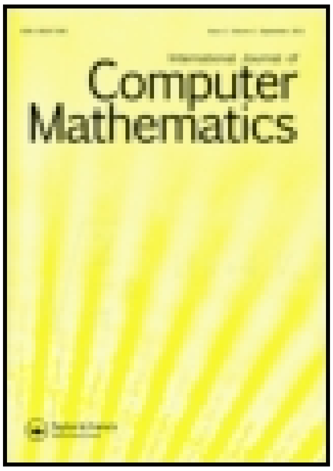

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

Embedded singly diagonally implicit runge-kutta methods (4,5) in (5,6). for the integration of stiff systems of odes

Fudziah Bt Ismail [a] & Mohamed B. Suleman [a]

[a] Department of Mathematics , Universiti Pertanian Malaysia , Serdang Selangor, UPM, 43400, Malaysia

Published online: 19 Mar 2007.

To cite this article: Fudziah Bt Ismail & Mohamed B. Suleman (1998) Embedded singly diagonally implicit runge-kutta methods (4,5) in (5,6). for the integration of stiff systems of odes, International Journal of Computer Mathematics, 66:3-4, 325-341, DOI: 10.1080/00207169808804644

To link to this article: http://dx.doi.org/10.1080/00207169808804644

Intern. J. Computer Math., Vol. 66, pp. 325–341
Reprints available directly from the publisher
Photocopying permitted by license only

EMBEDDED SINGLY DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS (4,5) IN (5,6). FOR THE INTEGRATION OF STIFF SYSTEMS OF ODEs

FUDZIAH BT ISMAIL and MOHAMED B. SULEIMAN

Department of Mathematics, Universiti Pertanian Malaysia, 43400 UPM Serdang Selangor, Malaysia

(Received 15 December 1996)

In this paper, fourth order, five-stage embedded in fifth order six-stage Singly Diagonally Implicit Runge-Kutta (SDIRK) methods are derived, and their regions of stability are determined. Numerical results of these methods are presented and compared with the results obtained when the same systems are solved using Rabeh's third order three-stage embedded in fourth order four-stage SDIRK method. Also presented are the stability regions of the methods.

Keywords: Singly diagonally implicit Runge-Kutta methods; embedded Runge-Kutta method; stiff ordinary differential equations

C. R. Categories: G.1.7

1. INTRODUCTION

The problem of integrating stiff systems of first order ordinary differential equations (ODEs) of the form

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

numerically using diagonally implicit Runge-Kutta method has received a considerable amount of attention. This is due to the fact that computational effort involved in using diagonally implicit runge-Kutta method (DIRK) is generally less than which is required by fully implicit Runge-Kutta method.

Such formula can be written as

$$^{(m)}k_i = f\left(x_n + c_i h,\ y_n + h\sum_{j=1}^{i} a_{ij}^{(m)} k_j\right), \quad (i = 1(1)q) \tag{1.2a}$$

$$\bar{y}_{n+1} = y_n + h\sum_{i}^{q} b_i^{(m)} k_i, \tag{1.2b}$$

If the diagonal element of the method are equal, $a_{ii} = \gamma$, then this class of methods are called Singly Diagonally Implicit Runge-Kutta method (SDIRK).

Suppose (1.2a) is solved successively using Newton-type iterations, where linear system with a coefficient matrix of the form

$$I - ha_{ii}\,\partial f/\partial y$$

is solved at each stage. Further if all a_{ii} are equal, then the stored LU-factorization of the matrix may be used repeatedly, thus the method is computationally more efficient. Such formulae were first suggested by Norsett [1] and further studied by Crouzeix [2], Alexander [3], Cash [4], Rabeh [5] also Cooper and Safy [6].

Embedded SDIRK formulae have a built in local truncation error estimate, as a result, the stepsize can be controlled at virtually no extra cost. The idea of embedding formulae to obtain local error estimate was first proposed by Fehlberg [7], using explicit Runge-Kutta method and Norsett and Thomsen [8] continued the work using semi-implicit Runge-Kutta method, whereby they developed SDIRK method of order two embedded in the method of order three. Currently the highest order embedded SDIRK method is third order embedded in fourth order, which was derived by Rabeh [5].

In this paper we derive SDIRK formulae of order four embedded in the method of order five.

Define the embedding method to (1.2) as

$$\left.\begin{aligned} {}^{(m)}k_i &= f\left(x_n + c_i h,\ y_n + h\sum_{j=1} a_{ij}\ {}^{(m)}k_j\right),\ i = 1(1)q+1. \\ y_{n+1} &= y_n + h\sum_{i=1}^{q+1} b_i\ {}^{(m)}k_i \end{aligned}\right\} \tag{1.3}$$

Then the local truncation error could be estimated using (1.3) and (1.2), given by

$$\left.\begin{aligned} \text{LTE} &= y_{n+1} - \bar{y}_{n+1} \\ &= h \sum_{i=1}^{q+1} (b_i - \bar{b}_i)\ {}^{(m)}k_i,\ \bar{b}_{q+1} = 0 \end{aligned}\right\} \tag{1.4}$$

The embedding method has $q+1$ stage and the embedded method has q stage only. Suppose the method (1.2) has order p and q-stage and the method (1.3) has order $p+1$ and $q+1$ stage. We will refer to the pair defined by (1.2) and (1.3) as SDIRK method (p, q) embedded in $(p+1, q+1)$.

Thus the pair we will derive in section 2 will be of the form $(4, 5)$ embedded in $(5, 6)$. Stability regions associated with the methods are presented in section 3, followed by numerical results in section 5.

2. DERIVATION OF EMBEDDED SDIRK METHOD

The coefficients of the matched pair of SDIRK methods $(4, 5)$ embedded in $(5, 6)$. are given in Table 2.1.

Note that $c_i = \sum_{j=1}^{i} a_{ij}$

$$\bar{b}_i - \text{for } (4,5) \text{ method}$$
$$b_i - \text{for } (5,6) \text{ method}$$

Here we give a few notations that are commonly used,

t - tree associated with elementary differentials.
$\Phi(t)$ - the elementary weights of the rooted tree t.
$\bar{\gamma}(t)$ - the density of the tree.

(here, $\bar{\gamma}(t)$ is different from γ in Tab. (2.1)).

TABLE 2.1

γ	γ					
c_2	a_{21}	γ				
c_3	a_{31}	a_{32}	γ			
c_4	a_{41}	a_{42}	a_{43}	γ		
c_5	a_{51}	a_{52}	a_{53}	a_{54}	γ	
c_6	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	γ
	b_1	b_2	b_3	b_4	b_5	b_6
	$\bar{b}_1$	$\bar{b}_2$	$\bar{b}_3$	$\bar{b}_4$	$\bar{b}_5$	

Butcher's [9] equations of order conditions are given in Table (2.2) for order 1 to 5, where the order conditions are given by

$$\Phi(t) = 1/\bar{\gamma}(t).$$

For the (4, 5) method, the first eight equations have to be satisfied and for the (5, 6) method we have to satisfy all the 17 equations.

Using Simplifying Equations

First consider the equations needed to be satisfied for the (5, 6) method. Equations (2.13) and (2.10) can be written as

$$\sum_i b_i\left(\sum a_{ij}c_j\right)^2 = 1/20$$

and

$$\sum_i b_i\left(\sum_j a_{ij}c_j\right)c_i^2 = 1/10$$

respectively.

Multiply (2.9) by 1/4, add to (2.13) and minus (2.10) will give us

$$(1/4)\sum_i b_i c_i^4 + \sum_i b_i\left(\sum_j a_{ij}c_j\right)^2 - \sum_i b_i\left(\sum_j a_{ij}c_j\right)c_i^2 = 0$$

$$\sum_i b_i\left(\sum_j a_{ij}c_j - (1/2)c_i^2\right)^2 = 0$$

Thus

$$\sum_j a_{ij}c_j = c_i^2/2 \quad (i = 2(1)6) \tag{2.18a}$$

(2.18a) does not hold for $i=1$, because we do not want $c_1 = a_{11} = 0$.

In this case we need

$$b_1 = 0 \tag{2.18b}$$

Butcher [9] refer to this pair as a row-simplifying assumption.

If the pair (2.18) is satisfied we can remove equations (2.13) and (2.10) provided (2.9) holds. Certain other pair of order conditions also have the

TABLE 2.2

graph of t	$\Phi(t) = 1/\bar{\gamma}(t)$	
	$\sum b_i = 1$	(2.1)
	$\sum b_i c_i = 1/2$	(2.2)
	$\sum b_i c_i^2 = 1/3$	(2.3)
	$\sum b_i a_{ij} c_j = 1/6$	(2.4)
	$\sum b_i c_i^3 = 1/4$	(2.5)
	$\sum b_i c_i a_{ij} c_j = 1/8$	(2.6)
	$\sum b_i a_{ij} c_j^2 = 1/12$	(2.7)
	$\sum b_i a_{ij} a_{jk} c_k = 1/24$	(2.8)
	$\sum b_i c_i^3 = 1/5$	(2.9)
	$\sum b_i c_i^2 a_{ij} c_j = 1/10$	(2.10)
	$\sum b_i c_i a_{ij} c_j^2 = 1/15$	(2.11)
	$\sum b_i c_i a_{ij} a_{jk} c_k = 1/30$	(2.12)
	$\sum b_i a_{ij} c_j a_{ik} c_k = 1/20$	(2.13)
	$\sum b_i a_{ij} c_j^3 = 1/20$	(2.14)
	$\sum b_i a_{ij} c_j a_{jk} c_k = 1/40$	(2.15)
	$\sum b_i a_{ij} a_{jk} c_k^2 = 1/60$	(2.16)
	$\sum b_i a_{ij} a_{jk} a_{kl} c_l = 1/120$	(2.17)

same type of relationship, they are (2.6) and (2.5), so we can remove (2.6). For equations (2.12) and (2.11), we can remove (2.12), if

$$\sum_i b_i c_i a_{i1} = 0 \tag{2.19}$$

Equations (2.4) can be written as equation (2.2) minus equation (2.3) giving

$$\sum_i b_i a_{ij} c_j = \sum_j b_j c_j - \sum_j b_j c_j^2,$$

or

$$\sum_i a_{ij} c_j = b_j(1 - c_j) \quad (j = 1(1)6), \tag{2.20}$$

which is referred to as column simplifying assumption. It had the effect of removing all the one-leg tree (tree that has only a single arc branching from the root) other than the one associated with equation (2.2). If (2.20) holds we can remove equations (2.4), (2.7), (2.8), (2.14), (2.15), (2.16) and (2.17).

Thus for the (5, 6) method, the equations needed to be satisfied are (2.1), (2.2), (2.3), (2.5), (2.9), (2.11), (2.18), (2.19) and (2.20). Denote the first eight equations associated with the (4, 5) embedded method by $(\hat{2.1}), \ldots, (\hat{2.8})$ and $\bar{b}_i$ are replaced by $\bar{b}_i$ $(i = 1(1)5)$. If (2.18a) holds then equations $(\hat{2.6})$ is equivalent to $(\hat{2.5})$ and $(\hat{2.4})$ is equivalent to $(\hat{2.3})$ if

$$\bar{b}_1 = 0 \tag{2.21}$$

Further equation $(\hat{2.8})$ and $(\hat{2.7})$ are equivalent if

$$\sum_i \bar{b}_i a_{i1} = 0 \tag{2.22}$$

Therefore equations needed to be satisfied for the (4, 5) method are $(\hat{2.1})$, $(\hat{2.2})$, $(\hat{2.3})$, $(\hat{2.5})$, $(\hat{2.7})$, $(\hat{2.21})$ and $(\hat{2.22})$.

From (2.18a) for $i = 2$, we have

$$a_{21} c_1 + a_{22} c_2 = c_2^2/2 \tag{2.23}$$

Substitute

$$c_1 = a_{11} = a_{22} = \gamma \text{ and } a_{21} = c_2 - \gamma, \text{ into (2.23) giving}$$
$$2\gamma(c_2 - \gamma) + 2\gamma c_2 = c_2^2$$

or

$$c_2 = 2\gamma \pm \gamma\sqrt{2} \tag{2.24}$$

From equation (2.20) for $j=6$ we have

$$b_6\, a_{66} = b_6(1 - c_6)$$
$$c_6 = 1 - \gamma.$$

After substituting b_1, $\bar{b}_1$,c_2 and c_6 into the system of equations we are left with equations (2.1), (2.2), (2.3), (2.5), (2.9), (2.11), [(2.18a) ($i=3(1)6$)], (2.19), [(2.20)($j=2(1)5$)], $(\hat{2.1})$, $(\hat{2.2})$, $(\hat{2.3})$, $(\hat{2.5})$ and $(\hat{2.7})$, since (2.22) can be substituted into $(\hat{2.7})$.

All together there are 21 equations to be solved with 23 unknowns. There are two free parameters which are chosen to be γ and c_3 and from (2.24) take $c_2 = 2\gamma + \gamma\sqrt{2}$. The system is solved using NAG routine.

In Tables (2.3–2.5) we present three SDIRK (4, 5) embedded in (5, 6). method using three values of γ.
For the above method we take $\gamma = 0.27805384$ and $c_3 = -0.7$.
And we call it as method F1(A).

TABLE 2.3

γ	γ					
$2\gamma+\gamma\sqrt{2}$	a_{21}	γ				
-0.7	a_{31}	1.06004630	γ			
0.25653741	a_{41}	−0.03682839	0.00789379	γ		
0.82839417	a_{51}	−0.03380531	0.00608165	0.53925854	γ	
$1-\gamma$	a_{61}	−0.63238912	0.04238439	0.55765932	0.75299105	γ
	0.0	−0.06318768	0.00318654	0.56747397	0.57361772	−0.0810905
	0.0	−0.02185092	0.00356539	0.5601091	0.45817642	

TABLE 2.4

γ	γ					
$2\gamma+\gamma\sqrt{2}$	a_{21}	γ				
-0.7	a_{31}	1.07091464	γ			
0.25109432	a_{41}	−0.03707993	0.00766549	γ		
0.56536452	a_{51}	−0.07272339	0.00883329	0.97869993	γ	
$1-\gamma$	a_{61}	−0.02745363	0.00216163	0.91306419	−0.05734425	γ
	0.0	−0.11018458	0.00277906	0.57280038	−0.16289019	0.4771267
	0.0	0.26135316	0.006310993	0.46956348	0.26277236	

TABLE 2.5

γ	γ					
$2\gamma+\gamma\sqrt{2}$	a_{21}	γ				
0.3	a_{31}	−0.07041631	γ			
−0.05789024	a_{41}	0.27469871	−1.45349159	γ		
0.49406019	a_{51}	−0.12085631	2.2243715	0.32725201	γ	
$1-\gamma$	a_{61}	−0.14595303	1.67648205	0.40578896	0.4013551	γ
	0.0	0.32448436	0.45800566	0.06676735	0.09204594	0.058696699
	0.0	0.35459613	0.42531509	0.07065555	0.14943322	

For the method in Table (2.4) $\gamma=4/15$ and $c_3=-0.7$, and we call it method F1(B).

For the method in Table (2.5) we take $\gamma=0.25$ and $c_3=0.3$, and we call it method F1(C).

For all the three methods the values of a_{i1} are given by

$$a_{i1} = c_i - \sum_{j=2}^{i} a_{ij} \quad (i = 1(1)6).$$

3. STABILITY OF THE METHODS

The method (1.3) applied to $y' = \lambda y$ yields

$$y_1 = R(h\lambda)y_0$$

with

$$R(\bar{h}) = 1 + \bar{h}b^T(I - \bar{h}A)^{-1}\mathbf{1} \tag{3.1}$$

where $\bar{h} = h\lambda$, $b^T = (b_1,\ldots,b_q)$

$$A = (a_{ij})_{i,j=1}^{q}$$
$$\mathbf{1} = (1,\ldots,1)^T$$

$R(\bar{h})$ is called the stability polynomial of the method. The stability polynomial of (3.1) can be written as

$$R(\bar{h}) = [\det(I - \bar{h}A + \bar{h}\mathbf{1}b^T)]/[\det(I - \bar{h}A)],$$

see Hairer and Wanner [10] for the proof.

For the six-stage SDIRK method $R(\bar{h})$ can be written as

$$R(\bar{h}) = P(\bar{h})/Q(\bar{h}) \tag{3.2}$$

For our method P and Q are polynomials of degree 6, since the method is of order 5, further

$$R(\bar{h}) = P(\bar{h})/Q(\bar{h}) = e^{\bar{h}} + O(\bar{h}^6) \tag{3.3}$$

see Butcher [9], where

$$P(\bar{h}) = 1 + d_1\bar{h} + d_2\bar{h}^2 + d_3\bar{h}^3 + d_4\bar{h}^4 + d_5\bar{h}^5 + d_6\bar{h}^6$$
$$Q(\bar{h}) = (1 - \bar{h}\gamma)^6,$$

Using equation (3.3), equating the left hand side and the right hand side and collecting terms of equal power of $\bar{h}$ we obtained the values of d_i $(i = 1(1)6)$ in terms of γ.

$$\begin{aligned}
d_1 &= (1 - 6\gamma)\\
d_2 &= (1/2 - 6\gamma + 15\gamma^2)\\
d_3 &= (1/6 - 3\gamma + 15\gamma^2 - 20\gamma^3)\\
d_4 &= (1/24 - \gamma + (15/2)\gamma^2 - 20\gamma^3 + 15\gamma^4)\\
d_5 &= (1/120 - (1/4)\gamma + (5/2)\gamma^2 - 10\gamma^3 + 15\gamma^4 - 6\gamma^5)
\end{aligned}$$

Notice that the terms that do not involve γ are equivalent to $1/\bar{\gamma}(t)$ [see Table (2.2)], where t are the trees which has no branches.

Thus

$$d_6 = (S - \gamma/20 + 5\gamma^2/8 - 10\gamma^3/3 + 15\gamma^4/2 - 6\gamma^5 - \gamma^6)$$

where

$$S = \sum_{ij} b_i a_{ij} a_{jk} a_{kl} a_{lm} c_m$$

and $S \neq 1/720$ because the method is not a method of order six, so it does not satisfy the conditions for the sixth-order method. In this case S can be calculated using the coefficients of the SDIRK (5, 6) method itself.

The stability region can be determined by the maximum modulus principle, that is the region enclosed by the set of points $\bar{h}$, for which, modulus of $R(\bar{h}) = 1$. Letting $R(\bar{h}) = \cos\theta + i \sin\theta$, the boundary is traced by solving the stability equations for values of $\theta \in [0,2\pi]$.

Here we only find the stability region of the SDIRK (5, 6) method, because the equations are solved using the method and the SDIRK (4, 5) method are used only to find local truncation error.

The stability regions for the three SDIRK method derived in section 2 are given in the following figures.

The stability region for SDIRK (5,6) method in F1(A) $S = (1.7339739) \times 10^{-3}$) lies outside the close region in figure (3.1)

The stability region for SDIRK (5,6) method in F1(B) $S = (1.65140337) \times 10^{-3}$) lies outside the close region in figure (3.2)

The stability region for SDIRK (5,6) method in F1(C) $S = (2.10384458) \times 10^{-3}$) lies inside the close region in figure (3.3).

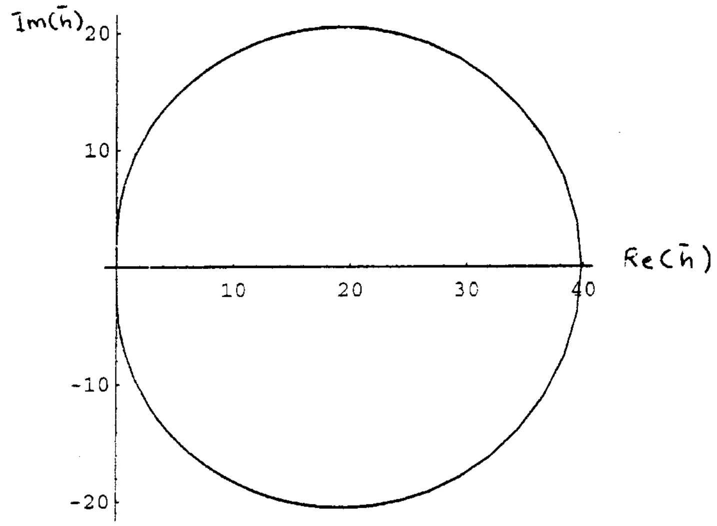

FIGURE 3.1

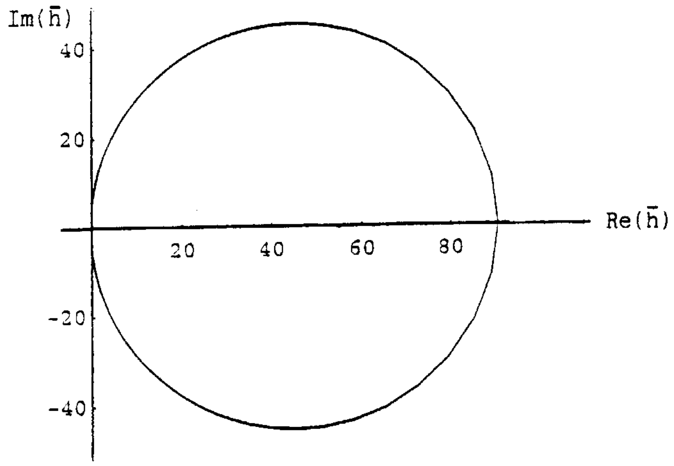

FIGURE 3.2

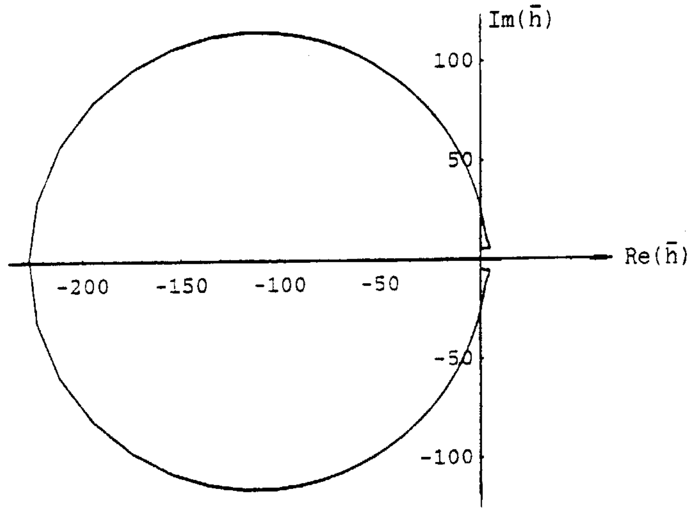

FIGURE 3.3

4. PROBLEM TESTED

The following are some of the problems tested.

Problem 1

$$\begin{aligned} y_1' &= -(55+y_3)y_1 + 65y_2 && y_1(0) = 1 \\ y_2' &= 0.0785(y_1 - y_2) && y_2(0) = 1 \\ y_3 &= 0.1y_1 && y_3(0) = 0 \\ & && 0 \le x \le 20 \end{aligned}$$

Problem 2

$$\begin{aligned} y_1' &= -10y_1 + 3y_2 \\ y_2' &= -3y_1 - 10y_2 \\ y_3' &= -4y_3 \\ y_4' &= -y_4 \\ y_5' &= -0.5y_5 && y_i(0) = 1,\ (i = 1(1)6) \\ y_6' &= -0.1y_6 && 0 \le x \le 20 \end{aligned}$$

Problem 3

$$\begin{aligned} y_1' &= -y_1 + y_2^2 + y_3^2 + y_4^2 \\ y_2' &= -10y_2 + 10(y_3^2 + y_4^2) \\ y_3' &= -40y_3 + 40y_4^2 && y_i(0) = 1,\ (i = 1(1)4) \\ y_4' &= -100y_4 + 2 && 0 \le x \le 20 \end{aligned}$$

Problem 4

$$\begin{aligned} y_1' &= -1800y_1 + 900y_2 && i = 2(1)8 \\ y_i' &= y_{i-1} - 2y_i + y_{i+1}, && y_i(0) = 0,\ (i = 1(1)9) \\ y_9' &= 1000y_8 - 2000y_9 + 1000 && 0 \le x \le 20. \end{aligned}$$

Problem 5

$$\begin{aligned} y_1' &= y_3 - 100y_1y_2 \\ y_2' &= y_3 + 2y_4 - 100y_1y_2 - 20000y_2^2 && y_1(0) = y_2(0) = 1 \\ y_3' &= -y_3 + 100y_1y_2 && y_3(0) = y_4(0) = 0 \\ y_4' &= -y_4 + 10000y_2^2 && 0 \le x \le 20 \end{aligned}$$

Problem 6

$$\begin{aligned} y_1' &= -10^4 y_1 + 100 y_2 - 10 y_3 + y_4 \\ y_2' &= 10^3 y_2 + 10 y_3 + 10 y_4 \\ y_3' &= -y_3 + 10 y_4 && y_i(0) = 1,\ (i = 1(1)4) \\ y_4' &= 0.1 y_4 && 0 \le x \le 20 \end{aligned}$$

Problem 7

$$\begin{aligned} y_1' &= -y_1 + 2 \\ y_2' &= -10 y_2 + y_1^2 \\ y_3' &= -40 y_3 + 4\,(y_1^2 + y_2^2) && y_i(0) = 0,\ (i = 1(1)4) \\ y_4' &= -100 y_4 + 10(y_1^2 + y_2^2 + y_3^2) && 0 \le x \le 20 \end{aligned}$$

5. NUMERICAL RESULTS AND CONCLUSION

In this section we briefly mentioned the implementation of the methods derived in section 2 on stiff systems of ODEs given in section 4, and compare the results with the results obtained when the systems are solved using Rabeh's [5] SDIRK (3, 3) embedded in (4, 4) method.

Initially the system is considered nonstiff, hence we do simple iteration, once there is an indication of stiffness through stepfailure and the trace of the Jacobian $\delta f/\delta y$ of (1.1) is negative, the whole system is changed to stiff and solve using Newton iterations.

Here we do two iterations on k_i and convergence test used are the same as in Suleiman *et al.* [11]

Convergence test for the nonstiff system is

$$h(\rho^2(1-\rho))|b_i|\|\Delta^{(1)}k_i\| < 0.2 \text{ tol}$$

and for the stiff system is

$$hb_i\|\Delta^{(1)}k_i\|\left(\frac{\|\Delta^{(1)}k_i\|}{\|\Delta^{(0)}k_i\|}\right)^2 < 0.1\text{tol}$$

where $\Delta^{(1)}k_i$ is the difference between the second and the first iteration of k_i and ρ is $|\delta f/\delta y|$. Numerical results are given in the Appendix.

Even though methods F1(A), F1(B) and F1(C) have more $k_i s$ to be evaluated at each step, the methods performed better compared to Rabeh's [5] in terms of Jacobian evaluations, number of steps and also the time taken to solve each system of equations on the computer HP 386 over tolerances 10^{-r}, $r=2,4,6,8$. This is obvious from Tables I and II especially for higher tolerances. Comparing the SDIRK methods (4, 5) in (5, 6) themselves, F1(A) and F1(B) are at par with each other and they performed better compared to the method F1(C). This is because F1(A) and F1(B) are stable throughout the left-half plane, while F1(C) is only stable in the close region with diameter 225 unit on the left-half plane.

APPENDIX

Table I below gives the numerical results of solving the system of ODEs in section 4, and Table II gives the time taken to solve the systems on computer HP 386 over all the tolerances. The notations used are as follows:

TOL – The chosen tolerance.
FCN – The number of function evaluations
STEP – The number of successful steps
FSTEP – The number of failed steps
JACO – The number of jacobian evaluations
And the time taken is in second.

Methods

F1(A) – SDIRK (4, 5) embedded in (5, 6) with $\gamma=0.27895384$ and $c_3=-0.7$
F1(B) – SDIRK (4, 5) embedded in (5, 6) with $\gamma=4/15$ and $c_3=-0.7$
F1(C) – SDIRK (4, 5) embedded in (5, 6) with $\gamma=0.25$ and $c_3=0.3$
R1 – Rabeh's SDIRK (3, 3) embedded in (4, 4) with $\gamma=0.43586652$.

TABLE I

Problem 1

	Tol 10^{-2}				*Tol 10^{-4}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	221	16	1	2	689	52	1	2
F1(B)	221	16	1	2	637	48	1	2
F1(C)	325	24	1	2	1105	84	1	2
R1	153	16	1	1	556	61	1	2

	Tol 10^{-6}				Tol 10^{-8}			
METHOD	*FCN*	*STEP*	*JACO*	*STEP*	*FCN*	*STEP*	*JACO*	*STEP*
F1(A)	3288	251	1	3	10630	815	1	3
F1(B)	3106	237	1	3	10032	769	1	3
F1(C)	4016	307	1	3	11878	911	1	3
R1	3347	370	1	2	17413	1932	1	3

Problem 2

	Tol 10^{-2}				Tol 10^{-4}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	261	19	1	1	598	44	1	2
F1(B)	261	19	1	1	572	42	1	2
F1(C)	287	21	1	1	637	47	1	2
R1	291	31	1	1	875	95	1	2

	Tol 10^{-6}				Tol 10^{-8}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	1391	105	1	2	3406	260	1	2
F1(B)	1352	102	1	2	3302	252	1	2
F1(C)	1495	113	1	2	3627	277	1	2
R1	2667	295	1	2	14618	1622	1	2

Problem 3

	Tol 10^{-2}				Tol 10^{-4}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	337	25	1	1	750	56	1	2
F1(B)	337	25	1	1	731	55	1	2
F1(C)	376	28	1	1	809	61	1	2
R1	370	40	1	1	1063	117	1	1

	Tol 10^{-6}				Tol 10^{-8}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	1770	134	1	3	4375	333	1	5
F1(B)	1710	130	1	3	4235	323	1	5
F1(C)	1917	145	1	4	4651	355	1	5
R1	3312	366	1	2	17319	1917	2	9

Problem 4

	Tol 10^{-2}				Tol 10^{-4}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	305	22	1	2	578	43	1	2
F1(B)	292	21	1	2	603	44	1	3
F1(C)	376	28	1	1	668	49	1	3
R1	294	31	1	1	806	87	1	2

	Tol 10^{-6}				*Tol 10^{-8}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	1370	103	1	4	3306	251	1	4
F1(B)	1331	100	1	3	3202	243	1	4
F1(C)	1474	111	1	3	3514	267	1	4
R1	2399	264	1	2	9248	1022	2	4

Problem 5

	Tol 10^{-2}				*Tol 10^{-4}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	834	46	4	24	956	64	5	6
F1(B)	504	30	3	12	1038	62	5	19
F1(C)	547	37	4	6	931	67	5	3
R1	5394	596	2	5	1077	117	2	2
	Tol 10^{-6}				*Tol 10^{-8}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	9634	734	5	3	5106	382	7	5
F1(B)	7462	552	6	17	4336	320	9	4
F1(C)	2100	156	6	3	5379	407	6	6
R1	2809	309	2	3	11795	1287	1	27

Problem 6

	Tol 10^{-2}				*Tol 10^{-4}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	454	34	1	1	986	74	1	2
F1(B)	441	33	1	1	934	70	1	2
F1(C)	493	37	1	1	1077	81	1	2
R1	523	57	1	1	1495	165	1	1
	Tol 10^{-6}				*Tol 10^{-8}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	2312	176	1	2	5665	433	1	3
F1(B)	2506	190	1	3	10814	830	1	2
F1(C)	2468	188	1	2	6029	461	1	3
R1	4644	514	1	2	86735	9633	2	4

Problem 7

	Tol 10^{-2}				*Tol 10^{-4}*			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	430	32	1	2	1404	106	1	3
F1(B)	417	31	1	2	1326	100	1	3
F1(C)	664	50	1	2	1846	140	1	3
R1	415	45	1	1	1458	160	1	2

	Tol 10^{-6}				*Tol* 10^{-8}			
METHOD	*FCN*	*STEP*	*JACO*	*FSTEP*	*FCN*	*STEP*	*JACO*	*FSTEP*
F1(A)	4264	326	1	3	10608	812	2	4
F1(B)	3952	302	1	3	11002	842	2	4
F1(C)	5018	384	1	3	13199	1013	1	4
R1	5561	615	1	3	21668	2398	1	10

TABLE II

TOTAL TIME TAKEN IN SECONDS TO SOLVE THE SYSTEMS OVER ALL THE TOLERANCES

PROBLEMS	*METHODS*			
	F1(A)	*F1(B)*	*F1(C)*	*R1*
1	6.6992	6.2578	7.7969	9.3906
2	6.3203	6.1016	6.7500	10.3281
3	4.7813	4.6094	5.0977	8.8398
4	11.6992	11.4180	12.5820	21.4180
5	10.6016	8.5664	5.7695	17.0781
6	6.0898	5.9297	6.5313	50.4219
7	11.992	10.6602	13.8984	12.0898

References

[1] Norsett, S. P. (1974). Semi-explicit Runge-Kutta methods, *Mathematics and Computation*, No. 6, University of Trondheim.

[2] Crouzeix, M. (1975). *On the approximation of ordinary differential equations using Runge-Kutta methods*, Phd thesis University of Paris.

[3] Alexander, R. (1977). Diagonally implicit Runge-Kutta methods for stiff ODEs, *Siam Journal Numer Anal*, **14**(6), 1006–1021.

[4] Cash, J. R. (1979). Diagonally Implicit Runge-Kutta Formulae with error estimates, *J. Inst. Maths. Applics*, **24**, 293–301.

[5] AL-Rabeh, A. (1987). Embedded DIRK methods for the numerical integration to stiff systems of ODEs, *International. J. Computer Math.*, **21**, 65–84.

[6] Cooper, G. J. and Sayfy, A. (1979). Semi-explicit A-stable Runge-Kutta methods, *Maths of Comp.*, **33**, 541–556.

[7] Fehlberg, E. (1964). *Z. Angnew. Math. Mech.*, **44**, T17–T29.

[8] Norsett, S. P. and Thomsen, P. G. (1984). Embedded SDIRK methods of basic order three, *BIT*, **24**, 634–646.

[9] Butcher, J. C. (1987). *The Numerical Analysis of Ordinary Differential Equations, Runge-Kutta and General Linear Methods*, John Wiley and Sons.

[10] Hairer, E. and Wanner, G. (1991). *Solving Equations II, Stiff and Differential-Algebraic Problems*, Springer-Verlag.

[11] Suleiman, M. B., Ismail, F. and Atan, K. A. B. M. (1996). Partitioning ODEs using Runge-Kutta method, *Applied Mathematics and Computation*, **79**, 289–309.