ELSEVIER

Journal of Computational and Applied Mathematics 54 (1994) 121–127

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Letter Section

A parallel DIRK method for stiff initial-value problems

Nguyen huu Cong [a,b,*]

[a] *Afdeling Numerieke Wiskunde, Centre for Mathematics and Computer Science, P.O. Box 94079, 1090 GB Amsterdam, Netherlands*

[b] *Faculty of Mathematics, Mechanics and Informatics, University of Hanoi, Thuong Dinh, Dong Da, Hanoi, Viet Nam*

Received 10 March 1994; revised 16 May 1994

Abstract

In this note we propose a fast parallel iteration process for solving a low-order implicit Runge–Kutta method. The resulting scheme can be regarded as a parallel singly diagonally implicit Runge–Kutta (PDIRK) method. On a two-processor computer, this method requires effectively the solution of two implicit relations per step. By two numerical experiments we compare this method with some sequential methods from the literature, and show its efficient behaviour.

Keywords: Runge–Kutta methods; Predictor–corrector methods; Parallelism

1. Introduction

Consider the stiff initial-value problem (IVP) for the system of first-order ordinary differential equations (ODEs)

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = f(y(t)), \quad y(t_0) = y_0, \ t_0 \leqslant t \leqslant T, \ y : \mathbb{R} \to \mathbb{R}^N, \ f : \mathbb{R}^N \to \mathbb{R}^N. \tag{1.1}$$

In this note we will concentrate on ODEs which have to be solved with low-accuracy demands (such as partial differential equations). Among the suitable methods for this purpose, the class of DIRK methods seems to be very attractive, since they combine good stability properties with

* *Correspondence to:* Faculty of Mathematics, Mechanics and Informatics, University of Hanoi, Thuong Dinh, Dong Da, Hanoi, Viet Nam. These investigations were supported by the University of Amsterdam who provided the author with a research grant for spending a total of two years at CWI (Centre for Mathematics and Computer Science) in Amsterdam.

SSDI 0377-0427(94)00068-9

low computational costs. The only disadvantage of these methods is their low stage order (usually equal to 1), causing order reduction in many stiff problems (cf. [2]). In [7], van der Houwen and Sommeijer proposed parallel iterated Runge–Kutta (PDIRK) methods for the parallel numerical integration of (1.1). These methods have a high(er) stage order but need quite a number of iterations (i.e., implicit stages) in order to be unconditionally stable. These methods are able to produce accurate results, but at a relatively high price. Therefore, they seem to be not the most suitable candidates to serve the present purpose: to generate low-accuracy results at a low price, using an unconditionally stable method.

In this note we construct a PDIRK method based on the concepts used in [7]. In that paper the convergence behaviour of these PDIRK methods was characterized by the magnitude of the iteration error function in the left half-plane (see also [8]). The PDIRK method in the present paper possesses step point order and stage order equal to 2. The parameters of the PDIRK method are chosen such that this method is L-stable and its iteration error function vanishes in the whole left half-plane. The resulting PDIRK method can be regarded as an L-stable singly diagonally implicit Runge–Kutta (SDIRK) method with four implicit stages. However, on two processors, only two sequential implicit stages per step are required. We expect this method to be an efficient integrator in the low-accuracy range. By means of two numerical experiments we will compare the efficiency of the method constructed in this note with that of a number of sequential DIRK methods from the literature and with the code LSODE (in which we restricted the order to 2).

2. PDIRK method

Starting with an s-stage implicit Runge–Kutta (IRK) method

$$\boldsymbol{Y}_n = \boldsymbol{e} \otimes y_n + h[A \otimes I_N]\boldsymbol{F}(\boldsymbol{Y}_n), \quad y_{n+1} = y_n + h[\boldsymbol{b}^{\mathrm{T}} \otimes I_N]\boldsymbol{F}(\boldsymbol{Y}_n),$$

which is referred to as the *corrector method,* we consider a parallel diagonally implicit iteration process of the form

$$\boldsymbol{Y}_n^{(0)} = \boldsymbol{e} \otimes y_n, \tag{2.1a}$$

$$\boldsymbol{Y}_n^{(j)} - h[D \otimes I_N]\boldsymbol{F}(\boldsymbol{Y}_n^{(j)}) = \boldsymbol{e} \otimes y_n + h[(A - D) \otimes I_N]\boldsymbol{F}(\boldsymbol{Y}_n^{(j-1)}), \quad j = 1, 2, \ldots, m, \tag{2.1b}$$

$$y_{n+1} = y_n + h[\boldsymbol{b}^{\mathrm{T}} \otimes I_N]\boldsymbol{F}(\boldsymbol{Y}_n^{(m)}). \tag{2.1c}$$

Here, the s-dimensional matrix D is of diagonal form with fixed positive diagonal entries. Owing to this form, the s components of each iterate $\boldsymbol{Y}_n^{(j)}$ can be solved in parallel. Hence, on an s-processor machine, (2.1) requires effectively the solution of m implicit relations per step. The freedom in the matrix D will be used to obtain fast convergence, i.e., to minimize the value of m.

Methods of the form (2.1) were first proposed in [7], and can be considered as DIRK methods. An extension to second-order ODEs can be found in [4,9].

2.1. The iteration error function

The iteration error of the process (2.1b) is studied on the basis of the model test equation $y'(t) = \lambda y(t)$, where λ runs through the eigenvalues of the Jacobian matrix $\partial f/\partial y$ (cf. [7,8]). For this equation we obtain the iteration error equation

$$\begin{aligned} &\boldsymbol{Y}_n - \boldsymbol{Y}_n^{(j)} = [Z(z)]^j [\boldsymbol{Y}_n - \boldsymbol{Y}_n^{(0)}], \quad j = 1, 2, \ldots, m, \\ &Z(z) := zD[I - zD]^{-1}[D^{-1}A - I], \quad z := \lambda h. \end{aligned} \tag{2.2}$$

The matrix $Z(z)$ is called the *iteration error matrix,* and its spectral radius $\rho(Z(z))$ the *iteration error function*. The region of convergence is defined by

$$\boldsymbol{C} = \{z: \rho(Z(z)) < 1, \operatorname{Re}(z) \leqslant 0\}. \tag{2.3}$$

In this note we will consider a diagonal matrix D with *constant* diagonal entries δ, i.e., $D = \delta I$. Then, the iteration error function reduces to the form

$$\rho(Z(z)) = \frac{|z\delta|}{|1 - z\delta|}\rho(\delta^{-1}A - I). \tag{2.4}$$

In the next subsection we will construct an L-stable PDIRK method with vanishing iteration error function $\rho(Z(z))$.

2.2. Construction of the PDIRK method

In this note we will restrict our considerations to the case $s = 2$ and we will first construct a suitable IRK method which can serve as the corrector.

For that purpose we will use the collocation principle and impose the simplifying conditions $B(2)$ and $C(2)$ (cf., e.g., [2,3]). Based on the collocation vector $\boldsymbol{c} = (\alpha, 1)^{\mathrm{T}}$, we find the corrector given by the Butcher array

$$\begin{array}{c|cc} \alpha & \dfrac{\alpha(2-\alpha)}{2(1-\alpha)} & \dfrac{\alpha^2}{2(\alpha-1)} \\ 1 & \dfrac{1}{2(1-\alpha)} & \dfrac{1-2\alpha}{2(1-\alpha)} \\ \hline & \dfrac{1}{2(1-\alpha)} & \dfrac{1-2\alpha}{2(1-\alpha)} \end{array} \tag{2.5}$$

where α is still a free parameter which will be used to obtain good convergence. Concerning the stability of this corrector method, we have the following theorem.

Theorem 2.1. *The two-stage implicit Runge–Kutta method defined by* (2.5) *is L-stable for any positive real-valued* α.

Proof. The stability function $R_{\text{corr}}(z)$ of the IRK method (2.5) is defined by (cf., e.g., [2,3])

$$R_{\text{corr}}(z) = 1 + z\boldsymbol{b}^{\mathrm{T}}(I - zA)^{-1}\boldsymbol{e} = \frac{\det(I - zA + z\boldsymbol{e}\boldsymbol{b}^{\mathrm{T}})}{\det(I - zA)}. \tag{2.6a}$$

A simple calculation leads to

$$R_{\text{corr}}(z) = \frac{2 + (1-\alpha)z}{2 - (1+\alpha)z + \alpha z^2}. \tag{2.6b}$$

Along the imaginary axis ($z = \mathrm{i}y$) we have

$$|R_{\text{corr}}(\mathrm{i}y)|^2 = \frac{4 + y^2(1-\alpha)^2}{(2 - \alpha y^2)^2 + y^2(1+\alpha)^2} \leqslant 1, \quad \text{for all real-valued } y \text{ and } \alpha.$$

Since $R_{\text{corr}}(z)$ is an analytical function of the left half-plane (its two poles are easily seen to lie in the right half-plane for all positive α), we may conclude that the IRK method (2.5) is L-stable for all positive α. □

Now we can exploit the freedom on the choice of the two parameters α and δ to minimize the iteration error function of the PDIRK method (2.1). For this purpose we will choose α and δ such that $\rho(Z(\infty)) = \rho(\delta^{-1}A - I) = 0$. This requirement leads to the condition

$$\delta = \tfrac{1}{4}(\alpha + 1), \quad \alpha = 3 \pm 2\sqrt{2}. \tag{2.7}$$

By means of (2.4) we have $\rho(Z(z)) \equiv 0$ for all z lying in the left half-plane. Thus, we have the following theorem.

Theorem 2.2. *The iteration error function of the PDIRK method defined by* (2.1), (2.5), *and* (2.7) *is identical to zero in the whole left half-plane* $\{z \in \mathbb{C}: \mathrm{Re}(z) \leqslant 0\}$.

Notice that in condition (2.7) we have two positive values of α, but it is natural to choose the smaller value $\alpha = 3 - 2\sqrt{2}$.

Theorem 2.2 implies that $[Z(z)]^2 \equiv O$ (zero matrix), in the whole left half-plane; consequently, with respect to the model equation, the PDIRK method (2.1), (2.5), (2.7) with $m \geqslant 2$ produces exactly the corrector solution.

Applying the PDIRK method to the model test equation $y'(t) = \lambda y(t)$, we obtain the recursion (cf. [7])

$$y_{n+1} = R_m(z)y_n, \qquad R_m(z) = R_{\text{corr}}(z) - z^2\boldsymbol{b}^{\mathrm{T}}[Z(z)]^m[I - zA]^{-1}\boldsymbol{c}, \tag{2.8}$$

where $R_{\text{corr}}(z)$ is the stability function of the RK corrector method (2.5) defined by (2.6). Hence, (2.8) together with $[Z(z)]^2 \equiv O$ implies that $R_2(z) = R_{\text{corr}}(z)$. Thus, we have the following corollary.

Corollary 2.3. *The PDIRK method defined by* (2.1), (2.5) *and* (2.7) *with* $m = 2$ *is L-stable and has step point and stage order both equal to* 2.

3. Numerical experiments

In this section we will report the numerical results obtained by the PDIRK method defined by (2.1), (2.5) and (2.7) with $m = 2$ (PDIRK$_2$ method) and by a number of sequential DIRK methods from the literature. We selected the following sequential DIRK methods:

Nørsett$_3$ and Nørsett$_4$: third- and fourth-order A-stable methods of Nørsett (cf. [5]);
HW$_4$: fourth-order L-stable method of Hairer and Wanner (cf. [3, p.107]);
CS$_5$: fifth-order A-stable method of Cooper and Sayfy (cf. [1]).

The computations were performed using fifteen-digits arithmetic. The accuracy is given by the number of correct digits NCD, obtained by writing the maximum norm of the error at the end of the integration interval in the form $10^{-\mathrm{NCD}}$. The sequential computational effort is measured by the number of sequential stages per unit interval. The (fixed) stepsize h is chosen such that the number of sequential stages per unit interval equals a prescribed number M (cf. [9]). In the tables of results, s^* denotes the effective number of stages per step, required by the various methods.

3.1. Prothero–Robinson-type problem

Our first example is the difficult system of (uncoupled) Prothero–Robinson-type equations (see [9])

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = J[y(t) - g(t)] + g'(t), \quad y(0) = g(0),$$
$$J = \mathrm{diag}(-10^{2(j-1)}), \quad g(t) = (1 + \sin(jt)), \qquad j = 1, \ldots, 6, \quad 0 \leqslant t \leqslant 20. \tag{3.1}$$

Prothero and Robinson [6] used a problem of this type to show the order reduction of RK methods. The exact solution of (3.1) is given by $y(t) = g(t)$ which has slowly (nonstiff) and rapidly (stiff) varying components. Table 3.1 lists the numerical results. For this problem, all methods show a second-order behaviour. Hence, for the DIRK methods, order reduction really occurs. The superiority of the PDIRK$_2$ method over all sequential DIRK methods used in this experiment is clearly demonstrated.

Table 3.1
Values of NCD and M for problem (3.1)

Methods	Order	s^*	$M = 60$	$M = 120$	$M = 240$	$M = 480$	$M = 960$
Nørsett$_3$	3	2	2.7	3.3	3.9	4.5	5.1
Nørsett$_4$	4	3	2.5	3.1	3.7	4.3	4.9
HW$_4$	4	5	3.6	4.5	5.5	6.0	6.3
CS$_5$	5	5	1.8	2.2	2.6	2.9	3.3
PDIRK$_2$	2	2	4.5	5.1	5.7	6.3	6.9

Table 3.2
Values of NCD and M for problem (3.2)

Methods	Order	s^*	$M=30$	$M=60$	$M=120$	$M=240$
Nørsett_3	3	2	3.8	4.4	5.1	5.7
Nørsett_4	4	3	3.5	4.1	4.8	5.4
HW_4	4	5	3.8	4.5	5.1	5.8
CS_5	5	5	2.6	3.2	4.0	5.0
PDIRK_2	2	2	4.7	5.3	5.9	6.6

3.2. Nonlinear partial differential equation

In order to show the performance of the PDIRK_2 method on problems to be solved with low accuracy demand, we consider the convection–diffusion problem (see [7])

$$\frac{\partial u(t,x)}{\partial t} = u(t,x)\frac{\partial^2 u(t,x)}{\partial x^2} - x\cos(t)\frac{\partial u(t,x)}{\partial x} - x^2\sin(t), \quad 0 \leqslant x \leqslant 1,\ 0 \leqslant t \leqslant 1. \quad (3.2)$$

The initial and Dirichlet boundary conditions are such that the exact solution is given by $u(t,x)=x^2\cos(t)$. Standard finite-difference discretization of the spatial derivatives on a uniform grid with meshsize $\frac{1}{40}$ leads to a system of 39 ODEs with exact solution $(\frac{1}{40}j)^2\cos(t)$, $j=1,\ldots,39$. Table 3.2 is the analogue of Table 3.1. Again, order reduction is shown for the DIRK methods, and PDIRK_2 turns out to be the most efficient method.

3.3. Comparison with the code LSODE

Next we compare PDIRK_2 with a BDF method which has in common with all the previous methods that only LU-decompositions are required of matrices with dimension N (i.e., the ODE dimension). For that purpose we applied to problem (3.2) the famous stiff code LSODE, in which we limited the order to 2 (to obtain L-stability). Table 3.3 compares the performances of LSODE and PDIRK_2. In spite of the fact that LSODE used a variable-stepsize strategy, it needed approximately twice as many integration steps (N_{steps}) to obtain the same accuracy (defined by NCD). As a consequence, the total number of implicit relations that has to be solved over the whole integration interval is about the same for both methods. However, the most expensive part in both algorithms is the LU-factorization of the Newton-iteration matrix, which has to be done after every change in the stepsize or update of the Jacobian. Since LSODE needs the double amount of steps, it is very likely that this aspect is in favour of the PDIRK_2 method. Hence, in the low-accuracy range, PDIRK_2 seems to be at least competitive with LSODE.

Table 3.3
Values of $\text{NCD}/N_{\text{steps}}$ for problem (3.2) obtained by LSODE and PDIRK_2

LSODE	3.6/13	4.0/18	4.6/34	5.2/67	5.8/136
PDIRK_2	3.7/5	4.0/7	4.6/14	5.3/28	5.9/56

4. Summary

We have constructed an L-stable six-stage SDIRK method which can be specified by the butcher array

0	0					
0	0	0				
$3-2\sqrt{2}$	$\frac{1}{4}(1-\sqrt{2})$	$\frac{1}{4}(7-5\sqrt{2})$	$\frac{1}{2}(2-\sqrt{2})$			
1	$\frac{1}{4}(1+\sqrt{2})$	$\frac{1}{4}(-1+\sqrt{2})$	0	$\frac{1}{2}(2-\sqrt{2})$		
$3-2\sqrt{2}$	0	0	$\frac{1}{4}(1-\sqrt{2})$	$\frac{1}{4}(7-5\sqrt{2})$	$\frac{1}{2}(2-\sqrt{2})$	
1	0	0	$\frac{1}{4}(1+\sqrt{2})$	$\frac{1}{4}(-1+\sqrt{2})$	0	$\frac{1}{2}(2-\sqrt{2})$
	0	0	0	0	$\frac{1}{4}(1+\sqrt{2})$	$\frac{1}{4}(3-\sqrt{2})$

When implemented on a two-processor computer, this method effectively requires only two implicit stages. It has step point and stage order both equal to 2 (similar to the underlying IRK corrector method). Numerical experiments show the high efficiency of the proposed PDIRK_2 method for difficult problems in a range of accuracies which are realistic for these problems.

References

[1] G.J. Cooper and A. Sayfy, Semiexplicit A-stable Runge–Kutta methods, *Math. Comp.* **33** (146) (1979) 541–556.

[2] K. Dekker and J.G. Verwer, Eds., *Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations,* CWI Monographs **2** (North-Holland, Amsterdam, 1984).

[3] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems,* Springer Ser. Comput. Math. **14** (Springer, Berlin, 1991).

[4] Nguyen huu Cong, A-stable diagonally implicit Runge–Kutta–Nyström methods for parallel computers, *Numer. Algorithms* **4** (1993) 263–281.

[5] S.P. Nørsett, Semi-explicit Runge–Kutta methods, Report Math. Comput. No. 6/74, Dept. Math., Univ. Trondheim, 1974.

[6] A. Prothero and A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, *Math. Comp.* **28** (125) (1974) 145–162.

[7] P.J. van der Houwen and B.P. Sommeijer, Iterated Runge–Kutta methods on parallel computers, *SIAM J. Sci. Statist. Comput.* **12** (1990) 1000–1028.

[8] P.J. van der Houwen and B.P. Sommeijer, Analysis of parallel diagonally implicit iteration of Runge–Kutta methods, *Appl. Numer. Math.* **11** (1–3) (1993) 169–188.

[9] P.J. van der Houwen, B.P. Sommeijer and Nguyen huu Cong, Parallel diagonally implicit Runge–Kutta–Nyström methods, *Appl. Numer. Math.* **9** (2) (1992) 111–131.