine we have the method given in Table 1.

The values of a_{i1} are obtained from the row condition $c_i = \sum_{j=1}^{i} a_{ij}$.

Stability of the Method

The stability polynomial is obtained when the method is applied to the linear test equation

$$y' = f(x, y) = \lambda y, \tag{12}$$

where

$$\begin{aligned} k_i &= f\left(x_n + c_i h, y_n + h\sum_{j=1}^{i} a_{ij}k_j\right) \\ &= \lambda\left(y_n + h\sum a_{ij}k_j\right) \\ &= \lambda y_n + h\lambda\sum a_{ij}k_j, \end{aligned}$$

and for diagonally implicit method

$$\begin{aligned} k_1 &= \lambda y_n + h\lambda a_{11}k_1 \\ k_2 &= \lambda y_n + h\lambda\left(a_{21}k_1 + a_{22}k_2\right) \\ &\vdots \\ k_i &= \lambda y_n + h\lambda\left(a_{i1}k_1 + \ldots + a_{ii}k_i\right), \end{aligned}$$

or

$$\begin{aligned} K_j &= \lambda Y_n + \bar{h}AK_j \\ \left(I - \bar{h}A\right)K_j &= \lambda Y_n \\ K_j &= \left(I - \bar{h}A\right)^{-1}\lambda Y_n, \end{aligned}$$

and

$$\begin{aligned} y_{n+1} &= y_n + h\sum b_j k_j \\ &= y_n + hb^t\left(t - \bar{h}A\right)^{-1}\lambda y_n \\ &= \left(1 + \bar{h}b^T\left(I - \bar{h}A\right)^{-1}\right)y_n, \end{aligned}$$

Thus $y_{n+1} = R(\bar{h})y_n$, where

$$R\left(\bar{h}\right) = 1 + \bar{h}b^T\left(I - \bar{h}A\right)^{-1}e$$

is called the stability polynomial of the method.

For diagonally implicit method, $R\left(\bar{h}\right)$ becomes a rational function $R\left(\bar{h}\right) = \dfrac{P\left(\bar{h}\right)}{Q\left(\bar{h}\right)}$, where P for our method is a polynomial of degree seven and $Q\left(\bar{h}\right) = \left(I - \bar{h}\right)^6$. If the method is of order p, then $e^{\bar{h}} - R\left(\bar{h}\right) - C\bar{h}^{p+1} + O\left(\bar{h}^{p+2}\right)$ (see Hairer & Wanner 1991). In other words $R\left(\bar{h}\right)$ is a rational approximation to $e^{\bar{h}}$ of order p.

Here

$$\begin{aligned} P\left(\bar{h}\right) = 1 &+ d_1\bar{h} + d_2\bar{h}^2 + d_3\bar{h}^3 + d_4\bar{h}^4 \\ &+ d_5\bar{h}^5 + d_6\bar{h}^6 + d_7\bar{h}^7. \end{aligned} \tag{13}$$

and $Q\left(\bar{h}\right) = \left(1 - \bar{h}\gamma\right)^6$, thus we have

$$\begin{aligned} P\left(\bar{h}\right) = \left(1 - \bar{h}\gamma\right)^6 \Bigg(1 + \bar{h} + \frac{\bar{h}^2}{2} + \frac{\bar{h}^{-3}}{6} + \frac{\bar{h}^4}{24} + \frac{\bar{h}^5}{120} + \\ O\left(\bar{h}^6\right)\Bigg) \end{aligned} \tag{14}$$

Using (13) and (14), and equating the left hand side and right hand side and collecting terms of equal powers of $\bar{h}$, the values of d_i (1(1)7) can be written in terms of γ as follows

$$\begin{aligned} d_1 &= 1 - 6\gamma, \\ d_2 &= \frac{1}{2} - 6\gamma + 15\gamma^2, \\ d_3 &= \frac{1}{6} - 3\gamma + 15\gamma^2 - 20\gamma^3, \\ d_4 &= \frac{1}{24} - \gamma + \frac{15}{2}\gamma^2 - 20\gamma^3 + 15\gamma^4, \\ d_5 &= \frac{1}{120} - \frac{1}{4}\gamma + \frac{5}{2}\gamma^2 - 10\gamma^3 + 15\gamma^4 - 6\gamma^5, \\ d_6 &= T_1 + \frac{\gamma}{20} - \frac{5}{8}\gamma^2 - \frac{10}{3}\gamma^3 + \frac{15}{2}\gamma^4 - 6\gamma^5 + \gamma^6, \\ d_7 &= T_2 + \frac{1}{8}\gamma^2 - \frac{5}{6}\gamma^3 + \frac{5}{2}\gamma^4 - 3\gamma^5 + \gamma^6, \end{aligned}$$

where $T_1 = \sum b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m$,

$T_2 = \sum b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m$, and , $T_1 \neq \dfrac{1}{720}$,

$T_2 \neq \dfrac{1}{5040}$, where

$\sum b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m, = \dfrac{1}{720}$

is one of the order conditions for the sixth order method and

$$\sum b_i a_{ij} a_{jk} a_{kl} a_{lm} a_{mn} c_m = \frac{1}{5040}$$

is one of the order conditions for the seventh order methods. Therefore, T_1 and T_2 can be calculated using coefficients of the SDIRK (5,7) method itself.

The stability region is the region enclosed by the set of points for which $R(\bar{h}) = 1$. Replacing 1 by $\cos\theta + i\sin\theta$, we can trace out this boundary by solving the equation for values of $\theta \in [0, 2\pi]$

$$R\left(\bar{h}\right) = \frac{P\left(\bar{h}\right)}{Q\left(\bar{h}\right)} = \cos\theta + i\sin\theta \quad \text{or}$$

$$P\left(\bar{h}\right) = \left(1 - \bar{h}\gamma\right)^6\left(\cos\theta + i\sin\theta\right),$$

Letting

$$F_1\left(\bar{h}\right) = P\left(\bar{h}\right) - \left(1 - \bar{h}\gamma\right)^6\left(\cos\theta + i\sin\theta\right) = 0,$$

and expanding the polynomial we have

$$\begin{aligned} &F_1\left(\bar{h}\right) = \\ &\left(-1 + \cos\theta + i\sin\theta\right) + \\ &\bar{h}\left(-6\gamma\cos\theta + 6\gamma - 1 - 6\gamma i\sin\theta\right) + \end{aligned}$$

$$\bar{h}^2\left(15\gamma^2\cos\theta-15\gamma^2+6\gamma-\frac{1}{2}+15\gamma^2 i\sin\theta\right)+$$
$$\bar{h}^3\left(-20\gamma^3\cos\theta+20\gamma^3-15\gamma^2+3\gamma-\frac{1}{6}-20\gamma^3 i\sin\theta\right)+$$
$$\bar{h}^4\left(15\gamma^4\cos\theta-15\gamma^4+20\gamma^3-\frac{15}{2}\gamma^2+\gamma-\frac{1}{24}+15\gamma^4 i\sin\theta\right)+$$
$$\bar{h}^5\left(-6\gamma^5\cos\theta+6\gamma^5-15\gamma^4+10\gamma^3-\frac{5}{2}\gamma^2+\frac{1}{4}\gamma-\frac{1}{120}-6\gamma^5 i\sin\theta\right)+$$
$$\bar{h}^6\left(\gamma^6\cos\theta-\gamma^6+6\gamma^5-\frac{15}{2}\gamma^4+\frac{10}{3}\gamma^3-\frac{5}{8}\gamma^2+\frac{1}{20}\gamma-T_1+\gamma^6 i\sin\theta\right)+$$
$$\bar{h}^7\left(-\gamma^6+3\gamma^5-\frac{5}{2}\gamma^4+\frac{5}{6}\gamma^3-\frac{1}{8}\gamma^2-T_2\right)=0.$$

Solve for $\bar{h}$ with γ, T_1 and T_2 depend on the coefficients of the method itself gives the stability region of the method, which is given in Figure1. Stability region of the method with T_1=2.094430752396×10^{-3} and T_2=2.09443075237×10^{-3} lies inside the close region of Figure 1.

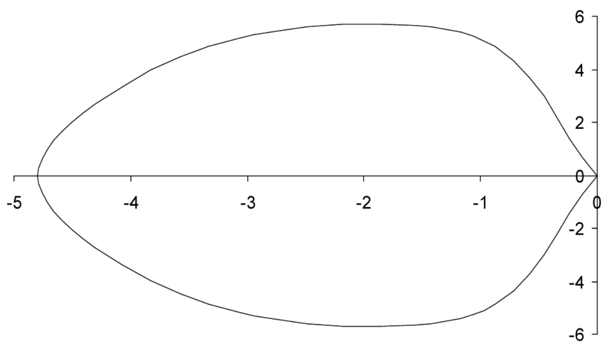

FIGURE 1. The stability region of the new method

IMPLEMENTATION

In this section, we briefly summarized the implementation of the method derived in the previous section on stiff systems of ODEs. The method is an implicit method, thus iterations are needed to obtain the numerical solutions. Initially the system is considered as nonstiff and simple iterations are used, once there is an indication of stiffness, the whole system is considered as stiff and Newton iterations are used. Here, two iterations are done and the convergence test for the simple iteration is

$$h\left(\frac{\rho^2}{1-\rho}\right)|b_i|\left\|\Delta^{(1)}k_i\right\|<0.2\ tol,$$

where tol is the tolerance chosen, convergence test for the Newton iteration is

$$h|b_i|\left\|\Delta^{(1)}k_i\right\|\left[\frac{\left\|\Delta^{(1)}k_i\right\|}{\left\|\Delta^{(0)}k_i\right\|}\right]<0.1\ tol.$$

$\Delta^{(m)}\ k$, is the difference between the (m+1)th and (m)th iteration of k_i and ρ is $\left|\frac{\partial f}{\partial y}\right|$.

h_{start} start is given by $h=\frac{tol}{2}$ and the subsequent stepsize is given by h = min {h_{acc}, h_{iter}}

where $h_{acc}=0.5\left[\frac{tol}{2LTE}\right]^{\frac{1}{p+1}}h$ and

$$h_{iter}=\frac{h}{10\rho}.$$

h_{acc} and h_{iter} are the values of h for which the solution is expected to satisfy the chosen tolerance and for which the iteration will converge respectively and in the case of failed step halve the stepsize and redo the process again. The indicator for stiffness here is when $h_{acc} > h_{iter}$.

NUMERICAL RESULTS AND CONCLUSION

In this section, some of the problems obtained from Enright et al. (1974) are tested upon. The numerical results are compared with the results obtained when the same set of problems are solved using 5(4) method developed by Kvaerno (2004). All methods share the same characteristics namely a zero first row and the last row in the coefficients matrix is identical with the output vector. The results are tabulated in Tables 2 to 5, and the notations used are as follows:

TOL ~ Tolerance used.

METHOD ~ N1 ~ The new 5(4) DIRK method
A1 ~ Kaervo's 5 (4) DIRK method.
FCN ~ the number of functions evaluated
STEP ~ The number of steps needed for the integration
JACO ~ The number of Jacobian evaluated
FS ~ The number of failed steps.

Problems tested are:

Problem 1.

$$y_1'=-y_1+y_2^2+y_3^2+y_4^2$$
$$y_2'=-10y_2+10\left(y_3^2+y_4^2\right)$$
$$y_3'=-40y_3+40y_4^2$$
$$y_4'=-100y_4+2$$
$$y_i(0)=1,(i=1(1)4)$$
$$0\le x\le 20$$

Problem 2

$$y_1'=-1800y_1+900y_2$$
$$y_1(0)=y_2(0)=1$$

$$y_i' = y_{i-1} - 2y_i + y_{i+1}$$
$$y_9' = -1000y_8 - 2000y_9 + 1000$$
$$y_1(0) = y_2(0) = 1$$
$$y_i(0) = 1, (i = 2(1)8)$$
$$0 \le x \le 20.$$

Problem 4.3.

$$y_1' = -10^4 y_1 + 100y_2 - 10y_3 + y_4$$
$$y_2' = -10^3 y_2 + 10y_3 - 10y_4$$
$$y_3' = -y_3 + y_4$$
$$y_i(0) = 1, (i = 1(1)4)$$
$$y_4' = -0.1y_4$$
$$0 \le x \le 20.$$

Problem 4.4.

$$y_1' = -(55 + y_3)y_1 + 65y_2$$
$$y_2' = 0.0785(y_1 - y_2)$$
$$y_3' = -0.1y_1$$
$$y_1(0) = 1,\ \ y_2(0) = 1,\ \ y_3(0) = 0$$

TABLE 2. Numerical results for problem 4.1, using tolerances 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}

TOL	METHOD	FCN	STEP	JACO	FS
10^{-2}	N1	256	17	1	1
	A1	338	23	1	1
10^{-4}	N1	702	49	1	2
	A1	920	65	1	2
10^{-6}	N1	2520	161	1	3
	A1	19523	634	1	4
10^{-8}	N1	4639	561	1	4
	A1	24892	5452	2	5

TABLE 3. Numerical results for problem 4.2, using tolerances 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}

TOL	METHOD	FCN	STEP	JACO	FS
10^{-2}	N1	314	24	1	2
	A1	441	29	1	2
10^{-4}	N1	605	41	1	2
	A1	868	58	1	3
10^{-6}	N1	1211	89	1	2
	A1	24534	356	1	4
10^{-8}	N1	4061	350	2	4
	A1	74835	6953	2	5

TABLE 4. Numerical results for problem 4.3, using tolerances 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}

TOL	METHOD	FCN	STEP	JACO	FS
10^{-2}	N1	331	22	1	1
	A1	428	29	1	1
10^{-4}	N1	929	64	1	1
	A1	1311	94	1	2
10^{-6}	N1	2437	170	1	3
	A1	15899	1758	1	3
10^{-8}	N1	9178	672	1	3
	A1	48589	4204	2	4

TABLE 5. Numerical results for problem 4.4, using tolerances 10^{-2}, 10^{-4}, 10^{-6}, 10^{-8}

TOL	METHOD	FCN	STEP	JACO	FS
10^{-2}	N1	309	14	1	1
	A1	321	19	1	1
10^{-4}	N1	369	32	1	2
	A1	386	33	1	2
10^{-6}	N1	2707	193	1	3
	A1	5489	965	2	3
10^{-8}	N1	5296	480	1	3
	A1	10463	3225	1	4

From the tables it was observed that, for all the tolerances and for all the problems method N1 is more efficient compared to A1 in terms of number of steps and number of function evaluations. The reason is that though N1 is of the same order as A1, stage order for N1 is almost 3 while for A1 is 2. As a conclusion it can be said that for stiff problems method N1 is more efficient compared to A1.

REFERENCES

Butcher J.C. 1987. *The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General Linear Methods,* New York: John Wiley and Sons.

Butcher, J.C. & Chen, D.J.L. 2000. A new Type of Singly-implicit Runge-Kutta method, *Applied Numerical Mathematics* 34: 179-188.

Dormand J.R. & Prince P.J. 1981. High order embedded Runge-Kutta formula. J. *Comput. Appl. Math* 7: 67-75.

Enright. W.H., Hull, T.E. & Lindberg B, 1974. Comparing Numerical Methods for stiff Systems of ODEs, *Technical Report No 69*, Department of Computer Science, University of Toronto, Canada.

Ismail, F. & Suleiman M.B. 1998. Embedded Singly Diagonally Implicit Runge-Kutta method (4,5) in (5,6) for the integration of stiff systems of ODEs, *International Journal of Computer Math* 66: 325-341.

Kvaerno A, 2004. Singly Diagonally Implicit Runge-Kutta Methods with an explicit first stage, *BIT* 44: 489-502.
Norsett, S.P. & Thompsen, P.G. 1984. Embedded SDIRK methods of basic order three, *BIT* 24: 634-464.

Jabatan Matematik
Fakulti Sains
Universiti Putra Malaysia
43400 UPM Serdang
Selangor, Malaysia

*Corresponding author; email: fudziah@math.upm.edu.my

Received: 9 April 2009
Accepted: 26 June 2009