Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

# A parallel multisplitting method with self-adaptive weightings for solving $H$-matrix linear systems

Ruiping Wen[1*] and Hui Duan[2]

*Correspondence: wenrp@163.com
[1]Higher Education Key Laboratory of Engineering and Scientific Computing, Taiyuan Normal University, Taiyuan, Shanxi 030012, P.R. China
Full list of author information is available at the end of the article

**Abstract**

In this paper, a parallel multisplitting iterative method with the self-adaptive weighting matrices is presented for the linear system of equations when the coefficient matrix is an $H$-matrix. The zero pattern in weighting matrices is determined in advance, while the non-zero entries of weighting matrices are determined by finding the optimal solution in a hyperplane of $\alpha$ points generated by the parallel multisplitting iterations. Especially, the nonnegative restriction of weighting matrices is released. The convergence theory is established for the parallel multisplitting method with self-adaptive weightings. Finally, a numerical example shows that the parallel multisplitting iterative method with the self-adaptive weighting matrices is effective.

**Keywords:** linear systems; self-adaptive weightings; convergence; parallel multisplitting; $H$-matrix

## 1 Introduction and preliminaries

To solve the large sparse linear system of equations

$$Ax = b, \quad A \in \mathbf{R}^{n\times n} \text{ nonsingular and } b \in \mathbf{R}^n, \tag{1.1}$$

O'Leary and White [1] first proposed parallel methods based on multisplittings of matrices in 1985, where several basic convergence results may be found. A multisplitting of $A$ is a collection of triples of $n \times n$ matrices $(M_i, N_i, E_i)_{i=1}^{\alpha}$ ($\alpha \le n$, a positive integer) with $M_i, N_i, E_i \in \mathbf{R}^{n\times n}$, and then the following method for solving system (1.1) was given:

$$\begin{aligned} &A = M_i - N_i, \quad i = 1, 2, \ldots, \alpha, \\ &M_i x_i^{(k)} = N_i x^{(k-1)} + b, \quad i = 1, 2, \ldots, \alpha; k = 1, 2, \ldots, \\ &x^{(k)} = \sum_{i=1}^{\alpha} E_i x_i^{(k)}, \end{aligned}$$

where $M_i$, $i = 1, 2, \ldots, \alpha$, are nonsingular and $E_i = \operatorname{diag}(e_1^{(i)}, e_2^{(i)}, \ldots, e_n^{(i)})$, $i = 1, 2, \ldots, \alpha$, satisfy $\sum_{i=1}^{\alpha} E_i = I$ ($I \in \mathbf{R}^{n\times n}$ is the identity matrix). By the way, $x^{(0)}$ is an initial vector of $x_* = A^{-1}b$.

There has been a lot of study (see [1–13]) on the parallel iteration methods for solving the large sparse system of linear equations (1.1). In particular, when the coefficient matrix $A$ is an $M$-matrix or an $H$-matrix, many parallel multisplitting iterative methods (see, e.g., [1–7, 10]) were presented, and the weighting matrices $E_i$, $i = 1, 2, \ldots, \alpha$, were generalized (see, e.g., [3, 4, 6–10])

$$\sum_{i=1}^{\alpha} E_i^{(k)} = I \text{ (or } \neq I), \quad E_i^{(k)} \geq 0, \text{diagonal}, i = 1, 2, \ldots, \alpha; k = 1, 2, \ldots, \tag{1.2}$$

but these weighting matrices were preset as multi-parameter.

As we know, the weighting matrices play an important role in parallel multisplitting iterative methods, but the weighting matrices in all the above-mentioned methods are determined in advance, they are not known to be good or bad, and this influences the efficiency of parallel methods. Recently, Wen and co-authors [13] discussed self-adaptive weighting matrices for a symmetric positive definite linear system of equations; Wang and co-authors [11] discussed self-adaptive weighting matrices for non-Hermitian positive definite linear system of equations. In this paper, we focus on the $H$-matrix, which originates from Ostrowski [14]. The $H$-matrix is a class of the important matrices that has many applications. For example, numerical methods for solving PDEs are a source of many linear systems of equations whose coefficients form $H$-matrices (see [15–18]). Are self-adaptive weighting matrices true for the linear system of equations when the coefficient matrix is an $H$-matrix? We will discuss this problem in this paper.

Here, we generalize the weighting matrices $E_i$ ($i = 1, 2, \ldots, \alpha$) to $E_i^{(k)}$ ($i = 1, 2, \ldots, \alpha$; $k = 1, 2, \ldots$), which can be divided into two steps: each splitting is first dealt with by a different processor, the weighting matrices $E_i^{(k)}$ ($i = 1, 2, \ldots, \alpha$; $k = 1, 2, \ldots$) will mask large portion of the matrix, so that each processor deals with a smaller matrix; later, the weighting matrices $E_i^{(k)}$ ($i = 1, 2, \ldots, \alpha$; $k = 1, 2, \ldots$) will be approximated to 'the best' choices well for $k$-step iteration in the hyperplane generated by $\{x_i^{(k)}, i = 1, 2, \ldots, \alpha\}$, $k = 1, 2, \ldots$. In this paper, we determine the weighting matrices $E_i^{(k)}$ ($i = 1, 2, \ldots, \alpha$; $k = 1, 2, \ldots$) for above reasons. Firstly, decreasing execution times is completed by the zero-entry's set in weighting matrices. Secondly, the self-adaptive weighting matrices are reached by finding the 'good' point in the hyperplane generated by $\{x_i^{(k)}, i = 1, 2, \ldots, \alpha\}$, $k = 1, 2, \ldots$. Thus, the scheme of finding the self-adaptive weighting matrices is established for the parallel multisplitting iterative method, it has two advantages as follows:

- The weighting matrices are not necessarily nonnegative;
- Only one splitting of $\alpha$ splittings is required to be convergent.

In the rest of this study, we first give some notations and preliminaries in Section 1, and then a parallel multisplitting iterative method with the self-adaptive weighting matrices is put forward in Section 2. The convergence of the parallel multisplitting iterative method is established in Section 3. Moreover, we give the computational results of the parallel multisplitting iterative method by a problem in Section 4. We end the paper with a conclusion in Section 5.

Here are some essential notations and preliminaries. $\mathbf{R}^{n\times n}$ is used to denote the $n \times n$ real matrix set, and $\mathbf{R}^n$ is the $n$-dimensional real vector set. $A^T$ represents the transpose of the matrix $A$, and $x^T$ denotes the transpose of the vector $x$. $\langle A \rangle$ stands for the comparison (Ostrowski) matrix of the matrix $A$, and $|A|$ is the absolute matrix of the matrix $A$.

In what follows, when $A$ is a strictly diagonally dominant matrix in column, $A$ is called a strictly diagonally dominant matrix.

**Definition 1.1** ([17]) The matrix $A$ is an $H$-matrix if there exists a positive diagonal matrix $D$ such that the matrix $DA$ is a strictly diagonally dominant matrix.

**Property 1.2** ([14]) The matrix $A$ is an $H$-matrix if and only if $\langle A\rangle$ is an $M$-matrix.

**Definition 1.3** ([8, 12]) Suppose that $A$ is an $H$-matrix. Let $A = M - N$, which is called an $H$-compatible splitting if $\langle A\rangle = \langle M\rangle - |N|$.

**Property 1.4** ([15]) Let $A$ be an $H$-matrix, then $|A^{-1}| \le \langle A\rangle^{-1}$.

## 2 Description of the method

Here, we present a parallel multisplitting iterative method with the self-adaptive weighting matrices.

Let

$$A = M_i - N_i, \quad i = 1, 2, \dots, \alpha. \tag{2.1}$$

$$E_i^{(k)} = \operatorname{diag}\bigl(e_1^{(i,k)}, \dots, e_n^{(i,k)}\bigr), \qquad \sum_{i=1}^{\alpha} E_i^{(k)} = I, \quad k = 1, 2, \dots. \tag{2.2}$$

By introducing

$$T_i = M_i^{-1} N_i, \quad i = 1, 2, \dots, \alpha. \tag{2.3}$$

The iteration matrix of $k$th step

$$T_k = \sum_{i=1}^{\alpha} E_i^{(k)} M_i^{-1} N_i = \sum_{i=1}^{\alpha} E_i^{(k)} T_i. \tag{2.4}$$

The preset nonzero set

$$S(i,k) = \bigl\{j | e_j^{(i,k)} \neq 0\bigr\}, \qquad S(k) = \bigcup_{i=1}^{\alpha} S(i,k). \tag{2.5}$$

**Method 2.1**

Step 0. Given a tolerance $\epsilon > 0$ and an initial vector $x^{(0)}$. For $i = 1, 2, \dots, \alpha$; $k = 1, 2, \dots$, give also the set $S(i,k)$ and for some $i_0$, $S(i_0, k) = \{1, 2, \dots, n\}$. The sequence of $x^{(k)}$, $k = 1, 2, \dots$, until converges.

Step 1. *Solution in parallel.* For the $i$th processor, $i = 1, 2, \dots, \alpha$: compute $x_j^{(i,k)}$, $j \in S(i,k)$ by

$$M_i x^{(i,k)} = N_i x^{(i,k-1)} + b, \tag{2.6}$$

where $x^{(i,k)} = (x_1^{(i,k)}, \dots, x_n^{(i,k)})^T$.

Step 2. For the $i_0$th processor, set

$$x = \sum_{i=1}^{\alpha} E_i^{(k)} x_i^{(k)} = \begin{pmatrix} \sum_{i=1}^{\alpha} e_1^{(i,k)} x_1^{(i,k)} \\ \sum_{i=1}^{\alpha} e_2^{(i,k)} x_2^{(i,k)} \\ \vdots \\ \sum_{i=1}^{\alpha} e_n^{(i,k)} x_n^{(i,k)} \end{pmatrix}.$$

Solve the following optimization problem:

$$\begin{aligned} &\min_{e_j^{(i,k)} \in S(k)} \|Ax - b\|_1 \\ &\text{s.t.} \sum_{i=1}^{\alpha} E_i^{(k)} = I. \end{aligned} \tag{2.7}$$

Or

$$\begin{aligned} &\|Ax - b\|_1 \leq \left\|Ax_{i_0}^{(k)} - b\right\|_1 \\ &\text{s.t.} \sum_{i=1}^{\alpha} E_i^{(k)} = I. \end{aligned} \tag{2.8}$$

Step 3. Compute

$$x^{(k)} = \sum_{i=1}^{\alpha} E_i^{(k)} x_i^{(k)}. \tag{2.9}$$

Step 4. If $\|Ax^{(k)} - b\|_1 \leq \epsilon$, stop; otherwise, $k \Leftarrow k + 1$, go to Step 1.

**Remark 2.1** The implementation of this method is that at each iteration there are $\alpha$ independent problems of the kind (2.6) with $x_j^{(i,k)}$, $j \in S(i,k)$ represents the solution to the local problem. The work for each equation in (2.6) is assigned to one processor, and communication is required only to produce the update given in (2.9). In general, some (most) of the diagonal elements in $E_i$ are zero and therefore the corresponding components of $x_j^{(i,k)}$, $j \in S(i,k)$ need not be calculated.

**Remark 2.2** We may use some optimization methods such as the simplex method (see [19]) to solve an approximated solution satisfying inequality (2.7). Usually, we can compute the optimal weighting at every two or three iterations to replace at each iteration step. Hence, the computational complexity of (2.7) is about $4n^2$ or $6n^2$ flops.

On the other hand, if $S(i_0,k) = n$, $S(i,k) = 1$, $i \neq i_0$, let $x = (1-t)x^{(i_0,k)} + t\bar{x}^{(k)}$, (2.7) becomes

$$\|Ax - b\|_1 = \left\|Ax^{(i_0,k)} - b - tA\left(x^{(i_0,k)} - \bar{x}^{(k)}\right)\right\|_1 = \|\bar{b} - \bar{a}t\|_1.$$

Now, we consider the following programming:

$$\min_x \sum_{j=1}^{n} |b_j - a_j x|. \tag{2.10}$$

**Assumptions**

(a) $b_j \geq 0, j = 1, 2, \ldots, n$.

(b) $\frac{b_1}{a_1} \leq \frac{b_2}{a_2} \leq \cdots \leq \frac{b_n}{a_n}$.

**Lemma 2.1** *Let programming* (2.10) *satisfy Assumptions and* $x_j = \frac{b_j}{a_j}$, $j = 1, 2, \ldots, n$. *Then there exists some* $j_0$ *such that* $x_{j_0}$ *is the solution of programming* (2.10).

*Proof* As we know, the solution $x^* \in [\frac{b_1}{a_1}, \frac{b_n}{a_n}]$. Let $P = \{\frac{b_1}{a_1}, \frac{b_2}{a_2}, \ldots, \frac{b_k}{a_k}, \ldots, \frac{b_n}{a_n}\}$ be a partition of $[\frac{b_1}{a_1}, \frac{b_n}{a_n}]$, it implies that $\frac{b_1}{a_1} < \frac{b_2}{a_2} < \cdots < \frac{b_k}{a_k} < \cdots < \frac{b_n}{a_n}$. Thus, we obtain a set of subinterval induced by the partition $P$.

In every subinterval, the function $\sum_{j=1}^{n} |b_j - a_j x|$ is a linear function, then the minimization point of the linear function is just a partition point. Hence, the lemma has been proved. □

**Corollary 2.2** *Let programming* (2.10) *satisfy Assumptions. Then*

$$\min\left\{\sum_{j=1}^{n}\left|b_j - a_j\frac{b_k}{a_k}\right|, \sum_{j=1}^{n}\left|b_j - a_j\frac{b_{k+1}}{a_{k+1}}\right|\right\} \leq \sum_{j=1}^{n} b_j$$

*with* $\frac{b_k}{a_k} \leq 0 \leq \frac{b_{k+1}}{a_{k+1}}$.

From Lemma 2.1 we obtain an approximated solution, its complexity is about $4n^2$ flops.

## 3 Convergence analysis

In this section, we discuss the convergence of Method 2.1 under the reasonable assumptions.

**Lemma 3.1** *Let* $A = M - N$ *be* $H$*-compatible splitting of the strictly diagonally dominant matrix* $A$, *then*

$$\left\|NM^{-1}\right\|_1 < 1. \tag{3.1}$$

*Proof* From the definition of $H$-compatible splitting, we know that $\langle A\rangle = \langle M\rangle - |N|$.

Let $N = (N_1^T, N_2^T, \ldots, N_n^T)^T$, $N_j = (n_{j1}, n_{j2}, \ldots, n_{jn})$. From Property 1.4, it holds that

$$\left\|NM^{-1}\right\|_1 \leq \left\||N|\langle M\rangle^{-1}\right\|_1 = \max_{1\leq j\leq n}\left|\left(e^T|N|\langle M\rangle^{-1}\right)_i\right|.$$

Let $e^T|N|\langle M\rangle^{-1} = x^T$, where $x^T = (x_1, x_2, \ldots, x_n)$. We have

$$e^T|N| = x^T\langle M\rangle.$$

Let $x_{j_0} = \max_{1\leq j\leq n} x_j$, which implies

$$\sum_{j=1}^{n}|n_{jj_0}| = m_{j_0j_0}x_{j_0} - \sum_{j\neq j_0} m_{jj_0}x_j \geq \left(m_{j_0j_0} - \sum_{j\neq j_0} m_{jj_0}\right)x_{j_0}, \quad x_{j_0} \leq \frac{\sum_{j=1}^{n}|n_{jj_0}|}{m_{j_0j_0} - \sum_{j\neq j_0} m_{jj_0}}.$$

From the $H$-compatible splitting $\langle A\rangle = \langle M\rangle - |N|$ and the strict diagonal dominance of $\langle A\rangle$, it holds that

$$\frac{\sum_{j=1}^{n} |n_{jj_0}|}{m_{j_0j_0} - \sum_{j\neq j_0} m_{jj_0}} < 1.$$

Hence, $\|NM^{-1}\|_1 < 1$. □

**Lemma 3.2** *Let $A = M - N$ be $H$-compatible splitting of the $H$-matrix $A$. Then there exists a positive diagonal matrix $D$ such that*

$$\left\|DNM^{-1}D^{-1}\right\|_1 < 1. \tag{3.2}$$

*Proof* There is a positive diagonal matrix $D$ such that $DA$ is a strictly diagonally dominant matrix since $A$ is an $H$-matrix. Thus

$$\langle DA\rangle = \langle DM\rangle - |DN|. \tag{3.3}$$

Let

$$\langle DA\rangle = \bar{A}, \qquad \langle DM\rangle = \bar{M}, \qquad |DN| = \bar{N}.$$

Then

$$\bar{A} = \bar{M} - \bar{N}$$

is a regular splitting of the strictly diagonally dominant matrix $\bar{A}$. From Lemma 3.1, we know that

$$\left\|\bar{N}\bar{M}^{-1}\right\|_1 < 1.$$

Hence,

$$\left\|DNM^{-1}D^{-1}\right\|_1 = \left\|\bar{N}\bar{M}^{-1}\right\|_1 < 1.$$ □

**Theorem 3.3** *Let $A = M_i - N_i$, $i = 1, 2, \ldots, \alpha$, be $\alpha$ splittings of the $H$-matrix $A$, and for some $i_0$, let $A = M_{i_0} - N_{i_0}$ be $H$-compatible splitting. Assume that $E_i^{(k)}$, $i = 1, 2, \ldots, \alpha$; $k = 1, 2, \ldots$ are yielded by* (2.7) *or* (2.8) *in Method* 2.1. *Then $\{x^{(k)}\}$ generated by Method* 2.1 *converges to the unique solution $x_*$ of* (1.1).

*Proof* Let $\varepsilon^{(k)} = x^{(k)} - x_*$. We have

$$\varepsilon^{(k+1)} = T_k\varepsilon^{(k)}, \qquad \varepsilon_i^{(k+1)} = T_i\varepsilon^{(k)}, \quad i = 1, 2, \ldots, \alpha. \tag{3.4}$$

On the other hand,

$$\min_{e_j^{(i,k)}\in S(k)} \|Ax - b\|_1 = \min_{e_j^{(i,k)}\in S(k)} \left\|A(x - x_*)\right\|_1. \tag{3.5}$$

Let $D$ be a positive diagonal matrix such that $DA$ is a strictly diagonally dominant matrix. Thus, from (2.8) (or (2.9)) and Lemma 3.2 we know that

$$\begin{aligned}\left\|DA\varepsilon^{(k+1)}\right\|_1 &= \left\|DAT_k\varepsilon^{(k)}\right\|_1 \le \min_{1\le i\le\alpha}\left\|DAT_i\varepsilon^{(k)}\right\|_1 \\ &\le \left\|DAT_{i_0}A^{-1}D^{-1}\right\|_1\left\|DA\varepsilon^{(k)}\right\|_1 = \left\|DN_{i_0}M_{i_0}^{-1}D^{-1}\right\|_1\cdot\left\|DA\varepsilon^{(k)}\right\|_1 \\ &\le \left\|D|N_{i_0}|\langle M_{i_0}\rangle^{-1}D^{-1}\right\|_1\cdot\left\|DA\varepsilon^{(k)}\right\|_1 \le r\left\|DA\varepsilon^{(k)}\right\|_1 \\ &\le \cdots \le r^{k+1}\left\|DA\varepsilon^{(0)}\right\|_1,\end{aligned}$$

where $r = \|D|N_{i_0}|\langle M_{i_0}\rangle^{-1}D^{-1}\|_1 < 1$.

Thus, $\lim_{k\to\infty}\|DA\varepsilon^{(k+1)}\|_1 = 0$, which implies that

$$\lim_{k\to\infty}\varepsilon^{(k+1)} = 0.$$

We have completed the proof of the theorem. □

## 4 Numerical experiments

In this section, a test problem to assess the feasibility and effectiveness of Method 2.1 in terms of both iteration number (denoted by IT) and computing time (in seconds, denoted by CPU) is given. All our tests are started from zero vector and terminated when the current iterate satisfied $\|r^{(k)}\|_1 < 10^{-6}$, where $r^{(k)}$ is the residual of the current, say $k$th iteration or the number of iterations is up to 20,000. For the latter the iteration is failing. We solve (2.7) or (2.8) in the optimization step by the simplex method (see [19]).

**Problem** Consider the generalized convection-diffusion equations in a two-dimensional case. The equation is

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + q\cdot\exp(x+y)\cdot x\cdot\frac{\partial u}{\partial x} + q\cdot\exp(x+y)\cdot y\cdot\frac{du}{dy} = f \tag{4.1}$$

with the homogeneous Dirichlet boundary condition. We use the standard Ritz-Galerkin finite element method by $P_1$ conforming triangular element to approximate the following continuous solutions $u = x\cdot y\cdot(1-x)\cdot(1-y)$ in the domain $\Omega = [0,1]\times[0,1]$, the step-sizes along both $x$ and $y$ directions are the same, that is, $h = \frac{1}{2^m}$, $m = 5,6,7$. Let $q = 1$.

After discretization, the matrix $A$ of this equation is given by

$$A = \begin{bmatrix} A_{11} & B_{12} & & & & \\ C_{21} & A_{22} & B_{23} & & & \\ & \ddots & \ddots & \ddots & & \\ & & C_{p-1,p-2} & A_{p-1,p-1} & B_{p-1,p} \\ & & & C_{p,p-1} & A_{p,p} \end{bmatrix} \in \mathbf{R}^{n\times n},$$

where $A_{i,i}$, $i = 1,\ldots,p$, are $s$-by-$s$ nonsymmetric matrices and $B_{i,i+1}^T \ne C_{i+1,i}$. Thus, $n = s\cdot p$ and $s = p = 2^m = 32, 64, 128$ from the above.

Let

$$A = D - L - U,$$

where $D$ is a block diagonal matrix, $L$ is the strictly block lower triangle matrix, $U$ is the strictly block upper triangle matrix.

We construct the multisplitting as follows:

(a) The block Jacobi splitting (denoted by BJ)

$$A = M_1 - N_1, \qquad M_1 = D.$$

(b) The block Gauss-Seidel splitting I (denoted by BGS-I)

$$A = M_2 - N_2, \qquad M_2 = D - L.$$

(c) The block Gauss-Seidel splitting II (denoted by BGS-II)

$$A = M_2 - N_2, \qquad M_2 = D - U.$$

The three weighting matrices are chosen as follows:

$$E_1 = \operatorname{diag}(\omega I, \omega I),$$

$$E_2 = \operatorname{diag}(\beta I, 0),$$

$$E_3 = \operatorname{diag}(0, \gamma I),$$

where $I$ is the $\frac{n}{2} \times \frac{n}{2}$ identity matrix.

In the first results presented in Table 1 we show the spectral radius of the corresponding iteration matrix for the classical iterative methods (a)-(c). It is well known that an iterative method is convergent when the spectral radius of the corresponding iteration matrix is less than one. We can see that the numbers listed in Table 1 approximate to 1 such that these iterative methods converge to the unique solution of (4.1) slowly. However, the behavior of the multisplitting method based on three splittings (a)-(c) can be improved by self-adaptive weightings, and the results are shown in Tables 2 and 3.

Here, we denote the basic parallel multisplitting iterative method with fixed weighting matrix by B-Meth (see [1]). In Basic Methods, we propose three groups of weighting matrices, which are generated by random selection. Thus, the corresponding parallel multisplitting iterative methods are denoted by B-Meth 1, B-Meth 2, B-Meth 3, respectively.

The speed-up is defined in the following:

$$\text{speed-up} = \frac{\text{CPU of the Basic Methods}}{\text{CPU of Method 2.1}}.$$

**Table 1 Radii of the classical iterative methods**

| ***n*** | **BJ** | **BGS-I** | **BGS-II** |
|---|---|---|---|
| $32 \times 32$ | 0.9923 | 0.984 | 0.9847 |
| $64 \times 64$ | 0.9980 | 0.9960 | 0.9961 |
| $128 \times 128$ | 0.9995 | 0.9990 | 0.9991 |

**Table 2 The comparison of computational results among the BGS-I, BGS-II, Method 2.1**

| *n* | | BGS-I | BGS-II | Method 2.1 |
|---|---|---|---|---|
| 32 × 32 | IT | 866 | 869 | 653 |
| | CPU(s) | 3.465 | 3.838 | 2.5130 |
| 64 × 64 | IT | 3,353 | 3,358 | 2,314 |
| | CPU(s) | 64.493 | 78.830 | 23.2740 |
| 128 × 128 | IT | 13,191 | 13,202 | 6,799 |
| | CPU(s) | 1,203.512 | 1,367.604 | 365.7850 |

Note: $k_{\mathrm{opt}} = 2$ can be chosen in Method 2.1.

**Table 3 The comparison of computational results between Method 2.1 and Basic Method**

| *n* | | Method 2.1 | B-Meth 1 | B-Meth 2 | B-Meth 3 |
|---|---|---|---|---|---|
| 32 × 32 | IT | 653 | 1,308 | 1,226 | 1,175 |
| | CPU(s) | 2.5130 | 4.4435 | 3.3955 | 3.1638 |
| 64 × 64 | IT | 2,314 | 4,571 | 4,601 | 4,489 |
| | CPU(s) | 23.2740 | 57.1624 | 57.1404 | 55.3278 |
| 128 × 128 | IT | 6,799 | 17,865 | 17,662 | 17,877 |
| | CPU(s) | 365.7850 | 1,048.6420 | 1,047.1182 | 1,040.2860 |

From the above numerical experiments, it is obtained that the average speed-up of the new parallel multisplitting iterative method (Method 2.1) is about 2.3 (the average value of all computational results by Basic Methods).

## 5 Conclusion

The parallel multisplitting iterative method with the self-adaptive weighting matrices has been proposed for the linear system of equations (1.1) when the coefficient matrix is an $H$-matrix. The convergence theory is established for the parallel multisplitting method with self-adaptive weightings. The numerical results show that the new parallel multisplitting iterative method with the self-adaptive weightings is effective.

**Competing interests**
The authors declare that they have no competing interests.

**Authors' contributions**
RW presented the optimization model of weighting matrices, carried out the convergence studies and drafted the manuscript. HD implemented the algorithm and helped to draft the manuscript. All authors read and approved the final manuscript.

**Author details**
[1]Higher Education Key Laboratory of Engineering and Scientific Computing, Taiyuan Normal University, Taiyuan, Shanxi 030012, P.R. China. [2]Department of Mathematics, Taiyuan Normal University, Taiyuan, Shanxi 030012, P.R. China.

**Acknowledgements**
The authors are very much indebted to the anonymous referees for their helpful comments and suggestions. And the authors are so thankful for the support from the NSF of Shanxi Province. This work is supported by grants from the NSF of Shanxi Province (201601D011004).

## Publisher's Note



Received: 17 January 2017 Accepted: 19 April 2017 Published online: 01 May 2017

**References**

1. O'Leary, DP, White, R: Multi-splittings of matrices and parallel solutions of linear systems. SIAM J. Algebraic Discrete Methods **6**, 630-640 (1985)

2. Bai, ZZ: Parallel matrix multisplitting block relaxation iteration methods. Math. Numer. Sin. **17**, 238-252 (1995)
3. Bai, ZZ: On the convergence of additive and multiplicative splitting iterations for systems of linear equations. J. Comput. Appl. Math. **154**, 195-214 (2003)
4. Bai, ZZ: On the convergence of parallel nonstationary multisplitting iteration methods. J. Comput. Appl. Math. **159**, 1-11 (2003)
5. Bai, ZZ: A new generalized asynchronous parallel multisplitting iteration method. J. Comput. Math. **17**, 449-456 (1999)
6. Evans, DJ: Blockwise matrix multi-splitting multi-parameter block relaxation methods. Int. J. Comput. Math. **64**, 103-118 (1997)
7. Sun, JC, Wang, DR: A unified framework for the construction of various matrix multisplitting iterative methods for large sparse system of linear equations. Comput. Math. Appl. **32**, 51-76 (1996)
8. Frommer, A, Mayer, G: Convergence of relaxed parallel multisplitting methods. Linear Algebra Appl. **119**, 141-152 (1989)
9. Neumann, M, Plemmons, RJ: Convergence of parallel multisplitting iterative methods for *M*-matrices. Linear Algebra Appl. **88**, 559-573 (1987)
10. Szyld, DB, Jones, MT: Two-stage and multisplitting methods for the parallel solution of linear systems. SIAM J. Matrix Anal. Appl. **13**, 671-679 (1992)
11. Wang, CL, Meng, GY, Yong, XR: Modified parallel multisplitting iterative methods for non-Hermitian positive definite systems. Adv. Comput. Math. **38**, 859-872 (2013)
12. Wang, DR: On the convergence of the parallel multisplitting AOR algorithm. Linear Algebra Appl. **154-156**, 473-486 (1991)
13. Wen, RP, Wang, CL, Yan, XH: Generalizations of nonstationary multisplitting iterative method for symmetric positive definite linear systems. Appl. Math. Comput. **216**, 1707-1714 (2010)
14. Varga, RS: On recurring theorems on diagonal dominance. Linear Algebra Appl. **13**, 1-9 (1976)
15. Berman, A, Plemmons, RJ: Nonnegative Matrices in the Mathematical Sciences. Classics in Applied Mathematics. SIAM, Philadelphia (1994)
16. Hirsh, RS, Rudy, DH: The role of diagonal dominance and cell Reynolds number in implicit difference methods for fluid mechanics. J. Comput. Phys. **16**, 304-310 (1974)
17. Li, BS: Generalizations of diagonal dominance in matrix theory. A thesis submitted to the Faculty of Graduate Studies and Research in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mathematics and Statistics, University of Regina (1997)
18. Meijerink, JA, Van De Vorst, HA: Iterative Solution of Linear Systems Arising from Discrete Approximations to Partial Differential Equations. Academisch Computer Centrum, Utrecht (1974)
19. Nelder, JA, Mead, R: A simplex method for function minimization. J. Comput. **7**, 308-313 (1965)