Journal of Computational and Applied Mathematics 45 (1993) 169–180
North-Holland

CAM 1281

Stability of parallel Volterra–Runge–Kutta methods

M.R. Crisci, E. Russo
Dipartimento di Matematica e Applicazioni, Università di Napoli, Italy

P.J. van der Houwen
Centre for Mathematics and Computer Science, Amsterdam, Netherlands

A. Vecchio
Istituto per Applicazioni della Matematica, CNR, Napoli, Italy

Received 19 January 1992
Revised 16 March 1992

Abstract

Crisci, M.R., E. Russo, P.J. van der Houwen and A. Vecchio, Stability of parallel Volterra–Runge–Kutta methods, Journal of Computational and Applied Mathematics 45 (1993) 169–180.

In this paper, we analyse parallel iteration of Volterra–Runge–Kutta methods (PIVRK methods) for solving second-kind Volterra integral equations on parallel computers. We focus on the determination of the region of convergence C and on the stability region S_m of the iterated method obtained after m iterations. Results are presented for the convolution test equation. It turns out that the stability region S_m does not necessarily converge to the stability region S of the corrector. However, for finite m, S_m need not to be contained in C or S and may be much larger than C.

Keywords: Diagonally implicit Volterra–Runge–Kutta methods; parallelism; stability.

1. Introduction

So far the design of numerical methods for integrating the second-kind Volterra equation (VIE)

$$y(t) = g(t) + \int_{t_0}^{t} K(t, x, y(x))\,\mathrm{d}x, \quad t_0 \leqslant t \leqslant T, \tag{1.1}$$

on parallel computers has not received much attention. There are two straightforward ap-

Correspondence to: Prof. P.J. van der Houwen, Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, Netherlands.

proaches for constructing parallel VIE solvers. The most obvious approach is based on Picard iteration:

$$y_j(t) = g(t) + \int_{t_0}^{t} K\big(t, x, y_{j-1}(x)\big)\,\mathrm{d}x, \quad t_0 \leqslant t \leqslant T. \tag{1.2}$$

Because of the explicit nature of the iteration (1.2), it is suitable as a starting point for numerical discretization on parallel systems. In the case of initial-value problems for ordinary differential equations (ODEs), such so-called *waveform relaxation methods* have been investigated (see [1,2,7,8] where further references can be found). With appropriate modifications, these Picard-based waveform relaxation methods for ODEs can be applied to VIEs.

An alternative approach solves (1.1) by *step-by-step methods* which contain sufficient parallelism to take advantage of parallel architectures. It is this approach that will be considered in this paper. In particular, we shall consider Volterra–Runge–Kutta methods (VRK methods). The rather costly lag term evalutations appearing in such methods can be efficiently computed on multi-processor computers, so that we shall concentrate on the solution of the implicit relations to be solved in each integration step. Using similar predictor-corrector techniques as proposed for parallel ODE solvers (cf. [9,10,12]), we develop a *diagonally implicit iteration method* for approximating the solution of the corrector equation. Each iteration in this method requires the sequential solution of just one system of equations whose dimension equals that of the VIE.

The main issue in this paper is the derivation of stability conditions for parallel, iterated VRK methods (PIVRK methods). The actual construction of such methods with good stability properties and its application to test problems will be subject of future research.

2. VRK methods

For notational convenience, we shall assume that the VIE (1.1) is a scalar equation. However, all results in this paper can be straightforwardly extended to systems. Discretizing the integration interval $[t_0, T]$ by step points $\{t_n\colon n = 0, \ldots, N\}$, and denoting the numerical approximations to the exact solution value $y(t_n)$ and to

$$F(t, t_n) = g(t) + \int_{t_0}^{t_n} K(t, x, y(x))\,\mathrm{d}x \tag{2.1}$$

by y_n and $F_n(t)$, respecitvely, the general s-stage VRK method is defined by

$$y_{n+1} = F_n(t_n + h) + h\boldsymbol{b}^{\mathrm{T}} K(t_n\boldsymbol{e} + h\boldsymbol{a}, t_n\boldsymbol{e} + h\boldsymbol{c}, \boldsymbol{Y}_n), \tag{2.2a}$$

where the s components $Y_{n,i}$ of the *stage vector* $\boldsymbol{Y}_n$ are determined by the stage vector equations

$$Y_{n,i} = F_n(t_n + \theta_i h) + hA_i K(t_n\boldsymbol{e} + h\boldsymbol{a}_i, t_n\boldsymbol{e} + h\boldsymbol{c}, \boldsymbol{Y}_n), \quad i = 1, \ldots, s. \tag{2.2b}$$

The VRK parameters are stored in the $s \times 1$ column vectors $\boldsymbol{a}$, $\boldsymbol{a}_i$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\theta = (\theta_i)$, and in the $1 \times s$ row matrices $A_i = (a_{ij})$. In (2.2), $\boldsymbol{e}$ denotes the unit vector $(1, \ldots, 1)^{\mathrm{T}}$ and componentwise notation is used for functions with vector arguments. In order to avoid confusion, we denote the entries of $\boldsymbol{a}_i$ by a_{ij}^{*}. We observe that for Pouzet-type VRK (PVRK) methods, we have $\boldsymbol{a} = \boldsymbol{e}$, $\theta = \boldsymbol{c}$, $\boldsymbol{a}_i = c_i\boldsymbol{e}$.

2.1. Parallelism in VRK methods

The bulk of the computational effort per step goes in the evaluation of the s lag terms $F_n(t_n + \theta_i h)$. This makes VRK methods so much more expensive than, for example, direct quadrature methods which require only one lag term per step. However, since these s lag terms can be evaluated independently, the sequential computational costs can be reduced by a factor s if we have s processors at our proposal. Thus, on parallel systems the sequential costs of VRK methods per step are comparable with those of direct quadrature methods.

2.2. Parallel iteration of the stage vector equation

By using a sufficient number of parallel processors, the sequential computational costs of evaluating the lag terms can be reduced to such an extent that solving the s-stage vector equations (2.2b) may easily dominate the overall costs per step. This leads us to looking for parallel methods for solving (2.2b). Since this system of equations is quite similar to the stage vector equation associated with implicit RK methods for ODEs, we may resort to parallel methods devised for solving implicit RK methods. Such methods have been proposed in a number of papers and fit into the family of iteration schemes

$$\begin{aligned} &Y_{n,i}^{(\nu+1)} - hD_i K\big(t_n \boldsymbol{e} + h\boldsymbol{a}_i,\ t_n \boldsymbol{e} + h\boldsymbol{c},\ \boldsymbol{Y}_n^{(\nu+1)}\big) \\ &\quad = F_n(t_n + \theta_i h) + h[A_i - D_i] K\big(t_n \boldsymbol{e} + h\boldsymbol{a}_i,\ t_n \boldsymbol{e} + h\boldsymbol{c},\ \boldsymbol{Y}_n^{(\nu)}\big), \end{aligned} \tag{2.3}$$

where $i = 1, \ldots, s$ and $\nu = 0, 1, \ldots, m-1$, and where $\boldsymbol{Y}_n^{(0)}$ is an initial approximation to $\boldsymbol{Y}_n$. The $1 \times s$ row matrices D_i are assumed to have zero entries except for the ith entry which will be denoted by d_i. Notice that (2.3) reduces to fixed-point iteration (or predictor-corrector iteration) if the d_i vanish.

Evidently, the iteration method (2.3) is suited for implementation on parallel computers, because in each iteration the components $Y_{n,i}^{(\nu+1)}$, $i = 1, \ldots, s$, can be computed independently. For references to papers where the above type of iteration methods has been used we mention [11]. Once $\boldsymbol{Y}_n^{(m)}$ is computed, we find y_{n+1} according to the step point formula

$$y_{n+1} = F_n(t_n + h) + h\boldsymbol{b}^{\mathrm{T}} K\big(t_n \boldsymbol{e} + h\boldsymbol{a},\ t_n \boldsymbol{e} + h\boldsymbol{c},\ \boldsymbol{Y}_n^{(m)}\big). \tag{2.4}$$

However, if for some value i^* of i the equalities

$$\theta_{i^*} = 1, \quad A_{i^*} = \boldsymbol{b}^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{a}_{i^*} = \boldsymbol{a} \tag{2.5}$$

hold, then it follows from (2.2) that $Y_{n,i^*} = y_{n+1}$, so that instead of (2.4) we may use the formula

$$y_{n+1} = Y_{n,i^*}^{(m)}. \tag{2.6}$$

However, it should be remarked that the order in h of the iteration error is reduced by one (compare a similar situation in the case of diagonally implicit iteration of RK methods for ODEs [12]). We shall call the two iterated VRK methods (2.3), (2.4) and (2.3), (2.6) PIVRK methods of type 1 and type 2, respectively, and we denote them by PIVRK1 and PIVRK2.

3. The region of convergence of PIVRK methods

From (2.2b) and (2.3) we derive the iteration error recursion

$$Y_{n,i}^{(\nu+1)} - Y_{n,i} - hD_i\big[K\big(t_n e + ha_i,\ t_n e + hc,\ Y_n^{(\nu+1)}\big) - K(t_n e + ha_i,\ t_n e + hc,\ Y_n)\big]$$
$$= h(A_i - D_i)\big[K\big(t_n e + ha_i,\ t_n e + hc,\ Y_n^{(\nu)}\big) - hA_i K(t_n e + ha_i,\ t_n e + hc,\ Y_n)\big]. \quad (3.1)$$

In the special case where the kernel K is linear in its third argument, we may write

$$Y_{n,i}^{(\nu+1)} - Y_{n,i} - hD_i L_{n,i}\big[Y_n^{(\nu+1)} - Y_n\big] = h(A_i - D_i)L_{n,i}\big[Y_n^{(\nu)} - Y_n\big], \quad (3.2)$$

where the $L_{n,i}(h)$ are diagonal matrices. This recursion can be written more compactly in the form

$$Y_n^{(\nu+1)} - Y_n = Z_n\big[Y_n^{(\nu)} - Y_n\big], \quad (3.3)$$

where Z_n is the *iteration matrix* of the PIVRK method.

In this paper, we confine our considerations to the convolution test equation

$$y(t) = 1 + \int_{t_0}^{t} [\lambda + \mu(t-x)]y(x)\,\mathrm{d}x, \quad \lambda \leqslant 0,\ \mu < 0, \quad (3.4)$$

so that the iteration matrix is given by

$$Z_n = R(z, w, D) := \big[I - zD + w^2 DC\big]^{-1}\big[z(A-D) - w^2(B-DC)\big],$$
$$z := h\lambda, \qquad w := -h\sqrt{-\mu}\,, \quad (3.5)$$
$$B = (b_{ij}) := \big(a_{ij}(a_{ij}^* - c_j)\big), \qquad C = (c_{ij}) := \mathrm{diag}(a_{ii}^* - c_i), \qquad D := \mathrm{diag}(d_i).$$

We note that for $w = 0$, i.e., $\mu = 0$, the test equation (3.4) reduces to the familiar ODE test equation and the PIVRK method reduces to the so-called PDIRK method analysed in [12].

The spectral radius of the iteration matrix Z_n will be called the *convergence function*. From (3.3) we conclude that we have convergence if, and only if, the convergence function $\rho(R(z, w, D))$ is less than 1. This can always be achieved by choosing sufficiently small stepsizes h. In order to obtain the convergence condition on h explicitly, we derive the region in the (z, w)-plane determined by the condition $\rho(R(z, w, D)) < 1$. In computing these regions, we often found that in the third-quarter plane ($z \leqslant 0$, $w \leqslant 0$), they contain an infinite region bounded by the parabola $z = -pw^2 + qw + r$ with nonnegative p, q and r (this parabola corresponds to a straight line in the (z, w^2)-plane originating from one of the linear Hurwitz conditions). This motivates the following definition.

Definition 3.1. Let the *region of convergence* of the PIVRK method be defined by the set $C(D)$ where the convergence function is less than 1. Then the PIVRK method will be called *A_0-convergent*, *B_0-convergent* and *$V_0(p, q, r)$-convergent* if $C(D)$ contains the sets $\{(z, 0): z < 0\}$, $\{(0, w): w < 0\}$ and $\{(z, w): z < -pw^2 + qw + r,\ w < 0\}$, respectively. It is called *V_0-convergent* if it is $V_0(0, 0, 0)$-convergent.

Remark 3.2. From (3.5) it immediately follows that a necessary condition for A_0-convergence and B_0-convergence is given by $\rho(I - D^{-1}A) < 1$ and $\rho(I - C^{-1}D^{-1}B) < 1$, respectively. Hence,

in the case of fixed-point iteration, where $D = O$, the corresponding method can be neither A_0-convergent nor B_0-convergent. Furthermore, for iterated *Pouzet-type* VRK methods, where $C = O$ (because $a_i = c_i e$), B_0-convergence is also excluded. In fact, for such methods the iteration matrix $R(0, w, D)$ reduces to $-w^2 B$, so that the set $\{(0, w)\colon w < 0\}$ is given by the finite interval $(-(\rho(B))^{-1/2}, 0)$. Notice that this interval does not depend on the matrix D.

Remark 3.3. From the definition of the variables z and w it follows that for a $V_0(p, q, r)$-convergent method with vanishing p, the iteration method converges if the test equation satisfies the condition

$$\lambda < -q\sqrt{-\mu} + \frac{r}{h}. \tag{3.6a}$$

Hence, $V_0(0, q, r)$-convergent methods with $r \geqslant 0$ applied to test problems with

$$\lambda < -q\sqrt{-\mu} \tag{3.6b}$$

do not impose a condition in the stepsize h. Unfortunately, we did not yet find $V_0(p, q, r)$-convergent methods with $p = 0$, so that we obtain a condition on the stepsize. In the particular case of $V_0(p, q, r)$-convergent methods with $p > 0$ and $q = 0$ the iteration method converges if the stepsize h is less than $h_{\max}$ where $h_{\max}$ is the largest root of the equation $-p\mu h^2 + \lambda h = r$. If $r \geqslant 0$, then $-p\mu h_{\max} + \lambda \geqslant 0$, so that we have the sufficient convergence condition

$$h \leqslant \frac{\lambda}{p\mu}, \quad \lambda < 0, \ \mu < 0. \tag{3.7a}$$

This inequality shows that $1/p$ may be considered as a sort of convergence boundary and that the convergence condition is not a serious limitation on the stepsize for problems where $\lambda \ll p\mu$. For small values of $|\lambda|$, the convergence condition (3.7a) should be replaced by

$$h \leqslant \sqrt{\frac{r}{-p\mu}}, \quad \lambda \approx 0, \ \mu < 0, \tag{3.7b}$$

showing that for $|\lambda|$ small it is $\sqrt{r/p}$ that plays the role of a convergence boundary.

Remark 3.4. In order to achieve $V_0(p, q, r)$-convergence for small values of p, one may follow a similar approach as in [12] which leads us to the minimization of the value of the convergence function along the "parabolic" direction $z = -pw^2 + qw + r$ as $w \to -\infty$. It is easily verified that

$$\rho\big(R(-pw^2 + qw + r, w, D)\big) \to \rho\big(I - (pD + DC)^{-1}(pA + B)\big), \quad \text{as } w \to -\infty, \tag{3.8}$$

so that one may try to choose D such that $\rho(I - (pD + DC)^{-1}(pA + B))$ is minimized for small p. Examples of such methods are given in [6].

Next, we shall specify a triangular-shaped region which is at least contained in the region of convergence. For that purpose, we need the following lemma.

Lemma 3.5. *Let O denote the zero matrix and let the set $E(D)$ be defined by*

$$E(D) := \{(z, w): z < 0, w < 0, \| R(z, w, D)\|_\infty < 1\}.$$

If the entries of the diagonal matrices C and D are nonnegative, then $E(O)$ is contained in $E(D)$.

Proof. Let (z, w) be in $E(O)$; then it follows from the definition of R (see (3.5)) and the condition on the matrices C and D that

$$\sum_{j=1}^{s} |za_{ij} - w^2 b_{ij}| < 1,$$

for all i. Hence,

$$\sum_{j=1}^{s} |z(a_{ij} - \delta_{ij}d_i) - w^2(b_{ij} - d_i c_{ij})| \leqslant -zd_i + w^2 c_{ii} + \sum_{j=1}^{s} |za_{ij} - w^2 b_{ij}|$$

$$\leqslant 1 - zd_i + w^2 c_{ii}.$$

Thus,

$$(1 - zd_i + w^2 c_{ii})^{-1} \sum_{j=1}^{s} |z(a_{ij} - \delta_{ij}d_i) - w^2(b_{ij} - d_i c_{ij})| \leqslant 1, \quad i = 1, \ldots, s,$$

which implies that (z, w) lies in $E(D)$. □

By means of this lemma and the observation that the set $E(D)$ is obviously contained in the convergence region $C(D)$, we can prove the following theorem.

Theorem 3.6. *Let the region $T(D)$ be defined by*

$$T(D) := \{(z, w): z < 0, w < 0, \| B - DC\|_\infty w^2 < 1 + \| A - D\|_\infty z\}.$$

If the entries of the diagonal matrices C and D are nonnegative, then both $T(O)$ and $T(D)$ are contained in the convergence region $C(D)$.

Proof. Let (z, w) be in $T(D)$; then

$$\| B - DC\|_\infty w^2 < 1 + \|A - D\|_\infty z = 1 - \| z(A - D)\|_\infty.$$

On the other hand, it follows from the definition of R and the condition on the matrices C and D that

$$\| R(z, w, D)\|_\infty \leqslant \| z(A - D) - w^2(B - DC)\|_\infty$$

$$\leqslant \| z(A - D)\|_\infty + \| w^2(B - DC)\|_\infty,$$

so that $\| R(z, w, D)\|_\infty < 1$. From the definition of $E(D)$ it then follows that (z, w) lies in $E(D)$. Hence, $T(D)$ is contained in $E(D)$. In particular, $T(O)$ is contained in $E(O)$, so that by virtue of Lemma 3.5, $T(O)$ is contained in $E(D)$. Thus, both $T(O)$ and $T(D)$ are contained in $E(D)$ and therefore in the convergence region $C(D)$. □

Table 3.1
Convergence boundaries $1/p$ and $\sqrt{r/p}$ associated with PIVRK methods generated by correctors and matrices D specified in [12]

	Corrector	D	$1/p$	$\sqrt{r/p}$
Two-stage Radau IIA	[12, (4.6a)]	[12, (4.6b)]	2.0	3.1
Two-stage Lagrange	[12, (4.4a)]	[12, (4.4b)]	2.5	3.8
Three-stage Radau IIA	[12, (4.10a)]	[12, (4.10b)]	2.3	3.8
Three-stage Lagrange	[12, (4.9a)]	[12, (4.9b)]	2.3	3.4

3.1. $V_0(p, 0, r)$-convergent PIVRK methods of Pouzet type

It is of interest to investigate the convergence regions of PIVRK methods. In this paper, we have restricted our considerations to Pouzet-type correctors and matrices D derived in [12] for ODEs. (In order to save space, we refer for a precise specification to the formulas given in [12].)

For a number of PIVRK methods, Table 3.1 lists the values of the convergence boundaries $1/p$ and $\sqrt{r/p}$ associated with the $V_0(p, 0, r)$-convergence regions. From this table and from (3.7) the convergence conditions for h can be derived.

4. The region of stability of PIVRK methods

The main purpose of this paper is the stability analysis of iterated VRK methods with respect to the convolution test equation (3.4). For the stability analysis of the VRK method itself (i.e., the corrector (2.2)), we refer to [4,5] where further references can be found.

Let the lag term $F_n(t)$ in (2.2) be of "extended" type. Then

$$y_n = F_n(t_n). \tag{4.1}$$

Furthermore, if J_n denotes the "integrated" stage vector, then

$$F_n(t) := g(t) + h\sum_{j=0}^{n-1} \boldsymbol{b}^{\mathrm{T}} K\big(te + h(\boldsymbol{a} - e),\ t_j e + hc,\ \boldsymbol{Y}_j\big), \qquad J_n := \sum_{j=0}^{n-1} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{Y}_j. \tag{4.2}$$

Applying the VRK method (2.2) to the test equation (3.4) reduces (2.2) to the recursion

$$\begin{aligned} \boldsymbol{Y}_n &= F_n(t_n)\boldsymbol{e} - w^2\theta J_n + \big[zA - w^2 B\big]\boldsymbol{Y}_n, \\ y_{n+1} &= F_n(t_n) - w^2 J_n + \big[z\boldsymbol{b} - w^2 \boldsymbol{b}_1\big]\boldsymbol{Y}_n \end{aligned} \tag{4.3}$$

(cf. [3, p.474]), where $\boldsymbol{b}_1 := (b_i(a_i - c_i))$.

We shall derive similar recursions for the iterated methods. It is readily seen that the analogue of (4.2) becomes

$$\begin{aligned} F_n(t) &:= g(t) + h\sum_{j=0}^{n-1} \boldsymbol{b}^{\mathrm{T}} K\big(te + h(\boldsymbol{a} - e),\ t_j e + hc,\ \boldsymbol{Y}_j^{(m)}\big), \\ J_n &:= h\sum_{j=0}^{n-1} \boldsymbol{b}^{\mathrm{T}} \boldsymbol{Y}_j^{(m)}. \end{aligned} \tag{4.4}$$

First, type 1 methods will be considered. Using (4.1) and (2.4), the analogue of (4.3) becomes

$$\begin{aligned} &[I - zD + w^2CD]Y_n^{(\nu+1)} = y_n e - w^2\theta J_n + [z(A-D) - w^2(B-CD)]Y_n^{(\nu)}, \\ &y_{n+1} = y_n - w^2 J_n + [zb - w^2 b_1]Y_n^{(m)}. \end{aligned} \tag{4.5}$$

Let us define the functions

$$\begin{aligned} &Q_m := R^m - [R^m - I]S^{-1}, \\ &S := I - zD + w^2CD - z(A-D) + w^2(B-CD) = I - zA + w^2 B, \end{aligned} \tag{4.6}$$

where R is the iteration matrix defined by (3.5) (for brevity, we omit the arguments of R). Furthermore, let the initial approximation be chosen according to

$$Y_n^{(0)} = F_n(y_n e + h\theta) = y_n e - w^2\theta J_n. \tag{4.7}$$

Then, by observing that for extended lag term formulas,

$$J_{n+1} = J_n + b^{\mathrm{T}} Y_n^{(m)} \tag{4.8a}$$

(cf. [3, p.481]), we deduce from (4.5)

$$Y_n^{(m)} = Q_m[y_n e - w^2\theta J_n], \quad y_{n+1} = y_n - w^2 J_n + [zb - w^2 b_1]^{\mathrm{T}} Y_n^{(m)}. \tag{4.8b}$$

Elimination of $Y_n^{(m)}$ from (4.8) yields the recursion

$$\begin{pmatrix} y_{n+1} \\ J_{n+1} \end{pmatrix} = M_m(z, w, D)\begin{pmatrix} y_n \\ J_n \end{pmatrix}, \tag{4.8'}$$

where the *stability matrix* $M_m(z, w, D)$ is given by

$$M_m(z, w, D) = \begin{pmatrix} 1 + [zb - w^2 b_1]^{\mathrm{T}} Q_m e & w^2\left(-1 + [zb - w^2 b_1]^{\mathrm{T}} Q_m \theta\right) \\ b^{\mathrm{T}} Q_m e & 1 - w^2 b^{\mathrm{T}} Q_m \theta \end{pmatrix}. \tag{4.9}$$

Likewise, we find for type 2 methods the stability matrix

$$M_m(z, w, D) = \begin{pmatrix} e_s^{\mathrm{T}} Q_m e & -w^2 e_s^{\mathrm{T}} Q_m \theta \\ b^{\mathrm{T}} Q_m e & 1 - w^2 b^{\mathrm{T}} Q_m \theta \end{pmatrix}. \tag{4.10}$$

The spectral radius of $M_m(z, w, D)$ will be called the *stability function*. The analogue of Definition 3.1 is given by the following definition.

Definition 4.1. Let the *region of stability* of the PIVRK method be defined by the set $S_m(D)$ where the stability function is less than 1. Then the PIVRK method will be called A_0*-stable*, B_0*-stable* and $V_0(p, q, r)$*-stable* if $S_m(D)$ contains the sets $\{(z, 0)\colon z < 0\}$, $\{(0, w)\colon w < 0\}$ and $\{(z, w)\colon z < -pw^2 + qw + r, w < 0\}$, respectively. It is called V_0*-stable* if it is $V_0(0, 0, 0)$-stable (cf. [3]).

Remark 4.2. The stability region $S_m(D)$ converges to the intersection of the convergence region $C(D)$ and the stability region S of the corrector as m tends to infinity. This can be explained by observing that the stability region $S_m(D)$ converges to a fixed stability region S_∞ if Q_m converges to the fixed matrix $-S^{-1}$ as $m \to \infty$. This happens if, and only if, the matrix

$R = R(z, w, D)$ in (4.6) has its eigenvalues within the unit disk, that is, if (z, w) lies in the convergence region $C(D)$. Furthermore, since the stability matrix $M_m(z, w, D)$ converges to that of the corrector if (z, w) lies in $C(D)$, we may conclude that $S_m(D)$ converges to the intersection of $C(D)$ and the stability region S of the corrector. As a consequence, the stability region of a PIVRK method does not necessarily converge to that of the corrector. However, for finite m, $S_m(D)$ need not be contained in $C(D)$ or S and may be much larger than $C(D)$.

Remark 4.3. From Remarks 3.2 and 4.2 it follows that PIVRK methods of Pouzet-type cannot be B_0-stable (because they cannot be B_0-convergent).

Remark 4.4. Similar to $V_0(0, q, r)$-convergent and $V_0(p, 0, r)$-convergent methods, $V_0(0, q, r)$-stable and $V_0(p, 0, r)$-stable methods impose a condition either on the parameters λ and μ (cf. (3.7a)) or on the stepsize h (cf. (3.7b)). Similar to the terminology used for characterizing the convergence region, we shall call the quantities $1/p$ and $\sqrt{r/p}$ the *stability boundaries associated with the stability region.*

4.1. Necessary stability conditions

We shall derive necessary conditions for stability by requiring that the stability function is at least bounded along certain directions in the (z, w)-plane. We shall distinguish stability along *horizontal* directions (w constant), *vertical* directions (z constant) and *parabolic* directions ($z = -pw^2$ with p constant).

Theorem 4.5. *Let m be fixed and let the matrices K, U and G be defined by*

$$K = I - D^{-1}A, \qquad U = I - (DC)^{-1}B, \qquad G = I - (pD + DC)^{-1}(pA + B), \tag{4.11}$$

where DC is assumed nonsingular. Then the conditions in Table 4.1 are necessary conditions to have a bounded stability function $\rho(M_m(z, w, D))$ along horizontal, vertical or parabolic directions.

Proof. For large values of $|z|$, $|w|$ and $|z| = |pw^2|$ we may respectively write

$$\begin{aligned} Q_m &= K^m + \frac{L_m}{z} + O\left(\frac{1}{z^2}\right), \qquad Q_m = U^m + \frac{V_m}{w^2} + \frac{W_m}{w^4} + O\left(\frac{1}{w^6}\right), \\ Q_m &= G^m + \frac{H_m}{w^2} + O\left(\frac{1}{w^4}\right), \end{aligned} \tag{4.12}$$

where K, U and G are defined in (4.11), where L_m does not depend on z (but may depend on

Table 4.1
Necessary stability conditions for bounded stability functions along various directions

Type	Horizontal direction	Vertical direction	Parabolic direction
1	$b^T K^m e = 0$	$b^T U^m \theta = b_1^T U^m e = 0$	$b^T G^m \theta = (b_1 + pb)^T G^m e = 0$
2	No condition required	$b^T U^m \theta = 0$	$b^T G^m \theta = 0$

w^2), where V_m and W_m do not depend on w (but may depend on z), and where H_m only depends on p. Substitution of (4.12) into (4.9) and (4.10) yields along horizontal directions

$$M_m(z, w, D) = z\begin{pmatrix} \boldsymbol{b}^{\mathrm{T}}K^m\boldsymbol{e} & \boldsymbol{b}^{\mathrm{T}}K^m\theta \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 1+\boldsymbol{b}^{\mathrm{T}}L_m\boldsymbol{e}-w^2\boldsymbol{b}_1^{\mathrm{T}}K^m\boldsymbol{e} & \boldsymbol{b}^{\mathrm{T}}L_m\theta - w^2\left(1+\boldsymbol{b}_1^{\mathrm{T}}K^m\theta\right) \\ \boldsymbol{b}^{\mathrm{T}}K^m\boldsymbol{e} & 1-w^2\boldsymbol{b}^{\mathrm{T}}K^m\theta \end{pmatrix} + \mathrm{O}\left(\frac{1}{z}\right), \tag{4.9'}$$

$$M_m(z, w, D) = \begin{pmatrix} \boldsymbol{e}_s^{\mathrm{T}}K^m\boldsymbol{e} & -w^2\boldsymbol{e}_s^{\mathrm{T}}K^m\theta \\ \boldsymbol{b}^{\mathrm{T}}K^m\boldsymbol{e} & 1-w^2\boldsymbol{b}^{\mathrm{T}}K^m\theta \end{pmatrix} + \mathrm{O}\left(\frac{1}{z}\right). \tag{4.10'}$$

Similarly, we find along vertical directions

$$M_m(z, w, D) = w^2\begin{pmatrix} -\boldsymbol{b}_1^{\mathrm{T}}U^m\boldsymbol{e} & -1+(z\boldsymbol{b}-\boldsymbol{b}_1)^{\mathrm{T}}U^m\theta - \boldsymbol{b}_1^{\mathrm{T}}V_m\theta \\ 0 & -\boldsymbol{b}^{\mathrm{T}}U^m\theta \end{pmatrix} + \begin{pmatrix} 1+z\boldsymbol{b}^{\mathrm{T}}U^m\boldsymbol{e}-\boldsymbol{b}_1^{\mathrm{T}}V_m\boldsymbol{e} & z\boldsymbol{b}^{\mathrm{T}}V_m\theta - \boldsymbol{b}_1^{\mathrm{T}}W_m\theta \\ \boldsymbol{b}^{\mathrm{T}}U^m\boldsymbol{e} & 1-\boldsymbol{b}^{\mathrm{T}}V_m\theta \end{pmatrix} + \mathrm{O}\left(\frac{1}{w^2}\right), \tag{4.9''}$$

$$M_m(z, w, D) = w^2\begin{pmatrix} 0 & -\boldsymbol{e}_s^{\mathrm{T}}U^m\theta \\ 0 & -\boldsymbol{b}^{\mathrm{T}}U^m\theta \end{pmatrix} + \begin{pmatrix} \boldsymbol{e}_s^{\mathrm{T}}U^m\boldsymbol{e} & -\boldsymbol{e}_s^{\mathrm{T}}V_m\theta \\ \boldsymbol{b}^{\mathrm{T}}U^m\boldsymbol{e} & 1-\boldsymbol{b}^{\mathrm{T}}V_m\theta \end{pmatrix} + \mathrm{O}\left(\frac{1}{w^2}\right). \tag{4.10''}$$

Finally, along parabolic directions, the matrix (4.9) assumes the form

$$M_m(z, w, D) = w^4\begin{pmatrix} 0 & E_1 \\ 0 & 0 \end{pmatrix} + w^2\begin{pmatrix} -(\boldsymbol{b}_1+p\boldsymbol{b})^{\mathrm{T}}G^m\boldsymbol{e} & E_2 \\ 0 & -\boldsymbol{b}^{\mathrm{T}}G^m\theta \end{pmatrix} + \mathrm{O}(1), \tag{4.9'''}$$

where we refrained from writing out the entries E_1 and E_2. The matrix (4.10) can be obtained from (4.10'') by replacing U with G and V_m with H_m.

For large values of $|z|$ and $|w|$ these matrices are dominated by the first few matrix terms. Hence, a necessary condition for stability is that the eigenvalues of these dominating matrices are bounded by 1. If these dominating matrices have entries depending on z or w, respectively, then we should require that these matrices have zero eigenvalues. This requirement leads to the necessary conditions in Table 4.1. □

4.2. $V_0(p, 0, r)$-stability

In order to get insight into the number of iterations needed to reach a sufficiently stable method, we investigate the stability regions of Pouzet methods arising from the correctors and matrices D derived in [12] for ODEs. It turned out that for these matrices D the type 1 methods do not satisfy the necessary condition $\boldsymbol{b}^{\mathrm{T}}K^m\boldsymbol{e} = 0$ of Table 4.1 and are therefore not $V_0(p, 0, r)$-stable; (however, although their stability regions are finite, they are quite large, so that they may still be of use for mildly stiff problems).

For type 2 methods, Table 4.2 lists the stability boundaries $1/p$ and $\sqrt{r/p}$ associated with the $V_0(p, 0, r)$-stability regions (finite stability regions are indicated by $*$). The last column in this

Table 4.2
Stability boundaries $1/p$ and $\sqrt{r/p}$ associated with PIVRK2 methods generated by correctors and matrices D specified in [12]

Corrector, matrix D	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=\infty$
Two-stage Radau IIA [12, (4.6a), (4.6b)]	1.1, 2.0	1.6, 2.0	1.7, 2.4	2.0, 2.0	2.0, 2.0	2.0, 2.0	2.0, 2.0	2.0, 2.0
Two-stage Lagrange [12, (4.4a), (4.4b)]	0.9, 2.4	0.7, 3.0	1.1, 2.4	1.1, 3.0	1.1, 3.0	1.4, 3.0	1.1, 3.0	1.1, 3.8
Three-stage Radau IIA [12, (4.9a), (4.9b)]	*, *	0.8, 3.9	0.8, 3.9	1.2, 3.9	1.0, 3.8	1.0, 3.8	1.2, 3.1	1.5, 3.4
Three-stage Lagrange [12, (4.9a), (4.9b)]	*, *	1.0, 3.1	1.2, 3.1	1.3, 3.4	1.6 3.1	1.3, 3.4	1.6, 3.1	1.6, 3.1

table has been obtained using the property that $S_m(D)$ converges to the intersection of $C(D)$ and the stability region S of the corrector. In general the *stability regions* of the PIVRK methods do *not* converge to that of the corrector. Typically, the stability region of the corrector contains an additional wedge along the negative w-axis. However, the PIVRK method does not converge here, so that the PIVRK method cannot be stable in this strip (for detailed stability plots, we refer to [6]).

References

[1] A. Bellen, Parallelism across the steps for difference and differential equations, in: A. Bellen, C.W. Gear and E. Russo, Eds., *Numerical Methods for Ordinary Differential Equations*, Lecture Notes in Math. **1386** (Springer, Berlin, 1989) 22–35.

[2] A. Bellen, Z. Jackiewicz and M. Zennaro, Stability analysis of time-point relaxation Heun method, C.N.R. Progetto Finalizzato "Sistemi Informatici e Calcolo Parallelo", Tech. Rep. #1/30, 1990.

[3] H. Brunner and P.J. van der Houwen, *The Numerical Solution of Volterra Equations,* CWI Monographs **3** (North-Holland, Amsterdam, 1986).

[4] M.R. Crisci, E. Russo and A. Vecchio, On the stability of one-step exact collocation methods for the numerical solution of the second-kind Volterra integral equation, *BIT* **29** (1989) 258–269.

[5] M.R. Crisci, E. Russo and A. Vecchio, Stability analysis of the de Hoog and Weiss implicit Runge–Kutta methods for the Volterra integral and integrodifferential equations, *J. Comput. Appl. Math.* **29** (3) (1990) 329–341.

[6] M.R. Crisci and A. Vecchio, Stability plots of parallel Volterra–Runge–Kutta methods for Volterra integral equations, Rapp. Tecnico No. 87/91, Ist. Appl. Mat., CNR, Napoli, 1991.

[7] C.W. Gear, Massive parallelism across space in ODEs, *Appl. Numer. Math.* **11** (1–3) (1993) 27–43.

[8] C.W. Gear and Xu Xuhai, Parallelism across time in ODEs, *Appl. Numer. Math.* **11** (1–3) (1993) 45–68.

[9] K.R. Jackson and S.P. Nørsett, The potential for parallelism in Runge–Kutta methods, Part I: RK formulas in standard form, Technical Report No. 239/90, Dept. Comput. Sci., Univ. Toronto, 1990.

[10] K.R. Jackson and S.P. Nørsett, The potential for parallelism in Runge–Kutta methods, Part II: RK predictor-corrector formulas, in preparation.

[11] P.J. van der Houwen and B.P. Sommeijer, Parallel ODE solvers, in: *ACM Conf. on Super-computing,* Amsterdam (ACM, New York, 1990) 71–81.

[12] P.J. van der Houwen and B.P. Sommeijer, Iterated Runge–Kutta methods on parallel computers, *SIAM J. Sci. Statist. Comput.* **12** (1991) 1000–1028.