 CMES, vol.91, no.1, pp.17-42, 2013

A Two-Side Equilibration Method to Reduce the Condition Number of an Ill-Posed Linear System

Chein-Shan Liu[1]

Abstract: In the present paper, we propose a novel two-side equilibration method to properly reduce the condition number of a given non-singular matrix only through a few operations. Then, two different conditioners together with the conjugate gradient method (CGM) are developed, which can overcome the defect of CGM, being not vulnerable to noisy disturbance exerted on an ill-posed linear system. The two-side CGM (TSCGM) and the pre-conditioning CGM (PrCGM) are convergent fast and accurate in solving linear inverse problems and the linear Hilbert problem under a large random noise.

Keywords: Ill-posed linear equations system, Inverse problem, Equilibrated conditioning, Conjugate gradient method

1 Introduction

In this paper we propose two *equilibrated conditioning* conjugate gradient methods to solve an ill-posed linear equations system:

$$\mathbf{B}\mathbf{x} = \mathbf{b}_0, \tag{1}$$

where $\det(\mathbf{B}) \neq 0$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$ might be an ill-conditioned, and generally unsymmetric coefficient matrix. The solution of such an ill-posed system of linear equations is an important issue for many engineering applications. In practice, in the linear equations which arise in engineering problems, the data $\mathbf{b}_0$ are usually not given exactly; instead of, the noises in $\mathbf{b}_0$ are unavoidable due to the measurement error. Therefore, we may encounter the problem that the numerical solution of an ill-posed linear equations system will deviate from the exact one to a great extent, when $\mathbf{b}_0$ is polluted by noise, whose measurement error will be largely amplified by the small singular values of $\mathbf{B}$.

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan.
E-mail: liucs@ntu.edu.tw

We can use the condition number to judge whether a given non-singular matrix $\mathbf{B}$ is ill-conditioned or not, which is defined as

$$\text{Cond}(\mathbf{B}) = \|\mathbf{B}\|\|\mathbf{B}^{-1}\|, \tag{2}$$

where the matrix norm is the Frobenius norm. At the same time the value of $\text{Cond}(\mathbf{B})$ measures the sensitivity of the solution $\mathbf{x}$ in Eq. (1) to a small variation of the given data $\mathbf{B}$ and $\mathbf{b}_0$. We have $\text{Cond}(\mathbf{B}) \geq 1$ and $\text{Cond}(\alpha\mathbf{B}) = \text{Cond}(\mathbf{B})$ for every scalar $\alpha \neq 0$. The last property shows that it is impossible to reduce $\text{Cond}(\mathbf{B})$ by multiplying all equations in Eq. (1) by a scalar. However, it is possible to diminish $\text{Cond}(\mathbf{B})$ by multiplying every row and every column of the matrix $\mathbf{B}$ by a suitable set of scaling numbers.

The scaling of linear algebraic equations is an important topic that has a long history of development. A matrix is equilibrated if all its rows or columns have the same norm, and under this condition the matrix is better conditioned. Theoretically, some optimal scalings have been proposed by Bauer (1963, 1969), van der Sluis (1969), Watson (1991), and Gautschi (2011). The problem is the search of some suitable diagonal matrices $\mathbf{Q}$ and $\mathbf{P}$, such that the condition number of $\mathbf{QBP}$ is reduced as much as possible [Vajargah (2012)]. In this paper we propose a simple procedure to find $\mathbf{P}$ and $\mathbf{Q}$ only through a few operations, which are derived explicitly.

The approaches to ill-posed linear problems can be categorized into three main classes: (a) regularizations of Eq. (1), (b) regularized algorithms to solve Eq. (1), and (c) a better pre-conditioning and/or post-conditioning to Eq. (1).

One of the matrix preconditioning techniques is based on an approximation of the inverse of the coefficient matrix. In the splitting method we assume that $\mathbf{B} = \mathbf{M} - \mathbf{N}$ and associate it with an iterative method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{M}^{-1}(\mathbf{b}_0 - \mathbf{B}\mathbf{x}_k). \tag{3}$$

Here $\mathbf{M}^{-1}$ plays the role of a preconditioner. The more $\mathbf{M}$ resembles $\mathbf{B}$, the faster the iterative method will converge. A natural and simplest way for the choice of the preconditioner is a diagonal matrix taken from the coefficient matrix, like as the Jacobi method. However, it usually does not have a remarkable reduction of the iteration number.

According to the idea of "equilibrated matrix", Liu (2012a) has developed a general purpose optimally scaled vector regularization method to treat ill-conditioned linear problems. The author and his coworkers have developed several methods fallen into the above three classes to solve the ill-posed linear problems: using the fictitious time integration method as a filter for ill-posed linear equations system [Liu and Atluri (2009a)], a modified polynomial expansion method [Liu and Atluri

(2009b)], the non-standard group-preserving scheme [Liu and Chang (2009)], a vector regularization method [Liu, Hong and Atluri (2010)], the preconditioners and postconditioners generated from a transformation matrix, obtained by Liu, Yeih and Atluri (2009) for solving the Laplace equation with a multiple-scale Trefftz basis functions, the relaxed steepest descent method [Liu (2011a)], optimal iterative algorithm [Liu and Atluri (2011)], an optimally scaled vector regularization method [Liu (2012a)], a generalized Tikhonov regularization method [Liu (2012b)], the best vector iterative algorithms [Liu (2012c, 2012d)], an adaptive Tikhonov regularization method [Liu (2013a)], an optimal tri-vector iterative algorithm [Liu (2013b)], an optimal preconditioner with an alternate relaxation parameter iterative algorithm [Liu (2013c)], as well as an optimal Krylov subspace iterative algorithm [Liu (2013d)].

This paper is a continuation of these efforts, which is organized as follows. The conjugate gradient method (CGM) and a preconditioned conjugate gradient method for solving linear equations system are reviewed in Section 2. In Section 3 we propose a simple two-side equilibration method to reduce the condition number of a given non-singular matrix only through a few sequential operations. Then, in Section 4 we describe two simple and direct two-side equilibrated conditioner methods for the solution of an ill-posed linear equations system. Section 5 is devoted to use the two-side CGM (TSCGM), and the pre-conditioning CGM (PrCGM) proposed in Section 4 together with method of fundamental solutions (MFS) and the Trefftz method to solve the Cauchy problems of Laplace equation, backward heat conduction problem, as well as an incomplete Cauchy problem of biharmonic equation. Finally, the conclusions are drawn in Section 6.

2 A preconditioned conjugate gradient method

Instead of Eq. (1), we can solve the normal equation:

$$\mathbf{C}\mathbf{x} = \mathbf{b}, \tag{4}$$

where

$$\mathbf{C} := \mathbf{B}^{\mathrm{T}}\mathbf{B}, \tag{5}$$

$$\mathbf{b} := \mathbf{B}^{\mathrm{T}}\mathbf{b}_0. \tag{6}$$

The conjugate gradient method (CGM), which is used to solve Eq. (4), is summarized as follows:
(i) Give an initial $\mathbf{x}_0$.
(ii) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{C}\mathbf{x}_0$ and $\mathbf{p}_1 = \mathbf{r}_0$.

(iii) For $k = 1, 2, \ldots$, we repeat the following iterations:

$$\begin{aligned}
\alpha_k &= \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}}\mathbf{C}\mathbf{p}_k}, \\
\mathbf{x}_k &= \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k, \\
\mathbf{r}_k &= \mathbf{b} - \mathbf{C}\mathbf{x}_k, \\
\beta_k &= \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \\
\mathbf{p}_{k+1} &= \mathbf{r}_k + \beta_k \mathbf{p}_k.
\end{aligned} \tag{7}$$

If $\mathbf{x}_k$ converges according to a given stopping criterion, such that,

$$\|\mathbf{r}_k\| < \varepsilon, \tag{8}$$

then stop; otherwise, go to step (iii).

It is well known that the convergence speed of CGM depends on the distribution of the eigenvalues of the coefficient matrix $\mathbf{C}$. When the coefficient matrix is typically extremely ill-conditioned, the convergence of CGM can be unacceptably slow. In this case, the CGM is not competitive without using a good preconditioner. That is, the preconditioning technique is a key ingredient for the success of CGM in applications. The idea of preconditioning technique is based on the consideration of the linear system with the same solution as the original equation. The problem is that each preconditioning technique is suited for a different type of problem. Until now no a robust preconditioning technique appears for all or at least much types of problems. Finding a good preconditioner to solve a given large scale linear system is often viewed as a combination of art and science.

To improve the convergence speed of iterative methods, an appropriate preconditioner can be incorporated. Based on the survey by Benzi (2002), a good preconditioner should satisfy the following requirements: (1) the preconditioned system should be easy to solve, and (2) the preconditioner should be cheap to construct and apply. In order to increase the convergence speed of CGM, we require to reduce the condition number of $\mathbf{C}$. For the purpose of comparison the preconditioned CGM with a preconditioning matrix $\mathbf{M}$ is written as follows:

(i) Give an initial $\mathbf{x}_0$.

(ii) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{C}\mathbf{x}_0$, $\mathbf{z}_0 = \mathbf{M}\mathbf{r}_0$ and $\mathbf{p}_1 = \mathbf{z}_0$.

(iii) For $k = 1, 2, \ldots$, we repeat the following iterations:

$$
\begin{aligned}
\alpha_k &= \frac{\mathbf{r}_{k-1}\cdot\mathbf{z}_{k-1}}{\mathbf{p}_k^{\mathrm{T}}\mathbf{C}\mathbf{p}_k},\\
\mathbf{x}_k &= \mathbf{x}_{k-1}+\alpha_k\mathbf{p}_k,\\
\mathbf{r}_k &= \mathbf{b}-\mathbf{C}\mathbf{x}_k,\\
\mathbf{z}_k &= \mathbf{M}\mathbf{r}_k,\\
\beta_k &= \frac{\mathbf{r}_k\cdot\mathbf{z}_k}{\mathbf{r}_{k-1}\cdot\mathbf{z}_{k-1}},\\
\mathbf{p}_{k+1} &= \mathbf{z}_k+\beta_k\mathbf{p}_k.
\end{aligned}
\tag{9}
$$

If $\mathbf{x}_k$ converges according to a given stopping criterion, such that, $\|\mathbf{r}_k\| < \varepsilon$, then stop; otherwise, go to step (iii).

3 Two sequential operations to reduce the condition number

Here we rewrite Eq. (1) as

$$\mathbf{QBPy} = \mathbf{Qb}_0, \tag{10}$$

where

$$\mathbf{x} = \mathbf{Py}. \tag{11}$$

It follows a new system for $\mathbf{y}$:

$$\mathbf{Ay} = \mathbf{b} := \mathbf{Qb}_0, \tag{12}$$

where

$$\mathbf{A} = \mathbf{QBP}. \tag{13}$$

Our purpose is first to find a two-side diagonal conditioiners $\mathbf{P}$ and $\mathbf{Q}$, such that the condition number of the new coefficient matrix $\mathbf{A}$ can be reduced much than that of $\mathbf{B}$.

Let us begin with the following diagonal post-conditioning matrix:

$$
\mathbf{P} = \begin{bmatrix}
1 & 0 & 0 & \dots & 0\\
0 & P_2 & 0 & \dots & 0\\
\vdots & \vdots & \vdots & \vdots & \vdots\\
0 & \dots & 0 & P_{n-1} & 0\\
0 & \dots & 0 & 0 & P_n
\end{bmatrix}. \tag{14}
$$

The problem is to search for some suitable values of P_k in Eq. (14), such that the condition number of

$$\mathbf{A} = \mathbf{BP} \tag{15}$$

is reduced as much as possible. Due to the property of $\mathrm{Cond}(\alpha\mathbf{B}) = \mathrm{Cond}(\mathbf{B})$ for every scalar $\alpha \neq 0$, we have normalized the first diagonal element of $\mathbf{P}$ to one without lost any generality.

A matrix is equilibrated if all its rows or columns have the same norm, and under this condition the matrix is better conditioned. According to the idea of equilibrated matrix [Liu (2012e)], we can choose P_k, such that each column of the coefficient matrix $\mathbf{A}$ has the same Euclidean norm, i.e.,

$$\sum_{j=1}^{n} A_{j1}^2 = \sum_{j=1}^{n} A_{j2}^2 = \ldots = \sum_{j=1}^{n} A_{jn}^2, \tag{16}$$

where A_{ij} denotes the ij-th component of $\mathbf{A}$. The square norm of the first column of $\mathbf{A}$ is

$$\sum_{j=1}^{n} A_{j1}^2 = \sum_{j=1}^{n} B_{j1}^2, \tag{17}$$

which is a fixed positive value for the matrix $\mathbf{B}$ being given. Hence, from Eqs. (16) and (17) we can solve P_k by

$$P_k = \left(\frac{\sum_{j=1}^{n} B_{j1}^2}{\sum_{j=1}^{n} B_{jk}^2} \right)^{\frac{1}{2}}, \quad k = 2, \ldots, n. \tag{18}$$

Now we investigate the effect by using the above scaling technique to reduce the condition number of the following matrix:

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 3 & 6 \end{bmatrix}. \tag{19}$$

We obtain the value of $\mathrm{Cond}(\mathbf{B}) = 63$. When we apply the above scaling technique we can obtain $\mathrm{Cond}(\mathbf{BP}) = 50.9411$. However, we can apply the following scales to further reduce the condition number:

$$P_k = \gamma \left(\frac{\sum_{j=1}^{n} B_{j1}^2}{\sum_{j=1}^{n} B_{jk}^2} \right)^{\frac{1}{2}}, \quad k = 2, \ldots, n, \tag{20}$$

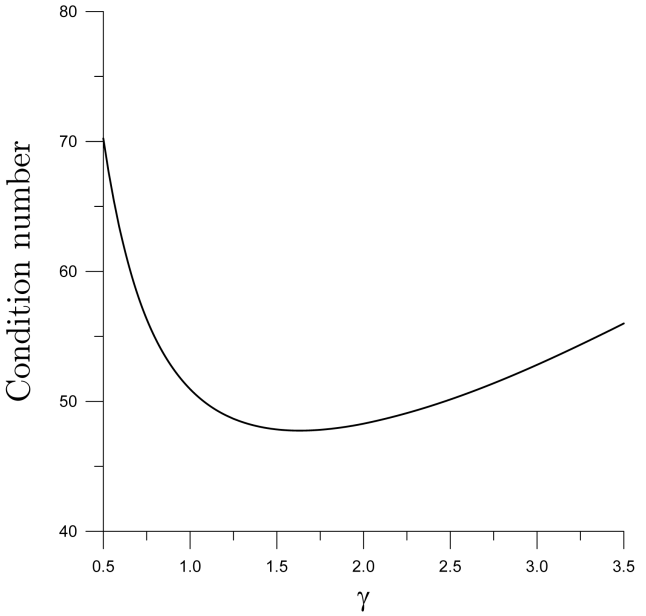

Figure 1: For a given matrix, plotting the condition number with respect to the amplification factor.

where γ is an amplification factor. In Fig. 1 we plot the condition number vs. γ in a range of $\gamma \in [0.5, 3.5]$. We can observe that when $\gamma = 1.625$ the condition number reduces to the smallest one with 47.7495.

Similarly, we can consider a left-conditioner $\mathbf{Q}$ to be

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & Q_2 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & Q_{n-1} & 0 \\ 0 & \dots & 0 & 0 & Q_n \end{bmatrix}, \tag{21}$$

and apply it to a given matrix $\mathbf{B}$ by the following scales:

$$Q_k = \gamma \left(\frac{\sum_{j=1}^{n} B_{1j}^2}{\sum_{j=1}^{n} B_{kj}^2} \right)^{\frac{1}{2}}, \quad k = 2, \dots, n. \tag{22}$$

When $\gamma = 1$, the resultant matrix $\mathbf{QB}$ has the same norm of each row.

Let us denote the operations in Eqs. (20) and (22), respectively, by

$$\mathbf{P} = \mathscr{P}(\mathbf{B}), \ \ \mathbf{Q} = \mathscr{Q}(\mathbf{B}). \tag{23}$$

Now we can construct a sequence of right-conditioners $\mathbf{P}_k, \ k = 1, \ldots, M$ and left-conditioners $\mathbf{Q}_k, \ k = 1, \ldots, M$ by

$$\begin{aligned}
&\mathbf{P}_1 = \mathscr{P}(\mathbf{B}), \ \ \mathbf{Q}_1 = \mathscr{Q}(\mathbf{B}_1), \\
&\mathbf{P}_2 = \mathscr{P}(\mathbf{B}_2), \ \ \mathbf{Q}_2 = \mathscr{Q}(\mathbf{B}_3), \\
&\vdots \\
&\mathbf{P}_M = \mathscr{P}(\mathbf{B}_{2M-2}), \ \ \mathbf{Q}_M = \mathscr{Q}(\mathbf{B}_{2M-1}),
\end{aligned} \tag{24}$$

where

$$\begin{aligned}
&\mathbf{B}_1 = \mathbf{B}\mathbf{P}_1, \ \ \mathbf{B}_2 = \mathbf{Q}_1\mathbf{B}_1, \\
&\mathbf{B}_3 = \mathbf{B}_2\mathbf{P}_2, \ \ \mathbf{B}_4 = \mathbf{Q}_2\mathbf{B}_3, \\
&\vdots \\
&\mathbf{B}_{2M-1} = \mathbf{B}_{2M-2}\mathbf{P}_M, \ \ \mathbf{B}_{2M} = \mathbf{Q}_M\mathbf{B}_{2M-1}.
\end{aligned} \tag{25}$$

Then, the final right-conditioner $\mathbf{P}$ and left-conditioner $\mathbf{Q}$ can be obtained by

$$\mathbf{P} = \mathbf{P}_1\mathbf{P}_2 \ldots \mathbf{P}_M, \ \ \mathbf{Q} = \mathbf{Q}_M\mathbf{Q}_{M-1} \ldots \mathbf{Q}_1. \tag{26}$$

The above process to obtain $\mathbf{P}$ and $\mathbf{Q}$ is first applying $\mathbf{P}_i$ to the resultant matrix and then applying $\mathbf{Q}_i$ to the resultant matrix, which is called the PQ process. We can also reverse the order by first applying $\mathbf{Q}_i$ to the resultant matrix and then applying $\mathbf{P}_i$ to the resultant matrix, which is called the QP process to distinct from the above PQ process. For a given symmetric matrix the PQ process and the QP process lead to the same result. However, for a given unsymmetric matrix the PQ process and the QP process may lead to the different results.

Besides the symmetric matrix given in Eq. (19) we consider another symmetric matrix:

$$B_{ij} = \frac{1}{i+j-1}, \ i, j = 1, \ldots, n, \tag{27}$$

which is the Hilbert matrix, where we fix $n = 5$. Here we take $M = 5$.

In Fig. 2 we plot the condition numbers of these matrices with respect to $k = 0, 1, 2, \ldots, 2M$. When $k = 0$, the condition number is that of the original matrix. For these two matrices we use $\gamma = 1.5$ and $\gamma = 1, 2$, respectively. It can be seen that

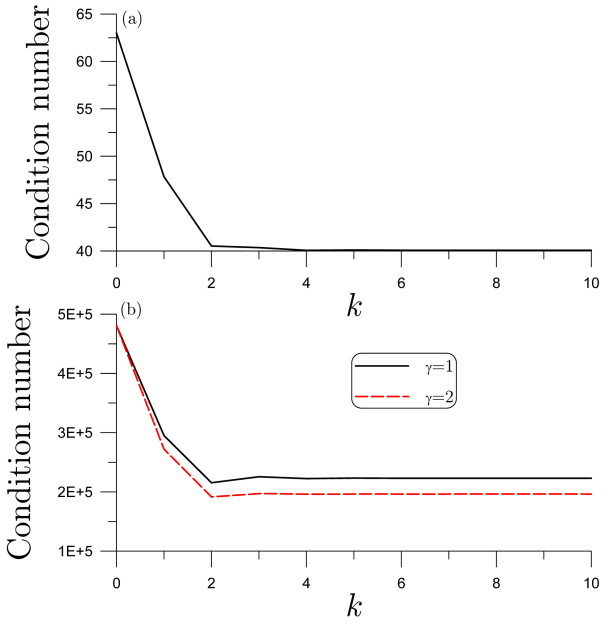

Figure 2: For two given symmetric matrices, plotting the condition number with respect to the number of operations.

the condition numbers for the first matrix given by Eq. (19) can be reduced quickly after a few operations with the number being $k = 2$ as shown in Fig. 2(a). For the Hilbert matrix, the PQ process leads to a fast reduction of the condition number by two operations, and the value of $\gamma = 2$ leads to a smaller condition number than that using $\gamma = 1$ as shown in Fig. 2(b).

Then we consider an unsymmetric matrix:

$$\mathbf{B} = \begin{bmatrix} 0.0926612 & 17.0784926 & 0.3127063 & 12.7526810 \\ 1.7811361 & 54.0213314 & 1.4953060 & 14.7655003 \\ 0.3460217 & 0.0680433 & 0.2626770 & 0.0227214 \\ 1.3745248 & 45.1500312 & 0.0505958 & 1.4314422 \end{bmatrix}. \tag{28}$$

Vajargah and Moradi (2012) have applied a genetic algorithm to solve a minimum problem of the condition number under the assumption of $\mathbf{D}_1$ and $\mathbf{D}_2$ being diagonal:

$$\min\{\mathrm{Cond}(\mathbf{D}_1\mathbf{B}\mathbf{D}_2)\},$$

and they found

$$\mathbf{D}_1 = \begin{bmatrix} -1022.446636286579 & 0 & 0 & 0 \\ 0 & 495.973718177785 & 0 & 0 \\ 0 & 0 & 2743.200397567089 & 0 \\ 0 & 0 & 0 & 545.478809477854 \end{bmatrix},$$

$$\mathbf{D}_2 = \begin{bmatrix} 2335.779656989865 & 0 & 0 & 0 \\ 0 & 74.778044753772 & 0 & 0 \\ 0 & 0 & -2923.322157077509 & 0 \\ 0 & 0 & 0 & -209.320088460085 \end{bmatrix}.$$

By using the above data, we apply the matrix inversion method developed by Liu, Hong and Atluri (2000) to find the inversion of $(\mathbf{D}_1\mathbf{B}\mathbf{D}_2)$ and check the accuracy of the invesrsion which is in the order of 10^{-15}. Then we calculate the minimum condition number to be 17.414806133, which is larger than that 14.4854765 given by Vajargah and Moradi (2012).

Under $\gamma = 0.9$, the condition number $\text{Cond}(\mathbf{B}) = 474.8583$ for the matrix given by Eq. (28) can be reduced by the PQ process quickly after a few operations with the number of operations being $k = 2$ to $\text{Cond}(\mathbf{B}_2) = 19.3652$, and the PQ process is faster than the QP process; however, after ten operations they lead to the same result as shown in Fig. 3(a). The condition number is about 18.9837, which is close to the minimum value 17.414806133 as reported in the above, not 14.4854765 as obtained by Vajargah and Moradi (2012). However, the present method is simpler than the genetic algorithm used by Vajargah and Moradi (2012). For the purpose of comparison, $\mathbf{Q}$ and $\mathbf{P}$ are written as follows:

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.339130167084 & 0 & 0 \\ 0 & 0 & 2.356851662291 & 0 \\ 0 & 0 & 0 & 0.584688010010 \end{bmatrix}, \tag{29}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.032205073516 & 0 & 0 \\ 0 & 0 & 1.361403106503 & 0 \\ 0 & 0 & 0 & 0.086106431746 \end{bmatrix}. \tag{30}$$

This example shows that the proposed sequential PQ and QP methods to find $\mathbf{Q}$ and $\mathbf{P}$ are effective and simple.

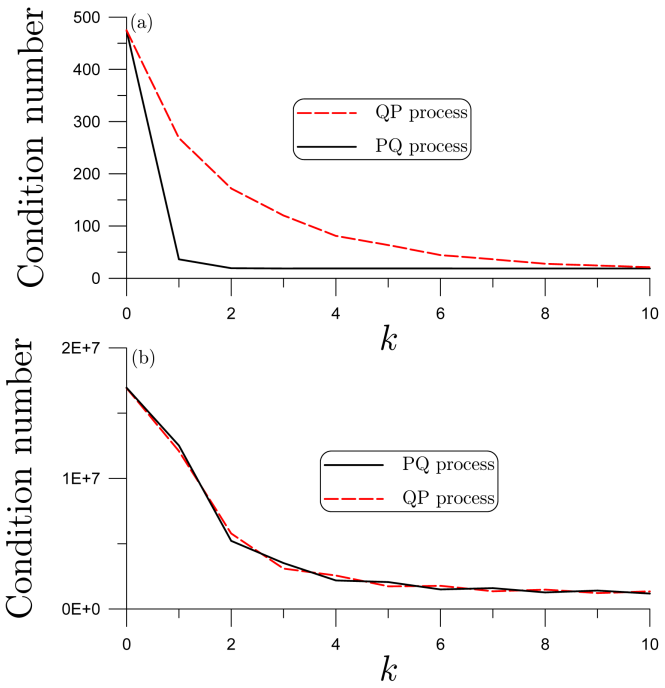

Figure 3: For two given unsymmetric matrices, plotting the condition number with respect to the number of operations and comparing the PQ process and the QP process.

The last case is an unsymmetric Vandermonde matrix:

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \\ x_1 & x_2 & \dots & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \dots & x_{n-1}^2 & x_n^2 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ x_1^{n-2} & x_2^{n-2} & \dots & x_{n-1}^{n-2} & x_n^{n-2} \\ x_1^{n-1} & x_2^{n-1} & \dots & x_{n-1}^{n-1} & x_n^{n-1} \end{bmatrix}, \tag{31}$$

where the nodes are generated from $x_i = (i-1)/(n-1)$, which are equidistant nodes in the unit interval $x_i \in [0,1]$. For the Vandermonde matrix given by Eq. (31) we fix $n = 10$ and $\gamma = 1$, and the PQ process and the QP process lead to the similar results as shown in Fig. 3(b). It is remarkable that the condition number can be reduced from the original 1.69285×10^7 to 1.34912×10^6 by using the proposed PQ and QP processes. Below we only use the PQ process to reduce the condition number.

4 A two-side equilibrated conditioning method

4.1 A two-side CGM

Now we can apply the following CGM to solve $\mathbf{y}$ and then $\mathbf{x} = \mathbf{P}\mathbf{y}$:
(i) Give M and γ and calculate $\mathbf{P}$ and $\mathbf{Q}$ by the PQ process.
(ii) Calculate $\mathbf{A} = \mathbf{QBP}$, $\mathbf{b} = \mathbf{A}^{\mathrm{T}}\mathbf{Q}\mathbf{b}_0$ and $\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$.
(iii) Give an initial $\mathbf{y}_0$.
(iv) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{C}\mathbf{y}_0$ and $\mathbf{p}_1 = \mathbf{r}_0$.
(v) For $k = 1, 2, \ldots$, we repeat the following iterations:

$$\begin{aligned}
\alpha_k &= \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}}\mathbf{C}\mathbf{p}_k}, \\
\mathbf{y}_k &= \mathbf{y}_{k-1} + \alpha_k \mathbf{p}_k, \\
\mathbf{r}_k &= \mathbf{b} - \mathbf{C}\mathbf{y}_k, \\
\beta_k &= \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \\
\mathbf{p}_{k+1} &= \mathbf{r}_k + \beta_k \mathbf{p}_k.
\end{aligned} \tag{32}$$

If $\mathbf{y}_k$ converges according to a given stopping criterion $\|\mathbf{r}_k\| < \varepsilon$, then stop; otherwise, go to step (v). The above iterative algorithm will be abbreviated as a two-side CGM (TSCGM).

4.2 A pre-conditioning CGM

We can also apply the following pre-conditioning CGM (PrCGM) to solve $\mathbf{y}$ and then $\mathbf{x} = \mathbf{P}\mathbf{y}$:
(i) Give M and γ and calculate $\mathbf{P}$ and $\mathbf{Q}$ by the PQ process.
(ii) Calculate $\mathbf{A} = \mathbf{BP}$, $\mathbf{b} = \mathbf{A}^{\mathrm{T}}\mathbf{b}_0$ and $\mathbf{C} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$.
(iii) Give an initial $\mathbf{y}_0$.
(iv) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{C}\mathbf{y}_0$, $\mathbf{z}_0 = \mathbf{Q}\mathbf{r}_0$ and $\mathbf{p}_1 = \mathbf{z}_0$.
(v) For $k = 1, 2, \ldots$, we repeat the following iterations:

$$\begin{aligned}
\alpha_k &= \frac{\mathbf{r}_{k-1} \cdot \mathbf{z}_{k-1}}{\mathbf{p}_k^{\mathrm{T}}\mathbf{C}\mathbf{p}_k}, \\
\mathbf{y}_k &= \mathbf{y}_{k-1} + \alpha_k \mathbf{p}_k, \\
\mathbf{r}_k &= \mathbf{b} - \mathbf{C}\mathbf{y}_k, \\
\mathbf{z}_k &= \mathbf{Q}\mathbf{r}_k, \\
\beta_k &= \frac{\mathbf{r}_k \cdot \mathbf{z}_k}{\mathbf{r}_{k-1} \cdot \mathbf{z}_{k-1}}, \\
\mathbf{p}_{k+1} &= \mathbf{z}_k + \beta_k \mathbf{p}_k.
\end{aligned} \tag{33}$$

If $\mathbf{x}_k$ converges according to a given stopping criterion, such that, $\|\mathbf{r}_k\| < \varepsilon$, then stop; otherwise, go to step (v).

5 Numerical examples

5.1 Example 1

Let us consider the inverse Cauchy problem for the Laplace equation:

$$\Delta u = u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 0, \tag{34}$$

$$u(\rho,\theta) = h(\theta), \;\; 0 \le \theta \le \beta\pi, \tag{35}$$

$$u_n(\rho,\theta) = g(\theta), \;\; 0 \le \theta \le \beta\pi, \tag{36}$$

where $h(\theta)$ and $g(\theta)$ are given functions and $\beta \le 1$. The inverse Cauchy problem is given as follows: To seek an unknown boundary function $f(\theta)$ on the part $\Gamma_2 := \{(r,\theta)|r = \rho(\theta), \;\beta\pi < \theta < 2\pi\}$ of the boundary under Eqs. (34)-(36) with the overspecified data on $\Gamma_1 := \{(r,\theta)|r = \rho(\theta), \; 0 \le \theta \le \beta\pi\}$.

It is well known that the method of fundamental solutions (MFS) can be used to solve the Laplace equation when a fundamental solution is known [Kupradze and Aleksidze (1964)]. In the MFS the solution of u at the field point $\mathbf{z} = (r\cos\theta, r\sin\theta)$ can be expressed as a linear combination of fundamental solutions $U(\mathbf{z},\mathbf{s}_j)$:

$$u(\mathbf{z}) = \sum_{j=1}^{n} c_j U(\mathbf{z},\mathbf{s}_j), \;\; \mathbf{s}_j \in \Omega^c. \tag{37}$$

For the Laplace equation (34) we have the fundamental solutions:

$$U(\mathbf{z},\mathbf{s}_j) = \ln r_j, \;\; r_j = \|\mathbf{z} - \mathbf{s}_j\|. \tag{38}$$

Previously, Liu (2008a) has proposed a new preconditioner to reduce the ill-condition of the MFS. In the practical application of MFS, by imposing the boundary conditions (35) and (36) on Eq. (37) we can obtain a linear equations system:

$$\mathbf{B}\mathbf{x} = \mathbf{b}_0, \tag{39}$$

where

$$\begin{aligned}
&\mathbf{z}_i = (z_i^1, z_i^2) = (\rho(\theta_i)\cos\theta_i, \rho(\theta_i)\sin\theta_i),\\
&\mathbf{s}_j = (s_j^1, s_j^2) = (R(\theta_j)\cos\theta_j, R(\theta_j)\sin\theta_j),\\
&B_{ij} = \ln\|\mathbf{z}_i - \mathbf{s}_j\|, \ \text{if } i \text{ is odd},\\
&B_{ij} = \frac{\eta(\theta_i)}{\|\mathbf{z}_i - \mathbf{s}_j\|^2}\left(\rho(\theta_i) - s_j^1\cos\theta_i - s_j^2\sin\theta_i\right.\\
&\quad \left. - \frac{\rho'(\theta_i)}{\rho(\theta_i)}[s_j^1\sin\theta_i - s_j^2\cos\theta_i]\right), \text{if } i \text{ is even},\\
&\mathbf{x} = (c_1, \ldots, c_n)^{\mathrm{T}}, \ \mathbf{b}_0 = (h(\theta_1), g(\theta_1), \ldots, h(\theta_m), g(\theta_m))^{\mathrm{T}},
\end{aligned} \tag{40}$$

in which $n = 2m$, and

$$\eta(\theta) = \frac{\rho(\theta)}{\sqrt{\rho^2(\theta) + [\rho'(\theta)]^2}}. \tag{41}$$

The above $R(\theta) = \rho(\theta) + D$ with an offset D can be used to locate the source points along a contour with the radius $R(\theta)$. When the linear equations system (39) is available, we can apply the TSCGM and PrCGM to solve it.

For the purpose of comparison we consider the following exact solution:

$$u(r,\theta) = r^2\cos(2\theta),$$

defined in a domain with the boundary $\rho(\theta) = \sqrt{10 - 6\cos(2\theta)}$, $0 \le \theta < 2\pi$.

We add a random noise with an intensity $s = 1\%$ on the boundary data, and the numerical solutions on the whole boundary are computed by the CGM, TSCGM and PrCGM with $D = 100$. Here we use the relative residual $\|\mathbf{y}_{k+1} - \mathbf{y}_k\| < \varepsilon$ as a convergence criterion. When the CGM under $\varepsilon = 10^{-10}$ is convergent with 51 iterations, the TSCGM with $\gamma = 0.9$, $M = 1$ and $n = 80$ is convergent with 9 iterations under $\varepsilon = 10^{-2}$, and the PrCGM with $\gamma = 2$, $M = 5$ and $n = 80$ is convergent with 6 iterations under $\varepsilon = 10^{-5}$. It can be seen that the conditioning effects on TSCGM and PrCGM make them convergence faster than the original CGM without conditioning.

We compare the numerical solutions with the exact one in Fig. 4(a), and the numerical errors are shown in Fig. 4(b). When the CGM is with the maximum error being 0.0336, the TSCGM is with the maximum error being 9.037×10^{-3}, and the PrCGM is with the maximum error being 2.847×10^{-3}. The TSCGM and PrCGM are much accurate than the CGM.

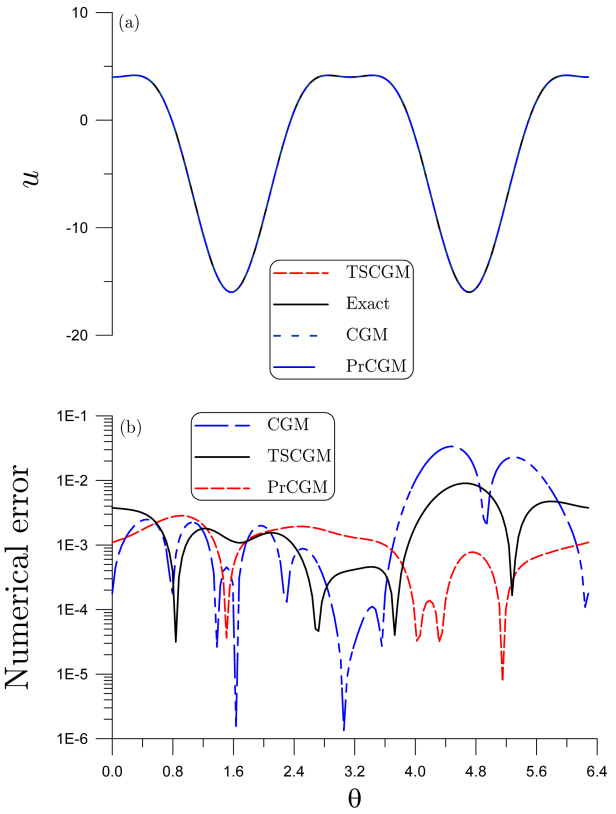

Figure 4: For an inverse Cauchy problem: (a) comparing the numerical solutions of CGM, PrCGM and TSCGM with the exact one, and (b) displaying the numerical errors.

5.2 Example 2

For this example we apply the following Trefftz method [Liu (2007a, 2007b)] to solve the inverse Cauchy problem of Laplace equation:

$$u(r,\theta) = a_0 + \sum_{k=1}^{m} \left[a_k \left(\frac{r}{R_0} \right)^k \cos k\theta + b_k \left(\frac{r}{R_0} \right)^k \sin k\theta \right], \tag{42}$$

where

$$R_0 \geq \rho_{\max} = \max_{\theta \in [0,2\pi]} \rho(\theta) \tag{43}$$

is a constant which is greater than the maximum length of the problem domain Ω. Besides, m is a positive integer chosen by the user, and $a_0, a_k, b_k,\ k = 1, \ldots, m$ are unknown coefficients to be determined. Liu and Atluri (2013) have proposed a modification of the above expansion method by using a set of optimal multiple-scales R_k, which are obtained by using the concept of equilibrated matrix.

We consider an example with the exact solution:

$$u = e^x \cos y = e^{r\cos\theta} \cos(r\sin\theta), \tag{44}$$

where the contour is described by an epitrochoid boundary shape:

$$\rho(\theta) = \sqrt{10 - 6\cos(2\theta)}. \tag{45}$$

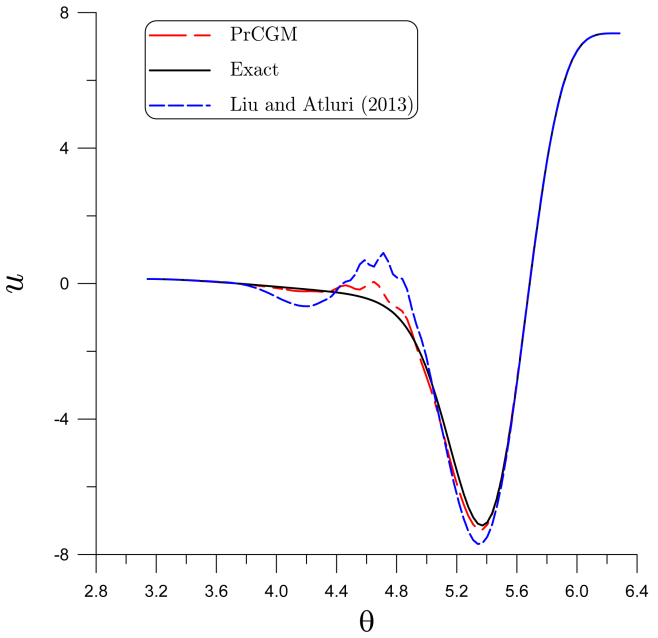

Figure 5: For an inverse Cauchy problem, comparing the numerical solutions of PrCGM and that of Liu and Atluri (2013) with the exact one.

Under the following parameters $R_0 = 5$, $m = 30$, $\beta = 1$, $\gamma = 0.9$, $M = 2$ and $s = 0.001$ we apply the method of PrCGM to solve this Cauchy problem, of which the result is compared with the exact solution in Fig. 5. Although under a stringent

Table 1: The comparisons of Liu and Atluri (2013) and the PrCGM used to solve Example 2

$\theta=\pi+k\Delta\theta$	Liu and Atluri (2013)	PrCGM
$k=20$	0.013317	0.005314
$k=30$	0.441388	0.067176
$k=40$	0.049667	0.114839
$k=50$	1.557880	0.364920
$k=60$	0.096412	0.259237
$k=70$	0.601039	0.164740
$k=80$	0.011124	0.043562

convergence criterion with $\varepsilon=10^{-9}$, the PrCGM is convergent with 287 iterations. The result obtained by the PrCGM with the maximum error being 0.566 is more accurate than that obtained by Liu and Atluri (2013), of which the comparisons at some points are made in Table 1, where $\Delta\theta=\pi/100$.

5.3 Example 3

In this example we consider an inverse Cauchy problem of the following biharmonic equation:

$$\Delta^2 u=0,\ \ (x,y)\in\Omega, \tag{46}$$

where Ω is an interior domain in the plane. This inverse Cauchy problem of biharmonic equation is under an incomplete set of data given by

$$u(\rho,\theta)=h(\theta),\ \ u_n(\rho,\theta)=g(\theta),\ \ 0\le\theta\le 2\beta\pi. \tag{47}$$

When $\beta=1$ we recover to the direct problem. Here we let $\beta<1$ and do not use the overspecified data, such that the present problem is a Cauchy problem with an incomplete set of given data.

For the purpose of comparison we suppose that the exact solution is

$$u(x,y)=x^3+y^3,$$

and the domain is defined by

$$\rho(\theta)=\sqrt{26-10\cos(4\theta)}. \tag{48}$$

For this example we apply the following Trefftz method [Liu (2008b)] to solve the inverse Cauchy problem of biharmonic equation:

$$\begin{aligned} u(r,\theta) &= a_0 + \sum_{k=1}^{m} \left[a_k \left(\frac{r}{R_0} \right)^k \cos k\theta + b_k \left(\frac{r}{R_0} \right)^k \sin k\theta \right] \\ &+ c_0 r^2 + \sum_{k=1}^{m} \left[c_k \left(\frac{r}{R_0} \right)^{k+2} \cos k\theta + d_k \left(\frac{r}{R_0} \right)^{k+2} \sin k\theta \right]. \end{aligned} \tag{49}$$

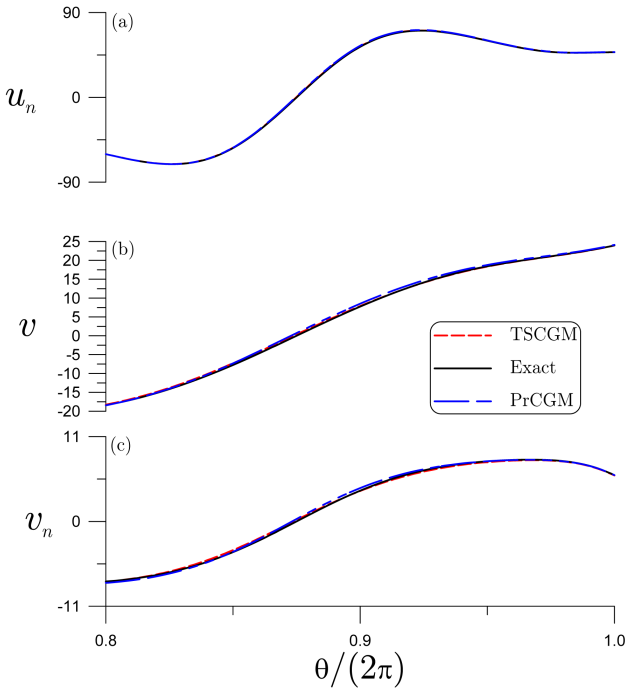

Figure 6: For inverse Cauchy problem of biharmonic equation, comparing the numerical solutions of PrCGM and TSCGM with the exact ones.

We fix the noise to be $s = 0.01$ and $\beta = 0.8$. Under the following parameters $R_0 = 10$, $m = 5$, $\gamma = 1$ and $M = 5$, we apply the TSCGM to solve this Cauchy problem, which is convergent with 30 iterations under $\varepsilon = 10^{-3}$. Under the following parameters $R_0 = 10$, $m = 5$, $\gamma = 1$ and $M = 5$, we apply the PrCGM to solve this Cauchy problem, which is convergent with 38 iterations under $\varepsilon = 10^{-4}$. The numerical results are compared with the exact solutions in Fig. 6, where $v = \Delta u$.

The results obtained by the TSCGM and PrCGM are very accurate. The maximum errors of u_n, v and v_n for the TSCGM are, respectively, 0.56, 0.46 and 0.28, while that for the PrCGM are, respectively, 1.19, 0.78 and 0.36.

5.4 Example 4

When the backward heat conduction problem (BHCP) is considered in a spatial interval of $0 < x < \ell$ by subjecting to the boundary conditions at two ends of a slab:

$$u_t(x,t) = \alpha u_{xx}(x,t), \;\; 0 < t < T, \;\; 0 < x < \ell, \tag{50}$$

$$u(0,t) = u_0(t), \;\; u(\ell,t) = u_\ell(t), \tag{51}$$

we solve u under a final time condition:

$$u(x,T) = u^T(x). \tag{52}$$

The fundamental solution of Eq. (50) is given as follows:

$$K(x,t) = \frac{H(t)}{2\sqrt{\alpha\pi t}} \exp\left(\frac{-x^2}{4\alpha t}\right), \tag{53}$$

where $H(t)$ is the Heaviside function.

The method of fundamental solutions (MFS) has a broad application in engineering computations. However, the MFS has a serious drawback in that the resulting linear equations system is always highly ill-conditioned, when the number of source points is increased [Golberg and Chen (1996)], or when the distances of source points are increased [Chen, Cho and Golberg (2006)].

In the MFS the solution of u at the field point $\mathbf{z} = (x,t)$ can be expressed as a linear combination of the fundamental solutions $U(\mathbf{z},\mathbf{s}_j)$:

$$u(\mathbf{z}) = \sum_{j=1}^{n} c_j U(\mathbf{z},\mathbf{s}_j), \;\; \mathbf{s}_j = (\eta_j,\tau_j) \in \Omega^c, \tag{54}$$

where n is the number of source points, c_j are unknown coefficients, and $\mathbf{s}_j$ are source points being located in the complement Ω^c of $\Omega = [0,\ell] \times [0,T]$. For the heat conduction equation we have the basis functions

$$U(\mathbf{z},\mathbf{s}_j) = K(x-\eta_j, t-\tau_j). \tag{55}$$

It is known that the location of source points in the MFS has a great influence on the accuracy and stability. In a practical application of MFS to solve the BHCP, the source points are uniformly located on two vertical straight lines parallel to

the t-axis not over the final time, which was adopted by Hon and Li (2009) and Liu (2011b), showing a large improvement than the line location of source points below the initial time. After imposing the boundary conditions and the final time condition to Eq. (54) we can obtain a linear equations system:

$$\mathbf{Bx} = \mathbf{b}_0, \tag{56}$$

where

$$\begin{aligned} &B_{ij} = U(\mathbf{z}_i, \mathbf{s}_j), \ \ \mathbf{x} = (c_1, \cdots, c_n)^{\mathrm{T}}, \\ &\mathbf{b}_0 = (u_\ell(t_i),\ i = 1, \ldots, m_1; u^T(x_j),\ j = 1, \ldots, m_2; u_0(t_k),\ k = m_1, \ldots, 1)^{\mathrm{T}}, \end{aligned} \tag{57}$$

and $n = 2m_1 + m_2$.

Here we compare the numerical solution with an exact solution:

$$u(x,t) = \cos(\pi x)\exp(-\pi^2 t).$$

For the case with $T = 1$ the value of final time data is in the order of 10^{-4}, which is quite small in a comparison with the value of the initial temperature $u_0(x) = \cos(\pi x)$ to be retrieved, which is in the order of $O(1)$. We add a relative random noise with an intensity $s = 10\%$ in the final time data. The convergence criterion is fixed to be $\varepsilon = 10^{-8}$ and $n = 2m_1 + m_2 = 40$. We first solve this problem by using the TSCGM with $M = 1$ and $\gamma = 1$, which is convergent with 65 iterations and the maximum error of initial condition obtained is 3.816×10^{-3}, of which the residual error and the error of solution are shown in Fig. 7. The residual error and the error of solution obtained by the PrCGM with $M = 2$ and $\gamma = 1$ are shown in Fig. 7, which is convergent with 65 iterations and the maximum error is 3.758×10^{-3}. The present results are much better than that calculated by Liu (2011a) which using the relaxed steepest descent method.

5.5 Linear Hilbert problem

Finding an n-order polynomial function $p(x) = a_0 + a_1 x + \ldots + a_n x^n$ to best match a continuous function $f(x)$ in the interval of $x \in [0,1]$:

$$\min_{\deg(p) \le n} \int_0^1 [f(x) - p(x)]^2 dx, \tag{58}$$

leads to a problem governed by Eq. (1), where $\mathbf{B}$ is the $(n+1) \times (n+1)$ Hilbert matrix defined by

$$B_{ij} = \frac{1}{i + j - 1}, \tag{59}$$

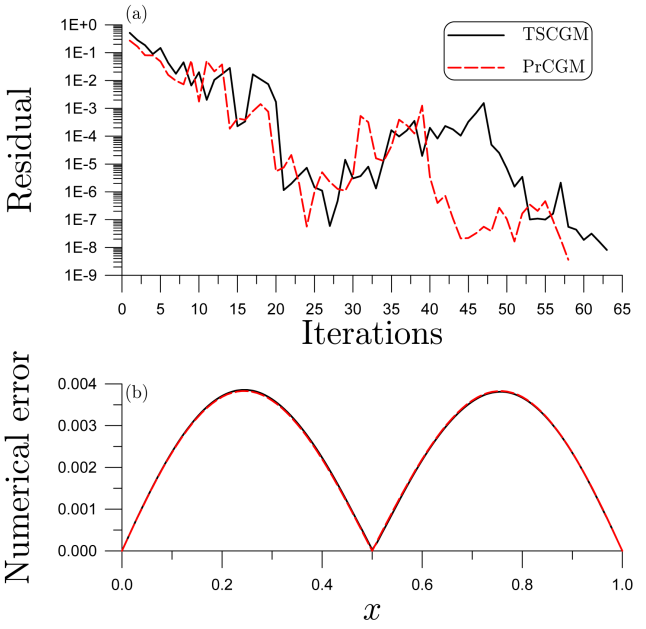

Figure 7: For BHCP solved by the TSCGM and PrCGM, showing (a) the residual errors, and (b) the numerical errors.

x is composed of the $n+1$ coefficients $a_0, a_1, \ldots, a_n$ appeared in $p(x)$, and

$$\mathbf{b}_0 = \begin{bmatrix} \int_0^1 f(x)dx \\ \int_0^1 xf(x)dx \\ \vdots \\ \int_0^1 x^n f(x)dx \end{bmatrix} \tag{60}$$

is uniquely determined by the function $f(x)$.

The Hilbert matrix is a notorious example of highly ill-conditioned matrices. Eq. (1) with the matrix **B** having a large condition number usually displays that an arbitrary small perturbation of data on the right-hand side may lead to an arbitrary large perturbation to the solution on the left-hand side.

In this example we consider a highly ill-conditioned linear system (1) with **B** given by Eq. (59). The ill-posedness of Eq. (1) increases fast with n. We consider an

exact solution with $x_j = 1$ and b_i is given by

$$b_0^i = \sum_{j=1}^{n} \frac{1}{i+j-1} + sR(i), \tag{61}$$

with $n = 300$ and $R(i)$ are random numbers between $[-1, 1]$.

When the noise is in the level of $s = 10^{-4}$, we let CGM run under the convergence criterion $\varepsilon = 10^{-9}$. The CGM converges very fast with 55 iterations; however, the maximum error of CGM is large up to 1.5.

For this highly ill-posed noised problem the TSCGM with $\gamma = 1$ and $M = 3$ is applicable and convergent with 10 iterations, and the maximum error is 8.67×10^{-3} as shown in Fig. 8. Then the PrCGM with $\gamma = 0.99$ and $M = 2$ is convergent with 15 iterations under $\varepsilon = 10^{-8}$, and the maximum error is 6.893×10^{-3} as shown in Fig. 8. These results are much better than that obtained by Liu (2012d). To our best knowledge, in the open literature there exists no method which can perform better than the TSCGM and PrCGM for this problem with $n = 300$.

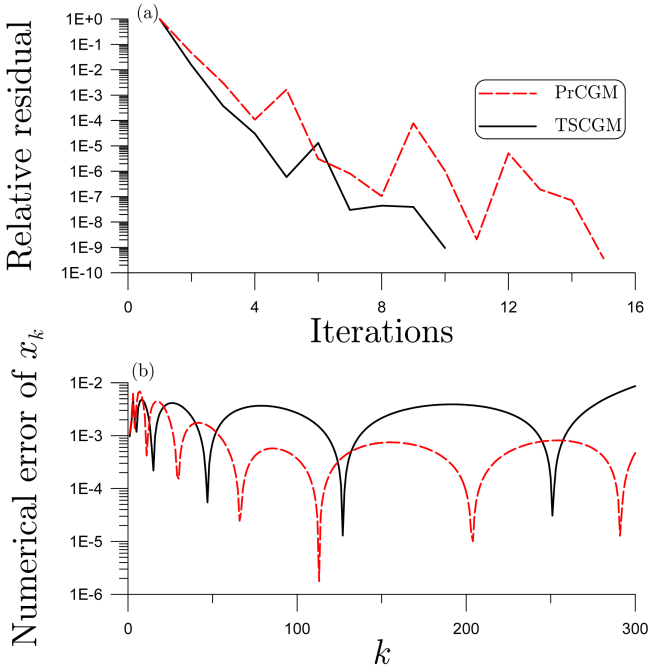

Figure 8: For a linear Hilbert problem with $n = 300$: (a) the residual errors, and (b) the numerical errors.

6 Conclusions

The present paper has proposed novel processes which can fast and significantly reduce the condition number of a given non-singular matrix only through a few operations. Thus we can solve the new system quite effectively and accurately by using the conjugate gradient method and the pre-conditioning conjugate gradient method to an ill-posed linear system. The numerical examples of the Hilbert linear equations system, backward heat conduction problem, and inverse Cauchy problems were addressed in this paper by using two different conditioners together with the CGM; they are, respectively, the two-side CGM and the pre-conditioning CGM based on the idea of equilibrated norm for the conditioned matrices. It is important that the efficiency and accuracy of the proposed novel algorithms are superior than the previous results computed by other algorithms developed by the author and his co-workers.

Acknowledgement: Taiwan's National Science Council project NSC-100-2221-E-002-165-MY3 and the 2011 Outstanding Research Award, granted to the author, are highly appreciated. Acknowledges that the author has been promoted to be a Lifetime Distinguished Professor of National Taiwan University, since 2013.

References

Bauer, F. L. (1963): Optimally scaled matrices. *Num. Math.*, vol. 5, pp. 73-87.

Bauer, F. L. (1969): Remarks on optimally scaled matrices. *Num. Math.*, vol. 13, pp. 1-3.

Benzi, M. (2002): Preconditioning techniques for large linear systems: a survey. *J. Comput. Phys.*, vol. 182, pp. 418-477.

Chen, C. S.; Cho, H. A.; Golberg, M. A. (2006): Some comments on the ill-conditioning of the method of fundamental solutions. *Eng. Anal. Bound. Elem.*, vol. 30, pp. 405-410.

Gautschi, W. (2011): Optimally scaled and optimally conditioned Vandermonde and Vandermonde-like matrices. *BIT Num. Math.*, vol. 51, pp. 103-125.

Golberg, M. A.; Chen, C. S. (1996): Discrete Projection Methods for Integral Equations. Computational Mechanics Publications, Southampton (1996).

Hon, Y. C.; Li, M. (2009): A discrepancy principle for the source points location in using the MFS for solving the BHCP. *Int. J. Comput. Meth.*, vol. 6, pp. 181-197.

Kupradze, V. D.; Aleksidze, M. A. (1964): The method of functional equations for the approximate solution of certain boundary value problems. *USSR Comput. Math. Math. Phy.*, vol. 4, pp. 82-126.

Liu, C.-S. (2007a): A modified Trefftz method for two-dimensional Laplace equation considering the domain's characteristic length. *CMES: Computer Modeling in Engineering & Sciences*, vol. 21, pp. 53-65.

Liu, C.-S. (2007b): An effectively modified direct Trefftz method for 2D potential problems considering the domain's characteristic length. *Eng. Anal. Bound. Elem.*, vol. 31, pp. 983-993.

Liu, C.-S. (2008a): Improving the ill-conditioning of the method of fundamental solutions for 2D Laplace equation. *CMES: Computer Modeling in Engineering & Sciences*, vol. 28, pp. 77-93.

Liu, C.-S. (2008b): A highly accurate MCTM for direct and inverse problems of biharmonic equation in arbitrary plane domains. *CMES: Computer Modeling in Engineering & Sciences*, vol. 30, pp. 65-75.

Liu, C.-S. (2011a) A revision of relaxed steepest descent method from the dynamics on an invariant manifold. *CMES: Computer Modeling in Engineering & Sciences*, vol. 80, pp. 57-86.

Liu, C.-S. (2011b): The method of fundamental solutions for solving the backward heat conduction problem with conditioning by a new post-conditioner. *Num. Heat Transfer, B: Fundamentals*, vol. 60, pp. 57-72.

Liu, C.-S. (2012a): Optimally scaled vector regularization method to solve ill-posed linear problems. *Appl. Math. Comp.*, vol. 218, pp. 10602-10616.

Liu, C.-S. (2012b): Optimally generalized regularization methods for solving linear inverse problems. *CMC: Computers, Materials & Continua*, vol. 29, pp. 103-127.

Liu, C.-S. (2012c): The concept of best vector used to solve ill-posed linear inverse problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 83, pp. 499-525.

Liu, C.-S. (2012d): A globally optimal iterative algorithm to solve an ill-posed linear system. *CMES: Computer Modeling in Engineering & Sciences*, vol. 84, pp. 383-403.

Liu, C.-S. (2012e): An equilibrated method of fundamental solutions to choose the best source points for the Laplace equation. *Eng. Anal. Bound. Elem.*, vol. 36, pp. 1235-1245.

Liu, C.-S. (2013a): A dynamical Tikhonov regularization for solving ill-posed linear algebraic systems. *Acta Appl. Math.*, vol. 123, pp. 285-307.

Liu, C.-S. (2013b): An optimal tri-vector iterative algorithm for solving ill-posed linear inverse problems, *Inv. Prob. Sci. Eng.*, dx.doi.org/10.1080/17415977.2012.717077 (2013).

Liu, C.-S. (2013c): An optimal preconditioner with an alternate relaxation parameter used to solve ill-posed linear problems. *CMES: Computer Modeling in Engineering & Sciences*, in press.

Liu, C.-S. (2013d): An optimal multi-vector iterative algorithm in a Krylov subspace for solving the ill-posed linear inverse problems. *CMC: Computers, Materials & Continua*, vol. 33, pp.175-198.

Liu, C.-S.; Atluri, S. N. (2009a): A fictitious time integration method for the numerical solution of the Fredholm integral equation and for numerical differentiation of noisy data, and its relation to the filter theory. *CMES: Computer Modeling in Engineering & Sciences*, vol. 41, pp. 243-261.

Liu, C.-S.; Atluri, S. N. (2009b): A highly accurate technique for interpolations using very high-order polynomials, and its applications to some ill-posed linear problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 43, pp. 253-276.

Liu, C.-S.; Atluri, S. N. (2011): An iterative method using an optimal descent vector, for solving an ill-conditioned system $\mathbf{Bx} = \mathbf{b}$, better and faster than the conjugate gradient method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 80, pp. 275-298.

Liu, C.-S.; Atluri, S. N. (2013): Numerical solution of the Laplacian Cauchy problem by using a better postconditioning collocation Trefftz method. *Eng. Anal. Bound. Elem.*, vol. 37, pp. 74-83.

Liu, C.-S.; Chang, C. W. (2009): Novel methods for solving severely ill-posed linear equations system. *J. Marine Sci. Tech.*, vol. 17, pp. 216-227.

Liu, C.-S. Hong, H. K.; Atluri, S. N. (2010): Novel algorithms based on the conjugate gradient method for inverting ill-conditioned matrices, and a new regularization method to solve ill-posed linear systems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 60, pp. 279-308.

Liu, C.-S.; Yeih, W.; Atluri, S. N. (2009): On solving the ill-conditioned system $\mathbf{Ax} = \mathbf{b}$: general-purpose conditioners obtained from the boundary-collocation solution of the Laplace equation, using Trefftz expansions with multiple length scales. *CMES: Computer Modeling in Engineering & Sciences*, vol. 44, pp. 281-311.

Vajargah, B. F.; Moradi, M. (2012): Diagonal scaling of ill-conditioned matrices by genetic algorithm, *J. Appl. Math. Stat. Inform.*, vol. 8, pp. 49-53.

van der Sluis, A. (1969): Condition numbers and equilibration of matrices. *Num. Math.*, vol. 14, pp. 14-23.

Watson, G. A. (1991): An algorithm for optimal ℓ_2 scaling of matrices. *IMA J. Num. Anal.*, vol. 11, pp. 481-492.

Wei, T.; Hon, Y. C.; Ling, L. (2007): Method of fundamental solutions with regularization techniques for Cauchy problems of elliptic operators. *Eng. Anal. Bound. Elem.*, vol. 31, pp. 373-385.