Available online at www.ispacs.com/cna
Volume 2012, Year 2012 Article ID cna-00115, 24 pages
doi: 10.5899/2012/cna-00115
Research Article

Modifications of Steepest Descent Method and Conjugate Gradient Method Against Noise for Ill-posed Linear Systems

Chein-Shan Liu *
Department of Civil Engineering, National Taiwan University, Taipei, Taiwan

Abstract
It is well known that the numerical algorithms of the steepest descent method (SDM), and the conjugate gradient method (CGM) are effective for solving well-posed linear systems. However, they are vulnerable to noisy disturbance for solving ill-posed linear systems. We propose the modifications of SDM and CGM, namely the modified steepest descent method (MSDM), and the modified conjugate gradient method (MCGM). The starting point is an invariant manifold defined in terms of a minimum functional and a fictitious time-like variable; however, in the final stage we can derive a purely iterative algorithm including an acceleration parameter. Through the Hopf bifurcation, this parameter indeed plays a major role to switch the situation of slow convergence to a new situation that the functional is stepwisely decreased very fast. Several numerical examples are examined and compared with exact solutions, revealing that the new algorithms of MSDM and MCGM have good computational efficiency and accuracy, even for the highly ill-conditioned linear equations system with a large noise being imposed on the given data.
Keywords: Ill-posed linear equations; Invariant manifold; Modified steepest descent method (MSDM); Modified conjugate gradient method (MCGM).

1 Introduction

In this paper we propose a robust and easily-implemented method to solve the following linear equations system:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{1.1}$$

*Corresponding author. Email address: liucs@ntu.edu.tw, Tel: 886-2-33664233

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a given positive definite matrix, and $\mathbf{x} \in \mathbb{R}^n$ is an unknown vector. The input data of $\mathbf{b} \in \mathbb{R}^n$ may be corrupted by noise. Therefore, we may encounter the problem that the numerical solution of Eq. (1.1) may deviate from the exact one to a great extent, when $\mathbf{A}$ is severely ill-conditioned and $\mathbf{b}$ is perturbed by noise.

The solution of ill-posed linear equations is an important issue for many scientific and engineering problems. A good numerical method to solve Eq. (1.1) might be beneficial in the applications to the optimization problems including linear programming and nonlinear programming, Newton's, Quasi-Newton's and homotopy methods for nonlinear equations system, finite difference and finite element methods for partial differential equations, etc.

To account of the sensitivity to noise it is usually using a regularization method to solve the ill-posed problem [8, 24, 25, 21], where a suitable regularization parameter is used to depress the bias in the computed solution by a better balance of approximation error and propagated data error. There are several methods developed after the pioneering work of Tikhonov and Arsenin [23]. Previously, the author and his coworkers have developed several methods to solve the ill-posed linear problems, like that using the fictitious time integration method as a filter for ill-posed linear system [12], a modified polynomial expansion method [13], the nonstandard group preserving scheme [15], a vector regularization method [16], the relaxed steepest descent method [11], as well as the optimal iterative algorithm [14].

A measure of the ill-posedness of Eq. (1.1) can be performed by using the condition number [22]:

$$\mathrm{cond}(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|, \tag{1.2}$$

where $\|\mathbf{A}\|$ is the Frobenius norm of $\mathbf{A}$. For arbitrary $\epsilon > 0$, there exists a matrix norm $\|\mathbf{A}\|$ such that $\rho(\mathbf{A}) \le \|\mathbf{A}\| \le \rho(\mathbf{A}) + \epsilon$, where $\rho(\mathbf{A})$ is a radius of the spectrum of $\mathbf{A}$. Therefore, the condition number of $\mathbf{A}$ can be estimated by

$$\mathrm{cond}(\mathbf{A}) = \frac{\max_{\sigma(\mathbf{A})} |\lambda|}{\min_{\sigma(\mathbf{A})} |\lambda|}, \tag{1.3}$$

where $\sigma(\mathbf{A})$ is the collection of all the eigenvalues of $\mathbf{A}$.

Roughly speaking, the numerical solution of Eq. (1.1) may lose the accuracy of k decimal points when $\mathrm{cond}(\mathbf{A}) = 10^k$. The problems with ill-conditioned $\mathbf{A}$ may appear in several fields. For example, finding an n-degree 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 $x \in [0, 1]$:

$$\min_{\deg(p)\le n} \int_0^1 [f(x) - p(x)]^2 dx, \tag{1.4}$$

leads to a problem governed by Eq. (1.1); $\mathbf{A}$ is the $(n+1)\times(n+1)$ Hilbert matrix defined by

$$A_{ij} = \frac{1}{i+j-1}, \tag{1.5}$$

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

$$\mathbf{b} = \begin{bmatrix} \int_0^1 f(x)dx \\ \int_0^1 xf(x)dx \\ \vdots \\ \int_0^1 x^n f(x)dx \end{bmatrix} \tag{1.6}$$

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

The Hilbert matrix is a famous example of highly ill-conditioned matrices. Eq. (1.1) with the coefficient matrix $\mathbf{A}$ having a large condition number usually displays that an arbitrarily small perturbation on the right-hand side will lead to an arbitrarily large perturbation to the solution on the left-hand side.

2 The ODEs on an invariant manifold

2.1 The steepest descent and conjugate gradient methods

Solving Eq. (1.1) by the steepest descent method [7] is equivalent to solve the following minimum problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \varphi(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^n} \left[\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x}\right]. \tag{2.7}$$

Then, by using the Ritz variational principle we can derive the following algorithm:
(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$.
(ii) For $k = 0, 1, 2 \ldots$, repeat the following computations:

$$\eta_k = \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^{\mathrm{T}}\mathbf{A}\mathbf{r}_k}, \tag{2.8}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{r}_k, \tag{2.9}$$

$$\mathbf{r}_{k+1} = \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}. \tag{2.10}$$

If $\|\mathbf{r}_{k+1}\| < \varepsilon$ for a prescribed convergence criterion ε then stop; otherwise, go to step (ii).

For the steepest descent method (SDM) the residual vector $\mathbf{r}_k$ is the steepest descent direction of the function φ at the point $\mathbf{x}_k$. But when $\|\mathbf{r}_k\|$ is rather small the calculated $\mathbf{r}_k$ may deviate from the real steepest descent direction to a great extent due to the round-off error of computing machine, which usually leads to the numerical instability of SDM.

An improvement of SDM is the conjugate gradient method (CGM), which enhances the search direction of the minimum by imposing the orthogonality of the residual vector at each iterative step [7]. The algorithm of the CGM can be summarized as follows:
(i) Give an initial $\mathbf{x}_0$.
(ii) Calculate $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$ and $\mathbf{p}_1 = \mathbf{r}_0$.
(iii) For $k = 1, 2 \ldots$, repeat the following computations:

$$\eta_k = \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}, \tag{2.11}$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \eta_k \mathbf{p}_k, \tag{2.12}$$

$$\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}, \tag{2.13}$$

$$\alpha_k = \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \tag{2.14}$$

$$\mathbf{p}_{k+1} = \alpha_k \mathbf{p}_k + \mathbf{r}_k. \tag{2.15}$$

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

2.2 The steplength in the steepest descent method

In addition to the CGM, several modifications to the SDM have been recurred in order to accelerate the convergence speed. The modifications even somewhat ad hoc have led to a new interest in the SDM, that the gradient vector itself is not a bad choice but rather that the original steplength leads to the slow convergence behavior. Barzilai and Borwein [1] were the first, who presented a new choice of steplength through two-point stepsize. Although their method did not guarantee the descent of the minimum functional values, Barzilai and Borwein [1] were able to produce a substantial improvement of the convergence speed for a certain test. The results of Barzilai and Borwein [1] have initiated many researches on the SDM, for example, Raydan [18, 19], Friedlander et al. [6], Raydan and Svaiter [20], Dai et al. [3], Dai and Liao [2], Dai and Yuan [4], Fletcher [5], and Yuan [26]. In this paper we will approach this problem from a quite different aspect of invariant manifold and bifurcation, and propose a new strategy to modify the steplength. We also compare our results with the following random SDM proposed by Raydan and Svaiter [20]:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \theta_k \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^{\mathrm{T}} \mathbf{A} \mathbf{r}_k} \mathbf{r}_k, \tag{2.16}$$

where θ_k are random numbers in $[0, 2]$. We will show that $\theta_k \in [0, 1]$ is a better choice from the viewpoint of invariant manifold. We also demonstrate that the newly modified SDM is performed better than the random SDM and the Barzilai-Borwein method (BBM):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{(\Delta \mathbf{r}_{k-1})^{\mathrm{T}} \Delta \mathbf{x}_{k-1}}{\|\Delta \mathbf{r}_{k-1}\|^2} \mathbf{r}_k, \tag{2.17}$$

where $\Delta \mathbf{r}_{k-1} = \mathbf{r}_k - \mathbf{r}_{k-1}$, and $\Delta \mathbf{x}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$. Initially, we can set $\mathbf{r}_0 = \mathbf{0}$ and $\mathbf{x}_0 = \mathbf{0}$.

2.3 An invariant manifold

From Eqs. (2.7) and (1.1) it is easy to prove that the minimum is

$$\min_{\mathbf{x} \in \mathbb{R}^n} \varphi(\mathbf{x}) = \varphi(\mathbf{x}^*) = -\frac{1}{2} \mathbf{x}^{*\mathrm{T}} \mathbf{A} \mathbf{x}^* < 0, \tag{2.18}$$

where $\mathbf{x}^*$ is a solution of Eq. (1.1).

We can take a different level set function from $\varphi(\mathbf{x})$ by

$$\phi(\mathbf{x}) = \varphi(\mathbf{x}) + c_0 = \frac{1}{2} \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x} - \mathbf{b}^{\mathrm{T}} \mathbf{x} + c_0, \tag{2.19}$$

where c_0 is a constant such that $\phi \geq 0$. Of course the minima of $\phi(\mathbf{x})$ and $\varphi(\mathbf{x})$ are happened at the same point $\mathbf{x} = \mathbf{x}^*$.

There are several regularization methods to deal with Eq. (1.1) when $\mathbf{A}$ is ill-conditioned. In this paper we consider an iterative regularization method for Eq. (1.1) by investigating an evolutional behavior of $\mathbf{x}$ from the ODEs defined on an invariant manifold, which is formed from $\phi(\mathbf{x})$:

$$h(\mathbf{x}, t) := Q(t) \phi(\mathbf{x}) = C. \tag{2.20}$$

Here, we let $\mathbf{x}$ be a function of a fictitious time-like variable t. We do not need to specify the function $Q(t)$ a priori, of which $C/Q(t)$ merely acts as a measure of the decreasing of

ϕ in time. Hence, we expect that in our algorithm $Q(t) > 0$ is an increasing function of t. We let $Q(0) = 1$, and C is determined from the initial condition $\mathbf{x}(0) = \mathbf{x}_0$ by

$$C = \phi(\mathbf{x}_0) > 0. \tag{2.21}$$

When $C > 0$ and $Q > 0$, the manifold defined by Eq. (2.20) is continuous and differentiable, and thus the following differential operation carried out on the manifold makes sense. For the requirement of consistency condition, by taking the time differential of Eq. (2.20) with respect to t and considering $\mathbf{x} = \mathbf{x}(t)$, we have

$$\dot{Q}(t)\phi(\mathbf{x}) + Q(t)(\mathbf{A}\mathbf{x} - \mathbf{b}) \cdot \dot{\mathbf{x}} = 0. \tag{2.22}$$

We suppose that $\mathbf{x}$ is governed by a gradient-flow:

$$\dot{\mathbf{x}} = -\lambda \frac{\partial \phi}{\partial \mathbf{x}} = -\lambda(\mathbf{A}\mathbf{x} - \mathbf{b}), \tag{2.23}$$

where λ is to be determined. Inserting Eq. (2.23) into Eq. (2.22) we can solve

$$\lambda = \frac{q(t)\phi}{\|\mathbf{r}\|^2}, \tag{2.24}$$

where

$$\mathbf{r} := \mathbf{A}\mathbf{x} - \mathbf{b}, \tag{2.25}$$

$$q(t) := \frac{\dot{Q}(t)}{Q(t)}. \tag{2.26}$$

Here $\mathbf{r}$ signifies the residual vector.

Thus, inserting Eq. (2.24) into Eq. (2.23) we obtain an evolution equation for $\mathbf{x}$ defined by a gradient-flow:

$$\dot{\mathbf{x}} = -q(t) \frac{\phi}{\|\mathbf{r}\|^2} \mathbf{r}. \tag{2.27}$$

In the present algorithm if $Q(t)$ can be guaranteed to be an increasing function of t, we may have an absolutely convergent property in solving Eq. (1.1) through the search of the minimum of ϕ by the following equation:

$$\phi(t) = \frac{C}{Q(t)}. \tag{2.28}$$

When t is quite large the above equation can enforce the functional ϕ tending to its minimum, and meanwhile the solution of Eq. (1.1) is obtained.

3 Dynamics of iterative algorithms

3.1 Discretizing, yet keeping x on the manifold

By applying the Euler method to Eq. (2.27) we can obtain the following algorithm:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \beta \frac{\phi}{\|\mathbf{r}\|^2} \mathbf{r}, \tag{3.29}$$

where

$$\beta = q(t)\Delta t. \tag{3.30}$$

In order to keep $\mathbf{x}$ on the manifold defined by Eq. (2.28) we can insert the above $\mathbf{x}(t+\Delta t)$ into $\phi(\mathbf{x}(t+\Delta t)) = C/Q(t+\Delta t)$, i.e.,

$$\frac{1}{2}\mathbf{x}^{\mathrm{T}}(t+\Delta t)\mathbf{A}\mathbf{x}(t+\Delta t) - \mathbf{b}^{\mathrm{T}}\mathbf{x}(t+\Delta t) + c_0 = \frac{C}{Q(t+\Delta t)}, \tag{3.31}$$

obtaining

$$\begin{aligned}\frac{C}{Q(t+\Delta t)} - c_0 &= \frac{1}{2}\mathbf{x}^{\mathrm{T}}(t)\mathbf{A}\mathbf{x}(t) - \mathbf{b}^{\mathrm{T}}\mathbf{x}(t) \\ &+ \beta\phi\frac{[\mathbf{b}-\mathbf{A}\mathbf{x}(t)]^{\mathrm{T}}\mathbf{r}}{\|\mathbf{r}\|^2} + \beta^2\phi^2\frac{\mathbf{r}^{\mathrm{T}}\mathbf{A}\mathbf{r}}{2\|\mathbf{r}\|^4}.\end{aligned} \tag{3.32}$$

Thus by Eqs. (2.25), (2.28) and (2.19) and through some manipulations we can derive the following scalar equation:

$$\frac{1}{2}a_0\beta^2 - \beta + 1 = \frac{Q(t)}{Q(t+\Delta t)}, \tag{3.33}$$

where

$$a_0 := \frac{\phi\mathbf{r}^{\mathrm{T}}\mathbf{A}\mathbf{r}}{\|\mathbf{r}\|^4}. \tag{3.34}$$

3.2 A trial dynamics

Based-on Eq. (3.33) to enforce the orbit of $\mathbf{x}$ being constrained by the manifold but without a careful judgement we may encounter a big trouble as specified below.

From the approximation of

$$Q(t+\Delta t) = Q(t) + \dot{Q}(t)\Delta t, \tag{3.35}$$

dividing by $Q(t)$, and by Eqs. (2.26) and (3.30) we have

$$\frac{Q(t)}{Q(t+\Delta t)} = \frac{1}{1+\beta}. \tag{3.36}$$

Inserting it into Eq. (3.33) we come to a cubic equation for β:

$$a_0\beta^2(1+\beta) - 2\beta(1+\beta) + 2(1+\beta) = 2, \tag{3.37}$$

where $\beta = 0$ is a double root, and which allows a non-zero solution of β:

$$\beta = \frac{2}{a_0} - 1. \tag{3.38}$$

Inserting the above β into Eq. (3.29) we can obtain

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \left[\frac{2}{a_0} - 1\right]\frac{\phi}{\|\mathbf{r}\|^2}\mathbf{r}. \tag{3.39}$$

However, this algorithm has an unfortunate fate that when a_0 grows from a small number to two, the algorithm will stagnate at a point which is not necessarily a solution. In the below we should avoid to follow this algorithm, and have to develop a better algorithm.

3.3 A better dynamics

The above derivation hints us that we must abandon the concept of keeping the orbit of $\mathbf{x}$ on the manifold with $Q(t)$ specified a priori by Eq. (3.36); otherwise, we only have an unuseful algorithm.

Let $s = Q(t)/Q(t+\Delta t)$. By Eq. (3.33) we can derive

$$\frac{1}{2}a_0\beta^2 - \beta + 1 - s = 0. \tag{3.40}$$

From Eq. (3.40), we can take the solution of β to be

$$\beta = \frac{1-\sqrt{1-2(1-s)a_0}}{a_0}, \quad \text{if } 1-2(1-s)a_0 \geq 0. \tag{3.41}$$

Let

$$1-2(1-s)a_0 = \gamma^2 \geq 0, \quad s = 1 - \frac{1-\gamma^2}{2a_0}; \tag{3.42}$$

such that the condition $1-2(1-s)a_0 \geq 0$ in Eq. (3.41) is automatically satisfied. Thus we have

$$\beta = \frac{1-\gamma}{a_0}. \tag{3.43}$$

Here $0 \leq \gamma < 1$ is a parameter. Inserting Eq. (3.43) for β into Eq. (3.29) and using Eq. (3.34) we obtain a new algorithm:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - (1-\gamma)\frac{\|\mathbf{r}(t)\|^2}{\mathbf{r}^{\mathrm{T}}(t)\mathbf{A}\mathbf{r}(t)}\mathbf{r}(t). \tag{3.44}$$

Remark 3.1. *It is known that in the SDM, we search the next* $\mathbf{x}(t+\Delta t)$ *from* $\mathbf{x}(t)$ *by minimizing the functional* φ *along the direction* $-\mathbf{r}(t)$*, i.e.,*

$$\min_{\alpha} \varphi(\mathbf{x}(t) - \alpha\mathbf{r}(t)). \tag{3.45}$$

Through some calculations we can obtain

$$\alpha = \frac{\|\mathbf{r}(t)\|^2}{\mathbf{r}^{\mathrm{T}}(t)\mathbf{A}\mathbf{r}(t)}. \tag{3.46}$$

Thus we have the following iterative algorithm:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \frac{\|\mathbf{r}(t)\|^2}{\mathbf{r}^{\mathrm{T}}(t)\mathbf{A}\mathbf{r}(t)}\mathbf{r}(t). \tag{3.47}$$

Similarly, from Eq. (3.40) we can choose β to minimize s, obtaining

$$\beta = \frac{1}{a_0}. \tag{3.48}$$

Inserting it into Eq. (3.29) and using Eq. (3.34) we can derive the same SDM algorithm again as in Eq. (3.47). Below, we will demonstrate that this minimization is not the best choice. Instead of, the algorithm in Eq. (3.44) will be better.

3.4 Two novel algorithms

Let $\mathbf{x}_k$ denote the numerical value of $\mathbf{x}$ at the k-th step. Thus, we can arrive to a purely iterative algorithm from Eq. (3.44):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^{\mathrm{T}} \mathbf{A} \mathbf{r}_k} \mathbf{r}_k, \tag{3.49}$$

where

$$\eta = 1 - \gamma. \tag{3.50}$$

Consequently, a modification of the SDM, namely a modified steepest descent method (MSDM) is available as follows:
(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$.
(ii) For $k = 0, 1, 2 \ldots$, repeat the following computations:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma) \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^{\mathrm{T}} \mathbf{A} \mathbf{r}_k} \mathbf{r}_k, \tag{3.51}$$

$$\mathbf{r}_{k+1} = \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}. \tag{3.52}$$

If $\|\mathbf{r}_{k+1}\| < \varepsilon$ then stop; otherwise, go to step (ii).

The above $0 \le \gamma < 1$ is a parameter determined by the user. If $\gamma = 0$ the present algorithm is reduced to the steepest descent method (SDM).

By the same token we can propose a newly modified algorithm of conjugate gradient method, namely a modified conjugate gradient method (MCGM):
(i) Give an initial $\mathbf{x}_0$.
(ii) Calculate $\mathbf{r}_0 = \mathbf{A}\mathbf{x}_0 - \mathbf{b}$ and $\mathbf{p}_1 = \mathbf{r}_0$.
(iii) For $k = 1, 2 \ldots$, repeat the following computations:

$$\eta_k = (1-\gamma) \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}} \mathbf{A} \mathbf{p}_k}, \tag{3.53}$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} - \eta_k \mathbf{p}_k, \tag{3.54}$$

$$\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}, \tag{3.55}$$

$$\alpha_k = \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \tag{3.56}$$

$$\mathbf{p}_{k+1} = \alpha_k \mathbf{p}_k + \mathbf{r}_k. \tag{3.57}$$

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

When $\gamma = 0$ the above algorithm is reduced to the conjugate gradient method.

4 Numerical examples

In order to assess the performance of the newly developed methods let us investigate the following numerical examples. Some results are compared with those obtained from the steepest descent method (SDM), and the conjugate gradient method (CGM). In order to emphasize the difference of our new algorithms we might call the present modifications as a modified steepest descent method (MSDM), and a modified conjugate gradient method (MCGM).

Example 4.1. *In this example we consider a two-dimensional but highly ill-conditioned linear system:*

$$\begin{bmatrix} 2 & 6 \\ 2 & 6.00001 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8 \\ 8.00001 \end{bmatrix}. \tag{4.58}$$

The condition number of this system is $cond(\mathbf{A}) = 1.59 \times 10^{13}$*, where* $\mathbf{A} = \mathbf{B}^T\mathbf{B}$ *and* $\mathbf{B}$ *denotes the coefficient matrix . The exact solution is* $(x, y) = (1, 1)$*.*

Previously, Liu et al. [16] have solved this problem by using a vector regularization method. They obtained a solution of $(x, y) = (1.00005, 1.00005)$ *when a random noise 0.01 is added on the data of* $(8, 8.00001)^T$*. No matter what regularization parameter is used in the Tikhonov regularization method for the above equation, they found that an incorrect solution of* $(x, y) = (1356.4, -450.8)$ *is obtained by the Tikhonov regularization method.*

Now we fix the noise to be 0.01, $\varepsilon = 10^{-9}$ *and starting from an initial condition* $(x_0, y_0) = (0.8, 0.5)$*. The Barzilai-Borwein method (BBM) does not converge with 500 iterations, and obtains an incorrect solution of* $(x, y) = (415.8, -137.3)$*. The residual error of BBM is shown in Figure 1(a). The SDM led to the same result, whose residual error is shown in Figure 1(b). Then we apply the MSDM to this problem by taking* $\gamma = 0.05$*, which led to an approximate solution of* $(x, y) = (0.9702, 1.01)$*. The residual error of MSDM is shown in Figure 1(c). The MSDM is quite different from the BBM and SDM. When the BBM and SDM are vulnerable to the disturbance of noise for an ill-posed linear system, the MSDM can work very well against the disturbance of noise.*

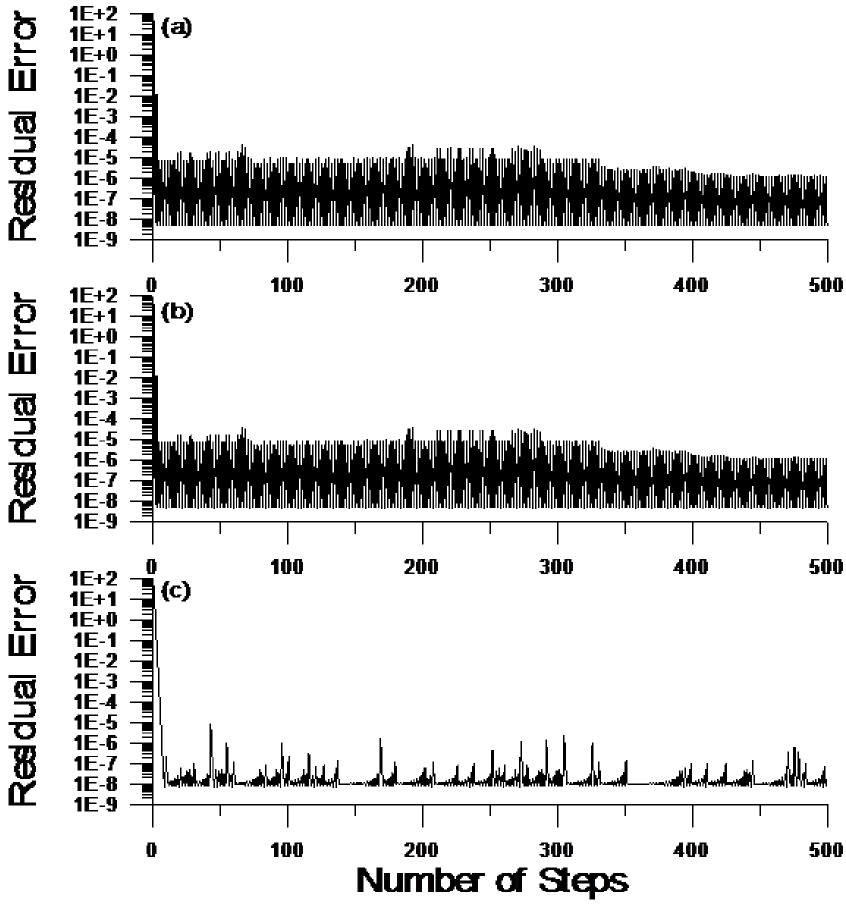

Figure 1: For Example 4.1 comparing the residual errors for (a) BBM, (b) SDM and (c) MSDM.

Example 4.2. *In this example we consider a highly ill-conditioned linear equations system (1.1) with* $\mathbf{A}$ *given by Eq. (1.5), whose ill-posedness increases very fast with* n.
In order to compare the numerical solutions with exact solution we suppose that $x_1 = x_2 = \ldots = x_n = 1$*, and then by Eq. (1.5) we have*

$$b_i = \sum_{j=1}^{n} \frac{1}{i+j-1} + \sigma R(i), \tag{4.59}$$

where we impose a noise on the data with $R(i)$ *being the random numbers in* $[-1, 1]$.
We solve this problem for the case with $n = 50$*. The resultant linear equations system is highly ill-conditioned, since the condition number is very large up to* 1.1748×10^{19}*. We fix the noise* $\sigma = 10^{-8}$*, which is the maximum noise that the SDM permits. With a stopping criterion* $\varepsilon = 10^{-7}$*, the SDM over 5000 iterations does not converge to the exact solution very accurately, as shown in Figure 2 by the solid line. Conversely, the MSDM with* $\gamma = 0.05$ *converges with 662 iterations, with the numerical error as shown in Figure 2 by the dashed line being much smaller than that of the SDM.*
Now, we explain the parameter γ *appeared in Eq. (3.51). In Figure 3 we compare* a_0*,* s*,* ϕ *and the residual errors for* $\gamma = 0.05$ *and* $\gamma = 0$*. From Figure 3(a) it can be seen that for the case with* $\gamma = 0$*, the values of* a_0 *tend to a constant and keep unchanged. By Eq. (3.34) it means that there exists an attracting set for the iterative orbit of* $\mathbf{x}$ *described by the following manifold:*

$$\frac{\phi \mathbf{r}^T \mathbf{A} \mathbf{r}}{\|\mathbf{r}\|^4} = \textit{Constant}. \tag{4.60}$$

Upon the iterative orbit is approached to this slow manifold, it is slowly to reduce the residual error as shown in Figure 3(d) by the solid line, wherea the ratio of s *is also keeping near to 1 as shown in Figure 3(b) by the solid line. Conversely, for the case* $\gamma = 0.05$*,* a_0 *is no more tending to a constant as shown in Figure 3(a) by the dashed line. Because the iterative orbit is not attracted by a slow manifold, the residual error as shown in Figure 3(d) by the dashed line can be reduced step-by-step, wherea the ratio of* s *is sometimes leaving the value that near to 1 as shown in Figure 3(b) by the dashed line. For the latter case the new algorithm of MSDM can give very accurate numerical solution with the residual error tending to* 10^{-7}*. Thus we can observe that when* γ *varies from zero to a positive value, the iterative dynamics given by Eq. (3.51) undergoes a Hopf bifurcation, like as the ODEs behavior observed by Liu [9, 10]. The original stable slow manifold existent for* $\gamma = 0$ *now becomes a ghost manifold for* $\gamma = 0.05$*, and thus the iterative orbit generated from the algorithm with the case* $\gamma = 0.05$ *does not be attracted by that manifold again, and instead of the intermittency happens, leading to an irregularly jumping behaviors in* a_0 *and in the residual error as shown respectively in Figures 3(a) and 3(d) by the dashed lines. In the scale in Figure 3(c), the difference of* ϕ *for SDM and MSDM is unclear because the resolution is not enough. We compare the* ϕ *for SDM and MSDM in Figure 4 in a finer scale, from which it can be seen that the functional* ϕ *is indeed decreased fast by using the MSDM.*
It is known that the CGM is easily disturbed by noise for an ill-posed linear system. Now we raise the noise to $\sigma = 10^{-4}$*. The stopping criterion is kept to be* $\varepsilon = 10^{-4}$*. In Figure 5 we compare the numerical errors by the solid line for CGM and the dashed line for MCGM with* $\gamma = 0.4$*. Although the CGM is unstable, the MCGM is still applicable for this seriously ill-posed case under a large noise, with the numerical error acceptable, which is smaller*

than 0.0428. If we give an early stopping criterion with $\varepsilon = 10^{-3}$, the CGM converges fast with the maximum error being 0.062 as shown in Figure 5 by the dashed-dotted line.

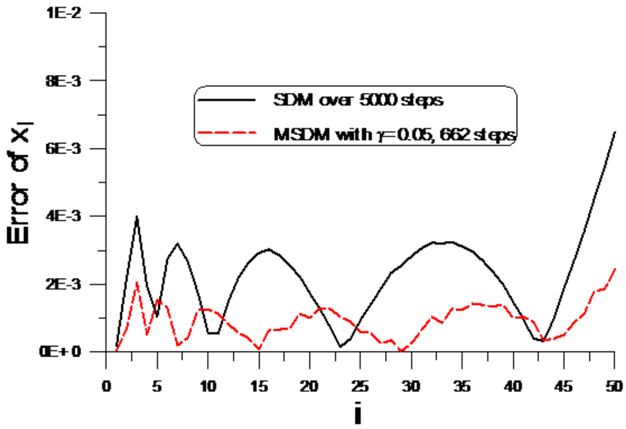

Figure 2: For Example 4.2 comparing numerical errors for SDM and MSDM.

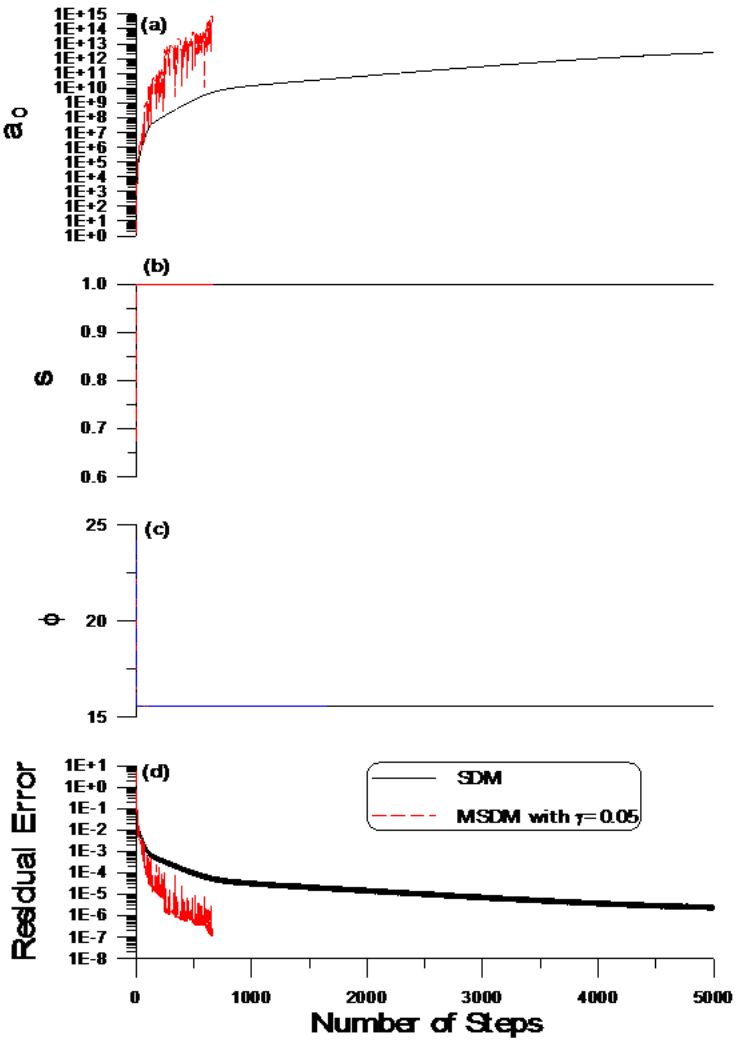

Figure 3: For Example 4.2 comparing (a) a_0, (b) s, (c) ϕ, and (d) residual errors for SDM and MSDM.

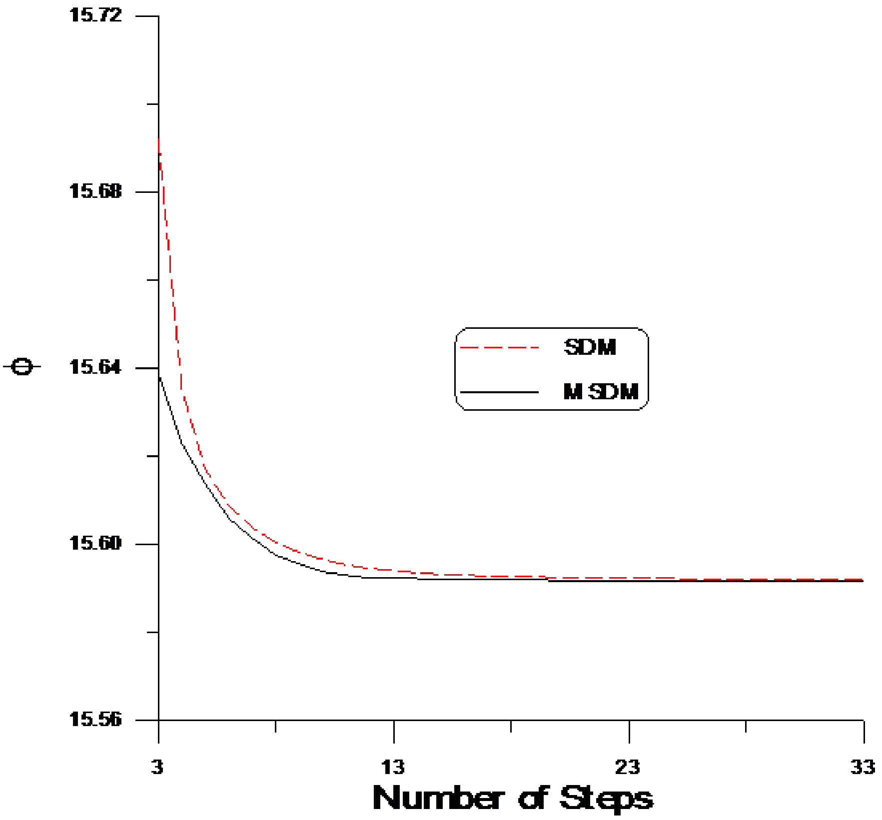

Figure 4: For Example 4.2 comparing ϕ of SDM and MSDM in a larger resolution scale.

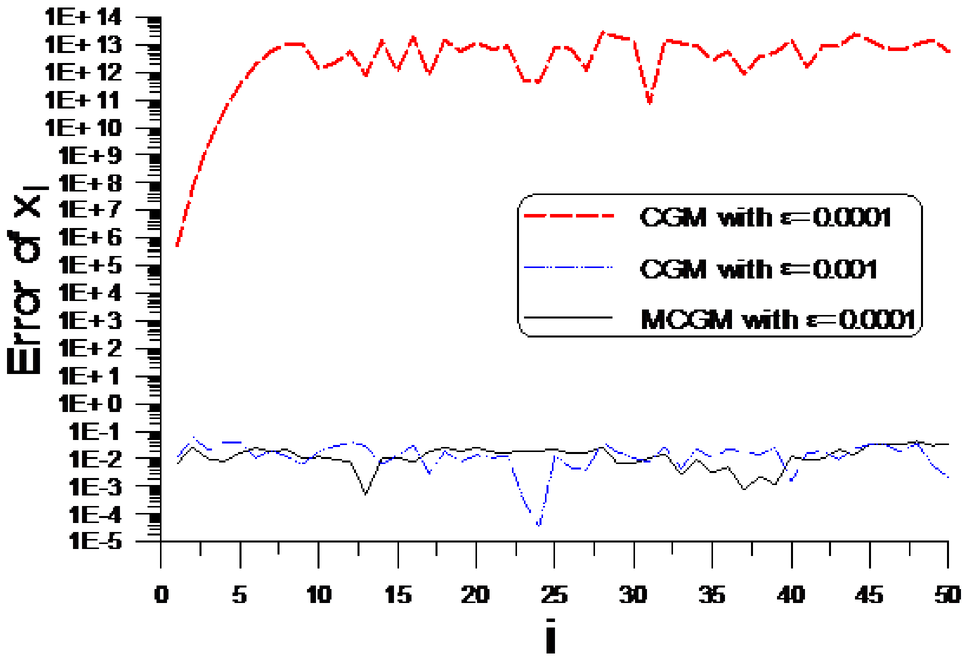

Figure 5: For Example 4.2 comparing numerical errors for CGM and MCGM.

Example 4.3. *As an application of the new algorithms of MSDM and MCGM we consider a polynomial interpolation. Liu and Atluri [13] have solved the ill-posed problem in the*

high-order polynomial interpolation by using a 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:*

$$p(x_i) = y_i, \quad i = 1, \ldots, m, \tag{4.61}$$

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*

$$p(x) = \sum_{i=1}^{m} a_i x^{i-1}, \tag{4.62}$$

where x^i *constitute a monomial basis. The statement that* $p(x)$ *interpolates the data points means that Eq. (4.61) must hold.*
If we substitute Eq. (4.62) into Eq. (4.61), we can obtain a system of linear equations for solving the coefficients a_i*, which in a matrix-vector form reads as*

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{m-2} & x_1^{m-1} \\ 1 & x_2 & x_2^2 & \ldots & x_2^{m-2} & x_2^{m-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 1 & x_{m-1} & x_{m-1}^2 & \ldots & x_{m-1}^{m-2} & x_{m-1}^{m-1} \\ 1 & x_m & x_m^2 & \ldots & x_m^{m-2} & x_m^{m-1} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{m-1} \\ a_m \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix}. \tag{4.63}$$

We have to solve the above system for a_i *to construct the interpolant* $p(x)$*.*
In order to compare the numerical solutions with exact solution we suppose that $a_1 = a_2 = \ldots = a_m = 1$*, and then by Eq. (4.63) we can obtain* $y_1, \ldots, y_m$*. Here we take* $m = 100$ *and* $x_i = -1 + 2i/100$ *to be the nodal points.*
We consider a noise being imposed on the data with $\sigma = 0.05$*, and fix the convergence criterion to be* $\varepsilon = 10^{-3}$*. In Figures 6 and 7 we compare* a_0*,* s*,* ϕ *and the residual errors for* $\gamma = 0$ *and* $\gamma = 0.05$ *for the MSDM. Through 19248 steps the SDM is convergent, and the MSDM is convergent much more quickly with 634 iterations.*
From Figure 6(a) it can be seen that for the case with $\gamma = 0$*, the values of* a_0 *tend to a constant and keep unchanged. By Eq. (4.60) it means that there exists an attracting set on the slow manifold for the iterative orbit of* $\mathbf{x}$ *generated by the SDM as that in Example 4.2. Such that from Figure 6(d) we can see a very slow convergence of SDM. In contrast, from Figure 7(a) it can be seen that for the case with* $\gamma = 0.05$*, the values of* a_0 *do not tend to a constant. As a result the residual error can quicky decrease with an intermittent fashion only through 634 iterations to the required convergence criterion as shown in Figure 7(b).*

Now we test the algorithm in Eq. (2.16) for this example. We first let $\theta_k \in [0, 2]$ *be random numbers. Under the same convergence criterion and the same initial condition, the above algorithm converges with 785 iterations. In Figures 7(c) and 7(d) we show* a_0 *and the residual error for this algorithm. It can be seen that they exhibit no intermittencies. In Figure 8 we compare the accuracies of the above random SDM, SDM and MSDM. The accuracy of random SDM is worser than other two algorithms. In summary the MSDM* *is convergent fastest and is more accurate than other two algotithms. Raydan and Svaiter [20] have argued that θ_k can be random numbers in the range of $[1, 2]$. However, we found that the random SDM with this range of θ_k converges very slowly as shown in Figures 9(a) and 9(b), even it is slightly faster than the original SDM. Basically, it is just a random disturbance of the invariant manifold, and its iterative orbit cannot leave the slow manifold far away. Upon comparing its a_0 and residual error as shown Figures 9(a) and 9(b) with those in Figures 6(a) and 6(d) we can see this point. Now we change the range of θ_k to $[0, 1]$. a_0 and the residual error are shown in Figures 9(c) and 9(d), which is convergent with 742 iterations. The convergence speed is faster than the other two random SDM algorithms. Also the accuracy is slightly increased. It is interesting that its a_0 and the residual error behavior are the mixture of randomness and intermittency as shown in Figures 9(c) and 9(d).*

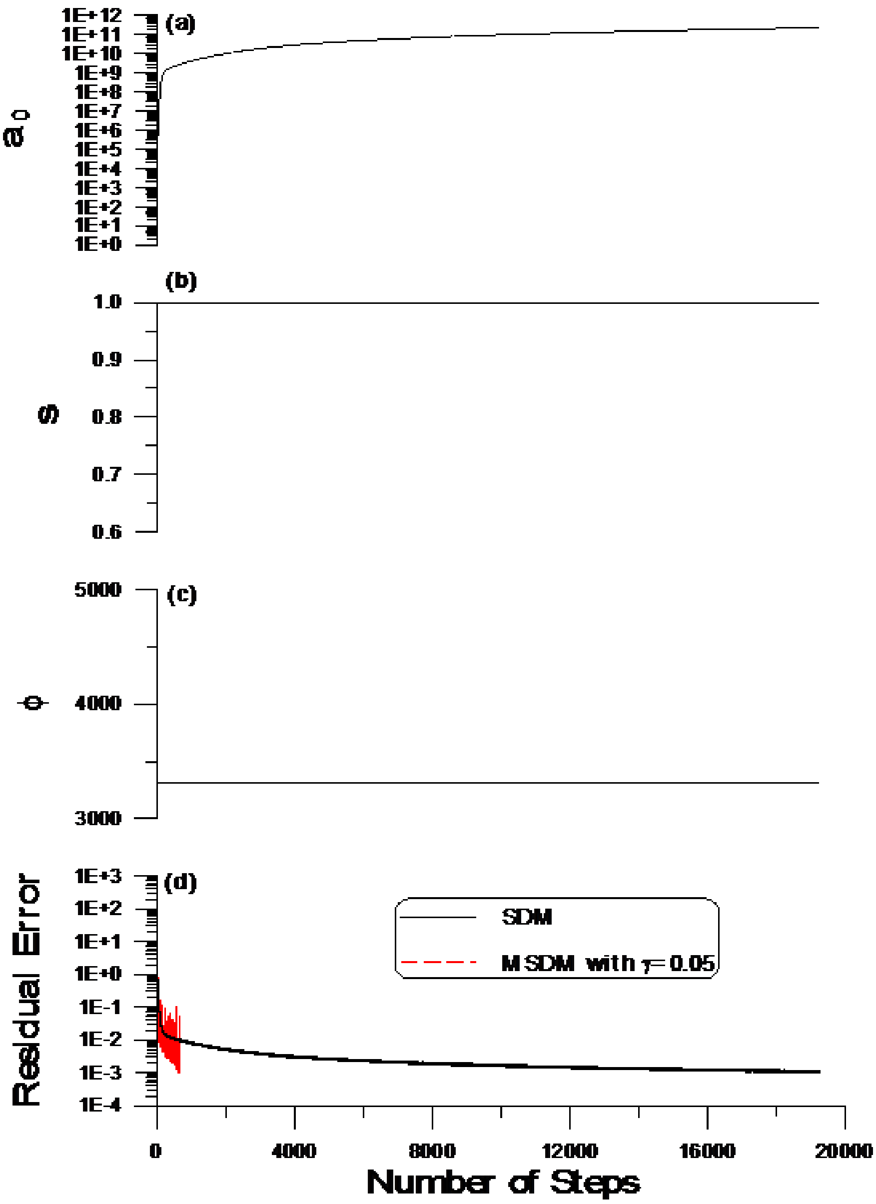

Figure 6: For Example 4.3 showing (a) a_0, (b) s, (c) ϕ and (d) residual error for SDM.

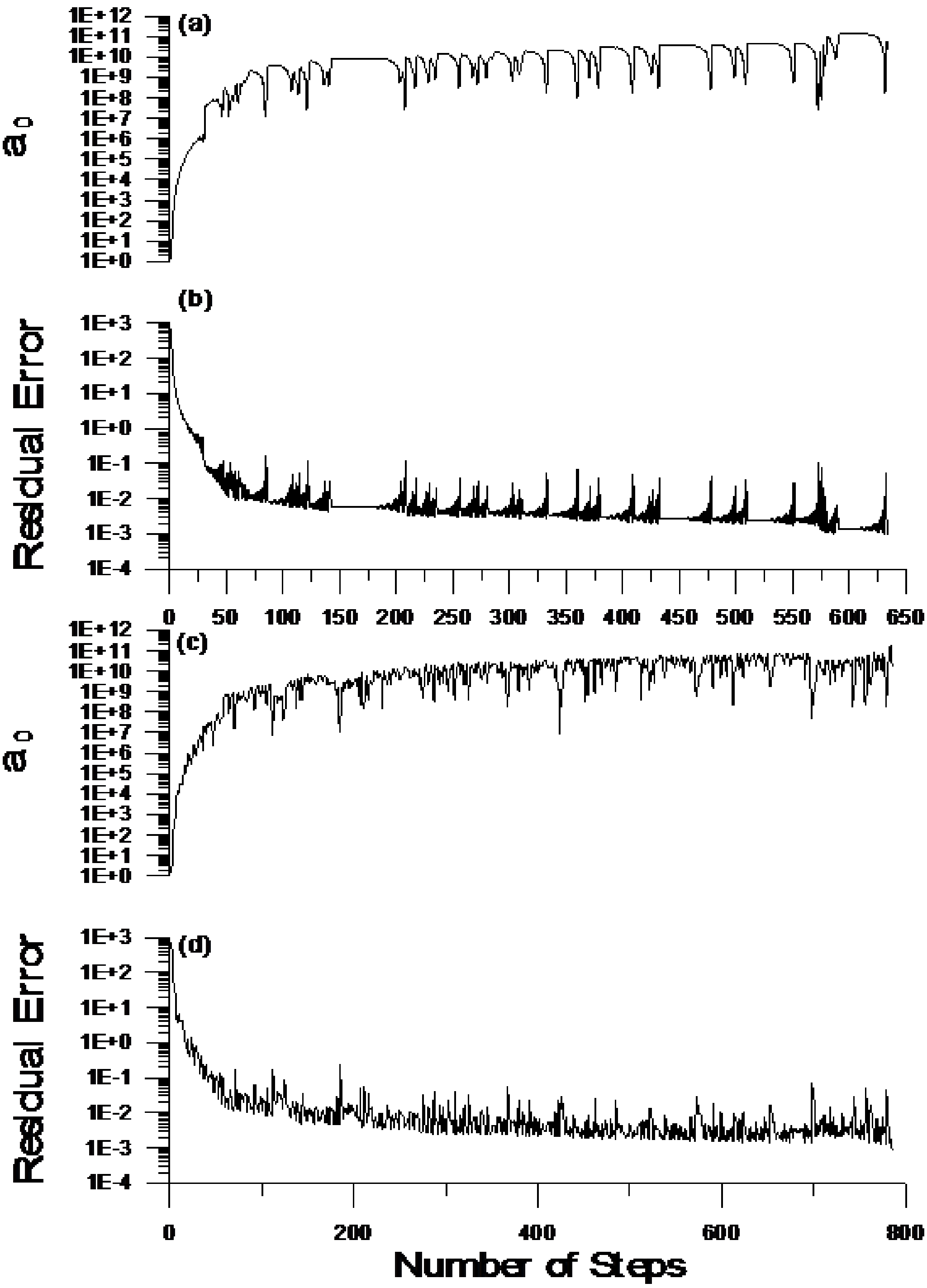

Figure 7: For Example 4.3 comparing (a) and (c) for a_0, and (b) and (d) of the residual errors for MSDM and random SDM with the interval of $[0, 2]$.

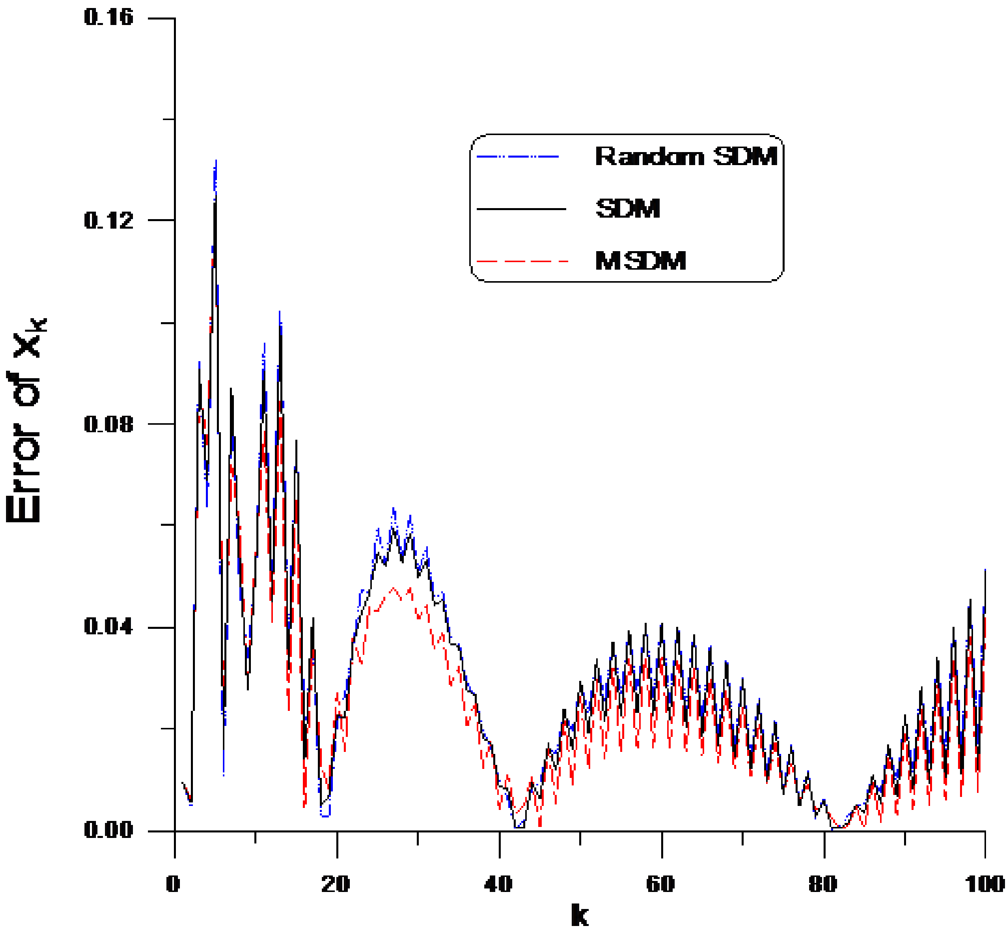

Figure 8: For Example 4.3 comparing accuracies of random SDM, SDM and MSDM.

Example 4.4. *The Runge phenomenon illustrates that the error can occur when one employs a polynomial interpolant of higher degree to interpolate a given function [17]. The function to be interpolated is*

$$f(x) = \frac{1}{1+x^2}, \quad x \in [-1, 1]. \tag{4.64}$$

Under a convergence criterion $\varepsilon = 10^{-8}$ *and with a noise intensity* $\sigma = 0.01$ *we solve this problem by using the CGM and the MCGM with* $\gamma = 0.15$. *The highest order of polynomials used in the interpolation is 100. In Figure 10(a) we compare the numerical solutions with exact solution, and show the numerical errors in Figure 10(b). The CGM converges very fast with 1431 iterations but its maximum error is large up to 0.213. When the MCGM is runned 5000 steps, its solution with the maximum error being 0.0082 is more accurate than the CGM. In Figure 11 we show* a_0 *and the residual error for CGM. It is interesting that the CGM leads to a random* a_0, *and thus it does not tend to the slow manifold described by Eq. (4.60). This is due to that the CGM uses two direction vectors in its iterative algorithm. When we take the gradient of Eq. (4.60), we can obtain two vectors* $\mathbf{r}$ *and* $\mathbf{Ar}$. a_0 *for the MCGM as shown in Figure 12(a) is quite different from that for the CGM. Again, we can see the intermittent phenomenon.*

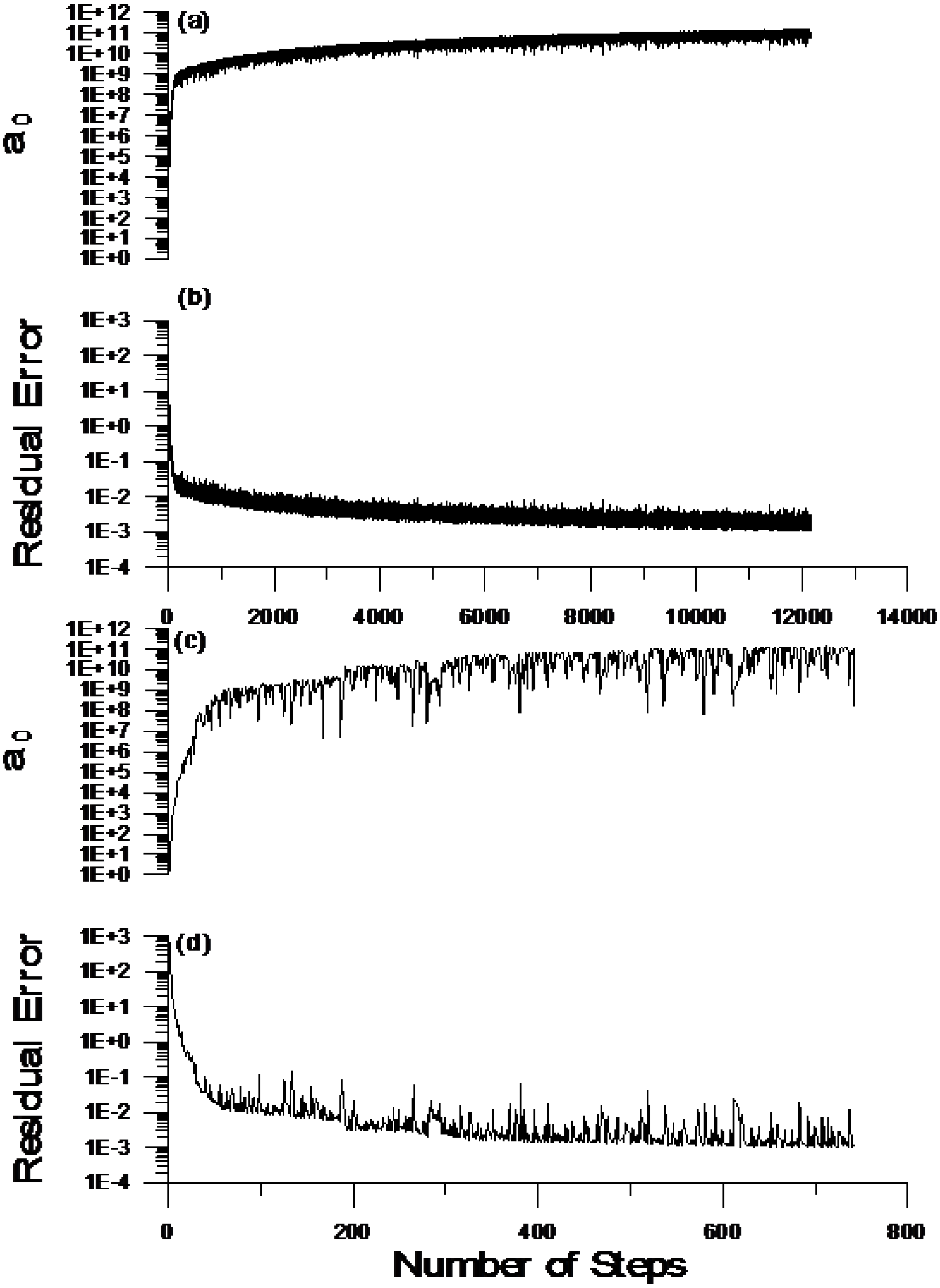

Figure 9: For Example 4.3 comparing (a) and (c) for a_0, and (b) and (d) of the residual errors for random SDM with the intervals of $[1, 2]$ *and* $[0, 1]$.

For the ill-posed linear system under noise, when the CGM gave a very accurate match to that noisy system with a residual error tending to 10^{-8} *as shown in Figure 11(b), the solution is not accurate as shown in Figure 10(b), which exhibits the Runge phenomenon. It means that the CGM is weak against noise, and the error due to noise is enlarged in the iterative process. Conversely, even the residual error for the MCGM is less poor with* *the order* 10^{-3} *as shown in Figure 12(b), the solution by the MCGM is better and reveals no Runge phenomena. When we apply the MSDM with* $\gamma = 0.05$ *to this problem under a convergence criterion* $\varepsilon = 10^{-4}$*, it is runned 1529 steps. The solution of MSDM with the maximum error being 0.0081 is more accurate than the CGM as shown in Figure 10(b) by the solid line. Overall, the MSDM is more accurate than the MCGM, even the maximum errors are the same.* a_0 *and the residual error for the MSDM as shown in Figure 13 are quite different from that for the CGM and MCGM. We can see the intermittent phenomena both in the curve of* a_0 *and in the residual error curve.*

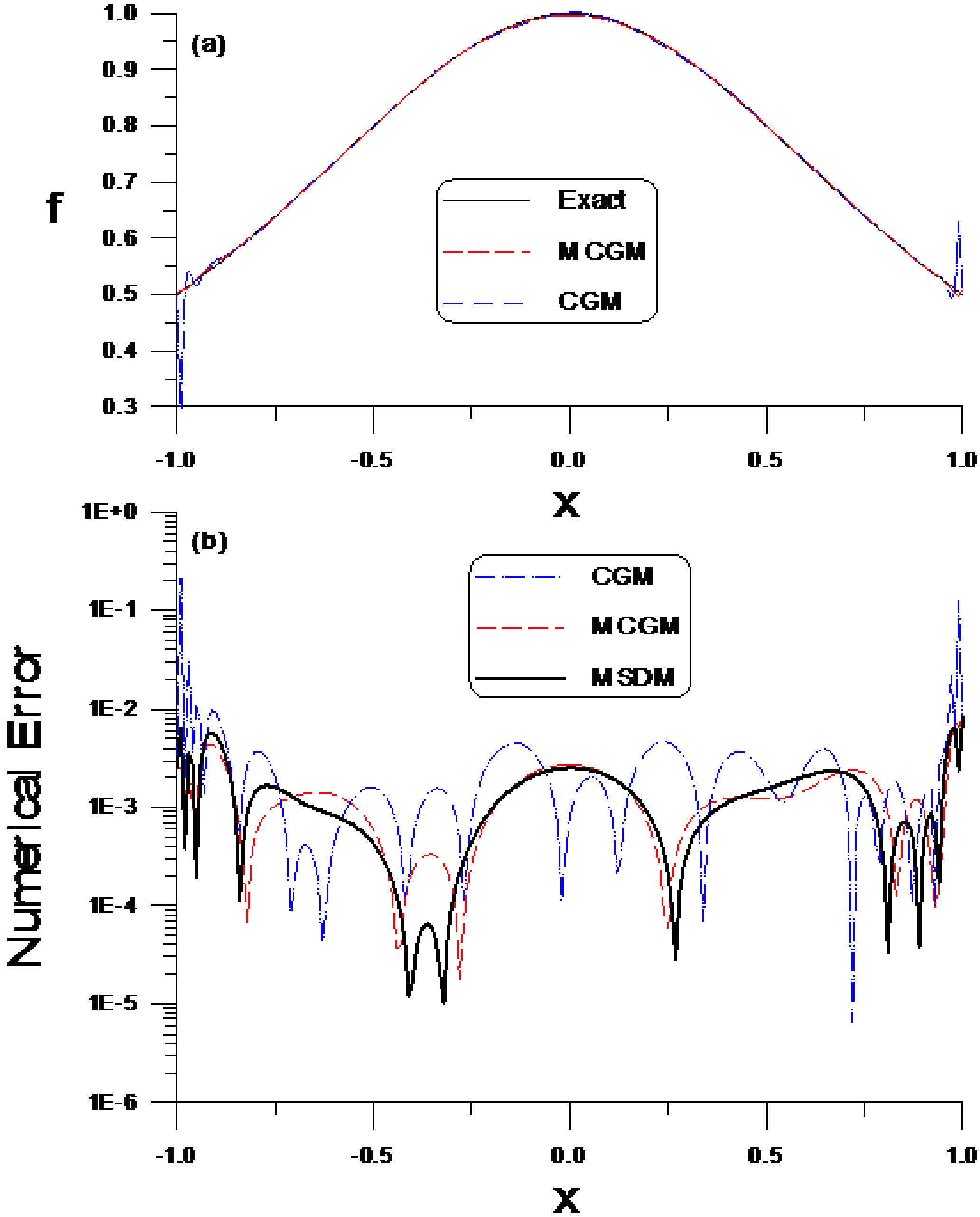

Figure 10: For a function interpolation with 100 orders polynomial: (a) comparing numerical solutions and exact solution, and (b) numerical errors of CGM, MCGM and MSDM.

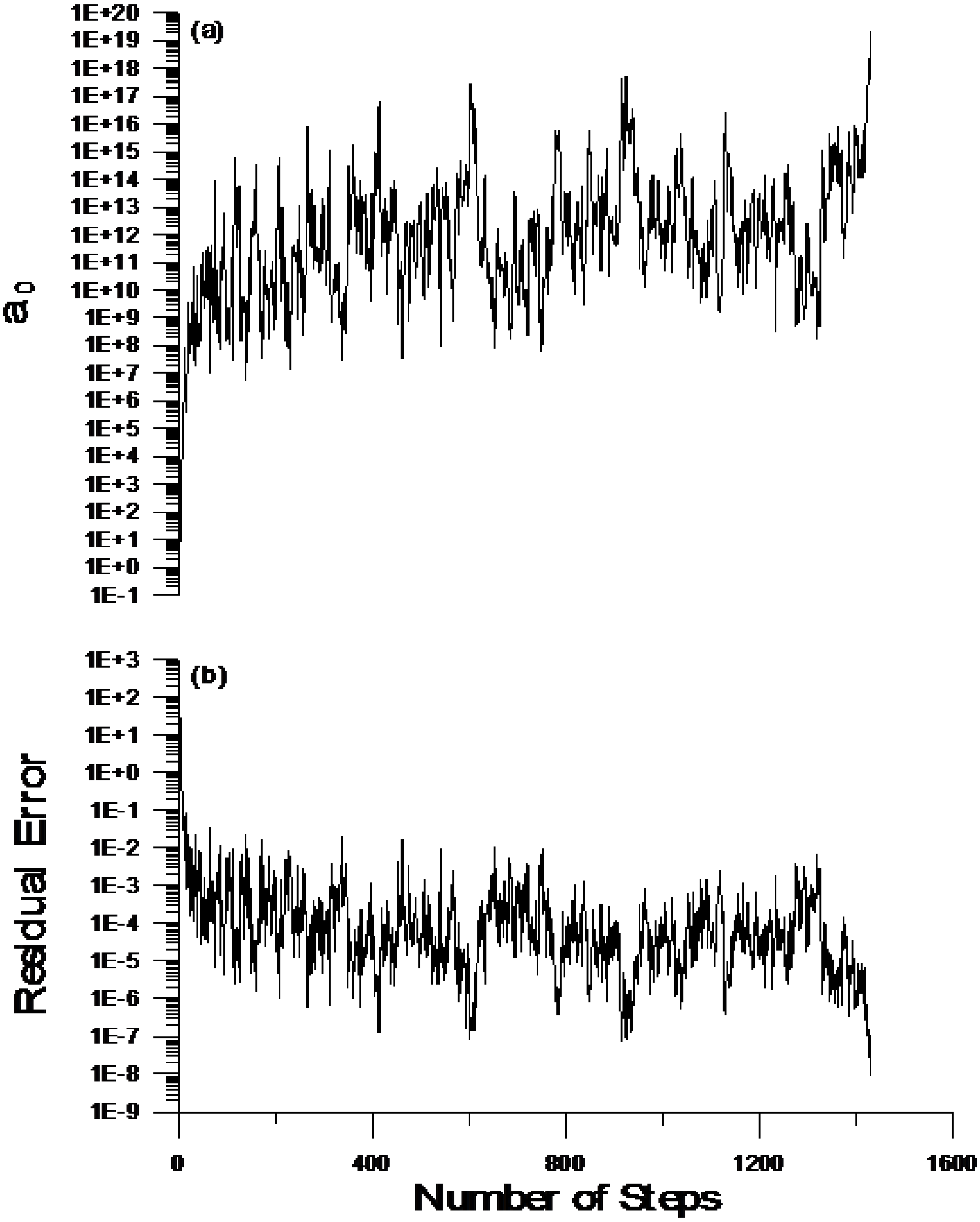

Figure 11: Displaying a_0 and the residual error for CGM.

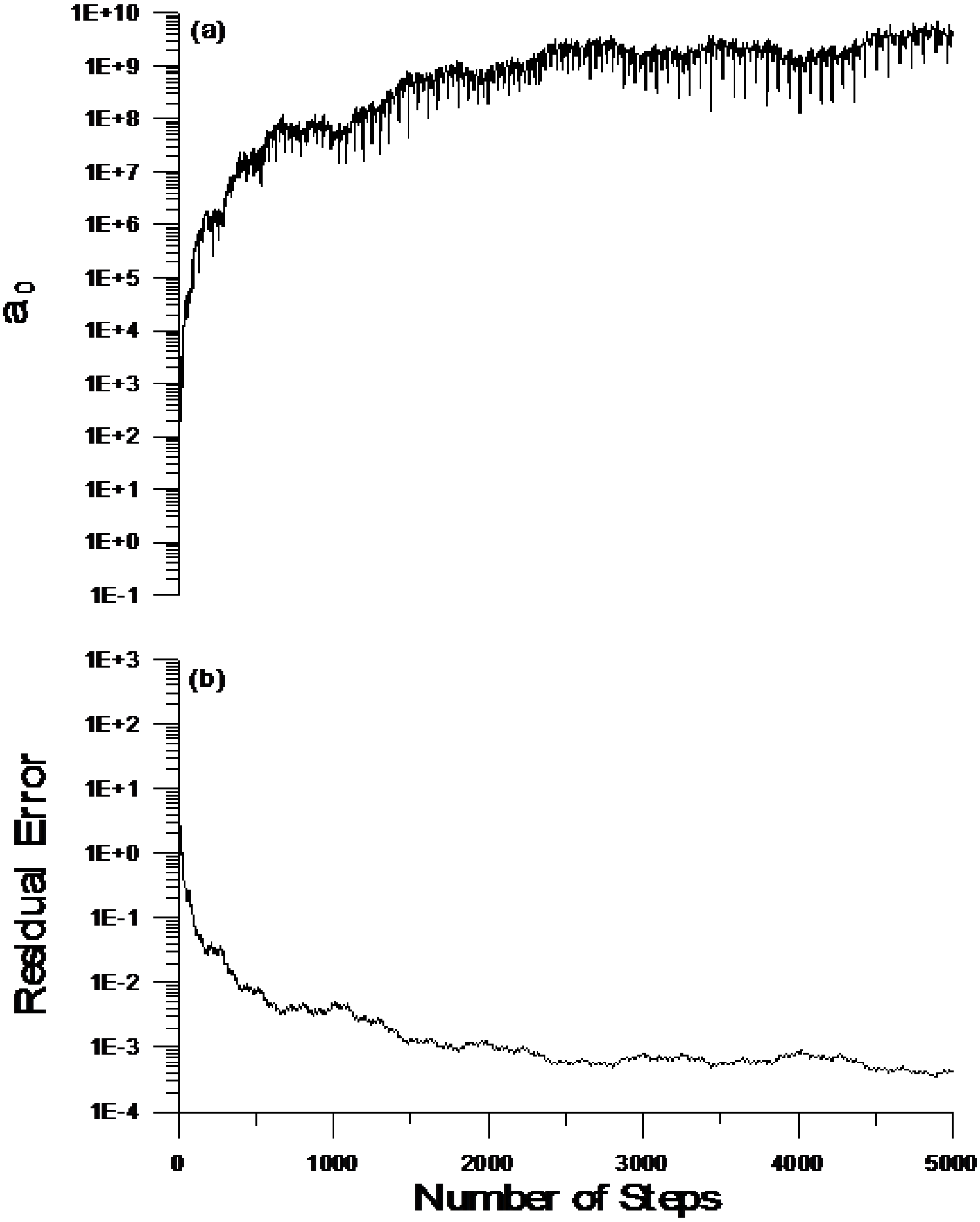

Figure 12: Displaying a_0 and the residual error for MCGM.

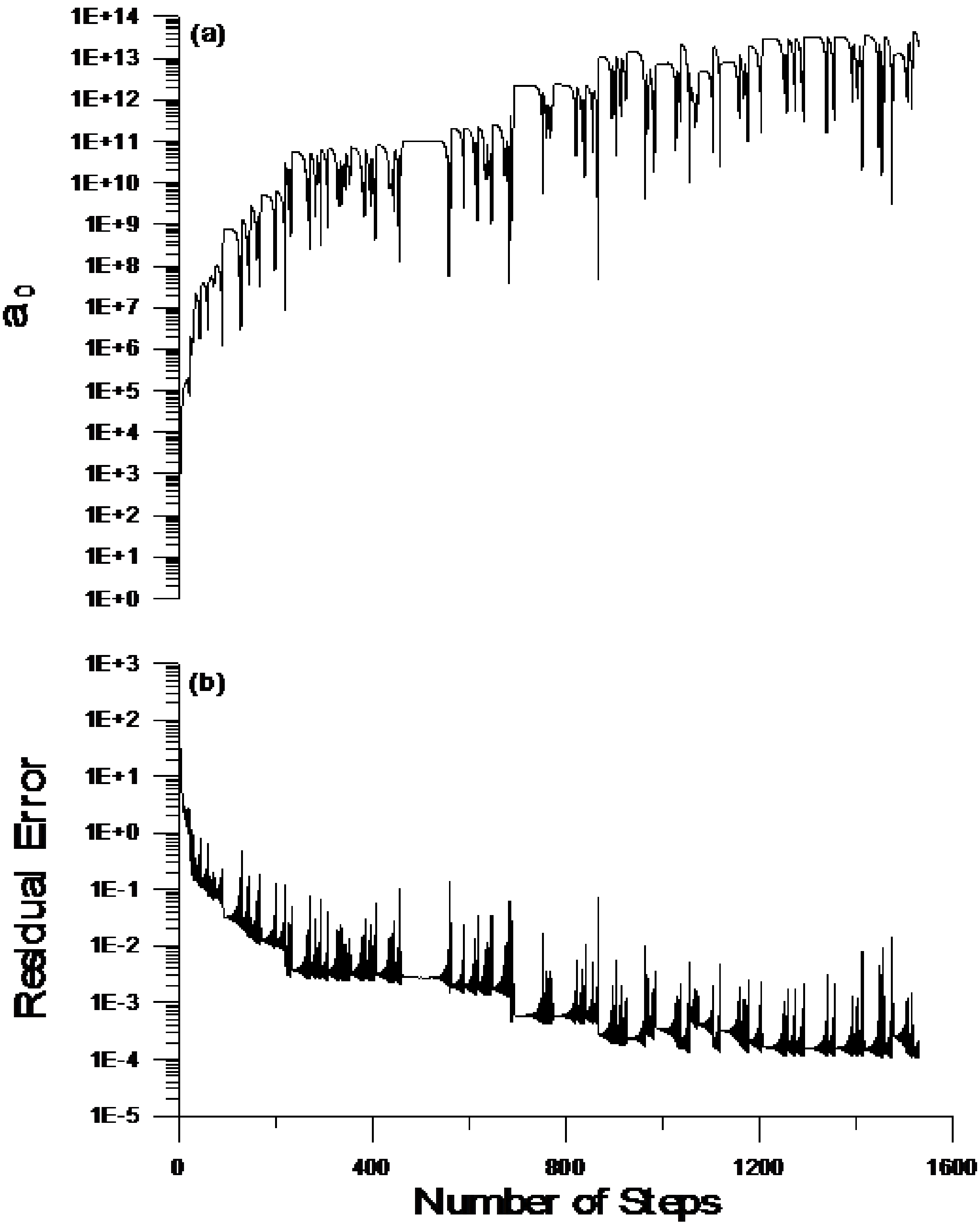

Figure 13: Displaying a_0 and the residual error for MSDM.

5 Conclusions

By embedding the minimization problem of a quadratical functional, which is formed to solve a linear system with a positive definite coefficient matrix, into a continuous manifold with a fictitious time, we can derive a governing system of nonlinear ODEs for the unknown vector. Then by employing the Euler scheme we have derived an iterative algorithm, which is a modification of the classical steepest descent method (SDM) with a parameter $0 \leq \gamma < 1$. This novel algorithm might be named a modified steepest descent method (MSDM), equipped with an acceleration parameter γ. We have proved that the minimizations in the SDM and in our formulation are led to the same algorithm, but they are not the best ones, which usually led to a quite slow convergence in the iterative solution. The parameter γ is a bifurcation parameter, which played a role to switch the situation from a slow convegence with $\gamma = 0$ to a quick convegence with $\gamma > 0$. This bifurcation is indeed an intermittent chaos which destabilizes the original slow manifold which is existent for $\gamma = 0$ in the SDM algorithm. Through several numerical tests we found that the MSDM outperformed very well not only in its convergence speed but also in its robustness against the imposed noise for the ill-posed linear system. By a similar idea we also refined the conjugate gradient method (CGM) to a modified conjugate gradient method (MCGM), which has a better performance against noise.

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.

References

[1] J. Barzilai, J.M. Borwein, Two point step size gradient methods, IMA J. Numer. Anal. 8 (1988) 141-148.
http://dx.doi.org/10.1093/imanum/8.1.141.

[2] Y.H. Dai, L.H. Liao, *R*-linear convergence of the Barzilai and Borwein gradient method, IMA J. Numer. Anal. 22 (2002) 1-10.
http://dx.doi.org/10.1093/imanum/22.1.1.

[3] Y.H Dai, J.Y. Yuan, Y. Yuan, Modified two-point stepsize gradient methods for unconstrained optimization, Comp. Optim. Appl. 22 (2002) 103-109.
http://dx.doi.org/10.1023/A:1014838419611.

[4] Y.H. Dai, Y. Yuan, Alternate minimization gradient method, IMA J. Numer. Anal. 23 (2003) 377-393.
http://dx.doi.org/10.1093/imanum/23.3.377.

[5] R. Fletcher, On the Barzilai-Borwein gradient method, In Optimization and Control with Applications, L Qi, K Teo, X Yang eds., pp. 235-256, Springer, New York (2005).

[6] A. Friedlander, J.M. Martinez, B. Molina, M. Raydan, Gradient method with retards and generalizations, SIAM J. Numer. Anal. 36 (1999) 275-289.
http://dx.doi.org/10.1137/S003614299427315X.

[7] S.L.S. Jacoby, J.S. Kowalik, J.T. Pizzo, Iterative Methods for Nonlinear Optimization Problems, Prentice-Hall, New Jersey (1972).

[8] K. Kunisch, J. Zou, Iterative choices of regularization parameters in linear inverse problems, Inverse Problems 14 (1998) 1247-1264.
http://dx.doi.org/10.1088/0266-5611/14/5/010.

[9] C.-S. Liu, Intermittent transition to quasiperiodicity demonstrated via a circular differential equation, Int. J. Non-Linear Mech. 35 (2000) 931-946.
http://dx.doi.org/10.1016/S0020-7462(99)00070-0.

[10] C.-S. Liu, A study of type I intermittency of a circular differential equation under a discontinuous right-hand side, J. Math. Anal. Appl. 331 (2007) 547-566.
http://dx.doi.org/10.1016/j.jmaa.2006.09.017.

[11] C.-S. Liu, A revision of relaxed steepest descent method from the dynamics on an invariant manifold, CMES: Comput. Model. Eng. Sci. 80 (2011) 57-86.

[12] C.-S. Liu, S.N. Atluri, 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: Comput. Model. Eng. Sci. 41 (2009) 243-261.

[13] C.-S. Liu, S.N. Atluri, A highly accurate technique for interpolations using very high-order polynomials and its applications to some ill-posed linear problems, CMES: Comput. Model. Eng. Sci. 43 (2009) 253-276.

[14] C.-S. Liu, S.N. Atluri, 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: Comput. Model. Eng. Sci. 80 (2011) 275-298.

[15] C.-S. Liu, C.W. Chang, Novel methods for solving severely ill-posed linear equations system, J. Marine Sci. Tech. 9 (2009) 216-227.

[16] C.-S. Liu, H.K. Hong, S.N. Atluri, 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: Comput. Model. Eng. Sci. 60 (2010) 279-308.

[17] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer-Verlag, Berlin (2000).

[18] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, IMA J. Numer. Anal. 13 (1993) 321-326.
http://dx.doi.org/10.1093/imanum/13.3.321.

[19] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, IAM J. Optim. 7 (1997) 26-33.
http://dx.doi.org/10.1137/S1052623494266365.

[20] M. Raydan, B.F. Svaiter, Relaxed steepest descent and Cauchy-Barzilai-Borwein method, Comp. Optim. Appl. 21 (2002) 155-167.
http://dx.doi.org/10.1023/A:1013708715892.

[21] E. Resmerita, Regularization of ill-posed problems in Banach spaces: convergence rates, Inverse Problems 21 (2005) 1303-1314.
http://dx.doi.org/10.1088/0266-5611/21/4/007.

[22] G. Stewart, Introduction to Matrix Computations, Academic Press, New York (1973).

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

[24] Y. Wang, T. Xiao, Fast realization algorithms for determining regularization parameters in linear inverse problems, Inverse Problems 17 (2001) 281-291. *http://dx.doi.org/10.1088/0266-5611/17/2/308.*

[25] J. Xie, J. Zou, An improved model function method for choosing regularization parameters in linear inverse problems, Inverse Problems 18 (2002) 631-643. *http://dx.doi.org/10.1088/0266-5611/18/3/307.*

[26] Y. Yuan, A new stepsize for the steepest descent method, J. Comp. Math. 24 (2006) 149-156.