SIAM J. SCI. STAT. COMPUT.
Vol. 2, No. 3, September 1981

MODIFIED DIAGONALLY IMPLICIT RUNGE-KUTTA METHODS*

ZAHARI ZLATEV†

Abstract. The experimental evidence indicates that the implementation of Newton's method in the numerical solution of systems of ordinary differential equations (ODE's) $y'=f(t,y)$, $y(a)=y_0$, $t\in[a,b]$ by implicit computational schemes may cause difficulties. This is especially true if (i) $f(t,y)$ and/or $f'_y(t,y)$ 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 in the situation described by (i) and (ii). In this paper some modified DIRKM's (MDIRKM's) are derived. The use of MDIRKM's is an attempt to improve the performance of Newton's method in the case where f and f'_y are quickly varying only in t. The stability properties of the MDIRKM's are studied. An error estimation technique for the new methods is proposed. Some numerical examples are presented.

Key words. ordinary differential equations (ODE's), numerical solution, Runge–Kutta methods, diagonally implicit schemes, order of accuracy, quasi-Newton iterative process, Gaussian elimination, matrix factorizations per step, starting approximations, absolute stability, *AN*-stability, *LN*-stability, error estimation, embedding, computational work per step, solving linear systems of ODE's

1. Introduction. Consider the initial value problem for first order systems of ordinary differential equations (following Stetter [23] we shall call this problem IVP1):

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

where s and p are positive integers.

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

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

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 [18] 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 formulae

$$(1.3)\qquad k_i(h_{n+1})=f\left(t_n+\alpha_i h_{n+1},\ y_n+h_{n+1}\left(\sum_{j=1}^{i-1}\beta_{ij}k_j(h_{n+1})+\gamma k_i(h_{n+1})\right)\right),\qquad i=1(1)m,$$

$$(1.4)\qquad y_{n+1}=y_n+h_{n+1}\sum_{i=1}^{m}p_ik_i(h_{n+1}),$$

where $h_{n+1}=t_{n+1}-t_n$ is the stepsize used at step $n+1$ ($n=0(1)N-1$). The coefficients of (1.3)–(1.4) are often written in the form

$$(1.5)\qquad \begin{array}{c|cccccc}
\alpha_1 & \gamma & & & & & \\
\alpha_2 & \beta_{21} & \gamma & & & & \\
\vdots & \vdots & \vdots & \ddots & & & \\
\alpha_m & \beta_{m1} & \beta_{m2} & \cdots & \beta_{mm-1} & \gamma \\
\hline
 & p_1 & p_2 & \cdots & p_{m-1} & p_m
\end{array}\qquad\text{or}\qquad \begin{array}{c|c}
\alpha & B \\
\hline
 & p^T
\end{array}.$$

* Received by the editors March 18, 1980, and in final revised form March 16, 1981. This work was sponsored in part by the Department of Computer Science, Aarhus University, Aarhus, Denmark.

† Department of Computer Science, Aarhus University, Aarhus, Denmark. Currently at the National Agency of Environmental Protection, Air Pollution Laboratory, DK-4000 Roskilde, Denmark.

It should be mentioned here that the following equalities hold for the DIRKM's:

$$\alpha_i = \sum_{j=1}^{i} \beta_{ij}, \quad (\beta_{ii} = \gamma), \quad i = 1(1)m, \tag{1.6}$$

but for the methods considered in this paper (1.6) will not be satisfied.

It should be emphasized that the functions k_i depend not only on the stepsize but also on the independent variable t_n. Therefore, it is more correct to use the notation $k_i(t_n + \alpha_i h_{n+1})$. Abbreviating this notation to $k_i(h_{n+1})$, we want to underline the important fact that if the first j $(j < m)$ functions (1.3) have already been computed by some iterative procedure and if the stepsize has to be changed because the iterative procedure fails to converge for k_{j+1}, then the first j functions (1.3) have to be recalculated.

The order of the method described by (1.3)–(1.4) or (1.5) can be defined as follows. Let

$$t_n = x, \quad h_{n+1} = h, \quad \Delta y = y(x+h) - y(x). \tag{1.7}$$

Assume that $y_n = y(x)$. Consider

$$\varphi_m(h) = \Delta y - h \sum_{i=1}^{m} p_i k_i(h). \tag{1.8}$$

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

$$\varphi_m(h) = \sum_{j=0}^{p} \left(\frac{h^j}{j!}\right) \varphi_m^{(j)}(0) + \left(\frac{h^{p+1}}{(p+1)!}\right) \varphi_m^{(p+1)}(\theta h). \tag{1.9}$$

The order of the method is p when

$$\varphi_m^{(j)}(0) = 0 \quad \text{for } j = 1(1)p, \qquad \varphi_m^{(p+1)}(0) \neq 0. \tag{1.10}$$

Assume now that a DIRKM of order $p \geqq 1$ 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. The quasi-Newton iterative process (QNIP) is commonly used in the integration codes. The use of QNIP in the solution of (1.3) is assumed from now on. Moreover, it is assumed that the simple Gaussian elimination (GE) is applied in the decomposition (the LU factorization) of the matrix $I - h\gamma f'_y$ (see § 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 problems 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 in 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 cannot be used during more than one step. 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 in Enright et al. [13, p. 23] are used to compute the average numbers of decompositions per steps for 5 codes and for 3 values of the error tolerance. A wide range of test-problems is used in [13]. It should be mentioned that the numerical results for some of the test-problems are not taken into

TABLE 1

The average numbers of the decompositions per step for the 5 codes tested by Enright et al. [13, *p.* 23]. *The codes are based on backward differentiation formulae* (GEAR), *the trapezoidal rule with extrapolation* (TRAPEX), *second derivative multistep formulae* (SDBASIC), *a fully implicit Runge-Kutta method* (IMPRK) *and a generalized Runge-Kutta technique* (GENRK).

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

account in Table 1. This is so, for example, for problem D6. The code IMPRK uses about 24.92 decompositions per step in the integration of D6 with $\varepsilon = 10^{-2}$ (see [13, p. 46]).

The numerical results obtained by SIRKUS (a code based on DIRKM's derived in [18]) in the integration of two chemical problems [2], [15] are shown in Table 2. Note that for the bigger problem ($s = 63$) the average numbers of decompositions per step are 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.

TABLE 2

The average numbers of decompositions per step found in the integration of two chemical problems by the code SIRKUS *which is based on* DIRKM's.

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

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.

In § 2 a theorem proved by Kantorovich in 1956 (see [16], [17]) is modified for the use of the QNIP in the solution of (1.3), when (1.1) is solved by a DIRKM. The theorem indicates that the QNIP can cause difficulties in the numerical integration when the IVP1 has not property S. Some modified DIRKM's (MDIRKM's) are derived in § 3. The stability properties of the MDIRKM's are discussed in § 4. An error estimation technique is proposed in § 5. Some applications of the MDIRKM's for linear IVP's 1 are given in § 6. A brief discussion of the results is presented in the last section.

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 ith system (1.3)). Let (for $i=1(1)m$ and $q=0,1,\cdots$)

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

Then the QNIP can be applied in the solution of (1.3) as follows:

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

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

For the QNIP the following theorem holds.

THEOREM 2.1. *Assume that*

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

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

$$\|\Gamma P(k_i^0(h))\|\leqq\bar{\eta}_i,\qquad i=1(1)m; \tag{2.5}$$

$$\|I-\Gamma P'(k_i^0(h))\|\leqq\delta_i,\qquad i=1(1)m; \tag{2.6}$$

$$\|\Gamma P''(k_i(h))\|\leqq K_i,\qquad k_i(h)\in\Omega_i,\quad i=1(1)m. \tag{2.7}$$

Then we have:

(i) *Existence and uniqueness. If*

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

$$r_i\geqq\frac{(1-\sqrt{1-2\bar{h}_i})(1-\delta_i)}{K_i},\qquad i=1(1)m, \tag{2.9}$$

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

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

(ii) *Convergence. If* (2.5)–(2.10) *hold, then the* QNIP *is convergent* (*i.e.*, $k_i^q(h)\in\Omega_i$, $i=1(1)m$, $q=0,1,\cdots$, *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* $q=0,1,\cdots$)

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

The above theorem is a modification of a result proved in [16] (see also [17, Chap. XVIII]). Similar results can be found in [19] (where some conditions containing the eigenvalues of f'_y are used, see [19, p. 28]). We prefer the formulation given by Kantorovich because it is very simple and allows us immediately to draw some conclusions about the qualitative behavior of the QNIP. Indeed, note that (2.6) measures the failure of Γ to be a good approximation to $[P'(k_i^0(h))]^{-1}$, and (2.5) measures the failure of $k_i^0(h)$ to be a good starting approximation. When the problem has property S both δ_i and $\bar{\eta}_i$ will normally be small ($\bar{\eta}_i$ are small because the extrapolation rules which are commonly used for obtaining starting approximations $k_i^0(h)$ work in general well in this case; δ_i are small because Γ is a good approximation to $[P'(k_i^0(h))]^{-1}$ even if it is calculated in a previous step $j<n$). This means that the *strategy of keeping the old decomposition will work well when* (1.1) *has property S.* If the IVP1 has property $\bar{S}$ or property S^* and if, in addition, ε is large, then the above strategy may cause difficulties. The results will be poorer when an attempt to keep the old decomposition even after small changes of the stepsize is carried out (this leads to a large number of rejected steps). If one of the above strategies is combined with restrictions in the changes of the stepsize, then the algorithms so found may perform very badly. However, the same algorithm can be efficient if the IVP1 has property S and/or if ε is small. See, for example, the performance of IMPRK for problem D6 [13, 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).

The above analysis shows that Nørsett's condition $\beta_{ii} = \gamma$ for $i = 1(1)m$ [18] normally ensures that at most one decomposition per step is needed if the problem has property S. However, if the IVP1 has property $\bar{S}$ or 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 the matrix $I - h\gamma f'_y(t_n + \alpha_1 h, y_n + h\gamma k_1^0(h))$ is decomposed at the beginning of each step. This is very unfortunate if the system 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 $O(s^2)$). It should also be pointed out that the transformation of (1.1) to autonomous form will not change the situation. If the nonautonomous problem (1.1) has not property S, then the transformed autonomous problem has not property S either. Moreover, in many practical problems t has a special physical meaning, and it is desirable to keep t as independent variable. Finally, if the problem is linear, then the transformation to autonomous form may cause some extra computations (because the transformed problem will in general be nonlinear).

Theorem 2.1 and the above analysis indicate that an attempt to improve the performance of the DIRKM's in the case where the problem has not property S may be worthwhile.

3. Modified diagonally implicit Runge-Kutta methods.

Replace (1.6) with the condition $\alpha_i = \gamma^*$ for $i = 1(1)m$. The method so found will be called a modified DIRKM (MDIRKM) if it has the same order as the *corresponding* DIRKM (the DIRKM which has the same coefficients γ, β_{ij} and p_i but the coefficients α_i satisfy (1.6)). An answer to the question 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.* Consider (1.10). It is obvious that $\varphi_m(0) = 0$ is satisfied. Since (see (1.3)–(1.10))

$$(3.1) \qquad (\Delta y)' = \frac{d(\Delta y)}{dh} = y'(x+h) = f(x+h, y(x+h)),$$

$$\begin{aligned}\frac{dk_i(h)}{dh} &= \gamma^* f'_t\left(x+\gamma^* h,\ y_n + h\sum_{j=1}^{i}\beta_{ij}k_j(h)\right)\\ &\quad + f'_y\left(x+\gamma^* h,\ y_n+h\sum_{j=1}^{i}\beta_{ij}k_j(h)\right)\left[\sum_{j=1}^{i}\beta_{ij}k_j(h)+h\sum_{j=1}^{i}\beta_{ij}\left(\frac{dk_j(h)}{dh}\right)\right],\end{aligned} \tag{3.2}$$

$$\varphi'_m(h) = f(x+h,\ y(x+h)) - \sum_{i=1}^{m} p_i k_i(h) - h\sum_{i=1}^{m} p_i\left(\frac{dk_i(h)}{dh}\right), \tag{3.3}$$

it is clear that

$$\varphi'_m(0) = \left(1-\sum_{i=1}^{m} p_i\right) f(x, y(x)), \tag{3.4}$$

and therefore $\varphi'_m(0)=0$ implies

$$\sum_{i=1}^{m} p_i = 1. \tag{3.5}$$

From

$$(\Delta y)'' = \frac{d^2(\Delta y)}{dh^2} = f'_t(x+h,\ y(x+h)) + f'_y(x+h,\ y(x+h))f(x+h,\ y(x+h)), \tag{3.6}$$

$$\varphi''_m(h) = (\Delta y)'' - 2\sum_{i=1}^{m} p_i\left(\frac{dk_i(h)}{dh}\right) - h\sum_{i=1}^{m} p_i\left(\frac{d^2k_i(h)}{dh^2}\right), \tag{3.7}$$

it follows that

$$\varphi''_m(0) = \left(1-2\gamma^*\sum_{i=1}^{m} p_i\right) f'_t(x, y(x)) + \left(1-2\sum_{i=1}^{m} p_i \sum_{j=1}^{i}\beta_{ij}\right) f'_y(x, y(x)) f(x, y(x)), \tag{3.8}$$

and therefore $\varphi''_m(0)=0$ implies

$$\gamma^* = 0.5 \quad \text{and} \quad \sum_{i=2}^{m} p_i \sum_{j=1}^{i-1}\beta_{ij} = 0.5-\gamma. \tag{3.9}$$

It is readily seen that the coefficients of the method can be chosen so that (3.5) and (3.9) are satisfied (and the order is 2). If, for example, $m=2$, then

$$\begin{array}{c|cc} 0.5 & \gamma & \\ 0.5 & (0.5-\gamma)/p & \gamma \\ \hline & 1-p & p \end{array} \tag{3.10}$$

can easily be found.

(b) *No* MDIRKM *of order* 3 *can be constructed.* This is trivial; no quadrature formula based on one point can be of order higher than 2. □

Assume that the IVP1 has property $\bar{S}$ and that ε is large. For all $\alpha_i = \gamma^*$, then one could expect $\bar{f}'_y(\tau, \eta)$ to be a good approximation to all matrices in the right-hand side of (2.1) and the QNIP will perform well at all stages during step n. If (1.6) are satisfied the above statements will often not hold. This shows that the MDIRKM's are introduced in an attempt to improve the performance of the QNIP when the IVP1 has property $\bar{S}$. However, if the IVP1 is linear, then the MDIRKM's are efficient not only when the problem has property $\bar{S}$ but also when the problem has property S^*, see § 6.

4. Stability properties of the MDIRKM's.

Let q be a complex constant. Assume that $\operatorname{Re} q \leqq 0$ and consider the model-equation

$$y' = qy, \qquad y \in \mathbb{R}. \tag{4.1}$$

It is well known (see, e.g., [5]) that the use of any RK method in the solution of (4.1) leads to

$$y_{n+1} = R(z) y_n, \qquad (n = 1(1)N), \tag{4.2}$$

where (see (1.5))

$$R(z) = 1 + zp^T (I - zB)^{-1} e, \quad z = h_{n+1} q, \quad e = (1, 1, \cdots, 1)^T. \tag{4.3}$$

The method is A-stable [12] if

$$|R(z)| \leqq 1 \quad \forall \operatorname{Re}(z) \leqq 0. \tag{4.4}$$

Since (4.3) does not depend on α, the following result is clear.

THEOREM 4.1. *An* MDIRKM *is A-stable if and only if the corresponding* DIRKM *is A-stable.*

If, for example, $m = 2$, then it is well known that for the DIRKM's

$$R(z) = [1 + (1 - 2\gamma) z + (\gamma^2 - 2\gamma + 0.5) z^2] / (1 - \gamma z)^2 \tag{4.5}$$

and the methods are A-stable if $\gamma \geqq 0.25$. Theorem 4.1 shows that this result holds also for the corresponding MDIRKM's (with $m = 2$).

It has already been mentioned that the MDIRKM's are efficient in the solution of linear systems (1.1), i.e., when

$$f(t, y) = A(t) y + b(t). \tag{4.6}$$

Therefore, it seems to be useful to investigate the AN-stability of the MDIRKM's. The notion AN-stability was introduced by Burrage and Butcher in [5]. Let $q(t)$ be a continuous complex-valued function with $\operatorname{Re} q(t) \leqq 0$ for $t \in [a, b]$. Consider the nonautonomous model-equation

$$y' = q(t) y, \qquad q(t) \in \mathbb{R}. \tag{4.7}$$

The implementation of any RK method to (4.7) leads to

$$y_{n+1} = K(Z) y_n, \quad Z = \operatorname{diag}(z_1, z_2, \cdots, z_m), \quad z_i = h_{n+1} q(t_n + \alpha_i h_{n+1}), \quad i = 1(1)m, \tag{4.8}$$

where (see (1.5) again)

$$K(Z) = 1 + p^T Z (I - BZ)^{-1} e, \qquad e = (1, 1, \cdots, 1)^T. \tag{4.9}$$

The method is said to be AN-stable if

$$|K(Z)| \leqq 1 \quad \forall \operatorname{Re}(z_i) \leqq 0, \quad i = 1(1)m. \tag{4.10}$$

While AN-stability implies A-stability, the converse statement is, in general, not true (see the example given in [5, p. 49]). However, for the MDIRKM's the following result holds.

THEOREM 4.2. *AN-stability is equivalent to A-stability for the* MDIRKM'*s.*

A remarkably simple criterion for AN-stability has been given by Burrage and Butcher [5]. This criterion can be described as follows. Consider the symmetric matrix M whose elements are $m_{ij} = p_i \beta_{ij} + p_j \beta_{ji} - p_i p_j$. For all $p_i \geqq 0$, if M is a semipositive definite matrix, then the RK method is said to be algebraically stable.

THEOREM 4.3 (Burrage and Butcher [5, *p.* 50]. *An algebraically stable* RK *method is AN-stable and, if* $\alpha_1, \cdots, \alpha_m$ *are distinct, it conversely holds that an AN-stable* RK *method is algebraically stable.*

The use of Theorem 4.3 to the 2-stage DIRKM's and the corresponding MDIRKM's gives

THEOREM 4.4. *The* 2-*stage* MDIRKM *described by* (3.10) *and the corresponding* DIRKM *are algebraically stable if* $p = 0.5$ *and* $\gamma \geqq 0.25$.

Proof. Matrix M is semipositive definite if (i) $\det(M) \geqq 0$, (ii) $m_{11} \geqq 0$ and (iii) $m_{22} \geqq 0$. Condition (i) leads after straightforward computations to $-4(0.5-\gamma)^2(0.5-p)^2 \leqq 0$ and (since $\gamma = 0.5$ produces a 1-stage method) $p = 0.5$. With $p = 0.5$, $m_{11} = m_{22} = \gamma - 0.25$ and (ii) and (iii) lead to $\gamma \geqq 0.25$. □

Theorem 4.3 and Theorem 4.4 give immediately the following result.

COROLLARY 4.1. *The* 2-*stage* DIRKM'*s are AN-stable if* $p = 0.5$ *and* $\gamma \geqq 0.25$.

By the use of Theorem 4.1 and the established fact about the A-stability of the 2-stage DIRKM's the following result can easily be obtained.

COROLLARY 4.2. *The* 2-*stage* MDIRKM'*s are AN-stable if* $\gamma \geqq 0.25$.

Corollary 4.1 and Corollary 4.2 show that if $p \neq 0.5$ and $\gamma \geqq 0.25$ then the 2-stage MDIRKM is AN-stable, while the corresponding DIRKM is only A-stable.

A situation where the use of an A-stable method in the solution of nonautonomous equations causes instability is given below. Consider the equation

$$y' = A \sin^2\left(\frac{\pi t}{c} - 3.430251901\right) y, \qquad A < 0, \quad c > 0. \tag{4.11}$$

Assume that the 2-stage DIRKM given by

$$\begin{array}{c|cc} 1-\sqrt{2}/2 & 1-\sqrt{2}/2 & \\ 27\sqrt{2}/2-18 & 14\sqrt{2}-19 & 1-\sqrt{2}/2 \\ \hline & (53-5\sqrt{2})/62 & (9+5\sqrt{2})/62 \end{array} \tag{4.12}$$

is applied in the solution of (4.11) with a constant stepsize $h = c$. A simple analysis shows that (4.12) will be unstable if $-Ac > 37.1$. The numerical results obtained for $A = -10{,}000$, $h = c = 0.1$, $y(0) = 1{,}000$ are given in Table 3. The results obtained with two other methods (which are AN-stable) are also given in Table 3. These methods are:

$$\begin{array}{c|cc} 1 & 1 & \\ 0 & -1 & 1 \\ \hline & 0.5 & 0.5 \end{array} \qquad \begin{array}{cc|c} 1-\sqrt{2}/2 & 1-\sqrt{2}/2 & \\ \sqrt{2}/2 & \sqrt{2}-1 & 1-\sqrt{2}/2 \\ \hline & 0.5 & 0.5 \end{array} \tag{4.13}$$

TABLE 3
The errors found in the numerical integration of (4.11) (E1 *is the error found by* (4.12), E2 *and* E3 *are the errors for the first and the second method* (4.13), *respectively*).

t	Number of steps	E1	E2	E3
1.0	10	5.95E+5	5.93E 0	6.11E−17
2.0	20	3.54E+7	3.52E−3	3.73E−37
3.0	30	2.11E+9	2.09E−6	2.28E−57
4.0	40	1.25E+11	1.24E−9	3.26E−66
5.0	50	7.45E+12	7.35E−13	0.0

For the 2-stage DIRKM's (when implemented in the solution of (4.7)),

$$K(Z)=\frac{[1+(1-p-\gamma)z_1+(p-\gamma)z_2+(\gamma^2-2\gamma+0.5)z_1z_2]}{(1-\gamma z_1)(1-\gamma z_2)}, \tag{4.14}$$

where (when a constant stepsize is used)

$$z_i=hq(t_n-\alpha_i h), \qquad i=1,2. \tag{4.15}$$

For the method (4.12), $\gamma^2-2\gamma+0.5=0$ and the choice $h=c$ gives $z_2\approx 0$ at each step. Therefore it must be emphasized that the above example is very artificially created. Nevertheless, the example shows that the use of AN-stable methods in the solution of nonautonomous problems should be preferred; see also the second experiment in § 6.

Compare the results for the two methods (4.13). The second method follows the behavior of the solution much better. This shows that it seems to be useful to introduce LN-stability (L-stability for nonautonomous problems).

DEFINITION 4.1. The notation $|Z|\rightarrow\infty$ will be used to express the fact that $|z_i|\rightarrow\infty$ for all i.

DEFINITION 4.2. The RK method is said to be *LN-stable* if it is AN-stable and if $|Z|\rightarrow\infty$ implies $K(Z)\rightarrow 0$.

The relations between the different kinds of stability for the RK methods when they are applied to (4.1) and (4.7) are seen in Fig. 4.1.

$$\begin{tikzcd}
L\text{-stability} \arrow[r, Rightarrow] & A\text{-stability} \\
LN\text{-stability} \arrow[u, Rightarrow] \arrow[r, Rightarrow] & AN\text{-stability} \arrow[u, Rightarrow]
\end{tikzcd}$$

FIG. 4.1

The following corollaries can be formulated and proved.

COROLLARY 4.3. *For the* MDIRKM'*s LN-stability and L-stability are equivalent.*

COROLLARY 4.4. *The* 2-*stage* DIRKM'*s are LN-stable if* $p=0.5$ *and* $\gamma=1\pm\sqrt{2}/2$.

COROLLARY 4.5. *The* 2-*stage* MDIRKM'*s are LN-stable if* $\gamma=1\pm\sqrt{2}/2$.

As an example let us note that the MDIRKM corresponding to (4.12) is LN-stable, while (4.12) is not.

The results in this section show that not only a better performance of the QNIP for problems which have property $\bar{S}$ can be expected when the MDIRKM's are used (see § 3), but also, the classes of AN-stable and LN-stable 2-stage MDIRKM's are considerably larger than the classes of AN-stable and LN-stable 2-stage DIRKM's.

LN-stable implicit Runge–Kutta methods are also considered by Burrage [4]. This concept is extended for a wider class of methods in [6].

5. 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's of order 2 will be described in this section. The device is based on "embedding" which was first used by Fehlberg [14] for explicit RK methods. 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 5.2).

DEFINITION 5.1. Consider the problem defined by

$$y'=f(t,y), \qquad y(t_n)=y_n. \tag{5.1}$$

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

$$T^p_{n+1}=(\varphi_m^{(p+1)}(0)/(p+1)!)h^{p+1}, \qquad h=h_{n+1} \tag{5.2}$$

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

THEOREM 5.1. *Assume that y_{n+1} is computed by a 2-stage* MDIRKM *of order* 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 *under consideration,*

$$k_i(h)=f\left(t_n+\alpha_i h,\ y_n+h\sum_{j=1}^{m}\beta_{ij}k_j(h)\right), \qquad i=3(1)m, \tag{5.3}$$

$$\hat{y}_{n+1}=y_n+h\sum_{i=1}^{m}\hat{p}_i k_i(h). \tag{5.4}$$

Then if the terms which contain h^4 are neglected in (1.9) *the principal part of the local truncation error can be written in the following way*:

$$T^2_{n+1}=\hat{y}_{n+1}-y_{n+1}=h\sum_{i=1}^{2}(\hat{p}_i-p_i)k_i(h)+h\sum_{i=3}^{m}\hat{p}_i k_i(h). \tag{5.5}$$

The problem is how to choose the auxiliary method (5.3)–(5.4). It is not possible to construct an MDIRKM of order 3 (see § 3). It is not desirable to use implicit formulae in (5.3) (this may cause extra decompositions). Therefore the only choice which will ensure that the computational cost of the error estimator formulae (5.3)–(5.4) is $O(s^2)$ is $\beta_{ij}=0$ for $i=3(1)m$ and $j\geqq i$. By this choice the following theorem can be proved.

THEOREM 5.2. *The smallest number m which allows us to construct an error estimator* (5.3)–(5.4) *with explicit formulae* (5.3) *for a* 2-*stage* MDIRKM *is* 4.

Proof. The method (5.4) will be of order 3 if its coefficients satisfy the following conditions.

$$\sum_{i=1}^{m}\hat{p}_i=1, \tag{5.6}$$

$$\sum_{i=1}^{m}\hat{p}_i\alpha_i=0.5, \tag{5.7}$$

$$\sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\beta_{ij}=0.5, \tag{5.8}$$

$$\sum_{i=1}^{m}\hat{p}_i\alpha_i^2=\frac{1}{3}, \tag{5.9}$$

$$\sum_{i=1}^{m}\hat{p}_i\alpha_i\sum_{j=1}^{i}\beta_{ij}=\frac{1}{3}, \tag{5.10}$$

$$\sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\alpha_j\beta_{ij}=\frac{1}{6}, \tag{5.11}$$

$$\sum_{i=1}^{m}\hat{p}_i\sum_{j=1}^{i}\beta_{ij}\sum_{\nu=1}^{j}\beta_{j\nu}=\frac{1}{6}, \tag{5.12}$$

$$\sum_{i=1}^{m}\hat{p}_i\left(\sum_{j=1}^{i}\beta_{ij}\right)^2=\frac{1}{3}. \tag{5.13}$$

(a) Let us choose $m=3$. Then it is easily seen that the system (5.6)–(5.13) has no solution (consider (5.6), (5.7) and (5.9) and take into account that (3.5) and (3.9) must also be satisfied).

(b) Let us choose $m=4$. Assume that $\gamma, \beta_{21}, \beta_{31}, \beta_{32}, \alpha_3$ and α_4 are chosen so that (5.13) is satisfied. Then the solution of (5.6)–(5.12) can be found (for $\beta_{21}\neq 0$, $\alpha_3 \neq \alpha_4$, $\alpha_3 \neq 0.5$ and $\alpha_4 \neq 0.5$) by the use of the following formulae (and the notation $\bar{\alpha}=(2\alpha_4-1)/(2\alpha_3-1)$, $\bar{\beta}=\beta_{31}+\beta_{32}-\gamma$, $\bar{\gamma}=1-6\gamma+6\gamma^2$):

$$(5.14)\quad \beta_{42}=\{(\alpha_4-\alpha_3)[\bar{\gamma}(2\alpha_4-1)+\gamma]+\bar{\alpha}(\beta_{32}\beta_{21}-\bar{\beta}\gamma)+[\bar{\alpha}(\alpha_4-\alpha_3)-\gamma]\bar{\beta}\}/\beta_{21},$$

$$(5.15)\quad \beta_{43}=-\bar{\alpha}(\alpha_4-\alpha_3),$$

$$(5.16)\quad \beta_{41}=\alpha_4-\alpha_3-\beta_{42}-\beta_{43}+\beta_{31}+\beta_{32},$$

$$(5.17)\quad \hat{p}_4=\tfrac{1}{6}(2\alpha_4-1)(\alpha_4-\alpha_3),$$

$$(5.18)\quad \hat{p}_3=-\bar{\alpha}\hat{p}_4,$$

$$(5.19)\quad \hat{p}_2=[0.5-\gamma-(\hat{p}_3+\hat{p}_4)\bar{\beta}-\hat{p}_4(\alpha_4-\alpha_3)]/\beta_{21},$$

$$(5.20)\quad \hat{p}_1=1-\hat{p}_2-\hat{p}_3-\hat{p}_4.$$

Straightforward calculations show that (5.13) can be rewritten as

$$(5.21)\quad 2\bar{\beta}(2\alpha_3-1-\bar{\beta}+\beta_{21})=(2\alpha_3-1)\{[1+\bar{\gamma}-(3-6\gamma)\beta_{21}](2\alpha_4-1)-\alpha_4+\alpha_3+\beta_{21}\}.$$

It is easy to see that (5.21) can be satisfied (for example by the choice of $\beta_{32}=0$, $\beta_{31}=\gamma$, $\alpha_3=\alpha_4-\beta_{21}-[1+\bar{\gamma}-(3-6\gamma)\beta_{21}](2\alpha_4-1)$). □

Consider now the integration method as a combination of a basic 2-stage MDIRKM of order 2 and a 4-stage error estimator (5.3)–(5.4) of order 3. If (1.1) is linear, then the requirement (5.13) is not necessary (this requirement appears when the coefficient in the term containing f''_{yy} is equated to zero). In this case the six parameters $\gamma, \beta_{21}, \beta_{31}, \beta_{32}, \alpha_3$ and α_4 could be used in order to construct an integration method which is optimal with regard to some of the following requirements: *accuracy, stability, computational work per step and simple implementation.* If (1.1) is not linear then at least one of the parameters must be used in order to satisfy (5.13). The other parameters can again be used in an attempt to optimize the integration method.

6. Application of MDIRKM's in the solution of linear systems. Assume that: (i) *the* IVP1 *is linear,* (ii) *the* IVP1 *has property* $\bar{S}$ *or* S^*, (iii) ε *is large.* In this section we shall show that in this situation the QNIP can successfully be replaced by the use of GE and, moreover, if the use of GE is assumed, then it is preferable to apply MDIRKM's. Some numerical results obtained by Y12NBF will be used to illustrate our conclusions. Therefore, before the discussion, a brief description of this code is needed (some more details are given in [21]). The integration method implemented in the code is given by

$$(6.1)\quad \begin{array}{c|cccc}
0.5 & 1-\sqrt{2}/2 & & & \\
0.5 & \sqrt{2}-1 & 1-\sqrt{2}/2 & & \\
0 & 0 & 0 & 0 & \\
1 & -\sqrt{2}+1 & \sqrt{2}-1 & 1 & 0 \\
\hline
p_i & 0.5 & 0.5 & & \\
\hat{p}_i & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & \frac{1}{6}
\end{array}$$

By this choice of the parameters γ, β_{21}, β_{31}, β_{32}, α_3 and α_4, an attempt to construct an integration scheme which is optimal with regard to the computational work per step and to the simplicity of implementation has been carried out. Since the QNIP is replaced by GE, the most expensive parts of the computational work per successful step are one decomposition, two function calls and two back substitutions. The computational work needed to obtain the coefficient matrix for $k_1(i)$ and $k_2(i)$ is reduced from $O(s^2)$ to $O(4s)$ arithmetic operations by the use of matrix $\bar{A}=(1-\sqrt{2}/2)^{-1}h^{-1}A$ instead of $A=I-h(1-\sqrt{2}/2)A(t_n+h/2)$. Vector $k_3(h)$ is computed before the computation of $k_1(h)$ and $k_2(h)$. In this way it is not necessary to recompute $k_3(h)$ when the step is rejected and has to be repeated with a smaller stepsize. The step is not rejected immediately after obtaining $\|\hat{y}_{n+1}-y_{n+1}\|_2>\varepsilon$. First the code will attempt to perform extrapolation as follows. Vectors y_{n+2} and y^*_{n+2} are calculated using starting values y_{n+1} and y_n and stepsizes h and $2h$, respectively. Only the basic method (the 2-stage MDIRKM) is used in these calculations. The approximation y_{n+2} is accepted if $\|y_{n+2}-y^*_{n+2}\|_2/7\leqq\varepsilon$ and in the next several steps only this extrapolation rule is used. If the extrapolation rule fails, then the stepsize $h_{n+1}=h$ is reduced and y_{n+1} is recomputed by (6.1). During the use of the extrapolation rule, the code takes 2 small steps and 1 large step. When we count the steps we give the number of the small steps. The code uses 1.5 decompositions, 3 back substitutions and 1.5 function calls per successful small step when the extrapolation rule is used.

Now we are ready to present some numerical results. Two very simple but illustrative examples are given below.

Example 6.1. Consider the problem

$$(6.2)\qquad y'=A\sin^2\left(\frac{t}{40}\right)(y-0.01t)+0.01,\qquad y(0)=1,\quad t\in[0,100],\quad A<0.$$

The exact solution of the problem is $y(t)=0.01t+\exp[A(t/2-10\sin(t/20))]$. If $-A$ is small, then f and f'_y are not very quickly varying in t, but when $-A$ is large they are.

The results given in Table 4 show that the computational work in the integration process practically does not depend on the magnitude of parameter A.

Example 6.2. Let us consider a more stringent problem:

$$(6.3)\qquad y'=A\sin^2\left(\frac{t}{2}\right)(y-t)+1,\qquad y(0)=1,\quad t\in[0,100],\quad A<0.$$

TABLE 4

Numerical results obtained in the integration of (6.2). *Error tolerance* $\varepsilon=10^{-1}$.

$-A$	Steps	Decompositions	Function calls	Substitutions	Accuracy
0.5	15	15	31	30	$2.5E-2$
50	18	24	35	59	$5.0E-2$
500	18	27	32	54	$7.4E-2$
10^5	19	29	33	58	$5.1E-2$
10^{10}	17	27	29	54	$5.0E-2$

The exact solution of the problem is $y(t)=t+\exp[A(t/2-\sin t/2)]$. If $-A$ is large the functions f and f'_y are varying very quickly in t; moreover, f is also varying in y.

It is seen from Table 5 that the computational work increases when $-A$ becomes large, but not very fast.

TABLE 5
Numerical results obtained in the integration of (6.3). *Error tolerance* $\varepsilon = 10^{-1}$.

$-A$	Steps	Decompositions	Function calls	Substitutions	Accuracy
0.01	15	15	31	30	$2.2E-2$
50	28	42	48	84	$1.0E-1$
10^3	31	47	52	94	$1.1E-1$
10^6	39	57	66	114	$1.0E-1$

An MDIRKM (as implemented in Y12NBE) has also been used in the solution of some problems of chemical origin ([21], [22]) for which all assumptions made at the beginning of this section hold. The problems and some previous experiments concerning these chemical problems are described in [20].

An implementation of this MDIRKM for linear problems with large and sparse matrices $A(t)$ has also been developed [22]. The sparse matrix algorithm is based on ideas described in [24], [25], [26], [28]. Numerical examples, with s up to 255, are given in [22].

A code based on the use of MDIRKM's and designed for nonlinear problems which have property $\bar{S}$ is under preparation at RECKU (the Regional Computing Centre at the University of Copenhagen).

7. 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^* if the problem is linear) and when ε is large. If this is not so then the DIRKM's of order $p \geqq 2$ may perform better. If the error tolerance is stringent then the code STRIDE [7], [10], [11], which is based on singly-implicit Runge–Kutta methods ([3], see also [9]; these methods are derived by the use of a transformation proposed in [8]) implemented in a variable stepsize variable formula manner, will work much better than any MDIRKM (whose order cannot exceed 2). This means that the MDIRKM's *must be used carefully.* If the problem is large and the user can establish that the nonlinear problem which has to be solved has property $\bar{S}$, then the use of MDIRKM's will normally be efficient (and sometimes very efficient). Often the problem has property $\bar{S}$ only on a part of the integration interval. If this is so the MDIRKM's should be used in conjunction with some other methods (STRIDE). The use of MDIRKM's with linear problems which have property $\bar{S}$ and even S^* is also very efficient. If the problem is not large (say, $s \leqq 10$), if ε is large and if the QNIP is replaced by GE (as in Y12NBF), then the MDIRKM's will be efficient also for linear problems which have property S; the computational cost of the decompositions is not very large (in comparison with the computational cost for the back substitutions) and the extra decompositions will be compensated by a great reduction of the numbers of function calls and back substitutions. If the problem is linear and large, then matrix $A(t)$ is usually sparse and some sparse technique can easily be implemented, and the use of MDIRKM's is again efficient if the above conditions are satisfied. Note that large linear problems (1.1) arise often in practice (e.g., in the solution of some parabolic partial differential equations [27], or in chemistry [22], 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.

Acknowledgments. Section 4 has been written following the constructive comments on a previous version of this paper made by an unknown referee. The author would like to thank him very much for the helpful suggestions.

REFERENCES

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

[2] H. Bildsøe, J. P. Jacobsen and K. Schaumburg, *Application of density matrix formalism in* NMR *spectroscopy* I. *Development of a calculation scheme and some simple examples*, J. Magnet. Resonance, 23 (1976), pp. 137–151.

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

[4] ———, *Stability and efficiency properties of implicit Runge-Kutta methods*, Ph.D. Thesis, Mathematics Department, Auckland University, Auckland, New Zealand, 1978.

[5] K. Burrage and J. C. Butcher, *Stability criteria for implicit Runge-Kutta methods*, SIAM J. Numer. Anal., 16 (1979), pp. 46–57.

[6] ———, *Nonlinear stability of a general class of differential equation methods*, BIT, 20 (1980), pp. 185–203.

[7] K. Burrage, J. C. Butcher and F. H. Chipman, *An implementation of singly-implicit Runge-Kutta methods*, BIT, 20 (1980), pp. 326–340.

[8] J. C. Butcher, *On the implementation of implicit Runge-Kutta methods.* BIT, 16 (1976), pp. 237–240.

[9] ———, *A transformed implicit Runge-Kutta method*, J. Assoc. Comput. Mach., 26 (1979), pp. 731–738.

[10] J. C. Butcher, K. Burrage and F. H. Chipman, STRIDE-*stable Runge-Kutta integrator for differential equations*, Computational Mathematics Report, March 1979, Department of Mathematics, University of Auckland, Auckland, New Zealand.

[11] F. H. Chipman, *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 Urbana-Champaign, Urbana, IL, 1979.

[12] G. Dahlquist, *A special stability problem for linear multistep methods*, BIT, 3 (1963), pp. 27–43.

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

[14] E. Fehlberg, *Klassische Runge-Kutta-Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme*, Computing, 6 (1970), pp. 61–71.

[15] J. P. Jacobsen, H. K. Bildsøe and K. Schaumburg, *Application of density matrix formalism in* NMR *spectroscopy* II. *The one-spin-*1 *case in anisotropic phase.* J. Magnet. Resonance, 23 (1976), pp. 153–164.

[16] L. V. Kantorovich, *On integral equations*, Uspekhi Mat. Nauk, 11 (1956), pp. 3–29.

[17] L. V. Kantorovich and G. P. Akilov, *Functional Analysis in Normed Spaces*, Pergamon Press, Oxford, 1964.

[18] S. P. Nørsett, *Semiexplicit Runge-Kutta methods*, Mathematics and Computation, No. 6/74, Department of Mathematics, University of Trondheim, Norway.

[19] H. H. Robertson and J. Williams, *Some properties of algorithms for stiff differential equations*, J. Inst. Math. Appl., 16 (1975), pp. 23–34.

[20] K. Schaumburg and J. Wasniewski, *Use of a semiexplicit Runge-Kutta algorithm in a spectroscopic problem*, Comput. Chem., 2 (1978), pp. 19–25.

[21] K. Schaumburg, J. Wasniewski and Z. Zlatev, *Solution of ordinary differential equations with time dependent coefficients. Development of a semiexplicit Runge-Kutta algorithm and application to a spectroscopic problem*, Comput. Chem., 3 (1979), pp. 57–63.

[22] K. Schaumburg, J. Wasniewski and Z. Zlatev, *The use of sparse matrix technique in the numerical integration of stiff systems of linear ordinary differential equations*, Comput. Chem., 4 (1980), pp. 1–12.

[23] H. J. Stetter, *Analysis of discretization methods for ordinary differential methods.* Springer, Berlin, 1973.

[24] Z. Zlatev, *Use of iterative refinement in the solution of sparse linear systems*, Report 1/79, Institute of Mathematics and Statistics, The Royal Veterinary and Agricultural University, Copenhagen, 1979, SIAM J. Numer. Anal., to appear.

[25] ———, *On some pivotal strategies in Gaussian elimination by sparse technique*, SIAM J. Numer. Anal., 17 (1980), pp. 18–30.

[26] Z. Zlatev, K. Schaumburg and J. Wasniewski, *Implementation of an iterative refinement option in a code for large and sparse systems.* Comput. Chem., 4 (1980), pp. 87–99.

[27] Z. Zlatev and P. G. Thomsen, *Application of backward differentiation methods to the finite element solution of time dependent problems*, Int. J. Num. Meth. Engng, 14 (1979), pp. 1051–1061.

[28] Z. Zlatev, J. Wasniewski and K. Schaumburg, *Y12M-solution of large and sparse systems of linear algebraic equations* (*documentation of subroutines*) Lecture Notes in Computer Science, Springer, Berlin, 1981.