ISSN 0105-8517

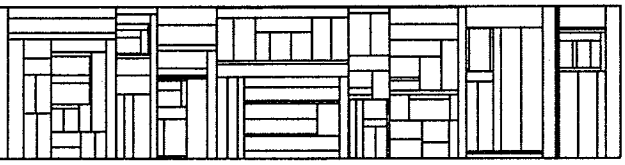
MODIFIED DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS

by

Zahari Zlatev

DAIMI PB-112
February 1980

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55



MODIFIED DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS

by

Zahari Zlatev

ABSTRACT The experimental evidence indicates that the implementation of Newton's method in the numerical solution of ordinary differential equations ($y'=f(t,y)$, $y(a)=y_0$, $t\in[a,b]$) by implicit computational schemes may cause difficulties. This is especially true in the situation where (i) $f(t,y)$ and/or $f'(t,y)$ are quickly varying in t and/or y and (ii) a low degree of accuracy is required. Such difficulties may also arise when diagonally implicit Runge-Kutta methods (DIRKM's) are used and when (i) and (ii) are satisfied. In this situation the choice of L-stable numerical methods and/or the choice of numerical methods which use minimal number of simple arithmetical operations per step will not be very successful if Newton's method fails to converge at many integration steps. In this paper some modifications in the DIRKM's are suggested so that the modified DIRKM's (MDIRKM's) will perform better than the coresponding DIRKM's when the functions f and f' are quickly varying only in t and (ii) is satisfied (more precisely, these modifications can be considered as an attempt to improve the convergence of the Newton's iteration in the above situation). An error estimation technique for the 2-stage MDIRKM's is proposed. Finally, it is shown that the MDIRKM's are more efficient than the corresponding DIRKM's when linear systems of ordinary differential equations are solved in the situation described by (i) and (ii).

1. Introduction

Consider the initial value problem for first order systems of ordinary differential equations (following Stetter [18] we shall call this problem IVP1)

$$(1.1)\quad y'=f(t,y),\quad y(a)=y_0,\quad t\in[a,b]\subset\mathbb{R},\quad y\in(\mathbb{C}^{(p+1)}[a,b])^s,$$

where s and p are positive integers.

Denote the true solution of the IVP1 by $y(t)$. Consider the grid

$$(1.2)\quad \mathbb{G}_N=\{t_\nu\in[a,b]\ /\ \nu=0(1)N, t_0=a, t_\nu<t_{\nu+1} \text{ for } \nu=0(1)N-1, t_N=b\}.$$

Very often numerical methods are used to obtain approximations y_ν to $y(t_\nu)$ at the points of the grid $\mathbb{G}_N$ according to some error tolerance ε. The methods introduced by Nørsett [13] will be discussed in this paper. Following Alexander [1] we shall call these methods diagonally implicit Runge-Kutta methods (DIRKM's). An m-stage DIRKM is based on the following formulae:

$$(1.3)\quad k_i(h_{n+1})=f(t_n+\alpha_i h_{n+1}, y_n+h_{n+1}\sum_{j=1}^{i}\beta_{ij}k_j(h_{n+1})),\quad i=1(1)m;$$

$$(1.4)\quad y_{n+1}=y_n+h_{n+1}\sum_{i=1}^{m}p_i k_i(h_{n+1})$$

where $h_{n+1}=t_{n+1}-t_n$ is the stepsize used at step $n+1$ $(n=0(1)N-1)$, $\beta_{ii}=\gamma$ $(i=1(1)m)$ and it is assumed that all y_j $(j=1(1)n)$ are already computed.

Denote: $t_n=x$, $h_{n+1}=h$ and $\Delta y=y(x+h)-y(x)$. Assume that

$y_n=y(x_n)$. Let

$$(1.5)\quad \varphi_m(h) = \Delta y - h\sum_{i=1}^{m} p_i k_i(h) \ .$$

Use the Taylor expansion ($0 < \theta < 1$)

$$(1.6)\quad \varphi_m(h) = \sum_{j=0}^{p} (h^j/j!)\varphi_m^{(j)}(0) + (h^{p+1}/(p+1)!)\varphi_m^{(p+1)}(\theta h) \ .$$

It is well-known that the DIRKM defined by (1.3)-(1.4) is of order (of consistency) p when

$$(1.7)\quad \varphi_m^{(j)}(0) = 0 \ , \quad j=0(1)p \ , \quad \varphi_m^{(p+1)}(0) \neq 0 \ .$$

Assume now that a DIRKM of order $p\geq 1$ is used in the numerical integration of (1.1). In general, some iterative process must be used in the computation of $k_i(h)$, $i=1(1)m$, because (1.3) are implicit. Newton's iterative process is commonly used in the integration codes. The use of this process for the solution of (1.3) is assumed in the further considerations. Moreover, it is assumed that the simple Gaussian elimination is applied in the decomposition (the LU factorization) of matrix $I-h\gamma f_y'$ (see Section 2). It is well-known that very often an old decomposition (obtained at some previous step j , $j<n$) can also be used at step n . Some examples where a new decomposition is normally computed only when the stepsize is changed can be constructed and arise in practice. Strategies, which attempt to keep the old decomposition even after small changes of the stepsize, have also been proposed and it has been verified that they work perfectly for some problems (1.1) . Unfortunately, there also arise situations where the old decomposition

of $I-h\gamma f'_y$ can not be used during more than one step. Moreover, for some problems (especially when a low degree of accuracy is required) even several decompositions per step are needed. This is true not only when DIRKM's are used but also for many other implicit methods. Two examples are given below in order to show that the average number of decompositions per step can be larger than one. In Table 1 the numerical results given by Enright et al [9, p. 23, Table 1] are used to compute the average numbers of decompositions per step for five different codes and for three values for the error tolerance. A wide range of test-problems is used in [9]. It should be mentioned that the numerical results for some of the test-problems are not taken into account in [9, p. 23, Table 1]. This is so e.g. for problem D6 . The implicit Runge-Kutta method, IMPRK , uses about 24.92 decompositions per step in the integration of D6 with $\varepsilon=10^{-2}$ (see [9, p.46]). The numerical results obtained by SIRKUS (a code

Tolerance	GEAR	SDBASIC	TRAPEX	IMPRK	GENRK
10^{-2}	0.27	1.47	1.72	6.67	2.67
10^{-4}	0.15	0.89	1.00	0.84	1.99
10^{-6}	0.09	0.61	0.55	0.23	1.87

Table 1

The average numbers of decompositions per step for the five codes tested by Enright et al (see [9, p. 23, Table 1]).

based on a DIRKM derived in [13]) in the integration of two chemical problems (described in [2,10]) are shown in Table 2. Note that for the bigger problem (s=63) the average numbers of de-

Tolerance	s = 15	s = 63
10^{-1}	1.79	2.27
10^{-2}	0.53	1.75
10^{-3}	0.12	0.89

Table 2

The average numbers of decompositions per step found in the integration of two chemical problems by the code SIRKUS .

compositions per step are much larger.

The results in Table 1 and Table 2 show that it is worthwhile attempting to answer the following questions. When can an old decomposition be used several times? If the problem is such that more than one decomposition per step will be needed when a DIRKM is used what can be done in order to improve the performance of the DIRKM under consideration?

The following definitions will be useful in our efforts to answer the above questions.

Definition 1.1. The IVP1 has property S if $f(t,y)$ and $f'_y(t,y)$ are slowly varying in t and y .

Definition 1.2. The IVP1 has property $\bar{S}$ if at least one of the functions $f(t,y)$ and $f'_y(t,y)$ is quickly varying in t and both functions are slowly varying in y .

Definition 1.3. The IVP1 has property S* if at least one of the functions $f(t,y)$ and $f'_y(t,y)$ is quickly varying in t and at least one of these functions is quickly varying in y .

Consider the case where the error tolerance is moderately large. In Section 2 a theorem proved by Kantorovich in 1956 (see

[11,12] is modified for the use of Newton's method for the solution of (1.3) when (1.1) is solved by a DIRKM. The theorem indicates that the average number of decompositions per step will be smaller than 1 if the IVP1 has property S. The theorem shows also that the success of the choice $\beta_{ii}=\gamma$ in the DIRKM's, in an attempt to reduce the number of simple arithmetic operations per step, depends on the convergence of the Newton's process. If the problem has property S and/or if the error tolerance is stringent, then the choice will be unconditionally successful (the same conclusion holds for the implicit Runge-Kutta methods derived by the Butcher transformation [5], see also [3] and [7]). If this is not so then the Newton iteration will often fail to converge and the number of decompositions per step may be larger than 1 . In Section 3 some modifications in the DIRKM's are proposed. The modified methods (MDIRKM's) can efficiently be used when the problems solved have property $\bar{S}$. An error estimation technique for the MDIRKM's is proposed in Section 4. In Section 5 it is shown that in the case where (1.1) is a linear IVP1 the MDIRKM's will perform better than the corresponding DIRKM's even if the problem has property S*. A brief discussion of the results is given in Section 6.

2. On the use of Newton's method in connection with DIRKM's

Assume that some approximations $k_i^0(h)$, i=1(1)m, to the solutions of (1.3) are available (only in this section the notation $k_i^*(h)$ will be used for the solution of the i'th system (1.3)). Let (for i=1(1)m and q=0,1,...)

$$(2.1)\quad \bar{f}_y'(\tau,\eta) \approx f_y'(t_n+\alpha_i h, y_n+h\sum_{j=1}^{i-1}\beta_{ij}k_j(h)+h\gamma k_i^q(h)) .$$

It is well-known that (1.3) can be solved by the quasi Newton iterative process (QNIP) defined by (i=1(1)m, q=0,1,...)

$$(2.2)\quad [I-h\gamma\bar{f}_y'(\tau,\eta)][k_i^{q+1}(h)-k_i^q(h)] = P(k_i^q(h)),$$

$$(2.3)\quad P(k_i(h)) = k_i(h)-f(t_n+\alpha_i h, y_n+h\sum_{j=1}^{i}\beta_{ij}k_j(h)).$$

For the QNIP the following theorem holds.

Theorem 2.1 Assume that

$$(2.4)\quad \Gamma = [I-h\gamma\bar{f}_y(\tau,\eta)]^{-1}$$

exists. Let the following conditions be satisfied when $k_i^0(h) \in \Omega_i$ (where Ω_i is the closed sphere defined by $\| k_i(h)-k_i^0(h) \| < r_i$, i=1(1)m):

$$(2.5) \quad \| \Gamma P_i(k_i^0(h)) \| \leq \bar{\eta}_i , \qquad i=1(1)m;$$

$$(2.6) \quad \| \Gamma P_i'(k_i^0(h)) \| \leq \delta_i , \qquad i=1(1)m;$$

$$(2.7) \quad \| \Gamma P_i''(k_i(h)) \| < K_i , \qquad k_i(h) \in \Omega_i , \qquad i=1(1)m .$$

Then we have:

(i) Existence and uniqueness. If

$$(2.8) \quad \bar{h}_i = K_i \bar{\eta}_i / (1-\delta_i)^2 < 0.5 , \quad \delta_i < 1 , \quad i=1(1)m;$$

$$(2.9) \quad r_i \geq (1-\sqrt{1-2\bar{h}_i})(1-\delta_i)/K_i , \qquad i=1(1)m;$$

when for any $i \in \{1,2,\ldots,m\}$ the equation (1.3) has a solution $k_i^*(h) \in \Omega_i$, which is unique if

$$(2.10) \quad r_i < (1+\sqrt{1-2\bar{h}_i})(1-\delta_i)/K_i , \qquad i=1(1)m .$$

(ii) Convergence. If (2.5)-(2.10) hold the QNIP is convergent (i.e. $k_i^q(h) \in \Omega_i$, $i=1(1)m$, $q=0,1,\ldots$, and $k_i^q(h) \to k_i^*(h)$ as $q \to \infty$).

(iii) Speed of convergence. If $\{k_i^q(h)\}$ is found by the QNIP then (for $i=1(1)m$ and $q=0,1,\ldots$)

$$(2.11) \quad \| k_i^*(h) - k_i^q(h) \| \leq [1-(1-\delta_i)\sqrt{1-2\bar{h}_i}]^{q+1}/K_i .$$

□

Remark 2.1 The above theorem is a slight modification of a theorem given by Kantorovich [11], see also [12,Chapter XVIII]. Similar results can be found in Robertson and Williams [14] (where some conditions containing the eigenvalues of f'_y are used, see [14, p. 28]). We prefer the formulation given by Kantorovich because it is very simple and allows us to draw immediately some conclusions about the qualitative behaviour of the QNIP (note that (2.6) measures the failure of Γ to be equal to $P_i^{-1}(k_i^0(h))$ and (2.5) measures the failure of $k_i^0(h)$ to be a good starting approximation).

Remark 2.2 Consider (2.8). From (2.6), (2.4) and (2.3) it follows that one can expect δ_i to be small if $f'_y(t,y)$ is slowly varying in t and y. From (2.5) and (2.3) it follows that one can expect $\bar{\eta}_i$ to be small if the starting approximations are good. In general, some extrapolation rules are used to obtain starting approximations when implicit Runge-Kutta methods are used. These rules will normally work well when $f(t,y)$ is slowly varying in t and y. Therefore one can expect that the same decomposition of $I-h\gamma\bar{f}'_y(\tau,\eta)$ can be used several times (even if ε is large) when the IVP1 has property S.

Remark 2.3 If the IVP1 has property $\bar{S}$ or property S* and if ε is large then the strategy of keeping the old decompositions as long as possible is not efficient; this leads to many rejected steps and extra computational work. In the above situation the results will be poorer if an attempt to keep the old decomposition even after small changes of the stepsize is

carried out (the number of rejections will be larger). If one of the above strategies is combined with restrictions in the changes of the stepsize then the algorithm so found may perform very badly (the convergence of the QNIP depends not only on h but rather on $\bar{f}'_y$ and f in this case). Therefore the last strategy may result both in a large number of steps and in a large number of rejections. However, note that the implementation of any of the above strategies will work very well if the IVP1 has property S and/or if the error tolerance is stringent. See e.g. the performance of IMPRK for problem D6 with $\varepsilon=10^{-2}$ and $\varepsilon=10^{-6}$ in [9, p. 46]. When $\varepsilon=10^{-2}$ the results are catastrophic: 5657 decompositions and 231 steps. When $\varepsilon=10^{-6}$ the results are much better: 69 decompositions and 15 steps (note too that the computing time is reduced by a factor larger than 100).

Remark 2.4 Nørsett's condition, $\beta_{ii}=\gamma$ (i=1(1)m), normally ensures that at most one decomposition per step is needed when the IVP1 has property S. However, if the IVP1 has property $\bar{S}$ or property S* and if ε is large then one should be prepared for an integration process where more than one decomposition per step will be needed even if $I-h\gamma f'_y(t_n+\alpha_1 h, y_n+h\gamma k_1^0(h))$ is decomposed at each step. This is very unfortunate if, in addition, the system (1.1) is large (the computational cost per decomposition is $O(s^3)$ simple arithmetic operations, while the computational cost of the QNIP without the decompositions is $O(s^2)$). Therefore some modifications in the DIRKM's in order to improve their performance in the above situation are desirable.

Remark 2.5 The performance of the QNIP in the implicit numerical schemes for solving IVP's 1 can also be improved by the use of predictor formulae which produce better starting approximations. This approach is discussed in [14].

3. Modified diagonally implicit Runge-Kutta methods

Introduce the sets: $A=\{\alpha_i/i=1(1)m\}$, $B=\{\beta_{ij}/i=1(1)m, j=1(1)i\}$, and $P=\{p_i/i=1(1)m\}$. Denote the method (1.3)-(1.4) by DIRKM(A,B;P). Consider also the set $A^*=\{\alpha_i=\gamma^*/i=1(1)m\}$. The method found from a DIRKM(A,B;P) by replacing A with A* will be called a modified diagonally implicit Runge-Kutta method (MDIRKM) coresponding to the DIRKM(A,B;P) if both methods are of the same order. An answer to the question of whether MDIRKM's can be constructed is given by the following theorem.

Theorem 3.1 MDIRKM's of order up to 2 can be constructed.

Proof (a) Order 2 is attainable. The order of consistency is 2 when (see (1.5)-(1.7))

$$(3.1)\qquad \varphi_m^{(0)}(0)=0, \quad \varphi_m^{(1)}(0)=0 \quad \text{and} \quad \varphi_m^{(2)}(0)=0 .$$

The first of these equalities is trivially satisfied. Consider the second equation. By the use of $(\Delta y)'=d(\Delta y)/dh=y'(x+h)=f(x+h,y(x+h))$ it is clear that

$$(3.2)\qquad \varphi_m^{(1)}(h)=f(x+h,y(x+h))-\sum_{i=1}^{m} p_i k_i(h)-h\sum_{i=1}^{m} p_i (dk_i(h)/dh)$$

where

$$(3.3)\qquad dk_i(h)/dh=\gamma^* f_t'(x+\gamma^* h, y_n+h\sum_{j=1}^{i}\beta_{ij}k_j(h))$$

$$+f_y'(x+\gamma^* h, y_n+h\sum_{j=1}^{i}\beta_{ij}k_j(h))[\sum_{j=1}^{i}\beta_{ij}k_j(h)+h\sum_{j=1}^{i}\beta_{ij}(dk_j(h)/dh)].$$

From (3.2) and (1.3) it follows that

$$(3.4)\qquad \varphi_m^{(1)}(0)=(1-\sum_{i=1}^{m} p_i)f(x,y(x))$$

and therefore $\varphi_m^{(1)}(0)=0$ implies

$$(3.5)\qquad \sum_{i=1}^{m} p_i=1 \ .$$

Consider the third equation (3.1). By the use of $(\Delta y)''=d^2(\Delta y)/dh^2=f_t'(x+h,y(x+h))+f_y'(x+h,y(x+h))f(x+h,y(x+h))$ the following equality can be obtained

$$(3.6)\qquad \varphi_m^{(2)}(h)=(\Delta y)''-2\sum_{i=1}^{m} p_i(dk_i(h)/dh)-h\sum_{i=1}^{m} p_i(d^2k_i(h)/dh^2) \ .$$

From (3.6), (1.3) and (3.3) it follows that

$$(3.7)\qquad \varphi_m^{(2)}(0)=(1-2\gamma^*\sum_{i=1}^{m} p_i)f_t'(x,y(x))$$

$$+(1-2\gamma^*\sum_{i=1}^{m} p_i\sum_{j=1}^{i}\beta_{ij})f_y'(x,y(x))f(x,y(x))$$

and it is clear that $\varphi_m^{(2)}(0)=0$ implies

$$(3.8)\qquad \gamma^*=0.5 \qquad \text{and} \qquad \sum_{i=2}^{m} p_i\sum_{j=1}^{i-1}\beta_{ij}=0.5-\gamma \ .$$

It is readily seen that the coefficients of the method can be chosen so that (3.5) and (3.8) are satisfied (and the order is 2). If e.g. $m=2$ is chosen then the same set of coefficients (p_1, p_2; β_{21} and γ) as the set considered by Nørsett [13, p. 43, formulae 6.9] will be found. Therefore MDIRKM's of order 2 can be constructed.

(b) No MDIRKM of order 3 can be constructed. This is trivial; a quadrature formula based on one point cannot be of

order higher than 2 .

□

Corollary 3.1 It is possible to construct L-stable MDIRKM's of order 2 .

Proof Assume that an MDIRKM of order 2 with m=2 corresponding to any of the Nørset methods in [13, p. 43, formulae 6.9] with γ satisfying $\gamma^2-2\gamma+0.5=0$ is constructed as described in the proof of Theorem 3.1. Apply the method so found to the test-equation $y'=\lambda y$ ($\lambda \in \mathbb{R}_-$). Then the two methods (the MDIRKM and the Nørsett method) are equivalent. Therefore the MDIRKM is L-stable (because the corresponding Nørsett method is).

□

Remark 3.1 In this paper we shall consider only L-stable MDIRKM's with m=p=2. For these methods $A^*=\{0.5, 0.5\}$. We shall compare the performance of such MDIRKM's with the performance of the corresponding DIRKM's (i.e. DIRKM's which have the same sets of coefficients B and P as the MDIRKM's under consideration).

Remark 3.2 If the IVP1 has property S and/or the error tolerance is stringent then the use of a MDIRKM will not give any advantage compared with the use of one of the corresponding DIRKM's. Note that if the IVP1 has property S and if it is rewritten in autonomous form, then both methods will perform in the same way. Note too, that in this case the use of DIRKM's of order larger than 2 may be more efficient than the use of a MDIRKM (see also Section 6).

Remark 3.3 Let ε be large. Assume that it has been established in some way that the IVP1 has property $\bar{S}$. Then

$$(3.9)\qquad [I-h\gamma f'_y(t_n+\alpha_1 h, y_n+h\gamma k_1^0(h))][k_i^{q+1}(h)-k_i^q(h)]=P(k_i^q(h))$$

can be used instead of (2.2) in the QNIP. This means that we agree to perform a decomposition at each step but we shall attempt to avoid the use of more than one decomposition per step. If any 2-stage DIRKM is used then (3.9) will not help very much (often the QNIP will not converge at the second stage because $\alpha_1 \neq \alpha_2$). It is clear that the use af a 2-stage MDIRKM (where $\alpha_1 = \alpha_2 = 0.5$) will be more efficient in this case.

Remark 3.4 The modification of the QNIP as in Remark 3.3 is the most efficient way to solve non-linear problems (1.1) (in the situation described above) with MDIRKM's. But this is not the only way to use the MDIRKM's. They can also be used with a QNIP based on (2.2). However, for the special situation considered here the use of the QNIP based on (3.9) is much more efficient.

Remark 3.5 Theorem 3.1 and Corollary 3.1 have been proved for linear systems of ordinary differential equations in [16], where the linear function (with regard to the second argument y) $A(t)y+b(t)$ is used instead of $f(t,y)$.

4. Error estimation technique

A device which can be used to control the local truncation error during the integration process performed by some 2-stage MDIRKM of order 2 will be described in this section. The following statements, which are well-known (and only slightly modified for our methods), are needed before the formulation of the main result in this section (Theorem 4.2).

Definition 4.1 Consider the IVP1 defined by

$$(4.1) \qquad y'=f(t,y) \ , \qquad y(t_n)=y_n \ .$$

Assume that an m-stage Runge-Kutta method (not necessarily an MDIRKM or a DIRKM) of order p is used to find y_{n+1}. Then

$$(4.2) \qquad T^p_{n+1}=(\varphi_m^{(p+1)}(0)/(p+1)!)h^{p+1} \qquad (h=h_{n+1})$$

will be called the principal part of the local truncation error.

Theorem 4.1 Assume that y_{n+1} is computed by an MDIRKM (with $m=p=2$). Consider another Runge-Kutta method of order 3 defined as follows: $k_1(h)$ and $k_2(h)$ are the vectors computed by the MDIRKM,

$$(4.3) \qquad k_i(h)=f(t_n+\alpha_i h, y_n+h\sum_{j=1}^{m}\beta_{ij}k_j(h)) \ , \qquad i=3(1)m \ ,$$

$$(4.4) \qquad \hat{y}_{n+1}=y_n+h\sum_{i=1}^{m}\hat{p}_i k_i(h) \ .$$

Then if the terms which contain h^4 are neglected in (1.6) the principal part of the local truncation error can be written in the following way

$$(4.5)\quad T^2_{n+1}=\hat{y}_{n+1}-y_{n+1}=h\sum_{i=1}^{2}(\hat{p}_i-p_i)k_i(h)+h\sum_{i=3}^{m}\hat{p}_i k_i(h)\ .$$

□

The problem is: how to choose the auxiliary method (4.3)-(4.4)? It is not possible to construct an MDIRKM of order 3 (see Theorem 3.1). It is not desirable to use implicit formulae in (4.3) (this may cause extra decompositions). Therefore the only choice, which will ensure that the arithmetical cost of the error estimator formulae (4.3)-(4.4) is $O(s^2)$, is $\beta_{ij}=0$, $i=3(1)m$, $j\geq i$. By this choice the following theorem can be proved.

Theorem 4.2 The smallest number m which allows us to construct an error estimator (4.3)-(4.4) with explicit formulae (4.3) for a 2-stage MDIRKM of order 2 is four.

Proof The method (4.4) will be of order 3 if its coefficients satisfy the following conditions:

$$(4.7)\quad \sum_{i=1}^{m}\hat{p}_i=1\ ,$$

$$(4.8)\quad \sum_{i=1}^{m}\hat{p}_i\alpha_i=0.5\ ,$$

$$(4.9)\quad \sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\beta_{ij}=0.5\ ,$$

$$(4.10)\quad \sum_{i=1}^{m}\hat{p}_i\alpha_i^2=1/3\ ,$$

$$(4.11)\quad \sum_{i=1}^{m}\hat{p}_i\alpha_i\sum_{j=1}^{i}\beta_{ij}=1/3\ ,$$

$$(4.12)\quad \sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\alpha_i\beta_{ij}=1/6\ ,$$

$$(4.13)\quad \sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\beta_{ij}\sum_{\nu=1}^{j}\beta_{j\nu}=1/6\ ,$$

$$(4.14)\quad \sum_{i=1}^{m}\hat{p}_i\Big(\sum_{j=1}^{i}\beta_{ij}\Big)^2=1/3\ .$$

(a) Let us choose $m=3$. Then it is easily seen that the system (4.7)-(4.14) has no solution (consider e.g. (4.7), (4.8) and (4.10) and take into account that (3.5) and (3.8) must also be satisfied). Note that when $m=3$ system (4.7)-(4.14) is a system of 8 equations with 6 unknown variables.

(b) Assume that $m=4$. Then (4.7)-(4.14) is a system of 8 equations with 11 unknowns. It can be proved that this system has a solution.

□

Remark 4.1 If the system is linear, then (4.14) can be removed. Nevertheless, again only a 4-stage error estimator with the 2-stage MDIRKM's can be constructed, see [16]. However, it is possible to construct special error estimators for linear systems; see more details in the next section, where the difference in the behaviour of the MDIRKM's for non-linear and for linear systems is discussed.

5. Application of MDIRKM's in the solution of linear systems

Assume that

$$f(t,y) = A(t)y + b(t) \quad . \tag{5.1}$$

The use of a 2-stage MDIRKM in this case may be very efficient. This can be explained as follows.

(i) The discretization of a linear IVP1 leads (at each step n, $n=1(1)N$) to the solution of two linear algebraic systems with the same coefficient matrix, $I-h\gamma A(t_n+0.5h)$, when a 2-stage MDIRKM is applied. The use of the corresponding DIRKM results in two linear algebraic systems also, however their coefficient matrices are different. This means that if one replaces the QNIP by the simple Gaussian elimination (GE) then the MDIRKM will require one decomposition per successful step, while two decompositions are needed with any 2-stage DIRKM.

(ii) If a two-stage MDIRKM is used with GE then the problem of finding good starting approximations $k_i^0(h)$ is avoided. Note that the problem of determination of good starting approximations is very important for the performance of the QNIP, see (2.8), (2.5), (2.2) and Remark 2.2. When linear multistep methods are implemented the starting approximations are normally computed by some explicit formula which is of the same order as the implicit formula used at step n $(n=1(1)N)$ and one may expect them to be good. It is not so easy to find good approximations when implicit Runge-Kutta methods are used. Therefore when the problem is linear the use of GE, where starting approximations are not needed, may be very efficient (see [16,17]). The problem of finding good starting approximations will also be avoided if DIRKM's are used with GE (but the use of DIRKM's with GE is not so efficient, see (i)). If DIRKM's are used with QNIP then an old decomposition (obtained at some previous step j) can be used to compute starting approximations at the current step n $(n>j)$, however this will be successful when the linear IVP1 has property S and/or when ε is small. If this is not so then one

can attempt to use the QNIP with (3.9). In this way no problems with the starting approximations arise at the first stage of the DIRKM. However, the second stage may cause difficulties (rejections of the step; this is very unfortunate because if this happens then the first stage has to be recomputed also).

(iii) The linearity of the IVP1 can be used to develop special computational schemes for linear problems (the computational scheme consists of the basic MDIRKM and of the error estimator (4.3)-(4.4)). In this case (4.14) is not needed (this condition arises from equating the coefficient before f''_{yy} to zero; note that for the DIRKM's where $\alpha_i = \sum_{j=1}^{i} \beta_{ij}$ (4.14) is equivalent to (4.10) and (4.11) and we can not see how the linearity of the IVP1 can be exploited to develop special computational schemes whose basic method is a DIRKM and which are valid only for linear problems). A special scheme for linear systems (see [16]) is given below.

$$k_3(h)=A(t_n)y_n+b(t_n) \quad , \tag{5.2}$$

$$A=I-h(1-\sqrt{2}/2)A(t_n+h/2) \quad , \tag{5.3}$$

$$Ak_1(h)=A(t_n+h/2)y_n+b(t_n+h/2) \quad , \tag{5.4}$$

$$Ak_2(h)=A(t_n+h/2)[y_n+h(\sqrt{2}-1)k_1(h)]+b(t_n+h/2) \quad , \tag{5.5}$$

$$y_{n+1}=y_n+(h/2)[k_1(h)+k_2(h)] \quad , \tag{5.6}$$

$$k_4(h)=A(t_n+h)\{y_n+h[(\sqrt{2}-1)(k_1(h)-k_2(h))+k_3(h)]\}+b(t_n+h) \quad , \tag{5.7}$$

$$\| T^2_{n+1} \|_2=(h/6)\| k_1(h)+k_2(h)-k_3(h)-k_4(h) \|_2 \quad . \tag{5.8}$$

This scheme is very efficient because (a) only one decom-

position per step is needed, (b) only two matrix computations per step are performed ($A(t_n+h)$ can be used at the next step), (c) if the step is rejected and has to be recomputed with a smaller stepsize then it is not necessary to recompute k_3. Note too that in the code described in [16] matrix $\bar{A}=(1-\sqrt{2}/2)^{-1}h^{-1}A$ is used instead of A. In this way the computational cost needed to obtain the coefficient matrix of the linear algebraic systems (5.4) and (5.5) is reduced from $O(s^2)$ to $O(4s)$ (here the fact that the right-hand sides of (5.4) and (5.5) have to be divided by $(1-\sqrt{2}/2)h$ is also taken into account).

The computational scheme described above has been tested in the solution of chemical problems arising in the nuclear resonance theory (see [2,10]) and has been compared with the code SIRKUS (based on a DIRKM described in [13]; it should be mentioned that some previous investigations had shown that SIRKUS is the best solver for these chemical problems among several codes tested in [15]). The computing time is 3-5 times smaller when the code Y12NBF (the code based on the above scheme) is used, see Table 3.

Tolerance	s = 15	s = 63
EPS0=500	11(47)	180(1067)
EPS0=400	13(51)	231(1081)
EPS0=300	13(53)	231(1105)
EPS0=200	16(56)	261(1200)
EPS0=100	21(65)	340(1285)

Table 3

The computing times obtained in the solution of two chemical problems on UNIVAC 1100/82 by Y12NBF and SIRKUS (the results for SIRKUS are given in brackets, the error tolerance, EPS0, is introduced in [15]).

It must be mentioned that the linearity is not exploited by SIRKUS. The results can be improved by using the linearity but will still be poorer than the results obtained by the code based on an MDIRKM.

In all considerations in this paragraph it is assumed that the problem has property $\bar{S}$ or property S* and that the error tolerance is not stringent.

(iv) The use of 2-stage MDIRKM's of order 2 with GE will be very efficient even if the linear IVP1 has property S* (and ε is large). This is not so when a 2-stage DIRKM is used because it is not efficient to replace the QNIP with GE (at least if s is large) and the fact that the IVP1 has property S* will cause difficulties in the performance of the QNIP (when the QNIP is based on (3.9) this is true for the second stage).

(v) It is easy to implement sparse matrix technique for the computational scheme (5.2)-(5.8). This has been done in [17] by the use of some ideas described in [19,20,22]. Numerical results are also given in [17].

6. Some concluding remarks

It is necessary to emphasize that the MDIRKM's will be efficient only when the IVP1 has property $\bar{S}$ (also property S* when the problem is linear) and ε is large. If this is not so then DIRKM's of order $p \geq 2$ may perform better. If the error tolerance is stringent then the code STRIDE (see [4,7,8]), which is based on singly implicit Runge-Kutta methods ([3], see also [6]; these methods are derived by the use of a transformation proposed in [5]) implemented in a variable stepsize variable formula manner, will work much better than any MDIRKM (whose order can not ex-

ceed 2). This means that the MDIRKM's must be used carefully. If the problem is large and if the user can establish that the non-linear IVP1 (which has to be solved) has property $\bar{S}$ then the use of MDIRKM's will normally be very efficient. The use of MDIRKM's with large linear problems may also be very efficient, especially if the linearity is exploited. Note that large linear problems arise often in practice (e.g. in the solution of some parabolic partial differential equations, see [21], or in chemistry, see[17]) and an investigation of the properties of the problem may result in a considerable improvement of the efficiency of the numerical integration when the right method is chosen.

References

1. R. Alexander(1977), Diagonally implicit Runge-Kutta methods for stiff ODE's, SIAM J. Numer. Anal. 14, pp. 1006-1021.

2. H. Bildsøe, J. P. Jacobsen and K. Schaumburg(1975), Application of density matrix formalism in NMR spectroscopy I.Development of a calculation scheme and some simple examples, Journal of Magnetic Resonance 23, pp. 137-151.

3. K. Burrage(1978), A special family of Runge-Kutta methods for solving stiff differential equations, BIT 18, pp. 22-41.

4. K. Burrage, J. C. Butcher and F. H. Chipman(1979), An implementation of singly-implicit Runge-Kutta methods, BIT,to appear.

5. J. C. Butcher(1976), On the implementation of implicit Runge-Kutta methods, BIT 16, pp. 237-240.

6. J. C. Butcher(1979), A transformed implicit Runge-Kutta method, J. Assoc. Comput. Mach. 26, pp.731-738.

7. J. C. Butcher, K. Burrage and F. H. Chipman(1979), STRIDE -

stable Runge-Kutta integrator for differential equations, Computational Mathematics Report - March 1979, Department of Mathematics, University of Auckland, Auckland, New Zealand.

8. F. Chipman(1979), Some experiments with STRIDE, Working papers for the 1979 SIGNUM Meeting on Numerical Ordinary Differential Equations (R.D.Skeel,ed.),Department of Computer Science, University of Illinois at Urbane-Champaign, Urbana, Illinois, USA.

9. W. H. Enright, T. E. Hull and B. Lindberg(1975), Comparing methods for stiff systems of ODE's, BIT 15, pp.10-48.

10. J. P. Jacobsen, H. K. Bildsøe and K. Schaumburg(1976), Application of density matrix formalism in NMR spectroscopy II. The one-splin-1 case in anisotropic phase, Journal of Magnetic Resonance 23, pp.153-164.

11. L. V. Kantorovich (1956), On integral equations, Uspekhi Math. Nauk 11, pp.3-29.

12. L. V. Kantorovich and G. P. Akilov(1964), Functional analusis in normed spaces, Pergamon Press, Oxford.

13. S. P. Nørsett(1974), Semi explicit Runge-Kutta methods, Department of Mathematics, University of Trondheim, Trondheim, Norway, Mathematics and Computation No.6/74,ISBN 82-7151-009-6.

14. H. H. Robertson and J. Williams(1975), Some properties of algorithms for stiff differential equations, J. Inst. Math. Appl. 16, pp. 23-34.

15. K. Schaumburg and J. Wasniewski(1978), Use of a semiexplicit Runge-Kutta integration algorithm in a spectroscopic problem, Computers and Chemistry 2, pp. 19-25.

16. K. Schaumburg, J. Wasniewski and Z. Zlatev(1979), Solution of ordinary differential equations with time dependent coef-

ficients. Development of a semiexplicit Runge Kutta algorithm and application to a spectroscopic problem, Computers and Chemistry, Vol. 3, pp. 57-63.

17. K. Schaumburg, J. Wasniewski and Z. Zlatev(1979), On the use of sparse matrix technique in the numerical integration of stiff systems of linear differential equations, Computers and Chemistry, to appear.

18. H. J. Stetter(1973), Analysis of discretization methods for ordinary differential equations, Springer, Berlin.

19. Z. Zlatev(1979), Use of iterative refinement in the solution of sparse linear systems, Institute of Mathematics and Statistics, The Royal Veterinary and Agricultural University, Copenhagen, Report No. 1/79.

20. Z. Zlatev(1980) , On some pivotal strategies in Gaussian elimination by sparse technique, SIAM J. Numer. Anal. 17, No.1, to appear.

21. Z. Zlatev and P. G. Thomsen(1979), Application of backward differentiation methods to the finite element solution of time dependent problems, Int. J. num. Meth. Engng 14, pp. 1051-1061.

22. Z. Zlatev and J. Wasniewski(1978), Package Y12M - solution of large and sparse systems of linear algebraic equations, Mathematics Institute, University of Copenhagen, Copenhagen, Denmark, Preprint Series No. 24 1978.