

Research Article

An Optimally Generalized Steepest-Descent Algorithm for Solving Ill-Posed Linear Systems

Chein-Shan Liu

Department of Civil Engineering, National Taiwan University, Taipei 106-17, Taiwan

Correspondence should be addressed to Chein-Shan Liu; liucs@ntu.edu.tw

Received 25 August 2013; Revised 29 October 2013; Accepted 29 October 2013

Academic Editor: Hui-Shen Shen

Copyright © 2013 Chein-Shan Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is known that the steepest-descent method converges normally at the first few iterations, and then it slows down. We modify the original steplength and descent direction by an optimization argument with the new steplength as being a merit function to be maximized. An optimal iterative algorithm with m -vector descent direction in a Krylov subspace is constructed, of which the m optimal weighting parameters are solved in closed-form to accelerate the convergence speed in solving ill-posed linear problems. The optimally generalized steepest-descent algorithm (OGSDA) is proven to be convergent with very fast convergence speed, accurate and robust against noisy disturbance, which is confirmed by numerical tests of some well-known ill-posed linear problems and linear inverse problems.

1. Introduction

The steepest-descent method (SDM), which can be traced back to Cauchy (1847), is the simplest gradient method for solving positive definite linear equations system. The SDM is effective for well-posed and low-dimensional linear problems; however, for large scale linear system and ill-posed linear system it converges very slowly.

Several modifications to the SDM have been addressed. These modifications 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 used in the SDM leads to the slow convergence behavior. Barzilai and Borwein [1] presented a new choice of steplength through two-point stepsize. Although their method did not guarantee the monotonic descent of the residual norms, Barzilai and Borwein [1] were able to produce a substantial improvement of the convergence speed for a certain test of a positive linear system. The results of Barzilai and Borwein [1] have encouraged many researches on the SDM, for example, Raydan [2, 3], Friedlander et al. [4], Raydan and Svaiter [5], Dai et al. [6], Dai and Liao [7], Dai and Yuan [8], Fletcher [9], and Yuan [10].

The iterative method for solving the system of algebraic equations can be derived from the discretization of certain ordinary differential equations (ODEs) system [11]. In

particular, some descent methods can be interpreted as the discretizations of gradient flows [12]. Indeed, the continuous algorithms have been investigated in many literature works for a long time, for example, Gavurin [13], Alber [14], Hirsch and Smale [15], and Chu [16]. The Lyapunov methods used in the analysis of iterative methods have been made by Ortega and Rheinboldt [17] and Bhaya and Kaszkurewicz [11, 18, 19]. Liu [20] has developed a state feedback controlling method together with a Lie-group differential algebraic equations method to solve ill-posed linear systems.

In this paper we solve

$$\mathbf{B}\mathbf{x} = \mathbf{b}_1, \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^n$ is an unknown vector determined from a given coefficient matrix $\mathbf{B} \in \mathbb{R}^{n \times n}$, which might be unsymmetric, and the input $\mathbf{b}_1 \in \mathbb{R}^n$ which might be polluted by random noise. A measure of the ill-posedness of (1) is the condition number of \mathbf{B} :

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

where $\|\mathbf{B}\|$ is the Frobenius norm of \mathbf{B} .

Instead of (1), we can solve a normal linear system:

$$\mathbf{C}\mathbf{x} = \mathbf{b}, \quad (3)$$

where

$$\mathbf{b} = \mathbf{B}^T \mathbf{b}_1, \quad \mathbf{C} = \mathbf{B}^T \mathbf{B} > \mathbf{0}. \quad (4)$$

Throughout this paper the superscript “T” signifies the transpose.

We consider iterative method for solving (3) and define for any vector \mathbf{x}_k the steepest-descent vector:

$$\mathbf{r}_k := \mathbf{C}\mathbf{x}_k - \mathbf{b}. \quad (5)$$

Ascher et al. [21] have viewed the gradient-descent method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta_k \mathbf{r}_k, \quad (6)$$

as a forward Euler scheme of a time-dependent ODE:

$$\dot{\mathbf{x}} = \mathbf{b} - \mathbf{C}\mathbf{x}. \quad (7)$$

The absolute stability bound

$$\eta_k \leq \frac{2}{\max_{\sigma(\mathbf{C})} \lambda} \quad (8)$$

must be obeyed if a uniform stepsize is employed, where $\sigma(\mathbf{C})$ is the set of all the eigenvalues of \mathbf{C} .

Specifically, (6) presents a steepest-descent method (SDM), if the steplength is taken to be

$$\eta_k = \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^T \mathbf{C} \mathbf{r}_k}. \quad (9)$$

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 a 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 an orthogonality to the residual vector at each iterative step.

The relaxed SDM (RSDM) for solving (3) is summarized as follows [22, 23].

- (i) Give γ and an initial value of \mathbf{x}_0 .
- (ii) For $k = 0, 1, \dots$, we repeat the following iterations:

$$\mathbf{r}_k = \mathbf{C}\mathbf{x}_k - \mathbf{b}, \quad \eta_k = \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^T \mathbf{C} \mathbf{r}_k}, \quad (10)$$

If \mathbf{x}_{k+1} converges according to a given stopping criterion $\|\mathbf{r}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii).

The steepest-descent method is the basis of several gradient-based methods [22, 23], and it is one of the most prominent iterative methods for solving positive definite linear equations system. Although the SDM works very well for most linear systems, the SDM does lose some of its luster for some ill-posed problems like inverse problems, image processing, and box-constrained optimization. Both the efficiency and robustness of the SDM iterative techniques can be improved by using a suitable preconditioning technique.

However, we do not go to the details of the preconditioning techniques of SDM, and instead we will propose an alternative approach by modifying the search direction and steplength used in the SDM by an argument from the optimality of a suitably defined merit function. In this paper we explore a generalization of the SDM by introducing the concept of optimal descent vector to solve linear equations, which is an optimal combination of the steepest-descent vector $-\mathbf{r}$ and the m -dimensional search vector in a Krylov subspace. Here we modify the direction \mathbf{r} as well as the steplength η from a theoretical optimization being performed on the derived steplength in an m -dimensional Krylov subspace.

The remaining parts of this paper are arranged as follows. A numerical iterative scheme with $-\mathbf{u}$ as a descent direction is proposed, and the convergence to the minimal point $\mathbf{x}^* = \mathbf{C}^{-1}\mathbf{b}$ of a quadratic functional is proven in Section 2. In Section 3 we propose a generalization of the steepest-descent method to solve positive linear equations systems, by using a new steplength as a merit function to be optimized in an m -dimensional Krylov subspace. Then, a genuine algorithm is constructed in Section 4, resulting in an optimally generalized steepest-descent algorithm (OGSDA) in terms of m optimal weighting parameters which is being derived explicitly to maximize the introduced steplength. The numerical examples are given in Section 5 to display some advantages of the OGSDA, which is compared with CGM, BBM, GMRES, and other algorithms. Finally, the conclusions are drawn in Section 6.

2. The Convergence of New Algorithm

Solving (3) by the steepest-descent method (SDM) is equivalent to solving the following minimum problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \varphi(\mathbf{x}) = \min_{\mathbf{x} \in \mathbb{R}^n} \left[\frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} - \mathbf{b}^T \mathbf{x} \right]. \quad (11)$$

In the SDM, we search the next $\mathbf{x}(t + \Delta t)$ from the current state $\mathbf{x}(t)$ by minimizing the functional φ along the direction $-\mathbf{r}(t)$; that is,

$$\min_{\eta} \varphi(\mathbf{x}(t) - \eta \mathbf{r}(t)). \quad (12)$$

Through some calculations we can obtain

$$\eta = \frac{\|\mathbf{r}(t)\|^2}{\mathbf{r}^T(t) \mathbf{C} \mathbf{r}(t)}. \quad (13)$$

Thus we have the following iterative algorithm of SDM:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \frac{\|\mathbf{r}(t)\|^2}{\mathbf{r}^T(t) \mathbf{C} \mathbf{r}(t)} \mathbf{r}(t). \quad (14)$$

A straightforward calculation reveals that

$$\varphi(\mathbf{x}(t + \Delta t)) < \varphi(\mathbf{x}(t)). \quad (15)$$

It means that the iterative sequences of SDM converge to the minimal point.

Then, we consider a more general scheme with the descent direction $-\mathbf{u}(t)$; that is,

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \frac{\mathbf{r}(t) \cdot \mathbf{u}(t)}{\mathbf{u}^T(t) \mathbf{C} \mathbf{u}(t)} \mathbf{u}(t). \quad (16)$$

Similarly we have the following result.

Lemma 1. Equation (16) is a result of the following minimization along the descent direction $-\mathbf{u}$:

$$\min_{\eta} \varphi(\mathbf{x}(t) - \eta \mathbf{u}(t)). \quad (17)$$

The functional φ is decreasing stepwise and the iterative scheme (16) is convergent to the minimal point $\mathbf{x}^* = \mathbf{C}^{-1} \mathbf{b}$.

Proof. Inserting

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) - \eta \mathbf{u}(t) \quad (18)$$

into the functional φ we have

$$\begin{aligned} \varphi(t + \Delta t) &= \frac{1}{2} [\mathbf{x}^T \mathbf{C} \mathbf{x} - 2\eta \mathbf{u}^T \mathbf{C} \mathbf{x} + \eta^2 \mathbf{u}^T \mathbf{C} \mathbf{u}] \\ &\quad - \mathbf{b}^T \mathbf{x} + \eta \mathbf{u}^T \mathbf{b}, \end{aligned} \quad (19)$$

where we simplify the notation of $\varphi(\mathbf{x}(t + \Delta t))$ by $\varphi(t + \Delta t)$ and omit the terms (t) in $\mathbf{x}(t)$ and $\mathbf{u}(t)$. Taking the derivative of (19) with respect to η and equating it to zero we can derive

$$\eta = \frac{\mathbf{u}^T \mathbf{r}}{\mathbf{u}^T \mathbf{C} \mathbf{u}}. \quad (20)$$

Because η is a steplength, we demand it to be positive; that is, $\mathbf{r} \cdot \mathbf{u} > 0$. By feeding the above η into (18) we can derive the iterative scheme (16).

Inserting (20) into (19) and subtracting the resultant by

$$\varphi(t) = \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} - \mathbf{b}^T \mathbf{x}, \quad (21)$$

we can obtain

$$\varphi(t + \Delta t) - \varphi(t) = \frac{1}{2} \eta^2 \mathbf{u}^T \mathbf{C} \mathbf{u} - \eta \mathbf{u}^T \mathbf{r} = -\frac{(\mathbf{u}^T \mathbf{r})^2}{2 \mathbf{u}^T \mathbf{C} \mathbf{u}}. \quad (22)$$

It implies

$$\varphi(t + \Delta t) = \varphi(t) - \frac{(\mathbf{u}^T \mathbf{r})^2}{2 \mathbf{u}^T \mathbf{C} \mathbf{u}} < \varphi(t), \quad (23)$$

and the functional φ is strictly decreasing. Because the minimal point $\mathbf{x}^* = \mathbf{C}^{-1} \mathbf{b}$ of φ is just the exact solution of (3), such that when the iterative sequences are generated from scheme (16), the functional φ decreases step-wisely to its minimal value and $\mathbf{x}(t)$ tends to the minimal point gradually. Thus we can conclude that the new scheme is convergent absolutely. \square

3. A Generalized SDM and Its Optimization

In Lemma 1 we have proven that the new scheme (16) is convergent, but it cannot tell us what descent direction $-\mathbf{u}$ is the best. Although the SDM uses the steepest-descent direction $-\mathbf{r}$, it is well known that the steplength η given in (13) quickly tends to a small value, causing the slowdown of the SDM scheme.

In order to overcome the slowness of SDM, Barzilai and Borwein [1] have presented a new choice of the steplength through two-point stepsize. The algorithm of the Barzilai-Borwein method (BBM) is

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{(\Delta \mathbf{r}_{k-1})^T \Delta \mathbf{x}_{k-1}}{\|\Delta \mathbf{r}_{k-1}\|^2} \mathbf{r}_k, \quad (24)$$

where $\Delta \mathbf{r}_{k-1} = \mathbf{r}_k - \mathbf{r}_{k-1}$ and $\Delta \mathbf{x}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$, with initial guesses $\mathbf{r}_0 = \mathbf{0}$ and $\mathbf{x}_0 = \mathbf{0}$. Nowadays the improvements of the original BBM have been developed in many literature works [24–30] to treat different ill-posed and inverse problems.

In the numerical solution of linear equations system the Krylov subspace method is one of the most important classes of projective type numerical methods [31–35]. The iterative algorithms that are applied to solve large scale linear systems are mostly the preconditioned Krylov subspace methods [36].

The GMRES used to solve (1) can be summarized as follows [37, 38].

- (i) Select m and give an initial \mathbf{x}_0 .
- (ii) For $k = 0, 1, \dots$, we repeat the following computations:

$$\mathbf{r}_k = \mathbf{b} - \mathbf{B} \mathbf{x}_k,$$

Arnoldi procedure to set up

$$\begin{aligned} \mathbf{v}_j^k, \quad j = 1, \dots, m, \quad \left(\text{from } \mathbf{v}_1^k = \frac{\mathbf{r}_k}{\|\mathbf{r}_k\|} \right), \\ \mathbf{V}_k = [\mathbf{v}_1^k, \dots, \mathbf{v}_m^k], \end{aligned} \quad (25)$$

Solve $(\bar{\mathbf{H}}_k^T \bar{\mathbf{H}}_k) \mathbf{y}_k = \|\mathbf{r}_k\| \bar{\mathbf{H}}_k^T \mathbf{e}_1$, obtaining \mathbf{y}_k ,

$$\mathbf{z}_k = \mathbf{V}_k \mathbf{y}_k,$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{z}_k.$$

If \mathbf{x}_{k+1} converges according to a given stopping criterion $\|\mathbf{r}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii). In the above, \mathbf{V}_k is an $n \times m$ matrix, while $\bar{\mathbf{H}}_k$ is an augmented Heissenberg upper triangular matrix with dimension $(m + 1) \times m$, and \mathbf{e}_1 is the first column of \mathbf{I}_{m+1} .

3.1. A Generalized SDM. We suppose that the original steepest-descent direction $-\mathbf{r}$ in the SDM is replaced by a new descent vector $-\mathbf{u}$:

$$\mathbf{u} = \mathbf{r} + \sum_{k=1}^m \alpha_k \mathbf{v}_k, \quad (26)$$

which is an optimal combination of \mathbf{r} and the m -vector \mathbf{v}_k , $k = 1, \dots, m$. The coefficients α_k will be optimized in Section 3.2.

Now we describe how to set up the m -vector \mathbf{v}_k , $k = 1, \dots, m$, by the Krylov subspace method. Suppose that we have an m -dimensional Krylov subspace generated by the coefficient matrix \mathbf{C} from the steepest-descent vector \mathbf{r} :

$$\mathcal{K}_m := \text{span} \{ \mathbf{C}\mathbf{r}, \dots, \mathbf{C}^m \mathbf{r} \}. \quad (27)$$

Then the Arnoldi process is used to normalize and orthogonalize the Krylov vectors $\mathbf{C}^j \mathbf{r}$, $j = 1, \dots, m$, such that the resultant vectors \mathbf{v}_i , $i = 1, \dots, m$, satisfy $\mathbf{v}_i \cdot \mathbf{v}_j = \delta_{ij}$, $i, j = 1, \dots, m$, where δ_{ij} is the Kronecker delta tensor.

The other way to obtain \mathbf{v}_i , $i = 1, \dots, m$, is simply by using the unit vectors in the Euclidean space, where \mathbf{v}_i is the i th column vector of the identity matrix \mathbf{I}_n . The resultant m -dimensional space is named the unit subspace.

3.2. Optimization of Algorithm. Let \mathbf{J} be an $n \times m$ matrix, which consists of

$$\mathbf{J} := [\mathbf{v}_1, \dots, \mathbf{v}_m]. \quad (28)$$

That is, the j th column of \mathbf{J} is the vector \mathbf{v}_j . Because $\mathbf{v}_1, \dots, \mathbf{v}_m$ are linearly independent vectors and $m < n$, the expansion matrix \mathbf{J} has a full rank with $\text{rank}(\mathbf{J}) = m$. Then, (26) can be written as

$$\mathbf{u} = \mathbf{r} + \mathbf{J}\boldsymbol{\alpha}, \quad (29)$$

where $\boldsymbol{\alpha} := (\alpha_1, \dots, \alpha_m)^T$.

Now, in view of (16) the new steplength becomes

$$\eta = \frac{\mathbf{r}^T \mathbf{u}}{\mathbf{u}^T \mathbf{C} \mathbf{u}}. \quad (30)$$

When $\mathbf{u} = \mathbf{r}$, the above η recovers to that defined in (13). In order to speed up the convergence we can search a way to maximize the steplength, such that we have the following maximization problem:

$$\max \left\{ \eta = \frac{\mathbf{r}^T \mathbf{u}}{\mathbf{u}^T \mathbf{C} \mathbf{u}} \right\}. \quad (31)$$

Here, we use the steplength η as a merit function to be optimized. Hence, we can prove the following result.

Theorem 2. For $\text{rank}(\mathbf{J}) = m$ and $\mathbf{u} \in \mathbf{r} + \mathcal{K}_m$, the descent direction $-\mathbf{u}$ which maximizes (31) is given by

$$\mathbf{u} = \mathbf{r} - \mathbf{E}\mathbf{C}\mathbf{r} + \lambda \mathbf{E}\mathbf{r}, \quad \mathbf{A} = \mathbf{J}^T \mathbf{C} \mathbf{J}, \quad (32)$$

$$\mathbf{E} = \mathbf{J} \mathbf{A}^{-1} \mathbf{J}^T, \quad b_0 = \mathbf{r}^T (\mathbf{C} \mathbf{E} \mathbf{C} - \mathbf{C}) \mathbf{r} < 0, \quad (33)$$

$$b_1 = \mathbf{r}^T (\mathbf{I}_n - \mathbf{C} \mathbf{E}) \mathbf{r} > 0, \quad (34)$$

$$b_2 = \mathbf{r}^T \mathbf{E} \mathbf{r} > 0, \quad (35)$$

$$\lambda = \frac{\sqrt{b_1^2 - b_0 b_2} - b_1}{b_2}. \quad (36)$$

Moreover, the steplength η defined in (30) is positive; that is,

$$\eta = \frac{1}{2\lambda} = \frac{b_2}{2\sqrt{b_1^2 - b_0 b_2} - 2b_1} > 0. \quad (37)$$

Proof. We divide this proof into three parts.

(A) *A Minimization Problem.* Inserting (29) into (31) we can obtain the following minimization problem:

$$\min_{\alpha_1, \dots, \alpha_m} \frac{\mathbf{u} \cdot (\mathbf{C} \mathbf{u})}{\mathbf{r} \cdot \mathbf{u}}, \quad (38)$$

where

$$\mathbf{r} \cdot \mathbf{u} = \|\mathbf{r}\|^2 + \mathbf{r}^T \mathbf{J} \boldsymbol{\alpha}, \quad (39)$$

$$\mathbf{u} \cdot (\mathbf{C} \mathbf{u}) = \mathbf{r}^T \mathbf{C} \mathbf{r} + 2\mathbf{r}^T \mathbf{C} \mathbf{J} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha}, \quad (40)$$

in which

$$\mathbf{A} := \mathbf{J}^T \mathbf{C} \mathbf{J} \quad (41)$$

is an $m \times m$ positive definite matrix. Because the matrix \mathbf{C} has a full rank with $\text{rank}(\mathbf{C}) = n$ and positive definite and \mathbf{J} has a full rank with $\text{rank}(\mathbf{J}) = m$, from the above equation it follows that \mathbf{A} has a full rank with $\text{rank}(\mathbf{A}) = m$. Then, we confirm that \mathbf{A} is an $m \times m$ positive definite matrix by (41).

From (39) and (40) we can derive

$$\mathbf{y}_1 := \nabla_{\boldsymbol{\alpha}} \mathbf{r} \cdot \mathbf{u} = \mathbf{J}^T \mathbf{r}, \quad (42)$$

$$\mathbf{y}_2 := \nabla_{\boldsymbol{\alpha}} \mathbf{u} \cdot (\mathbf{C} \mathbf{u}) = 2\mathbf{J}^T \mathbf{C} \mathbf{r} + 2\mathbf{A} \boldsymbol{\alpha}, \quad (43)$$

where $\nabla_{\boldsymbol{\alpha}}$ denotes the gradient with respect to $\boldsymbol{\alpha}$.

By using

$$\nabla_{\boldsymbol{\alpha}} \frac{\mathbf{u} \cdot (\mathbf{C} \mathbf{u})}{\mathbf{r} \cdot \mathbf{u}} = \mathbf{0} \implies \mathbf{r} \cdot \mathbf{u} \nabla_{\boldsymbol{\alpha}} \mathbf{u} \cdot (\mathbf{C} \mathbf{u}) - \mathbf{u} \cdot (\mathbf{C} \mathbf{u}) \nabla_{\boldsymbol{\alpha}} \mathbf{r} \cdot \mathbf{u} = \mathbf{0}, \quad (44)$$

we can derive the following governing equation of vector form to solve the optimal parameter $\boldsymbol{\alpha}$:

$$\mathbf{r} \cdot \mathbf{u} \mathbf{y}_2 - \mathbf{u} \cdot (\mathbf{C} \mathbf{u}) \mathbf{y}_1 = \mathbf{0}. \quad (45)$$

Now we prove that $\boldsymbol{\alpha}$ solved from the above equation is a minimal point of the functional in (38). We can compute the Jacobian matrix of (45), which is denoted by $\mathbf{J}_{\boldsymbol{\alpha}}$:

$$\mathbf{J}_{\boldsymbol{\alpha}} = 2\mathbf{r} \cdot \mathbf{u} \mathbf{A} + \mathbf{y}_2 \mathbf{y}_1^T - \mathbf{y}_1 \mathbf{y}_2^T. \quad (46)$$

It is an $m \times m$ positive definite matrix, because for any $\mathbf{z} \neq \mathbf{0} \in \mathbb{R}^m$ we have

$$\mathbf{z}^T \mathbf{J}_{\boldsymbol{\alpha}} \mathbf{z} = 2\mathbf{r} \cdot \mathbf{u} \mathbf{z}^T \mathbf{A} \mathbf{z} > 0, \quad (47)$$

where we have used the positive definite property of \mathbf{A} , and $\mathbf{r} \cdot \mathbf{u} > 0$. Because η defined by (20) is a steplength, we demand it to be positive; that is, $\mathbf{r} \cdot \mathbf{u} > 0$ (see the proof given below for $\mathbf{r} \cdot \mathbf{u} > 0$).

(B) *A Quadratic Equation.* Equations (40) and (43) can be written as

$$\mathbf{u} \cdot (\mathbf{C}\mathbf{u}) = \mathbf{r}^T \mathbf{C} \mathbf{r} + 2\mathbf{y}_0^T \mathbf{J} \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{A} \boldsymbol{\alpha}, \quad (48)$$

$$\mathbf{y}_2 = 2\mathbf{J}^T \mathbf{y}_0 + 2\mathbf{A} \boldsymbol{\alpha}, \quad (49)$$

where

$$\mathbf{y}_0 := \mathbf{C} \mathbf{r}. \quad (50)$$

From (45) we can observe that \mathbf{y}_2 is proportional to \mathbf{y}_1 , which is supposed to be

$$\mathbf{y}_2 = \frac{\mathbf{u} \cdot (\mathbf{C}\mathbf{u})}{\mathbf{r} \cdot \mathbf{u}} \mathbf{y}_1 = 2\lambda \mathbf{y}_1, \quad (51)$$

where 2λ is a multiplier to be determined. From (51), two equations follow

$$\mathbf{y}_2 = 2\lambda \mathbf{y}_1, \quad (52)$$

$$\mathbf{u} \cdot (\mathbf{C}\mathbf{u}) = 2\lambda \mathbf{r} \cdot \mathbf{u}. \quad (53)$$

Then, by (42), (49), and (52) we can solve $\boldsymbol{\alpha}$ by

$$\boldsymbol{\alpha} = \lambda \mathbf{A}^{-1} \mathbf{J}^T \mathbf{r} - \mathbf{A}^{-1} \mathbf{J}^T \mathbf{y}_0. \quad (54)$$

Inserting (54) for $\boldsymbol{\alpha}$ into (39) and (48) we have that

$$\begin{aligned} \mathbf{r} \cdot \mathbf{u} &= \|\mathbf{r}\|^2 - \mathbf{y}_0^T \mathbf{E} \mathbf{r} + \lambda \mathbf{r}^T \mathbf{E} \mathbf{r}, \\ \mathbf{u} \cdot (\mathbf{C}\mathbf{u}) &= \mathbf{r}^T \mathbf{C} \mathbf{r} - \mathbf{y}_0^T \mathbf{E} \mathbf{y}_0 + \lambda^2 \mathbf{r}^T \mathbf{E} \mathbf{r}, \end{aligned} \quad (55)$$

where

$$\mathbf{E} := \mathbf{J} \mathbf{A}^{-1} \mathbf{J}^T \quad (56)$$

is an $n \times n$ positive semi-definite matrix.

Now, from (53) and (55) we can derive a quadratic equation to solve λ :

$$a_2 \lambda^2 + a_1 \lambda + a_0 = 0, \quad (57)$$

where

$$\begin{aligned} a_0 &:= \mathbf{y}_0^T \mathbf{E} \mathbf{y}_0 - \mathbf{r}^T \mathbf{C} \mathbf{r}, \\ a_1 &:= 2\|\mathbf{r}\|^2 - 2\mathbf{y}_0^T \mathbf{E} \mathbf{r}, \quad a_2 := \mathbf{r}^T \mathbf{E} \mathbf{r}. \end{aligned} \quad (58)$$

(C) *A Closed-Form Solution of Optimal Parameter $\boldsymbol{\alpha}$.* Now we prove $a_0 < 0$. By using (50) we have

$$a_0 = \mathbf{r}^T [\mathbf{C} \mathbf{E} \mathbf{C} - \mathbf{C}] \mathbf{r} < 0, \quad (59)$$

if we can prove

$$\mathbf{C} \mathbf{E} \mathbf{C} - \mathbf{C} < \mathbf{0}. \quad (60)$$

From (56) and (41) it follows that

$$\mathbf{E} \mathbf{C} \mathbf{J} = \mathbf{J} (\mathbf{J}^T \mathbf{C} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{C} \mathbf{J} = \mathbf{J}, \quad (61)$$

$$\mathbf{J}^T \mathbf{C} \mathbf{E} = \mathbf{J}^T \mathbf{C} \mathbf{J} (\mathbf{J}^T \mathbf{C} \mathbf{J})^{-1} \mathbf{J}^T = \mathbf{J}^T, \quad (62)$$

$$\mathbf{J}^T [\mathbf{C} \mathbf{E} \mathbf{C}] \mathbf{J} = \mathbf{J}^T [\mathbf{C} \mathbf{J} (\mathbf{J}^T \mathbf{C} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{C}] \mathbf{J} = \mathbf{J}^T \mathbf{C} \mathbf{J}. \quad (63)$$

They indicate that \mathbf{E} is a Penrose pseudoinverse of \mathbf{C} in the Krylov subspace; hence, (60) follows.

According to (58), it is obvious that $a_2 > 0$; hence, by (59) we have

$$a_1^2 - 4a_0 a_2 > 0. \quad (64)$$

As a consequence, (57) has a closed-form positive real solution:

$$\lambda = \frac{\sqrt{a_1^2 - 4a_0 a_2} - a_1}{2a_2}. \quad (65)$$

Inserting it into (54) we can obtain the closed-form solution of $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \frac{\sqrt{a_1^2 - 4a_0 a_2} - a_1}{2a_2} \mathbf{A}^{-1} \mathbf{J}^T \mathbf{r} - \mathbf{A}^{-1} \mathbf{J}^T \mathbf{y}_0. \quad (66)$$

Upon comparing (53) and (30) we can derive (37) with the steplength η being positive. Moreover, as a direct result of (30) and (37) we have $\mathbf{r} \cdot \mathbf{u} > 0$. \square

Remark 3. Because of $a_2 > 0$ and $a_0 < 0$ in (57), it has a positive solution as shown in (65) and in addition a negative solution as follows:

$$\lambda = \frac{-\sqrt{a_1^2 - 4a_0 a_2} - a_1}{2a_2}, \quad (67)$$

where

$$a_1 = 2\mathbf{r}^T (\mathbf{I}_n - \mathbf{C} \mathbf{E}) \mathbf{r} > 0, \quad (68)$$

in view of (58), (50), and (62). Hence, by (30) and (37) we have

$$\mathbf{r} \cdot \mathbf{u} = \|\mathbf{r}\|^2 - \mathbf{y}_0^T \mathbf{E} \mathbf{r} + \lambda \mathbf{r}^T \mathbf{E} \mathbf{r} < 0, \quad (69)$$

which contradicts (20), where η is a steplength being positive. Below we will give a numerical example to demonstrate this fact.

Corollary 4. *If $m = n$, then one has*

$$\boldsymbol{\alpha} = \mathbf{J}^{-1} \mathbf{C}^{-1} \mathbf{r} - \mathbf{J}^{-1} \mathbf{r}. \quad (70)$$

Proof. If $m = n$, then \mathbf{J} defined by (28) is an $n \times n$ nonsingular matrix. Simultaneously, \mathbf{E} defined by (56) is equal to \mathbf{C}^{-1} , $a_0 = 0$, $a_1 = 0$, $a_2 > 0$, and meanwhile (66) reduces to (70), where we have taken $\lambda = 1$ according to the above condition. \square

Remark 5. If $m = n$, by (29) and (70) we have

$$\mathbf{u} = \mathbf{r} + \mathbf{J}\boldsymbol{\alpha} = \mathbf{r} + \mathbf{J}[\mathbf{J}^{-1}\mathbf{C}^{-1}\mathbf{r} - \mathbf{J}^{-1}\mathbf{r}] = \mathbf{C}^{-1}\mathbf{r}, \quad (71)$$

and inserting it into (31) we can prove $\eta = 1$. In any way, under this condition we have the fastest convergent behavior, because the solution $\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$ is already obtained. Usually, we cannot take $m = n$, which leads to a quite difficult task to find the inverse of $\mathbf{A} = \mathbf{J}^T\mathbf{C}\mathbf{J}$ when $m = n$ is a large number.

4. An Optimal Generalized Steepest-Descent Algorithm

Thus, we arrive at the following *optimally generalized steepest-descent algorithm* (OGSDA).

- (i) Select m and $0 \leq \gamma < 1$, and give an initial value of \mathbf{x}_0 .
- (ii) For $k = 0, 1, \dots$, we repeat the following computations:

$$\mathbf{r}_k = \mathbf{C}\mathbf{x}_k - \mathbf{b},$$

Arnoldi procedure to set up

$$\mathbf{v}_j^k, \quad j = 1, \dots, m, \quad \left(\text{from } \mathbf{v}_1^k = \frac{\mathbf{C}\mathbf{r}_k}{\|\mathbf{C}\mathbf{r}_k\|} \right),$$

$$\mathbf{J}_k = [\mathbf{v}_1^k, \dots, \mathbf{v}_m^k],$$

$$\mathbf{A}_k = \mathbf{J}_k^T \mathbf{C} \mathbf{J}_k,$$

$$\mathbf{E}_k = \mathbf{J}_k \mathbf{A}_k^{-1} \mathbf{J}_k^T,$$

Compute (36), obtaining λ_k ,

$$\mathbf{u}^k = \mathbf{r}_k - \mathbf{E}_k \mathbf{C} \mathbf{r}_k + \lambda_k \mathbf{E}_k \mathbf{r}_k,$$

$$\eta_k = \frac{1}{2\lambda_k},$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1 - \gamma) \eta_k \mathbf{u}^k.$$

If \mathbf{x}_{k+1} converges according to a given stopping criterion $\|\mathbf{r}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii).

Remark 6. Depending on the different choice of \mathbf{v}_j , $j = 1, \dots, m$, the optimal algorithm is further classified into two types. When \mathbf{v}_j , $j = 1, \dots, m$, are constructed from the Krylov subspace and followed by the Arnoldi process to orthonormalize the Krylov vectors, the resultant optimal algorithm is named the OGSDA with Krylov subspace method. On the other hand, if \mathbf{v}_j is simply the j th column of the unit matrix \mathbf{I}_m , then the resultant optimal algorithm is named the OGSDA with unit subspace method. Both methods have their own advantage. In the Krylov subspace method, the expansion matrix \mathbf{J} is not fixed, and it can adjust its configuration at each iterative step to accelerate the convergence, which however leads to a consumption of computational time in the calculation of the inverse of $\mathbf{A} = \mathbf{J}^T\mathbf{C}\mathbf{J}$ to obtain \mathbf{A}^{-1} and \mathbf{E} . On the other hand, in the unit subspace method, the expansion

matrix \mathbf{J} is fixed, and we only need to calculate $\mathbf{A} = \mathbf{J}^T\mathbf{C}\mathbf{J}$, \mathbf{A}^{-1} and \mathbf{E} one time, which is much time saving than that by using the Krylov subspace method; however, it cannot adjust the configuration of the descent direction at each iterative step, which might lead to an ill-condition in the course of computation.

We can evaluate the cost of computing the approximate solution \mathbf{x}_k by OGSDA with k steps. We can estimate the cost of the computation of the scalar λ , which at each iterative step requires an inversion of $m \times m$ matrix, the matrix-vector multiplications of $n \times n$ -matrix and n -vector three times, and the inner products of two n -vectors five times. This portion is very time saving with totally $3kn + km^2$, because of $m \ll n$. The computation of the Arnoldi vectors needs totally $k(k+1)n + kn^2$ multiplications. In the computation of \mathbf{u} it requires $2kn$ multiplications. The k steps OGSDA therefore requires totally $k(k+6)n + kn^2 + km^2$ multiplications. Dividing by the total number of steps k we can obtain that each step requires $(k+6)n + n^2 + m^2$ multiplications on the average. The computational cost of OGSDA is slightly larger than that of GMRES and BBM per step. However, because the convergence speed of the OGSDA is very fast, the total computational time is less than that of GMRES and BBM.

5. Numerical Examples

In order to evaluate the performance of the OGSDA, we test some well-known ill-posed linear problems of the Hilbert problem, the backward heat conduction problem, the heat source identification problem, and the inverse Cauchy problem.

5.1. Example 1. First by testing the performance of OGSDA on the solution of linear equations system, we consider the following convex quadratic programming problem with equality constraint:

$$\min \left\{ f = \frac{1}{2} \mathbf{x}^T \mathbf{P} \mathbf{x} + \mathbf{q}^T \mathbf{x} \right\}, \quad (73)$$

$$\mathbf{Q} \mathbf{x} = \mathbf{b}_0, \quad (74)$$

where \mathbf{P} is an $n_1 \times n_1$ matrix, \mathbf{Q} is an $m_1 \times n_1$ matrix, and \mathbf{b}_0 is an m_1 -vector, which means that (74) provides m_1 linear constraints. According to the Lagrange theory we need to solve (1) with the following \mathbf{b} and \mathbf{B} :

$$\mathbf{b}_1 := \begin{bmatrix} -\mathbf{q} \\ \mathbf{b}_0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{P} & \mathbf{Q}^T \\ \mathbf{Q} & \mathbf{0}_{m_1 \times m_1} \end{bmatrix}. \quad (75)$$

For definite we take $n_1 = 3$ and $m_1 = 2$ with

$$\min \left\{ f = x_1^2 + 2x_2^2 + x_3^2 - 2x_1x_2 + x_3 \right\}, \quad (76)$$

$$x_1 + x_2 + x_3 = 4, \quad 2x_1 - x_2 + x_3 = 2.$$

Under the following parameters $m = 2$, $\gamma = 0.2$, and $\varepsilon = 10^{-5}$ and starting from the initial values of $x_1 = x_2 = x_3 = 1$ and two Lagrange multipliers $\lambda_1 = -2$ and $\lambda_2 = 2$ we

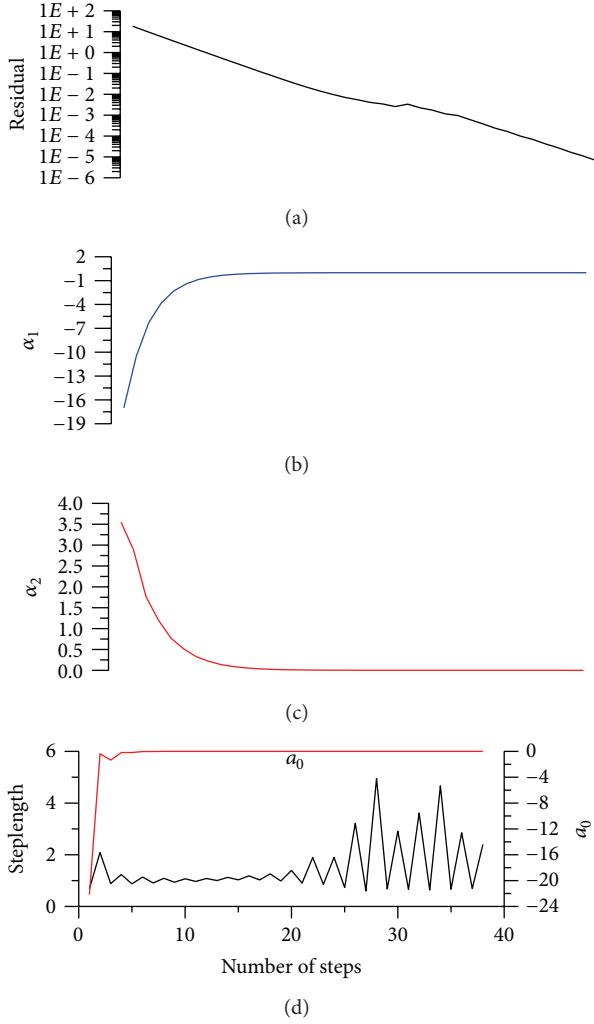


FIGURE 1: Example 1 showing (a) residual, ((b) and (c)) optimal parameters, and (d) steplength and a_0 .

apply the OGSDA to solve the resultant five-dimensional linear equations system, which is convergent with 38 steps as shown in Figure 1(a) for the residual. In Figures 1(b) and 1(c) we show, respectively, the optimal parameters α_1 and α_2 and the values of η_k and a_0^k . It can be seen that $a_0^k \in [-22.16, -5.72 \times 10^{-13}]$, as proven by (59) $a_0^k < 0$. Now we raise m to $m = n = 5$ and also the convergence criterion to $\varepsilon = 10^{-5}$ and $\gamma = 0$ is fixed. The numerical results obtained by the OGSDA with Krylov subspace is convergent with only three steps. The final solution is a minimal value $\min f = 3.977273$ at the point $(x_1, x_2, x_3) = (1.909090909092, 1.954545454479, 0.13636363636)$.

In order to claim that the solution of λ given by (67) is incorrect, we use the corresponding algorithm to solve the above problem under the parameters $m = 2$ and $\gamma = 0.2$. First the algorithm does not converge within 500 steps under a weak convergence criterion with $\varepsilon = 10^{-2}$ as shown in Figure 2(a). Then the algorithm gives negative steplength of η as shown in Figure 2(b).

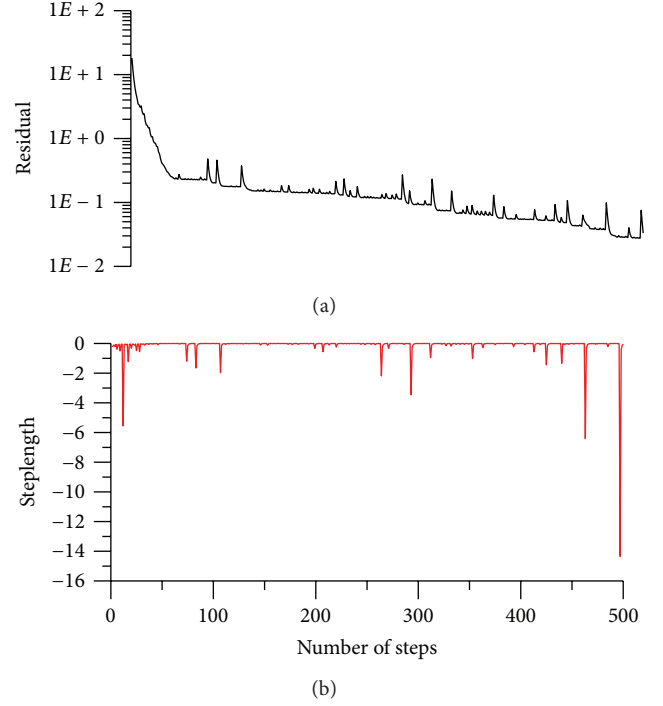


FIGURE 2: For Example 1, using (67) the algorithm does not converge and gives a negative steplength.

5.2. *Example 2.* Finding an n -order polynomial function $p(x) = a_0 + a_1x + \dots + a_nx^n$ to best match a continuous function $f(x)$ in the interval of $x \in [0, 1]$

$$\min_{\deg(p) \leq n} \int_0^1 [f(x) - p(x)]^2 dx, \quad (77)$$

leads to a problem governed by (3), where \mathbf{C} is the $(n+1) \times (n+1)$ Hilbert matrix defined by

$$C_{ij} = \frac{1}{i+j-1}, \quad (78)$$

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

$$\mathbf{b} = \begin{bmatrix} \int_0^1 f(x) dx \\ \int_0^1 xf(x) dx \\ \vdots \\ \int_0^1 x^n f(x) dx \end{bmatrix} \quad (79)$$

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

The Hilbert matrix is a notorious example of highly ill-conditioned matrices. Equation (3) with the matrix \mathbf{C} having a large condition number usually displays that an arbitrarily small perturbation of data on the right-hand side may lead to an arbitrarily large perturbation to the solution on the left-hand side.

In this example we consider a highly ill-conditioned linear system (3) with \mathbf{C} given by (78). The ill-posedness of

TABLE 1: Comparing the numerical results for Example 2 with different methods.

Solutions	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
Exact	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
SVD	0.9999	1.008	0.985	0.995	1.007	1.012	1.009	0.999	0.984
SDM	1.00001	0.99980	1.00091	0.99910	0.99928	1.00035	1.00103	1.00062	0.99898
CGM	1.00000	1.00001	0.99924	1.00019	0.99988	0.99985	1.00008	1.00019	0.99987
RSDM	1.00000	0.99986	1.00087	0.99888	0.99928	1.00058	1.00128	1.00066	0.99856
OGSDA	0.9999999999	1.000000005	0.9999999185	1.000000569	0.9999979488	1.000004126	0.9999953195	1.000002797	0.9999993151

(3) increases fast with n . We consider an exact solution with $x_j = 1$ and b_i is given by

$$b_i = \sum_{j=1}^n \frac{1}{i+j-1} + \sigma R(i), \quad (80)$$

with $R(i)$ being random numbers between $[-1, 1]$.

Liu [22] has applied the relaxed steepest-descent method (RSDM) with $\gamma = 0.06$, starting from $x_1 = \dots = x_9 = 0.5$, to solve the Hilbert problem with $n = 9$ with a stopping criterion $\varepsilon = 10^{-8}$ for the relative residual. Through 50000 iterations the numerical solution converges to the exact solution very accurately as shown in Table 1 with the maximum error being 1.44×10^{-3} . When we apply the OGSDA with the unit subspace method under $m = n = 9$ and $\gamma = 10^{-5}$ to solve this problem it converges only three steps and with the maximum error being 4.68×10^{-6} . It is very time saving because we only need to calculate A^{-1} one time, by which using the matrix conjugate gradient method (MCGM) developed by Liu et al. [39] we only spend 103 steps under the convergence criterion $\varepsilon_1 = 10^{-5}$ to find A^{-1} . For the purpose of comparison, the values obtained by other numerical methods are also listed in Table 1.

As shown in Corollary 4 when m is taken to be $m = n$ we have the fastest convergent solution. Usually, we can take a smaller m for the easy solution of A^{-1} . For example, if we take $m = 5$ in the OGSDA with the Krylov subspace method, we can obtain the solution with the maximum error being 4.45×10^{-4} through only four steps. Under the same conditions, the CGM is convergent with six steps with the maximum error being 7.857×10^{-3} .

Next we consider a more difficult case with $n = 300$. In the computation a random noise in the level of $\sigma = 10^{-6}$ is imposed on the right-hand side, and the convergence criteria used in CGM and OGSDA are both $\varepsilon = 10^{-2}$ for the relative residual. The CGM converges very fast with 4 iterations as shown in Figure 3(a); however, the maximum error of CGM is large up to 0.25. For this noised case the OGSDA is applicable and convergent with 4 iterations, where $\gamma = 0.15$ and $m = 10$ are used. The maximum error of OGSDA is 0.0113. In Figure 3 we compare (a) relative residuals, (b) steplength of OGSDA, and (c) numerical errors of OGSDA, CGM, GMRES, and BBM. It is obvious that for this highly ill-posed problem the OGSDA is convergent fast, and accurate. The solutions obtained by the CGM, GMRES, and BBM are not good, although the GMRES with $m = 9$ converges with

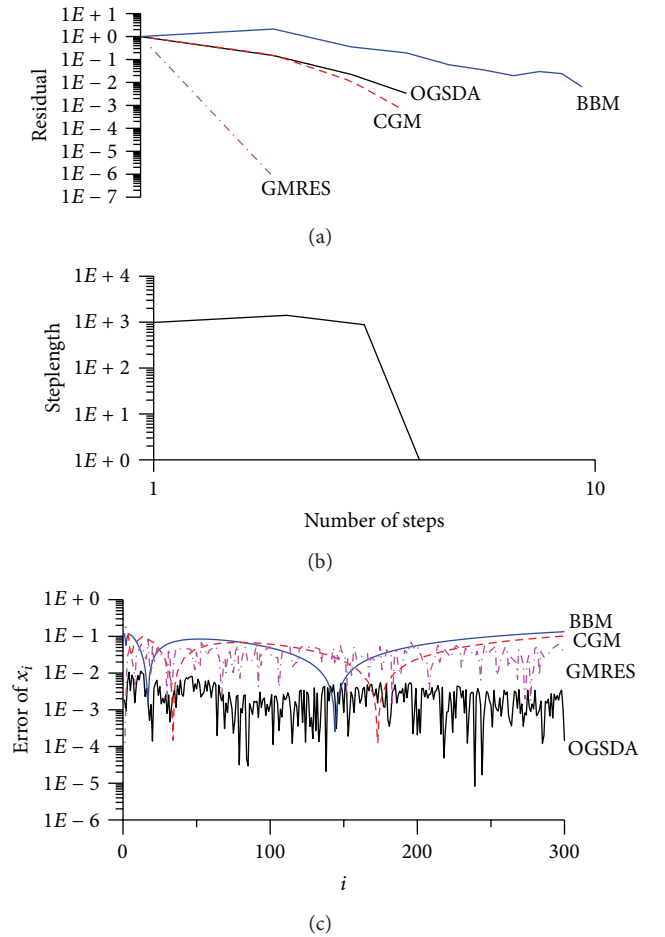


FIGURE 3: Example 2 of the Hilbert problem with $n = 300$, showing (a) relative residuals, (b) steplength of OGSDA, and (c) comparing the numerical errors of CGM, GMRES, OGSDA, and BBM.

only two steps, CGM with four steps, and the BBM with ten steps.

5.3. Example 3. When we apply a central difference scheme to the following two-point boundary value problem:

$$\begin{aligned} -u''(x) &= f(x), \quad 0 < x < 1, \\ u(0) &= a, \quad u(1) = b, \end{aligned} \quad (81)$$

we can derive a linear equations system

$$\mathbf{C}\mathbf{u} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \quad (82)$$

$$= \begin{bmatrix} (\Delta x)^2 f(\Delta x) + a \\ (\Delta x)^2 f(2\Delta x) \\ \vdots \\ (\Delta x)^2 f((n-1)\Delta x) \\ (\Delta x)^2 f(n\Delta x) + b \end{bmatrix},$$

where $\Delta x = 1/(n+1)$ is the spatial length and $u_i = u(i\Delta x)$, $i = 1, \dots, n$, are unknown values of $u(x)$ at the grid points $x_i = i\Delta x$. $u_0 = a$ and $u_{n+1} = b$ are the given boundary conditions. The above matrix \mathbf{C} is known as a central difference matrix.

The eigenvalues of \mathbf{C} are found to be

$$4 \sin^2 \frac{k\pi}{2(n+1)}, \quad k = 1, 2, \dots, n, \quad (83)$$

which together with the symmetry of \mathbf{C} indicates that \mathbf{C} is positive definite, and

$$\text{Cond}(\mathbf{C}) = \frac{\sin^2(n\pi/2(n+1))}{\sin^2(\pi/2(n+1))} \quad (84)$$

is a large number when the grid number n is very large.

In this numerical test we fix $n = 200$ and thus the condition number of \mathbf{C} is 16210.7. Let us consider the boundary value problem in (81) with $f(x) = \sin \pi x$. The exact solution is

$$u(x) = a + (b-a)x + \frac{1}{\pi^2} \sin \pi x, \quad (85)$$

where we fix $a = 1$ and $b = 2$.

A relative random noise with intensity $\sigma = 0.01$ is added by $\hat{f}(i\Delta x) = f(i\Delta x)[1 + \sigma R(i)]$ into the right-hand side data of (82). Under a moderate convergence criterion with $\varepsilon = 10^{-7}$ for the relative residual, we compare (a) relative residuals and (c) numerical errors of OGSDA, CGM, GMRES, and BBM in Figure 4. The number of iterations of CGM is 1325, and the maximum error is 3.14×10^{-5} . Then we apply the OGSDA with $\gamma = 0.25$ and $m = 30$ to this problem, whose number of iterations is 66, and the maximum error is 1.9×10^{-5} . The number of iterations of GMRES with $m = 10$ is 811, and the maximum error is 4.21×10^{-5} , while the number of iterations of BBM is 1708, and the maximum error is 1.06×10^{-5} .

It can be seen that the OGSDA is more convergent, faster, and more accurate than CGM and GMRES and much faster than the BBM; however, the accuracy of BBM is slightly better than that obtained by the OGSDA.

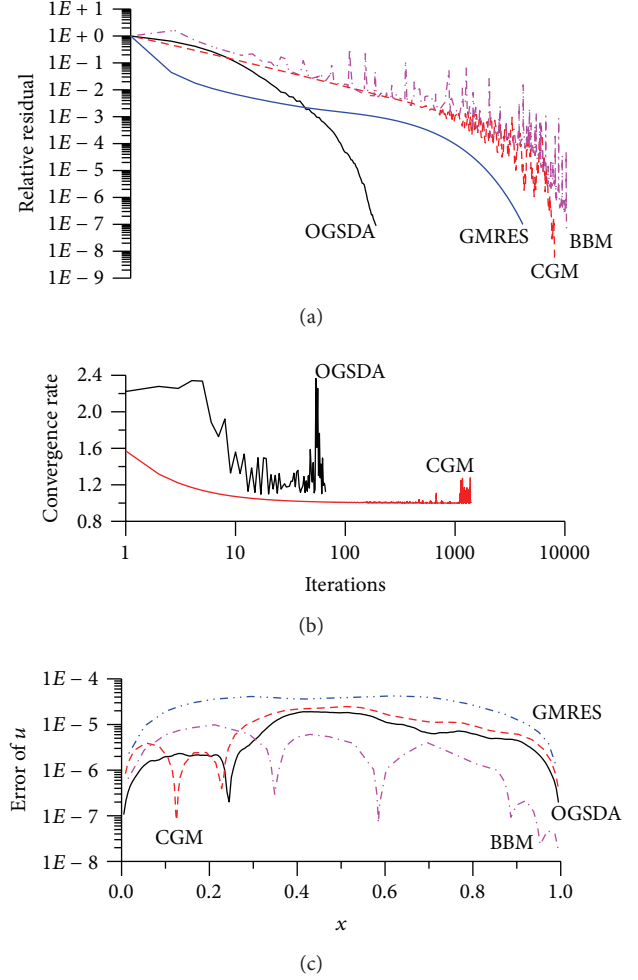


FIGURE 4: Example 3, comparing (a) relative residuals, (b) convergence rates, and (c) numerical errors of CGM, OGSDA, GMRES, and BBM.

In order to demonstrate the efficiency of the present OGSDA with that of the CGM we can compare the convergence rate of these two algorithms. According to Liu [40, 41] the convergence rate is given by

$$\text{Convergence Rate}_{\text{OGSDA}} := \frac{1}{\sqrt{s_k}},$$

$$s_k = 1 - \frac{(1 - \gamma^2) [(\mathbf{C}\mathbf{x}_k - \mathbf{b}) \cdot \mathbf{C}\mathbf{u}^k]^2}{\|\mathbf{C}\mathbf{x}_k - \mathbf{b}\|^2 \|\mathbf{C}\mathbf{u}^k\|^2},$$

$$\text{Convergence Rate}_{\text{CGM}} := \frac{1}{\sqrt{s_k}}, \quad s_k = 1 - \frac{\eta_k}{\|\mathbf{B}\mathbf{x}_k - \mathbf{b}_1\|^2}, \quad (86)$$

where $\eta_k = \|\mathbf{r}_k\|^2 / \mathbf{p}_k^T \mathbf{C} \mathbf{p}_k$, in which \mathbf{p}_k is the supplemental vector used in the CGM. In Figure 4(b) we compare the convergence rates of CGM and OGSDA, of which we can see that the convergence rate of OGSDA is much larger than that of CGM.

5.4. *Example 4.* When the backward heat conduction problem (BHCP) is considered in a spatial interval of $0 < x < \ell$ by subjecting to the boundary conditions at two ends of a slab:

$$u_t(x, t) = \kappa u_{xx}(x, t), \quad 0 < t < T, \quad 0 < x < \ell, \quad (87)$$

$$u(0, t) = u_0(t), \quad u(\ell, t) = u_\ell(t), \quad (88)$$

we solve u under a final time condition:

$$u(x, T) = u^T(x). \quad (89)$$

The fundamental solution of (87) is given by

$$K(x, t) = \frac{H(t)}{2\sqrt{\kappa\pi t}} \exp\left(\frac{-x^2}{4\kappa t}\right), \quad (90)$$

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

The method of fundamental solutions (MFS) has a serious drawback that the resulting linear equations system is always highly ill-conditioned, when the number of source points is increased, or when the distances of source points are increased.

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

$$u(\mathbf{z}) = \sum_{j=1}^n c_j U(\mathbf{z}, \mathbf{s}_j), \quad \mathbf{s}_j = (\eta_j, \tau_j) \in \Omega^c, \quad (91)$$

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

$$U(\mathbf{z}, \mathbf{s}_j) = K(x - \eta_j, t - \tau_j). \quad (92)$$

It is known that the location of source points in the MFS has a great influence on the accuracy and stability. In a practical application of MFS to solve the BHCP, the source points are uniformly located on two vertical straight lines parallel to the t -axis, not over the final time, which was adopted by Hon and Li [42] and Liu [43], showing a large improvement than the line location of source points below the initial time. After imposing the boundary conditions and the final time condition to (91) we can obtain a linear equations system:

$$\mathbf{B}\mathbf{x} = \mathbf{b}_1, \quad (93)$$

where

$$\begin{aligned} B_{ij} &= U(\mathbf{z}_i, \mathbf{s}_j), \quad \mathbf{x} = (c_1, \dots, c_n)^T, \\ \mathbf{b}_1 &= (u_\ell(t_i), i = 1, \dots, m_1; u^T(x_j), \\ &\quad j = 1, \dots, m_2; u_0(t_k), k = m_1, \dots, 1)^T, \end{aligned} \quad (94)$$

and $n = 2m_1 + m_2$.

Since the BHCP is highly ill-posed, the ill-condition of the coefficient matrix \mathbf{B} in (93) is serious. To overcome the ill-posedness of (93) we can use the OGSDA to solve this problem. Here we compare the numerical solution with an exact solution:

$$u(x, t) = \cos(\pi x) \exp(-\pi^2 t). \quad (95)$$

For the case with $T = 1$ the value of final time data is in the order of 10^{-4} , which is small by comparing with the value of the initial temperature $f(x) = u_0(x) = \cos(\pi x)$ to be retrieved, which is $O(1)$. First we impose a relative random noise with an intensity $\sigma = 10\%$ being imposed on the final time data. Under the following parameters $m_1 = 11$, $m_2 = 6$, $m = 10$, $\gamma = 0.25$, and $\varepsilon = \varepsilon_1 = 10^{-5}$, we solve this problem by the OGSDA with the Krylov subspace method. With 28 iterations the OGSDA can obtain a very accurate solution with the maximum error being 9.877×10^{-4} , which is better than 1.247×10^{-3} calculated by Liu [35] using the optimal multivector iterative algorithm (OMVIA). In Figures 5(a) and 5(b) we plot the relative residual and steplength, of which we can see that the minimal steplength is 1. The numerical result is compared with the exact initial value in Figure 5(c), whose numerical error as shown in Figure 5(d) indicates that the present OGSDA is very robust against noise. Under the same parameters we also apply the BBM and GMRES with $m = 10$ to solve the BHCP, of which both algorithms cannot converge to satisfy the convergence criterion with $\varepsilon = 10^{-5}$, and we let BBM run 100 steps and the GMRES run 30 steps with the residuals being shown in Figure 5(a). The maximum error obtained by the BBM is 0.151, while that for the GMRES is 0.809 as shown in Figure 5(d). For this problem we can quickly provide a very accurate numerical result by using the OGSDA, which is much better than the other three algorithms.

5.5. *Example 5.* Let us consider the inverse Cauchy problem for the Laplace equation:

$$\Delta u = u_{rr} + \frac{1}{r} u_r + \frac{1}{r^2} u_{\theta\theta} = 0, \quad (96)$$

$$u(\rho, \theta) = h(\theta), \quad 0 \leq \theta \leq \pi, \quad (97)$$

$$u_n(\rho, \theta) = g(\theta), \quad 0 \leq \theta \leq \pi, \quad (98)$$

where $h(\theta)$ and $g(\theta)$ are given function. The inverse Cauchy problem is specified as follows.

To seek an unknown boundary function $f(\theta)$ on the part $\Gamma_2 := \{(r, \theta) \mid r = \rho(\theta), \pi < \theta < 2\pi\}$ of the boundary under (96)–(98) with the overspecified data being given on $\Gamma_1 := \{(r, \theta) \mid r = \rho(\theta), 0 \leq \theta \leq \pi\}$.

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

$$u(\mathbf{x}) = \sum_{j=1}^n c_j U(\mathbf{x}, \mathbf{s}_j), \quad \mathbf{s}_j \in \Omega^c. \quad (99)$$

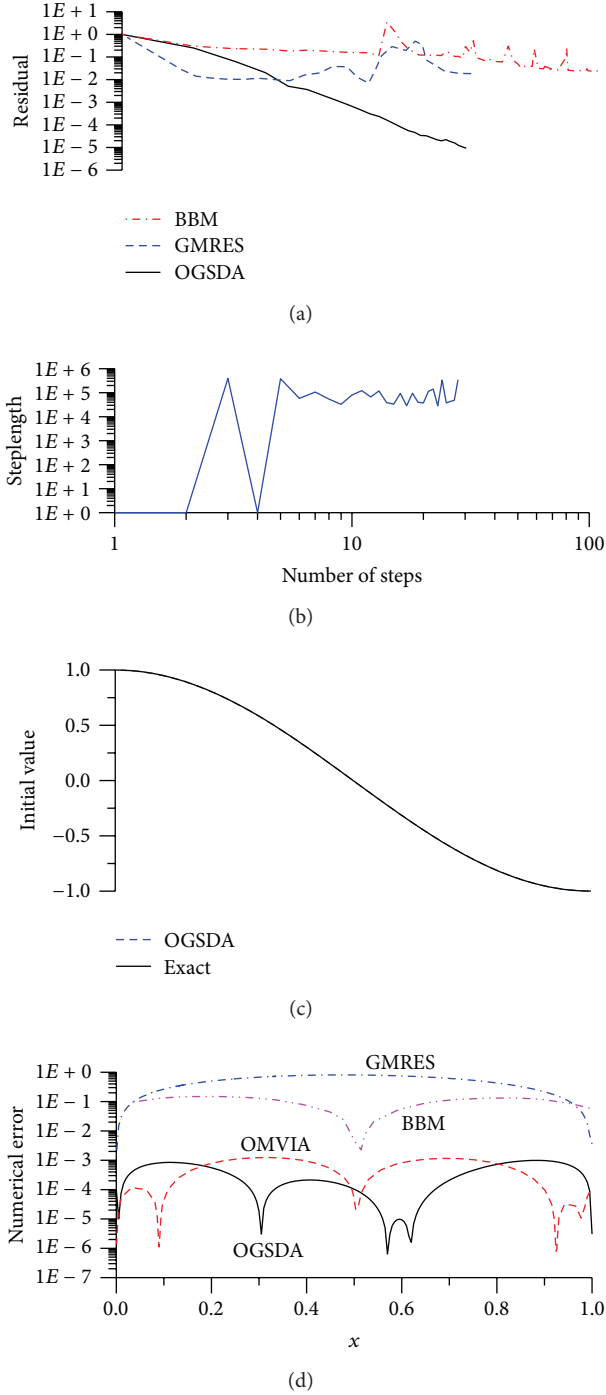


FIGURE 5: Example 4 showing (a) relative residual and (b) steplength of OGSDA, (c) comparing numerical and exact initial value, and (d) comparing numerical errors of OMVIA, OGSDA, BBM, and GMRES.

For the Laplace equation (96) we have the fundamental solutions:

$$U(\mathbf{x}, \mathbf{s}_j) = \ln r_j, \quad r_j = \|\mathbf{x} - \mathbf{s}_j\|. \quad (100)$$

In the practical application of MFS, by imposing the boundary conditions (97) and (98) at N points on (99) we can obtain a linear equations system:

$$\mathbf{B}\mathbf{c} = \mathbf{b}_1, \quad (101)$$

where

$$\mathbf{x}_i = (x_i^1, x_i^2) = (\rho(\theta_i) \cos \theta_i, \rho(\theta_i) \sin \theta_i),$$

$$\mathbf{s}_j = (s_j^1, s_j^2) = (R(\theta_j) \cos \theta_j, R(\theta_j) \sin \theta_j),$$

$$B_{ij} = \ln \|\mathbf{x}_i - \mathbf{s}_j\|, \quad \text{if } i \text{ is odd,}$$

$$B_{ij} = \frac{\eta(\theta_i)}{\|\mathbf{x}_i - \mathbf{s}_j\|^2} \left(\rho(\theta_i) - s_j^1 \cos \theta_i - s_j^2 \sin \theta_i - \frac{\rho'(\theta_i)}{\rho(\theta_i)} \times [s_j^1 \sin \theta_i - s_j^2 \cos \theta_i] \right), \quad \text{if } i \text{ is even,}$$

$$\mathbf{c} = (c_1, \dots, c_n)^T,$$

$$\mathbf{b}_1 = (h(\theta_1), g(\theta_1), \dots, h(\theta_N), g(\theta_N))^T, \quad (102)$$

in which $n = 2N$, and

$$\eta(\theta) = \frac{\rho(\theta)}{\sqrt{\rho^2(\theta) + [\rho'(\theta)]^2}}. \quad (103)$$

The above $R(\theta) = \rho(\theta) + D$ with an offset D can be used to locate the source points along a contour with a radius $R(\theta)$. When the linear equations system (101) is established, we can apply the OGSDA to solve it.

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

$$u(x, y) = \cos x \cosh y + \sin x \sinh y, \quad (104)$$

defined in a domain with a complex amoeba-like irregular shape as a boundary:

$$\rho(\theta) = \exp(\sin \theta) \sin^2(2\theta) + \exp(\cos \theta) \cos^2(2\theta). \quad (105)$$

After imposing the boundary conditions (97) and (98) at N points on (99) we can obtain a linear equations system. The noise being imposed on the measured data h and g is $\sigma = 0.01$.

We solve this problem by the OGSDA with $m = 10$ and $\gamma = 0.2$. Through 15 iterations the residual is smaller than the convergence criterion $\varepsilon = 10^{-5}$ as shown in Figure 6(a), and also the steplength is shown there. We compare the recovered data computed by the OGSDA with the exact one in Figure 6(b). The numerical error as shown in Figure 6(c) is smaller than 0.062. It can be seen that the OGSDA with Krylov subspace can accurately recover the unknown boundary condition.

We also apply the GMRES with $m = 5$ and the BBM to this inverse Cauchy problem, whose residuals are in

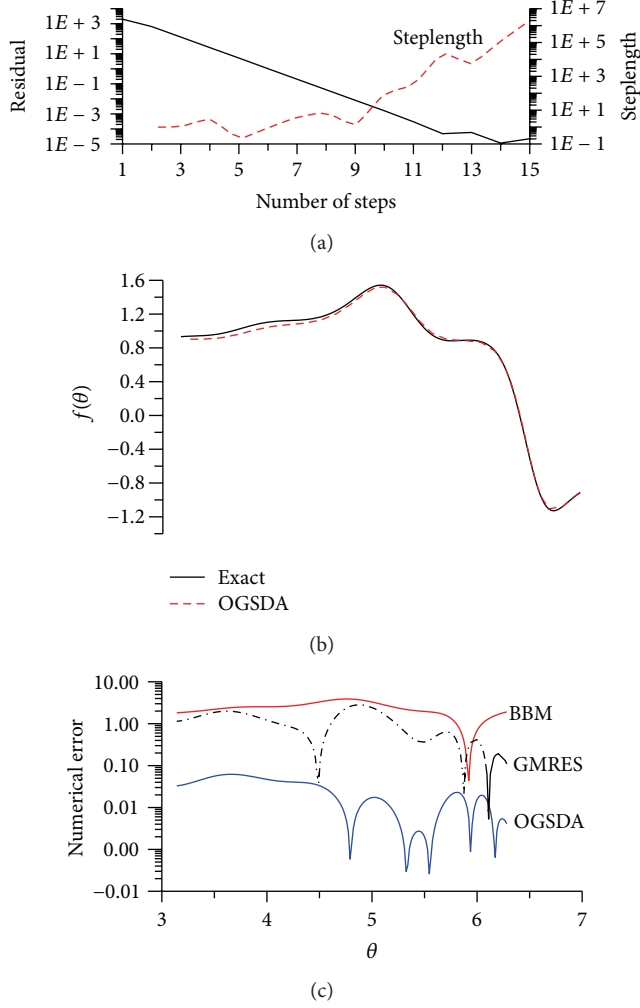


FIGURE 6: Example 5 showing (a) residual and steplength of OGSDA, (b) comparing numerical and exact solutions, and (c) numerical errors of OGSDA, GMRES, and BBM.

Figure 7. They fast tend to a plateau and then the residuals are no more reduced. The numerical solutions are very bad with the maximum error of BBM being 3.79 and the maximum error of GMRES being 2.81. As compared with those obtained by OGSDA, the accuracy and convergence speed of OGSDA are much better than those obtained by the GMRES and the BBM.

5.6. Example 6. In this section we apply the OGSDA to identify an unknown space-dependent heat source function $H(x)$ for a one-dimensional heat conduction equation:

$$u_t(x, t) = u_{xx}(x, t) + H(x), \quad 0 < x < \ell, \quad 0 < t < t_f, \quad (106)$$

$$u(0, t) = u_0(t), \quad u(\ell, t) = u_\ell(t), \quad (107)$$

$$u(x, 0) = f(x). \quad (108)$$

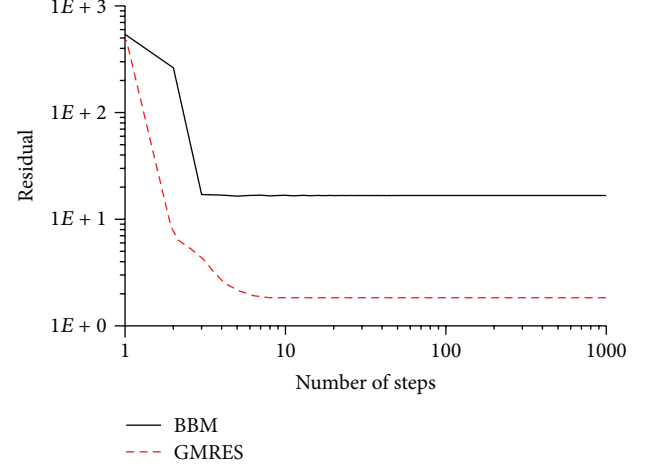


FIGURE 7: Example 5 showing residuals of GMRES and BBM.

In order to identify $H(x)$ we impose a Neumann type boundary condition:

$$u_x(0, t) = q(t). \quad (109)$$

We propose a numerical differential method by letting $v = u_t$. Taking the differentials of (106), (107), and (109) with respect to t and letting $v = u_t$ we can derive

$$v_t(x, t) = v_{xx}(x, t), \quad 0 < x < \ell, \quad 0 < t < t_f, \quad (110)$$

$$v(0, t) = \dot{u}_0(t), \quad (111)$$

$$v(\ell, t) = \dot{u}_\ell(t), \quad (112)$$

$$v_x(0, t) = \dot{q}(t). \quad (113)$$

This is an inverse heat conduction problem (IHCP) for $v(x, t)$ without using the initial condition.

Therefore as being a numerical method, we can first solve the above IHCP for $v(x, t)$ by using the MFS in Section 5.4 to obtain a linear equations system and then the method introduced in Section 4 to solve the resultant linear equations system. Thus, we can construct $u(x, t)$ by

$$u(x, t) = \int_0^t v(x, \xi) d\xi + f(x), \quad (114)$$

which automatically satisfies the initial condition in (108).

From (114) it follows that

$$u_{xx}(x, t) = \int_0^t v_{xx}(x, \xi) d\xi + f''(x), \quad (115)$$

which together with $u_t = v$ being inserted into (106) leads to

$$v(x, t) = \int_0^t v_{xx}(x, \xi) d\xi + f''(x) + H(x). \quad (116)$$

Inserting (110) for $v_{xx} = v_t$ into the above equation and integrating it we can derive the following equation to recover $H(x)$:

$$H(x) = v(x, 0) - f''(x). \quad (117)$$

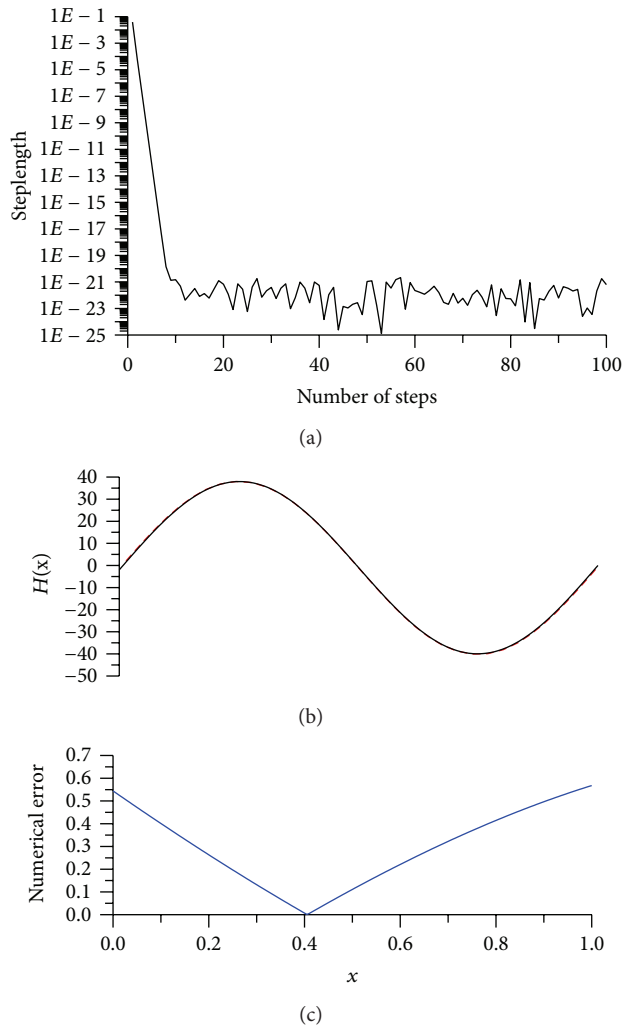


FIGURE 8: Example 6 showing (a) steplength of OGSDA, (b) comparing numerical and exact solutions, and (c) numerical error of OGSDA.

This approach exhibits threefold ill-posedness: one is the use of the IHCP to solve $v(x, t)$ which is used to provide the data of $v(x, 0)$ used in (117), one is all the boundary conditions being obtained from the first-order differentials of measured data as shown in (111)–(113), and another is the second-order differential of the data $f(x)$ in (117).

We consider

$$\begin{aligned} u(x, t) &= x^2 + 2xt + \sin(2\pi x), \\ H(x) &= 2x - 2 + 4\pi^2 \sin(2\pi x). \end{aligned} \quad (118)$$

In (117) we disregard the ill-posedness of $f''(x)$ and suppose that the data $f''(x)$ are given exactly. We solve this problem by the OGSDA with unit subspace method, where we fix $m = 10$ and $\gamma = 0.05$. The maximum number of iterations is set to be 100. A random noise with intensity $\sigma = 0.05$ is added on the data $\hat{q}(t)$. We have obtained the numerical solution with the steplength being shown in Figure 8(a), which is very small. We compare the heat source computed by the OGSDA with

the exact one in Figure 8(b). The numerical error is smaller than 0.568 as shown in Figure 8(c).

6. Conclusions

In the present paper, we have derived an optimal algorithm including an m -vector optimal search direction in an m -dimensional Krylov subspace or a unit subspace as an optimally generalized steepest-descent algorithm (OGSDA) to solve highly ill-posed linear systems. This algorithm has good computational efficiency and accuracy in solving the ill-posed linear equations for the linear Hilbert problem with $n = 300$ and three famous linear inverse problems of backward heat conduction problem, inverse Cauchy problem, and inverse heat source problem. In particular, the OGSDA has a better filtering effect against noise, such that for all the numerical examples of inverse problems the numerical results recovered are quite smooth. The robustness of the OGSDA was confirmed by imposing noisy disturbance on the given data even with an intensity being large up to 10%. In all the numerical examples the convergence steps are among 4 to 100 steps, which is very time saving, no one case running over one second in the PC computer. For highly ill-posed problems the performance of OGSDA without needing of large value m of the Krylov subspace dimension is better than other algorithms investigated in this paper.

Conflict of Interests

This paper is a purely academic research, and the author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The project NSC-102-2221-E-002-125-MY3 and the 2011 Outstanding Research Award from the National Science Council of Taiwan and the 2011 Taiwan Research Front Award from Thomson Reuters are highly appreciated. The author thanks anonymous referee for the comments which improved the quality of this paper.

References

- [1] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.
- [2] M. Raydan, "On the barzilai and borwein choice of steplength for the gradient method," *Journal of Numerical Analysis*, vol. 13, no. 3, pp. 321–326, 1993.
- [3] M. Raydan, "The Barzilaiai and Borwein gradient method for the large scale unconstrained minimization problem," *Journal on Optimization*, vol. 7, no. 1, pp. 26–33, 1997.
- [4] A. Friedlander, J. M. Martinez, B. Molina, and M. Raydan, "Gradient method with retards and generalizations," *Journal on Numerical Analysis*, vol. 36, no. 1, pp. 275–289, 1999.
- [5] M. Raydan and B. F. Svaiter, "Relaxed steepest descent and Cauchy-Barzilai-Borwein method," *Computational Optimization and Applications*, vol. 21, no. 2, pp. 155–167, 2002.

- [6] Y. Dai, J. Yuan, and Y.-X. Yuan, "Modified two-point stepsize gradient methods for unconstrained optimization," *Computational Optimization and Applications*, vol. 22, no. 1, pp. 103–109, 2002.
- [7] Y.-H. Dai and L.-Z. Liao, "R-linear convergence of the Barzilai and Borwein gradient method," *Journal of Numerical Analysis*, vol. 22, no. 1, pp. 1–10, 2002.
- [8] Y.-H. Dai and Y.-X. Yuan, "Alternate minimization gradient method," *Journal of Numerical Analysis*, vol. 23, no. 3, pp. 377–393, 2003.
- [9] R. Fletcher, "On the Barzilai-Borwein Method," in *Optimization and Control with Applications*, L. Qi, K. Teo, and X. Yang, Eds., vol. 96 of *Applied Optimization*, pp. 235–256, Springer, New York, NY, USA, 2005.
- [10] Y.-X. Yuan, "A new stepsize for the steepest descent method," *Journal of Computational Mathematics*, vol. 24, no. 2, pp. 149–156, 2006.
- [11] A. Bhaya and E. Kaszkurewicz, *Control Perspectives on Numerical Algorithms and Matrix Problems*, Advances in Design and Control, Book 10, SIAM, Philadelphia, Pa, USA, 2006.
- [12] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Springer, Berlin, Germany, 1994.
- [13] M. K. Gavurin, "Nonlinear functional equations and continuous analogs of iterative methods," *Izvestiya Vysshikh Uchebnykh Zavedenii Matematika*, vol. 5, pp. 18–31, 1958.
- [14] Y. I. Alber, "Continuous processes of the Newton type," *Differential Equations*, vol. 7, pp. 1461–1471, 1971.
- [15] M. W. Hirsch and S. Smale, "On algorithms for solving $f(x) = 0$," *Communications on Pure and Applied Mathematics*, vol. 32, pp. 281–312, 1979.
- [16] M. T. Chu, "On the continuous realization of iterative processes," *SIAM Review*, vol. 30, no. 3, pp. 375–387, 1988.
- [17] I. M. Ortega and W. C. Rheinboldt, *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, New York, NY, USA, 1970.
- [18] A. Bhaya and E. Kaszkurewicz, "Steepest descent with momentum for quadratic functions is a version of the conjugate gradient method," *Neural Networks*, vol. 17, no. 1, pp. 65–71, 2004.
- [19] A. Bhaya and E. Kaszkurewicz, "A control-theoretic approach to the design of zero finding numerical methods," *IEEE Transactions on Automatic Control*, vol. 52, no. 6, pp. 1014–1026, 2007.
- [20] C.-S. Liu, "A state feedback controller used to solve an ill-posed linear system by a $GL(n, \mathbb{R})$ iterative algorithm," *Communications in Numerical Analysis*, vol. 2013, Article ID cna-00181, 22 pages, 2013.
- [21] U. M. Ascher, K. van den Doel, H. Huang, and B. F. Svaiter, "Gradient descent and fast artificial time integration," *Mathematical Modelling and Numerical Analysis*, vol. 43, no. 4, pp. 689–708, 2009.
- [22] C.-S. Liu, "A revision of relaxed steepest descent method from the dynamics on an invariant manifold," *Computer Modeling in Engineering and Sciences*, vol. 80, no. 1, pp. 57–86, 2011.
- [23] C.-S. Liu, "Modifications of steepest descent method and conjugate gradient method against noise for ill-posed linear systems," *Communications in Numerical Analysis*, vol. 2012, Article ID cna-00115, 24 pages, 2012.
- [24] Y.-H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, "The cyclic Barzilai-Borwein method for unconstrained optimization," *Journal of Numerical Analysis*, vol. 26, no. 3, pp. 604–627, 2006.
- [25] G. Frassoldati, L. Zanni, and G. Zanghirati, "New adaptive stepsize selections in gradient methods," *Journal of Industrial and Management Optimization*, vol. 4, no. 2, pp. 299–312, 2008.
- [26] W. W. Hager, B. A. Mair, and H. Zhang, "An affine-scaling interior-point CBB method for box-constrained optimization," *Mathematical Programming*, vol. 119, no. 1, pp. 1–32, 2009.
- [27] S. Bonettini, R. Zanella, and L. Zanni, "A scaled gradient projection method for constrained image deblurring," *Inverse Problems*, vol. 25, no. 1, Article ID 015002, 2009.
- [28] G. Yu, L. Qi, and Y. Dai, "On nonmonotone chambolle gradient projection algorithms for total variation image restoration," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 143–154, 2009.
- [29] T. Lukić, J. Lindblad, and N. Sladoje, "Regularized image denoising based on spectral gradient optimization," *Inverse Problems*, vol. 27, no. 8, Article ID 085010, 2011.
- [30] S. Setzer, G. Steidl, and J. Morgenthaler, "A cyclic projected gradient method," *Computational Optimization and Applications*, vol. 54, no. 2, pp. 417–440, 2013.
- [31] J. Dongarra and F. Sullivan, "Guest editors introduction to the top 10 algorithms," *Computing in Science & Engineering*, vol. 2, no. 1, pp. 22–23, 2000.
- [32] Y. Saad, "Krylov subspace methods for solving large unsymmetric linear systems," *Mathematics of Computation*, vol. 37, no. 155, pp. 105–126, 1981.
- [33] R. W. Freund and N. M. Nachtigal, "QMR: a quasi-minimal residual method for non-Hermitian linear systems," *Numerische Mathematik*, vol. 60, no. 1, pp. 315–339, 1991.
- [34] J. Van Den Eshof and G. L. G. Sleijpen, "Inexact Krylov subspace methods for linear systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 1, pp. 125–153, 2005.
- [35] C.-S. Liu, "An optimal multi-vector iterative algorithm in a Krylov subspace for solving the ill-posed linear inverse problems," *Computers Materials & Continua*, vol. 33, no. 2, pp. 175–198, 2013.
- [36] V. Simoncini and D. B. Szyld, "Recent computational developments in Krylov subspace methods for linear systems," *Numerical Linear Algebra with Applications*, vol. 14, no. 1, pp. 1–59, 2007.
- [37] Y. Saad and M. H. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 856–869, 1986.
- [38] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, Pa, USA, 2nd edition, 2003.
- [39] C.-S. Liu, H.-K. Hong, and 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," *Computer Modeling in Engineering and Sciences*, vol. 60, no. 3, pp. 279–308, 2010.
- [40] C.-S. Liu, "An optimal tri-vector iterative algorithm for solving ill-posed linear inverse problems," *Inverse Problems in Science and Engineering*, vol. 21, no. 4, pp. 650–681, 2013.
- [41] C.-S. Liu, "A dynamical Tikhonov regularization for solving ill-posed linear algebraic systems," *Acta Applicandae Mathematicae*, vol. 123, no. 1, pp. 285–307, 2013.
- [42] Y. C. Hon and M. Li, "A discrepancy principle for the source points location in using the MFS for solving the BHCP," *Inte-*

International Journal of Computational Methods, vol. 6, no. 2, pp. 181–197, 2009.

- [43] C.-S. Liu, “The method of fundamental solutions for solving the backward heat conduction problem with conditioning by a new post-conditioner,” *Numerical Heat Transfer, Part B*, vol. 60, no. 1, pp. 57–72, 2011.