ELSEVIER

Available at WWW.MATHEMATICSWEB.ORG POWERED BY SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 151 (2003) 201–213

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

Convergence of the parallel chaotic waveform relaxation method for stiff systems

Dongjin Yuan[a], Kevin Burrage[b,*]

[a] *Department of Mathematics, Yangzhou University, JiangSu 225002, People's Republic of China*
[b] *Department of Mathematics, University of Queensland, St. Lucia, Qld. 4072, Australia*

Received 13 February 2001; received in revised form 13 July 2002

Abstract

In this paper, we establish several algorithms for parallel chaotic waveform relaxation methods for solving linear ordinary differential systems based on some given models. Under some different assumptions on the coefficient matrix A and its multisplittings we obtain corresponding sufficient conditions of convergence of the algorithms. Also a discussion on convergence speed comparison of synchronous and asynchronous algorithms is given.

MSC: 65L05; 65F10

Keywords: Multisplitting; Parallel; Chaotic; Convergence; Waveform relaxation; Stiff systems; H-matrix

1. Introduction

In the area of simulation of large electrical circuits the equation describing the circuit often yields an m-dimensional stiff initial value problem

$$x'(t) = f(t, x(t)), \quad x(0) = x_0$$

with $t \in [0, T]$, $x \in C^1([0, T]; R^m)$, $f \in C([0, T], R^m; R^m)$, $x_0 \in R^m$. If one has to simulate a very large scale integrated (VLSI) circuit, the dimension of the problem can be in the millions. One must solve an m-dimensional system of equations at each timepoint before advancing to the next timepoint. In the beginning of the 1980s a new approach for solving these problems was developed at the Electronic Research Laboratory at Berkeley that circumvents these difficulties. In this approach, called the

* Corresponding author.
E-mail addresses: dongjinyuan@yahoo.com (D. Yuan), kb@maths.uq.edu.au (K. Burrage).

0377-0427/02/$ - see front matter
PII: S0377-0427(02)00749-5

waveform relaxation algorithm, the full system is decomposed into smaller subsystems, which can be solved independently by different processors on a parallel computer. The subsystems are integrated over certain time intervals, the so-called windows. Inputs from other subsystems are taken from the previous iteration. By using a parallel computer, multisplitting methods were first introduced in [13] for solving large linear system of equations. Up to now, in general, there are two modes of parallel multisplitting iterations for solving not only large systems of linear algebraic equations but large systems of ordinary differential equations as well. One is synchronous and another one is chaotic (asynchronous).

The parallel multisplitting synchronous iterative method was first introduced in [13] on a parallel computer and so far there are many papers [2,15] to investigate and modify such a topic. In this mode the computational load of an iteration is divided in some way (for example, pointwise or blockwise) among the available processors, and it is assumed that the processors exchange all necessary information of the current iteration before a new iteration can begin. One of the main disadvantages of the synchronous method is the speed of computation is limited to that of the slowest processor (or that which has the most computational load), so that the fastest processor (or that which has less computational load) spends considerable amounts of time in an idle or wait status.

The second mode, which can avoid loss of time and efficiency in processor utilization, is the chaotic (asynchronous) in which the computation and communication are carried out on the various processors completely independently of the progress on other processors. This chaotic iterative method of complete independence of each processor was first presented in [6] (for point iterative schemes). Later many researchers then have developed this approach, such as [3,11,16], from several different ways in this aspect but only for large linear algebraic systems. In this paper we want to use the mode in waveform relaxation algorithm for large systems of ordinary differential equations.

We restrict ourselves to linear problems; this means that we are dealing only with linear m-dimensional initial problems which are given by

$$x'(t) + Ax(t) = f(t), \quad x(0) = x_0 \tag{1}$$

with $t \in [0, T]$, $x \in C^1([0, T]; R^m)$, $f \in C([0, T]; R^m)$, $A \in R^{m\times m}$. The decoupling process can mathematically be described by a splitting of the matrix A as

$$A = M - N. \tag{2}$$

The waveform relaxation algorithm is given by the iteration

$$x'_{n+1}(t) + Mx_{n+1}(t) = Nx_n(t) + f(t), \quad x_{n+1}(0) = x_0. \tag{3}$$

As a starting function for x_0 usually constant initial values are chosen, i.e., $x_0(t) \equiv x_0$. Iteration (3) is solved not only for one timepoint but for a fixed time interval $[0, T_0]$ which is usually a subset of the domain of integration $[0, T]$. After convergence has occurred on the window $[0, T_0]$ one proceeds to the next window $[T_0, T_1]$ until finally the end of the domain of integration T is reached.

It is worthy of remark that in [4,9,14], the authors used an overlapping method and an implicit integration method by parallel multisplitting, which are all synchronous, for solving (1) and obtained the convergence of the method, in which the matrix A in (1) is only an M-matrix.

The main purpose of this paper is to establish three parallel chaotic algorithms based on some given models in [3,11,16] for solving linear ordinary differential systems (1), in which the matrix

A can be an H-matrix, and investigate the corresponding convergence of these algorithms. Also a discussion on convergence speed comparison of synchronous and asynchronous algorithms will be given.

2. Notation and algorithms

Let us first introduce some of the notation and terminology [1] which will be used in this paper.

For $x=(x_1,x_2,\ldots,x_m)^{\top}\in R^m$ and $A=(a_{ij})\in R^{m\times m}$, then by $x\geqslant 0$ we mean that $x_i\geqslant 0$ for $i=1,\ldots,m$, and by $A\geqslant 0$ that $a_{ij}\geqslant 0$ for $i,j=1,\ldots,m$, in which case we say that x and A are nonnegative. For $A,B\in R^{m\times m}$, we write $A\leqslant B$ if $a_{ij}\leqslant b_{ij}$ hold for all entries of $A=(a_{ij})$ and $B=(b_{ij})$.

By $|A|=(|a_{ij}|)$ we define the absolute value of $A\in R^{m\times m}$, it is a nonnegative matrix satisfying $|AB|\leqslant |A|\cdot|B|$. The notation $\langle A\rangle=(\langle a\rangle_{ij})$ represents the comparison matrix of $A\in R^{m\times m}$ where

$$\langle a\rangle_{ij}=\begin{cases}|a_{ij}| & \text{if } i=j,\\ -|a_{ij}| & \text{if } i\neq j.\end{cases}$$

A matrix $A=(a_{ij})\in R^{m\times m}$ is an M-matrix if it is nonsingular with $A^{-1}\geqslant 0$ and $a_{ij}\leqslant 0$ for all $i\neq j$. It is an H-matrix if $\langle A\rangle$ is an M-matrix, and an L-matrix if $a_{ii}>0$ for $i=1,\ldots,m$, and $a_{ij}\leqslant 0$ for all $i\neq j$.

A splitting (2) of a matrix A is called a nonnegative splitting if $M^{-1}N\geqslant 0$ and an M-splitting if M is an M-matrix and $N\geqslant 0$. For any integer $L\geqslant 2$ a multisplitting of $A\in R^{m\times m}$ is a collection of L triples (M_l,N_l,E_l) of $m\times m$ real matrices, $l=1,2,\ldots,L$, for which each E_l is nonnegative diagonal, each M_l is invertible and the equations

$$A=M_l-N_l,\quad l=1,2,\ldots,L \tag{4}$$

and

$$\sum_{l=1}^{L}E_l=I \tag{5}$$

are satisfied.

We can now split the matrix A L times and we will take the right-hand side as input. $y_{l,n+1}(t)$ will denote the unknowns of the lth splitting. We then have

$$\begin{aligned}&y'_{l,n+1}(t)+M_l y_{l,n+1}(t)=N_l x_n(t)+f(t),\quad l=1,\ldots,L,\\ &y_{l,n+1}(0)=x_0.\end{aligned} \tag{6}$$

Relation (6) can represent the lth subsystem. In the case of synchronous communication, after having solved each subsystem since there is no iteration between two different subsystems we compute a new approximation to the solution of (1) by

$$x_{n+1}(t)=\sum_{l=1}^{L}E_l y_{l,n+1}(t). \tag{7}$$

Now we consider the case of asynchronous (chaotic). We can denote

$$y_{l,n+1}(t) = F_{l,(n;t)}(x_n(t), t). \tag{8}$$

We point out that there are many methods, such as semi-implicit mid-point rule or implicit Euler method, for solving problem (8).

For a nonnegative integer $\mu_{l,n}$, in which l, the number of the processor and n, the index of the iteration step, let

$$F_l^{\mu_{l,n}} = \begin{cases} F_l \cdot F_l \cdots F_l, & \mu_{l,n} \geqslant 1, \\ I, & \mu_{l,n} = 0. \end{cases}$$

The $F_l^{\mu_{l,n}}$ is the $\mu_{l,n}$th composition of the affine operator.

Using the above notation, (8) and some given models in [3,5,9,11,16] we can now describe three algorithms based on the parallel chaotic waveform relaxation method.

Algorithm 2.1.
initialize
$x_0(t) \equiv x_0 \ \forall t \in [0, T]$
$n := 0$
repeat
solve for $l = 1, 2, \ldots, L$ {in parallel}
$Y_{l,n+1}(t) := F_{l,(n;t)}^{\mu_{l,n}}(x_n(t), t), \ Y_{l,n+1}(0) = x_0, \ \mu_{l,n} \geqslant 1$
$x_{n+1}(t) := \sum_{l=1}^{L} E_l Y_{l,n+1}(t)$
$n := n + 1$
until convergence
End of algorithm

Here $\mu_{l,n} = \lfloor (\max_{1 \leqslant l \leqslant L} \tau_{l,n})/\tau_{l,n} \rfloor$, $l = 1, 2, \ldots, L$, with $\tau_{l,n}$ being the computing time of the lth processor for nth iteration. For example, by letting $L = 5$, $\tau_{1,n} = 0.125$, $\tau_{2,n} = 1$, $\tau_{3,n} = 0.5$, $\tau_{4,n} = 0.325$, and $\tau_{5,n} = 0.25$, we obtain $\mu_{1,n} = 8$, $\mu_{2,n} = 1$, $\mu_{3,n} = 2$, $\mu_{4,n} = 3$, $\mu_{5,n} = 4$.

We point out the algorithm in [11] is only the special case of the above algorithm when $\mu_{l,n} = 1$.

By using a suitable positive relaxation parameter ω, we then get the following relaxed algorithm which is based on Algorithm 2.1:

Algorithm 2.2.
initialize
$x_0(t) \equiv x_0 \ \forall t \in [0, T]$
$n := 0$
repeat
solve for $l = 1, 2, \ldots, L$ {in parallel}
$Y_{l,n+1}(t) := F_{l,(n;t)}^{\mu_{l,n}}(x_n(t), t), \ Y_{l,n+1}(0) = x_0, \ \mu_{l,n} \geqslant 1$
$x_{n+1}(t) := \omega \sum_{l=1}^{L} E_l Y_{l,n+1}(t) + (1 - \omega) x_n(t), \ \omega > 0$
$n := n + 1$
until convergence
End of algorithm

Next, we will consider the more complicated situation, which is similar to Algorithm 2 in [11,16]. In this case the following new terminology is necessary. A sequence of sets Q_n with $Q_n \subseteq \{1,\dots,L\}$ is admissible if every integer $1,\dots,L$ appears infinitely often in the Q_n, while such an admissible sequence is regulated if there exists a positive integer J such that each of the integers $1,\dots,L$ appears at least once in any J consecutive sets of the sequence. Assume that $\{Q_n\}$ is admissible and regulated, then we can get the following algorithm:

Algorithm 2.3.
initialize
$x_0(t) \equiv x_0 \ \forall t \in [0,T]$
$n := 0$
repeat
$x_n(0) = x_0$
$x_{n+1}(t) := (I - \omega \sum_{l\in Q_n} E_l)x_n(t) + \omega \sum_{l\in Q_n} E_l F^{\mu_{l,n}}_{l,(n;t)}(z_{n-r_n+1}(t),t)$,
$z_{n-r_n+1}(t) = (x^1_{n-r(1,n)}(t), x^2_{n-r(2,n)}(t),\dots,x^m_{n-r(m,n)}(t))^\top$,
$\mu_{l,n} \geqslant 1,\ \omega > 0,\ \emptyset \neq Q_n \subseteq \{1,\dots,L\}$
$n := n+1$
until convergence
End of algorithm

Remark 2.1. The terminology, overlapping multisplitting, was presented by [9] and other references. By denoting $s=1,\dots,m$, one can choose L subsets $s_1,s_2,\dots,s_L$ of s satisfying the condition $\bigcup_{l=1}^{L} s_l = s$. If m_l denotes the number of elements of s_l for $l=1,\dots,L$, then $\sum_{l=1}^{L} m_l \geqslant m$, where the equality holds if $s_1,s_2,\dots,s_L$ are disjoint. If there exists at least one pair of indices $i \neq j$ with $i,j \in 1,\dots,L$ so that $s_i \cap s_j \neq \emptyset$, then the multisplitting will be called an overlapping multisplitting; otherwise it will be named a disjoint multisplitting.

Remark 2.2. In order to save computing time, we can define the elements of the E_l by setting $E_l(i,i)=0$ for $i \notin s_l$. Since, by computing values for components that will be thrown away afterwards, not only computing time is wasted but also more memory is used. Therefore we will compute a component i of subsystem l only if $i \in s_l$. Using the subsets $s_1,s_2,\dots,s_L$ we can define L projection matrices $P_1,P_2,\dots,P_L$ in the following way: P_l are the diagonal matrices with $P_l(i,i) = 1 \Leftrightarrow i \in s_l$. By using these projection matrices we project problem (1) into L different subspaces. We solve now the projected problems in each subspaces but we use components from outside the particular subspace as an input. Observe that in a disjoint multisplitting the matrices P_l and E_l coincide. It is clear that

$$P_l E_l = E_l P_l = E_l, \quad l = 1,2,\dots,L.$$

We see that the iterate $x_{n+1}(t)$ is computed by

$$x_{n+1}(t) = \sum_{l=1}^{L} E_l F_{l,(n;t)}(x_n(t),t) = \sum_{l=1}^{L} E_l P_l F_{l,(n;t)}(x_n(t),t).$$

This means that in the lth subsystem we only have to compute $P_l F_{l,(n;t)}(x_n(t),t)$. In a practical implementation of the algorithm for the lth subsystem, not the matrix A but the projected matrix $P_l A$ is split.

3. Convergence of the algorithms

In this section we will adapt the approach presented in [4,9] and other previous references [7,10,12] to analyze the multisplitting algorithm. Since the proofs of some results in this section can be obtained in an analogous way or can be found in [9], we will omit them here.

First we introduce the following notation:

$$k_l(t) = \exp(-tM_l)N_l, \quad l = 1,2,\ldots,L,$$

where $k_l(t)$ is the kernel of the linear integral operator K_l acting on elements of the space $X = \mathscr{L}^p([0,T],R^m)$, $1 \leqslant p \leqslant \infty$ defined by

$$K_l u(t) = \int_0^t k_l(t-s)u(s)\,\mathrm{d}s, \quad l = 1,2,\ldots,L.$$

If we denote

$$\varphi_l(t) = \exp(-tM_l)x_0 + \int_0^t \exp((s-t)M_l)f(s)\,\mathrm{d}s, \quad l = 1,2,\ldots,L,$$

we can write the solution $y_l(t)$ of a subsystem (6) in the $(n+1)$st sweep using the "variation of constants formular" as

$$y_{l,n+1}(t) = K_l x_n(t) + \varphi_l(t).$$

Having computed the solution of each subsystem we have to weight the solutions by E_l matrices and to sum the weighted solutions over all subsystems to get a new approximation to the numerical solution of (1). Therefore the following notation will be used necessarily:

$$k(t) = \sum_{l=1}^{L} E_l k_l(t),$$

$$\mathscr{K}u(t) = \sum_{l=1}^{L} E_l K_l u(t),$$

$$\varphi(t) = \sum_{l=1}^{L} E_l \varphi(t)_l(t).$$

Using this notation the next iteration can be written as

$$x_{n+1}(t) = \mathscr{K}x_n(t) + \varphi(t). \tag{9}$$

Then we get

Lemma 3.1 (See from Burrage [4] and Jeltsch and Pohl [9]). *The following statements are equivalent*:

1. $x(t)$ *is a solution of* (1);
2. $x(t)$ *is a solution of* $x(t) = K_l x(t) + \varphi_l(t),\ x(0) = x_0,\ l = 1, 2, \ldots, L$;
3. $x(t)$ *is the solution of the fixed-point equation* $x(t) = \mathscr{K} x(t) + \varphi(t)$.

Lemma 3.2 (See also from Burrage [4] and Jeltsch and Pohl [9]). *For any finite interval* $[0, T]$,

$$\rho(\mathscr{K}) = \lim_{n\to\infty} \|\mathscr{K}^n\|_T^{1/n} \leqslant \lim_{n\to\infty} \left[\frac{(cT)^n}{n!}\right]^{1/n} = 0. \tag{10}$$

where $c = \sum_{l=1}^{L} c_l$ *with* $\|k_l\|_T \leqslant c_l,\ l = 1, \ldots, L$.

We summarize the above results in the following theorem.

Theorem 3.1. *The Algorithms* 2.1–2.3 *converge on any finite interval to the numerical solution of* (1) *without restriction on matrix* A *in* (1) *and the multisplitting*.

Remark 3.1. Here we should point out that (10) holds only on finite interval $[0, T]$. When one treats stiff ordinary differential equations, Lipschitz constants are usually large; even so, solutions are smooth. Hence the "classical" estimates on finite interval would force us to use small windows. To treat the stiff situation adequately we should show convergence independently of the size of Lipschitz constants, or what is equivalent to this for arbitrary window size. Hence we shall give bounds and show convergence independent of the interval. We pay our main attention to stiff ordinary differential equations where the step size can be fixed, but we can let the number of steps go to infinity. Here we only request that A is an H-matrix.

Before starting our main results concerning the above algorithms we should first introduce the following lemmas, which are necessary for the proofs of these results.

Lemma 3.3 (See from Song and Yuan [16]). *If* A *is an* H-*matrix*, *then*

(a) $|A^{-1}| \leqslant \langle A\rangle^{-1}$;
(b) *there exists a diagonal matrix* P *whose diagonal entries are positive such that* AP *is by rows strictly diagonally dominant*, *i.e.*,

$$\langle A\rangle Pe > 0 \tag{11}$$

with $e = (1, \ldots, 1)^{\top}$.

Lemma 3.4 (See also from Song and Yuan [16]). *Let* A *be an* M-*matrix*, *and let the splitting*

$$A = M - N$$

be an M-*splitting. If* P *is the diagonal matrix defined in Lemma* 3.3, *then*

$$\|P^{-1}M^{-1}NP\|_\infty < 1. \tag{12}$$

Lemma 3.5 (See from Pohl [14]). *Let matrices* $A=(a_{ij})$ *and* $B=(b_{ij})$ *be given with* $a_{ij} \leqslant 0$ *and* $b_{ij} \leqslant 0$ *for* $i \neq j$, *and let* A *be an* M-*matrix. Then* $A \leqslant B$ *implies that* B *is an* M-*matrix.*

Lemma 3.6 (See from Burrage [4]). *Let* $X=\mathscr{L}^p(R_+,R^m)$ *with* $1 \leqslant p \leqslant \infty$, *and* $K(z)=(zI+M)^{-1}N$ *is the Laplace transform of the kernel* $k(t)$ *of* $\mathscr{K}$. *Then*

$$\rho(\mathscr{K}) = \operatorname*{Sup}_{R(z)\geqslant 0} \rho((zI+M)^{-1}N).$$

Lemma 3.7 (See also from Pohl [14]). *Consider a matrix* $B \in R^{m\times m}$ *with* $B \geqslant 0$ *and* $u \in R^m$ *with* $u > 0$ *be given, then* $Bu < u$ *implies* $\rho(B) < 1$.

Using the above Lemmas 3.3–3.7, now we can prove one of our main results, which is a sufficient condition for the convergence of Algorithm 2.1.

Theorem 3.2. *Let* $A \in R^{m\times m}$ *be an* H-*matrix and let the matrix* P *be defined in Lemma* 3.3. *If the multisplitting* (M_l, N_l, E_l) *satisfies*

$$\langle A\rangle = \langle M_l\rangle - |N_l|, \quad l = 1,2,\ldots,L, \tag{13}$$

then the sequence $\{x_n(t)\}$ *generated by Algorithm* 2.1 *converges to the numerical solution of the stiff system* (1) *for an initial vector* $x(0)=x_0$.

Proof. From Lemma 3.6, we see that our goal is to show the following inequality:

$$\operatorname*{Sup}_{R(z)\geqslant 0} \rho\left(\sum_{l=1}^{L} E_l((zI+M_l)^{-1}N_l)^{\mu_{l,n}}\right) < 1. \tag{14}$$

By (13) we have

$$\langle zI+A\rangle = \langle zI+M_l\rangle - |N_l|, \quad l = 1,2,\ldots,L,\ R(z) \geqslant 0.$$

Since A is an H-matrix, then $\langle A\rangle$ is an M-matrix. By Lemma 3.5 $\langle zI+A\rangle$ is also an M-matrix. On the other hand $\langle M_l\rangle$ are M-matrices, for $l=1,\ldots,L$. Using Lemma 3.5 again, it is clear that $\langle zI+M_l\rangle$ are M-matrices, for $l=1,\ldots,L$. Hence $\langle zI+M_l\rangle - |N_l|$ are M-splitting of $\langle zI+A\rangle$, for $l=1,\ldots,L$. Using Lemma 3.4 we have

$$\|P^{-1}\langle zI+M_l\rangle^{-1}|N_l|P\|_\infty < 1, \quad l=1,\ldots,L$$

and by Lemma 3.3 we can obtain

$$\begin{aligned}\|P^{-1}(zI+M_l)^{-1}N_lP\|_\infty &\leqslant \|P^{-1}|(zI+M_l)^{-1}\|N_l|P\|_\infty \\ &\leqslant \|P^{-1}\langle zI+M_l\rangle^{-1}|N_L|P\|_\infty < 1, \quad l=1,\ldots,L.\end{aligned}$$

Let us denote $e=(1,1,\ldots,1)^{\top}\in R^m$. For $l=1,2,\ldots,L$, we then obtain

$$\begin{aligned}|(zI+M_l)^{-1}N_l|Pe &= P(P^{-1}|(zI+M_l)^{-1}N_l|P)e\\ &\leqslant \|P^{-1}(zI+M_l)^{-1}N_lP\|_{\infty}Pe\\ &\leqslant \max_{1\leqslant l\leqslant L}\|P^{-1}(zI+M_l)^{-1}N_lP\|_{\infty}Pe.\end{aligned}$$

We denote

$$\sigma=\max_{1\leqslant l\leqslant L}\|P^{-1}(zI+M_l)^{-1}N_lP\|_{\infty}\quad\text{and}\quad u=Pe, \tag{15}$$

then we have

$$|(zI+M_l)^{-1}N_l|u\leqslant\sigma u,\quad l=1,\ldots,L \tag{16}$$

with $\sigma\in(0,1)$ and $u>0\in R^m$.

Let us define

$$H=\sum_{l=1}^{L}E_l((zI+M_l)^{-1}N_l)^{\mu_{l,n}},\quad R(z)\geqslant 0,$$

then it follows from (16) that

$$\begin{aligned}|H|u &\leqslant \sum_{l=1}^{L}E_l|(z+M_l)^{-1}N_l|^{\mu_{l,n}}u\\ &\leqslant \sum_{l=1}^{L}E_l|(z+M_l)^{-1}N_l|u<\sum_{l=1}^{L}E_lu=u.\end{aligned}$$

By Lemma 3.7, we obtain

$$\rho(H)\leqslant\rho(|H|)<1.$$

It is evident that (13) holds immediately from the above inequality. Thus, we have completed the proof. □

Theorem 3.3. *Suppose that the assumptions of Theorem* 3.2 *hold. Then the sequence* $\{x_n(t)\}$ *generated by Algorithm* 2.2 *converges to the numerical solution of the stiff system* (1) *for an initial vector* $x(0)=x_0$ *when* $\omega\in(0,2/(1+\sigma))$ *with* σ *satisfying* (15).

Proof. Let us define

$$H=\omega\sum_{l=1}^{L}E_l((zI+M_l)^{-1}N_l)^{\mu_{l,n,i}}+(1-\omega)I,\quad R(z)\geqslant 0.$$

It is clear that we only need to prove there exists a vector $u\in R^m$ with $u>0$ such that $|H|u<u$. If we also take $u=Pe$, then it follows from (16) that

$$|H|u=\omega\sum_{l=1}^{L}E_l|(zI+M_l)^{-1}N_l|^{\mu_{l,n}}u+|1-\omega|u$$

$$\leqslant \omega \sum_{l=1}^{L} E_l |(zI + M_l)^{-1} N_l| u + |1 - \omega| u$$

$$\leqslant \omega\sigma u + |1 - \omega| u = (\omega\sigma + |1 - \omega|) u.$$

Let $\beta = \omega\sigma + |1 - \omega|$, it is easy to show that $\beta < 1$ for all $\omega \in (0, 2/(1 + \sigma))$ with σ satisfying (15). Consequently we have $|H|u < u$, then

$$\rho\left(\omega \sum_{l=1}^{L} E_l ((zI + M_l)^{-1} N_l)^{\mu_{l,n}} u + (1 - \omega) u\right) < 1,$$

which completes the proof. □

Using the proof of Theorem 3.2 and [16, Theorem 2.8] we get immediately our final result on the convergence of Algorithm 2.3.

Theorem 3.4. *Suppose that the assumptions of Theorem* 3.2 *hold. Then the sequence* $\{x_n(t)\}$ *generated by Algorithm* 2.3 *converges to the numerical solution of the stiff system* (1) *for an initial vector* $x(0) = x_0$ *when* $\omega \in (0, 2/(1 + \sigma))$ *with* σ *satisfying* (15) *and* $\emptyset \neq Q_n \subseteq \{1, \ldots, L\}$.

We should point out that we can get easily the conclusion that the convergence rate of the asynchronous iteration is geometric (for $\mu_{l,n} > 1$) and superior to that of the synchronous iteration. We now observe the convergence rates of the algorithm in [8], which is synchronous, and Algorithm 2.1 in this paper. Let us denote

$$H' = \sum_{l=1}^{L} E_l (zI + M_l)^{-1} N_l, \quad R(z) \geqslant 0,$$

where H' is the iterative matrix of the algorithm in [8].

Since there exists a vector $u > 0 \in R^m$ and a scalar $\sigma \in (0, 1)$ such that

$$|(zI + M_l)^{-1} N_l| u \leqslant \sigma u, \quad l = 1, 2, \ldots, L,$$

then we have $\rho((zI + M_l)^{-1} N_l) < 1$.

It is clear that we obtain

$$\rho\left(\sum_{l=1}^{L} E_l ((zI + M_l)^{-1} N_l)^{\mu_{l,n}}\right) \leqslant \rho\left(\sum_{l=1}^{L} E_l (zI + M_l)^{-1} N_l\right) < 1.$$

This implies

$$\rho(H) \leqslant \rho(H') < 1.$$

where H, defined in Section 3, is the iterative matrix of Algorithm 2.1.

As for Algorithms 2.2 and 2.3, since they are better than Algorithm 2.1 for some suitable relaxation parameters, we need not to compare the convergence speed of them with that of synchronous iteration.

Table 1
$T = 0.25$

m	100	125	150	175	200	400	700	1000
TOL $= 10^{-4}$	25	34	45	64	101	199	507	1014
TOL $= 10^{-8}$	39	53	72	102	161	322	856	1735

Table 2
$T = 0.5$

m	100	125	150	175	200	400	700	1000
TOL $= 10^{-4}$	40	54	70	103	158	313	793	1604
TOL $= 10^{-8}$	57	77	104	145	232	462	1233	2530

To conclude this paper, numerical results are reported in this section to show the convergence properties on a stiff problems. The test equation is derived from the heat equation. Discretizing the heat equation in one space variable, given by

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad t > 0, \ -1 < x < 1,$$
$$u(-1,t) = u(1,t) = 0, \quad u(x,0) = u_0(x),$$

where $u_0(x)$ is a given initial function, leads to the linear system of the form

$$y'(t) + Qy(t) = 0, \quad t > 0, \ y(0) = y_0,$$

where $y = [y_1, y_2, \ldots, y_m]^{\mathrm{T}} = [y(t,x_1), y(t,x_2), \ldots, y(t,x_m)]^{\mathrm{T}}$,

$$Q = \frac{a^2}{h_x^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} \in R^{m \times m}$$

and h_x is the stepsize for discretizing the spatial variable. Here $a = 1$.

Tables 1–3 give the number of iterations needed to reduce the errors below the required tolerances, TOL $= 10^{-4}$ and 10^{-8} on the window $[0, 0.25]$, $[0, 0.5]$ and $[0, 1]$ based on Algorithm 2.1, where $\mu_{1,n}$=1.5, $\mu_{2,n}$=2.5, $\mu_{3,n}$=3, $\mu_{4,n}$=1.25. Also Table 4 gives the number of iterations about TOL$=10^{-4}$ and 10^{-8} on the window $[0, 0.25]$ based on the synchronous method. By comparing Table 1 with Table 4, we see that the convergence rate of Algorithm 2.1 is superior to that of synchronous case for this test equation.

Table 3
$T = 1$

m	100	125	150	175	200	400	700	1000
TOL $= 10^{-4}$	66	92	118	174	269	538	1342	2715
TOL $= 10^{-8}$	89	122	163	231	412	743	1957	4014

Table 4
$T = 0.25$

m	100	125	150	175	200	400	700	1000
TOL $= 10^{-4}$	41	56	74	108	171	356	938	1989
TOL $= 10^{-8}$	64	89	119	170	273	576	1586	3402

Remark 3.2. Our numerical results indicate that the algorithms presented in this article are more informative, providing a much better comparison of the efficiencies of the convergence rates. However, we also should point out that when the dimension of the coefficient matrix is very small, there is little value in the asynchronous methods because of the overheads in parallel execution.

Acknowledgements

The authors would like to thank the referees for their valuable comments and constructive suggestions on the original manuscript of this paper.

References

[1] A. Berman, R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.
[2] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation, Prentice-Hall, Toronto, 1989.
[3] R. Bru, L. Elsner, M. Neumman, Models of parallel chaotic iterative methods, Linear Algebra Appl. 103 (1988) 175–192.
[4] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, Oxford, 1995.
[5] K. Burrage, C. Dyke, B. Pohl, On the performance of parallel waveform relaxation for differential systems, Appl. Numer. Math. 20 (1996) 39–55.
[6] D. Chazan, W. Miranker, Chaotic relaxation, Linear Algebra Appl. 2 (1969) 199–222.
[7] Z. Jackiewicz, M. Kwapisz, Convergence of waveform relaxation methods for differential-algebraic equations, SIAM J. Numer. Anal. 33 (6) (1996) 2303–2317.
[8] J. Janssen, S. Vandewall, Multigrid waveform relaxation on spatial finite element meshes: the continuous-time case, SIAM J. Numer. Anal. 33 (2) (1996) 456–474.
[9] R. Jeltsch, B. Pohl, Waveform relaxation with overlapping splittings, SIAM J. Sci. Comput. 16 (1995) 40–49.
[10] Y. Jiang, O. Wing, A note on convergence conditions of waveform relaxation algorithms for nonlinear differential–algebraic equations, Appl. Numer. Math. 36 (2001) 281–297.
[11] P.E. Kloeden, D. Yuan, Convergence of relaxed chaotic parallel iterative methods, Bull. Austral. Math. Soc. 50 (1994) 167–176.

[12] U. Miekkala, O. Nevanlinna, Convergence of dynamic iteration methods for initial value problem, SIAM J. Sci. Statist. Comput. 8 (4) (1987) 459–482.
[13] D.P. O'Leary, R.E. White, Multisplittings of matrices and parallel solution of linear systems, SIAM J. Algebraic Discrete Methods 6 (1985) 630–640.
[14] B. Pohl, On the convergence of the discretized multisplitting waveform relaxation algorithm, Appl. Numer. Math. 11 (1993) 251–258.
[15] Y. Song, Convergence of parallel multisplitting method for H-matrices, Internat. J. Comput. Math. 48 (1993).
[16] Y. Song, D. Yuan, On the convergence of relaxed parallel chaotic iterations for H-matrix, Internat. J. Comput. Math. 52 (1994) 195–209.