Copyright © 2010 Tech Science Press CMES, vol.60, no.3, pp.279-308, 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

Chein-Shan Liu[1], Hong-Ki Hong[1] and Satya N. Atluri[2]

Abstract: We propose novel algorithms to calculate the inverses of ill-conditioned matrices, which have broad engineering applications. The vector-form of the conjugate gradient method (CGM) is recast into a matrix-form, which is named as the matrix conjugate gradient method (MCGM). The MCGM is better than the CGM for finding the inverses of matrices. To treat the problems of inverting ill-conditioned matrices, we add a vector equation into the given matrix equation for obtaining the left-inversion of matrix (and a similar vector equation for the right-inversion) and thus we obtain an over-determined system. The resulting two modifications of the MCGM, namely the MCGM1 and MCGM2, are found to be much better for finding the inverses of ill-conditioned matrices, such as the Vandermonde matrix and the Hilbert matrix. We propose a natural regularization method for solving an ill-posed linear system, which is theoretically and numerically proven in this paper, to be better than the well-known Tikhonov regularization. The presently proposed natural regularization is shown to be equivalent to using a new preconditioner, with better conditioning. The robustness of the presently proposed method provides a significant improvement in the solution of ill-posed linear problems, and its convergence is as fast as the CGM for the well-posed linear problems.

Keywords: Ill-posed linear system, Inversion of ill-conditioned matrix, Left-inversion, Right-inversion, Regularization vector, Vandermonde matrix, Hilbert matrix, Tikhonov regularization

---

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan. E-mail: liucs@ntu.edu.tw
[2] Center for Aerospace Research & Education, University of California, Irvine

1 Introduction

In this paper we propose novel regularization techniques to solve the following linear system of algebraic equations:

$$\mathbf{V}\mathbf{x} = \mathbf{b}_1, \tag{1}$$

where $\det(\mathbf{V}) \neq 0$ and $\mathbf{V}$ may be an ill-conditioned, and generally unsymmetric matrix. The solution of such an ill-posed system of linear equations is an important issue for many engineering problems while using the boundary element method [Han and Olson (1987); Wen, Aliabadi and Young (2002); Atluri (2005); Karlis, Tsinopoulos, Polyzos and Beskos (2008)], MLPG method [Atluri, Kim and Cho (1999); Atluri and Shen (2002); Tang, Shen and Atluri (2003); Atluri (2004); Atluri and Zhu (1998)], or the method of fundamental solutions [Fairweather and Karageorghis (1998); Young, Tsai, Lin and Chen (2006); Tsai, Lin, Young and Atluri (2006); Liu (2008)].

In practical situations of linear equations which arise in engineering problems, the data $\mathbf{b}_1$ are rarely given exactly; instead, noises in $\mathbf{b}_1$ are unavoidable due to the measurement error. Therefore, we may encounter the problem wherein the numerical solution of an ill-posed system of linear equations may deviate from the exact one to a great extent, when $\mathbf{V}$ is severely ill-conditioned and $\mathbf{b}_1$ is perturbed by noise.

To account for the sensitivity to noise, it is customary to use a "regularization" method to solve this sort of ill-posed problem [Kunisch and Zou (1998); Wang and Xiao (2001); Xie and Zou (2002); Resmerita (2005)], wherein a suitable regularization parameter is used to suppress the bias in the computed solution, by seeking a better balance of the error of approximation and the propagated data error. Several regularization techniques were developed, following the pioneering work of Tikhonov and Arsenin (1977). For a large scale system, the main choice is to use the iterative regularization algorithm, wherein the regularization parameter is represented by the number of iterations. The iterative method works if an early stopping criterion is used to prevent the introduction of noisy components into the approximated solutions.

The Vandermonde matrices arise in a variety of mathematical applications. Some example situations are polynomial interpolations, numerical differentiation, approximation of linear functionals, rational Chebyshev approximation, and differential quadrature. In these applications, finding the solution of a linear system with the Vandermonde matrix as a coefficient matrix, and the inversion of Vandermonde matrix are required. So an efficient method to finding the inversion of Vandermonde matrix is desirable. The condition number of Vandermonde matrix may be quite large [Gautschi (1975)], causing large errors when computing the inverse of a large scale Vandermonde matrix. Several authors have therefore proposed algorithms which exploit the structure of Vandermonde matrix to numerically compute stable solutions in operations different from those required by the Gaussian elimination [Higham (1987, 1988); Björck and Pereyra (1970); Calvetti and Reichel (1993)]. These methods rely on constructing first a Newton interpolation of the polynomial and then converting it to the monomial form. Wertz (1965) suggested a simple numerical procedure, which can greatly facilitate the computation of the inverse of Vandermonde matrix. Neagoe (1996) has found an analytic formula to calculate the inverse of Vandermonde matrix. However, a direct application of Neagoe's formula will result in a tedious algorithm with $O(n^3)$ flops. Other analytical inversions were also reported by El-Mikkawy (2003), Skrzipek (2004), Jog (2004), and Eisinberg and Fedele (2006). Some discussions about the numerical algorithms for the inversion of Vandermonde matrix are summarized by Gohberg and Olshevsky (1997).

Indeed, the polynomial interpolation is an ill-posed problem and it makes the interpolation by higher-degree polynomials as not being easy for numerical implementation. In order to overcome those difficulties, Liu and Atluri (2009a) have introduced a characteristic length into the high-order polynomials expansion, which improved the accuracy for the applications to some ill-posed linear problems. At the same time, Liu, Yeih and Atluri (2009) have developed a multi-scale Trefftz-collocation Laplacian conditioner to deal with the ill-conditioned linear systems. Also, Liu and Atluri (2009b), using a Fictitious Time Inegration Method, have introduced a new filter theory for ill-conditioned linear systems. In this paper we will propose a new, simple and direct regularization technique to overcome the above-mentioned ill-conditioned behavior for the general ill-posed linear system of equations. This paper is organized as follows. For use in the following sections, we describe the conjugate gradient method for a linear system of equations in Section 2. Then we construct a matrix conjugate gradient method (MCGM) for a linear system of matrix equations in Section 3, where the left-inversion of an ill-conditioned matrix is computed. In Section 4 we propose two modifications of the matrix conjugate gradient method (MCGM) by adding a vector equation in the left-inversion matrix equation and combining them with the right-inversion matrix equation. Those two algorithms for the inversion of ill-conditioned matrix are called MCGM1 and MCGM2, respectively. Then we project the algorithm MCGM1 into the vector space of linear systems in Section 5, where we indeed describe a novel, simple, and direct regularization of the linear system for the solution of ill-posed linear system of equations, which is then compared with the Tikhonov regularization. In Section 6 we give the numerical examples of the Vandermonde matrix and the Hilbert matrix, to test the accuracy of our novel algorithms for the inversion of matrix via four error measurements. Section 7 is devoted to the applications of the novel regularization method developed in Section 5 to the polynomial interpolation and the best polynomial approximation. Finally, some conclusions are drawn in Section 8.

2 The conjugate gradient method for solving $\mathbf{Ax} = \mathbf{b}$

The conjugate gradient method (CGM) is widely used to solve a positive definite linear system. The basic idea is to seek approximate solutions from the Krylov subspaces.

Instead of Eq. (1), we consider the normalized equation:

$$\mathbf{Ax} = \mathbf{b}, \tag{2}$$

where

$$\mathbf{A} := \mathbf{V}^{\mathrm{T}}\mathbf{V}, \tag{3}$$

$$\mathbf{b} := \mathbf{V}^{\mathrm{T}}\mathbf{b}_1. \tag{4}$$

The conjugate gradient method (CGM), which is used to solve the vector Eq. (2), is summarized as follows:

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

(ii) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ and $\mathbf{p}_1 = \mathbf{r}_0$.

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

$$\alpha_k = \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}, \tag{5}$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \alpha_k \mathbf{p}_k, \tag{6}$$

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k, \tag{7}$$

$$\eta_k = \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \tag{8}$$

$$\mathbf{p}_{k+1} = \mathbf{r}_k + \eta_k \mathbf{p}_k. \tag{9}$$

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

$$\|\mathbf{r}_k\| < \varepsilon, \tag{10}$$

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

In the present paper we seek to find the inverse of $\mathbf{V}$ [see Eq. (1)], denoted numerically by $\mathbf{U}$. To directly apply the above CGM to finding $\mathbf{U}$ by $\mathbf{VU} = \mathbf{I}_m$, we have to solve for an $m \times m$ matrix $\mathbf{U} = [\mathbf{u}_1^{\mathrm{T}}, \ldots, \mathbf{u}_m^{\mathrm{T}}]$, where the i-th column of $\mathbf{U}$ is computed via $\mathbf{Vu}_i = \mathbf{e}_i$, in which $\mathbf{e}_i$ is the i-th column of the identity matrix $\mathbf{I}_m$. This will increase the number of multiplications and the additions by m times, although the computer CPU time may not increase as much bacause most elements of $\mathbf{e}_i$ are zeros.

3 The matrix conjugate gradient method for inverting V

Let us begin with the following matrix equation:

$$\mathbf{V}^{\mathrm{T}}\mathbf{U}^{\mathrm{T}} = \mathbf{I}_m, \text{ i.e., } (\mathbf{UV})^{\mathrm{T}} = \mathbf{I}_m, \tag{11}$$

if $\mathbf{U}$ is the inversion of $\mathbf{V}$. Numerically, we can say that this $\mathbf{U}$ is a left-inversion of $\mathbf{V}$. Then we have

$$\mathbf{AU}^{\mathrm{T}} = (\mathbf{VV}^{\mathrm{T}})\mathbf{U}^{\mathrm{T}} = \mathbf{V}, \tag{12}$$

from which we can solve for $\mathbf{U}^{\mathrm{T}} := \mathbf{C}$.

The matrix conjugate gradient method (MCGM), which is used to solve the matrix Eq. (12), is summarized as follows:

(i) Assume an initial $\mathbf{C}_0$.

(ii) Calculate $\mathbf{R}_0 = \mathbf{V} - \mathbf{AC}_0$ and $\mathbf{P}_1 = \mathbf{R}_0$.

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

$$\alpha_k = \frac{\|\mathbf{R}_{k-1}\|^2}{\mathbf{P}_k \cdot (\mathbf{AP}_k)}, \tag{13}$$

$$\mathbf{C}_k = \mathbf{C}_{k-1} + \alpha_k \mathbf{P}_k, \tag{14}$$

$$\mathbf{R}_k = \mathbf{V} - \mathbf{AC}_k, \tag{15}$$

$$\eta_k = \frac{\|\mathbf{R}_k\|^2}{\|\mathbf{R}_{k-1}\|^2}, \tag{16}$$

$$\mathbf{P}_{k+1} = \mathbf{R}_k + \eta_k \mathbf{P}_k. \tag{17}$$

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

$$\|\mathbf{R}_k\| < \varepsilon, \tag{18}$$

then stop; otherwise, go to step (iii). In above the capital boldfaced letters denote $m \times m$ matrices, the norm $\|\mathbf{R}_k\|$ is the Frobenius norm (similar to the Euclidean norm for a vector), and the inner product is for matrices. When $\mathbf{C}$ is calculated, the inversion of $\mathbf{V}$ is given by $\mathbf{U} = \mathbf{C}^{\mathrm{T}}$.

4 Two modifications of the matrix conjugate gradient method for inverting V

In our experience the MCGM is much better than the original CGM for finding the inversion of a weakly ill-conditioned matrix. However, when the ill-posedness is stronger, we need to modify the MCGM. The first modification is by adding a natural vector equation into Eq. (11), borrowed from Eq. (1):

$$\mathbf{V}\mathbf{x}_0 = \mathbf{y}_0, \tag{19}$$

through which, given $\mathbf{x}_0$, say $\mathbf{x}_0 = \mathbf{1} = [1,\dots,1]^{\mathrm{T}}$, we can straightforwardly calculate $\mathbf{y}_0$, because $\mathbf{V}$ is a given matrix. Hence, we have

$$\mathbf{y}_0^{\mathrm{T}}\mathbf{U}^{\mathrm{T}} = \mathbf{x}_0^{\mathrm{T}}, \text{ i.e., } \mathbf{x}_0 = \mathbf{U}\mathbf{y}_0. \tag{20}$$

Together, Eqs. (11) and (20) constitute an over-determined system to calculate $\mathbf{U}^{\mathrm{T}}$. This over-determined system can be written as

$$\mathbf{B}\mathbf{U}^{\mathrm{T}} = \begin{bmatrix} \mathbf{I}_m \\ \mathbf{x}_0^{\mathrm{T}} \end{bmatrix}, \tag{21}$$

where

$$\mathbf{B} := \begin{bmatrix} \mathbf{V}^{\mathrm{T}} \\ \mathbf{y}_0^{\mathrm{T}} \end{bmatrix} \tag{22}$$

is an $n \times m$ matrix with $n = m+1$. Multiplying Eq. (21) by $\mathbf{B}^{\mathrm{T}}$, we obtain an $m \times m$ matrix equation again:

$$[\mathbf{V}\mathbf{V}^{\mathrm{T}} + \mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{U}^{\mathrm{T}} = \mathbf{V} + \mathbf{y}_0\mathbf{x}_0^{\mathrm{T}}, \tag{23}$$

which, similar to Eq. (12), is solved by the MCGM. This algorithm for solving the inverse of an ill-conditioned matrix is labelled here as the MCGM1 method. *The flow chart to compute the left-inversion of* $\mathbf{V}$ *is summarized in Fig. 1.*

The above algorithm is suitable for finding the left-inversion of $\mathbf{V}$; however, we also need to solve

$$\mathbf{V}\mathbf{U} = \mathbf{I}_m, \tag{24}$$

Box 1: Flow Chart for Left-Inversion of V when V is ill-conditioned

(i) Choose a suitable $\mathbf{x}_0$ and $\mathbf{y}_0 = \mathbf{V}\mathbf{x}_0$.

(ii) Let $\mathbf{A} = \mathbf{V}^{\mathrm{T}}\mathbf{V} + \mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}$.

(iii) Let $\mathbf{B} = \mathbf{V} + \mathbf{y}_0\mathbf{x}_0^{\mathrm{T}}$.

(iv) Assume an initial $\mathbf{C}_0$.

(v) Calculate $\mathbf{R}_0 = \mathbf{B} - \mathbf{A}\mathbf{C}_0$ and $\mathbf{P}_1 = \mathbf{R}_0$.

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

$\alpha_k = \frac{\|\mathbf{R}_{k-1}\|^2}{\mathbf{P}_k \cdot (\mathbf{A}\mathbf{P}_k)}$,

$\mathbf{C}_k = \mathbf{C}_{k-1} + \alpha_k \mathbf{P}_k$,

$\mathbf{R}_k = \mathbf{B} - \mathbf{A}\mathbf{C}_k$,

$\eta_k = \frac{\|\mathbf{R}_k\|^2}{\|\mathbf{R}_{k-1}\|^2}$,

$\mathbf{P}_{k+1} = \mathbf{R}_k + \eta_k \mathbf{P}_k$.

If $\|\mathbf{R}_k\| < \varepsilon$, then stop; otherwise, go to step (vi).

(vii) Let $\mathbf{V}^{-1} = \mathbf{C}^{\mathrm{T}}$.

Figure 1: The flow chart to compute the left-inversion of a given matrix **V**.

when we want **U** also as a right-inversion of **V**. Mathematically, the left-inversion is equal to the right-inversion. But numerically they are hardly equal, especially for ill-conditioned matrices.

For the right-inversion we can supplement, as in Eq. (19), another equation:

$$\mathbf{y}_1^{\mathrm{T}}\mathbf{U} = \mathbf{x}_1^{\mathrm{T}}, \text{ i.e., } \mathbf{y}_1 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_1. \tag{25}$$

Then the combination of Eqs. (24), (25), (11) and (20) leads to the following over-determined system:

$$\begin{bmatrix} \mathbf{V} & \mathbf{0} \\ \mathbf{y}_1^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{y}_0^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{x}_1^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_m \\ \mathbf{0} & \mathbf{x}_0^{\mathrm{T}} \end{bmatrix}. \tag{26}$$

Then, multiplying the transpose of the leading matrix, we can obtain an $2m \times 2m$ matrix equation:

$$\begin{bmatrix} \mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_1\mathbf{y}_1^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}\mathbf{V}^{\mathrm{T}}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{U}^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \mathbf{V}^{\mathrm{T}}+\mathbf{y}_1\mathbf{x}_1^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}+\mathbf{y}_0\mathbf{x}_0^{\mathrm{T}} \end{bmatrix}, \tag{27}$$

which is then solved by the MCGM for the following two $m \times m$ matrix equations:

$$[\mathbf{V}\mathbf{V}^{\mathrm{T}}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{U}^{\mathrm{T}} = \mathbf{V}+\mathbf{y}_0\mathbf{x}_0^{\mathrm{T}}, \tag{28}$$

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_1\mathbf{y}_1^{\mathrm{T}}]\mathbf{U} = \mathbf{V}^{\mathrm{T}}+\mathbf{y}_1\mathbf{x}_1^{\mathrm{T}}. \tag{29}$$

This algorithm for solving the inversion problem of ill-conditioned matrix is labelled as the MCGM2 method. The MCGM2 can provide both the solutions of $\mathbf{U}$ as well as $\mathbf{U}^{\mathrm{T}}$, and thus we can choose one of them as the inversion of $\mathbf{V}$. For the inversion of matrix we prefer the right-inversion obtained from Eq. (29).

5 A new simple and direct regularization of an ill-posed linear system

5.1 A natural regularization

Besides the primal system in Eq. (1), sometimes we need to solve the dual system with

$$\mathbf{V}^{\mathrm{T}}\mathbf{y} = \mathbf{b}_1. \tag{30}$$

Applying the operators in Eq. (23) to $\mathbf{b}_1$ and utilizing the above equation, i.e., $\mathbf{y} = \mathbf{U}^{\mathrm{T}}\mathbf{b}_1$, we can obtain

$$[\mathbf{V}\mathbf{V}^{\mathrm{T}}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{y} = \mathbf{V}\mathbf{b}_1+(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0, \tag{31}$$

where $\mathbf{y}_0 = \mathbf{V}\mathbf{x}_0$.

Replacing the $\mathbf{V}$ in Eq. (31) by $\mathbf{V}^{\mathrm{T}}$, we have a similar equation for the primal system in Eq. (1):

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{x} = \mathbf{V}^{\mathrm{T}}\mathbf{b}_1+(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0, \tag{32}$$

where $\mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_0$.

In Eq. (32), $\mathbf{x}_0$ is a regularization vector, which can be chosen orthogonal to the input data $\mathbf{b}_1$, such that

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{x} = \mathbf{b}, \tag{33}$$

where $\mathbf{b}$ is defined in Eq. (4). It bears certain similarity with the following Tikhonov regularization equation:

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V}+\alpha\mathbf{I}_m]\mathbf{x}=\mathbf{b}, \tag{34}$$

where α is a regularization parameter. However, we need to stress that Eqs. (31)-(33) are simple and direct regularization equations for an ill-posed linear system. The Tikhonov regularization perturbs the original system to a new one by adding a regularization parameter α. The present novel regularization method does not perturb the original system, but mathematically converts it to a new one through a regularization vector $\mathbf{y}_0=\mathbf{V}\mathbf{x}_0$. *The flow chart to compute the solution of* $\mathbf{V}\mathbf{x}=\mathbf{b}_1$ *is summarized in Fig. 2.*

Box 2: Flow Chart for Solving $\mathbf{V}\mathbf{x}=\mathbf{b}_1$ when V is ill-conditioned, and $\mathbf{b}_1$ is noisy

(i) Choose a suitable $\mathbf{x}_0$.

(ii) Let $\mathbf{y}_0=\mathbf{V}\mathbf{x}_0$ or $\mathbf{y}_0=\mathbf{V}^{\mathrm{T}}\mathbf{x}_0$.

(iii) Let $\mathbf{A}=\mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}$ and $\mathbf{b}=\mathbf{V}^{\mathrm{T}}\mathbf{b}_1+(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0$.

(iv) Assume an initial $\mathbf{c}_0$.

(v) Calculate $\mathbf{r}_0=\mathbf{b}-\mathbf{A}\mathbf{c}_0$ and $\mathbf{p}_1=\mathbf{r}_0$.

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

$\alpha_k=\frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k\cdot(\mathbf{A}\mathbf{p}_k)}$,

$\mathbf{c}_k=\mathbf{c}_{k-1}+\alpha_k\mathbf{p}_k$,

$\mathbf{r}_k=\mathbf{b}-\mathbf{A}\mathbf{c}_k$,

$\eta_k=\frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}$,

$\mathbf{p}_{k+1}=\mathbf{r}_k+\eta_k\mathbf{p}_k$.

If $\|\mathbf{r}_k\|<\varepsilon$, then stop; otherwise, go to step (vi).

(vii) Let $\mathbf{x}=\mathbf{c}$.

Figure 2: The flow chart to compute the solution of a given ill-posed linear system $\mathbf{V}\mathbf{x}=\mathbf{b}_1$.

Regularization can be employed when one solves Eq. (1), when $\mathbf{V}$ is highly ill-conditioned. Hansen (1992) and Hansen and O'Leary (1993) have given an illuminating explanation that the Tikhonov regularization of linear problems is a trade-off between the size of the regularized solution and the quality to fit the given data:

$$\min_{\mathbf{x}\in\mathbb{R}^m} \varphi(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^m} \left[\|\mathbf{V}\mathbf{x}-\mathbf{b}_1\|^2 + \alpha\|\mathbf{x}\|^2\right]. \tag{35}$$

A generalization of Eq. (35) can be written as

$$\min_{\mathbf{x}\in\mathbb{R}^m} \varphi(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^m} \left[\|\mathbf{V}\mathbf{x}-\mathbf{b}_1\|^2 + \mathbf{x}^{\mathrm{T}}\mathbf{Q}\mathbf{x}\right], \tag{36}$$

where $\mathbf{Q}$ is a non-negative definite matrix. In our case in Eq. (33), $\mathbf{Q} := \mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}$. From the above discussions it can be seen that the present regularization method is the most natural one, because the regularization vector $\mathbf{y}_0$ is generated from the original system.

A simple example illustrates that the present regularization method is much better than the well-known Tikhonov regularization method. Before embarking on a further analysis of the present regularization method, we give a simple example of the solution of a linear system of two linear algebraic equations:

$$\begin{bmatrix} 2 & 6 \\ 2 & 6.00001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8 \\ 8.00001 \end{bmatrix}. \tag{37}$$

The exact solution is $(x,y)=(1,1)$. We use the above novel regularization method to solve this problem with $\mathbf{x}_0=(1,1)^{\mathrm{T}}$ and $\mathbf{y}_0=\mathbf{V}\mathbf{x}_0$ is calculated accordingly. It is interesting to note that the condition number is greatly reduced from $\mathrm{Cond}(\mathbf{V}^{\mathrm{T}}\mathbf{V}) = 1.59\times10^{13}$ to $\mathrm{Cond}(\mathbf{V}^{\mathrm{T}}\mathbf{V}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}) = 19.1$. Then, when we add a random noise 0.01 on the data of $(8,8.00001)^{\mathrm{T}}$, we obtain a solution of $(x,y)=(1.00005,1.00005)$ through two iterations by employing the CGM to solve the resultant linear system (32). However, no matter what parameter of α is used in the Tikhonov regularization method for the above equation, we get an incorrect solution of $(x,y) = (1356.4,-450.8)$ through four iterations by employing the CGM to solve the linear system.

5.2 The present natural regularization is equivalent to using a preconditioner

Now, we prove that the solution of Eq. (32) is mathematically equivalent to the solution of Eq. (2). If $\mathbf{A}$ can be inverted exactly, the solution of Eq. (2) is written as

$$\tilde{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}. \tag{38}$$

Similarly, for Eq. (32) we have

$$\mathbf{x} = [\mathbf{A}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]^{-1}[\mathbf{b}+(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0]. \tag{39}$$

By using the Sherman-Morrison formula:

$$[\mathbf{A}+\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}}{1+\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{y}_0}, \tag{40}$$

we obtain

$$\begin{aligned}\mathbf{x} &= \mathbf{A}^{-1}\mathbf{b} - \frac{\mathbf{A}^{-1}\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}{1+\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{y}_0} \\ &+ (\mathbf{x}_0\cdot\mathbf{b}_1)\left[\mathbf{A}^{-1}\mathbf{y}_0 - \frac{\mathbf{A}^{-1}\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{y}_0}{1+\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{y}_0}\right].\end{aligned} \tag{41}$$

By using Eq. (38) and through some algebraic manipulations we can derive

$$\mathbf{x} = \tilde{\mathbf{x}} + \frac{\mathbf{x}_0\cdot\mathbf{b}_1 - \mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}{1+\mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{y}_0}\mathbf{A}^{-1}\mathbf{y}_0. \tag{42}$$

Further using the relation:

$$\mathbf{x}_0\cdot\mathbf{b}_1 - \mathbf{y}_0^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b} = \mathbf{x}_0\cdot\mathbf{b}_1 - \mathbf{x}_0^{\mathrm{T}}\mathbf{V}(\mathbf{V}^{\mathrm{T}}\mathbf{V})^{-1}\mathbf{V}^{\mathrm{T}}\mathbf{b}_1 = 0,$$

we can prove that

$$\mathbf{x} = \tilde{\mathbf{x}}. \tag{43}$$

Next, we will explain that the naturally regularized Eq. (32) is equivalent to a preconditioned equation. Let $\mathbf{A} = \mathbf{V}^{\mathrm{T}}\mathbf{V}$. Then $\mathbf{A}$ is positive definite because of $\det(\mathbf{V}) \neq 0$. Let $\mathbf{x}_0 = \mathbf{V}\mathbf{z}_0$, where $\mathbf{z}_0$ instead of $\mathbf{x}_0$, is a free vector. Then by $\mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_0$ we have $\mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{V}\mathbf{z}_0 = \mathbf{A}\mathbf{z}_0$.
Inserting $\mathbf{y}_0 = \mathbf{A}\mathbf{z}_0$ into Eq. (32) and using Eqs. (3) and (4) we can derive

$$[\mathbf{A}+\mathbf{A}\mathbf{z}_0\mathbf{z}_0^{\mathrm{T}}\mathbf{A}]\mathbf{x} = \mathbf{b} + (\mathbf{z}_0\cdot\mathbf{b})\mathbf{A}\mathbf{z}_0, \tag{44}$$

where $\mathbf{x}_0\cdot\mathbf{b}_1 = \mathbf{z}_0\cdot\mathbf{b}$ was used.
Let

$$\mathbf{P} := \mathbf{I}_m + \mathbf{A}\mathbf{z}_0\mathbf{z}_0^{\mathrm{T}} \tag{45}$$

be a preconditioned matrix. Then Eq. (44) can be written as

$$\mathbf{P}\mathbf{A}\mathbf{x} = \mathbf{P}\mathbf{b}, \tag{46}$$

which is just Eq. (2) multiplied by a preconditioner $\mathbf{P}$.

By definition (45), it is easy to prove that

$$(\mathbf{PA})^{\mathrm{T}} = \mathbf{AP}^{\mathrm{T}} = \mathbf{A} + \mathbf{A}\mathbf{z}_0\mathbf{z}_0^{\mathrm{T}}\mathbf{A} = \mathbf{PA}, \tag{47}$$

which means that the new system matrix in Eq. (46) is symmetric and positive definite because $\mathbf{A}$ is positive definite.

From the above results, we can understand that the naturally regularized Eq. (32) is equivalent to the original equation (2) multiplied by a preconditioner. This regularization mechanism is different from the Tikhonov regularization, which is an approximation of the original system. Here, we do not disturb the original system, but the use of the discussed preconditioner leads to a better conditioning of the coefficient matrix $\mathbf{PA}$ (see the simple example given in Section 5.1 and the next section).

5.3 Reducing the condition number by the use of the present type of a natural regularization

At the very beginning, if the supplemented equations (19) and (25) are written as $\beta\mathbf{y}_0 = \beta\mathbf{V}\mathbf{x}_0$ and $\beta\mathbf{y}_0 = \beta\mathbf{V}^{\mathrm{T}}\mathbf{x}_0$, where β plays the role of a weighting factor for weighting the supplemented equation in the least-squares solution, then we can derive

$$\text{Dual System: } [\mathbf{V}\mathbf{V}^{\mathrm{T}} + \beta^2\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{y} = \mathbf{V}\mathbf{b}_1 + \beta^2(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0, \ \ \mathbf{y}_0 = \mathbf{V}\mathbf{x}_0, \tag{48}$$

$$\text{Primal System: } [\mathbf{V}^{\mathrm{T}}\mathbf{V} + \beta^2\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{x} = \mathbf{V}^{\mathrm{T}}\mathbf{b}_1 + \beta^2(\mathbf{x}_0\cdot\mathbf{b}_1)\mathbf{y}_0, \ \ \mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_0. \tag{49}$$

Below we only discuss the primal system, while the results are also true for the dual system. Suppose that $\mathbf{A}$ has a singular-value decomposition:

$$\mathbf{A} = \mathbf{W}\mathrm{diag}\{s_i\}\mathbf{W}^{\mathrm{T}}, \tag{50}$$

where s_i are the singular values of $\mathbf{A}$ with $0 < s_1 \leq s_2 \leq \ldots \leq s_m$. Thus, Eq. (2) has an exact solution:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{W}\mathrm{diag}\{s_i^{-1}\}\mathbf{W}^{\mathrm{T}}\mathbf{b}. \tag{51}$$

However, this solution may be incorrect when the data of $\mathbf{b}$ are noisy. The effect of regularization is to modify s_i^{-1} for those singular values which are very small, by

$$\omega(s_i^2)s_i^{-1},$$

where $\omega(s)$ is called a filter function. So, instead of Eq. (51) we can obtain a regularized solution:

$$\mathbf{x} = \mathbf{W}\mathrm{diag}\{\omega(s_i^2)s_i^{-1}\}\mathbf{W}^{\mathrm{T}}\mathbf{b}, \tag{52}$$

where $\omega(s_i^2)s_i^{-1} \to 0$ when $s_i \to 0$. Obviously, from the Tikhonov regularization, we can derive a filter function such that

$$\omega(s) = \frac{s}{s+\alpha}, \tag{53}$$

which is named the Tikhonov filter function, and α is a regularization parameter. The above discussions were elaborated on, in the paper by Liu and Atluri (2009b).

Suppose that $\mathbf{e}_1$ is the corresponding eigenvector of s_1 for $\mathbf{A}$:

$$\mathbf{A}\mathbf{e}_1 = s_1\mathbf{e}_1. \tag{54}$$

If the free vector $\mathbf{x}_0$ is chosen to be

$$\mathbf{x}_0 = \mathbf{V}\mathbf{e}_1, \tag{55}$$

then we have

$$\mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_0 = \mathbf{A}\mathbf{e}_1 = s_1\mathbf{e}_1. \tag{56}$$

Inserting Eq. (56) into the system matrix in the primal system (49), we have

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V} + \beta^2\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{e}_1 = \mathbf{A}\mathbf{e}_1 + \beta^2 s_1^2\|\mathbf{e}_1\|^2\mathbf{e}_1 = (s_1 + \beta^2 s_1^2)\mathbf{e}_1, \tag{57}$$

where the eigenvector $\mathbf{e}_1$ is normalized by taking $\|\mathbf{e}_1\|^2 = 1$. Eq. (57) means that the original eigenvalue s_1 for $\mathbf{A}$ is modified to $s_1 + \beta^2 s_1^2$ for the primal system in Eq. (49).

Unlike the parameter α in the Tikhonov regularization, which must be a small value in order to not disturb the original system too much, we can choose the parameter β to be large enough, such that the condition number of the primal system in Eq. (49) can be reduced to

$$\mathrm{Cond}[\mathbf{V}^{\mathrm{T}}\mathbf{V} + \beta^2\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}] = \frac{s_m}{s_1 + \beta^2 s_1^2} \ll \mathrm{Cond}(\mathbf{A}) = \frac{s_m}{s_1}. \tag{58}$$

For the ill-conditioned linear system in Eq. (2), the $\mathrm{Cond}(\mathbf{A})$ can be quite large due to the small s_1. However, the regularized primal system in Eq. (49) provides a mechanism to reduce the condition number by a significant amount. This natural regularization not only modifies the left-hand side of the system equations but also the right-hand side. This situation is quite different from the Tikhonov regularization, which only modifies the left-hand side of the system equations, and thus the modification parameter α is restricted to be small enough. In our regularization, β can be quite large, because we do not disturb the original system any more.

More interestingly, as shown in Eqs. (52) and (53), while the Tikhonov regularization disturbs all singular values by a quantity α, which causes solution error, the present regularization does not disturb other singular values, because of

$$[\mathbf{V}^{\mathrm{T}}\mathbf{V}+\beta^2\mathbf{y}_0\mathbf{y}_0^{\mathrm{T}}]\mathbf{e}_i=\mathbf{A}\mathbf{e}_i+\beta^2 s_1^2(\mathbf{e}_1\cdot\mathbf{e}_i)\mathbf{e}_i=s_i\mathbf{e}_i,\ \ i\geq 2, \tag{59}$$

where s_i and $\mathbf{e}_i$ are the corresponding eigenvalues and eigenvectors of $\mathbf{A}$, and $\mathbf{e}_1\cdot\mathbf{e}_i=0,\ i\geq 2$ due to the positiveness of $\mathbf{A}$.

6 Error assessment through numerical examples

We evaluate the accuracy of the inversion $\mathbf{U}$ for $\mathbf{V}$ by

$$e_1=|\|\mathbf{U}\mathbf{V}\|-\sqrt{m}|, \tag{60}$$
$$e_2=\|\mathbf{U}\mathbf{V}-\mathbf{I}_m\|, \tag{61}$$
$$e_3=|\|\mathbf{V}\mathbf{U}\|-\sqrt{m}|, \tag{62}$$
$$e_4=\|\mathbf{V}\mathbf{U}-\mathbf{I}_m\|, \tag{63}$$

where m is the dimension of $\mathbf{V}$. In order to distinguish the above algorithms introduced in Sections 3 and 4 we call them MCGM, MCGM1, and MCGM2, respectively.

6.1 Vandermonde matrices

First we consider the following Vandermonde matrix:

$$\mathbf{V}=\begin{bmatrix} 1 & 1 & \ldots & 1 & 1 \\ x_1 & x_2 & \ldots & x_{m-1} & x_m \\ x_1^2 & x_2^2 & \ldots & x_{m-1}^2 & x_m^2 \\ \vdots & \vdots & \ldots & \vdots & \vdots \\ x_1^{m-2} & x_2^{m-2} & \ldots & x_{m-1}^{m-2} & x_m^{m-2} \\ x_1^{m-1} & x_2^{m-1} & \ldots & x_{m-1}^{m-1} & x_m^{m-1} \end{bmatrix}, \tag{64}$$

where the nodes are generated from $x_i=(i-1)/(m-1)$, which are eqidistant nodes in the unit interval. Gohberg and Olshevsky (1997) have demonstrated the ill-condition of this case that $\mathrm{Cond}(\mathbf{V})=6\times 10^7$ when $m=10$, and $\mathrm{Cond}(\mathbf{V})=4\times 10^{18}$ when $m=30$.

The CGM to finding the inversion of matrix has been demonstrated in Section 2. For $m=9$ the CGM with $\varepsilon=10^{-9}$ leads to the acceptable $e_3=2.36\times 10^{-6}$ and

Table 1: Comparing the e_i with different methods

Errors of	e_1	e_2	e_3	e_4
CGM	4.42	2.679	2.36×10^{-6}	2.06×10^{-5}
MCGM	4.14×10^{-6}	1.26×10^{-5}	4.67×10^{-3}	0.17
MCGM1	1.85×10^{-6}	5.90×10^{-6}	6.47×10^{-1}	2.08
MCGM2(R)	2.82×10^{-4}	4.15×10^{-2}	5.14×10^{-6}	1.50×10^{-5}
MCGM2(L)	5.31×10^{-6}	1.57×10^{-5}	4.77×10^{-3}	1.69×10^{-1}

$e_4 = 2.06\times10^{-5}$, but with the worser $e_1 = 4.42$ and $e_2 = 6.79$ as shown in Table 1, because the CGM is to finding the right-inversion by $\mathbf{VU} = \mathbf{I}_m$.

For the comparison with the result obtained from the MCGM, the $\mathbf{UV}-\mathbf{I}_m$ obtained from the CGM is recorded below:

$$\mathbf{UV}-\mathbf{I}_m =$$
$$\left[\begin{array}{ccccc} -0.358(-2) & -0.356(-2) & -0.350(-2) & -0.315(-2) & -0.172(-2) \\ 0.182(-1) & 0.181(-1) & 0.179(-1) & 0.158(-1) & 0.606(-2) \\ -0.359(-1) & -0.359(-1) & -0.359(-1) & -0.305(-1) & -0.116(-2) \\ 0.303(-1) & 0.306(-1) & 0.315(-1) & 0.237(-1) & -0.271(-1) \\ 0.811(-3) & 0.186(-3) & -0.186(-2) & 0.490(-2) & 0.606(-1) \\ -0.234(-1) & -0.228(-1) & -0.206(-1) & -0.242(-1) & -0.636(-1) \\ 0.200(-1) & 0.196(-1) & 0.184(-1) & 0.195(-1) & 0.370(-1) \\ -0.742(-2) & -0.730(-2) & -0.691(-2) & -0.708(-2) & -0.116(-1) \\ 0.108(-2) & 0.106(-2) & 0.101(-2) & 0.102(-2) & 0.153(-2) \end{array}\right.$$
$$\left.\begin{array}{cccc} 0.239(-2) & 0.118(-1) & 0.302(-1) & 0.623(-1) \\ -0.235(-1) & -0.933(-1) & -0.233 & -0.482 \\ 0.924(-1) & 0.319 & 0.783 & 1.62 \\ -0.197 & -0.620 & -1.50 & -3.10 \\ 0.255 & 0.748 & 1.78 & 3.70 \\ -0.207 & -0.575 & -1.36 & -2.82 \\ 0.103 & 0.276 & 0.646 & 1.34 \\ -0.291(-1) & -0.755(-1) & -0.176 & -0.365 \\ 0.357(-2) & 0.903(-2) & 0.209(-1) & 0.435(-1) \end{array}\right]. \tag{65}$$

Obviously, the CGM provides a poor inversion with a large error 3.7.

Using the same $m = 9$ and $\varepsilon = 10^{-9}$ the MCGM leads to much better $e_1 = 4.14\times10^{-6}$, and $e_2 = 1.26\times10^{-5}$ than those of the CGM, and the acceptable $e_3 = 4.67\times$

10^{-3} and $e_4 = 0.167$, where $\mathbf{UV}-\mathbf{I}_m$ is recorded below:

$$\mathbf{UV}-\mathbf{I}_m =$$

$$\left[\begin{array}{ccccc} -0.100(-8) & 0.144(-7) & -0.533(-7) & 0.972(-7) & -0.107(-6) \\ 0.134(-7) & -0.912(-7) & 0.278(-6) & -0.492(-6) & 0.549(-6) \\ -0.448(-7) & 0.322(-6) & -0.104(-5) & 0.195(-5) & -0.229(-5) \\ 0.747(-7) & -0.553(-6) & 0.181(-5) & -0.342(-5) & 0.405(-5) \\ -0.909(-7) & 0.678(-6) & -0.221(-5) & 0.412(-5) & -0.482(-5) \\ 0.832(-7) & -0.576(-6) & 0.178(-5) & -0.315(-5) & 0.351(-5) \\ -0.325(-7) & 0.220(-6) & -0.691(-6) & 0.128(-5) & -0.149(-5) \\ 0.998(-8) & -0.648(-7) & 0.198(-6) & -0.360(-6) & 0.418(-6) \\ -0.748(-9) & 0.719(-8) & -0.255(-7) & 0.486(-7) & -0.570(-7) \end{array}\right.$$

$$\left.\begin{array}{cccc} 0.783(-7) & -0.380(-7) & 0.111(-7) & -0.148(-8) \\ -0.393(-6) & 0.175(-6) & -0.441(-7) & 0.477(-8) \\ 0.173(-5) & -0.817(-6) & 0.220(-6) & -0.259(-7) \\ -0.307(-5) & 0.146(-5) & -0.395(-6) & 0.467(-7) \\ 0.362(-5) & -0.171(-5) & 0.464(-6) & -0.553(-7) \\ -0.251(-5) & 0.113(-5) & -0.290(-6) & 0.328(-7) \\ 0.111(-5) & -0.514(-6) & 0.135(-6) & -0.156(-7) \\ -0.312(-6) & 0.144(-6) & -0.380(-7) & 0.435(-8) \\ 0.431(-7) & -0.206(-7) & 0.572(-8) & -0.697(-9) \end{array}\right]. \tag{66}$$

From Table 1 it can be seen that the MCGM2 provides the most accurate inversion than other three methods. In this solution we let $\mathbf{x}_1 = \mathbf{x}_0 = \mathbf{1}$ in Eqs. (28) and (29). While the MCGM2(R) means the right-inversion, the MCGM2(L) means the left-inversion. Whether one uses MCGM2(R) or MCGM2(L), the fact is that MCGM2 has a better performance than the MCGM1 for the inversion of an ill-conditioned matrix.

In order to compare the accuracy of inverting the Vandermonde matrices, by using the MCGM1 and MCGM2, we calculate the four error estimations $e_k,\ k = 1,\ldots,4$ in Fig. 3 in the range of $5 \le m \le 30$, where the convergent criteria are $\varepsilon = 10^{-6}$ for the MCGM1 and $\varepsilon = 10^{-5}$ for the MCGM2. From Fig. 3(a) it can be seen that both the MCGM1 and MCGM2 have the similar e_1 and e_2; but as shown in Fig. 3(b) the MCGM2 yields much better results in e_3 and e_4 than the MCGM1. It means that the MCGM2 is much better in finding the inversion of Vandermonde matrix.

Under the same $m = 9$ and $\varepsilon = 10^{-9}$ the MCGM1 leads to a better $e_1 = 1.85\times 10^{-6}$ and $e_2 = 5.89\times 10^{-6}$ than those of the CGM and MCGM as shown in Table 1. This fact indicates that the MCGM1 is more accurate than the CGM and MCGM to solve the linear system (1). For example, we let $x_i = i,\ i = 1,\ldots,9$ be the exact solutions.

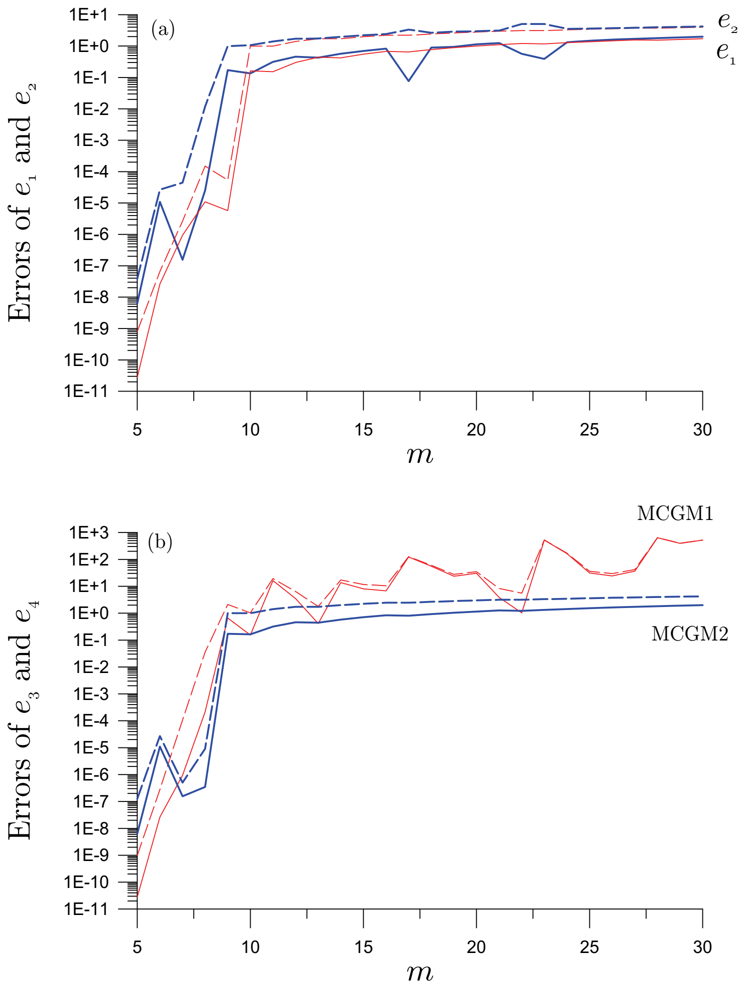

Figure 3: Plotting the errors of (a) e_1 and e_2 and (b) e_3 and e_4 with respect to m for the MCGM1 and MCGM2 applied to the Vandermonde matrices.

Then we solve Eq. (1) with $\mathbf{x} = \mathbf{U}\mathbf{b}_1$ by the MCGM and MCGM1, whose absolute errors are compared with those obtained by the CGM in Table 2. It can be seen that for this ill-posed linear problem, the MCGM and MCGM1 are much better than the CGM.

The above case already revealed the advantages of the MCGM and MCGM1 methods than the CGM. The accuracy of MCGM and MCGM1 is about four to seven orders higher than that of the CGM. Here we have directly used the CGM to find

Table 2: Comparing the numerical errors for a Vandermonde linear system with different methods

Errors of	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
CGM	0.181(-4)	0.573(-4)	0.450(-4)	0.289(-4)	0.423(-4)	0.223(-4)	0.527(-4)	0.278(-4)	0.498(-5)
MCGM	0.182(-11)	0.102(-9)	0.960(-9)	0.157(-8)	0.175(-8)	0.140(-8)	0.146(-8)	0.509(-10)	0.300(-10)
MCGM1	0.382(-10)	0.269(-9)	0.524(-9)	0.116(-8)	0.268(-8)	0.169(-8)	0.640(-9)	0.138(-9)	0.000

the solution of linear system, and not through the CGM to find the inversion of the system matrix. As shown in Eq. (65), if we use the inversion $\mathbf{U}$ of $\mathbf{V}$ to calculate the solution by $\mathbf{x} = \mathbf{U}\mathbf{b}_1$, the numerical results would be much worse than those listed in Table 2 under the item CGM.

Furthermore, we consider a more ill-posed case with $m = 50$, where we let $x_i = i,\ i = 1,\dots,50$ be the exact solutions. In Fig. 4 we compare the absolute errors obtained by the CGM, the MCGM and the MCGM1, which are, respectively, plotted by the dashed-dotted line, solid-line and dashed-line. It can be seen that the accuracy of the MCGM and MCGM1 is much better than the CGM, and the MCGM1 is better than the MCGM, where both the stopping criteria of the MCGM and MCGM1 are set to be $\varepsilon = 10^{-6}$, and that of the CGM is 10^{-10}.

6.2 Hilbert matrices

The Hilbert matrix

$$H_{ij} = \frac{1}{i-1+j} \tag{67}$$

is notoriously ill-conditioned, which can be seen from Table 3 [Liu and Chang (2009)].

Table 3: The condition numbers of Hilbert matrix

m	cond($\mathbf{H}$)	m	cond($\mathbf{H}$)
3	5.24×10^2	7	4.57×10^8
4	1.55×10^4	8	1.53×10^{10}
5	4.77×10^5	9	4.93×10^{11}
6	1.50×10^7	10	1.60×10^{13}

It is known that the condition number of Hilbert matrix grows as $e^{3.5m}$ when m is very large. For the case with $m = 200$ the condition number is extremely huge of the order 10^{348}. The exact inverse of the Hilbert matrix has been derived by Choi

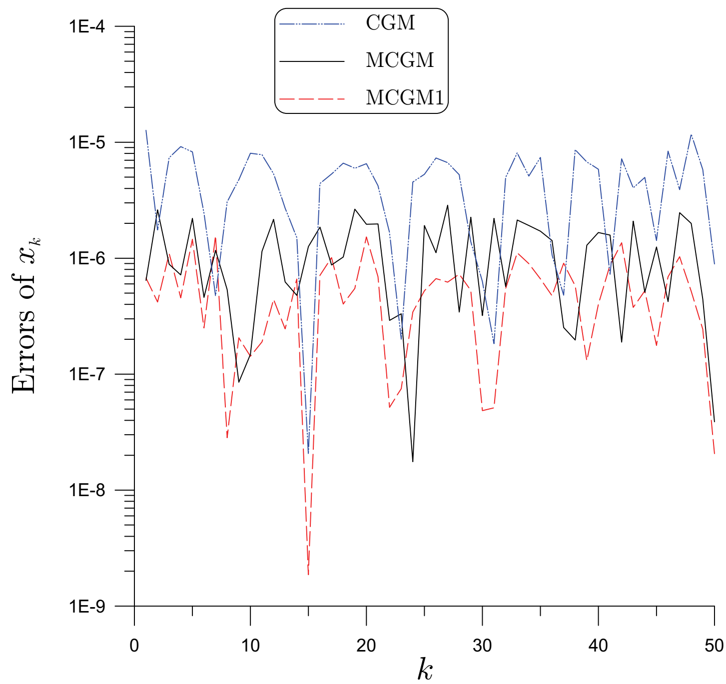

Figure 4: For a Vandermonde linear system with $m = 50$ comparing the numerical errors of the CGM, MCGM and MCGM1.

(1983):

$$(H^{-1})_{ij} = (-1)^{(i+j)}(i+j-1)\begin{pmatrix} m+i-1 \\ m-j \end{pmatrix}\begin{pmatrix} m+j-1 \\ m-i \end{pmatrix}\begin{pmatrix} i+j-2 \\ i-1 \end{pmatrix}^2. \tag{68}$$

Since the exact inverse has large integer entries when m is large, a small perturbation of the given data will be amplified greatly, such that the solution is contaminated seriously by errors. The program can compute the inverse by using the exact integer arithmetic for $m = 13$. Past that number the double precision approximation should be used. However, due to overflow the inverse can be computed only for m which is much smaller than 200.

In order to compare the accuracy of inversion of the Hilbert matrices, by using the MCGM1 and MCGM2, we calculate the four error estimations e_k, $k = 1,\ldots,4$ in Fig. 5 in the range of $5 \le m \le 30$, where the convergent criteria are $\varepsilon = 10^{-7}$ for the MCGM1 and $\varepsilon = 5 \times 10^{-6}$ for the MCGM2. In the MCGM2 we let $\mathbf{x}_0 = \mathbf{1}$, and

$$\mathbf{x}_1 = \left[\mathbf{I}_m - \frac{\|\mathbf{x}_0\|^2}{\mathbf{x}_0^{\mathrm{T}} \mathbf{H} \mathbf{x}_0} \mathbf{H} \right] \mathbf{x}_0.$$

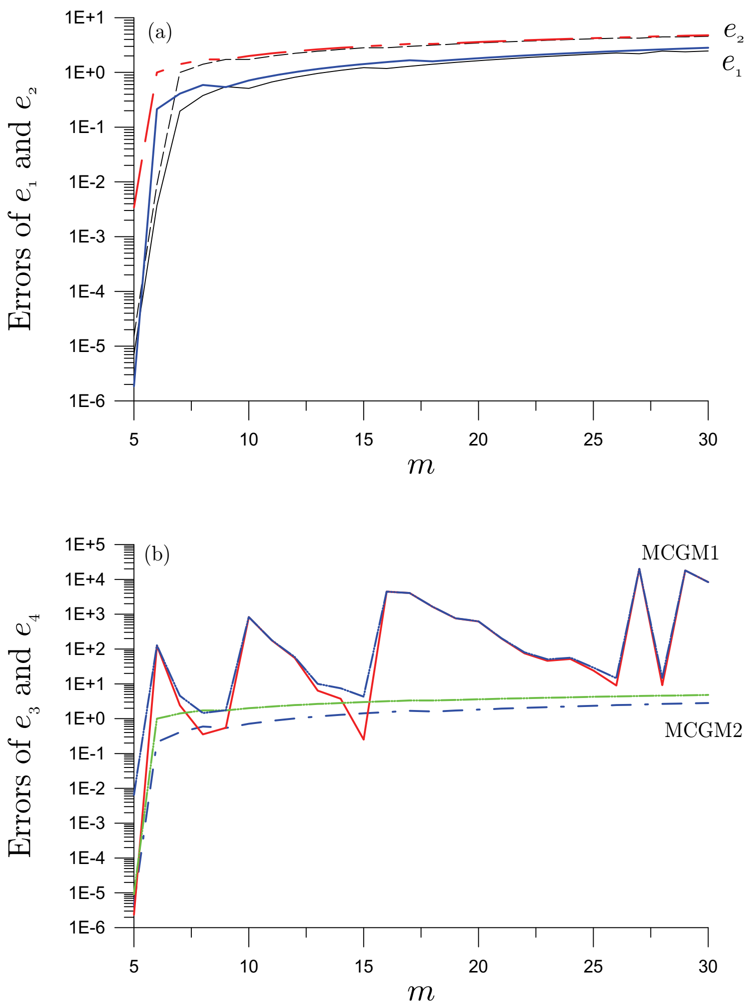

Figure 5: Plotting the errors of (a) e_1 and e_2 and (b) e_3 and e_4 with respect to m for the MCGM1 and MCGM2 applied to the Hilbert matrices.

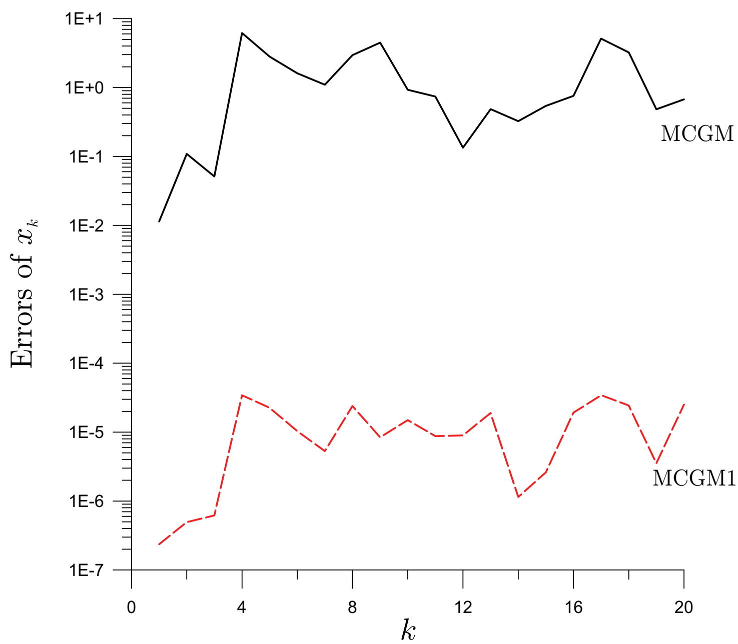

Figure 6: For a Hilbert linear system with $m = 20$ comparing the numerical errors of the MCGM and MCGM1.

From Fig. 5(a) it can be seen that both the MCGM1 and MCGM2 have the similar e_1 and e_2; but as shown in Fig. 5(b) the MCGM2 has much better results in e_3 and e_4 than the MCGM1. It means that the MCGM2 is much better in finding the inversion of Hilbert matrix.

We consider a highly ill-conditioned Hilbert linear system with $m = 20$. Under the same $\varepsilon = 10^{-8}$ the MCGM1 leads to better $(e_1, e_2, e_3, e_4) = (0.414, 3.82, 360276, 360276)$ than those of the MCGM with $(e_1, e_2, e_3, e_4) = (400.72, 400.72, 8.65 \times 10^8, 8.65 \times 10^8)$. This fact indicates that the MCGM1 is more accurate than the MCGM to solve the Hilbert linear system. We let $x_i = 1,\ i = 1, \ldots, 20$ be the exact solutions, and the absolute errors of numerical results are compared in Fig. 6, of which one can see that the MCGM1 is much accurate than the MCGM.

From Table 2, Figs. 3 and 6 it can be seen that the MCGM1 can provide a very accurate solution of $\mathbf{x}$ in terms of $\mathbf{x} = \mathbf{U}\mathbf{b}_1$, because the MCGM1 is a feasible algorithm to finding the left-inversion of ill-conditioned matrix. However, we do not suggest to directly use $\mathbf{x} = \mathbf{U}\mathbf{b}_1$ to find the solution of $\mathbf{x}$, when the data $\mathbf{b}_1$ are noisy. The reason is that the noise in $\mathbf{b}_1$ would be enlarged when the elements in $\mathbf{U}$ are quite large. Then, we apply the Tikhonov regularization with $\alpha = 10^{-5}$, and the presently described regularizarion methods to solve the linear system (1) with the Hilbert matrix, where a random noise with intensity $s = 0.001$ and mean 0.5 is added in the data on the right-hand side. We let $x_i = i,\ i = 1,\dots,20$ be the exact solutions, and the absolute errors of numerical results are compared in Fig. 7, of which one can see that the presently described regularization (NR) in Eq. (32) is more accurate than the Tikhonov regularization (TR). The numerical results are listed in Table 4.

Table 4: Comparing numerical results for a Hilbert linear system under noise

Solutions	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
Exact	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
TR	0.90	3.34	1.58	2.19	3.85	5.76	7.61	9.25	10.67	11.87
NR	1.05	1.43	3.99	4.39	4.68	5.33	6.32	7.52	8.82	10.12
Solutions	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
Exact	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0	19.0	20.0
TR	12.87	13.69	14.35	14.89	15.32	15.66	15.92	16.12	16.26	16.35
NR	11.39	12.57	13.67	14.68	15.58	16.39	17.11	17.75	18.31	18.80

7 Applications of the presently proposed regularization

7.1 Polynomial interpolation

As an application of the new regularization in Eq. (31) we consider a polynomial interpolation. Liu and Atluri (2009a) have solved the ill-posed problem in the high-order polynomial interpolation by using the scaling technique.

Polynomial interpolation is the interpolation of a given set of data by a polynomial. In other words, given some data points, such as obtained by sampling of a measurement, the aim is to find a polynomial which goes exactly through these points.

Given a set of m data points (x_i, y_i) where no two x_i are the same, one is looking for a polynomial $p(x)$ of degree at most $m-1$ with the following property:

$$p(x_i) = y_i,\ \ i = 1,\dots,m, \tag{69}$$

where $x_i \in [a,b]$, and $[a,b]$ is a spatial interval of our problem domain.

The unisolvence theorem states that such a polynomial $p(x)$ exists and is unique, and can be proved by using the Vandermonde matrix. Suppose that the interpolation polynomial is in the form of

$$p(x) = \sum_{i=1}^{m} a_i x^{i-1}, \tag{70}$$

where x^i constitute a monomial basis. The statement that $p(x)$ interpolates the data points means that Eq. (69) must hold.

If we substitute Eq. (70) into Eq. (69), we can obtain a system of linear equations in the coefficients a_i, which in a matrix-vector form reads as

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{m-2} & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{m-2} & x_2^{m-1} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & x_{m-1} & x_{m-1}^2 & \dots & x_{m-1}^{m-2} & x_{m-1}^{m-1} \\ 1 & x_m & x_m^2 & \dots & x_m^{m-2} & x_m^{m-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n-1} \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix}. \tag{71}$$

We have to solve the above system for a_i to construct the interpolant $p(x)$. As suggested by Liu and Atluri (2009a) we use a scaling factor R_0 in the coefficients $b_i = a_i R_0^{i-1}$ to find b_i and then a_i. In view of Eq. (30), the above is a dual system with $\mathbf{V}$ defined by Eq. (64).

The Runge phenomenon illustrates that the error can occur when constructing a polynomial interpolant of higher degree [Quarteroni, Sacco and Saleri (2000)]. The function to be interpolated is

$$f(x) = \frac{1}{1+25x^2}, \quad x \in [-1,1]. \tag{72}$$

We apply the regularization technique in Section 5 by solving Eq. (30), which is regularized by Eq. (31), to obtain $b_i = a_i R_0^{i-1}$, where $R_0 = 1.2$, and then a_i are inserted into the interpolant in Eq. (70) to solve this problem.

Under a random noise $s = 0.01$ on the data $\mathbf{b}_1$ we take $\mathbf{x}_0$ perpendicular to $\mathbf{b}_1$ by

$$\mathbf{x}_0 = \left[\mathbf{I}_m - \frac{\|\mathbf{b}_1\|^2}{\mathbf{b}_1^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}\mathbf{b}_1}\mathbf{V}^{\mathrm{T}}\right]\mathbf{b}_1. \tag{73}$$

In Fig. 8(a) we compare the exact function with the interpolated polynomial. Although m is large up to 120, no oscillation is observed in the interpolant by the novel regularization method, where the interpolated error as shown in Fig. 8(b) is smaller than 0.0192. The CGM used to solve the regularized Eq. (31) is convergent rather fast under $\varepsilon = 10^{-7}$. On the other hand, we also applied the Tikhonov regularization method to calculate this example with $\alpha = 10^{-5}$. However, its result is not good, and the maximal error can be large up to 0.16.

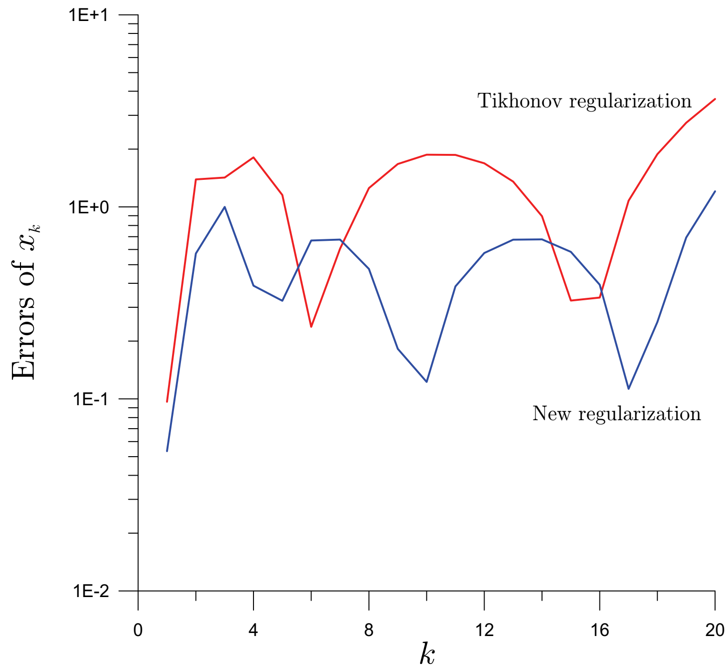

Figure 7: For a Hilbert linear system with $m = 20$ comparing the numerical errors of the Tikhonov regularization and the new regularization in the present paper.

7.2 Best polynomial approximation

The problems with an ill-conditioned $\mathbf{V}$ may appear in several fields. For example, finding an $(m-1)$-order polynomial function $p(x) = a_0 + a_1x + \ldots + a_{m-1}x^{m-1}$ to best match a continuous function $f(x)$ in the interval of $x \in [0,1]$:

$$\min_{\deg(p)\leq m-1} \int_0^1 |f(x)-p(x)|dx, \tag{74}$$

leads to a problem governed by Eq. (1), where $\mathbf{V}$ is the $m \times m$ Hilbert matrix defined by Eq. (67), $\mathbf{x}$ is composed of the m coefficients $a_0, a_1, \ldots, a_{m-1}$ appearing in $p(x)$, and

$$\mathbf{b} = \begin{bmatrix} \int_0^1 f(x)dx \\ \int_0^1 xf(x)dx \\ \vdots \\ \int_0^1 x^{m-1}f(x)dx \end{bmatrix} \tag{75}$$

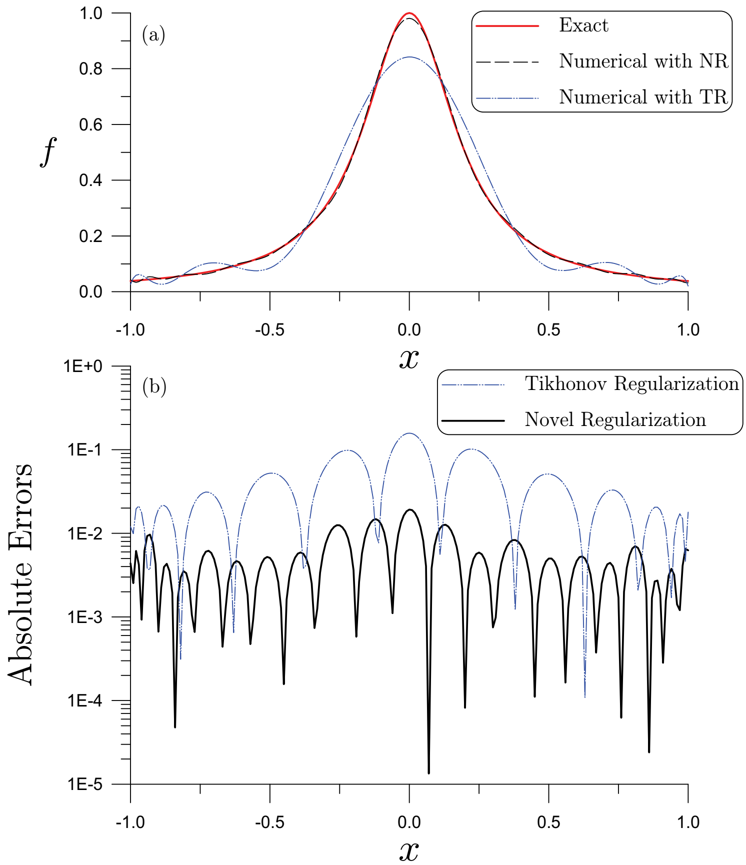

Figure 8: (a) Comparing the exact function and the polynomial interpolant calculated by the novel regularization (NR) method and the Tikhonov regularization (TR) method, and (b) the numerical errors.

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

Encouraged by the above well-conditioning behavior of the Hilbert linear system after the presently proposed regularization, now, we are ready to solve this very difficult problem of a best approximation of the function e^x by an $(m-1)$-order polynomial. We compare the exact solution e^x with the numerical solutions without noise and with a random noise $s = 0.001$ with zero mean in Fig. 9(a), where $m = 12$ and $m = 3$ were used, respectively. The absolute errors are also shown in Fig. 9(b). The results are rather good. The present results are better than those in Liu, Yeih and Atluri (2009), which are calculated by the preconditioning technique.

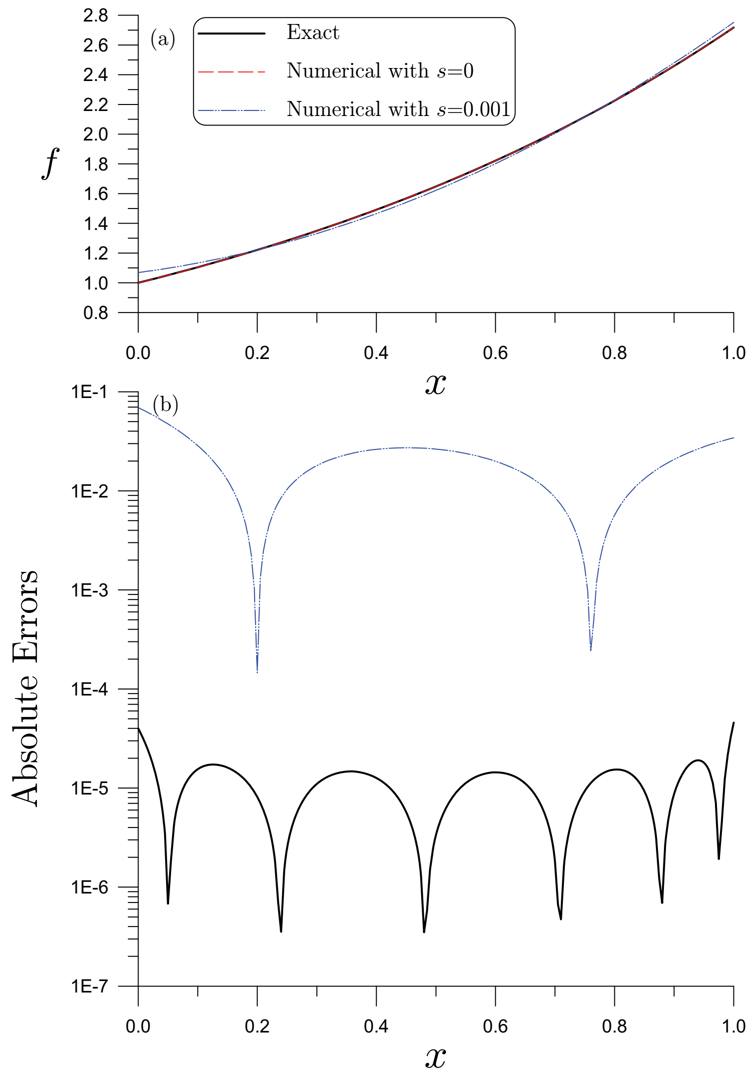

Figure 9: (a) Comparing the exact function and the best polynomial approximation calculated by the new regularization method, and (b) the numerical errors.

8 Conclusions

We have proposed a matrix conjugate gradient method (MCGM) to directly invert ill-conditioned matrices. Two novel algorithms MCGM1 and MCGM2 were developed in this paper, for the first time, to find the inversion of **V**, which can overcome the ill-posedness of severely ill-conditioned matrices appearing in linear equations: $\mathbf{Vx} = \mathbf{b}_1$. By adding two compatible vector equations into the matrix equations, we obtained an over-determined system for the inversion of an ill-conditioned matrix. The solution is then a least-squares one, which can relax the ill-posedness of ill-conditioned matrices. Eqs. (28) and (29) constitute a regularized pair of dual and primal systems of matrix equations for the two-sided inversions of an ill-conditioned matrix. When $\mathbf{V}$ is a non-symmetric matrix we can let $\mathbf{x}_1 = \mathbf{x}_0$; otherwise, $\mathbf{x}_1$ must be different from $\mathbf{x}_0$. Thus, the MCGM1 can provide an accurate solution of $\mathbf{x}$ by $\mathbf{x} = \mathbf{U}\mathbf{b}_1$, when there exists no noise on the data of $\mathbf{b}_1$. In contrast to the Tikhonov regularization, we have projected the regularized matrix equation into the vector space of linear equations, and obtained a novel vector regularization method for the ill-posed linear system. In this new theory, there exists a feasible generalization from the scalar regularization parameter α for the Tikhonov regularization technique to a broad vector regularization parameter $\mathbf{y}_0 = \mathbf{V}\mathbf{x}_0$ or $\mathbf{y}_0 = \mathbf{V}^{\mathrm{T}}\mathbf{x}_0$ for a novel regularization technique presented in this paper. Through some numerical tests of the Vandermonde and Hilbert linear systems we found that the presently proposed algorithms converge rather fast, even for the highly ill-posed linear problems. This situation is quite similar to the CGM for the well-posed linear problems. The new algorithms have better computational efficiency and accuracy, which may be applicable to many engineering linear problems with ill-posedness.

References

Atluri, S. N. (2004): The Meshless Method (MLPG) for Domain & BIE Discretizations. Tech Science Press.

Atluri, S. N. (2005): Methods of Computer Modeling in Engineering & the Sciences. Tech Science Press.

Atluri, S. N.; Kim, H. G.; Cho, J. Y. (1999): A critical assessment of the truly meshless local Petrov-Galerkin (MLPG), and local boundary integral equation (LBIE) methods. *Comp. Mech.*, vol. 24, pp. 348-372.

Atluri, S. N.; Shen, S. (2002): The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite element and boundary element methods. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 11-51.

Atluri, S. N.; Zhu, T. L. (1998): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comp. Mech.*, vol. 22, pp. 117-127.

Björck, A.; Pereyra, V. (1970): Solution of Vandermonde systems of equations. *Math. Comput.*, vol. 24, pp. 893-903.

Calvetti, D.; Reichel, L. (1993): Fast inversion of Vanderomnde-like matrices involving orthogonal polynomials. *BIT*, vol. 33, pp. 473-484.

Choi, M. D. (1983): Tricks or treats with the Hilbert matrix. *Amer. Math. Monthly*, vol. 90, pp. 301-312.

Eisinberg, A.; Fedele, G. (2006): On the inversion of the Vandermonde matrix. *Appl. Math. Comp.*, vol. 174, pp. 1384-1397.

El-Mikkawy, M. E. A. (2003): Explicit inverse of a generalized Vandermonde matrix. *Appl. Math. Comp.*, vol. 146, pp. 643-651.

Fairweather, G.; Karageorghis, A. (1998): The method of fundamental solutions for elliptic boundary value problems. *Adv. Comp. Math.*, vol. 9, pp. 69-95.

Gautschi, W. (1975): Norm estimates for inverses of Vandermonde matrices. *Num. Math.*, vol. 23, pp. 337-347.

Gohberg, I.; Olshevsky, V. (1997): The fast generalized Parker-Traub algorithm for inversion of Vandermonde and related matrices. *J. Complexity*, vol. 13, pp. 208-234.

Han, P. S.; Olson, M. D. (1987): An adaptive boundary element method. *Int. J. Num. Meth. Eng.*, vol. 24, pp. 1187-1202.

Hansen, P. C. (1992): Analysis of discrete ill-posed problems by means of the L-curve. *SIAM Rev.*, vol. 34, pp. 561-580.

Hansen, P. C.; O'Leary, D. P. (1993): The use of the L-curve in the regularization of discrete ill-posed problems. *SIAM J. Sci. Comput.*, vol. 14, pp. 1487-1503.

Higham, N. J. (1987): Error analysis of the Björck-Pereyra algorithms for solving Vandermonde systems. *Num. Math.*, vol. 50, pp. 613-632.

Higham, N. J. (1988): Fast solution of Vandermonde-like systems involving orthogonal polynomials. *IMA J. Num. Anal.*, vol. 8, pp. 473-486.

Jog, C. S. (2004): The accurate inversion of Vandermonde matrices. *Comp. Math. Appl.*, vol. 47, pp. 921-929.

Karlis, G. F.; Tsinopoulos, S. V.; Polyzos, D.; Beskos, D. E. (2008): 2D and 3D boundary element analysis of mode-I cracks in gradient elasticity. *CMES: Computer Modeling in Engineering & Sciences*, vol. 26, pp. 189-207.

Kunisch, K.; Zou, J. (1998): Iterative choices of regularization parameters in linear inverse problems. *Inverse Problems*, vol. 14, pp. 1247-1264.

Liu, C.-S. (2008): 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.; Atluri, S. N. (2009a): 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. (2009b): 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.; 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.; 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.

Neagoe, V. E. (1996): Inversion of the Van der Monde matrix. *IEEE Sign. Proce. Lett.*, vol. 3, pp. 119-120.

Quarteroni, A.; Sacco, R.; Saleri, F. (2000): Numerical Mathematics. Springer-Verlag, Berlin.

Resmerita, E. (2005): Regularization of ill-posed problems in Banach spaces: convergence rates. *Inverse Problems*, vol. 21, pp. 1303-1314.

Skrzipek, M. R. (2003): Inversion of Vandermonde-like matrices. *BIT*, vol. 44, pp. 291-306.

Tang, Z.; Shen, S.; Atluri, S. N. (2003): Analysis of materials with strain-gradient effects: a meshless local Petrov-Galerkin (MLPG) approach, with nodal displacements only. *CMES: Computer Modeling in Engineering & Sciences*, vol. 4, pp. 177-196.

Tikhonov, A. N.; Arsenin, V. Y. (1977): Solutions of Ill-Posed Problems. John-Wiley & Sons, New York.

Tsai, C. C.; Lin, Y. C.; Young, D. L.; Atluri, S. N. (2006): Investigations on the accuracy and condition number for the method of fundamental solutions. *CMES: Computer Modeling in Engineering and Sciences*, vol. 16, pp. 103-114.

Wang, Y.; Xiao, T. (2001): Fast realization algorithms for determining regularization parameters in linear inverse problems. *Inverse Problems*, vol. 17, pp. 281-291.

Wen, P. H.; Aliabadi, M. H.; Young, A. (2002): Boundary element analysis of curved cracked panels with mechanically fastened repair patches. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 1-10.

Wertz, H. J. (1965): On the numerical inversion of a recurrent problem: the Vandermonde matrix. *IEEE Trans. Auto. Control*, vol. AC-10, p. 492.

Xie, J.; Zou, J. (2002): An improved model function method for choosing regularization parameters in linear inverse problems. *Inverse Problems*, vol. 18, pp. 631-643.

Young, D. L.; Ruan J. W. (2005): Method of fundamental solutions for scattering problems of electromagnetic waves. *CMES: Computer Modeling in Engineering & Sciences*, vol. 7, pp. 223-232.

Young, D. L.; Tsai, C. C.; Lin, Y. C.; Chen, C. S. (2006): The method of fundamental solutions for eigenfrequencies of plate vibrations. *CMC: Computers, Materials & Continua*, vol. 4, pp. 1-10.