CMC, vol.33, no.2, pp.175-198, 2013

An Optimal Multi-Vector Iterative Algorithm in a Krylov Subspace for Solving the Ill-Posed Linear Inverse Problems

Chein-Shan Liu[1]

Abstract: An optimal m-vector descent iterative algorithm in a Krylov subspace is developed, of which the m weighting parameters are optimized from a properly defined objective function to accelerate the convergence rate in solving an ill-posed linear problem. The optimal multi-vector iterative algorithm (OMVIA) is convergent fast and accurate, which is verified by numerical tests of several linear inverse problems, including the backward heat conduction problem, the heat source identification problem, the inverse Cauchy problem, and the external force recovery problem. Because the OMVIA has a good filtering effect, the numerical results recovered are quite smooth with small error, even under a large noise up to 10%.

Keywords: Linear inverse problems, Ill-posed linear equations system, Optimal multi-vector iterative algorithm (OMVIA), Future cone, Invariant-manifold, Krylov subspace method

1 Introduction

The iterative algorithm for solving algebraic equations can be derived from the discretization of a certain ordinary differential equations (ODEs) system [Bhaya and Kaszkurewicz (2006); Chehab and Laminie (2005); Liu and Atluri (2008)]. Particularly, some descent methods can be interpreted as the discretizations of gradient flows [Helmke and Moore (1994)]. For a large scale system the major choice is using an iterative algorithm, where an early stopping criterion is used to prevent the reconstruction of noisy component in the approximate solution. The author and his coworkers have developed several methods to solve the ill-posed system of linear algebraic equations, like using the fictitious time integration method as a filter [Liu and Atluri (2009a)], a modified polynomial expansion method [Liu and Atluri (2009b)], the Laplacian preconditioners and postconditioners [Liu, Yeih and Atluri

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

(2009)], a vector regularization method [Liu, Hong and Atluri (2010)], a relaxed steepest descent method [Liu (2011a, 2012a)], an optimal iterative algorithm with an optimal descent vector [Liu and Atluri (2011a)], the best vector iterative method [Liu (2012b)], the globally optimal vector iterative method [Liu (2012c)], the optimally scaled vector regularization method [Liu (2012d)], an optimally generalized Tikhonov regularization method [Liu (2012e)], an adaptive Tikhonov regularization method [Liu (2013a)], as well as an optimal tri-vector iterative algorithm [Liu (2013b)].

There are regularization method, the preconditioner or postconditioner method, the iterative method, and the combination of these methods to solve the ill-posed linear problem. In this paper we will develop an iterative algorithm with a descent vector spanned in a Krylov subspace and with an inner iteration to determine the optimized weighting parameters to solve the following ill-posed linear equations system:

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

where $\mathbf{x} \in \mathbb{R}^n$ is an unknown vector, to be determined from a given coefficient matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, which might be unsymmetric, and the input $\mathbf{b} \in \mathbb{R}^n$, which might be disturbed by random noise. The linear inverse problems are usually being converted into the above form.

There are a lot of numerical methods that converge significantly faster than the steepest descent method (SDM), and unlike the conjugate gradient method (CGM), they insist their search directions to be the gradient vector at each iteration [Barzilai and Borwein (1988)]. The SDM performs poorly, yielding the iteration counts that grow linearly with Cond($\mathbf{B}$) [Akaike (1959)], whose unwelcome slowness has to do with the choice of the gradient descent direction $\mathbf{R}$. Liu (2013b) has explored a variant of the SDM by feding the concept of an optimal descent tri-vector to solve the ill-posed linear equations system, which is an optimal combination of the steepest descent vector $\mathbf{R}$, the residual vector $\mathbf{r}$ and a supplemental vector $\mathbf{BR}$, where not only the direction $\mathbf{R}$ but also the steplength are modified from a theoretical foundation of optimization being realized on an invariant manifold. This novel method was performed better than the generalized minimal residual method (GMRES) [Saad and Schultz (1986)], the CGM, and other gradient descent variant methods. The concept of *optimal vector driving algorithm* was first developed by Liu and Atluri (2011b, 2012) for solving nonlinear equations, and then by Liu and Atluri (2011a), and Liu (2012b, 2012c) for solving linear equations. As a continuation of these efforts, we further explore the concept of optimal iterative algorithm with an m-vector in a Krylov subspace as a descent direction to solve Eq. (1). In the solution of linear equations system the Krylov subspace method is one of the most important classes of numerical methods [Dongarra and Sullivan (2000)]. The iterative algorithms that are applied to solve large-scale linear algebraic systems are mostly the preconditioned Krylov subspace methods [Simoncini and Szyld (2007)].

The remaining parts of this paper are arranged as follows. In Section 2 we start from an invariant-manifold of a future cone in the Minkowski space and a Krylov subspace method to derive a nonlinear system of ODEs for the numerical solution of Eq. (1). Then, a genuine dynamics on the invariant-manifold is constructed in Section 3, resulting in an m-vector optimal descent algorithm in terms of m weighting parameters which to be optimized explicitly from a simple objective function. The numerical examples of linear inverse problems are given in Section 4 to display some advantages of the present optimal multi-vector iterative algorithm (OMVIA). Finally, the conclusions are drawn in Section 5.

2 The Krylov subspace method

For the linear equations system (1), which is expressed to be $\mathbf{r} = \mathbf{0}$ in terms of the residual vector:

$$\mathbf{r} = \mathbf{B}\mathbf{x} - \mathbf{b}, \tag{2}$$

we can introduce a scalar homotopy function:

$$h(\mathbf{x},t) = \frac{Q(t)}{2}\|\mathbf{r}(\mathbf{x})\|^2 - \frac{1}{2}\|\mathbf{r}_0\|^2 = 0, \tag{3}$$

where $Q(t) > 0$ is a monotonically increasing function of t, $\mathbf{x}_0$ is an initial value of $\mathbf{x}$, and $\mathbf{r}_0 = \mathbf{r}(\mathbf{x}_0)$. In terms of

$$\mathbf{X} = \begin{bmatrix} \frac{\mathbf{r}}{\|\mathbf{r}_0\|} \\ \frac{1}{\sqrt{Q(t)}} \end{bmatrix}, \tag{4}$$

Eq. (3) represents a cone:

$$\mathbf{X}^{\mathrm{T}}\mathbf{g}\mathbf{X} = 0 \tag{5}$$

in the Minkowski space $\mathbb{M}^{n+1}$, endowed with an indefinite Minkowski metric tensor:

$$\mathbf{g} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{1\times n} & -1 \end{bmatrix}. \tag{6}$$

Because the last component $1/\sqrt{Q(t)}$ of $\mathbf{X}$ is positive, the cone in Eq. (5) is a future cone [Liu (2001)].

When $Q > 0$ and $Q(t) \in \mathbb{C}^1[0,\infty)$, the manifold defined by Eq. (3) is continuous and differentiable, and thus, as a consequence of the "consistency condition", we have

$$\frac{1}{2}\dot{Q}(t)\|\mathbf{r}(\mathbf{x})\|^2 + Q(t)\mathbf{R}\cdot\dot{\mathbf{x}} = 0, \tag{7}$$

which is obtained by taking the differential of Eq. (3) with respect to t and considering $\mathbf{x} = \mathbf{x}(t)$ and $h(\mathbf{x}(t),t) = 0$ for all t. Corresponding to the residual vector $\mathbf{r}$ in Eq. (2), the above

$$\mathbf{R} := \mathbf{B}^{\mathrm{T}}\mathbf{r} \tag{8}$$

is the steepest descent vector.

We suppose that the evolution of $\mathbf{x}$ is governed by a vector $\mathbf{u}$:

$$\dot{\mathbf{x}} = \lambda\mathbf{u}, \tag{9}$$

where the descent vector

$$\mathbf{u} = \sum_{k=1}^{m} \alpha_k \mathbf{u}_k \tag{10}$$

to be optimized, is a suitable combination of the m-vector $\mathbf{u}_k$, $k = 1,\ldots,m$, while the coefficients α_k are to be optimized in Section 3.3.

Now we describe how to set up the m-vector $\mathbf{u}_k$, $k = 1,\ldots,m$ by the Krylov subspace method and the Arnoldi procedure. Suppose that we have an m-dimensional Krylov subspace generated by the coefficient matrix $\mathbf{B}$ from the steepest descent vector $\mathbf{R}$:

$$\mathscr{K}_m := \text{span}\{\mathbf{R},\mathbf{BR},\ldots,\mathbf{B}^{m-1}\mathbf{R}\}. \tag{11}$$

Then the Arnoldi procedure is used to set up the m-vector $\mathbf{u}_k$, $k = 1,\ldots,m$, which uses the Gram-Schmidt orthogonalization technique, such that $\mathbf{u}_k \in \mathscr{K}_m$, $k = 1,\ldots,m$, and

$$\mathbf{u}_i \cdot \mathbf{u}_j = \delta_{ij}, \tag{12}$$

where δ_{ij} is the Kronecker delta symbol.

Inserting Eq. (9) into Eq. (7) we can derive

$$\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{r}\|^2}{\mathbf{r}^{\mathrm{T}}\mathbf{v}}\mathbf{u}, \tag{13}$$

where

$$\mathbf{v} := \mathbf{B}\mathbf{u} = \sum_{k=1}^{m} \alpha_k \mathbf{v}_k = \sum_{k=1}^{m} \alpha_k \mathbf{B}\mathbf{u}_k, \tag{14}$$

$$q(t) := \frac{\dot{Q}(t)}{2Q(t)}. \tag{15}$$

Hence, in our algorithm if $Q(t)$ can be guaranteed to be a monotonically increasing function of t, we might have an absolutely convergent property in solving the linear equations system (1):

$$\|\mathbf{r}(\mathbf{x})\|^2 = \frac{C}{Q(t)}, \tag{16}$$

where

$$C = \|\mathbf{r}(\mathbf{x}_0)\|^2 \tag{17}$$

is determined by the initial value $\mathbf{x}_0$. In the cone defined by Eq. (5), we can observe that the path of $\mathbf{X}$ traces on the cone and gradually moves down to the vertex point along the cone.

We do not need to specify the function $Q(t)$ a priori, of which $\sqrt{C/Q(t)}$ merely acts as a measure of the residual error of $\mathbf{r}$ in time. In Section 3.2 we can compute Q at each iteration step. Hence, we impose in our algorithm that $Q(t) > 0$ is a monotonically increasing function of t. When t increases to a large value, the above equation (16) will enforce the residual error $\|\mathbf{r}(t)\|$ tending to zero, and meanwhile the solution of Eq. (1) is obtained approximately.

3 Dynamics on the invariant-manifold

3.1 Keeping x *on the invariant manifold*

Now we discretize the foregoing continuous dynamics (13) into a discrete time dynamics by applying the forward Euler scheme:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \beta \frac{\|\mathbf{r}\|^2}{\mathbf{r}^{\mathrm{T}}\mathbf{v}} \mathbf{u}, \tag{18}$$

where

$$\beta = q(t)\Delta t \tag{19}$$

is a discretized stepsize. Correspondingly, $\mathbf{u}$ is a search direction of the algorithm endowed with a steplength $\beta\|\mathbf{r}\|^2/(\mathbf{r}^{\mathrm{T}}\mathbf{v})$.

In order to keep $\mathbf{x}$ on the manifold (16) we can consider the evolution of $\mathbf{r}$ along the path $\mathbf{x}(t)$ by

$$\dot{\mathbf{r}} = \mathbf{B}\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{r}\|^2}{\mathbf{r}^{\mathrm{T}}\mathbf{v}}\mathbf{v}. \tag{20}$$

Similarly, we use the forward Euler scheme to integrate Eq. (20), obtaining

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) - \beta\frac{\|\mathbf{r}\|^2}{\mathbf{r}^{\mathrm{T}}\mathbf{v}}\mathbf{v}, \tag{21}$$

from which by taking the square-norms of both sides and using Eq. (16) we can obtain

$$\frac{C}{Q(t+\Delta t)} = \frac{C}{Q(t)} - 2\beta\frac{C}{Q(t)} + \beta^2\frac{C}{Q(t)}\frac{\|\mathbf{r}\|^2}{(\mathbf{r}^{\mathrm{T}}\mathbf{v})^2}\|\mathbf{v}\|^2. \tag{22}$$

Thus the following scalar equation is derived:

$$a_0\beta^2 - 2\beta + 1 - \frac{Q(t)}{Q(t+\Delta t)} = 0, \tag{23}$$

where

$$a_0 := \frac{\|\mathbf{r}\|^2\|\mathbf{v}\|^2}{(\mathbf{r}^{\mathrm{T}}\mathbf{v})^2} \geq 1 \tag{24}$$

by using the Cauchy-Schwarz inequality:

$$\mathbf{r}^{\mathrm{T}}\mathbf{v} \leq \|\mathbf{r}\|\|\mathbf{v}\|.$$

As a result of Eq. (23), $h(\mathbf{x},t) = 0$, $t \in \{0,1,2,\ldots\}$ remains to be an invariant-manifold in the space-time domain $(\mathbf{x},t)$ for the discrete time dynamical system $h(\mathbf{x}(t),t) = 0$.

3.2 An iterative dynamics

Let

$$s = \frac{Q(t)}{Q(t+\Delta t)} = \frac{\|\mathbf{r}(\mathbf{x}(t+\Delta t))\|^2}{\|\mathbf{r}(\mathbf{x}(t))\|^2}, \tag{25}$$

which is an important quantity to assess the convergent property of our numerical algorithm for solving the linear equations system (1).

From Eqs. (23) and (25) it follows that

$$a_0\beta^2 - 2\beta + 1 - s = 0; \tag{26}$$

hence, by setting

$$s = 1 - \frac{1-\gamma^2}{a_0}, \tag{27}$$

we can take a preferred solution of β to be

$$\beta = \frac{1-\gamma}{a_0}. \tag{28}$$

Consequently, from Eqs. (18), (24) and (28) we can derive the following algorithm:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - (1-\gamma)\frac{\mathbf{r}^{\mathrm{T}}\mathbf{v}}{\|\mathbf{v}\|^2}\mathbf{u}, \tag{29}$$

where

$$0 \le \gamma < 1 \tag{30}$$

is a parameter.

Under conditions (24) and (30), from Eqs. (25) and (27) we can prove that the new algorithm satisfies

$$\frac{\|\mathbf{r}(t+\Delta t)\|}{\|\mathbf{r}(t)\|} = \sqrt{s} < 1, \tag{31}$$

which means that the residual error is absolutely decreased. In other words, the convergence rate of present iterative algorithm is

$$\text{Convergence Rate} := \frac{\|\mathbf{r}(t)\|}{\|\mathbf{r}(t+\Delta t)\|} = \frac{1}{\sqrt{s}} > 1. \tag{32}$$

The property in Eq. (32) is very important, since it guarantees that the new algorithm is absolutely convergent to the true solution. *A smaller s will lead to a faster convergence.*

When s is determined by Eq. (27), from Eq. (25), Q can be sequentially computed by

$$Q(t+\Delta t) = \frac{Q(t)}{s} = \frac{a_0 Q(t)}{a_0 - 1 + \gamma^2}, \tag{33}$$

with $Q(0) = 1$. Furthermore, by Eq. (31) we have

$$Q(t+\Delta t) > Q(t),$$

which means that $Q(t) \longrightarrow \infty$, and by Eq. (16) leads to $\|\mathbf{r}(t)\|^2 \longrightarrow 0$. Hence, we can obtain the true solution.

3.3 Optimization of α_k

In algorithm (29) we not yet specify how to choose the m weighting parameters α_k which appear in the vector $\mathbf{v}$ defined by Eq. (14). They can be determined such that a_0 defined by Eq. (24), hence s defined by Eq. (27), are minimized with respect to α_k, because a smaller s will lead to a larger convergence rate as shown in Eq. (32). Therefore, we come to a minimization problem:

$$\min_{\alpha_1,\ldots,\alpha_m} \left\{ a_0 = \frac{\|\mathbf{r}\|^2 \|\mathbf{v}\|^2}{(\mathbf{r}\cdot\mathbf{v})^2} \right\}, \tag{34}$$

which however would lead to a set of m highly nonlinear algebraic equations to solve α_k. This is a quite difficult task to obtain the solution of α_k.

Instead of Eq. (34), we can consider a simpler minimization problem by

$$\min_{\alpha_1,\ldots,\alpha_m} \{P = \|\mathbf{v}-\mathbf{r}\|^2\}, \tag{35}$$

where the objective function P can be written out explicitly:

$$P = \|\sum_{k=1}^m \alpha_k \mathbf{v}_k - \mathbf{r}\|^2. \tag{36}$$

By taking $\partial P/\partial \alpha_k = 0$, $k = 1,\ldots,m$ we can derive the following equation to solve α_k, $k = 1,\ldots,m$:

$$\begin{bmatrix} \mathbf{v}_1\cdot\mathbf{v}_1 & \mathbf{v}_2\cdot\mathbf{v}_1 & \cdots & \mathbf{v}_m\cdot\mathbf{v}_1 \\ \mathbf{v}_1\cdot\mathbf{v}_2 & \mathbf{v}_2\cdot\mathbf{v}_2 & \cdots & \mathbf{v}_m\cdot\mathbf{v}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_1\cdot\mathbf{v}_m & \mathbf{v}_2\cdot\mathbf{v}_m & \cdots & \mathbf{v}_m\cdot\mathbf{v}_m \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \mathbf{r}\cdot\mathbf{v}_1 \\ \mathbf{r}\cdot\mathbf{v}_2 \\ \vdots \\ \mathbf{r}\cdot\mathbf{v}_m \end{bmatrix}. \tag{37}$$

This is a linear equations system with an $m \times m$ coefficient matrix, and with m unknown α_k, $k = 1,\ldots,m$. We can apply the CGM to solve the above system, which is convergent very fast.

Remark: The best choice of $\mathbf{u}$ would be $\mathbf{u} = \mathbf{B}^{-1}\mathbf{r}$, which by Eq. (14) leads to $\mathbf{v} = \mathbf{r}$, and by Eq. (24) further leads to the smallest value of $a_0 = 1$. However, if one has such the best choice of $\mathbf{u} = \mathbf{B}^{-1}\mathbf{r}$ at hand, Eq. (1) is already solved by $\mathbf{x} = \mathbf{B}^{-1}\mathbf{b}$. Instead of letting $\mathbf{v} = \mathbf{r}$, in the present formulation we let $\mathbf{v}$ approach $\mathbf{r}$ by a minimization in Eq. (35); hence, we can obtain the minimizations of a_0 and s by a simpler method. Although we require a few inner iterations to determine the weighting coefficients α_k, the total number of iterations in solving the ill-posed linear problem can be significiantly reduced. This makes the OMVIA superior than other iterative algorithms even they do not need an inner iteration.

3.4 An optimal multi-vector iterative algorithm

Since the time-like variable is discretized to be $t \in \{0,1,2,\ldots\}$, we can let $\mathbf{x}_k$ denote the numerical value of $\mathbf{x}$ at the k-th step. Thus, we arrive at a purely iterative algorithm by Eq. (29), which is labelled as the *optimal multi-vector iterative algorithm* (OMVIA):

(i) Select m and $0 \le \gamma < 1$, and give an initial $\mathbf{x}_0$.

(ii) For $k = 0,1,2,\ldots$, we repeat the following computations:

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

$$\mathbf{R}_k = \mathbf{B}^{\mathrm{T}}\mathbf{r}_k,$$

Arnoldi procedure to set up $\mathbf{u}_j,\ \mathbf{v}_j = \mathbf{B}\mathbf{u}_j,\ j = 1,\ldots,m,$

CGM to solve Eq. (37), obtaining $\alpha_j,\ j = 1,\ldots,m,$

$$\mathbf{u}^k = \sum_{j=1}^{m} \alpha_j \mathbf{u}_j,$$

$$\mathbf{v}^k = \mathbf{B}\mathbf{u}^k,$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma)\frac{\mathbf{r}_k \cdot \mathbf{v}^k}{\|\mathbf{v}^k\|^2}\mathbf{u}^k. \tag{38}$$

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 number of k is for the count of outer iterations; however, in the solution of α_k by the CGM, there is also a number for the count of inner iterations. Below we will take a count of both the numbers of outer iterations and inner iterations together as a total number of iterations.

4 Numerical examples of linear inverse problems

In order to evaluate the performance of the optimal multi-vector iterative algorithm (OMVIA), we test four well-known ill-posed linear inverse problems of the backward heat conduction problem, the heat source identification problem, the inverse Cauchy problem, and the external force recovery problem.

4.1 Backward heat conduction problem

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:

$$u_t(x,t) = \kappa u_{xx}(x,t), \;\; 0 < t < T, \;\; 0 < x < \ell, \tag{39}$$

$$u(0,t) = u_0(t), \;\; u(\ell,t) = u_\ell(t), \tag{40}$$

we solve u under a final time condition:

$$u(x,T) = u^T(x). \tag{41}$$

The fundamental solution to Eq. (39) is given as follows:

$$K(x,t) = \frac{H(t)}{2\sqrt{\kappa\pi t}} \exp\left(\frac{-x^2}{4\kappa t}\right), \tag{42}$$

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 $U(\mathbf{z},\mathbf{s}_j)$:

$$u(\mathbf{z}) = \sum_{j=1}^{n} c_j U(\mathbf{z},\mathbf{s}_j), \;\; \mathbf{s}_j = (\eta_j, \tau_j) \in \Omega^c, \tag{43}$$

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

$$U(\mathbf{z},\mathbf{s}_j) = K(x-\eta_j, t-\tau_j). \tag{44}$$

It is known that the location of source points in the MFS has a great influence on the accuracy and stability. In a practical application of MFS to solve the BHCP, the source points are uniformly located on two vertical straight lines parallel to the t-axis, not over the final time, which was adopted by Hon and Li (2009) and Liu (2011b), showing a large improvement than the line location of source points below the initial time. After imposing the boundary conditions and the final time condition to Eq. (43) we can obtain a linear equations system:

$$\mathbf{B}\mathbf{x} = \mathbf{b}, \tag{45}$$

where

$$\begin{aligned} &B_{ij} = U(\mathbf{z}_i, \mathbf{s}_j), \ \ \mathbf{x} = (c_1, \cdots, c_n)^{\mathrm{T}}, \\ &\mathbf{b} = (u_\ell(t_i),\ i = 1, \ldots, m_1; u^T(x_j),\ j = 1, \ldots, m_2; u_0(t_k),\ k = m_1, \ldots, 1)^{\mathrm{T}}, \end{aligned} \tag{46}$$

and $n = 2m_1 + m_2$.

Example 1: Since the BHCP is highly ill-posed, the ill-condition of the coefficient matrix $\mathbf{B}$ in Eq. (45) is serious. To overcome the ill-posedness of Eq. (45) we can use the OMVIA to solve this problem. Here we compare the numerical solution with an exact solution:

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

For the case with $T = 1$ the value of final time data is in the order of 10^{-4}, which is small by comparing with the value of the initial temperature $f(x) = u_0(x) = \cos(\pi x)$ to be retrieved, which is $O(1)$. We solve this problem by the OMVIA with $m = 3$ and $\gamma = 0.05$. Through totally 7524 iterations (including the inner iterations) the residual norm is smaller than 0.01 as shown in Fig. 1(a), where the value of a_0 is also shown. Under a relative random noise with an intensity $\sigma = 10\%$ being imposed on the final time data, we compare the initial time data computed by the OMVIA with the exact one $u(x,0) = \cos(\pi x)$ in Fig. 1(b). The numerical error as shown in Fig. 1(c) is smaller than 0.001247. It indicates that the present iterative algorithm is very robust against noise, and we can provide a very accurate numerical result by using the OMVIA. For the purpose of comparison, in Fig. 1(c) we also plot the numerical error computed by the optimally generalized Tikhonov regularization method (OGTRM) [Liu (2012e)], of which we can see that the OMVIA is more accurate than the OGTRM about one order.

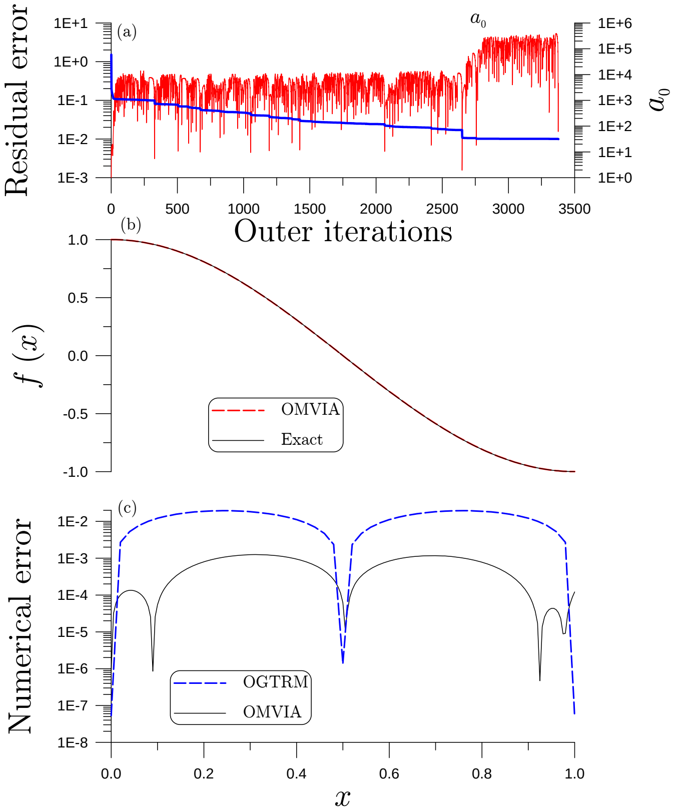

Figure 1: For example 1: (a) showing residual error and a_0, (b) comparing the numerical and exact solutions, and (c) showing the numerical error of recovered initial condition.

4.2 Heat source identification problem

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

$$u_t(x,t) = u_{xx}(x,t) + H(x),\ 0 < x < \ell,\ 0 < t < t_f, \tag{47}$$

$$u(0,t) = u_0(t),\ \ u(\ell,t) = u_\ell(t), \tag{48}$$

$$u(x,0) = f(x). \tag{49}$$

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

$$u_x(0,t) = q(t). \tag{50}$$

We propose a numerical differential method by letting $v = u_t$. Taking the differentials of Eqs. (47) and (48) and (50) with respect to t, and letting $v = u_t$ we can derive

$$v_t(x,t) = v_{xx}(x,t),\ 0 < x < \ell,\ 0 < t < t_f, \tag{51}$$

$$v(0,t) = \dot{u}_0(t), \tag{52}$$

$$v(\ell,t) = \dot{u}_\ell(t), \tag{53}$$

$$v_x(0,t) = \dot{q}(t). \tag{54}$$

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 4.1 to obtain a linear equations system and then the method introduced in Section 3.4 to solve the resultant linear equations system. Thus, we can construct $u(x,t)$ by

$$u(x,t) = \int_0^t v(x,\xi)d\xi + f(x), \tag{55}$$

which automatically satisfies the initial condition in Eq. (49).

From Eq. (55) it follows that

$$u_{xx}(x,t) = \int_0^t v_{xx}(x,\xi)d\xi + f''(x), \tag{56}$$

which together with $u_t = v$ being inserted into Eq. (47), leads to

$$v(x,t) = \int_0^t v_{xx}(x,\xi)d\xi + f''(x) + H(x). \tag{57}$$

Inserting Eq. (51) for $v_{xx} = v_t$ into the above equation and integrating it we can derive the following equation to recover $H(x)$:

$$H(x) = v(x,0) - f''(x). \tag{58}$$

This approach exhibits three-fold ill-posednesses: 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 Eq. (58), one is all the boundary conditions being obtained from the first-order differentials of measured data as shown in Eqs. (52)-(54), and another is the second-order differential of the data $f(x)$ in Eq. (58).

However, we can prove the following unique theorem for the recovery of $H(x)$.

Theorem 1: The reconstruction of $H(x)$ in Eqs. (47)-(50) is unique.

Proof: Suppose that $u_0 = u_\ell = f = q = 0$ in Eqs. (48)-(50), we need to verify $H = 0$ in Eq. (47). By the transformation $v = u_t$, $v(x,t)$ satisfies Eq. (51) with zero overspecified boundary conditions. By the uniqueness of continuation of heat conduction equation we know that $v = 0$ in $0 < x < \ell$, $0 < t < t_f$; hence, we have $u_t = 0$ and $u(x,t) = c$ being a constant, and by $u(x,0) = 0$ we have $c = 0$. This ends the proof. □

Here, as that in Section 4.1 we can apply the MFS to solve Eqs. (51)-(54), of which by the collocation to satisfy the overspecified boundary conditions we can derive a linear equations system.

Example 2: To compare our numerical result with that obtained by Farcas and Lesnic (2006), and Yang, Deng, Yu and Luo (2009) for the recovery of a spatially-dependent heat source, we consider

$$\begin{aligned} &u(x,t) = \sin(\pi x)(2 - e^{-\pi^2 t}), \\ &H(x) = 2\pi^2 \sin(\pi x). \end{aligned} \tag{59}$$

In Eq. (58) we disregard the ill-posedness of $f''(x)$, and suppose that the data $f''(x)$ are given exactly. We solve this problem by the OMVIA with $m = 16$ and $\gamma = 0.05$. The maximum number of outer iterations is set to be 40, or the convergence criterion is taken to be $\varepsilon = 0.1$. A random noise with an intensity $\sigma = 0.05$ is added on the data $\dot{q}(t)$. Through totally 1346 iterations we can find a solution, of which the residual error is shown in Fig. 2(a), where the value of a_0 is also shown. We compare the heat source computed by the OMVIA with the exact one in Fig. 2(b). The numerical error is smaller than 0.06 as shown in Fig. 2(c). The iterative algorithm OMVIA has provided a rather accurate numerical result, even a 5% noise is added on the measured data $\dot{q}(t)$. The present result is better than that obtained by Farcas and Lesnic (2006), and Yang, Deng, Yu and Luo (2009).

Example 3: Then we consider

$$\begin{aligned} &u(x,t) = x^2 + 2xt + \sin(2\pi x), \\ &H(x) = 2x - 2 + 4\pi^2 \sin(2\pi x). \end{aligned} \tag{60}$$

Similarly, in Eq. (58) we disregard the ill-posedness of $f''(x)$, and suppose that the data $f''(x)$ are given exactly. We solve this problem by the OMVIA with $m = 10$ and $\gamma = 0.05$. The maximum number of outer iterations is set to be 500, or the convergence criterion is taken to be $\varepsilon = 0.1$. A random noise with intensity $\sigma = 0.05$

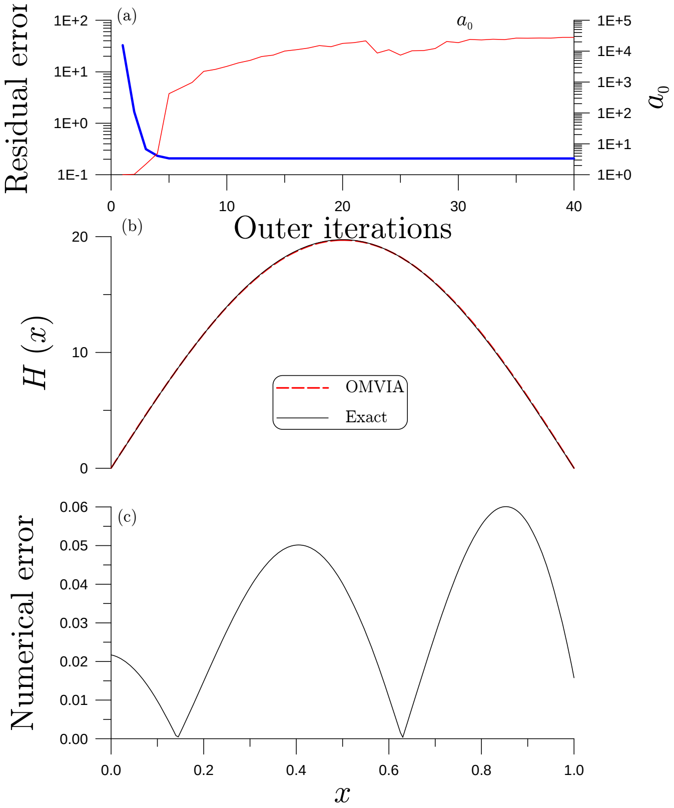

Figure 2: For example 2: (a) showing residual error and a_0, (b) comparing the numerical and exact solutions, and (c) showing the numerical error of recovered heat source.

is added on the data $\dot{q}(t)$. Through totally 7036 iterations we have obtained the numerical solution with the residual error being shown in Fig. 3(a), where the value of a_0 is also shown. We compare the heat source computed by the OMVIA with the exact one in Fig. 3(b). The numerical error is smaller than 0.052 as shown in Fig. 3(c).

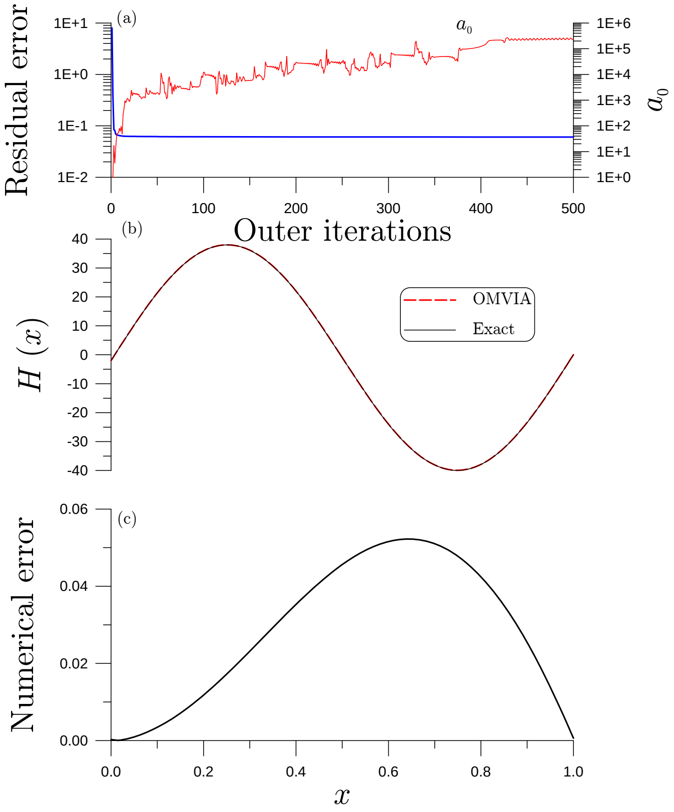

Figure 3: For example 3: (a) showing residual error and a_0, (b) comparing the numerical and exact solutions, and (c) showing the numerical error of recovered heat source.

4.3 *Inverse Cauchy problem*

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

$$\Delta u = u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta} = 0, \tag{61}$$

$$u(\rho,\theta) = h(\theta), \quad 0 \le \theta \le \pi, \tag{62}$$

$$u_n(\rho,\theta) = g(\theta), \quad 0 \le \theta \le \pi, \tag{63}$$

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 $\Gamma_2 := \{(r,\theta)|r = \rho(\theta),\ \pi < \theta < 2\pi\}$ of the boundary under Eqs. (61)-(63) with the overspecified data being given on $\Gamma_1 := \{(r,\theta)|r = \rho(\theta),\ 0 \leq \theta \leq \pi\}$.

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 [Kupradze and Aleksidze (1964)]. 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 $U(\mathbf{x},\mathbf{s}_j)$:

$$u(\mathbf{x}) = \sum_{j=1}^{n} c_j U(\mathbf{x},\mathbf{s}_j),\ \ \mathbf{s}_j \in \Omega^c. \tag{64}$$

For the Laplace equation (61) we have the fundamental solutions:

$$U(\mathbf{x},\mathbf{s}_j) = \ln r_j,\ \ r_j = \|\mathbf{x}-\mathbf{s}_j\|. \tag{65}$$

Previously, Liu (2008) has proposed a new preconditioner to reduce the ill-condition of the MFS. In the practical application of MFS, by imposing the boundary conditions (62) and (63) at N points on Eq. (64) we can obtain a linear equations system:

$$\mathbf{Bc} = \mathbf{b}, \tag{66}$$

where

$$\begin{aligned}
&\mathbf{x}_i = (x_i^1, x_i^2) = (\rho(\theta_i)\cos\theta_i, \rho(\theta_i)\sin\theta_i),\\
&\mathbf{s}_j = (s_j^1, s_j^2) = (R(\theta_j)\cos\theta_j, R(\theta_j)\sin\theta_j),\\
&B_{ij} = \ln\|\mathbf{x}_i-\mathbf{s}_j\|,\ \ \text{if } i \text{ is odd},\\
&B_{ij} = \frac{\eta(\theta_i)}{\|\mathbf{x}_i-\mathbf{s}_j\|^2}\left(\rho(\theta_i) - s_j^1\cos\theta_i - s_j^2\sin\theta_i - \frac{\rho'(\theta_i)}{\rho(\theta_i)}[s_j^1\sin\theta_i - s_j^2\cos\theta_i]\right), \text{if } i \text{ is even},\\
&\mathbf{c} = (c_1,\ldots,c_n)^{\mathrm{T}},\ \ \mathbf{b} = (h(\theta_1), g(\theta_1), \ldots, h(\theta_N), g(\theta_N))^{\mathrm{T}},
\end{aligned} \tag{67}$$

in which $n = 2N$, and

$$\eta(\theta) = \frac{\rho(\theta)}{\sqrt{\rho^2(\theta) + [\rho'(\theta)]^2}}. \tag{68}$$

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 (66) is available, we can apply the OMVIA to solve it.

Example 4: For the purpose of comparison we consider the following exact solution:

$$u(x,y) = \cos x \cosh y + \sin x \sinh y, \tag{69}$$

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

$$\rho(\theta) = \exp(\sin\theta)\sin^2(2\theta) + \exp(\cos\theta)\cos^2(2\theta). \tag{70}$$

After imposing the boundary conditions (62) and (63) at N points on Eq. (64) we can obtain a linear equations system. Here we fix $n = 40$ and take $D = 2$ to distribute the source points. The noise being imposed on the measured data h and g is $\sigma = 0.01$.

We solve this problem by the OMVIA with $m = 3$ and $\gamma = 0.01$. Through totally 1174 iterations the residual norm is smaller than 0.1 as shown in Fig. 4(a), where the value of a_0 is also shown. We compare the recovered data computed by the OMVIA with the exact one in Fig. 4(b). The numerical error as shown in Fig. 4(b) is smaller than 0.11. It can be seen that the OMVIA can accurately recover the unknown boundary condition. The result is better than that calculated by Liu (2013b) by using the OTVIA, which leads to the maximum error being 0.22.

4.4 Example 5: external force recovery problem

Let us consider the following inverse problem to recover the external force $F(t)$ for an ODE:

$$\ddot{y}(t) + \dot{y}(t) + y(t) = F(t). \tag{71}$$

In a time interval of $t \in [0, t_f]$ the discretized data $y_i = y(t_i)$ are supposed to be measurable, which are subjected to a random noise with an intensity σ. Usually, it is very difficult to recover the external force $F(t_i)$ from Eq. (71) by the direct differentials of the noisy data of the displacements, because the differential is a highly ill-posed linear operator.

To approach this inverse problem by the polynomial interpolation, we begin with

$$p_N(x) = c_0 + \sum_{k=1}^{N} c_k x^k. \tag{72}$$

Now, the coefficient c_k is split into two coefficients a_k and b_k to absorb more interpolation points; in the meanwhile, $\cos(k\theta_k)$ and $\sin(k\theta_k)$ are introduced to reduce the condition number of the coefficient matrix [Liu (2011c)]. We suppose that

$$c_k = \frac{a_k \cos(k\theta_k)}{R_{2k}^k} + \frac{b_k \sin(k\theta_k)}{R_{2k+1}^k}, \tag{73}$$

and

$$\theta_k = \frac{2k\pi}{N}, \quad k = 1, \ldots, N. \tag{74}$$

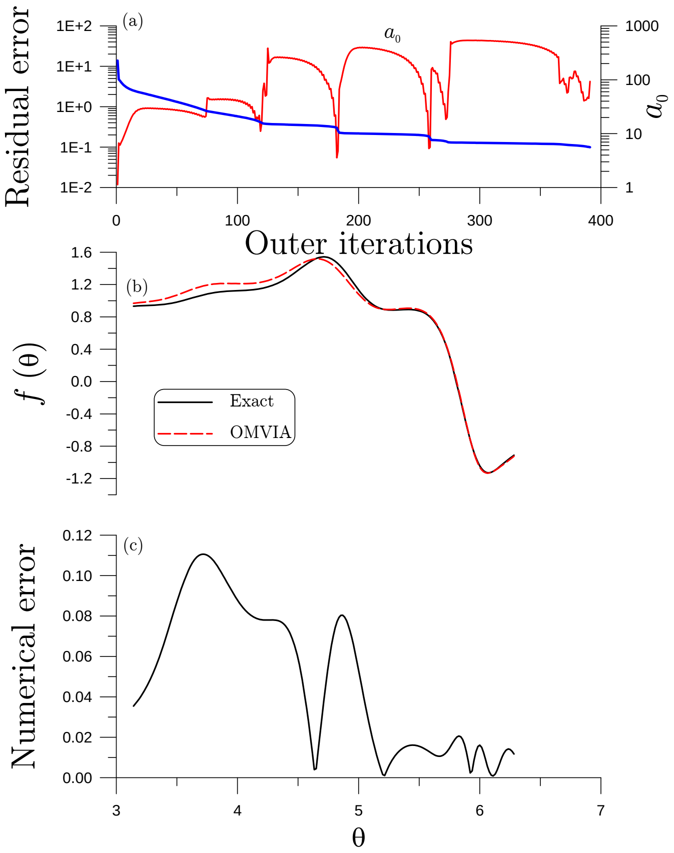

Figure 4: For example 4: (a) showing residual error and a_0, (b) comparing the numerical and exact solutions, and (c) showing the numerical error of recovered boundary condition.

The considered problem domain is $[a,b]$, and the interpolating points are:

$$a = x_0 < x_1 < x_2 < \ldots < x_{2N-1} < x_{2N} = b. \tag{75}$$

Substituting Eq. (73) into Eq. (72), we can obtain

$$p_N(x) = a_0 + \sum_{k=1}^{N} \left[a_k \left(\frac{x}{R_{2k}} \right)^k \cos(k\theta_k) + b_k \left(\frac{x}{R_{2k+1}} \right)^k \sin(k\theta_k) \right], \tag{76}$$

where we let $c_0 = a_0$. Here, a_k and b_k are unknown coefficients. In order to obtain them, we impose the following n interpolated conditions:

$$p(x_i) = y_i, \;\; i = 0, \ldots, n-1, \tag{77}$$

where $n = 2N+1$. Thus, we obtain a linear equations system to determine a_k and b_k:

$$\begin{bmatrix} 1 & \frac{x_0\cos\theta_1}{R_2} & \frac{x_0\sin\theta_1}{R_3} & \cdots & \left(\frac{x_0}{R_{2N}}\right)^N \cos N\theta_N & \left(\frac{x_0}{R_{2N+1}}\right)^N \sin N\theta_N \\ 1 & \frac{x_1\cos\theta_1}{R_2} & \frac{x_1\sin\theta_1}{R_3} & \cdots & \left(\frac{x_1}{R_{2N}}\right)^N \cos N\theta_N & \left(\frac{x_1}{R_{2N+1}}\right)^N \sin N\theta_N \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{x_{2N-1}\cos\theta_1}{R_2} & \frac{x_{2N-1}\sin\theta_1}{R_3} & \cdots & \left(\frac{x_{2N-1}}{R_{2N}}\right)^N \cos N\theta_N & \left(\frac{x_{2N-1}}{R_{2N+1}}\right)^N \sin N\theta_N \\ 1 & \frac{x_{2N}\cos\theta_1}{R_2} & \frac{x_{2N}\sin\theta_1}{R_3} & \cdots & \left(\frac{x_{2N}}{R_{2N}}\right)^N \cos N\theta_N & \left(\frac{x_{2N}}{R_{2N+1}}\right)^N \sin N\theta_N \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_N \\ b_N \end{bmatrix}$$

$$= \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{2N-1} \\ y_{2N} \end{bmatrix}. \quad (78)$$

We note that the norm of the first column of the above coefficient matrix is $\sqrt{2N+1}$. According to the concept of equilibrated matrix [Liu (2012f)], we can derive the optimal scales for the current interpolation with a half-order technique as

$$R_{2k} = \beta_0 \left(\frac{1}{2N+1} \sum_{j=0}^{2N} x_j^{2k} (\cos k\theta_k)^2 \right)^{1/(2k)}, \; k = 1,2,\ldots,N, \quad (79)$$

$$R_{2k+1} = \beta_0 \left(\frac{1}{2N+1} \sum_{j=0}^{2N} x_j^{2k} (\sin k\theta_k)^2 \right)^{1/(2k)}, \; k = 1,2,\ldots,N, \quad (80)$$

where β_0 is a scaling factor. The improved method uses N order polynomial to interpolate $n = 2N+1$ data nodes, while regular method with a full-order can only interpolate $N+1$ data points.

Now we fix $N = 10$ and $t_f = 5$ and consider the exact solution to be $F(t) = \cos t$, which is obtained by inserting the exact $y(t) = \sin t$ into Eq. (71). The parameter used is $\beta_0 = 1.24$, and the noise being imposed on the measured data is $\sigma = 0.01$.

The maximum number of outer iterations is set to be 600, or the convergence criterion is taken to be $\varepsilon = 0.001$. We solve this problem by the OMVIA with $m = 5$ and $\gamma = 0.1$. Through totally 2930 iterations we obtain an approximate solution, of which the residual error is shown in Fig. 5(a), where the value of a_0 is also shown. We compare the recovered data computed by the OMVIA with the exact one in Fig. 5(b). The numerical error as shown in Fig. 5(b) is smaller than 0.221. The maximum error obtained by the relaxed steepest descent method (RSDM) [Liu (2011a)] is a larger value 0.452.

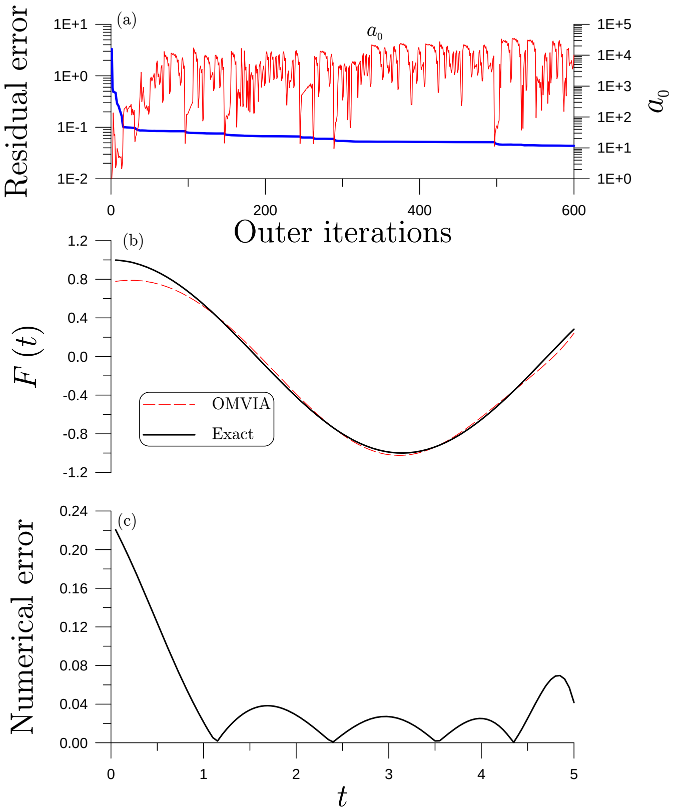

Figure 5: For example 5: (a) showing residual error and a_0, (b) comparing the numerical and exact solutions, and (c) showing the numerical error of recovered external force.

5 Conclusions

In the present paper, we have derived a purely iterative algorithm including a relaxation parameter γ chosen by the user, and with an m-vector optimal search direction in an m-dimensional Krylov subspace to solve a highly ill-posed linear system. This algorithm is an *Optimal Multi-Vector Iterative Algorithm* (OMVIA), which *has a* *good computational efficiency and accuracy* in solving the ill-posed linear equations system. In particular, the OMVIA has a better filtering effect against noise, such that as shown in all the numerical examples the numerical results recovered are quite smooth. Numerical tests on the linear inverse problems have confirmed the robustness of the OMVIA against noisy disturbance even with an intensity being large up to 10%.

Acknowledgement: Highly appreciated are the project NSC-100-2221-E-002-165-MY3 and the 2011 Outstanding Research Award from the National Science Council of Taiwan, and the 2011 Taiwan Research Front Award from Thomson Reuters.

References

Akaike, H. (1959): On a successive transformation of probability distribution and its application to the analysis of the optimum gradient method. *Ann. Inst. Stat. Math. Tokyo*, vol. 11, pp. 1-16.

Barzilai, J.; Borwein, J. M. (1988): Two point step size gradient methods. *IMA J. Numer. Anal.*, vol. 8, pp. 141-148.

Bhaya, A.; Kaszkurewicz, E. (2006): Control Perspectives on Numerical Algorithms and Matrix Problems. SIAM, Advances in Design and Control 10.

Chehab, J. P.; Laminie, J. (2005): Differential equations and solution of linear systems. *Numer. Algor.*, vol. 40, pp. 103-124.

Dongarra, J.; Sullivan, F. (2000): Guest editors' introduction to the top 10 algorithms. *Comput. Sci. Eng.*, vol. 2, pp. 22-23.

Farcas, A.; Lesnic, D. (2006): The boundary-element method for the determination of a heat source dependent on one variable. *J. Eng. Math.*, vol. 54, pp. 375-388.

Helmke, U.; Moore, J. B. (1994): Optimization and Dynamical Systems, Springer, Berlin.

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

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

Liu, C.-S. (2001): Cone of non-linear dynamical system and group preserving schemes. *Int. J. Non-Linear Mech.*, vol. 36, pp. 1047-1068.

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

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

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

Liu, C.-S. (2011c): A highly accurate multi-scale full/half-order polynomial interpolation. *CMC: Computers, Materials & Continua*, vol. 25, pp. 239-263.

Liu, C.-S. (2012a): Modifications of steepest descent method and conjugate gradient method against noise for ill-posed linear systems. *Commun. Numer. Anal.*, vol. 2012, Article ID cna-00115, 24 pages.

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

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

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

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

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

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

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

Liu, C.-S.; Atluri, S. N. (2008): A novel time integration method for solving a large system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 31, pp. 71-83.

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

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

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

Liu, C.-S.; Atluri, S. N. (2011b): An iterative algorithm for solving a system of nonlinear algebraic equations, $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, using the system of ODEs with an optimum α in $\dot{\mathbf{x}} = \lambda[\alpha\mathbf{F} + (1-\alpha)\mathbf{B}^{\mathrm{T}}\mathbf{F}]$; $B_{ij} = \partial F_i/\partial x_j$. *CMES: Computer Modeling in Engineering & Sciences*, vol. 73, pp. 395-431.

Liu, C.-S.; Atluri, S. N. (2012): A globally optimal iterative algorithm using the best descent vector $\dot{\mathbf{x}} = \lambda[\alpha_c\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}]$, with the critical value α_c, for solving a system of nonlinear algebraic equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. *CMES: Computer Modeling in Engineering & Sciences*, vol. 84, pp. 575-601.

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

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

Saad, Y.; Schultz, M. H. (1986): GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856-869.

Simoncini, V.; Szyld, D. B. (2007): Recent computational developments in krylov subspace methods for linear systems. *Numer. Linear Algebra Appl*, vol. 14, pp. 1-59.

Yang, L.; Deng, Z. C.; Yu, J. N.; Luo, G. W. (2009): Optimization method for the inverse problem of reconstructing the source term in a parabolic equation. *Math. Comput. Simula.*, vol. 80, pp. 314-326.