Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Journal of Computational and Applied Mathematics 162 (2004) 431–444

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Preconditioned WR–LMF-based method for ODE systems☆

Xiao-qing Jin*, Vai-kuong Sin, Li-li Song

Faculty of Science and Technology, University of Macau, C. Postal 3001, Macau, China

Received 13 September 2001; received in revised form 2 May 2003

Abstract

The waveform relaxation (WR) method was developed as an iterative method for solving large systems of ordinary differential equations (ODEs). In each WR iteration, we are required to solve a system of ODEs. We then introduce the boundary value method (BVM) which is a relatively new method based on the linear multistep formulae to solve ODEs. In particular, we apply the generalized minimal residual method with the Strang-type block-circulant preconditioner for solving linear systems arising from the application of BVMs to each WR iteration. It is demonstrated that these techniques are very effective in speeding up the convergence rate of the resulting iterative processes. Numerical experiments are presented to illustrate the effectiveness of our methods.

MSC: 15A18; 65F10; 65F15; 65L05; 65N22

Keywords: ODE; WR method; BVM; GMRES method; Block-circulant preconditioner

1. Introduction

Consider the following linear ordinary differential equations (ODEs):

$$\begin{aligned} &\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} + Q\mathbf{y}(t) = \mathbf{g}(t), \quad t \in (t_0, T], \\ &\mathbf{y}(t_0) = \mathbf{z}, \end{aligned} \tag{1}$$

☆ The research is supported by the Research Grants RG026/00-01S/JXQ/FST and RG024/01-02S/JXQ/FST from University of Macau.
* Corresponding author.
E-mail address: xqjin@umac.mo (X.-q. Jin).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2003.08.034

where $\mathbf{y}(t)$, $\mathbf{g}(t):\mathbb{R}\to\mathbb{R}^m$, $\mathbf{z}\in\mathbb{R}^m$ and $Q\in\mathbb{R}^{m\times m}$. By splitting the matrix Q as

$$Q = M + N, \tag{2}$$

we then construct an iteration of the form

$$\begin{aligned} &\frac{\mathrm{d}\mathbf{y}^{(k+1)}(t)}{\mathrm{d}t} + M\mathbf{y}^{(k+1)}(t) = -N\mathbf{y}^{(k)}(t) + \mathbf{g}(t), \quad t\in(t_0,T],\\ &\mathbf{y}^{(k+1)}(t_0) = \mathbf{z}, \end{aligned} \tag{3}$$

where $k=0,1,\ldots$, and $\mathbf{y}^{(0)}$ is a given initial guess usually given by $\mathbf{y}^{(0)}(t)=\mathbf{z}$ for $t\in[t_0,T]$. Iteration (3) is called the waveform relaxation (WR) method and is also called dynamic iteration, see [5]. The fact that WR methods are specifically designed for parallel algorithms has been discussed in [4]. This paper, however, presents a different point of view from [4] by using sequential algorithms. The reasons for adopting sequential algorithms are that sometimes parallel computing is not available on the one hand, and WR methods in parallel computing are very much machine-dependent on the other. Analysis of WR methods in sequential algorithms is also helpful for the justification of the efficiency of any proposed parallel algorithms. We recall that the WR method is originated from electrical network simulation, see [12]. It differs from standard iterative techniques in that it is a continuous-in-time analogue of stationary method by iterating with functions.

The Jacobi, Gauss–Seidel and successive overrelaxation (SOR) versions of the WR technique are classical methods. To be precise, the matrix Q is decomposed as

$$Q = L + D + U,$$

where D is a diagonal matrix, L is a strictly lower triangular matrix and U is a strictly upper triangular matrix. The splittings

$$M = D, \qquad N = L + U,$$

$$M = D + L, \qquad N = U$$

and

$$M = (D + \omega L)/\omega, \qquad N = Q - M \quad \text{with } \omega > 0,$$

define, respectively, the Jacobi, Gauss–Seidel and SOR versions of WR iterations, where ω is relaxation parameter.

In this paper, we mainly consider the case of Q in (2) being Toeplitz (a matrix is said to be Toeplitz if its entries are constant along its diagonals). Such kind of ODE system often appears in numerical partial differential equations see [2,7]. By using the well-known circulant and skew-circulant decomposition of Toeplitz matrix, we decompose the matrix Q as $Q=M+N$, where M is a circulant matrix and N is a skew-circulant matrix see [6,8,18]. More precisely, for a Toeplitz matrix

$$T_n = [t_{i-j}]_{i,j=1}^{n} = [t_k]_{k=-n+1}^{n-1},$$

we can decompose the matrix T_n as

$$T_n = C_n + S_n, \tag{4}$$

where

$$C_n = \begin{bmatrix} c_0 & c_1 & \cdots & c_{n-2} & c_{n-1} \\ c_{n-1} & c_0 & \ddots & \ddots & c_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_2 & \ddots & \ddots & c_0 & c_1 \\ c_1 & c_2 & \cdots & c_{n-1} & c_0 \end{bmatrix}, \qquad S_n = \begin{bmatrix} s_0 & s_1 & \cdots & s_{n-2} & s_{n-1} \\ -s_{n-1} & s_0 & \ddots & \ddots & s_{n-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -s_2 & \ddots & \ddots & s_0 & s_1 \\ -s_1 & -s_2 & \cdots & -s_{n-1} & s_0 \end{bmatrix}$$

with

$$c_0 + s_0 = t_0, \quad c_k = \tfrac{1}{2}(t_k + t_{-n+k}), \quad s_k = \tfrac{1}{2}(t_k - t_{-n+k}), \quad k = 1, 2, \ldots, n-1.$$

The WR method with this new scheme is called the $C + S$ version.

The convergence behavior of WR methods has been studied exhaustively in a series of papers [13–15], where the authors formulated the convergence characteristics of the method in terms of the spectral radius of the corresponding waveform relaxation operator. To accelerate WR iterations, the multigrid technique was studied in [17] and the preconditioning technique was discussed in [5].

In this paper, we apply boundary value methods (BVMs) see [3,11] to each WR iteration (3). This method requires the solution of nonsymmetric linear systems that are often large and sparse. We then propose to use the generalized minimal residual (GMRES) method [9,16] with the Strang-type block-circulant preconditioner [7,10] to solve these linear systems. The main purpose of this paper is to investigate the effectiveness of preconditioning technique on the speed of the resulting iterative processes.

The paper is organized as follows. In Section 2, we briefly study the error estimate of WR iterations. In Section 3, we introduce the BVM and the Strang-type preconditioner. The invertibility of the Strang-type preconditioner is discussed in Section 4. The convergence rate and operation cost of the preconditioned GMRES method are studied in Section 5, and, finally, numerical experiments are given in Section 6 to illustrate the effectiveness of our methods.

2. Error bound of WR iterations

In this section, We shall consider the space of continuous functions $\mathscr{C}[t_0, T]$ equipped with the sup-norm:

$$\|\mathbf{y}\|_T \equiv \sup\{\|\mathbf{y}(t)\| : t \in [t_0, T]\},$$

where $\mathbf{y}(t) : \mathbb{R} \to \mathbb{R}^m$ and $\|\cdot\|$ is some norm on $\mathbb{R}^m$. Specially, in the following, we use the norm $\|\cdot\|_2$. It is well-known (see, e.g., [14]) that if the iterations $\{\mathbf{y}^{(k)}\}_{k=0}^{\infty}$ defined by (3) converge, the

error of the kth iteration $\mathbf{y}^{(k)}$ can be bounded by

$$\|\mathbf{y}-\mathbf{y}^{(k)}\|_T \leqslant \frac{(C(T-t_0))^k}{k!}\,\|\mathbf{y}-\mathbf{y}^{(0)}\|_T,$$

where $\mathbf{y}$ is the true solution of (1) and C is a constant such that

$$\|\mathrm{e}^{-Mt}N\|_T \leqslant C.$$

Here $\|\cdot\|_T$ denotes the corresponding induced matrix norm. However, if C is large, then the convergence of finding $\mathbf{y}$ through iteration could be very slow. Usually, it is not easy to estimate C, see [5,14].

In the following, we will show that when the Jacobian matrix Q is Toeplitz, the bound of $\|\mathrm{e}^{-Mt}N\|_T$ is easy to be calculated for the $C+S$ version of the WR iteration. It is well-known that any $m\times m$ circulant matrix can be diagonalized by the $m\times m$ Fourier matrix F_m, see [6,10]. Hence, we have the following decomposition:

$$M = F_m \Lambda_M F_m^*, \tag{5}$$

where $\Lambda_M = \mathrm{diag}(\lambda_1,\lambda_2,\ldots,\lambda_m)$ is a diagonal matrix holding the eigenvalues of M. Since N is a skew-circulant matrix, it can also be diagonalized as

$$N = D_m F_m \Lambda_N F_m^* D_m^*, \tag{6}$$

where $\Lambda_N = \mathrm{diag}(\gamma_1,\gamma_2,\ldots,\gamma_m)$ is a diagonal matrix containing the eigenvalues of the matrix N and $D_m = \mathrm{diag}(1,\delta,\delta^2,\ldots,\delta^{m-1})$ with $\delta = \sqrt[m]{-1}$, see [6]. By using (5) and (6), we obtain

$$\begin{aligned}
\|\mathrm{e}^{-Mt}N\|_T &= \sup_{\|y\|_2=1}\sup_{t\in[t_0,T]} \|\mathrm{e}^{-F_m\Lambda_M F_m^* t} D_m F_m \Lambda_N F_m^* D_m^* y(t)\|_2 \\
&\leqslant \sup_{t\in[t_0,T]} \|F_m\,\mathrm{diag}\left(\sum_{n=0}^{\infty}(-\lambda_1 t)^n/n!,\ldots,\quad \sum_{n=0}^{\infty}(-\lambda_m t)^n/n!\right) F_m^* \\
&\quad \times D_m F_m\,\mathrm{diag}(\gamma_1,\ldots,\gamma_m) F_m^* D_m^*\|_2 \\
&\leqslant \sup_{t\in[t_0,T]} \{\|\mathrm{diag}(\mathrm{e}^{-\lambda_1 t},\ldots,\mathrm{e}^{-\lambda_m t})\|_2 \cdot \|\mathrm{diag}(\gamma_1,\ldots,\gamma_m)\|_2\} \\
&\leqslant \sup_{t\in[t_0,T]} \max_{1\leqslant i\leqslant m} |\mathrm{e}^{-\lambda_i t}| \max_{1\leqslant i\leqslant m} |\gamma_i| \\
&= \max_{1\leqslant i\leqslant m} |\mathrm{e}^{-\Re(\lambda_i)t_0}| \max_{1\leqslant i\leqslant m} |\gamma_i| = C,
\end{aligned}$$

where $\Re(\cdot)$ is the real part of a complex number.

3. BVM and Strang-type preconditioner

In WR iterations, we are required to solve a series of ODE systems (3) and therefore BVMs are proposed. BVMs are numerical methods based on the linear multistep formulae (LMF) for solving ODEs. Although initial value methods (IVMs) are more efficient than BVMs, the advantage in using BVMs over IVMs comes from their stability properties see for instance [3]. A BVM approximates the solution of (3) by means of a discrete boundary value problem. By using a v-step LMF over a uniform mesh

$$t_j = t_0 + jh, \quad j = 0, \ldots, s,$$

where $h = (T - t_0)/s$ is the step size, we have

$$\sum_{i=-v}^{\mu-v} \alpha_{i+v} \mathbf{y}_{n+i}^{(k+1)} = h \sum_{i=-v}^{\mu-v} \beta_{i+v} \mathbf{f}_{n+i}, \quad n = v, \ldots, s - \mu + v \tag{7}$$

with boundary values

$$\mathbf{y}_0^{(k+1)}, \ldots, \mathbf{y}_{v-1}^{(k+1)}, \quad \mathbf{y}_{s-\mu+v+1}^{(k+1)}, \ldots, \mathbf{y}_s^{(k+1)}. \tag{8}$$

In (7), $\mathbf{y}_n^{(k+1)}$ is an approximation to the true solution $\mathbf{y}^{(k+1)}(t_n)$ of (3) and

$$\mathbf{f}_n \equiv -M\mathbf{y}_n^{(k+1)} - N\mathbf{y}_n^{(k)} + \mathbf{g}(t_n)$$

with $\mathbf{y}_n^{(k)}$ obtained by the kth dynamic iteration.

For the boundary values (8), the dynamic iteration (3) only provides one initial condition $\mathbf{y}^{(k+1)}(t_0) = \mathbf{z}$. For the others, we can supply other two sets (initial and final) of additional difference equations with the same order of accuracy of (7) see [3], say,

$$\sum_{i=0}^{\mu} \alpha_i^{(j)} \mathbf{y}_i^{(k+1)} = h \sum_{i=0}^{\mu} \beta_i^{(j)} \mathbf{f}_i, \quad j = 1, \ldots, v-1 \tag{9}$$

and

$$\sum_{i=0}^{\mu} \alpha_{\mu-i}^{(j)} \mathbf{y}_{s-i}^{(k+1)} = h \sum_{i=0}^{\mu} \beta_{\mu-i}^{(j)} \mathbf{f}_{s-i}, \quad j = s - \mu + v + 1, \ldots, s. \tag{10}$$

Now, by combining (7), (9), (10) and the initial condition

$$\mathbf{y}_0^{(k+1)} = \mathbf{y}^{(k+1)}(t_0) = \mathbf{z},$$

we obtain a linear system

$$T\vec{\mathbf{y}}^{(k+1)} = G\vec{\mathbf{y}}^{(k)} + \vec{\mathbf{d}}, \tag{11}$$

where

$$T = A \otimes I_m + hB \otimes M,$$
$$\vec{\mathbf{y}}^{(k+1)} = ((\mathbf{y}_0^{(k+1)})^{\mathrm{T}}, \ldots, (\mathbf{y}_s^{(k+1)})^{\mathrm{T}})^{\mathrm{T}} \in \mathbb{R}^{(s+1)m},$$
$$G = -h(B \otimes N) \tag{12}$$

and

$$\vec{\mathbf{d}} = \mathbf{e_1} \otimes \mathbf{z} + h(B \otimes I_m)\vec{\mathbf{g}} \in \mathbb{R}^{(s+1)m}$$

with $\mathbf{e}_1 = (1, 0, \ldots, 0)^{\mathrm{T}} \in \mathbb{R}^{(s+1)}$ and $\vec{\mathbf{g}} = (\mathbf{g}(t_0), \mathbf{g}(t_1), \ldots, \mathbf{g}(t_s))^{\mathrm{T}} \in \mathbb{R}^{(s+1)m}$. In (12), $I_m \in \mathbb{R}^{m\times m}$ is the identity matrix, the matrix $A \in \mathbb{R}^{(s+1)\times(s+1)}$ is defined by

$$A \equiv \begin{bmatrix}
1 & \cdots & 0 & & & & \\
\alpha_0^{(1)} & \cdots & \alpha_\mu^{(1)} & & & & \\
\vdots & \vdots & \vdots & & & & \\
\alpha_0^{(v-1)} & \cdots & \alpha_\mu^{(v-1)} & & & 0 & \\
\alpha_0 & \cdots & \alpha_\mu & & & & \\
 & \alpha_0 & \cdots & \alpha_\mu & & & \\
 & & \ddots & \ddots & \ddots & & \\
 & & & \ddots & \ddots & \ddots & \\
 & & & & \alpha_0 & \cdots & \alpha_\mu \\
 & & 0 & & \alpha_0^{(s-\mu+v+1)} & \cdots & \alpha_\mu^{(s-\mu+v+1)} \\
 & & & & \vdots & \vdots & \vdots \\
 & & & & \alpha_0^{(s)} & \cdots & \alpha_\mu^{(s)}
\end{bmatrix}$$

and $B \in \mathbb{R}^{(s+1)\times(s+1)}$ is defined similarly by using $\{\beta_i\}_{i=0}^{\mu}$ instead of $\{\alpha_i\}_{i=0}^{\mu}$ in A and the first row of B is zero. Obviously, (11) is a classical stationary iteration for linear systems. It is well-known that the convergence of such an iteration requires $\rho(T^{-1}G) < 1$, where $\rho(\cdot)$ is the spectral radius. Since the size of the matrix T is very large when h is small and (or) m is large, if a direct method is used to solve the system (11), the operation cost can be very expensive, see numerical comparisons in [2,7].

Therefore Krylov subspace methods, such as the GMRES method [9,16], are proposed to solve the linear system (11). In order to speed up the convergence rate of Krylov subspace methods, we use the Strang-type block-circulant preconditioner. The Strang-type block-circulant preconditioner for T in (12) is defined by

$$S = s(A) \otimes I_m + hs(B) \otimes M, \tag{13}$$

where $s(A)$ is a circulant matrix given by

$$s(A) \equiv \begin{bmatrix} \alpha_v & \cdots & \alpha_\mu & & & \alpha_0 & \cdots & \alpha_{v-1} \\ \vdots & \ddots & & \ddots & & & \ddots & \vdots \\ \alpha_0 & & \ddots & & \ddots & & & \alpha_0 \\ & \ddots & & \ddots & & \ddots & 0 & \\ & & \ddots & & \ddots & & \ddots & \\ & 0 & & \ddots & & \ddots & & \ddots \\ \alpha_\mu & & & & \ddots & & \ddots & & \alpha_\mu \\ \vdots & \ddots & & & & \ddots & & \ddots & \vdots \\ \alpha_{v+1} & \cdots & \alpha_\mu & & & \alpha_0 & \cdots & \alpha_v \end{bmatrix}$$

and $s(B)$ is defined similarly by using $\{\beta_i\}_{i=0}^{\mu}$ instead of $\{\alpha_i\}_{i=0}^{\mu}$ in $s(A)$. Here $\{\alpha_i\}_{i=0}^{\mu}$ and $\{\beta_i\}_{i=0}^{\mu}$ are the coefficients in (7). When Q in (2) is Toeplitz, by using the decomposition (4), we choose M as a circulant matrix and N as a skew-circulant matrix.

4. Invertibility of Strang-type preconditioner

Now, we prove that under stability assumption on a given BVM, the Strang-type preconditioner is invertible. The stability of a BVM is closely related to two characteristic polynomials $\rho(z)$ and $\sigma(z)$ defined by

$$\rho(z) \equiv z^v \sum_{i=-v}^{\mu-v} \alpha_{i+v} z^i \quad \text{and} \quad \sigma(z) \equiv z^v \sum_{i=-v}^{\mu-v} \beta_{i+v} z^i,$$

where $\{\alpha_i\}_{i=0}^{\mu}$ and $\{\beta_i\}_{i=0}^{\mu}$ are given in (7). The $A_{v,\mu-v}$-stability polynomial is defined by

$$\pi(z,q) \equiv \rho(z) - q\sigma(z),$$

where $z, q \in \mathbb{C}$. Let

$$\mathbb{C}^- \equiv \{q \in \mathbb{C} : \Re(q) < 0\} \quad \text{and} \quad \mathbb{C}^+ \equiv \{q \in \mathbb{C} : \Re(q) > 0\}.$$

We have

Definition 1. The region

$$\mathscr{D}_{v,\mu-v} \equiv \{q \in \mathbb{C} : \pi(z,q) \text{ has } v \text{ zeros inside } |z| = 1 \text{ and } \mu - v \text{ zeros outside } |z| = 1\}$$

is called the region of $A_{v,\mu-v}$-stability of a given BVM and the BVM is said to be $A_{v,\mu-v}$-stable if

$$\mathbb{C}^- \subseteq \mathscr{D}_{v,\mu-v}.$$

Theorem 1. *If the BVM for* (3) *is* $A_{v,\mu-v}$*-stable and the eigenvalues of* M *satisfy*

$$\lambda_k(M) \in \mathbb{C}^+$$

for $k = 1,\ldots,m$*, then the preconditioner* S *defined by* (13) *is invertible.*

Proof. Follows the proof of Theorem 1 in [7] directly. □

If there is a $\lambda_l(M)$ which does not satisfy the condition in Theorem 1, say, $\lambda_l(M) \notin \mathbb{C}^+$, and M is assumed to be diagonalized by a unitary matrix (for instance, M is circulant), we can "move" $\lambda_l(M)$ into $\mathbb{C}^+$ by subtracting $\lambda_{\min} - \varepsilon$ from the main diagonal of the matrix M, where

$$\lambda_{\min} = \min_l \{\Re(\lambda_l(M)) : \lambda_l(M) \notin \mathbb{C}^+\}$$

and ε is a positive real number. After such a modification, a new matrix $\tilde{M}$ can be written as

$$\tilde{M} = M - (\lambda_{\min} - \varepsilon)I_m.$$

It yields a new decomposition of the matrix Q:

$$Q = \tilde{M} + \tilde{N},$$

where

$$\tilde{N} = N + (\lambda_{\min} - \varepsilon)I_m.$$

Obviously, all the eigenvalues of $\tilde{M}$ are now in $\mathbb{C}^+$ and therefore Theorem 1 is still applicable.

5. Convergence rate and operation cost

In this section, we first take a look at the convergence rate of the GMRES method. We know that if the preconditioned matrix is nonsingular and can be decomposed as

$$S^{-1}T = I + L,$$

where I is the identity matrix, T and S are defined by (12) and (13), respectively, the convergence rate of the GMRES method will be bounded by $\operatorname{rank}(L) + 1$, see [10]. We therefore have the following theorem.

Theorem 2. *The preconditioned matrix* $S^{-1}T$ *is nonsingular and can be decomposed as* $S^{-1}T = I + L$ *where* $\operatorname{rank}(L) \leqslant 2m\mu$*. If the GMRES method is applied to solving*

$$S^{-1}T\vec{\mathbf{y}}^{(k+1)} = S^{-1}\vec{\mathbf{b}},$$

the method will converge in at most $2m\mu + 1$ *iterations in exact arithmetic.*

Proof. Follows the proof of Theorem 2 in [7] directly. □

We should emphasize that the numerical examples in the next section show a much faster convergence rate than that predicted by the estimate provided by Theorem 2. Therefore, a detailed analysis for the convergence rate could be carried out in the future work.

Now, we consider the operation cost in each iteration of the GMRES method. Since $s(A)$ and $s(B)$ are both circulant matrices, we have the following decompositions:

$$s(A) = F_{s+1}\Lambda_A F_{s+1}^*, \quad s(B) = F_{s+1}\Lambda_B F_{s+1}^*, \tag{14}$$

where Λ_A and Λ_B are diagonal matrices containing the eigenvalues of $s(A)$ and $s(B)$, respectively, and F_{s+1} is the $(s+1)\times(s+1)$ Fourier matrix. It yields a decomposition of $S^{-1}(T\mathbf{v})$, i.e.,

$$S^{-1}(T\mathbf{v}) = (F_{s+1}\otimes I_m)(\Lambda_A \otimes I_m + h\Lambda_B \otimes M)^{-1}(F_{s+1}^* \otimes I_m)(T\mathbf{v}),$$

where $\mathbf{v}$ is some vector. This matrix–vector multiplication is the main work in each iteration of the GMRES method see, e.g. [6,10]. We then consider the following four cases for Toeplitz matrix:

$$Q = [q_{i-j}]_{i,j=1}^{m} = [q_k]_{k=-m+1}^{m-1},$$

(i) In the Jacobi version of WR iterations, since T is a block-Toeplitz matrix with Toeplitz blocks (plus a small rank perturbation), by using the Strang's embedding algorithm with the fast Fourier transform (FFT), see [6,10], $T\mathbf{v}$ can be computed within $\mathcal{O}(ms\log ms)$ operations. Meanwhile, note that

$$S = s(A)\otimes I_m + q_0 hs(B)\otimes I_m,$$

therefore, S^{-1} can be calculated within $\mathcal{O}(ms\log s)$ operations. Thus, it requires $\mathcal{O}(ms\log ms)$ operations to compute $S^{-1}(T\mathbf{v})$.

(ii) In the Gauss–Seidel version of WR iterations, we note that

$$\Lambda_A \otimes I_m + h\Lambda_B \otimes M$$

is a block diagonal matrix with lower triangular Toeplitz blocks. Therefore, we have to solve $s+1$ lower triangular Toeplitz systems of size $m\times m$. When using the superfast direct Toeplitz solver, see [1,18], it requires $\mathcal{O}(sm\log^2 m)$ operations to calculate $S^{-1}\mathbf{w}$ for some vector $\mathbf{w}$. As in (i), $T\mathbf{v}$ can also be computed within $\mathcal{O}(ms\log ms)$ operations. Therefore, it requires $\mathcal{O}(ms\log ms + ms\log^2 m)$ operations to compute $S^{-1}(T\mathbf{v})$.

(iii) For the SOR version of WR iterations, with a similar analysis given as in (ii), we know that it also requires $\mathcal{O}(ms\log ms + ms\log^2 m)$ operations to compute $S^{-1}(T\mathbf{v})$.

(iv) In the $C+S$ version of WR iterations, since the matrix M is a circulant matrix, by using (5) and (14), we have

$$S^{-1}(T\mathbf{v}) = (F_{s+1}\otimes F_m)(\Lambda_A \otimes I_m + h\Lambda_B \otimes \Lambda_M)^{-1}(F_{s+1}^* \otimes F_m^*)(T\mathbf{v}).$$

By using the FFT, S^{-1} can be calculated within $\mathcal{O}(ms\log ms)$ operations. As in (i), $T\mathbf{v}$ can also be computed within $\mathcal{O}(ms\log ms)$ operations. Therefore, it requires $\mathcal{O}(ms\log ms)$ operations to compute $S^{-1}(T\mathbf{v})$.

Consequently, by Theorem 2, the total complexity of each WR iteration is bounded by $\mathcal{O}(m^2 s\log ms)$ operations by using the $C+S$ version and the Jacobi version, while is bounded by $\mathcal{O}(m^2 s\log ms + m^2 s\log^2 m)$ operations by using the Gauss–Seidel version and the SOR version.

6. Numerical experiments

So far, we have introduced our method which combines WR iterations, the BVM and the GMRES method together with the Strang-type preconditioner for solving (1). To illustrate the efficiency of this method, in this section, two numerical examples are performed. From the preceding analysis, we specially choose diagonal dominant Toeplitz matrices as our Jacobian matrices in order to guarantee the convergence of WR iterations. The BVM we used here is the fifth order GAM [3], the following equations,

$$\mathbf{y}_n - \mathbf{y}_{n-1} = \frac{h}{720}(-19\mathbf{f}_{n-2} + 346\mathbf{f}_{n-1} + 456\mathbf{f}_n - 74\mathbf{f}_{n+1} + 11\mathbf{f}_{n+2}), \quad n = 2, \ldots, s-2$$

can be used with the following initial equation:

$$\mathbf{y}_1 - \mathbf{y}_0 = \frac{h}{720}(-19\mathbf{f}_4 + 106\mathbf{f}_3 - 264\mathbf{f}_2 + 646\mathbf{f}_1 + 251\mathbf{f}_0),$$

and the two final additional equations,

$$\mathbf{y}_{s-1} - \mathbf{y}_{s-2} = \frac{h}{720}(-19\mathbf{f}_s + 346\mathbf{f}_{s-1} + 456\mathbf{f}_{s-2} - 74\mathbf{f}_{s-3} + 11\mathbf{f}_{s-4}),$$

$$\mathbf{y}_s - \mathbf{y}_{s-1} = \frac{h}{720}(251\mathbf{f}_s + 646\mathbf{f}_{s-1} - 264\mathbf{f}_{s-2} + 106\mathbf{f}_{s-3} - 19\mathbf{f}_{s-4}).$$

All experiments are performed in MATLAB and we use the MATLAB-provided M-file "gmres" (see MATLAB on-line documentation) to solve the preconditioned systems. In our calculations, the stopping criterion in the GMRES method (which is applied within each WR iteration) is

$$\frac{\|\mathbf{r}_q\|_2}{\|\mathbf{r}_0\|_2} < 10^{-6},$$

where $\mathbf{r}_q$ is the residual after the qth iteration in the GMRES method and the zero vector is the initial guess. The stopping criterion of WR iterations is

$$\frac{\|\mathbf{y}^{(k+1)} - \mathbf{y}^{(k)}\|_2}{\|\mathbf{y}^{(k)}\|_2} \leqslant 10^{-6},$$

where $\mathbf{y}^{(k)}$ is the solution after the kth WR iteration. All programs are run on a 1.7 GHz PC with 512 Mbytes of memory.

Example 1. Consider

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} + Q\mathbf{y}(t) = 0, \quad t \in (0, 1],$$

$$\mathbf{y}(0) = (1, 2, \ldots, m)^{\mathrm{T}},$$

Table 1
Number of WR iterations for convergence in Example 1

m	s	$C+S$	Jacobi	Gauss–Seidel	SOR ($\omega = 0.987$)
20	16	11	18	11	11
	32	11	18	11	11
	64	11	18	11	11
	128	11	17	11	11
40	16	11	18	11	11
	32	11	19	11	11
	64	11	17	11	11
	128	10	17	11	11
60	16	11	18	11	11
	32	11	17	11	11
	64	11	17	11	11
	128	10	17	10	10

where

$$Q = \begin{bmatrix} 6 & -2 & 1 & & & \\ -2 & 6 & -2 & 1 & & \\ 1 & -2 & 6 & \ddots & \ddots & \\ & 1 & \ddots & \ddots & \ddots & 1 \\ & & \ddots & \ddots & \ddots & -2 \\ & & & 1 & -2 & 6 \end{bmatrix}.$$

Tables 1 and 2 show, respectively, the number of WR iterations and the number of megaflops (Mflops) required for convergence with different combinations of matrix sizes m and s. For the SOR method, an optimal value of ω has been tried to locate for a matrix with size $m = 20$ and $s = 16$ by calculating the spectral radius $\rho(T^{-1}G)$ versus relaxation parameter ω, and the result is given in Fig. 1. Fig. 1 shows that an optimal value of ω seems to be $\omega \approx 0.987$ with corresponding spectral radius $\rho \approx 0.041$. From Table 2, we found that the number of megaflops of the SOR method with $\omega = 0.987$ is the smallest one. As expected, the number of WR iterations required for convergence remains almost a constant for increasing m and s.

Example 2. Consider

$$\frac{\mathrm{d}\mathbf{y}(t)}{\mathrm{d}t} + Q\mathbf{y}(t) = 0, \quad t \in (0, 1],$$

$$\mathbf{y}(0) = (1, 2, \ldots, m)^{\mathrm{T}},$$

Table 2
Number of Mflops for convergence in Example 1

m	s	$C+S$	Jacobi	Gauss–Seidel	SOR ($\omega = 0.987$)
20	16	12.410	16.327	14.732	14.732
	32	23.026	33.621	25.191	25.191
	64	45.391	65.154	47.237	47.237
	128	97.632	121.104	112.654	115.321
40	16	25.971	29.172	26.950	26.950
	32	49.122	50.192	48.317	46.119
	64	93.521	101.463	96.175	96.175
	128	171.524	195.021	185.495	185.495
60	16	39.623	42.834	40.178	40.178
	32	81.434	90.762	82.329	82.329
	64	171.812	230.198	193.693	193.693
	128	355.705	465.014	386.611	386.611

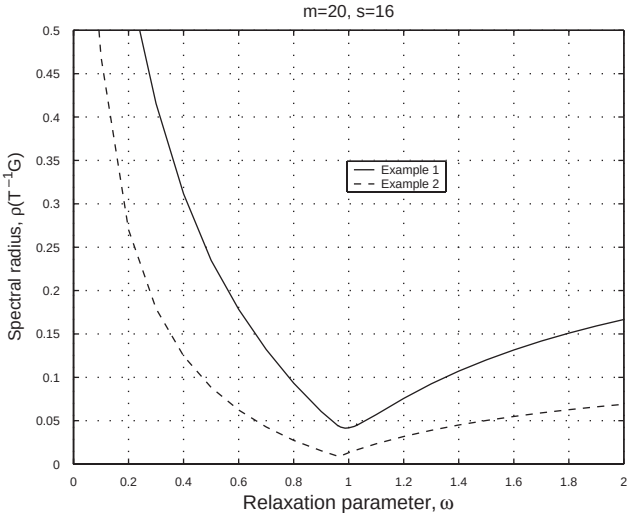

Fig. 1. Spectral radius $\rho(T^{-1}G)$ versus relaxation parameter ω.

where

$$Q = \begin{bmatrix} 2 & \frac{1}{3} & \cdots & \frac{1}{3^{m-2}} & \frac{1}{3^{m-1}} \\ -\frac{1}{2} & 2 & \frac{1}{3} & \cdots & \frac{1}{3^{m-2}} \\ \vdots & -\frac{1}{2} & 2 & \ddots & \vdots \\ -\frac{1}{2^{m-2}} & \vdots & \ddots & \ddots & \frac{1}{3} \\ -\frac{1}{2^{m-1}} & -\frac{1}{2^{m-2}} & \cdots & -\frac{1}{2} & 2 \end{bmatrix}.$$

Table 3
Number of WR iterations for convergence in Example 2

m	s	$C+S$	Jacobi	Gauss–Seidel	SOR ($\omega = 0.965$)
20	16	7	8	7	6
	32	7	7	7	6
	64	7	7	7	6
	128	6	7	7	6
40	16	7	7	7	7
	32	7	7	7	6
	64	6	7	7	6
	128	6	7	7	6
60	16	7	7	8	7
	32	6	7	7	6
	64	6	7	7	6
	128	6	7	7	6

Table 4
Number of Mflops for convergence in Example 2

m	s	$C+S$	Jacobi	Gauss–Seidel	SOR ($\omega = 0.965$)
20	16	9.351	11.289	11.678	10.971
	32	18.293	22.575	26.316	25.728
	64	38.710	47.692	62.242	60.768
	128	78.485	97.842	130.024	123.071
40	16	20.115	24.862	30.419	29.655
	32	38.994	49.278	65.355	63.493
	64	78.716	98.560	127.82	124.947
	128	150.914	194.286	251.335	241.040
60	16	31.581	39.938	53.250	50.878
	32	60.838	77.242	98.226	95.419
	64	127.095	161.433	223.283	215.184
	128	242.671	317.087	418.843	407.939

Tables 3 and 4 give the number of WR iterations and the number of megaflops (Mflops) required for convergence with different combinations of m and s. Again, Fig. 1 demonstrates that an optimal value for ω is $\omega \approx 0.965$ with corresponding spectral radius $\rho \approx 0.009$. From Table 4, we found that the number of megaflops of the SOR method with $\omega = 0.965$ is smaller than that of the Gauss–Seidel method. The number of WR iterations required for convergence remains almost a constant again for increasing m and s. From Table 4, the number of megaflops of our method is the smallest one.

Acknowledgements

We would like to thank the referees for their valuable suggestions and detailed comments which make the paper more presentable.

References

[1] G. Ammar, W. Gragg, Superfast solution of real positive definite toeplitz systems, SIAM J. Matrix Anal. Appl. 9 (1988) 61–76.
[2] D. Bertaccini, A circulant preconditioner for the systems of LMF-based ODE codes, SIAM J. Sci. Comput. 22 (2000) 767–786.
[3] L. Brugnano, D. Trigiante, Solving Differential Problems by Multistep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998.
[4] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, Oxford, 1995.
[5] K. Burrage, Z. Jackiewicz, S. Nørsett, R. Renaut, Preconditioning waveform relaxation iterations for differential systems, BIT 36 (1996) 54–76.
[6] R. Chan, M. Ng, Conjugate gradient methods for toeplitz systems, SIAM Rev. 38 (1996) 427–482.
[7] R. Chan, M. Ng, X. Jin, Strang-type preconditioner for systems of LMF-based ODE codes, IMA J. Numer. Anal. 21 (2001) 451–462.
[8] P. Davis, Circulant Matrices, Wiley, New York, 1979.
[9] G. Golub, C. van Loan, Matrix Computations, 3rd Edition, Johns Hopkins University Press, Baltimore, 1996.
[10] X. Jin, Developments and Applications of Block Toeplitz Iterative Solvers, Kluwer Academic Publishers & Science Press, Beijing, 2002.
[11] J. Lambert, Numerical Methods for Ordinary Differential Systems, Wiley, Chichester, 1991.
[12] E. Lelarasmee, A. Ruehli, A. Sangiovanni-Vincentelli, The waveform relaxation method for time-domain analysis of large-scale integrated circuits, IEEE Trans. CAD IC Systems 1 (1982) 131–145.
[13] U. Miekkala, O. Nevanlinna, Convergence of dynamic iteration methods for initial value problems, SIAM J. Sci. Statist. Comput. 8 (1987) 459–482.
[14] O. Nevanlinna, Remarks on Picard–Lindelöf iteration, Part I, BIT 29 (1989) 328–346.
[15] O. Nevanlinna, Remarks on Picard–Lindelöf iteration, Part II, BIT 29 (1989) 535–562.
[16] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[17] S. Vandewalle, Parallel Multigrid Waveform Relaxation for Parabolic Problems, B.G. Teubner, Stuttgart, 1993.
[18] Z. Xu, K. Zhang, Q. Lu, Fast Algorithm for Toeplitz-like Matrices, Northwest Industrial University Publishers, Xi'an, 1999 (in Chinese).