Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational and Applied Mathematics 220 (2008) 74–84

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

On a new iterative method for solving linear systems and comparison results

Yan-Fei Jing*, Ting-Zhu Huang[1]

School of Applied Mathematics, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054, P.R. China

Received 19 April 2007

Abstract

In Ujević [A new iterative method for solving linear systems, Appl. Math. Comput. 179 (2006) 725–730], the author obtained a new iterative method for solving linear systems, which can be considered as a modification of the Gauss–Seidel method. In this paper, we show that this is a special case from a point of view of projection techniques. And a different approach is established, which is both theoretically and numerically proven to be better than (at least the same as) Ujević's. As the presented numerical examples show, in most cases, the convergence rate is more than one and a half that of Ujević.

MSC: 65F10

Keywords: Linear system; Projection technique; Petrov–Galerkin condition; Comparison result; Gauss–Seidel method

1. Introduction

In [21], Ujević obtained a new iterative method for solving linear systems, which can be considered as a modification of the Gauss–Seidel method. In fact, virtually the new iterative method can be termed as a "one-dimensional double successive projection method" (referred to as 1D-DSPM) while an elementary Gauss–Seidel method is nothing but a "one-dimensional single successive projection method" (referred to as 1D-SSPM) [17], as will be seen shortly.

We still consider the iterative solution of $n \times n$ nonsingular linear systems of equations

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix (referred to as an *SPD matrix*) and $b \in \mathbb{R}^n$ is given and $x \in \mathbb{R}^n$ is unknown. For solving such linear systems, there has been an explosion of activity in iterative methods spurred by demand due to extraordinary technological advances in engineering and sciences. We refer the reader to the excellent survey [18]. Most of the existing practical iterative techniques for solving large linear systems of equations utilize a projection process in one way or another; see, e.g., [6,19,15,3,5,13,20]. Householder's book [11] contains a fairly good

* Corresponding author.
E-mail addresses: 00jyfvictory@163.com, yanfeijing@hotmail.com (Y.-F. Jing), tzhuang@uestc.edu.cn, tingzhuhuang@126.com (T.-Z. Huang).
[1] Supported by NCET of China (NCET-04-0893).

0377-0427/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2007.07.035

overview of iterative methods—specially oriented towards projection methods. Projection techniques are present in different forms in many other areas of scientific computing and can be formulated in abstract Hilbert functional spaces and finite element spaces. For more general, including nonlinear, projection processes, the reader is recommended to consult [14].

Projection techniques are the process in which one attempts to solve a set of equations by solving each separate equation by a correction that is small in some norm. These techniques could be used for over- or under-determined linear systems, such as those that arise in tomography problems. This has led to the methods of [7,12], which were later identified as instances of Gauss–Jacobi and/or Gauss–Seidel for related systems with $A^{\mathrm{T}}A$ or AA^{T}. The idea of projection process is to extract an approximate solution to (1) from a subspace of $\mathbb{R}^n$. For more details, refer to [17]. Denote $\mathscr{K}$ and $\mathscr{L}$ the search subspace and the constraints subspace, respectively, and let m be their dimension and $x_0 \in \mathbb{R}^n$ be an initial guess to the solution. Typically, a projection method onto the subspace $\mathscr{K}$ and orthogonal to $\mathscr{L}$ is a process which does its possible to find an approximate solution $x \in \mathbb{R}^n$ to (1) by imposing the Petrov–Galerkin conditions that x belong to the affine space $x_0 + \mathscr{K}$ and that the new residual vector be orthogonal to $\mathscr{L}$, i.e.,

$$\text{Find} \quad x \in x_0 + \mathscr{K} \quad \text{such that } b - Ax \perp \mathscr{L}. \tag{2}$$

From this point of view, the new iterative method proposed in [21] can be viewed as a special case of the projection techniques in which two pairs of $\mathscr{K}$ and $\mathscr{L}$ of dimension one are chosen separately while it makes double corrections at each step of the process cycled for $i = 1, \ldots, n$. Therefore, we term it as 1D-DSPM. Further analysis will be presented in Section 2. In addition, an elementary Gauss–Seidel method is a projection method with $\mathscr{K} = \mathscr{L} = \mathrm{span}\{e_i\}$, where e_i is the ith column of the identity matrix. And single correction is made at each step of these projection steps cycled for $i = 1, \ldots, n$. A different approach in Section 3 is obtained with the combination of the double subspaces $\mathscr{K}$ chosen at each step of those of Ujević's and still proceeds to make double corrections at each step of the projection steps cycled for $i = 1, \ldots, n$; that is we impose the Patrov–Galerkin conditions onto the subspace $\mathscr{K}$ and orthogonal to the identical subspace $\mathscr{L}$ of dimension two at each step. We call it as "two-dimensional double successive projection method" (referred to as 2D-DSPM). As the theory in this section indicates, 2D-DSPM gives better (at least the same) reduction of the error than 1D-DSPM. The presented numerical examples in Section 4 show that, in most cases, the convergence rate of 2D-DSPM is more than one and a half that of 1D-DSPM.

Before ending this section, we describe some of the notation we use throughout. Denote e_i the ith column of the identity matrix of appropriate order. By $x_*,\ x_k,\ x_{k+1} \in \mathbb{R}^n$ for any nonnegative integer k, we denote the exact, the current approximate and the latter approximate solution to (1), respectively.

By $\langle x, y\rangle = y^{\mathrm{T}}x$ we denote a vector inner product between the vectors $x, y \in \mathbb{R}^n$. For any positive definite matrix $M \in \mathbb{R}^{n\times n}$, the M-inner product is defined as $\langle x, y\rangle_M = \langle Mx, y\rangle = y^{\mathrm{T}}Mx$. Moreover, if M is an SPD matrix, the corresponding norm is

$$\|x\|_M^2 = \langle Mx, y\rangle = x^{\mathrm{T}}Mx = (Mx)^{\mathrm{T}}x = \langle x, Mx\rangle \quad \text{for any } x \in \mathbb{R}^n.$$

For simplicity and unification of the following illustration and computation, denote $\mathscr{K}_1 = \mathrm{span}\{v_1\}$ and $\mathscr{K}_2 = \mathrm{span}\{v_2\}$ the corresponding representatives of the candidate subspaces at each step in 1D-DSPM, where $0 \neq v_1,\ v_2 \in \mathbb{R}^n$ and $v_1,\ v_2$ is of linear independence. At the same time, the subspace at each step in 2D-DSPM will be $\mathscr{K} = \mathrm{span}\{v_1, v_2\}$.

Since the coefficient matrix $A \in \mathbb{R}^{n\times n}$ considered in (1) is an SPD matrix, simply denote the inner products

$$a = \langle Av_1, v_1\rangle, \quad c = \langle Av_1, v_2\rangle = \langle Av_2, v_1\rangle, \quad d = \langle Av_2, v_2\rangle,$$

and denote

$$p_1 = \langle Ax_k - b, v_1\rangle, \quad p_2 = \langle Ax_k - b, v_2\rangle.$$

2. Interpretation of Ujević's new iterative method in terms of projection techniques

Let us first recall some knowledge of the new iterative method derived in [21]. Making use of some trivial substitutions and computation, we represent again the principles of 1D-DSPM ((3.3) and (3.8) of [21]) in our uniform notation as

follows

$$\begin{cases} x_{k+1} = x_k + \alpha_1 v_1 + \beta_2 v_2, \\ f(x_{k+1}) - f(x_k) = -\dfrac{p_1^2}{2a} - \dfrac{(cp_1 - ap_2)^2}{2a^2 d}, \quad k = 0, 1, 2, \ldots, \end{cases} \tag{3}$$

where, $\alpha_1 = -p_1/a$, $\beta_2 = (cp_1 - ap_2)/ad$, $f(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle$.

As stated in [21], 1D-DSPM updates two components of the approximate solution x_k at the same time while choosing $v_1 = e_i$, $v_2 = e_j$, where j depends on i, $j \neq i$, and it can be considered as a modification of the Gauss–Seidel method. From the projection point of view, we will have a two-step investigation of 1D-DSPM at each step of the process cycled for $i = 1, \ldots, n$.

The first step is to choose the subspaces $\mathscr{K}_1 = \mathscr{L}_1 = \text{span}\{v_1\}$, $x_0 = x_k$, and Eq. (2) turns to

$$\text{Find} \quad \tilde{x}_{k+1} \in x_k + \mathscr{K}_1 \quad \text{such that } b - A\tilde{x}_{k+1} \perp \mathscr{L}_1, \tag{4}$$

where,

$$\tilde{x}_{k+1} = x_k + \tilde{\alpha} v_1.$$

Eq. (6) can be represented in terms of inner products as

$$\langle b - A\tilde{x}_{k+1}, v_1\rangle = 0, \tag{5}$$

which is

$$\begin{aligned} \langle b - Ax_k - \tilde{\alpha} Av_1, v_1\rangle &= \langle b - Ax_k, v_1\rangle - \tilde{\alpha}\langle Av_1, v_1\rangle \\ &= -p_1 - \tilde{\alpha} a \\ &= 0, \end{aligned}$$

giving rise to $\tilde{\alpha} = -p_1/a$, which is the same with α_1 in (3).

The next step is in a similar way to choose the subspaces $\mathscr{K}_2 = \mathscr{L}_2 = \text{span}\{v_2\}$, $x_0 = \tilde{x}_{k+1}$, and this time Eq. (2) turns to

$$\text{Find} \quad x_{k+1} \in \tilde{x}_{k+1} + \mathscr{K}_2 \quad \text{such that } b - Ax_{k+1} \perp \mathscr{L}_2, \tag{6}$$

where,

$$x_{k+1} = \tilde{x}_{k+1} + \tilde{\beta} v_2.$$

Eq. (4) can be represented in terms of inner products as

$$\langle b - Ax_{k+1}, v_2\rangle = 0, \tag{7}$$

which is

$$\begin{aligned} \langle b - A\tilde{x}_{k+1} - \tilde{\beta} Av_2, v_2\rangle &= \langle b - Ax_k - \tilde{\alpha} Av_1 - \tilde{\beta} Av_2, v_2\rangle \\ &= \langle b - Ax_k, v_2\rangle - \tilde{\alpha}\langle Av_1, v_2\rangle - \tilde{\beta}\langle Av_2, v_2\rangle \\ &= -p_2 - \tilde{\alpha} c - \tilde{\beta} d \\ &= -p_2 + \frac{cp_1}{a} - \tilde{\beta} d \\ &= 0, \end{aligned}$$

giving rise to $\tilde{\beta} = (cp_1 - ap_2)/ad$, which is the same with β_2 in (3). After (4) and (6), the same next approximate solution x_{k+1} to (1) can be obtained as in Ujević's new iterative method.

Up to now, it is clear that 1D-DSPM is a special case of the projection methods.

3. 2D-DSPM and its theoretical comparison results with 1D-DSPM

In this section, we present a different approach with the combination of the double subspaces $\mathscr{K}_1, \mathscr{K}_2$ chosen at each step of those of Ujević's in the previous section and still proceeds to make double corrections at each step of the projection steps cycled for $i = 1, \ldots, n$; that is we impose the Patrov–Galerkin conditions onto the subspace $\mathscr{K} = \text{span}\{v_1, v_2\}$ and orthogonal to the identical subspace $\mathscr{L} = \text{span}\{v_1, v_2\}$ at each step, making Eq. (2) become

$$\text{Find} \quad x_{k+1} \in x_k + \mathscr{K} \quad \text{such that } b - Ax_{k+1} \perp \mathscr{L}, \tag{8}$$

where,

$$x_{k+1} = x_k + \alpha v_1 + \beta v_2.$$

Eq. (8) can be represented in terms of inner products as

$$\begin{cases} \langle b - Ax_{k+1}, v_1\rangle = 0, \\ \langle b - Ax_{k+1}, v_2\rangle = 0, \end{cases} \tag{9}$$

which is

$$\begin{cases} \langle b - Ax_k - \alpha Av_1 - \beta Av_2, v_1\rangle &= \langle b - Ax_k, v_1\rangle - \alpha\langle Av_1, v_1\rangle - \beta\langle Av_2, v_1\rangle \\ &= -p_1 - a\alpha - c\beta \\ &= 0, \\ \langle b - Ax_k - \alpha Av_1 - \beta Av_2, v_2\rangle &= \langle b - Ax_k, v_2\rangle - \alpha\langle Av_1, v_2\rangle - \beta\langle Av_2, v_2\rangle \\ &= -p_2 - c\alpha - d\beta \\ &= 0. \end{cases} \tag{10}$$

For the solutions of α and β in (10), the following well-known lemma is needed.

Lemma 1. *a, c, d defined in Section 1 satisfy the following inequalities*

$$\begin{cases} a > 0, \\ d > 0, \\ ad - c^2 > 0. \end{cases} \tag{11}$$

Proof. As defined in Section 1, $a = \langle Av_1, v_1\rangle$, $c = \langle Av_1, v_2\rangle = \langle Av_2, v_1\rangle$, $d = \langle Av_2, v_2\rangle$, where $0 \neq v_1$, $v_2 \in \mathbb{R}^n$ and v_1, v_2 is of linear independence, the first two inequalities are easy to see according to the properties of vector inner products. The proof of the last inequality begins by expanding $\langle A(v_1 - \lambda v_2), v_1 - \lambda v_2\rangle$ with $\lambda \in \mathbb{R}$ as follows

$$\langle A(v_1 - \lambda v_2), v_1 - \lambda v_2\rangle = \langle Av_1, v_1\rangle - 2\lambda\langle Av_1, v_2\rangle + \lambda^2\langle Av_2, v_2\rangle.$$

Since $v_2 \neq 0$ and v_1, v_2 are supposed to be linearly independent, take $\lambda = \langle Av_1, v_2\rangle/\langle Av_2, v_2\rangle$. Then $\langle A(v_1 - \lambda v_2), v_1 - \lambda v_2\rangle > 0$ shows the above equality

$$\begin{aligned} 0 < \langle A(v_1 - \lambda v_2), v_1 - \lambda v_2\rangle &= \langle Av_1, v_1\rangle - 2\frac{\langle Av_1, v_2\rangle^2}{\langle Av_2, v_2\rangle} + \frac{\langle Av_1, v_2\rangle^2}{\langle Av_2, v_2\rangle} \\ &= \langle Av_1, v_1\rangle - \frac{\langle Av_1, v_2\rangle^2}{\langle Av_2, v_2t\rangle} \\ &= a - \frac{c^2}{d}, \end{aligned}$$

which yields the third inequality. □

Thus, solving the equations abstracted from (10)

$$\begin{cases} -p_1 - a\alpha - c\beta = 0, \\ -p_2 - c\alpha - d\beta = 0. \end{cases}$$

We get

$$\begin{cases} \alpha = \dfrac{cp_2 - dp_1}{ad - c^2}, \\ \beta = \dfrac{cp_1 - ap_2}{ad - c^2}. \end{cases} \tag{12}$$

We now describe 2D-DSPM as follows.

Algorithm 1. Two-dimensional double successive projection method (2D-DSPM)

1. Choose an initial guess $x_0 \in \mathbb{R}^n$ to (1)
2. For $k = 0, 1, 2, \ldots,$ until convergence, Do:
3. $z_1 = x_k$
4. For $i = 1, \ldots, n,$ Do:
5. $z_{i+1} = z_i + \alpha v_1 + \beta v_2$
6. EndDo for i
7. $x_{k+1} = z_{n+1}$
8. EndDo for k.

where α and β are computed as (12). As noted in [21], $v_1,\ v_2$ can be arbitrary elements of $\mathbb{R}^n$. However, suitable pairs of $v_1,\ v_2$ at each step should be chosen for the efficiency in real applications. The relation of the effect with respect to error reduction between 2D-DSPM and 1D-DSPM will be shown both in Theorem 4 and Corollary 8. The result for specific choices of $v_1,\ v_2$ will be seen in Corollary 5 and striking numerical comparison results will be given in Section 4 later.

We also consider the following problem:

$$f(x) = \tfrac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle \to \inf. \tag{13}$$

Before giving the theoretical comparison results with respect to error reduction between 2D-DSPM and 1D-DSPM, we first show that the reduction of f in the above form is equivalent to the reduction of the error in Lemma 2 and then we present in Theorem 3 the reduction between $f(x_k)$ and $f(x_{k+1})$ when x_{k+1} is computed with 2D-DSPM.

Lemma 2 (*Ujević [21]*). *The reduction between $f(x_k)$ and $f(x_{k+1})$ is of equivalence to the reduction of error $= x - x_*$ in the A-norm when f is in the form of* (13).

Proof. The proof is easy as follows

$$\begin{aligned} \|x_{k+1} - x_*\|_A - \|x_k - x_*\|_A &= \langle Ax_{k+1} - Ax_*, x_{k+1} - x_*\rangle - \langle Ax_k - Ax_*, x_k - x_*\rangle \\ &= \langle Ax_{k+1}, x_{k+1}\rangle - 2\langle b, x_{k+1}\rangle - (\langle Ax_k, x_k\rangle - 2\langle b, x_k\rangle) \\ &= 2f(x_{k+1}) - 2f(x_k). \qquad \square \end{aligned} \tag{14}$$

Theorem 3. *2D-DSPM gives the reduction between $f(x_k)$* and *$f(x_{k+1})$ as follows*

$$f(x_k) - f(x_{k+1}) = \frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)}. \tag{15}$$

Proof. Since $x_{k+1} = x_k + \alpha v_1 + \beta v_2$ computed in 2D-DSPM, with some trivial computation,

$$\begin{aligned} f(x_{k+1}) &= f(x_k + \alpha v_1 + \beta v_2) \\ &= f(x_k) + \alpha\langle Ax_k - b, v_1\rangle + \beta\langle Ax_k - b, v_2\rangle \\ &\quad + \tfrac{1}{2}\alpha^2\langle Av_1, v_1\rangle + \alpha\beta\langle Av_1, v_2\rangle + \tfrac{1}{2}\beta^2\langle Av_2, v_2\rangle \\ &= f(x_k) + p_1\alpha + p_2\beta + \tfrac{1}{2}a\alpha^2 + c\alpha\beta + \tfrac{1}{2}d\beta^2. \end{aligned} \tag{16}$$

Let

$$g(\alpha, \beta) = p_1\alpha + p_2\beta + \tfrac{1}{2}a\alpha^2 + c\alpha\beta + \tfrac{1}{2}d\beta^2.$$

A better reduction between $f(x_k)$ and $f(x_{k+1})$ requires to minimize $g(\alpha, \beta)$ in terms of α, β. And the necessity to minimize $g(\alpha, \beta)$ gives birth to the following equations

$$\begin{cases} \dfrac{\partial g}{\partial \alpha} = p_1 + a\alpha + c\beta = 0, \\ \dfrac{\partial g}{\partial \beta} = p_2 + c\alpha + d\beta = 0, \end{cases}$$

which have the same solutions presented in (12). By trivial symbolic computation, the expressions of α, β in (12) show (16),

$$f(x_{k+1}) = f(x_k) - \frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)},$$

which completes the proof. □

Now we depict the comparison results with respect to error reduction between 2D-DSPM and 1D-DSPM.

Theorem 4. *2D-DSPM gives a better (at least the same) reduction of the function f in the form of* (13) *than 1D-DSPM*; *in other words*, *2D-DSPM gives a better (at least the same) reduction of the error than 1D-DSPM.*

Proof. As (3) reveals, 1D-DSPM gives the reduction between $f(x_k)$ and $f(x_{k+1})$ as

$$f(x_k) - f(x_{k+1}) = \frac{p_1^2}{2a} + \frac{(cp_1 - ap_2)^2}{2a^2d}. \tag{17}$$

The work we have to do is to compare (15) and (17). Subtract (17) from (15), we have

$$\frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)} - \left(\frac{p_1^2}{2a} + \frac{(cp_1 - ap_2)^2}{2a^2d}\right) = \frac{c^2(cp_1 - ap_2)^2}{2a^2d(ad - c^2)} \geqslant 0,$$

which proves the first part of the assertion and the second part follows immediately by (14). □

Corollary 5. *If* $\langle Av_1, v_2\rangle = 0$ *or* $\langle Ax_k - b, cv_1 - av_2\rangle = 0$ *at each step of 2D-DSPM*, *then 1D-DSPM and 2D-DSPM have the same reduction effect*; *If* $\langle Ax_k - b, cv_1 - av_2\rangle = 0$ *at each step of both 1D-DSPM and 2D-DSPM*, *the reduction effects of 1D-SSPM*, *1D-DSPM and 2D-DSPM are all the same*, *and both of 1D-DSPM and 2D-DSPM regress to 1D-SSPM*, *i.e., the Gauss–Seidel method.*

Proof. From Theorems 3 and 4, it can be easily showed as follows. On the one hand, if $\langle Av_1, v_2\rangle = 0$, that is to say $c = 0$, which yields for both 1D-DSPM and 2D-DSPM the same reduction effect as $(p_1^2/2a) + (p_2^2/2d)$ in terms of f.

On the other hand, if $\langle Ax_k - b, cv_1 - av_2\rangle = 0$, which means

$$cp_1 - ap_2 = 0,$$

then 1D-SSPM, 1D-DSPM and 2D-DSPM all give the reduction of $p_1^2/2a$ in terms of f and in such cases 1D-DSPM and 2D-DSPM both regress to 1D-SSPM, which completes the proof. □

It should be noted that different choices of pairs of v_1, v_2 at each step determine different reductions of the error, which is also of dependence on the interrelationship between the two vectors with respect to the coefficient matrix as well as on the relation between the current residual vector and the two vectors v_1, v_2, as is somewhat revealed in the above corollary and will be seen in the numerical examples.

From Theorem 4 and Corollary 5, we can see that the reduction of the error in 2D-DSPM is better than (at least the same as) that in 1D-DSPM. Before giving a further relation between (15) and (17) in the next theorem, a lemma is needed.

Lemma 6. *If $cp_1(cp_1 - ap_2) \geqslant 0$ and $dp_1 \neq cp_2$ are satisfied, then the following inequality holds*

$$0 \leqslant \frac{c^2(cp_1 - ap_2)^2}{a^2(dp_1 - cp_2)^2} < 1. \tag{18}$$

Proof. First (18) exists because of its nonzero denominator for $dp_1 \neq cp_2$. Then if $a^2(dp_1 - cp_2)^2 - c^2(cp_1 - ap_2)^2 > 0$, then (18) holds. In fact, by trivial computation, we have

$$a^2(dp_1 - cp_2)^2 - c^2(cp_1 - ap_2)^2 = (ad - c^2)((ad + c^2)p_1^2 - 2acp_1p_2). \tag{19}$$

By Lemma 1 and assumptions, we have

$$\begin{cases} ad - c^2 > 0, \\ (ad + c^2)p_1^2 - 2acp_1p_2 > 2c^2p_1^2 - 2acp_1p_2 = 2cp_1(cp_1 - ap_2) \geqslant 0. \end{cases}$$

From the above inequalities, (19) is positive, and we are done. □

Theorem 7. *If the conditions in Lemma* 6 *are satisfied, then the reduction of the function f in the form of* (13) *between 2D-DSPM and 1D-DSPM have the following relation*

$$1 \leqslant \frac{\Delta f_{2D-DSPM}}{\Delta f_{1D-DSPM}} \leqslant \frac{1}{1 - c^2(cp_1 - ap_2)^2/a^2(dp_1 - cp_2)^2}, \tag{20}$$

where $\Delta f_{2D-DSPM}$ and $\Delta f_{1D-DSPM}$ denote (15) *and* (17), *respectively*.

Proof. The left-hand side of (20) is straightforward from the proof of Theorem 4. The proof of the right-hand side inequality begins by setting

$$\omega \Delta f_{2D-DSPM} = \Delta f_{1D-DSPM}$$

with a scalar ω,

$$\omega \frac{dp_1^2 + ap_2^2 - 2cp_1p_2}{2(ad - c^2)} = \frac{p_1^2}{2a} + \frac{(cp_1 - ap_2)^2}{2a^2d}.$$

Expanding the above equality follows

$$\omega = 1 - \frac{c^2(cp_1 - ap_2)^2}{a^2(d^2p_1^2 - 2cdp_1p_2 + adp_2^2)}.$$

Observing that $ad > c^2$, we obtain

$$\omega \geqslant 1 - \frac{c^2(cp_1 - ap_2)^2}{a^2(dp_1 - cp_2)^2} > 0,$$

where the above equality holds in the cases appearing in Corollary 5, and the result follows immediately by Lemma 6. □

Corollary 8. *The relation of the error reduction between 2D-DSPM and 1D-DSPM has the same form as* (20).

Proof. It is direct by lemma 2 and the preceding theorem. □

Here, the convergence of 2D-DSPM can be guaranteed for solving linear systems of equations of (1).

4. Numerical comparison results

In this section, we compare our 2D-DSPM approach with the given methods presented in [21] with two classes of matrices: one is as the presented in [21, Example 1]; the other is the coefficient matrix of the linear systems generated by the discretization of two-dimensional partial differential equations appearing in [1,2,9,16,4,8], etc. First we present in detail a generalized algorithm similar to the particular method stated in [21] for implementing the process of the iterative solution to (1).

We choose: $v_1 = e_i$, $v_2 = e_j$, where j depends on i, $j \neq i$, $i = 1, \ldots, n$. Then from Algorithm 1, we have exactly at each step

$$x_{k+1} = x_k + \alpha_i e_i + \beta_i e_j \quad \text{for } k = 0, 1, \ldots,$$

where

$$\begin{cases} \alpha_i = \dfrac{a_{ij} p_j - a_{jj} p_i}{a_{ii} a_{jj} - a_{ij}^2}, \\ \beta_i = \dfrac{a_{ij} p_i - a_{ii} p_j}{a_{ii} a_{jj} - a_{ij}^2}, \\ p_i = \langle a_i, x_k \rangle - b_i, \\ p_j = \langle a_j, x_k \rangle - b_j. \end{cases}$$

Here, a_i, a_{ij} denote the ith column and the (i, j)th entry of $A \in \mathbb{R}^{n \times n}$, respectively, and b_i denotes the ith element of $b \in \mathbb{R}^n$, for $i, j = 1, \ldots, n$. In a generalized way, we choose $j = i - ij_{\text{gap}}$ for $i = 1, \ldots, n$ and $j = i - ij_{\text{gap}} + n$ if $i \leqslant ij_{\text{gap}}$, where ij_{gap} is an introduced positive integral parameter which is less than n. The above results provide the next algorithm.

Algorithm 2. A particular implementation of 2D-DSPM in a generalized way

1. Choose an initial guess $x_0 \in \mathbb{R}^n$ to (1) and a prescribed $ij_{\text{gap}} (< n)$
2. Until convergence, Do:
3. $x = x_0$
4. For $i = 1, \ldots, n$, Do:
5. $j = i - ij_{\text{gap}}$
6. If $i \leqslant ij_{\text{gap}}$, *then*
7. $j = i - ij_{\text{gap}} + n$
8. Endif
9. $p_i = \langle a_i, x \rangle - b_i$
10. $p_j = \langle a_j, x \rangle - b_j$
11. $\mu_i = a_{ii} a_{jj} - a_{ij}^2$
12. $\alpha_i = \dfrac{a_{ij} p_j - a_{jj} p_i}{\mu_i}$
13. $x_i = x_i + \alpha_i$
14. $\beta_i = \dfrac{a_{ij} p_i - a_{ii} p_j}{\mu_i}$
15. $x_j = x_j + \beta(i)$
16. EndDo for i
17. $x_0 = x$
18. Stopping criteria
19. EndDo

Table 1
Comparison results for Example 1

ij_{gap}	it_{1D}	it_{2D}
1	6	7
2	13	6
100	13	6
500	13	7
999	13	7

Table 2
Comparison results for Example 2

ij_{gap}	it_{1D}	it_{2D}
1	8	8
2	14	8
3	14	9
100	14	9
500	15	10
999	14	8

It can be observed that the Algorithm 1 in [21] is the special case when $ij_{\text{gap}} = 1$ in 1D-DSPM. In the following, the systems of linear equations (1) will be solved with a PC-Pentium(R) 4, CPU 3.06 GHz, 512 M of RAM and performed in MATLAB 6.5 with machine precision 10^{-16}. Let $b = Ae$, where e is the $n \times 1$ vector whose elements are all equal to unity, such that $x = (1, 1, \ldots, 1)^{\mathrm{T}}$ is the exact solution to (1). The stopping criteria is $\|x_{k+1} - x_k\| < 10^{-6}$. All these tests here are started with an initial guess equal to $x_0 = (x^1, \ldots, x^n)$, $x^i = 0.001 * i$, $i = 1, \ldots, n$.

Now, we give the numerical comparison results in terms of iteration number of 1D-DSPM and 2D-DSPM as follows. Denote it_{1D}, it_{2D} the iteration number of 1D-DSPM and 2D-DSPM, respectively.

Example 1 (*Ujević [21]*). Let the matrix A be given by

$$a_{ii} = 4n, \quad a_{i,i+1} = a_{i+1,i} = n, \quad a_{ij} = 0.5 \quad \text{for } i = 1, \ldots, n, \ j \neq i, \ i+1.$$

In order to compare the convergence rate between 2D-DSPM and 1D-DSPM, we also choose $n = 1000$.

We separately solve the above problem by Algorithm 2 and by its counterpart in [21] with different ij_{gap}, which means to choose different pairs of v_1, v_2 at each step. Then we shall see, in most cases, the convergence rate of 2D-DSPM is more than twice that of 1D-DSPM according to the comparison results listed in Table 1.

A further observation is made that at each step of 2D-DSPM when $ij_{\text{gap}} = 1$, one of the first part of Corollary 5 holds; that is the reduction effect is about the same for 1D-DSPM and 2D-DSPM. In fact, $\langle Ax_k - b, cv_1 - av_2\rangle$ at each step in 2D-DSPM shows $(0, -0.3638, -0.3638, 0, 0, 0) \times 10^{-8}$ while $\langle Ax_k - b, cv_1 - av_2\rangle$ at each step in 1D-DSPM shows $(-3.5490, 0.0625, 0.0003, -0.0001, -0.0000, -0.0000) \times 10^3$. Therefore, in a sense of efficiency, in [21, Algorithm 1] obtained the optimal choices for v_1, v_2. However, it seems more difficult in real applications to choose the suitable pairs of v_1, v_2 beforehand. Whereas, 2D-DSPM always seems to give its best error reduction in this case whatever v_1, v_2 are chosen.

Example 2. Let the matrix A be the same as in the previous example except that the diagonal entries turn to

$$a_{ii} = 3n \quad \text{for } i = 1, \ldots, n.$$

And the other conditions remain the same with those in Example 1. We can obtain the relation of the convergence rate between 2D-DSPM and 1D-DSPM is about one and a half as seen in Table 2.

Table 3
Comparison results for Example 3

ij_{gap}	Case 1		Case 2		Case 3	
	it_{1D}	it_{2D}	it_{1D}	it_{2D}	it_{1D}	it_{2D}
1	391	376	321	302	281	287
2	611	391	481	312	476	302
100	611	322	481	255	476	249
500	611	323	481	256	476	250
999	611	391	481	313	476	302

Example 3. For convenience of comparison, consider the two-dimensional partial differential equations on the unit square region $\Omega = [0, 1] \times [0, 1]$ of the form

$$a(x, y)u_{xx} + b(x, y)u_{yy} + c(x, y)u_x + d(x, y)u_y + e(x, y)u(x, y) = f(x, y), \tag{21}$$

where $a(x, y),\ b(x, y),\ c(x, y),\ d(x, y),\ e(x, y)$ are given real valued functions. Here, we mention three cases for the choices of these functions with Dirichlet-type boundary conditions for the purpose of comparison.

Case 1: $a(x, y) = -1,\ b(x, y) = -1,\ c(x, y) = 0,\ d(x, y) = 10(x + y),\ e(x, y) = 10(x - y),\ f(x, y) = 0$;

Case 2: $a(x, y) = -1,\ b(x, y) = -1,\ c(x, y) = -10(x + y),\ d(x, y) = -10(x - y),\ e(x, y) = 1,\ f(x, y) = 0$;

Case 3: $a(x, y) = -1,\ b(x, y) = -1,\ c(x, y) = 10e^{xy},\ d(x, y) = 10e^{-xy},\ e(x, y) = 0,\ f(x, y) = 0$.

Using five-point finite difference scheme to discretize these above problems with a uninform grid of mesh spacing $\Delta x = \Delta y = 1/(m + 1)$ in x and y directions, respectively, we can obtain different symmetric positive definite matrices of order $m \times m$ as m varies. For details on symmetric positive definite matrices arising in discretizations, refer to [10, Chapter 7]. In particular, the choice of $m = 32$ results in $m \times m = 32 \times 32$ matrices. The comparison results for the three cases mentioned above between 1D-DSPM and 2D-DSPM are shown in Table 3, which further strengthen the superiority of 2D-DSPM to 1D-DSPM.

5. Concluding remarks

In this paper, Ujević's new iterative method [21] is investigated from a point of view of projection techniques, which can be considered as a special case of the projection methods as analyzed in Section 2, and we term it as 1D-DSPM. A different approach with the name of 2D-DSPM has been established, which shows a better (at least the same) effective error reduction than 1D-DSPM by theoretical analysis and the corresponding comparison results at large. Its particular implementation in a generalized way is given, whose convergence rate is always more than one and a half that of Ujević's with the same $v_1,\ v_2$, shown by the presented numerical examples.

It should be observed that the convergence rates with different choices of pairs of $v_1,\ v_2$ at each step of the process cycled for $i = 1, \ldots, n$ behave differently with respect to the error reduction. Since the optimal pairs v_1, v_2 seem to be difficult to predict in real scientific computing, we just gave some numerical comparison results for different choices of v_1, v_2 by trial and error in this paper. Therefore, what we are concerned about next is the study on these behaviors and on the optimal choices for the pairs of v_1 and v_2 in advance.

Finally, we have to point out a serious problem which has been observed by Prof. Eugene L. Wachspress. It is well known that successive overrelaxation leads to a great improvement when the underlying iteration is consistently ordered. Although we may improve on Gauss–Seidel, we lose the ordering and can no longer realize the SOR again. This is a major reason for applying overlapping block iteration.

Acknowledgments

We thank Prof. Eugene L. Wachspress for carefully reading our draft and helping us point out the serious problem addressed in the last part of this paper.

References

[1] M. Benzi, W. Joubert, A. van Duin, Orderings for incomplete factorization preconditioning of nonsymmetric problems, SIAM J. Sci. Comput. 20 (1999) 1652–1670.
[2] M. Benzi, D.B. Szyld, G. Mateescu, Numerical experiments with parallel ordering for ILU preconditioners, Electron. Trans. Numer. Anal. 8 (1999) 88–114.
[3] A. Bj´orck, T. Elfving, Accelerated projection methods for computing psedu-inverse solutions of systems of linear equations, BIT 19 (1976) 145–163.
[4] M.I.G. Bloor, M.J. Wilson, Generating parametrization of wing geometries using partial differential equations, Comput. Methods Appl. Mech. Engrg. 148 (1997) 125–138.
[5] R. Bramley, A. Sameh, Row projection methods for large nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 13 (1993) 168–193.
[6] C. Brezinski, Projection Methods for Systems of Equations, North-Holland, Amsterdam, 1997.
[7] G. Cimmino, Calcolo approssimato per le soluzioni dei sistemi di equazioni lineari, Ric. Sci. Progr. Tecn. Econom. Naz. 9 (1938) 326–333.
[8] D.J. Evans, Direct methods of solution of partial differential equations with periodic boundary conditions, Math. Comput. Simulation XXI (1979) 270–275.
[9] G.H. Golub, D. Vanderstraeten, On the preconditioning of matrices with skew-symmetric splittings, Numer. Algorithms 25 (2000) 223–239.
[10] R.A. Horn, C.R. Johnson, Matrix Analysis, Originally published by Cambridge University Press in 1986.
[11] A.S. Householder, Theory of Matrices in Numerical Analysis, Blaisdell, Johnson, CO, 1964.
[12] G.S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichngen, Bull. Internal. Acad. Pol. Sci. Lett. A 35 (1937) 355–357.
[13] C. Kamath, A. Sameh, A projection method for solving nonsymmetric linear systems on multiprocessors, Parallel Comput. 9 (1988–89) 291–312.
[14] M.A. Krasnoselskii, et al., Approximate Solutions of Operator Equations, Wolters-Nordhoff, Groningen, 1972.
[15] L. Lopez, V. Simoncini, Analysis of projection methods for rational function approximation to the matrix exponential operator, SIAM J. Numer. Anal. 44 (2006) 613–635.
[16] Y. Saad, Preconditioning techniques for nonsymmetric and indefinite linear systems, J. Comput. Appl. Math. 24 (1988) 89–105.
[17] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., The PWS Publishing Company, Boston, 1996 SIAM, Philadelphia, 2003.
[18] Y. Saad, H.A. van der Vorst, Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math. 43 (2005) 1155–1174.
[19] V. Simoncini, Variable accuracy of matrix-vector products in projection methods for eigencomputation. Technical Report, Universit*à* di Bologna, 2004; SIAM J. Numer. Anal.
[20] K. Tanabe, Projection method for solving a singular system of linear equations and its applications, Numer. Math. 17 (1971) 203–214.
[21] N. Ujević, A new iterative method for solving linear systems, Appl. Math. Comput. 179 (2006) 725–730.