Vol.46, No.3 Journal of Shanghai Normal University (Natural Sciences) Jun., 2017
DOI:10.3969/J.ISSN.100-5137.2017.03.008

Comments on new iterative methods for solving linear systems

Wang Ke[1], Tan Lijun[1], Wang Shiheng[2*]

(1. College of Sciences, Shanghai University, Shanghai 200444, China;

2. Department of Basic Education, Nanyang Vocational College of Agriculture, Nanyang 473000, China)

Abstract: Some new iterative methods were presented by Du, Zheng and Wang for solving linear systems in [3], where it is shown that the new methods, comparing to the classical Jacobi or Gauss-Seidel method, can be applied to more systems and have faster convergence. This note shows that their methods are suitable for more matrices than positive matrices which the authors suggested through further analysis and numerical examples.

Key words: iterative method; linear system; classical iteration

CLC number: O 241.6 **Document code:** A **Article ID:** 1000-5137(2017)03-0406-04

2000 MSC: 15A06, 65F10

关于解线性系统的新迭代方法的注释

王　珂[1], 谭丽君[1], 王世恒[2*]

(1. 上海大学 理学院，上海 200444; 2. 南阳农业职业学院 基础部，南阳 473000)

摘　要：在文 [3] 中作者们提出了几种求解线性系统的新迭代方法，与经典的 Jacobi 或 Gauss-Seidel 方法相比，这些方法可以被应用到更多的线性系统且有更快的收敛速度. 通过分析和数值算例说明他们的方法适合更一般的矩阵，而不仅仅是文 [3] 作者们提到的只适合正矩阵.

关键词：迭代法；线性系统；经典迭代

The nonsingular linear system

$$\boldsymbol{A}x = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{A} = (a_{ij}) \in \mathcal{R}^{n\times n}$, $x \in \mathcal{R}^n$ and $\boldsymbol{b} \in \mathcal{R}^n$, has many applications in scientific computing[1−2]. In [3], Du, Zheng and Wang discussed some new iterative methods for solving (1). This note will show that their methods are suitable for matrices other than positive matrices.

Received date: 2016-10-19

Foundation item: This research was supported by National Natural Science Foundation of China (11301330); grants of "The First-class Discipline of Universities in Shanghai" and Gaoyuan Discipline of Shanghai.

∗**Corresponding author:** Wang Shiheng, associate professor, reseach area: advanced algebra. E-mail: 77917092@qq.com

For (1), Du, Zheng and Wang[3] proposed the following two iterative schemes (2) and (3),

$$x^{(k+1)} = T_1 x^{(k)} + D_1^{-1} b, \quad k = 0, 1, 2, \cdots, \tag{2}$$

where $\boldsymbol{T}_1 = \boldsymbol{D}_1^{-1}\boldsymbol{E}_1$,

$$\boldsymbol{D}_1 = \begin{bmatrix} a_{11} & & & & & \\ & a_{22} & & & & \\ & & \ddots & & & \\ a_{l1} & a_{l2} & \cdots & a_{ll} & \cdots & a_{ln} \\ & & & & \ddots & \\ & & & & & a_{nn} \end{bmatrix}, \quad 1 \leqslant l \leqslant n,$$

and $\boldsymbol{E}_1 = -(\boldsymbol{A} - \boldsymbol{D}_1)$, and

$$x^{(k+1)} = T_2 x^{(k)} + D_2^{-1} b, \quad k = 0, 1, 2, \cdots, \tag{3}$$

where $\boldsymbol{T}_2 = \boldsymbol{D}_2^{-1}\boldsymbol{E}_2$,

$$\boldsymbol{D}_2 = \begin{bmatrix} a_{11} & & & & & \\ & \ddots & & & & \\ & & a_{ii} & a_{ij} & & \\ & & & \ddots & & \\ & & & a_{jj} & & \\ & & & & \ddots & \\ & & & & & a_{nn} \end{bmatrix}, \quad 1 \leqslant i, j \leqslant n,$$

and $\boldsymbol{E}_2 = -(\boldsymbol{A} - \boldsymbol{D}_2)$.

The convergence theorems for (2) and (3) are as below.

Theorem 1 [3] Let $\boldsymbol{A} = (a_{ij})_{n\times n}$, $a_{ij} > 0$, $\sum\limits_{j=1, j\neq i}^{n} |a_{ij}| \leqslant |a_{ii}|$, $n \geqslant 3$. Then the iteration matrix $\boldsymbol{T}_1$ of the method (2) satisfies $\rho(\boldsymbol{T}_1) < 1$, i.e., the iterative method (2) converges.

Theorem 2 [3] Let $\boldsymbol{A} = (a_{ij})_{n\times n}$, $a_{ij} > 0$, $\sum\limits_{j=1, j\neq i}^{n} |a_{ij}| \leqslant |a_{ii}|$, $n \geqslant 3$. Then the iteration matrix $\boldsymbol{T}_2$ of the method (3) satisfies $\rho(\boldsymbol{T}_2) < 1$, i.e., the iterative method (3) converges.

For Theorem 1, they addressed the following remarks.

Remark 1 [3] 1. The size n of the matrix $\boldsymbol{A}$ has to satisfy $n \geqslant 3$. For instance, let

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Then $\boldsymbol{T}_1 = \boldsymbol{D}_1^{-1}\boldsymbol{E}_1 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$, and $\rho(\boldsymbol{T}_1) = 1$.

2. The condition $a_{ij} > 0$ is necessary. For instance, let $\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. Then

$$\boldsymbol{D}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } \boldsymbol{E}_1 = -\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix},$$

and

$$\boldsymbol{T}_1=\boldsymbol{D}_1^{-1}\boldsymbol{E}_1=-\begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

whose eigenvalues are $0,\ 0,\ 1$ and -1. So $\rho(\boldsymbol{T}_1)=1$.

3. Notice that the iterative method (2) converges even if the matrix $\boldsymbol{A}$ is just diagonally dominant.

In fact, the first two of Remark 1 are not necessary. For the first one, assume $\boldsymbol{A}=\begin{bmatrix}a&b\\c&d\end{bmatrix}$, then

$$\boldsymbol{T}_1=\boldsymbol{D}_1^{-1}\boldsymbol{E}_1=\begin{bmatrix}\dfrac{bc}{ad}&0\\-\dfrac{c}{d}&0\end{bmatrix}\text{ or }\begin{bmatrix}0&-\dfrac{b}{a}\\0&\dfrac{bc}{ad}\end{bmatrix},$$

so $\rho(\boldsymbol{T}_1)=\left|\dfrac{bc}{ad}\right|<1$ as $|bc|\leqslant|ad|$ and the equivalence does not hold with $\boldsymbol{A}$ nonsingular. What the authors addressed is only the case where $\boldsymbol{A}$ is singular.

As for the second one, it's the same. The authors just presented an example with a singular coefficient matrix $\boldsymbol{A}$.

The following examples illustrate the effectiveness of the new methods (2) and (3) to diagonally dominant matrices with zero and/or negative entries, compared with the classical Jacobi and Gauss-Seidel iterative methods. The initial guess is 0 and the stopping criterion is

$$\frac{\left\|r^{(k)}\right\|_2}{\left\|r^{(0)}\right\|_2}<10^{-6},$$

where $r^{(k)}$ is the residual vector after k iterations. The numerical results are listed in Tables 1-3, where Jacobi, GS, I and II stand for Jacobi method, Gauss-Seidel method, methods (2) and (3), respectively.

Example 1 Consider the $n\times n$ linear system (1) with

$$\boldsymbol{A}=(a_{ij})_{n\times n}=\begin{cases}a_{ij}=1, & i=j,\\ a_{ij}=1, & i=n,\ j=1,\\ a_{ij}=0, & i=n,\ j=2,\cdots,n-1,\quad 1\leqslant i,j\leqslant n,\\ a_{ij}=-\dfrac{1}{n}, & \text{other } i\neq j,\end{cases}$$

and $\boldsymbol{b}=(1,0,\cdots,0)^{\mathrm{T}}$.

Table 1 Iterations (IT), CPU time (t) and relative error (ERR) for Example 1

n	Jacobi			GS			I			II		
	IT	t	ERR	IT	t	ERR	IT	t	ERR	IT	t	ERR
300	633	0.1	1.0×10^{-6}	358	0.2	9.9×10^{-7}	3	0.0	3.7×10^{-8}	3	0.0	3.7×10^{-8}
400	788	0.2	1.0×10^{-6}	449	0.4	1.0×10^{-6}	3	0.0	1.6×10^{-8}	3	0.0	1.6×10^{-8}
500	930	0.6	1.0×10^{-6}	534	1.0	1.0×10^{-6}	3	0.0	8.0×10^{-9}	3	0.0	8.0×10^{-9}

Table 1 shows that the new iterative methods (2) and (3) are much better than Jacobi and Gauss-Seidel methods. For all iterations, CPU times and precisions of methods (2) and (3) are the same.

Example 2 Consider the $n \times n$ dense linear system (1) with

$$\boldsymbol{A} = (a_{ij})_{n\times n} = \begin{cases} a_{ij} = -\dfrac{1}{2} - \dfrac{i}{2n}, & i < j, \\ a_{ij} = a_{ji}, & i > j, \\ a_{ij} = -\sum\limits_{k\neq i} a_{ik} + 1 + \dfrac{i}{n}, & i = j, \end{cases} \quad 1 \leqslant i, j \leqslant n,$$

and $\boldsymbol{b} = (1, 1, \cdots, 1)^{\mathrm{T}}$.

Table 2 Iterations (IT), CPU time (t) and relative error (ERR) for Example 2

n	Jacobi			GS			I			II		
	IT	t	ERR	IT	t	ERR	IT	t	ERR	IT	t	ERR
1000	181	0.5	6.3×10^{-7}	125	0.8	4.8×10^{-7}	157	0.4	4.0×10^{-7}	181	0.5	6.2×10^{-7}
2000	221	2.0	8.0×10^{-7}	158	3.6	5.0×10^{-7}	187	1.6	9.4×10^{-7}	221	2.0	7.9×10^{-7}
3000	249	4.5	7.1×10^{-7}	174	8.9	7.9×10^{-7}	219	3.8	9.8×10^{-7}	249	4.5	7.2×10^{-7}

In this example, the new iterative method (2) has less iterations and CPU time than Jacobi method, and has less CPU time than Gauss-Seidel method; the method (3) is as good as Jacobi method and better than Gauss-Seidel method.

Example 3 Consider the $n \times n$ dense linear system (1) with

$$\boldsymbol{A} = (a_{ij})_{n\times n} = \begin{cases} a_{ij} = n, & i = j, \\ a_{ij} = 1 - n, & i = 1,\ j = n, \\ a_{ij} = n - 1, & i = n,\ j = 1, \\ a_{ij} = \dfrac{1}{n-1}, & i = 1,\ j = 2, \cdots, n-1, \\ a_{ij} = -\dfrac{1}{n-1}, & i = n,\ j = 2, \cdots, n-1, \\ a_{ij} = 1, & \text{other } i \neq j, \end{cases} \quad 1 \leqslant i, j \leqslant n,$$

and $\boldsymbol{b} = (1, 2, \cdots, n)^{\mathrm{T}}$.

Table 3 Iterations (IT), CPU time (t) and relative error (ERR) for Example 3

n	Jacobi			GS			I			II		
	IT	t	ERR	IT	t	ERR	IT	t	ERR	IT	t	ERR
300	3454	0.4	1.0×10^{-6}	1725	0.7	1.0×10^{-6}	1724	0.3	1.0×10^{-6}	1358	0.2	9.9×10^{-7}
400	4548	0.9	1.0×10^{-6}	2272	1.8	1.0×10^{-6}	2271	0.5	1.0×10^{-6}	1813	0.4	1.0×10^{-6}
500	5629	2.7	1.0×10^{-6}	2812	4.5	1.0×10^{-6}	2811	1.5	1.0×10^{-6}	2269	1.2	1.0×10^{-6}

In Table 3, the method (2) has less CPU time than Jacobi and Gauss-Seidel methods. The method (3) has less both iterations and CPU time than Jacobi and Gauss-Seidel methods.

References:

[1] Varga R. Matrix iterative analysis [M]. New Jersey: Prentice-Hall, Englewood Cliffs, 1962.

[2] Young D. Iterative solution of large linear systems [M]. New York: Academic Press, 1971.

[3] Du J, Zheng B, Wang L. New iterative methods for solving linear systems [J]. Journal of Applied Analysis and Computation, 2011, 1:351-360.

(责任编辑：包震宇)