Journal of Marine Science and Technology

Volume 17 | Issue 3 Article 7

NOVEL METHODS FOR SOLVING SEVERELY ILL-POSED LINEAR EQUATIONS SYSTEM

Chein-Shan Liu
Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., liucs@ntu.edu.tw

Chih-Wen Chang
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

NOVEL METHODS FOR SOLVING SEVERELY ILL-POSED LINEAR EQUATIONS SYSTEM

Chein-Shan Liu* and Chih-Wen Chang**

Key words: ill-posed linear equations, nonstandard group-preserving scheme, unconditional stable scheme, regularization, L-curve, fictitious time integration method (FTIM).

ABSTRACT

We treat an ill-posed system of linear equations by transforming it into a linear system of stiff ordinary differential equations (SODEs), adding a differential term on the left-hand side. In order to overcome the difficulty of numerical instability when integrating the SODEs, Liu [20] has combined nonstandard finite difference method and group-preserving scheme, namely the nonstandard group-preserving scheme (NGPS), to obtain an unconditional stable numerical method for SODEs. This paper applies the NGPS to the SODEs resulting from the ill-posed linear equations, and proves that the new algorithms are unconditional stable. To strengthen accuracy, an L-curve is used to select a suitable regularization parameter. Moreover, we also combine the NGPS with a newly developed fictitious time integration method (FTIM) from Liu and Atluri [29] to solve the ill-posed linear equations. Several numerical examples are examined and compared with exact solutions, revealing that the new algorithms have better computational efficiency and accuracy even for the highly ill-conditioned linear equations with a large disturbance on the given data.

I. INTRODUCTION

In this paper we will propose some robust and easily-implemented new methods to solve the following linear equations system:

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

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a given positive definite matrix, and $\mathbf{x} \in \mathbb{R}^n$ is an unknown vector.

The input data of $\mathbf{b} \in \mathbb{R}^n$ may be corrupted by noise. In a practical use of (1) in engineering problems, the data $\mathbf{b}$ are rarely given exactly; instead of the noises are unavoidable due to measurement and modeling errors. Therefore, we may encounter the problem that the numerical solution of (1) may deviate from the exact one to a great extent, when $\mathbf{A}$ is severely ill-conditioned and $\mathbf{b}$ is perturbed by noise.

The solution of ill-posed linear equations is an important issue for many engineering problems. We are specially interesting on the solution of the above equation under noise, when the condition number of $\mathbf{A}$ is very large. A good numerical method to solve (1) may be beneficial in the applications to the optimization problems including linear programming and nonlinear programming, Newton's, Quasi-Newton's and homotopy methods for nonlinear equations system, finite difference and finite element methods for partial differential equations, etc.

Many numerical methods used in computational mechanics, as demonstrated by Atluri [1], Atluri *et al*. [2], Atluri and Shen [3], Atluri and Zhu [4, 5], and Zhu *et al*. [45], lead to the requirement by solving linear equations system. Collocation methods, as those used by Liu [21-23] for the modified Trefftz method of Laplace equation also need to solve a large system of linear equations.

To account of the sensitivity to noise it is usually using a regularization method to solve this sort of ill-posed problem [15, 37, 42, 44], where a suitable regularized parameter is used to depress the bias in the computed solution by a better balance of approximation error and propagated data error. There are several techniques developed after the pioneer work of Tikhonov and Arsenin [41]. For a large scale system the main choice is using the iterative regularization algorithm, where a regularized parameter is represented by the number of iterations. The iterative method works if an early stopping criterion is used to prevent from reconstruction of noisy components in the approximated solutions.

A measure of the ill-posedness of (1) can be performed by calculating the condition number of $\mathbf{A}$ [40]:

$$\mathrm{cond}(\mathbf{A}) = \left\|\mathbf{A}\right\| \left\|\mathbf{A}^{-1}\right\|, \tag{2}$$

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

Author for correspondence: Chein-Shan Liu (e-mail: liucs@ntu.edu.tw).
**Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.*
***Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

$$\mathrm{cond}(\mathbf{A}) = \frac{\max_{\sigma(\mathbf{A})} |\lambda|}{\min_{\sigma(\mathbf{A})} |\lambda|}, \tag{3}$$

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

Speaking roughly, the numerical solution of (1) may lose the accuracy of k decimal points when $\mathrm{cond}(\mathbf{A}) = 10^k$. The problems of this sort with ill-conditioned $\mathbf{A}$ may appear in several fields. For example, finding an n-degree polynomial function $r(x) = a_0 + a_1 x + \ldots + a_n x^n$ to best match a continuous function $f(x)$ in the interval of $x \in [0, 1]$:

$$\min_{\deg(r)\le n} \int_0^1 |f(x) - r(x)|\, dx, \tag{4}$$

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

$$A_{ij} = \frac{1}{i - 1 + j}, \tag{5}$$

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

$$\mathbf{b} = \begin{bmatrix} \int_0^1 f(x)dx \\ \int_0^1 xf(x)dx \\ \vdots \\ \int_0^1 x^n f(x)dx \end{bmatrix} \tag{6}$$

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

The Hilbert matrix is a famous example of highly ill-conditioned matrices, which can be seen from Table 1. Equation (1) with the matrix $\mathbf{A}$ having a large condition number usually displays that an arbitrary small perturbation on the right-hand side may lead to an arbitrary large perturbation of the solution on the left-hand side.

On the other hand, when we apply the central difference scheme on the following two-point boundary value problem:

$$\begin{aligned} &-u''(x) = f(x), \quad 0 < x < 1, \\ &u(0) = a, \ u(1) = b, \end{aligned} \tag{7}$$

we get

$$\mathbf{Au} = \begin{bmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} (\Delta x)^2 f(\Delta x) + a \\ (\Delta x)^2 f(2\Delta x) \\ \vdots \\ (\Delta x)^2 f((n-1)\Delta x) \\ (\Delta x)^2 f(n\Delta x) + b \end{bmatrix}, \tag{8}$$

where $\Delta x = 1/(n + 1)$ is the spatial length, and $u_i = u(i\Delta x)$, $i = 1, \ldots, n$, 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 $\mathbf{A}$ is known as a central difference matrix.

Table 1. The condition numbers of Hilbert matrix.

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

Table 2. The condition numbers of central difference matrix.

n	$\mathrm{cond}(\mathbf{A})$	n	$\mathrm{cond}(\mathbf{A})$
60	1.5074×10^3	80	2.6584×10^3
100	4.1336×10^3	120	5.9331×10^3
140	8.0568×10^3	160	1.0505×10^4
180	1.3277×10^4	200	1.6373×10^4

Taking the inverse of $\mathbf{A}$ in (1) or (8) we may obtain the unknown vector $\mathbf{x}$, or the unknown vector $\mathbf{u}$. However, there exhibits a great difficulty when $\mathbf{A}$ has a large condition number. The eigenvalues of $\mathbf{A}$ are found to be [19]

$$4\sin^2 \frac{k\pi}{2(n+1)}, \ k = 1, 2, \ldots, n, \tag{9}$$

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

$$\mathrm{cond}(\mathbf{A}) = \frac{\sin^2 \dfrac{n\pi}{2(n+1)}}{\sin^2 \dfrac{\pi}{2(n+1)}} \tag{10}$$

may be getting a large number when the grid number n is very large. See Table 2 for a list of some condition numbers.

II. THE REQUIREMENT OF METHODOLOGY

There are several regularization methods to deal with (1) when $\mathbf{A}$ is ill-conditioned. In this paper we consider an iterative regularization method for (1) by investigating the long term behavior of the following equation:

$$\dot{\mathbf{x}} = \mathbf{b} - \mathbf{Ax} =: \mathbf{r}(\mathbf{x}), \tag{11}$$

where the superimposed dot denotes the differential with respect to t, which is an independent variable. The fixed point, i.e., $\mathbf{r}(\mathbf{x}) = \mathbf{0}$, of the above equation is the solution of (1). When t approaches to a large value we may expect that $\mathbf{x}$ tends to the solution of (1).

The regularization in (11) is performed by integrating the initial value problem only up to a value $t = 1/\gamma$, where γ is a regularization parameter [8, 10].

In fact, the integration of (11) by a forward Euler method leads to the Richardson's iteration method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h(\mathbf{b} - \mathbf{A}\mathbf{x}_k), \tag{12}$$

where the factor h is a time stepsize. For a stability reason h is constrained by

$$0 < h < \frac{2}{\|\mathbf{A}\|} \le \frac{2}{\lambda_{\max}}, \tag{13}$$

in which $\lambda_{\max}$ is the maximum eigenvalue of $\mathbf{A}$.

If $\mathbf{A}$ is not a positive definite matrix, we may also consider the normal equation by multiplying (1) by $\mathbf{A}^{\mathrm{T}}$:

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}, \tag{14}$$

where the superscript T denotes the transpose.

Landweber [16] has proposed an iteration method to find the solution of the above equation:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + h(\mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x}_k), \tag{15}$$

where, in order to ensure the convergence of numerical solution, h is a fixed time stepsize satisfying

$$0 < h < \frac{2}{\|\mathbf{A}^{\mathrm{T}}\mathbf{A}\|}. \tag{16}$$

It is known that the reciprocal of iteration number plays a role of regularization parameter [9, 17]. How to speed up the convergence rate of the Landweber iteration is also discussed by Hanke [11]. Apart from its easy implementation, the Landweber iteration method presents a better regularization and robustness feature. Recently, Rieder [38] has proposed a Runge-Kutta regularization method for the ill-posed linear problems.

Since $\mathbf{A}$ is ill-conditioned, Eq. (11) is a stiff ODEs system. When applying a numerical integration technique to solve (11), it is usually required that the numerical method should be unconditional stable and that it can preserve the fixed point behavior.

In order to understand this problem let us consider a simple ODEs system:

$$\dot{\mathbf{x}} = \begin{bmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{bmatrix} \mathbf{x} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}. \tag{17}$$

Under the initial conditions $x_1(0) = 2$ and $x_2(0) = 1$, the solutions are

$$\begin{aligned} x_1(t) &= 1.5e^{-t} + 0.5e^{-1000t} + \frac{b_1 + b_2}{2}[1 - e^{-t}] + \frac{b_1 - b_2}{2000}[1 - e^{-1000t}], \\ x_2(t) &= 1.5e^{-t} + 0.5e^{-1000t} + \frac{b_1 + b_2}{2}[1 - e^{-t}] - \frac{b_1 - b_2}{2000}[1 - e^{-1000t}]. \end{aligned} \tag{18}$$

The eigenvalues of the system matrix in (17) are $\lambda_1 = -1000$ and $\lambda_2 = -1$. A number of integration schemes when applied to (17) require that both $|h\lambda_1|$ and $|h\lambda_2|$ be bounded by a certain number. For example, it is necessary that $1000h < 2$ for a stable calculation of (17) by the Euler method. This condition imposes a severe restriction on the time stepsize h used in the numerical integration.

The steady-state solutions are obtained from (18) by letting $t \to \infty$,

$$\begin{aligned} x_1 &= \frac{1}{1000}[500.5b_1 + 499.5b_2], \\ x_2 &= \frac{1}{1000}[499.5b_1 + 500.5b_2]. \end{aligned} \tag{19}$$

Assume that $b_1 = 1$ and $b_2 = 1$ and thus the exact solutions are $x_1 = x_2 = 1$. In order to get an accurate solution with the error smaller than 5×10^{-6} from exact solution, it requires at least a time $t_0 = 5\ln 10 = 11.513$ for t in (18), and hence 5756 steps are required for the Euler method. For every one order increasing of the condition number of $\mathbf{A}$, the step numbers also increase one order under the same required accuracy. For example, if the condition number increases up to 10^{10} then the Euler method requires 5756×10^7 steps in order to achieve a solution with an accuracy of 5×10^{-6}. From this demonstration it is clear that the Euler integrator is not appropriate to treat the ill-posed problem with a high condition number.

As mentioned above we usually require our integration of (11) to a large time extent in order to get a steady-state solution. If the time stepsize of a numerical scheme is restricted to be very small due to a reason of stability, it is hardly been used in the integration of stiff equation (11). For a highly ill-posed problem $\mathbf{A}$ has a large condition number, which also renders (11) very stiff, and the approach to a steady state usually requires a time very long. It is thus very difficult to apply the conventional numerical scheme to search the steady-state solution, since it is very expensive of computational time.

III. APPLIED THE NGPS TO ILL-POSED LINEAR EQUATIONS

An effective scheme is developed here by considering the nonstandard finite difference method for solving severely stiff problems, which is basing on the group preserving scheme proposed by Liu [18, 20] stated as follows for self-content.

1. Group Preserving Scheme

Liu [18] has embedded the n-dimensional system (11) into the following n+1-dimensional augmented system:

$$\frac{d}{dt}\begin{bmatrix} \mathbf{x} \\ \|\mathbf{x}\| \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n\times n} & \dfrac{\mathbf{r}(\mathbf{x})}{\|\mathbf{x}\|} \\ \dfrac{\mathbf{r}^{\mathrm{T}}(\mathbf{x})}{\|\mathbf{x}\|} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \|\mathbf{x}\| \end{bmatrix}. \tag{20}$$

Since the system matrix is an element of the Lie algebra $so_o(n, 1)$, the Lie group generated from it is known as a Lorentz group $SOo(n, 1)$.

The group-preserving scheme (GPS) can preserve the above internal symmetry group $SOo(n, 1)$ of the augmented system. We refer Liu [18] for the following integration method:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\|\mathbf{x}_k\|^2 + \tau \mathbf{r}_k \cdot \mathbf{x}_k}{\|\mathbf{x}_k\|^2 - \tau^2 \|\mathbf{r}_k\|^2} h \mathbf{r}_k = \mathbf{x}_k + \eta_k \mathbf{r}_k, \tag{21}$$

where $\mathbf{x}_k$ denotes the numerical value of $\mathbf{x}$ at the discrete time t_k, $\tau := h/2$, $\mathbf{r}_k$ denotes $\mathbf{r}(\mathbf{x}_k)$, and η_k is an adaptive factor.

Some properties of preserving the fixed point behavior of the above numerical scheme (21) have been investigated by Liu [18], showing that it is efficient and accurate for the numerical solutions of ODEs. However, for the use of GPS in the stiff ODEs the stepsize may require to be very small; hence, we need to introduce the following nonstandard GPS developed by Liu [20].

2. Nonstandard Group Preserving Scheme

The main idea of nonstandard finite difference [32-35] is replacing the Euler forward approximation of $\dot{\mathbf{x}}_k$:

$$\dot{\mathbf{x}}_k \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}, \tag{22}$$

by a nonstandard forward approximation:

$$\dot{\mathbf{x}}_k \approx \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\phi(h)}, \tag{23}$$

where $\phi(h)$ is a denominator function with the properties of $\phi(h) > 0$ and $\phi(h) = h + O(h^2)$.

For linear stiff ODE we may let

$$\phi(h) := \frac{1 - \exp(-\rho h)}{\rho}, \tag{24}$$

where ρ is a number not smaller than the Lipschitz constant of (11):

$$\rho \geq L = \|\mathbf{A}\| \geq \max\{|\lambda_i| : i = 1, 2, ..., n\}. \tag{25}$$

The replacement of h by $\phi(h)$ in (23) inspired Liu [20] to replace the h in (21) by $\phi(h)$; consequently, one has

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{4\|\mathbf{x}_k\|^2 + 2\phi \mathbf{r}_k \cdot \mathbf{x}_k}{4\|\mathbf{x}_k\|^2 - \phi^2 \|\mathbf{r}_k\|^2} \phi \mathbf{r}_k. \tag{26}$$

From (24) and $\|\mathbf{r}_k\| \leq L \|\mathbf{x}_k\|$ it follows that

$$\frac{\phi \|\mathbf{r}_k\|}{\|\mathbf{x}_k\|} \leq \phi L < 1, \ \forall h > 0. \tag{27}$$

Hence, the denominator in (26) is positive, i.e.,

$$4\|\mathbf{x}_k\|^2 - \phi^2 \|\mathbf{r}_k\|^2 = \|\mathbf{x}_k\|^2 \left(4 - \frac{\phi^2 \|\mathbf{r}_k\|^2}{\|\mathbf{x}_k\|^2} \right) > 0. \tag{28}$$

It guarantees that the adaptive factor in (26) is always positive, that is,

$$\eta_k := \frac{4\|\mathbf{x}_k\|^2 + 2\phi \mathbf{r}_k \cdot \mathbf{x}_k}{4\|\mathbf{x}_k\|^2 - \phi^2 \|\mathbf{r}_k\|^2} \phi > 0, \ \forall h > 0. \tag{29}$$

The combination of nonstandard difference method with group preserving scheme, namely the nonstandard group preserving scheme (NGPS), renders the new numerical scheme (26) always stable. This result is very important for stiff differential equations, because as demonstrated by Shampine and Gear [39] the dominant factor to choose a suitable stepsize for stiff differential equations is its stability, not its accuracy.

Furthermore, scheme (26) preserves the fixed point and the property of original differential equations system. Under the above condition (29), it is obvious that

$$\mathbf{x}_{k+1} = \mathbf{x}_k \Leftrightarrow \mathbf{r}_k = \mathbf{0}. \tag{30}$$

This means that $\mathbf{x}_k$ is a fixed point of the discretized mapping (26) if and only if the point $\mathbf{x}$ is a fixed point of the system (11).

Liu [20] has proved that the mapping (26) preserves the property of stable fixed point for all $h > 0$. For the linear stiff equation (11) the fixed point is an asymptotically stable one, since $-\mathbf{A}$ has negative eigenvalues. Therefore, the application of NGPS to this equation may be beneficial from those good properties.

IV. NUMERICAL METHODS FOR LINEAR EQUATIONS

1. Numerical Algorithm of NGPS (Algorithm 1)

Substituting (11) for $\mathbf{r}$ into (26) we can obtain

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{r}_k, \tag{31}$$

where

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k \tag{32}$$

is a residual vector at the k-th step, and the adapting factor η_k is

$$\eta_k := \frac{4\|\mathbf{x}_k\|^2 + 2\phi \mathbf{r}_k \cdot \mathbf{x}_k}{4\|\mathbf{x}_k\|^2 - \phi^2 \|\mathbf{r}_k\|^2}\phi. \tag{33}$$

When applying the NGPS to solve (1), the numerical procedures can be summarized as follows (**Algorithm 1**):

(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
(ii) For $k = 0, 1, 2\ldots$ we repeat the following calculations. If $\|\mathbf{r}_k\| < \varepsilon$ for a given stopping criterion ε, then stop; otherwise, let $k = k + 1$ and find the next $\mathbf{x}_{k+1}$ by (31), and $\mathbf{r}_{k+1}$ by (32).

We can prove that the above algorithm is *unconditional stable*. By using the inequality $\mathbf{r}_k \cdot \mathbf{x}_k \le \|\mathbf{r}_k\| \, \|\mathbf{x}_k\|$, from (33) it follows that

$$\eta_k \le \frac{2\|\mathbf{r}_k\|}{2\|\mathbf{r}_k\| - \phi\|\mathbf{r}_k\|}\phi. \tag{34}$$

Because of $\rho \ge \|\mathbf{A}\|$, by taking (27) and (24) into account we can obtain

$$\eta_k < 2\phi \le \frac{2}{\rho} \le \frac{2}{\|\mathbf{A}\|}. \tag{35}$$

The above equation guarantees that the iteration given in (31) converges, no matter what $h > 0$ is used. For the Richardson iteration given in (12) or the Landweber iteration given in (15), h is constrained either by (13) or by (16).

It can be seen that the NGPS algorithm is controlled by two parameters h and ρ in the ranges of $h > 0$ and $\rho \ge \|\mathbf{A}\| \ge \lambda_{\max}$. Equation (35) reveals that a larger ρ will lead to a smaller η_k, which in turns renders to a slower convergence rate. The h controls the accuracy; however, for the linear equation (1) we do not take care the accuracy of its transient state governed by (11), but instead of we are concerned with its convergence rate to the steady state. For this reason h can be chosen as large as possible in order to quickly tend to the steady-state solution. For example, the smallest eigenvalue of the Hilbert matrix with $n = 9$ is about 1.52×10^{-9}, and the largest eigenvalue is about 1.75. If in (11) we use this Hilbert matrix, there are fast changing components and also slowly changing components. Tending to a steady state may require the time t larger than 10^9 for that $\exp[-10^{-9} t]$ can approach to zero. If the time stepsize h is small we may require many steps to approach the steady state. In Section V we will give numerical examples to show that a suitable selection of (h, ρ) may lead to a better NGPS algorithm to treat the ill-posed linear problems. When the noisy effect is also considered, we may need to select a more better (h, ρ, α) to calculate the solution, where α plays a role of regularization parameter as to be shown in Section IV.3.

2. The Steepest Descent and Conjugate Gradient Methods

Solving (1) by the steepest descent method [14] is equivalent to solve the following minimum problem:

$$\min_{\mathbf{x}\in R^n} \varphi(\mathbf{x}) = \min_{\mathbf{x}\in R^n} [\frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{x}^{\mathrm{T}}\mathbf{b}]. \tag{36}$$

Then, by using the Ritz variational principle we can derive the following algorithm:

(i) Give an initial $\mathbf{x}_0$, and then $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$.
(ii) For $k = 0, 1, 2\ldots$ we repeat the following calculations: If $\|\mathbf{r}_k\| < \varepsilon$ then stop; otherwise, let $k = k + 1$ and find the next $\mathbf{x}_{k+1}$ and $\mathbf{r}_k$ by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{r}_k, \tag{37}$$

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k, \tag{38}$$

$$\eta_k = \frac{\|\mathbf{r}_k\|^2}{\mathbf{r}_k^{\mathrm{T}}\mathbf{A}\mathbf{r}_k}. \tag{39}$$

Go to step (ii).

As compared with the algorithm in Section IV.1 one can find that the algorithm of steepest descent method (SDM) is similar to the NGPS, besides that the calculations of the adapting factor η_k by (39) for the SDM, and by (33) for the NGPS. However, the SDM is not unconditional stable, because η_k, given in (39), satisfying

$$\frac{1}{\lambda_{\min}} \ge \eta_k \ge \frac{1}{\lambda_{\max}}, \tag{40}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are respectively the smallest and the largest eigenvalues of $\mathbf{A}$, is not guaranteed to satisfy the stable condition (13).

For the SDM the residual vector $\mathbf{r}_k$ is the steepest descent direction of the function φ at the point $\mathbf{x}_k$. But when $\|\mathbf{r}_k\|$ is rather small the calculated $\mathbf{r}_k$ may deviate from the real steepest descent direction to a great extent due to 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 searching direction of the minimum by imposing the orthogonality of the residual vectors at each iterative step [14]. The algorithm of the CGM can be summarized as follows:

(i) Give an initial $\mathbf{x}_0$.
(ii) Calculate $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ and $\mathbf{p}_1 = \mathbf{r}_0$.
(iii) For $k = 1, 2\ldots$ we repeat the following calculations:

$$\eta_k = \frac{\|\mathbf{r}_{k-1}\|^2}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k}, \tag{41}$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \eta_k \mathbf{p}_k, \tag{42}$$

$$\mathbf{r}_k = \mathbf{r}_{k-1} - \eta_k \mathbf{A}\mathbf{p}_k, \tag{43}$$

$$a_k = \frac{\|\mathbf{r}_k\|^2}{\|\mathbf{r}_{k-1}\|^2}, \tag{44}$$

$$\mathbf{p}_{k+1} = \mathbf{p}_k + a_k \mathbf{p}_k. \tag{45}$$

If $\mathbf{x}_k$ converges according to a given stopping criterion:

$$\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \varepsilon, \tag{46}$$

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

3. A Regularization of NGPS (Algorithm 2)

A regularization can be employed when one solves (1) under a highly ill-conditioned **A**. Hansen [12] and Hansen and O'Leary [13] have given an illuminating explain that the Tikhonov regularization of linear problems is a trade-off between the size of the regularized solution and the quality to fit the given data:

$$\min_{\mathbf{x}\in R^n} \varphi(\mathbf{x}) = \min_{\mathbf{x}\in R^n} [\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \alpha \|\mathbf{x}\|^2]. \tag{47}$$

Therefore, instead of (11) we can apply the NGPS to solve

$$\dot{\mathbf{x}} = \mathbf{b} - \mathbf{A}\mathbf{x} - \alpha \mathbf{x}, \tag{48}$$

where α is a regularized parameter, which can be determined by a technique of L-curve. As that done in Section IV.1, we can obtain an iterative method given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{r}_k, \tag{49}$$

where

$$\mathbf{r}_k := \mathbf{b} - \mathbf{A}\mathbf{x}_k - \alpha \mathbf{x}_k, \tag{50}$$

$$\eta_k := \frac{4\|\mathbf{x}_k\|^2 + 2\phi \mathbf{r}_k \cdot \mathbf{x}_k}{4\|\mathbf{x}_k\|^2 - \phi^2 \|\mathbf{r}_k\|^2} \phi. \tag{51}$$

The numerical procedures can be written as follows (**Algorithm 2**):

(i) Give an initial $\mathbf{x}_0$.
(ii) For k = 0, 1, 2… we repeat the following calculations. If Eq. (46) is fulfilled, then stop; otherwise, let $k = k + 1$ and find the next $\mathbf{x}_{k+1}$ by (49).

4. A FTIM Technique

Liu and Atluri [29] have introduced a novel method by embedding the nonlinear algebraic equations into a system of nonautonomous first order ODEs:

$$\dot{\mathbf{x}} = -\frac{\nu}{1+t} \mathbf{r}(\mathbf{x}). \tag{52}$$

This numerical technique has been called a fictitious time integration method (FTIM).

The above idea by introducing a fictitious time t was first proposed by Liu [24] to treat an inverse Sturm-Liouville problem by transforming an ODE into a PDE. Then, Liu and his coworkers [25, 26, 31] extended this idea to develop new methods for estimating parameters in the inverse vibration problems. More recently, Liu [27] has used the FTIM technique to solve the nonlinear complementarity problems, whose numerical results are very well. Then, Liu [28] used the FTIM to solve the boundary value problems of elliptic type partial differential equations. Liu and Atluri [30] also employed this technique of FTIM to solve mixed-complementarity problems and optimization problems.

In this paper we will use the NGPS introduced in Section III.2 to integrate the above equation by inserting (11) for **r**, and the criterion of stopping iterations is given by

$$\|\mathbf{r}_k\| \le \varepsilon_1. \tag{53}$$

V. NUMERICAL EXAMPLES

In order to assess the performance of the newly developed methods let us investigate the following examples.

1. Example 1

We first consider a very simple example:

$$\begin{bmatrix} 1000 & 0 \\ -0.909 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{54}$$

with exact solution:

$$x_1 = \frac{b_1}{1000}, \quad x_2 = b_2 + \frac{0.909}{1000} b_1. \tag{55}$$

The condition number of this problem is 1000. Let us consider

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} - \begin{bmatrix} 1000 & 0 \\ -0.909 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \tag{56}$$

We apply both the NGPS method and the Landweber iteration method in (15) to this simple problem. Since the NGPS is designed for the solution of stiff equation, we can take a large time stepsize h = 100 when ρ is fixed to be the larger eigenvalue, i.e., ρ = 1000. When we apply the Landweber iteration method to this problem the stepsize h is restricted to be $h < 2 \times 10^{-6}$, and we take $h = 10^{-6}$.

In the case when the data (b_1, b_2) = (1, 1) are contaminated by a random noise, we are concerned with the stability of our calculation method, which is investigated by adding a random noise into b_1 and b_2 by $sR(i)$, where $R(i)$ are random numbers

between –1 and 1. Under the noise level in the range of $s \in [0.0001, 0.001]$, we have compared the numerical solutions obtained by the NGPS and the Landweber method by the relative errors of x_1 and x_2 as shown in Fig. 1(a). Obviously, it can be seen that the results of the NGPS are much better than that calculated by the Landweber method. For the NGPS both the erros of x_1 and x_2 are in the order of 10^{-3}, but the Landweber method gives an unacceptable solution of x_2.

Under the same initial condition $(x_1, x_2) = (0.1, 0.1)$ and the same stopping criterion $\varepsilon = 10^{-3}$, the iteration numbers of the Landweber method are much larger than that of the NGPS method as shown in Fig. 1(b), and the computational time of the Landweber method spent in this calculation is about twenty times of the NGPS. For the FTIM we use the NGPS to integrate (52) by using $h = 0.01$, $\rho = 10$, $v = -10$, and $\varepsilon_1 = 10^{-4}$. As shown in Fig. 1 by the dashed-dotted lines, the iteration numbers of the FTIM are smaller than that of the NGPS, and the accuracy of x_2 is also improved than that of the NGPS.

The main drawback of the Landweber iteration method is its large number of iterations needed to obtain a converged solution, and this situation is more worse when the stopping criterion is imposed more strictly for more ill-conditioned linear equations. To speed up the method, several semi-iterative methods have been investigated, for example, the v-method [6], and the polynomial acceleration method [11].

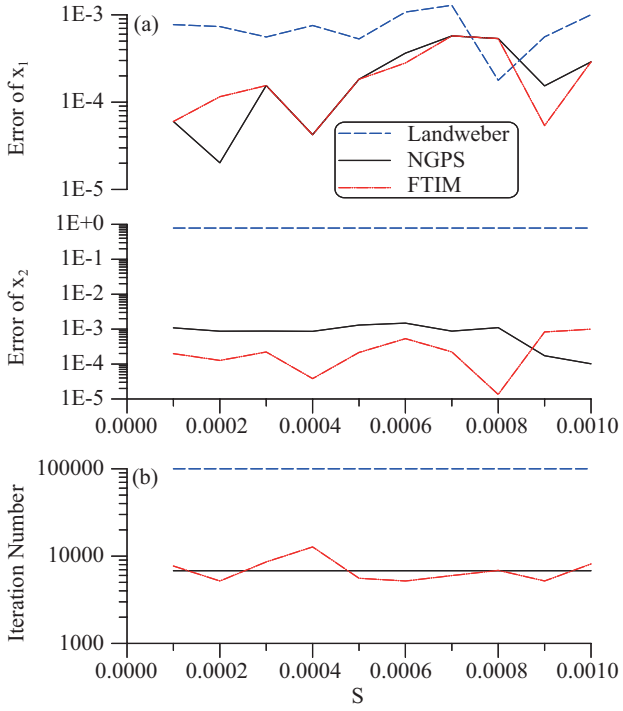

Fig. 1. For Example 1 the numerical erros and iteration numbers of NGPS, FTIM and the Landweber method under different noise levels are compared in (a) and (b).

2. Example 2

Let us consider the boundary value problem in (7) with $f(x) = \sin \pi x$. The exact solution is

$$u(x) = a + (b-a)x + \frac{1}{\pi^2}\sin \pi x. \tag{57}$$

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

In the calculation of this example by the NGPS we have fixed $\Delta x = 0.02$, $\rho = 10$ and $h = 10$. Starting from a set of initial conditions we can employ the iteration procedure as specified in Section IV.1 to calculate the solution, where the stopping criterion is taken to be $\varepsilon = 10^{-5}$. We first consider a linear initial condition as shown in Fig. 2(a). Through 13424 iterations the numerical solution converges to the exact solution very accurately as shown in Fig. 2(b) by the dashed line, while the exact solution is shown by the solid line. From Fig. 2(c) it can be seen that the numerical error is smaller than 5×10^{-4} even the quantity of $\Delta x = 0.02$ used in the discretization is of the second order.

Next we let $u_i = R(i)/2 + 1 + (\sin \pi x_i)/\pi^2$ be our initial conditions as shown in Fig. 2(a). The total number of iterations is 22044, which is much large than that for the linear initial condition. The numerical error as shown in Fig. 2(c) is coincident with the one under the linear initial condition. The computational times are about two to three seconds in the above two calculations. These results show that the NGPS algorithm works very well and accurately independent of the initial conditions.

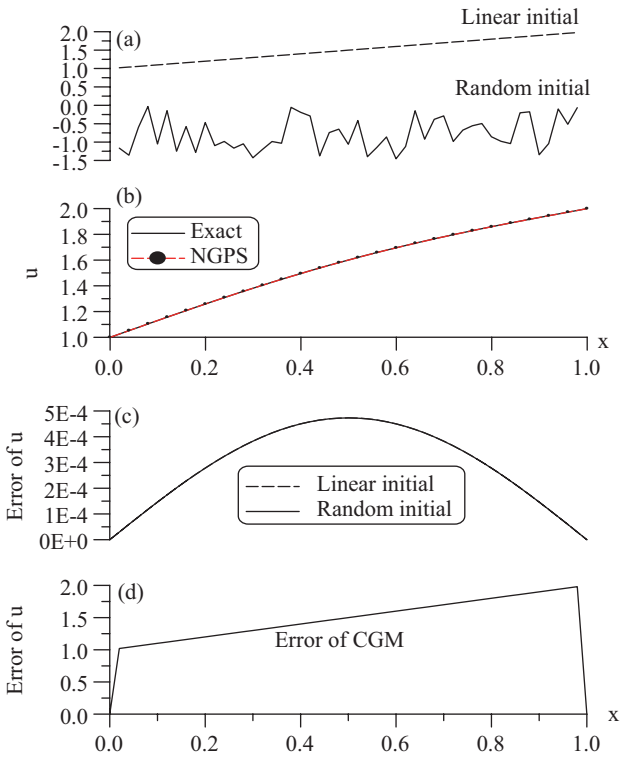

Fig. 2. Applying the NGPS for Example 2 we employ two different initial conditions in (a), and (b) comparing numerical and exact solutions, (c) the numerical errors of NGPS, and (d) the numerical error of CGM.

Table 3. For Example 2 comparing the maximum error (ME) and iteration number (IN) for different h and ρ.

ρ	h	ME	IN
1	0.1	5.031×10^{-3}	7977
	0.5	5.024×10^{-3}	1929
	10	3.492×10^{-5}	608199
	100	4.823×10^{-5}	64879
5	0.1	5.032×10^{-3}	9646
	1	5.029×10^{-3}	3821
	10	5.030×10^{-3}	3795
	100	5.030×10^{-3}	3795
10	0.1	5.033×10^{-3}	12008
	2	5.031×10^{-3}	7591
	10	5.030×10^{-3}	7591
	100	5.030×10^{-3}	7591
50	0.1	5.034×10^{-3}	38211
	5	5.034×10^{-3}	37954
	10	5.034×10^{-3}	37954
	100	5.034×10^{-3}	37954

In Table 3 we compare the maximum errors of our numerical results under different ρ and h, and the iteration numbers are also indicated when using the stopping criterion $\varepsilon = 10^{-4}$. It can be seen that for a fixed ρ, when h increases the iteration number decreases and saturates to a certain number. For a fixed h, when ρ increases the iteration number is also increased. In all cases the maximum errors can be controlled within a small range of $[5.024 \times 10^{-3}, 5.044 \times 10^{-3}]$. However, there are two particular cases for $\rho = 1$ and $h = 10$ and 100, the errors of which are greatly reduced to the order of 10^{-5}, but the iteration numbers are also increased to 608199 for $h = 10$ and 64879 for $h = 100$.

We have applied the SDM to this problem; however, it is very unstable no matter what initial conditions are used. Then, we apply the CGM to this problem and through 50 iterations it converges to a solution under the criterion (46) with $\varepsilon = 10^{-10}$. However, this solution is not a true solution of this problem, because the error as shown in Fig. 2(d) is very large.

Under a large noise level with $s = 0.001$, we have compared the numerical solutions obtained by the NGPS and the Landweber iteration method (12) with exact solution as shown in Fig. 3. Under the initial condition $u_i = 1.7$ for all i and the stopping criterion $\varepsilon = 6 \times 10^{-3}$, the Landweber iteration method through 42 iterations leads to a maximum error of 5.473×10^{-2} with a time stepsize $h = 0.9$, which is the maximum time stepsize that the Landweber method is stable for this problem. Under the same initial condition and the same stopping criterion $\varepsilon = 2 \times 10^{-4}$, **Algorithm 1** of the NGPS through 1894 iterations leads to a maximum error of 2.326×10^{-2} with $h = 1$ and $\rho = 2$. It can be seen that the NGPS method is slightly better than the Landweber method as shown in Fig. 3.

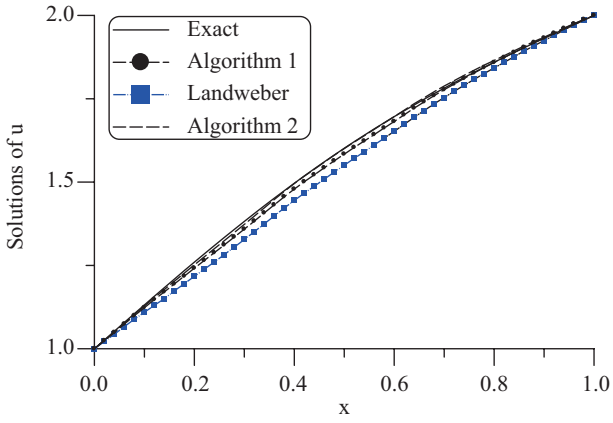

Fig. 3. For Example 2 under a large noise the numerical solutions of Algorithms 1 and 2 of the NGPS and the Landweber are compared with the exact solution.

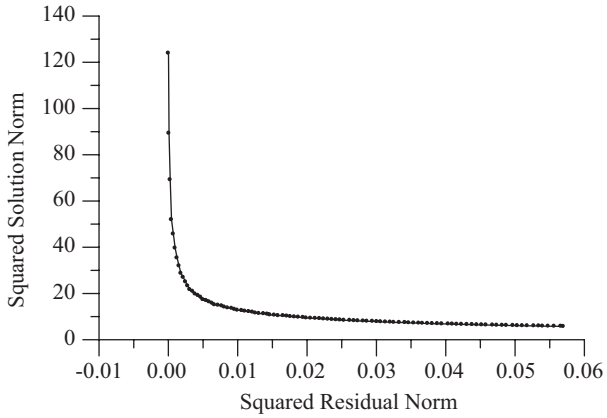

Fig. 4. The L-curve of Algorithm 2 of the NGPS obtained by varying the regularized parameter.

It can be seen that the non-regularized solution is already rather better. However, in order to increase the accuracy of the NGPS method, we can employ a regularization technique to this problem. According to **Algorithm 2** in Section IV.3 we first plot an L-curve in Fig. 4 under the stopping criterion in (46) with $\varepsilon = 2 \times 10^{-4}$. Then we select a regularized parameter to be $\alpha = 0.000064$, and apply **Algorithm 2** of the NGPS method to this problem under the same initial condition and the same stopping criterion $\varepsilon = 5 \times 10^{-4}$. Through 929 iterations it leads to a maximum error 9.641×10^{-3}. It can be seen that the results of **Algorithm 2** as shown in Fig. 3 with dashed line is slightly better than that calculated by **Algorithm 1** of the NGPS, and is much better than that calculated by the Landweber method.

3. Example 3

In this example we consider a highly ill-conditioned linear equation (1) with **A** given by (5). The ill-posedness of (1) increases fast with n.

Table 4. Comparing the numerical results for Example 3 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
NGPS	1.00001	0.99980	1.00090	0.99909	0.99928	1.00037	1.00105	1.00062	0.99887
FTIM	1.00001	0.99981	1.00082	0.99922	0.99933	1.00029	1.00092	1.00058	0.99901
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

1) n = 9

In order to compare the numerical solutions with exact solutions we suppose that $x_1 = x_2 = \ldots = x_n = 1$, and then by (5) we have

$$b_i = \sum_{j=1}^{n} \frac{1}{i+j-1}. \tag{58}$$

We first calculate this problem for the case with $n = 9$. As shown in Table 1, the resulting linear equation is highly ill-conditioned, since the condition number is very large up to 4.93×10^{11}.

In the calculation by the NGPS we have fixed $\rho = 2$ and $h = 0.5$. Starting from a set of initial conditions with $x_1 = \ldots = x_9 = 0.5$, we employ the iteration procedure in Section IV.1 to this problem with a stopping criterion $\varepsilon = 10^{-8}$. Through 182441 iterations the numerical solution converges to the exact solution very accurately as shown in Table 4, where the values obtained by the singular value decomposition (SVD) technique [36] are also listed for the purpose of comparison. In the calculation by the FTIM we have fixed $\rho = 10$, $h = 10^{-3}$, $\varepsilon_1 = 10^{-7}$ and $v = -20000$. Starting from a set of initial conditions with $x_1 = \ldots = x_9 = 0.5$, we employ the iteration procedure as specified in Section IV.4 to this problem with a stopping criterion $\varepsilon_1 = 10^{-7}$. Through 47403 iterations the numerical solution converges to the exact solution very accurately as shown in Table 4. The accuracy of FTIM is slightly better than that of the NGPS, and is also convergent fast than the NGPS.

In Fig. 5(a) we have used the solid line to denote the exact values and the dashed line to denote the numerical values obtained by the NGPS. From Fig. 5(b) it can be seen that the maximum numerical error of NGPS is equal to 1.12768×10^{-3}. But the maximum error of SVD is 1.6×10^{-2}. Obviously, the NGPS results in a great improvement of the numerical results than that of the SVD.

For Example 2 in Section V.2 both the SDM and the CGM are failed to find solution, which may be due to a large dimension of the problem and the narrow band character of the matrix **A**. But for Example 3 we can apply these methods to find the solutions, which are listed in Table 4. Both the NGPS and the SDM produce the same errors, and the CGM is slightly better than that of the NGPS and SDM.

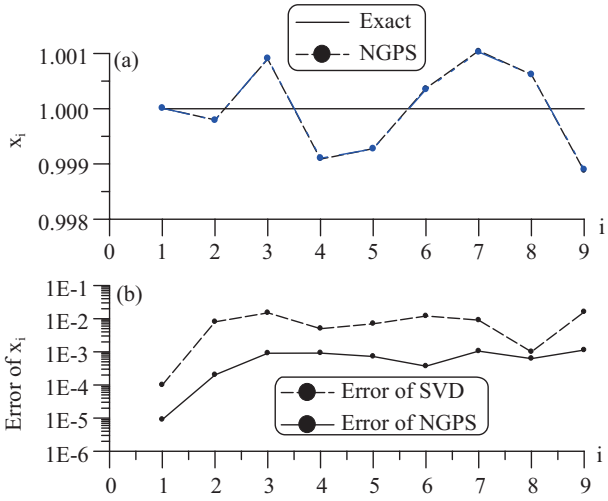

Fig. 5. For Example 3: (a) comparing the numerical result of NGPS with the exact solution, and (b) the numerical errors of NGPS and SVD.

In Fig. 6(a) we fix $\rho = 2$ and $\varepsilon = 10^{-6}$ and allow the stepsize varying from $h = 1$ to $h = 10$. It can be seen that the maximum errors are located within a narrow range of $[6.3388 \times 10^{-3}, 6.3692 \times 10^{-3}]$. It reflects that the NGPS method is stable for different time stepsize. Also, we have investigated the influence of ρ on the maximum errors in Fig. 6(b), from which it can be seen that the maximum errors are located within a narrow range of $[9.396 \times 10^{-3}, 9.401 \times 10^{-3}]$. In these calculations we have fixed $h = 1$ and $\varepsilon = 10^{-5}$. The NGPS method is also stable even one employs different ρ.

When applying the CGM to this problem we found that it is very sensitive to the noise; hence, we cannot calculate the result under a noise level $s = 10^{-5}$.

In the calculation of this noised problem by the NGPS we have fixed $\rho = 2$ and $h = 5$, and $\varepsilon = 10^{-5}$ is also used in the calculation by the SDM. The NGPS converges to a rather accurate solution as shown in Table 5 through 3139 iterations. In the calculation by the FTIM we have fixed $\rho = 1$, $h = 0.005$ and $v = -1000$. The FTIM converges very fast with 766 iterations to a rather accurate solution as shown in Table 5. On the other hand, the SDM converges to a slightly bad solution as shown in Table 5 through 31576 iterations. Unfortunately, for x_4 the SDM leads to an error about 21%.

Table 5. Comparing numerical results for Example 3 under noise.

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
NGPS	1.00016	1.00038	0.99902	0.99119	1.00490	1.01430	1.00135	0.99808	0.98977
FTIM	1.00021	1.00074	0.99568	0.99464	1.00761	1.00961	1.00572	0.99771	0.986885
SDM	1.00102	0.97669	1.12472	0.79421	1.00401	1.19965	0.93218	0.98919	0.97715

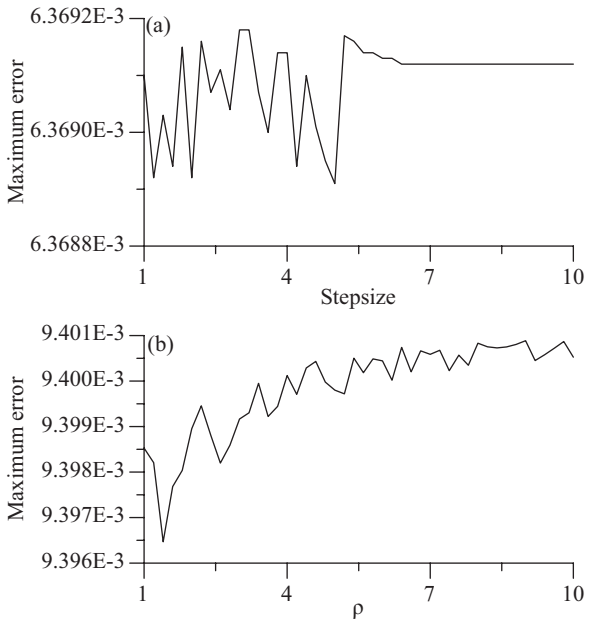

Fig. 6. When applying the NGPS for Example 3 we displaying the influence of (a) the time stepsize and (b) the factor ρ on the maximum error of solutions.

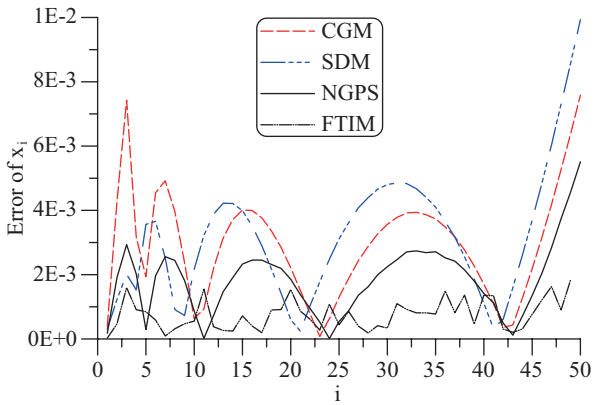

Fig. 7. For a highly ill-posed case of Example 3 with $n = 50$ we comparing the numerical errors of NGPS, FTIM, CGM and SDM.

2) $n = 50$

Let us increase the ill-posedness of this problem with $n = 50$ and with a noise $s = 10^{-8}$, which is the maximum noise that allows us to apply the CGM in the computation of this problem. For this problem the condition number is about 1.1748×10^{19}. In the calculation of this problem by the NGPS we have fixed $\rho = 2$ and $h = 5$, and $\varepsilon = 10^{-5}$ is also used in the calculations by the SDM and the CGM. The NGPS provides a rather accurate solution with the errors in the order of 10^{-3} as shown in Fig. 7 when comparing with the exact solutions $x_1 = \ldots = x_{50} = 1$. The accuracy of the SDM and the CGM are worse than that of the NGPS. The largest error for the SDM at x_{50} is about 0.01. In the calculation of this problem by the FTIM we have fixed $\rho = 10$, $h = 10^{-4}$, $\nu = -30000$ and $\varepsilon_1 = 10^{-8}$. The accuracy of the FTIM is much better than other methods as shown by the dashed-dotted line in Fig. 7.

Next, we consider

$$x_i = 2\sin(p_i)\exp[p_i(1-p_i)],\ p_i = i\times\frac{1}{n},$$

$$b_i = \sum_{j=1}^{n}\frac{1}{i+j-1}x_j + s[0.5+R(i)] \tag{59}$$

with $n = 50$ and $0 < p_i \le 1$. This noise has a mean value 0.5. This problem is more difficult than the one with constant $x_1 = \ldots = x_n = 1$.

When the noise is imposed in the levels of $s = 10^{-4}$ and $s = 10^{-2}$, the NGPS is still applicable. In Fig. 8(a) we compare the exact solution given in (59) with the numerical solution of the NGPS by using $\varepsilon = 2 \times 10^{-4}$, $h = 100$ and $\rho = 2$ for the case with $s = 10^{-4}$. The maximum error is about 0.068966. We also use the FTIM to calculate this problem under $\nu = -1000$, $\varepsilon_1 = 3 \times 10^{-4}$, $h = 0.001$ and $\rho = 15$. The result is plotted in Fig. 8(a) by the dashed-dotted line. It is convergent fast than that of the NGPS. In Fig. 8(b) we compare the exact solution with the numerical solution by using $\varepsilon = 5 \times 10^{-2}$, $h = 500$ and $\rho = 500$ for the case with $s = 10^{-2}$. At the two ends there are some discrepancies and the maximum error is about 0.172513. Calculating this problem by FTIM, we use $\nu = -1000$, $\varepsilon_1 = 2.5 \times 10^{-2}$, $h = 1$ and $\rho = 5$. The result is plotted in Fig. 8(b) by the dashed-dotted line. It is convergent fast than NGPS, and is slightly inaccurate than NGPS. These results are better than that calculated by using the Tikhonov regularization technique [43].

3) $n = 200$

It is known that the condition number of Hilbert matrix grows like as $e^{3.5n}$ when n is very large. For the case with $n =$

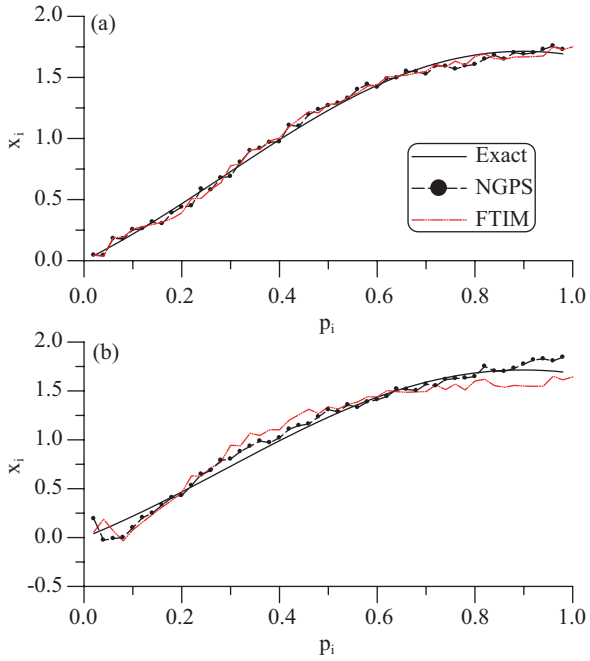

Fig. 8. Comparing the numerical solutions of NGPS and FTIM with a non-constant exact solution for Example 3 with n = 50: (a) s = 0.0001, and (b) s = 0.01.

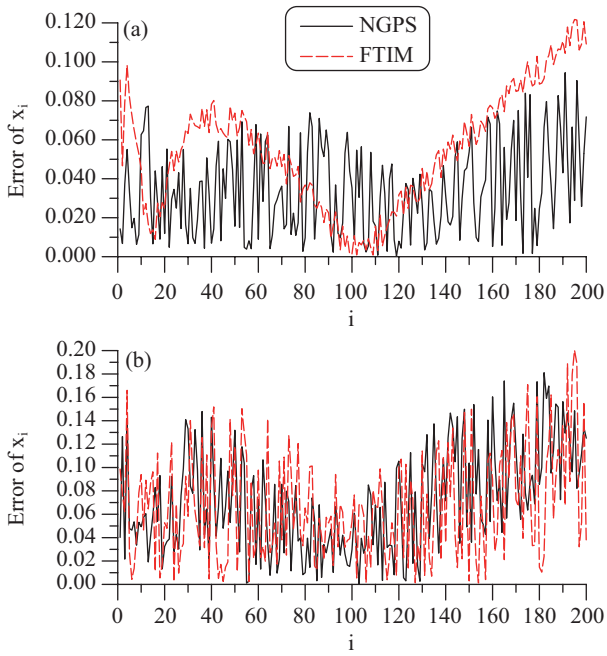

Fig. 9. The numerical errors of NGPS and FTIM for Example 3 with a fixed n = 200 and different (a) s = 0.001, and (b) s = 0.01.

200 the condition number is extremely huge up to 10^{348}. The exact inverse of the Hilbert matrix has been derived by Choi [7]:

$$(A)^{-1}_{ij} = (-1)^{(i+j)}(i+j-1)\binom{n+i-1}{n-j}\binom{n+j-1}{n-i}\binom{i+j-2}{i-1}^2. \quad (60)$$

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

Under this severe condition of both a tremendous ill-posedness of **A** with n = 200 and a sensible noise with $s = 10^{-3}$, the NGPS is still applicable to this problem by relaxing the convergence criterion to $\varepsilon = 10^{-2}$, increasing the stepsize to h = 1000 and with ρ = 2. Indeed, the NGPS gives a solution very fast only through 94 iterations, and the result is still acceptable with an error in the second order as shown in Fig. 9(a). When $s = 10^{-2}$ we also plot the numerical error in Fig. 9(b). For this case we use **Algorithm 2** in Section IV.3 to calculate the solution with α = 0.0001, while the convergence criterion ε = 0.08 and the stepsize h = 2000 were used. It spends 13 iterations to obtain the solution with a maximum error 0.1811. Similarly, we also plot the numerical errors by using the FTIM in Fig. 9 with dashed lines. Both cases spend 30 iterations, under the same v = –1000, ρ = 10 and h = 10, but with ε_1 = 0.05 for s = 0.001 and ε_1 = 0.1 for s = 0.01.

VI. CONCLUSION

In order to tackle of the numerical instability of some conventional iteration methods on solving ill-posed linear problems, we have developed two new algorithms based on a combination of nonstandard finite difference method with group-preserving scheme, namely the nonstandard group-preserving scheme (NGPS). We proved that the NGPS is unconditional stable, therefore, allowing a larger stepsize in the calculations without inducing numerical instability, as well as speeding up the convergence of iterations. We also investigated the effect of a newly developed fictitious time integration method (FTIM) on the solutions of ill-posed linear equations. When the time integration in the FTIM was carried out by the NGPS, the convergent behavior of FTIM is better than other methods. Several numerical examples were examined, some of which were compared with exact solutions revealing that the NGPS and FTIM can work very well even for highly ill-conditioned linear equations under a large noise perturbation. Through a regularization of NGPS we can obtain a more accurate algorithm by selecting a regularized parameter through the use of L-curve. Through this study, we have obtained two easily-implemented and unconditional stable iteration methods to solve the ill-posed linear problems.

REFERENCES

1. Atluri, S. N., *Methods of Computer Modeling in Engineering and Sciences*, Tech. Science Press (2002).

2. Atluri, S. N., Liu, H. T., and Han, Z. D., "Meshless local Petrov-Galerkin (MLPG) mixed collocation method for elasticity problems," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 14, pp. 141-152 (2006).
3. Atluri, S. N. and Shen, S., "The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite and boundary element methods," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 3, pp. 11-51 (2002).
4. Atluri, S. N. and Zhu, T. L., "A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics," *Computational Mechanics*, Vol. 22, pp. 117-127 (1998).
5. Atluri, S. N. and Zhu, T. L., "A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation," *Computational Modeling and Simulation in Engineering*, Vol. 3, pp. 187-196 (1998).
6. Brakhage, H., *On Ill-Posed Problems and the Method of Conjugate Gradients*. *Inverse and Ill-Posed Problems*, pp. 165-175, Engl, H. W. and Groetsch, C. W. eds., Academic Press, New York (1987).
7. Choi, M. D., "Tricks or treats with the Hilbert matrix," *American Mathematical Monthly*, Vol. 90, pp. 301-312 (1983).
8. Engl, H. W., Hanke, M., and Neubauer, A., *Regularization of Inverse Problems*, Kluwer Academic Publishers, Dordrecht (1996).
9. Engl, H. W., "Regularization methods for the stable solutions of inverse problems," *Survey Mathematics Industry*, Vol. 3, pp. 71-143 (1993).
10. Gilyazov, S. F. and Goldman, N. L., *Regularization of Ill-Posed Problems by Iteration Methods*, Kluwer Academic Publishers, Dordrecht (2000).
11. Hanke, M., "Accelerated Landweber iterations for the solutions of ill-posed problems," *Numerical Mathematics*, Vol. 60, pp. 341-373 (1991).
12. Hansen, P. C., "Analysis of discrete ill-posed problems by means of the L-curve," *SIAM Review*, Vol. 34, pp. 561-580 (1992).
13. Hansen, P. C. and O'Leary, D. P., "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM Journal on Scientific Computing*, Vol. 14, pp. 1487-1503 (1993).
14. Jacoby, S. L. S., Kowalik, J. S., and Pizzo, J. T., *Iterative Methods for Nonlinear Optimization Problems*, Prentice-Hall, New Jersey (1972).
15. Kunisch, K. and Zou, J., "Iterative choices of regularization parameters in linear inverse problems," *Inverse Problems*, Vol. 14, pp. 1247-1264 (1998).
16. Landweber, L., "An iteration formula for Fredholm integral equations of the first kind," *America Journal of Mathematics*, Vol. 73, pp. 615-624 (1951).
17. Latham, G. A., "Best L^2 Tikhonov analogue for Landweber iteration," *Inverse Problems*, Vol. 14, pp. 1527-1537 (1998).
18. Liu, C.-S., "Cone of non-linear dynamical system and group preserving schemes," *International Journal of Non-Linear Mechanics*, Vol. 36, pp. 1047-1068 (2001).
19. Liu, C.-S., "Group preserving scheme for backward heat conduction problems," *International Journal of Heat and Mass Transfer*, Vol. 47, pp. 2567-2576 (2004).
20. Liu, C.-S., "Nonstandard group-preserving schemes for very stiff ordinary differential equations," *CMES*: *Computer Modeling in Engineering & Sciences*, Vol. 9, pp. 255-272 (2005).
21. Liu, C.-S., "A modified Trefftz method for two-dimensional laplace equation considering the domain's characteristic length," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 21, pp. 53-65 (2007).
22. Liu, C.-S., "A highly accurate solver for the mixed-boundary potential problem and singular problem in arbitrary plane domain," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 20, pp. 111-122 (2007).
23. Liu, C.-S., "An effectively modified direct Trefftz method for 2D potential problems considering the domain's characteristic length," *Engineering analysis with boundary elements*, Vol. 31, pp. 983-993 (2007).
24. Liu, C.-S., "Solving an inverse Sturm-Liouville problem by a Lie-group method," *Boundary Value Problems*, Vol. 2008, Article ID 749865 (2008).
25. Liu, C.-S., "Identifying time-dependent damping and stiffness functions by a simple and yet accurate method," *Journal of Sound and Vibration*, Vol. 318, pp. 148-165 (2008).
26. Liu, C.-S., "A Lie-group shooting method for simultaneously estimating the time-dependent damping and stiffness coefficients," *CMES*: *Computer Modeling in Engineering & Sciences*, Vol. 27, pp. 137-149 (2008).
27. Liu, C.-S., "A time-marching algorithm for solving non-linear obstacle problems with the aid of an NCP-function," *CMC: Computers, Materials & Continua*, Vol. 8, pp. 53-65 (2008).
28. Liu, C.-S., "A fictitious time integration method for two-dimensional quasilinear elliptic boundary value problems," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 33, pp. 179-198 (2008).
29. Liu, C.-S. and Atluri, S. N., "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 (2008).
30. Liu, C.-S. and Atluri, S. N., "A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to nonlinear optimization," *CMES*: *Computer Modeling in Engineering & Sciences*, Vol. 34, pp. 155-178 (2008).
31. Liu, C.-S., Chang, J. R., Chang, K. H., and Chen, Y. W. "Simultaneously estimating the time-dependent damping and stiffness coefficients with the aid of vibrational data," *CMC*: *Computers, Materials & Continua*, Vol. 7, pp. 97-107 (2008).
32. Mickens, R. E., *Nonstandard Finite Difference Models of Differential Equations*, World Scientific, Singapore (1994).
33. Mickens, R. E., "Discretizations of nonlinear differential equations using explicit nonstandard methods," *Journal of Computational and Applied Mathematics*, Vol. 110, pp. 181-185 (1999).
34. Mickens, R. E. and Ramadhani, I., "Finite-differences having the correct linear stability properties for all finite stepsizes III," *Computers & Mathematics with Applications*, Vol. 27, pp. 77-89 (1994).
35. Mickens, R. E. and Smith, A., "Finite-difference models of ordinary differential equations: influence of denominator functions," *Journal of the Franklin Institute*, Vol. 327, pp. 143-149 (1990).
36. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in Fortran*. *The Art of Scientific Computing*, Second Ed., Cambridge University (1992).
37. Resmerita, E., "Regularization of ill-posed problems in Banach spaces: convergence rates," *Inverse Problems*, Vol. 21, pp. 1303-1314 (2005).
38. Rieder, A., "Runge-Kutta integrators yield optimal regularization schemes," *Inverse Problems*, Vol. 21, pp. 453-471 (2005).
39. Shampine, L. F. and Gear, C. W., "A user's view of solving stiff ordinary differential equations," *SIAM Review*, Vol. 21, pp. 1-17 (1979).
40. Stewart, G., *Introduction to Matrix Computations*, Academic Press, New York (1973).
41. Tikhonov, A. N. and Arsenin, V. Y., *Solutions of Ill-Posed Problems*, John-Wiley & Sons, New York (1977).
42. Wang, Y. and Xiao, T., "Fast realization algorithms for determining regularization parameters in linear inverse problems," *Inverse Problems*, Vol. 17, pp. 281-291 (2001).
43. Xiao, T., Yu, S., and Wang, Y., *The Numerical Solutions of Inverse Problems*, Science Publisher, Peiking, China (in Chinese) (2005).
44. Xie, J. and Zou, J., "An improved model function method for choosing regularization parameters in linear inverse problems," *Inverse Problems*, Vol. 18, pp. 631-643 (2002).
45. Zhu, T., Zhang, J., and Atluri, S. N., "A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and nonlinear boundary value problems," *Engineering Analysis with Boundary Elements*, Vol. 23, pp. 375-389 (1999).