ELSEVIER

Journal of Computational and Applied Mathematics 87 (1997) 1–9

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

A note on the efficient implementation of implicit methods for ODEs[1]

Pierluigi Amodio[a], Luigi Brugnano[b,*]

[a] *Dipartimento di Matematica, Via Orabona 4, 70125 Bari, Italy*

[b] *Dipartimento di Matematica "U. Dini", Viale Morgagni 67/A, 50134 Firenze, Italy*

Received 3 June 1997; received in revised form 28 July 1997

Abstract

The use of implicit methods for ODEs, e.g. implicit Runge–Kutta schemes, requires the solution of nonlinear systems of algebraic equations of dimension $s \cdot m$, where m is the size of the continuous differential problem to be approximated. Usually, the solution of this system represents the most time-consuming section in the implementation of such methods. Consequently, the efficient solution of this section would improve their performance. In this paper, we propose a new iterative procedure to solve such equations on sequential computers.

Keywords: Implicit methods for ODEs; Modified Newton iteration; Splittings

AMS classification: 65L06; 65L05; 65H10; 65F30

1. Introduction

The single application of many implicit schemes for ODEs to the problem

$$y' = f(t, y), \qquad y(t_0) = y_0 \in \mathbb{R}^m, \tag{1}$$

results in the solution of a system of algebraic equations in the form

$$F(\boldsymbol{y}) := \boldsymbol{y} - h(C \otimes I_m)\boldsymbol{f} - \boldsymbol{g}(y_0) = \boldsymbol{0}, \tag{2}$$

where h is the stepsize, C is a full $s \times s$ matrix, I_m is the identity matrix of size m (when not specified, I will denote the identity matrix whose dimension will be clear from the context), $\boldsymbol{g}: \mathbb{R}^m \to \mathbb{R}^{s \cdot m}$ is a known function, and

$$\boldsymbol{y} = (y_1, \ldots, y_s)^{\mathrm{T}}, \qquad \boldsymbol{f} = (f_1, \ldots, f_s)^{\mathrm{T}}, \quad f_i = f(t_i, y_i),$$

* Corresponding author. E-mail: na.brugnano@na-net.ornl-gov.

[1] Work supported by M.U.R.S.T.

PII S0377-0427(97)00167-2

are the vectors with the unknown approximations to the solution at the grid-points $t_1, \dots, t_s$. This is the case, e.g., when an implicit s-stage Runge–Kutta method is used on problem (1). A different, more recent instance is given by the use of a block boundary value method (B_2VM) [1, 2] on the same problem, which would lead to a discrete problem in the form

$$(A \otimes I_m)y - h(B \otimes I_m)f + (a \otimes y_0) - h(b \otimes f_0) = 0,$$

where A and B are $s \times s$ matrices and a, b are given vectors. In particular, the matrix A is nonsingular and, therefore, multiplication on the left by $A^{-1} \otimes I_m$ produces a discrete problem in the form (2).

It is customary to use the modified Newton iteration to solve problem (2),

$$(I - hC \otimes J_0)\Delta y_i = -F(y_{i-1}), \quad i = 1, 2, \dots, \tag{3}$$

where J_0 is the Jacobian of the function f evaluated at (t_0, y_0). The use of the modified iteration has the obvious advantage of requiring the matrix $(I - hC \otimes J_0)$ to be factored only once, to carry out the iteration (3). If we do not consider the function and Jacobian evaluations, this implies that the leading term in the arithmetic complexity is given by $\frac{2}{3}(s \cdot m)^3$ flops, where we count as one *flop* one of the four basic *f*loating *op*erations with real quantities. The first relevant attempt to reduce this cost is due to Butcher [4], who essentially suggests to transform the matrix C into its Jordan form. This allows to lower the arithmetic complexity in the range between $\frac{2}{3}s \cdot m^3$ and $\frac{4}{3}s \cdot m^3$ flops, depending on the eigenvalues of C (eventually by solving complex systems as done, for example, in the implementation of the code RADAU5 [5]). We also mention that, in order to take full advantage from Butcher's procedure, Runge–Kutta methods with only one real eigenvalue have been proposed [8, 3]. However, such methods are less favorable than Runge–Kutta methods with complex eigenvalues, in terms of accuracy and stability.

A different approach is to use an *inner* iterative procedure for each *outer* iteration in (3). One of such inner–outer iteration schemes has been recently proposed in [6, 7], where they essentially consider the following procedure:

$$(I - hL \otimes J_0)\,\Delta y_i^{(j)} = (h(C - L) \otimes J_0)\,\Delta y_i^{(j-1)} - F(y_{i-1}), \quad j = 1, \dots, \mu, \tag{4}$$

instead of each step in (3). Here, μ is a suitable parameter and the matrix L is obtained by the LU factorization of the matrix C, where U has unit diagonal entries. In the following, we shall assume the matrix C to be nonsingular.

Considering that the iteration matrix in (4) is a matrix function of the Jacobian J_0, in order to study the asymptotic convergence rate of the inner iteration, it is sufficient to consider the convergence behavior in correspondence of each eigenvalue λ of J_0. Consequently, by setting $q = h\lambda$, it will be sufficient to study the spectral radius $\rho(q)$ of the *amplification matrix*

$$M(q) := q(I - qL)^{-1}(C - L) \equiv qL(I - qL)^{-1}(U - I). \tag{5}$$

According to van der Houwen and de Swart [7], the *region of convergence* of the inner iteration is given by

$$\Gamma = \{q \in \mathbb{C}: \rho(q) < 1\},$$

and the method is said to be *A-convergent* if $\mathbb{C}^- \subseteq \Gamma$.

A necessary condition for having A-convergence is the matrix L to have positive diagonal entries. In this case the eigenvalues of $M(q)$ are analytical functions of q in $\mathbb{C}^-$, and A-convergence is equivalent to require

$$\rho^* := \max_{x\in\mathbb{R}} \rho(\mathrm{i}x) \leqslant 1, \tag{6}$$

where, as usual, i is the imaginary unit.

A remarkable property of the inner iteration (4) is that

$$\rho(q) \to 0 \quad \text{as } q \to \infty. \tag{7}$$

In fact, from (5) it follows that, for $|q| \gg 0$,

$$M(q) \approx I - U,$$

which is a nilpotent matrix of order s (since U has unit diagonal entries). Consequently, the inner iteration is very suited for stiff problems. In the following section, we shall improve this approach.

2. Modification of the inner iteration

We observe that, for large m, the leading term in the arithmetic complexity of the procedure (4) is $\frac{2}{3}s \cdot m^3$ flops. It is due to the factorization of the $m \times m$ matrices

$$I_m - h\ell_i J_0, \quad i = 1,\ldots,s,$$

where ℓ_i is the ith diagonal entry of the matrix L. This may not be a severe limitation if the algorithm is implemented in parallel on s processors, since all the above factorizations are independent of each other [6, 7], but it would result in a less competitive algorithm on a sequential computer. However, if all the entries ℓ_i are equal, then only one of the above factorizations is needed and, consequently, the leading term in the arithmetic complexity reduces to $\frac{2}{3}m^3$. This is exactly what we plan to achieve for many methods of practical interest.

For this purpose, the next result is useful.

Theorem 1. *Let $C \in \mathbb{R}^{s\times s}$, $\det(C) > 0$. Then there exists a transformation matrix T such that*

$$TCT^{-1} = L\,U, \tag{8}$$

where U is upper triangular with unit diagonal entries and L is lower triangular with the elements on the main diagonal all equal to

$$\ell = \sqrt[s]{\det(C)}. \tag{9}$$

Proof. The proof is obtained recursively, by using suitable elementary matrices. We shall take such matrices upper triangular, but they could also be chosen differently. To begin with, let us consider the matrix

$$T_1 = I - \boldsymbol{e}_1 \boldsymbol{u}_1^{\mathrm{T}},$$

where, in general,
- e_i is the ith unit vector in $\mathbb{R}^s$, and
- $u_i \in \mathbb{R}^s$ has the first i entries equal to zero,

such that (see (9))

$$T_1 C T_1^{-1} = \begin{pmatrix} \ell & z_1^{\mathrm{T}} \\ x_1 & C_1 \end{pmatrix}. \tag{10}$$

If $C=(c_{ij})$, and $c=(c_{11},\dots,c_{s1})^{\mathrm{T}}$ is its first column, one then obtains that u_1 is any vector such that

$$c_{11} - u_1^{\mathrm{T}} c = \ell.$$

For simplicity, we suppose that $c_{21},\dots,c_{s1}$ are not all equal to zero, even though the whole procedure can be modified to handle this case, too. The next step is to compute the first elementary Gauss matrix L_1 corresponding to the matrix (10),

$$L_1 = I - g_1 e_1^{\mathrm{T}}, \quad g_1 = \frac{1}{\ell}\begin{pmatrix} 0 \\ x_1 \end{pmatrix},$$

from which

$$(L_1 T_1) C T_1^{-1} = \begin{pmatrix} \ell & z_1^{\mathrm{T}} \\ \mathbf{0} & \hat{C}_1 \end{pmatrix}.$$

The same procedure is then repeated on the matrix $\hat{C}_1$, and so on, until the following factorization is obtained:

$$\hat{U} \equiv \begin{pmatrix} \ell & * & \dots & * \\ & \ell & \ddots & \vdots \\ & & \ddots & * \\ & & & \ell \end{pmatrix} = (L_{s-1} T_{s-1} \cdots L_1 T_1) C (T_{s-1} \cdots T_1)^{-1}$$

$$= (\hat{L}_{s-1} \cdots \hat{L}_1)(T_{s-1} \cdots T_1) C (T_{s-1} \cdots T_1)^{-1}, \tag{11}$$

where, for all i,

$$T_i = I - e_i u_i^{\mathrm{T}}, \quad L_i = I - g_i e_i^{\mathrm{T}}, \tag{12}$$

and the first i entries of the vector g_i are zero. The key point is that the matrices $\hat{L}_i$, obtained by

$$\left(\prod_{j>i} T_j\right) L_i = \left(I - \sum_{j>i} e_j u_j^{\mathrm{T}}\right)(I - g_i e_i^{\mathrm{T}})$$

$$= I - \sum_{j>i} e_j u_j^{\mathrm{T}} - g_i e_i^{\mathrm{T}} + \sum_{j>i} e_j (u_j^{\mathrm{T}} g_i) e_i^{\mathrm{T}}$$

$$= I - \sum_{j>i} e_j u_j^{\mathrm{T}} - \left(g_i - \sum_{j>i} e_j (u_j^{\mathrm{T}} g_i)\right) e_i^{\mathrm{T}}$$

$$\equiv I - \sum_{j>i} e_j u_j^{\mathrm{T}} - \hat{g}_i e_i^{\mathrm{T}}$$

$$= (I - \hat{\boldsymbol{g}}_i \boldsymbol{e}_i^{\mathrm{T}}) \left(I - \sum_{j>i} \boldsymbol{e}_j \boldsymbol{u}_j^{\mathrm{T}} \right)$$

$$\equiv \hat{L}_i \left(\prod_{j>i} T_j \right)$$

are still elementary Gauss matrices. As a consequence, by setting

$$\hat{L}^{-1} = (\hat{L}_{s-1} \cdots \hat{L}_1), \qquad T = (T_{s-1} \cdots T_1),$$

from (11) we get the LU factorization of the matrix $\hat{C} = TCT^{-1}$. The thesis follows considering that the factors of the factorization (8) are $L = \ell \hat{L}$ and $U = (1/\ell)\hat{U}$. □

We observe that in most cases of interest the matrices T_i in (12) can be chosen so that

$$\boldsymbol{u}_i = \alpha_i \boldsymbol{e}_{i+1}, \quad \alpha_i \in \mathbb{R},$$

namely, the matrix T in Theorem 1 is upper bidiagonal,

$$T = \begin{pmatrix} 1 & \alpha_1 & & \\ & \ddots & \ddots & \\ & & 1 & \alpha_{s-1} \\ & & & 1 \end{pmatrix}. \tag{13}$$

This is indeed true for all methods considered in Section 3. In this case, the matrix T can be easily obtained by using the Matlab function `makeT` reported in the Appendix.

We observe that the use of the presented technique requires, at each outer iteration, the variable transformation

$$z = (T \otimes I_m) y,$$

whose complexity is $O(s^2 \cdot m)$ flops in general, and $2s \cdot m$ flops when T is upper bidiagonal. Consequently, this cost can be considered negligible with respect to the other operations involved, when $m \gg 1$.

Remark 1. The result of Theorem 1 can be generalized to the case $\det(C) \neq 0$, if we allow the diagonal entries of the matrix U to be either 1 or -1. This is done by replacing (9) with

$$\ell = \sqrt[s]{|\det(C)|}.$$

Moreover, the result can be easily adapted to handle the case of Runge–Kutta schemes having the matrix of the tableau with a zero first row (e.g. Lobatto IIIA schemes), or in the case of methods having the last column with zero entries (e.g. Lobatto IIIB schemes). In the former case, in fact, one applies the above procedure to the matrix obtained by discarding the first row and the first column, while in the latter case the procedure is applied to the matrix obtained by neglecting the first row and the last column.

By using the factorization (8), it follows that the iteration (4) is transformed to

$$(I - hL \otimes J_0)\Delta z_i^{(j)} = (h(TCT^{-1} - L) \otimes J_0)\Delta z_i^{(j-1)} - (T \otimes I_m)F(y_{i-1}), \quad j = 1, \ldots, \mu,$$
$$\Delta y_i^{(\mu)} = (T^{-1} \otimes I_m)\Delta z_i^{(\mu)}. \tag{14}$$

As a consequence, the new amplification matrix is given by

$$M_T(q) := q(I - qT^{-1}LT)^{-1}(C - T^{-1}LT) \equiv T^{-1}qL(I - qL)^{-1}(U - I)T. \tag{15}$$

We observe that the property (7) continues to hold for this matrix.

It is evident that the spectral properties of such matrix, having fixed the matrix C, will depend on the matrix T. In the following section, we shall compare the amplification matrices (5) and (15). In fact, in order that the presented technique be competitive with that proposed in [6], it is sufficient to show that the convergence properties of the matrix (15) are not worse than those of the matrix (5).

3. Comparison of amplification factors

In this section we compare the properties of the amplification matrices (5) and (15) corresponding to some Runge–Kutta methods. In Table 1 we report the parameter defined in (6) for the matrix (5) and the corresponding parameter for the matrix (15). By comparing the values in each row of the table, it seems that the iteration (14) is at least as effective as the iteration (4). However, the parameter ρ^* is significant to describe the asymptotic behavior of powers of the matrices $M(q)$ and $M_T(q)$, whereas, when the iterations (4) or (14) are used, it would be preferable to take μ, the number of inner iterations, as small as possible, in order to reduce the arithmetic complexity of the corresponding outer iteration. Consequently, in place of the parameter (6), which could not be able to describe the behavior of the amplification matrices when the number of inner iterations is small, the following parameters are also considered for the matrix $M(q)$ [6] (hereafter, the norm used is the infinity norm),

Table 1
Parameter (6) for the matrices (5) and (15) corresponding to some Runge–Kutta schemes

		ρ^*	
Method	s	M	M_T
Gauss	2	0.14	0.13
	3	0.30	0.25
Radau IA	3	0.46	0.31
Radau IIA	3	0.37	0.31
Lobatto IIIA	3	0.14	0.13
	4	0.30	0.25
Lobatto IIIB	3	0.25	0.13
	4	0.41	0.25
Lobatto IIIC	2	0.33	0.29
	3	0.54	0.41

Table 2
Parameters (16)–(18) for the matrices (5) and (15) (for various values of μ), corresponding to the Runge–Kutta schemes in Table 1

		ρ_1^*		ρ_2^*		ρ_3^*	
Method	s	M	M_T	M	M_T	M	M_T
Gauss	2	0.15	0.14	0.14	0.13	0.14	0.13
	3	0.38	0.31	0.33	0.28	0.32	0.27
Radau IA	3	2.45	0.95	0.85	0.43	0.65	0.37
Radau IIA	3	0.45	0.37	0.40	0.34	0.39	0.33
Lobatto IIIA	3	0.14	0.13	0.14	0.13	0.14	0.13
	4	0.38	0.33	0.33	0.28	0.32	0.27
Lobatto IIIB	3	1.00	0.46	0.40	0.21	0.33	0.17
	4	2.24	0.77	0.79	0.36	0.60	0.30
Lobatto IIIC	2	1.00	0.83	0.48	0.41	0.41	0.36
	3	3.00	1.32	1.00	0.57	0.76	0.49
		$\tilde{\rho}_1$		$\tilde{\rho}_2$		$\tilde{\rho}_3$	
Method	s	M	M_T	M	M_T	M	M_T
Gauss	2	0.08	0.08	0.08	0.08	0.08	0.08
	3	0.13	0.10	0.12	0.10	0.12	0.09
Radau IA	3	0.30	0.17	0.28	0.15	0.27	0.15
Radau IIA	3	0.21	0.16	0.20	0.15	0.20	0.15
Lobatto IIIA	3	0.08	0.08	0.08	0.08	0.08	0.08
	4	0.13	0.10	0.13	0.10	0.12	0.09
Lobatto IIIB	3	0.17	0.08	0.17	0.08	0.17	0.08
	4	0.22	0.11	0.20	0.10	0.20	0.10
Lobatto IIIC	2	0.50	0.41	0.50	0.41	0.50	0.41
	3	0.50	0.29	0.47	0.27	0.46	0.26
		$\rho_1^{(\infty)}$		$\rho_2^{(\infty)}$		$\rho_3^{(\infty)}$	
Method	s	M	M_T	M	M_T	M	M_T
Gauss	2	0.15	0.14	0	0	0	0
	3	0.33	0.21	0.19	0.15	0	0
Radau IA	3	2.45	0.95	0.68	0.37	0	0
Radau IIA	3	0.45	0.33	0.26	0.21	0	0
Lobatto IIIA	3	0.13	0.12	0	0	0	0
	4	0.23	0.21	0.17	0.14	0	0
Lobatto IIIB	3	1.00	0.46	0	0	0	0
	4	2.24	0.77	0.62	0.30	0	0
Lobatto IIIC	2	1.00	0.83	0.50	0	0	0
	3	3.00	1.32	0.82	0.50	0	0

- the *averaged amplification factor*,

$$\rho_\mu^* = \max_{x\in\mathbb{R}} \sqrt[\mu]{\|M^\mu(\mathrm{i}x)\|}, \quad \mu = 1,2,\ldots, \tag{16}$$

which is always greater than ρ^*. Obviously, $\rho_\mu^* \to \rho^*$ as $\mu \to \infty$;

- the *nonstiff averaged amplification factor*,

$$\tilde{\rho}_\mu = \sqrt[\mu]{\|(C-L)^\mu\|}, \tag{17}$$

which is useful when $q \approx 0$ since in this case $M(q) \approx q(C-L)$;

- the *stiff averaged amplification factor*

$$\rho_\mu^{(\infty)} = \lim_{x\to\infty} \sqrt[\mu]{\|M^\mu(\mathrm{i}x)\|}, \tag{18}$$

which may not be zero, for $\mu < s$.

Obviously, when replacing the matrix $M(q)$ with $M_T(q)$, the above quantities become, respectively,

$$\rho_\mu^* = \max_{x\in\mathbb{R}} \sqrt[\mu]{\|M_T^\mu(\mathrm{i}x)\|},$$

$$\tilde{\rho}_\mu = \sqrt[\mu]{\|(C - T^{-1}LT)^\mu\|},$$

$$\rho_\mu^{(\infty)} = \lim_{x\to\infty} \sqrt[\mu]{\|M_T^\mu(\mathrm{i}x)\|}, \quad \mu = 1,2,\ldots$$

In Table 2 we list the above parameters for the same methods considered in Table 1, for both matrices (5) and (15). In all cases, it turns out that the iterative procedure (14), based on the transformation matrix T chosen accordingly to what stated in Theorem 1, is better than the original iteration (4). Consequently, the new iteration will provide an improvement over that described in [6], at least on sequential computers.

Acknowledgements

The authors are very indebted to Professor Donato Trigiante for the stimulating discussions, and to the reviewers for their helpful comments.

Appendix

```
function T = makeT( C, detC )
%
% T = makeT( C ), Computes the transformation matrix (13)
%                 when C is nonsingular.
%
s = max( size( C ) );
if nargin==1, detC = det(C)^(1/s); end
T = eye( s );
```

```
if s>1
    T(1,2) = -(C(1,1)-detC)/C(2,1);
    T1 = T; T1(1,2) = -T1(1,2);
    C = T*C*T1; C(2:s,:) = C(2:s,:)-C(2:s,1)*C(1,:)/C(1,1);
    T(2:s,2:s) = makeT( C(2:s,2:s), detC );
end
return
```

References

[1] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon & Breach, London, 1997.

[2] L. Brugnano, D. Trigiante, On the potentiality of sequential and parallel codes based on extended trapezoidal rules (ETRs), Appl. Numer. Math. 25 (1997) 1–16.

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

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

[5] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential–Algebraic Equations, Springer Series in Computational Mathematics, vol. 14, Springer, Berlin, 1991.

[6] P.J. van der Houwen, J.J.B. de Swart, Triangularly implicit iteration methods for ODE-IVP solvers, SIAM J. Sci. Comput. 18 (1997) 41–55.

[7] P.J. van der Houwen, J.J.B. de Swart, Parallel linear system solvers for Runge–Kutta methods, Adv. Comput. Math. 7 (1–2) (1997) 157–181.

[8] S.P. Nørsett, Runge–Kutta methods with a multiple real eigenvalue only, BIT 16 (1976) 388–393.