Waveform relaxation methods for implicit differential equations

van der Houwen, P.J.; van der Veen, W.A.

Published in:
Advances in Computational Mathematics

DOI:
10.1023/A:1018942718589

Link to publication

Citation for published version (APA):
van der Houwen, P. J., & van der Veen, W. A. (1997). Waveform relaxation methods for implicit differential equations. *Advances in Computational Mathematics*, *7*, 183-197. https://doi.org/10.1023/A:1018942718589

Waveform relaxation methods for implicit differential equations [*]

P. J. van der Houwen and W. A. van der Veen

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

We apply a Runge–Kutta-based waveform relaxation method to initial-value problems for implicit differential equations. In the implementation of such methods, a sequence of nonlinear systems has to be solved iteratively in each step of the integration process. The size of these systems increases linearly with the number of stages of the underlying Runge–Kutta method, resulting in high linear algebra costs in the iterative process for high-order Runge–Kutta methods. In our earlier investigations of iterative solvers for implicit initial-value problems, we designed an iteration method in which the linear algebra costs are almost independent of the number of stages when implemented on a parallel computer system. In this paper, we use this parallel iteration process in the Runge–Kutta waveform relaxation method. In particular, we analyse the convergence of the method. The theoretical results are illustrated by a few numerical examples.

Keywords: numerical analysis, implicit differential equations, convergence, waveform relaxation, Runge–Kutta methods, parallelism.

AMS subject classification: 65L05.

1. Introduction

Consider the initial-value problem (IVP) for the implicit differential equation (IDE)

$$\phi(t, \boldsymbol{y}', \boldsymbol{y}) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}}, \ \boldsymbol{y}, \phi \in \mathbb{R}^d. \tag{1.1}$$

It will be assumed that the initial conditions for $\boldsymbol{y}(t_0)$ and $\boldsymbol{y}'(t_0)$ are consistent and that the IVP has a unique solution. Furthermore, defining the Jacobian matrices $K := \phi_{\boldsymbol{u}}(t, \boldsymbol{u}, \boldsymbol{v})$ and $J := -\phi_{\boldsymbol{v}}(t, \boldsymbol{u}, \boldsymbol{v})$, it will be assumed that in the neighbourhood of the solution, the characteristic equation $\det(\lambda K - J) = 0$ associated with the linearization of (1.1) has only zeros in the nonpositive halfplane. The pair of matrices $\{K, J\}$ will be said to be a *stable pair* if they satisfy this requirement. In the convergence analysis of iteration methods for solving the numerical discretization of (1.1), the property of matrix pairs will play a central role.

[*] The research reported in this paper was partly supported by the Technology Foundation (STW) in the Netherlands.

A large class of numerical discretizations of (1.1) is defined by

$$\begin{aligned} &\boldsymbol{y}_n = \left(\boldsymbol{e}_s^T \otimes I\right)\boldsymbol{Y}_n, \\ &\boldsymbol{\Phi}\left(\boldsymbol{e}t_{n-1} + \boldsymbol{c}h, \left(h^{-1}A^{-1} \otimes I\right)\left(\boldsymbol{Y}_n - (E \otimes I)\boldsymbol{Y}_{n-1}\right), \boldsymbol{Y}_n\right) = 0. \end{aligned} \tag{1.2}$$

Here, $\boldsymbol{y}_n$ is the numerical approximation to the exact solution value $\boldsymbol{y}(t_n)$, A and E denote s-by-s matrices, $\boldsymbol{e}_s$ is the sth unit vector, h is the step size $t_n - t_{n-1}$ (to be assumed constant in the analysis presented in this paper), $\otimes$ denotes the Kronecker product, and I is the d-by-d identity matrix (in the following, we shall use the notation I for any identity matrix; its dimension will always be clear from the context). The s components $\boldsymbol{Y}_{ni}$ of the sd-dimensional stage vector $\boldsymbol{Y}_n$ represent approximations to the exact solution values $\boldsymbol{y}(t_{n-1}+c_ih)$, where the c_i are the components of the abscissa vector $\boldsymbol{c} = (c_i)$ and where $c_s = 1$. Furthermore, for any pair of vectors $\boldsymbol{Y}_n' = (\boldsymbol{Y}_{ni}')$ and $\boldsymbol{Y}_n = (\boldsymbol{Y}_{ni})$, we define the function

$$\boldsymbol{\Phi}\left(\boldsymbol{e}t + \boldsymbol{c}h, \boldsymbol{Y}_n', \boldsymbol{Y}_n\right) := \left(\phi\left(t + c_ih, \boldsymbol{Y}_{ni}', \boldsymbol{Y}_{ni}\right)\right). \tag{1.3}$$

The method (1.2) is completely defined by the triple $\{A, E, \boldsymbol{c}\}$. We remark that (1.2) reduces to a (stiffly accurate) RK method for IDEs if A equals the Butcher matrix of the RK method, $\boldsymbol{c} := A\boldsymbol{e}$, and $E := (\boldsymbol{0}, \ldots, \boldsymbol{0}, \boldsymbol{e})$, $\boldsymbol{e}$ being the s-dimensional vector with unit entries (see [5]).

In [13], parallel iteration methods for solving the stage vector $\boldsymbol{Y}_n$ from the nonlinear system (1.2) have been proposed. In this paper, we want to combine these parallel iteration techniques with the waveform relaxation (WR) approach. The resulting numerical solution methods have a considerable amount of intrinsic parallelism. However, the price to be paid is a decrease of the speed of convergence of the iteration methods. This paper studies how the convergence of the WR method is influenced by the number of WR iterations, the number of modified Newton iterations, and the number of inner iterations (for solving the linear Newton systems). The theoretical results are illustrated by a few numerical examples.

2. WR methods

The derivation of WR methods starts with representing the IDE (1.1) in the form

$$\psi\left(t, \boldsymbol{y}', \boldsymbol{y}', \boldsymbol{y}, \boldsymbol{y}\right) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}}, \ \boldsymbol{y}, \psi \in \mathbb{R}^d, \tag{2.1}$$

where $\psi(t, \boldsymbol{u}', \boldsymbol{v}', \boldsymbol{u}, \boldsymbol{v})$ is a *splitting function* satisfying $\psi(t, \boldsymbol{u}', \boldsymbol{u}', \boldsymbol{u}, \boldsymbol{u}) = \phi(t, \boldsymbol{u}', \boldsymbol{u})$. This splitting function is chosen such that the Jacobian matrices $K^* := \partial\psi/\partial\boldsymbol{u}'$ and $J^* = -\partial\psi/\partial\boldsymbol{u}$ have a simple structure, so that, given an approximation $\boldsymbol{y}^{(k-1)}$ to the solution $\boldsymbol{y}$ of (1.1), a next approximation $\boldsymbol{y}^{(k)}$ is more easily solved from the system

$$\psi\left(t, \boldsymbol{y}'^{(k)}, \boldsymbol{y}'^{(k-1)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k-1)}\right) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}}, \ \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k-1)}, \psi \in \mathbb{R}^d, \tag{2.2}$$

than $\boldsymbol{y}$ is solved from (1.1). Here, $k = 1, 2, \ldots, q$, and $\boldsymbol{y}^{(0)}$ denotes an initial approximation to the solution of (1.1). The iteration process (2.2) is called *continuous WR* *iteration* with WR iterates $\boldsymbol{y}^{(k)}$. Such iteration processes were introduced in Lelarasmee [7] and Lelarasmee et al. [8]. For linear problems, its convergence has been extensively studied in [9].

In the case of *explicit* differential equations (i.e., $K = K^* = I$), a popular choice for the splitting function ψ is such that the matrix J^* is σ-by-σ block-diagonal (block-Jacobi WR method). Then, each iteration of the WR method (2.2) requires the integration of σ uncoupled systems over the interval $[t_0, t_{\text{end}}]$ (note that these integrations can be done in parallel on σ processors). In the IDE case ($K \neq I$), we obtain a block-Jacobi WR method if both J^* and K^* are σ-by-σ block-diagonal. As an example, we consider the case where (2.2) is of the form

$$\begin{aligned}
&\psi_1\bigl(t, \boldsymbol{u}'^{(k)}, \boldsymbol{y}'^{(k-1)}, \boldsymbol{u}^{(k)}, \boldsymbol{y}^{(k-1)}\bigr) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}},\ \boldsymbol{u}^{(k)}, \psi_1 \in \mathbb{R}^{d_1},\\
&\psi_2\bigl(t, \boldsymbol{v}'^{(k)}, \boldsymbol{y}'^{(k-1)}, \boldsymbol{v}^{(k)}, \boldsymbol{y}^{(k-1)}\bigr) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}},\ \boldsymbol{v}^{(k)}, \psi_2 \in \mathbb{R}^{d_2}.
\end{aligned} \tag{2.2$'$}$$

Here, $d_1 + d_2 = d$ and $\boldsymbol{y} = (\boldsymbol{u}^T, \boldsymbol{v}^T)^T$. Obviously, K^* and J^* are both 2-by-2 block-diagonal. More generally, whenever K^* and J^* are both σ-by-σ block-diagonal, we find a set of σ subsystems with the generic form

$$\psi\bigl(t, \boldsymbol{y}'^{(k)}, \boldsymbol{x}'^{(k-1)}, \boldsymbol{y}^{(k)}, \boldsymbol{x}^{(k-1)}\bigr) = \boldsymbol{0}, \quad t_0 \leqslant t \leqslant t_{\text{end}}, \tag{2.3}$$

where $\boldsymbol{x}^{(k-1)}$ is defined by the σ subsystems solutions of the preceding WR iteration and $\boldsymbol{y}^{(k)}$ is the new subsystem solution. For further details we refer to [2, p. 276 ff.].

The convergence of the continous WR iteration (2.2) is faster as the integration interval $[t_0, t_{\text{end}}]$ is smaller. In fact, for ordinary differential equations (ODEs) which arise for $K = I$, and for sufficiently smooth splitting functions ψ, we have the well-known estimate

$$\bigl\|\boldsymbol{y}^{(k)}(t) - \boldsymbol{y}(t)\bigr\| \leqslant \frac{L^k(t-t_0)^k}{k!} \max_{t_0 \leqslant \tau \leqslant t_{\text{end}}} \bigl\|\boldsymbol{y}^{(0)}(\tau) - \boldsymbol{y}(\tau)\bigr\|,$$

where L is a constant depending on the splitting function (for example, for the standard test equation defined by $\phi = \boldsymbol{y}' - \lambda\boldsymbol{y}$ with splitting function $\psi = \boldsymbol{u}' - \lambda\boldsymbol{v}$, we have $L = |\lambda|$). This estimate indicates that convergence is improved if $t_{\text{end}} - t_0$ is small. Therefore, we do not apply the WR method on the whole interval $[t_0, t_{\text{end}}]$, but successively on a number of smaller subintervals (also called *windows*) of length ωh where ω is a usually small integer and h the step size.

2.1. Discrete WR iteration

Let us integrate the IVP for (2.2) numerically by the step-by-step method $\{A, E, \boldsymbol{c}\}$ defined in (1.2). Introducing the residual function

$$\begin{aligned}
\boldsymbol{R}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}) := \Psi\bigl(&\boldsymbol{e}t_{n-1} + \boldsymbol{c}h, \bigl(h^{-1}A^{-1} \otimes I\bigr)\bigl(\boldsymbol{U} - (E \otimes I)\boldsymbol{V}\bigr),\\
&\bigl(h^{-1}A^{-1} \otimes I\bigr)\bigl(\boldsymbol{X} - (E \otimes I)\boldsymbol{V}\bigr), \boldsymbol{U}, \boldsymbol{X}\bigr),
\end{aligned} \tag{2.4a}$$

and dividing $[t_0, t_{\text{end}}]$ into subintervals (or windows) $[t_{\kappa\omega}, t_{\kappa\omega+\omega}]$, we obtain on $[t_{\kappa\omega}, t_{\kappa\omega+\omega}]$ the scheme

```
for $k = 1$ to $q$
    $\boldsymbol{Y}^{(k)}_{\kappa\omega} := \boldsymbol{Y}^{(q)}_{\kappa\omega}$
    for $n = \kappa\omega + 1$ to $\kappa\omega + \omega$                    (2.4b)
        solve $\boldsymbol{Y}^{(k)}_n$ from $\boldsymbol{R}(\boldsymbol{Y}^{(k)}_n, \boldsymbol{Y}^{(k)}_{n-1}, \boldsymbol{Y}^{(k-1)}_n) = \boldsymbol{0}$,
        set   $\boldsymbol{y}^{(k)}_n = (\boldsymbol{e}_s^T \otimes I)\boldsymbol{Y}^{(k)}_n$.
```

Here, $\boldsymbol{y}^{(k)}_n$, $\boldsymbol{Y}^{(k)}_n$, and Ψ are the analogues of $\boldsymbol{y}_n$, $\boldsymbol{Y}_n$ and Φ occurring in (1.2). The scheme (2.4) will be called the *discrete WR iteration process* with (discrete) WR iterates $\boldsymbol{Y}^{(k)}_n$ and $\boldsymbol{y}^{(k)}_n$.

If (2.4) converges on all windows as $q \to \infty$, then $\boldsymbol{Y}^{(q)}_n$ converges to the solution $\boldsymbol{Y}_n$ of $\boldsymbol{R}(\boldsymbol{Y}_n, \boldsymbol{Y}_{n-1}, \boldsymbol{Y}_n) = \boldsymbol{0}$, that is, to the stage vector $\boldsymbol{Y}_n$ defined in (1.2). As a consequence, $(\boldsymbol{e}_s^T \otimes I)\boldsymbol{Y}^{(q)}_n$ approximates the solution of (1.1) at t_n with order p in h, p being the order of accuracy of the underlying method (1.2).

The iteration scheme (2.4) has a certain amount of intrinsic parallelism, because for a given subinterval $[t_{\kappa\omega}, t_{\kappa\omega+\omega}]$ and given k, the ω iterates $\{\boldsymbol{Y}^{(k)}_{\kappa\omega+1}, \boldsymbol{Y}^{(k-1)}_{\kappa\omega+2}, \ldots, \boldsymbol{Y}^{(k+1-\omega)}_{\kappa\omega+\omega}\}$ can be computed in parallel (parallelism across the steps within a window, see, e.g., [14] and [1]). Hence, effectively, the subinterval $[t_{\kappa\omega}, t_{\kappa\omega+\omega}]$ does not require the computation of $q\omega$ iterates, but only $q + \omega - 1$ iterates, so that the number of *effective* (or *sequential*) WR iterations per step is $1 + \omega^{-1}(q - 1)$. Here, each iterate has dimension sd. Note that this holds for any splitting function ψ.

There is an additional amount of intrinsic parallelism if the splitting function ψ is such that J^* and K^* are σ-by-σ block-diagonal. In such cases, the IVP can be decoupled into a set of σ subsystems of the form (2.3) each of which can be integrated by the method $\{A, E, \boldsymbol{c}\}$ defined in (1.2). Since these integrations can be done concurrently, the strategy described above can be applied to each subsystem. Thus, the effective costs per step reduce to the computation of $1 + \omega^{-1}(q - 1)$ WR iterates of dimension sd^*, where d^* is the maximal dimension of the subsystems.

2.2. *The Newton iteration process*

In an actual application of (2.4), each time step requires the solution of $\boldsymbol{Y}^{(k)}_n$ from the (nonlinear) system $\boldsymbol{R}(\boldsymbol{Y}^{(k)}_n, \boldsymbol{Y}^{(k)}_{n-1}, \boldsymbol{Y}^{(k-1)}_n) = \boldsymbol{0}$. Given the WR iteration index k and the time step index n, we shall use the following iteration process:

```
$\boldsymbol{Y}^{(k,0)}_n := \boldsymbol{Y}^{(k-1,m)}_n$,
    for $j = 1$ to $m$
        solve $\boldsymbol{Y}^{(k,j)}_n$ from                                  (2.5)
        $N_0(\boldsymbol{Y}^{(k,j)}_n - \boldsymbol{Y}^{(k,j-1)}_n)$
            $= -h(A \otimes I)\boldsymbol{R}(\boldsymbol{Y}^{(k,j-1)}_n, \boldsymbol{Y}^{(k)}_{n-1}, \boldsymbol{Y}^{(k-1,m)}_n)$,
```

where N_0 is the (modified) Newton matrix

$$N_0 := I \otimes K^* - A \otimes hJ^*. \tag{2.6}$$

Here, the Jacobian matrices K^* and J^* of the splitting function ψ are both evaluated at the step point t_{n-1}. The modified Newton process (2.5), will be assumed to be convergent.

The combination of the WR iteration method (2.4) and the modified Newton method {(2.5), (2.6)} is a nested iteration process containing four loops with indices $\kappa,\ k,\ n$ and j. The three iteration parameters $q,\ \omega$, and m determine the range of the indices $k,\ n$ and j. The number of effective modified Newton iterations (i.e., linear system solves) per step in {(2.4), (2.5)} is given by $m(1 + \omega^{-1}(q-1))$.

Remark 2.1. In practice, it may be an efficient strategy to perform only a few Newton iterations, because the WR iterate $\boldsymbol{Y}_n^{(k)}$ may still be far away from the solution $\boldsymbol{Y}_n$ of (1.2). Hence, it seems a waste to perform many Newton iterations for computing a close approximation to $\boldsymbol{Y}_n^{(k)}$, which itself is a poor approximation to $\boldsymbol{Y}_n$. In the extreme case where $m = 1$, the method {(2.4), (2.5)} reduces to

```
for $k = 1$ to $q$
    $\boldsymbol{Y}_{\kappa\omega}^{(k)} := \boldsymbol{Y}_{\kappa\omega}^{(q)}$
    for $n = \kappa\omega + 1$ to $\kappa\omega + \omega$
        solve $\boldsymbol{Y}_n^{(k)}$ from                                   (2.7)
        $N_0(\boldsymbol{Y}_n^{(k)} - \boldsymbol{Y}_n^{(k-1)})$
            $= -h(A \otimes I)\boldsymbol{R}(\boldsymbol{Y}_n^{(k-1)}, \boldsymbol{Y}_{n-1}^{(k)}, \boldsymbol{Y}_n^{(k-1)})$,
        set $\boldsymbol{y}_n^{(k)} = (\boldsymbol{e}_s^T \otimes I)\boldsymbol{Y}_n^{(k)}$.
```

A comparison with {(2.4), (2.5)} shows that in (2.7) we have a more frequent updating of the righthand side, so that it is expected that (2.7) shows a better overall convergence than {(2.4), (2.5)} with $m > 1$, that is, for constant qm, the accuracy is expected to be best for $m = 1$. However, it should also be observed that small m implies more frequent communication when implemented on a parallel computer system, so that given the number of WR iterations q, the effective costs for $m = 1$ and $m = 2$ or $m = 3$ may well be comparable.

Let us consider the case where the matrices K^* and J^* are σ-by-σ lower block-triangular matrices (K_{ij}^*) and (J_{ij}^*). In order to see the amount of parallelism inherent in the resulting modified Newton matrix we reorder the rows and columns in N_0. Let the partitioning of the vector $\boldsymbol{y}$ in (1.1) corresponding to the blocks (K_{ij}^*) and (J_{ij}^*) be denoted by $\boldsymbol{y} = (\boldsymbol{u}^T, \boldsymbol{v}^T, \ldots)^T$, and let us replace the sd-dimensional vectors $\boldsymbol{Y}$ in (2.5) by permuted vectors

$$\widetilde{\boldsymbol{Y}} = P\boldsymbol{Y} := (\boldsymbol{U}^T, \boldsymbol{V}^T, \ldots)^T,$$

where P is such that $\boldsymbol{U}, \boldsymbol{V}, \ldots$ are stage vectors associated with $\boldsymbol{u}, \boldsymbol{v}, \ldots$ in the same way as $\boldsymbol{Y}$ is associated with $\boldsymbol{y}$. Then the permuted version of the linear system in (2.5) becomes

$$\begin{aligned}&\widetilde{N}_0\big(\widetilde{\boldsymbol{Y}}_n^{(k,j)} - \widetilde{\boldsymbol{Y}}_n^{(k,j-1)}\big) = -P(hA \otimes I)\boldsymbol{R}\big(\boldsymbol{Y}_n^{(k,j-1)}, \boldsymbol{Y}_{n-1}^{(k)}, \boldsymbol{Y}_n^{(k-1,m)}\big),\\ &\widetilde{N}_0 := PN_0P^{-1}.\end{aligned} \tag{2.5$'$}$$

It is easily verified that for any matrix C and any σ-by-σ block matrix $M = (M_{ij})$, the matrix $P(C \otimes M)P^{-1}$ becomes a σ-by-σ block matrix with entries $C \otimes M_{ij}$. Hence,

$$\begin{aligned}\widetilde{N}_0 &:= (I \otimes K_{ij}^*) - (A \otimes hJ_{ij}^*)\\ &= \begin{pmatrix} I \otimes K_{11}^* - A \otimes hJ_{11}^* & O & \ldots \\ I \otimes K_{21}^* - A \otimes hJ_{21}^* & I \otimes K_{22}^* - A \otimes hJ_{22}^* & \ldots \\ \vdots & \vdots & \vdots \end{pmatrix}.\end{aligned} \tag{2.6$'$}$$

This expression shows that solving (2.5) by a direct method requires the LU decomposition of the σ diagonal blocks $I \otimes K_{ii}^* - A \otimes hJ_{ii}^*$. Hence, there are σ LU decompositions to be performed which can all be done in parallel. The maximal dimension of the matrices to be decomposed equals sd^*, d^* denoting the dimension of the largest blocks in K^* and J^*, so that the effective LU costs on σ processors is O$((sd^*)^3)$, each time the matrix N_0 in (2.5) is updated. Apart from these LU costs, each modified Newton iteration requires the evaluation of the function $\boldsymbol{R}$ and a forward/backward substitution. The evaluation of $\boldsymbol{R}$ can again be distributed over σ processors.

2.3. *Iterative solution of the Newton systems*

The LU decompositions needed in the modified Newton process may be costly if d^* is still large. Therefore, the linear Newton systems in (2.5) will be solved iteratively by an *inner* iteration process (in this connection, we may interpret the Newton process $\{(2.5), (2.6)\}$ as an *outer* iteration process). We shall use the iteration method

```
$\boldsymbol{U}^{(0)} := \boldsymbol{Y}_n^{(k,j-1)}$
    $\boldsymbol{C}^{(k,j)} := N_0\boldsymbol{Y}_n^{(k,j-1)} - h(A \otimes I)\boldsymbol{R}(\boldsymbol{Y}_n^{(k,j-1)}, \boldsymbol{Y}_{n-1}^{(k)}, \boldsymbol{Y}_n^{(k-1,m)})$        (2.9)
    for $\nu = 1$ to $r$
        solve $\boldsymbol{U}^{(\nu)}$ from $N(\boldsymbol{U}^{(\nu)} - \boldsymbol{U}^{(\nu-1)}) = -N_0\boldsymbol{U}^{(\nu-1)} + C^{(k,j)}$,
```

where the iteration matrix N is chosen such that the linear system for the inner iterate $\boldsymbol{U}^{(\nu)}$ is easily solved and where r is chosen such that $\boldsymbol{U}^{(r)}$ is an "acceptable" approximation to $\boldsymbol{Y}_n^{(k,j)}$. Evidently, if (2.8) converges as $r \to \infty$, then $\boldsymbol{U}^{(r)}$ converges to the solution $\boldsymbol{Y}_n^{(k,j)}$ of (2.5) irrespective the choice for N. However, as we will see in the experiments, it is possible to choose "convenient" matrices N such that in an actual computation, one or two inner iterations are sufficient (see section 4). In fact, we shall define N by

$$N := I \otimes K^* - T \otimes hJ^*, \tag{2.10}$$

where T is lower triangular with positive diagonal entries (cf. [12,13]). In order to see the intrinsic parallelism of the inner iteration process, we proceed as in the preceding section. Again assuming that K^* and J^* are both lower block-triangular, we obtain the (permuted) iteration matrix

$$\begin{aligned}\widetilde{N} &:= (I \otimes K^*_{ij}) - (T \otimes hJ^*_{ij}) \\ &= \begin{pmatrix} I \otimes K^*_{11} - T \otimes hJ^*_{11} & O & \dots \\ I \otimes K^*_{21} - T \otimes hJ^*_{21} & I \otimes K^*_{22} - T \otimes hJ^*_{22} & \dots \\ \vdots & \vdots & \vdots \end{pmatrix}.\end{aligned} \tag{2.9$'$}$$

Hence, (2.8) requires the LU decomposition of the σ matrices $I \otimes K^*_{ii} - T \otimes hJ^*_{ii}$. But, since T is also lower triangular, the LU decomposition of each of these matrices falls apart into the LU decomposition of the s matrices $K^*_{ii} - T_{jj}hJ^*_{ii}$, $j = 1, \dots, s$, which can all be done in parallel. The maximal dimension of the matrices to be decomposed equals d^*, so that the computational complexity is reduced to O($(d^*)^3$), provided that $s\sigma$ processors are available (if only p processors are available, with $p < s\sigma$, then effectively, the computational complexity is about O($s\sigma p^{-1}(d^*)^3$)). Apart from these LU decompositions, each inner iteration again requires a forward/backward substitution. Furthermore, by diagonalizing T by a Butcher transformation, the forward/backward substitution can be distributed over s processors. If K^* and J^* are both *block-diagonal*, then even the forward/backward substitution can be distributed over $s\sigma$ processors.

3. Convergence results

In this section, we study the convergence of the inner iteration method (2.8) and, for linear IVPs, the convergence of the (discrete) WR iteration method (2.4). We recall that the outer iteration process, that is, the modified Newton process (2.5), is always assumed to be convergent.

3.1. *The inner iteration method*

The convergence of (2.8) can be studied by deriving the error recursion for $\boldsymbol{U}^{(\nu)} - \boldsymbol{Y}_n^{(k,j)}$, i.e.,

$$\begin{aligned} &\boldsymbol{U}^{(\nu)} - \boldsymbol{Y}_n^{(k,j)} = M_1\big(\boldsymbol{U}^{(\nu-1)} - \boldsymbol{Y}_n^{(k,j)}\big), \\ &M_1 := (I \otimes K^* - T \otimes hJ^*)^{-1}\big((A - T) \otimes hJ^*\big). \end{aligned} \tag{3.1}$$

For convergence, the spectral radius $\rho(M_1)$ of M_1 should be less than 1. In [13], amplification matrices of the type M_1 have been analysed and led to the following definition and convergence theorem:

Definition 3.1. Let

$$Z(z) := z(I - zT)^{-1}(A - T). \tag{3.2}$$

Then, $\mathbb{B}(A)$ is the set of lower triangular matrices T such that the spectrum of $Z(z)$ is within the unit circle for Re$(z) \leqslant 0$.

Theorem 3.1. Let N be defined as in (2.9) with $T \in \mathbb{B}(A)$. Then, the inner iteration process (2.8) converges for all $h > 0$ if, and only if, $\{K^*, J^*\}$ is stable.

For the construction of lower triangular matrices T that are in $\mathbb{B}(A)$, we refer to [12].

3.2. The discrete WR iteration method

The convergence of discrete WR methods of RK type of the form (2.3) has extensively been studied, in particular for the ODE case where $K = K^* = I$ (see, e.g., [2,6], and the references in [2]). For linear problems, where second-order terms in the error recursion can be ignored, the convergence analysis is quite straightforward. In this section, we give a brief derivation of a few convergence results.

From (2.4) and (1.2) it follows that for linear problems the WR iteration error $\boldsymbol{Y}_n^{(k)} - \boldsymbol{Y}_n$ satisfies the recursion

$$\begin{aligned} &\boldsymbol{Y}_n^{(k)} - \boldsymbol{Y}_n = M_2\big(\boldsymbol{Y}_n^{(k-1)} - \boldsymbol{Y}_n\big) + M_3\big(\boldsymbol{Y}_{n-1}^{(k)} - \boldsymbol{Y}_{n-1}\big), \\ &M_2 := N_0^{-1}\big(I \otimes (K^* - K) - A \otimes h(J^* - J)\big), \\ &M_3 := N_0^{-1}(E \otimes K), \quad N_0 := I \otimes K^* - A \otimes hJ^*. \end{aligned} \tag{3.3}$$

This recursion is of a similar form as the error recursion of the PDIRKAS GS method analysed in [11] and can be represented as

$$\varepsilon^{(k)} = Q^k \varepsilon^{(0)}, \quad \varepsilon^{(k)} := \begin{pmatrix} \boldsymbol{Y}_{\kappa\omega+1}^{(k)} - \boldsymbol{Y}_{\kappa\omega+1} \\ \boldsymbol{Y}_{\kappa\omega+2}^{(k)} - \boldsymbol{Y}_{\kappa\omega+2} \\ \vdots \\ \boldsymbol{Y}_{\kappa\omega+\omega}^{(k)} - \boldsymbol{Y}_{\kappa\omega+\omega} \end{pmatrix}, \tag{3.4}$$

$$Q := \begin{pmatrix} M_2 & O & O & O & \dots \\ M_3M_2 & M_2 & O & O & \dots \\ M_3^2M_2 & M_3M_2 & M_2 & O & \dots \\ M_3^3M_2 & M_3^2M_2 & M_3M_2 & M_2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}.$$

Hence, we have convergence if the spectral radius $\rho(Q)$ of Q is less than 1, i.e., if $\rho(M_2) < 1$. An estimate for $\rho(M_2)$ can be obtained along the lines of a similar approach as in [13]. Theorem 3.2 presents conditions for convergence using the logarithmic matrix norm $\mu[\cdot]$ associated with the Euclidean norm $\|\cdot\|$, i.e., for any square matrix S, we have $\mu[S] = (1/2)\lambda_{\max}(S + S^H)$, where S^H is the complex transposed of S and $\lambda_{\max}(\cdot)$ denotes the algebraically largest eigenvalue.

Theorem 3.2. Let the IVP (1.1) be linear, let the spectrum $\sigma(A)$ of A be in the positive halfplane, and define (if K^* is nonsingular)

$$\widetilde{K} := (K^*)^{-1}K, \qquad \widetilde{J} := (K^*)^{-1}J, \qquad \widetilde{J}^* := (K^*)^{-1}J^*. \tag{3.5}$$

Then, the WR iteration process (2.4) converges if one of the following three conditions is satisfied for all $\alpha \in \sigma(A)$:

$$\big\|(K - K^*) - \alpha h(J - J^*)\big\| < -\mu[-K^* + \alpha hJ^*], \tag{3.6a}$$

$$\big\|\big(\widetilde{K} - I\big) - \alpha h\big(\widetilde{J} - \widetilde{J}^*\big)\big\| < \frac{\mathrm{Re}(\alpha)}{|\alpha|} - h|\alpha|\mu\big[\widetilde{J}^*\big], \quad K^* \text{ nonsingular}, \tag{3.6b}$$

$$\big\|(J^*)^{-1}J - I\big\| < \frac{\mathrm{Re}(\alpha)}{|\alpha|}, \qquad \mu\big[\widetilde{J}^*\big] \leqslant 0,$$
$$K^* = K, \quad K^* \text{ and } J^* \text{ nonsingular}. \tag{3.6c}$$

Proof. Let the eigenvectors and eigenvalues of M_2 be denoted by $\boldsymbol{a} \otimes \boldsymbol{w}$ and $\widetilde{\mu}$, where $\boldsymbol{a}$ is an eigenvector of A with eigenvalue α. Then,

$$(K^* - \alpha hJ^*)^{-1}\big((K^* - K) - \alpha h(J^* - J)\big)\boldsymbol{w} = \widetilde{\mu}\boldsymbol{w}, \tag{3.7}$$

so that

$$\rho(M_2) < \big\|(K - K^*) - \alpha h(J - J^*)\big\|\big\|(K^* - \alpha hJ^*)^{-1}\big\|.$$

By virtue of a property of the logarithmic norm, we have for any nonsingular, complex matrix C with $\mu[-C] < 0$, the estimate $\|C^{-1}\| < -(\mu[-C])^{-1}$. Hence, if $\mu[-K^* + \alpha hJ^*] < 0$, then $\|(K^* - \alpha hJ^*)^{-1}\| < -(\mu[-K^* + \alpha hJ^*])^{-1}$. This leads to condition (3.6a). Note that here K^* is allowed to be singular.

If K^* is nonsingular, then we may write

$$\rho(M_2) < \big\|\big(\alpha^{-1} - h\widetilde{J}^*\big)^{-1}\big\|\big\|\alpha^{-1}\big(\widetilde{K} - I\big) - h\big(\widetilde{J} - \widetilde{J}^*\big)\big\|.$$

Proceeding as above, we derive (3.6b).

Finally, we consider the case where $K^* = K$ and where both K^* and J^* are nonsingular. From (3.7) we derive the inequality

$$\rho(M_2) < \big\|\big(I - \alpha h\widetilde{J}^*\big)^{-1}\big(\alpha h\widetilde{J}^*\big)\big\|\big\|I \otimes \big((J^*)^{-1}J - I\big)\big\|.$$

In this case, we use a theorem of von Neumann. Von Neumann's theorem states that, given a matrix X with $\mu[X] \leqslant 0$ and a rational function R of z which is bounded in the lefthand halfplane $\mathrm{Re}(z) \leqslant 0$, then with respect to the Euclidean norm, the value of $\|R(X)\|$ is bounded by the maximum of $\{|R(z)|\colon \mathrm{Re}(z) \leqslant 0\}$ (see, e.g., [5, p. 179]). Thus, assuming that $\mu_2[\widetilde{J}^*] \leqslant 0$, condition (3.6c) follows from

$$\big\|\big(I - \alpha h\widetilde{J}^*\big)^{-1}\big(\alpha h\widetilde{J}^*\big)\big\| \leqslant \max_{\mathrm{Re}(z)\leqslant 0}\big|z\alpha(1 - z\alpha)^{-1}\big| = \frac{|\alpha|}{\mathrm{Re}(\alpha)}. \qquad \square$$

Let us compare the convergence conditions of this theorem for the particular case where $K^* = K$. Then (3.6) simplifies to

$$\|J - J^*\| < -\frac{1}{h|\alpha|}\mu[-K + \alpha hJ^*], \tag{3.7a}$$

$$\big\|K^{-1}(J - J^*)\big\| < \frac{\mathrm{Re}(\alpha)}{h|\alpha|^2} - \mu\big[K^{-1}J^*\big], \quad K \text{ nonsingular}, \tag{3.7b}$$

$$\big\|(J^*)^{-1}J - I\big\| < \frac{\mathrm{Re}(\alpha)}{|\alpha|}, \quad \mu\big[K^{-1}J^*\big] \leqslant 0, \qquad K \text{ and } J^* \text{ nonsingular}. \tag{3.7c}$$

The conditions (3.7a), (3.7b) and (3.7c) respectively provide an absolute estimate, a scaled absolute estimate and a relative estimate for the difference between J and J^*. Note that condition (3.7c) implies unconditional convergence with respect to h. For example, for the four-stage Radau IIA corrector, we have unconditional convergence if $\|(J^*)^{-1}J - I\| < 0.56$. If A has its eigenvalues in the positive halfplane, then condition (3.7b) shows that unconditional convergence is also possible if $\|K^{-1}(J - J^*)\| < -\mu[K^{-1}J^*]$.

4. Numerical experiments

The crucial aspect of the iteration process $\{(2.4), (2.5)\}$, is the convergence behaviour for splitting functions ψ for which the matrix N_0 allows a fast solution of the associated linear systems. Equally crucial is the effect of the number of inner and outer iterations r and m, and the window length ωh.

In this section, we illustrate the performance for a few test problems.

For the predictor we chose the "last step point" formula $\boldsymbol{Y}_n^{(0)} = \boldsymbol{e} \otimes \boldsymbol{y}_{n-1}$, and we used the four-stage Radau IIA corrector whose Butcher matrix is (within 14 digits) given by

$$A = \begin{pmatrix} 0.11299947932316 & -0.04030922072352 & 0.02580237742034 & -0.0099046765073 \\ 0.23438399574740 & 0.20689257393536 & -0.04785712804854 & 0.01604742280652 \\ 0.21668178462325 & 0.40612326386737 & 0.18903651817006 & -0.02418210489983 \\ 0.22046221117677 & 0.38819346884317 & 0.32884431998006 & 0.06250000000000 \end{pmatrix}$$

Following [12], we choose for the matrix T the lower triangular factor L of the Crout decomposition LU of A, i.e.,

$$\begin{aligned} T &= L \\ &= \begin{pmatrix} 0.11299947932312 & 0 & 0 & 0 \\ 0.23438399574745 & 0.29050212926461 & 0 & 0 \\ 0.21668178462320 & 0.48341807916606 & 0.30825766001501 & 0 \\ 0.22046221117877 & 0.46683683945825 & 0.44141588145851 & 0.11764705882353 \end{pmatrix}. \end{aligned} \tag{4.1}$$

This choice implies that the amplification matrix $Z(z)$ defined in (3.2) becomes strictly upper triangular at infinity, i.e., $Z(\infty) = I - T^{-1}A = I - U$. As a consequence, the stiff iteration error components are strongly damped in the iteration process. Moreover, we verified numerically that the matrix T given in (4.1) lies in $\mathbb{B}(A)$. Hence, it follows from theorem 3.1 that for each k and j the inner iterates $\boldsymbol{U}^{(\nu)}$ in $\{(2.8)\}$ unconditionally converge as $\nu \to \infty$ whenever the pair $\{K^*, J^*\}$ is stable. Note that there is no need to give the entries of T with extreme accuracy. As long as T lies in $\mathbb{B}(A)$, convergence is ensured (see definition 3.1).

In all experiments, constant step sizes have been used (if needed, we adapted the initial condition such that the integration starts outside the transient phase), and the matrices K and J were updated in each step. We recall that per update, the effective LU-costs are $\mathrm{O}((d^*)^3)$, where d^* is the maximal dimension of the diagonal blocks in the matrices K^* and J^*.

For given numbers of WR iterations q, outer iterations m, inner iterations r, and given window size ω, the tables of results present the minimal number of correct digits cd of the components of $\boldsymbol{y}$ at the end point $t = t_{\text{end}}$ of the integration interval (i.e., the absolute errors are written as $10^{-\text{cd}}$). We recall that the total number of effective inner iterations per step is given by $r_{\text{total}} = mr(1 + \omega^{-1}(q-1))$, which may serve as an estimate for the effective costs that are additional to the LU-costs. For the small window sizes used in practice and the usually large number of WR iterations needed to solve the IVP, we may approximately set $r_{\text{total}} = mrq\omega^{-1}$.

4.1. HIRES problem of Schäfer

Our first test problem is provided by the HIRES problem given in [5, p. 157] which originates from Schäfer [10] for explaining the "High Irradiance Responses" of photomorphogenesis (see also Gottwald [4] and the CWI testset [3]). This problem

Table 1
WR method $\{(2.4), (2.5), (2.8)\}$ applied to HIRES with block-Jacobi splitting (4.1a), $h = 15$ and $r = 1\backslash 2$.

ω	m	$q = 3$	$q = 5$	$q = 7$	$q = 9$	$q = 11$	$q = 13$	$q = 15$	$q = \infty$
1	1	1.4\1.9	2.6\3.6	3.7\5.7	4.9\6.2	6.1\7.0	7.8\8.2	7.9\7.9	7.9
	2	1.8\1.9	3.6\3.8	5.3\6.1	7.1\7.8	7.8\7.9	7.9\7.9		
	3	1.9\1.9	3.8\3.8	5.9\6.1	7.7\7.8	7.9\7.9			
2	1	1.0\1.2	2.0\2.6	3.0\4.1	4.0\6.1	5.1\6.4	6.4\7.9	7.4\8.0	
	2	1.2\1.2	2.5\2.6	4.0\4.2	5.5\6.0	7.1\7.6	7.8\7.9	7.9\7.9	
	3	1.2\1.2	2.6\2.6	4.2\4.2	5.9\6.0	7.5\7.6	7.8\7.9		
4	1	0.7\0.8	1.4\1.7	2.2\2.8	3.0\4.0	3.9\5.6	4.9\6.4	6.1\6.9	
	2	0.8\0.9	1.7\1.7	2.8\2.8	4.0\4.1	5.2\5.4	6.6\6.9	7.6\7.8	
	3	0.9\0.9	1.7\1.7	2.8\2.8	4.1\4.1	5.4\5.4	6.9\6.9	7.8\7.8	

was integrated over the interval $[5, 305]$. Writing the system as $\boldsymbol{y}' = \boldsymbol{f}(\boldsymbol{y})$, we may define the block-Jacobi splitting function

$$\psi(\boldsymbol{u}', \boldsymbol{v}', \boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}' - \boldsymbol{f}(\boldsymbol{u}) + 0.035(u_5 - v_5)\boldsymbol{e}_3 + 0.69(u_4 - v_4)\boldsymbol{e}_6, \tag{4.1a}$$

with the associated Jacobian splitting $K^* = K = I$ and

$$J^* = \begin{pmatrix} J_{11} & O \\ O & J_{22} \end{pmatrix}, \qquad J - J^* = \begin{pmatrix} O & J_{12} \\ J_{21} & O \end{pmatrix},$$

$$J_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.035 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad J_{21} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.69 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The results in table 1 show that the outer iteration process converges quite fast and that even a single outer iteration already produces a relatively high accuracy. The inner iteration process converges equally fast and two inner iterations usually suffices to find the modified Newton iterate. However, the WR iteration process requires relatively many iterations to reach the corrector solution, particularly on larger windows. Furthermore, note that for a constant total number of effective inner iterations $r_{\text{total}} = mrq\omega^{-1}$, the accuracy rapidly decreases as m increases (cf. remark 2.1). Thus, the best iteration strategy seems to be one outer iteration and one or two inner iterations.

The performance of the WR iteration can be improved if we apply block-Gauss–Seidel splitting:

$$\begin{gathered} \psi(\boldsymbol{u}', \boldsymbol{v}', \boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}' - \boldsymbol{f}(\boldsymbol{u}) + 0.035(u_5 - v_5)\boldsymbol{e}_3, \\ J^* = \begin{pmatrix} J_{11} & O \\ J_{21} & J_{22} \end{pmatrix}, \qquad J - J^* = \begin{pmatrix} O & J_{12} \\ O & O \end{pmatrix}. \end{gathered} \tag{4.1b}$$

Table 2 presents the analogue of table 1 and clearly shows the increased rate of convergence.

Table 2
WR method $\{(2.4), (2.5), (2.8)\}$ applied to HIRES with block Gauss–Seidel splitting (4.3b), $h = 15$ and $r = 1\backslash 2$.

ω	m	$q = 3$	$q = 5$	$q = 7$	$q = 9$	$q = 11$	$q = 13$	$q = 15$	$q = \infty$
1	1	3.2\3.8	4.2\4.7	5.1\5.5	5.8\6.3	6.6\7.2	7.5\8.2	7.9\7.9	7.9
	2	4.2\5.1	6.1\6.6	8.0\8.0	7.9\7.9	7.9\7.9	7.9\7.9		
	3	5.1\5.9	7.9\8.0	7.9\7.9					
2	1	3.1\3.6	4.1\4.6	4.9\5.4	5.6\6.2	6.4\7.0	7.2\8.1	7.9\7.9	
	2	4.1\5.2	5.8\6.3	7.4\8.2	7.9\7.9	7.9\7.9	7.9\7.9		
	3	5.3\4.7	7.1\8.2	7.9\7.9					
4	1	3.1\3.5	3.7\4.3	4.6\5.1	5.2\5.8	5.9\6.6	6.6\7.4	7.4\7.9	
	2	4.2\4.2	5.2\5.7	6.7\7.2	7.9\7.9	7.9\7.9	7.9\7.9	7.9\7.9	
	3	4.1\4.0	6.0\6.5	7.9\7.9					

4.2. *The transistor amplifier*

Our second test problem is the semi-explicit representation of the transistor amplifier given in [3]. It is a nonlinear, eight-dimensional problem of index 1 on the interval $[0, 0.2]$ given by

$$\dot{\boldsymbol{u}}(t) = \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{v}), \quad \boldsymbol{g}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{0}, \quad \boldsymbol{u}, \boldsymbol{f} \in \mathbb{R}^5,\ \boldsymbol{v}, \boldsymbol{g} \in \mathbb{R}^3, \tag{4.2}$$

so that

$$K = \begin{pmatrix} I & O \\ O & O \end{pmatrix}, \qquad J = \begin{pmatrix} \boldsymbol{f}_{\boldsymbol{u}} & \boldsymbol{f}_{\boldsymbol{v}} \\ \boldsymbol{g}_{\boldsymbol{u}} & \boldsymbol{g}_{\boldsymbol{v}} \end{pmatrix}.$$

The structure of K and J suggests the use of a block-Gauss–Seidel splitting with

$$K^* = K, \qquad J^* = \begin{pmatrix} \boldsymbol{f}_{\boldsymbol{u}} & O \\ \boldsymbol{g}_{\boldsymbol{u}} & \boldsymbol{g}_{\boldsymbol{v}} \end{pmatrix},$$

which reduces the effective costs of each LU-update by a factor $8^3/5^3 \approx 4$.

The results in table 3 show the same trends as in the preceding tables.

5. Summary and concluding remarks

The numerical integration method proposed in this paper is based on a Runge–Kutta type integration formula (1.2) which is solved iteratively by three nested iteration processes: the discrete WR process (2.4), the modified Newton process or outer iteration process (2.5), and the linear system solver or inner iteration process (2.8). It aims at the solution of IDEs of which the Jacobian matrices K and J are approximated by lower triangular σ-by-σ block matrices K^* and J^*. On $\omega\sigma s$ processors, the total effective cost per step approximately consists of carrying out $r_{\text{total}} = mrq\omega^{-1}$ inner iterations. Here, ω is the window length and q, m and r respectively denote the number of WR iterations, outer iterations and inner iterations. Each Jacobian update

Table 3
WR method $\{(2.4),(2.5),(2.8)\}$ applied to the transistor amplifier with the splitting (4.2), $h = 2 \times 10^{-4}$ and $r = 1\backslash 2$.

ω	m	$q = 3$	$q = 5$	$q = 7$	$q = 9$	$q = 11$	$q = 13$	$q = 15$	$q = \infty$
1	1	0.9\∗	1.0\1.7	1.8\3.3	2.8\4.8	3.8\5.9	4.9\7.3	6.0\8.7	9.7
	2	0.8\1.4	2.7\3.0	4.7\5.1	5.7\6.8	7.5\8.3	9.3\9.5	9.6\9.8	
	3	1.4\1.2	2.7\2.7	4.7\4.4	6.8\6.8	8.2\8.1	9.6\9.6	9.7\9.7	
2	1	0.3\0.4	0.2\0.3	0.3\0.6	0.5\1.0	0.7\1.7	0.9\2.3	1.2\2.1	
	2	0.6\0.8	2.7\1.8	2.6\2.8	2.7\3.8	3.2\4.9	3.8\5.8	4.3\6.5	
	3	0.3\0.3	0.9\1.5	2.0\2.0	2.2\3.1	3.1\4.3	5.1\5.1	4.5\7.6	
4	1	∗\0.2	0.7\0.4	0.4\0.6	0.5\0.9	0.6\1.2	0.7\1.6	0.9\2.0	
	2	0.5\0.7	∗\0.7	∗\1.7	∗\1.8	∗\2.1	∗\3.0	∗\4.1	
	3	0.6\0.4	0.7\1.0	2.0\1.8	2.0\2.2	2.4\3.1	3.1\4.3	4.9\4.7	

or change of step size requires s concurrent LU-decompositions of matrices of maximal dimension d^*, where d^* is the maximal blocksize occurring in K^* and J^*, that is, effectively only $\mathrm{O}((d^*)^3)$ operations per update. Furthermore, each inner iteration requires a forward/backward substitution of dimension $\leqslant sd^*$ which can be distributed over s processors, that is, only $\mathrm{O}(r_{\text{total}}(d^*)^2)$ operations per step.

The numerical experiments with the method $\{(2.4),(2.5),(2.8)\}$ presented in section 4 clearly show:

(i) The better the approximations K^* and J^*, the faster the convergence of the WR iterates.

(ii) One or two inner iterations are sufficient, i.e., $r \leqslant 2$.

(iii) For constant r_{total}, the accuracy is best if only one outer iteration is performed, i.e., $m = 1$.

In an actual implementation, the values of q, m and r should be determined dynamically during the integration process. At present, the full method $\{(2.4),(2.5),(2.8)\}$ is tested on a sequential computer system and only the case where $\omega = r = 1$ and $K = K^* = I$, $J^* = J$ (and hence $q = 1$) has been implemented on the four-processor Cray-C98/4256. The results reported in [12] show that with respect to the code RADAU5 of Hairer and Wanner [5], to be considered as one of the best sequential codes, the speed-ups are in the range $[2.4, 3.1]$. Implementation of the full method $\{(2.4),(2.5),(2.8)\}$ on the Cray-C98/4256 will be subject of future research.

References

[1] A. Bellen and F. Tagliaferro, A combined WR-parallel steps method for ordinary differential equations, in: *Proc. Conf. Parallel Computing: Achievements, Problems and Prospects*, Capri, Italy (1990).

[2] K. Burrage, *Parallel and Sequential Methods for Ordinary Differential Equations* (Clarendon Press, Oxford, 1995).

[3] W. M. Lioen, J. J. B. de Swart and W. A. van der Veen, Test set for IVP solvers, available via WWW at URL: http://www.cwi.nl/cwi/projects/IVPtestset.shtml (1996).

[4] B. A. Gottwald, MISS: A simple simulation system for biological and chemical processes, EDV in Medizin und Biologie 3 (1977) 85–90 (in German).

[5] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II; Stiff and Differential-Algebraic Problems,* Springer Series in Computational Mathematics 14 (Springer, Berlin, 1991).

[6] K. J. in 't Hout, On the convergence of waveform relaxation methods for stiff nonlinear ordinary differential equations, Appl. Numer. Math. 18 (1995) 175–190.

[7] E. Lelarasmee, The waveform relaxation method for the time domain analysis of large scale nonlinear dynamical systems, Ph.D. thesis, University of California, Berkeley (1982).

[8] E. Lelarasmee, A. E. Ruehli and A. L. Sangiovanni-Vincentelli, The waveform relaxation method for time domain analysis of large scale integrated circuits, IEEE Trans. CAD IC Syst. 1 (1982) 131–145.

[9] U. Miekkela and O. Nevanlinna, Convergence of dynamic iteration methods for initial value problems, SIAM J. Sci. Statist. Comput. 8 (1987) 459–482.

[10] E. Schäfer, A new approach to explain the “High Irradiance Responses” of photomorphogenesis on the basis of phytochrome, J. Math. Biology 2 (1975) 41–56.

[11] P. J. van der Houwen, B. P. Sommeijer and W. A. van der Veen, Parallelism across the steps in iterated Runge–Kutta methods for stiff initial value problems, Numer. Algorithms 8 (1994) 293–312.

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

[13] P. J. van der Houwen and W. A. van der Veen, Solving implicit differential equations on parallel computers, CWI Report NM-R9526 (1995), submitted for publication.

[14] J. K. White and A. L. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits* (Kluwer, Dordrecht, 1987).