Copyright © 2013 Tech Science Press CMES, vol.96, no.6, pp.409-434, 2013

A Novel Method for Solving Ill-conditioned Systems of Linear Equations with Extreme Physical Property Contrasts

Cheng-Yu Ku[1]

Abstract: This paper proposes a novel method, named the dynamical Jacobian-inverse free method (DJIFM), with the incorporation of a two-sided equilibrium algorithm for solving ill-conditioned systems of linear equations with extreme physical property contrasts. The DJIFM is based on the construction of a scalar homotopy function for transforming the vector function of linear or nonlinear algebraic equations into a time-dependent scalar function by introducing a fictitious time-like variable. The DJIFM demonstrated great numerical stability for solving linear or nonlinear algebraic equations, particularly for systems involving ill-conditioned Jacobian or poor initial values that cause convergence problems. With the incorporation of a newly developed two-sided equilibrium algorithm, the solution of layered problems with extreme contrasts in the physical property that are typically highly ill-conditioned can be solved. The proposed method was then adopted for the solution of several highly ill-conditioned numerical examples, including the linear Hilbert matrix, linear Vandermonde matrix, layered linear and nonlinear groundwater flow problems. The results revealed that using the DJIFM and the two-sided equilibrium algorithm can improve the convergence and increase the numerical stability for solving layered problems.

Keywords: Ill-conditioned, layered problem, the dynamical Jacobian-inverse free method (DJIFM), Jacobian, linear algebraic equations.

1 Introduction

Layered problems with extreme contrasts in physical properties are often encountered in engineering problems, such as in the analyses of heat conductivity, in the temperature of layered plates, the presence of oil and natural gas in a reservoir

[1] Department of Harbor and River Engineering, National Taiwan Ocean University, Keelung 20224, Taiwan.
Corresponding author. Email: chkst26@mail.ntou.edu.tw

of rock formations, and the seepage problems in layered soils. These physical problems are highly complex because the materials typically consist of layers with extreme contrasts in physical properties. For example, the hydraulic conductivity of sand is in the order of 1 to 10^{-3} cm/s, whereas the hydraulic conductivity of clay is less than 10^{-6} cm/s [Das (2010)]. Hence, a contrast of 10^{-6} is commonly encountered in engineering practice. Other applications where the coefficients have large discontinuities include electrical power networks, groundwater flow, semiconductors, and electromagnetic modeling [Coomer and Graham (1996); Vuik, Segal, Meijerinky and Wilma (2001)].

The numerical methods used in computational mechanics, including the finite difference method, finite element method, boundary element method, and meshless method [Atluri (2002)], typically lead to a system of linear algebraic equations (LAEs) or nonlinear algebraic equations (NAEs) that must be solved. Problems regarding radical changes in material properties frequently generate ill-conditioned matrices. The illness of a system of LAEs arises from the internal interfaces between media of radically different properties. In practical applications, numerous systems of LAEs are often encountered. The matrices of systems of LAEs are usually sparse, but because of fill-ins, a direct method requires too much memory to fit in the core. Therefore, iterative methods are likely the most suitable candidates for solving the linear systems of equations. Over the years, various contributions have been made toward the numerical solution of LAEs [Liu (2013a); Liu (2014)]. An iterative-based method, such as the conjugate gradient method (CGM) [Press, et al. (2007); Liu and Chang (2009)], is perhaps the best known for producing increasingly accurate approximations to the solutions of a system of LAEs.

It is well known that the convergence of the CGM depends on the distribution of the eigenvalues of the system of LAEs. When a system of LAEs is extremely ill-conditioned, the convergence of the regular numerical method can be unacceptably slow or might not be able to converge [Vuik, Segal and Meijerinky (1999); Liu (2012)]. To enable the use of the numerical method for extremely ill-conditioned problems [Liu, Hong and Atluri (2010)], Liu (2013b) introduced a two-sided equilibrium method to reduce the condition number of an ill-posed linear system and combined the two-sided equilibrium method with the CGM to solve linear inverse problems and the linear Hilbert problem under enormous random noise.

In addition to the CGM, the fictitious time integration method (FTIM) was first used to solve a nonlinear system of algebraic equations (NAEs) by introducing fictitious time [Liu (2008); Ku, Yeih, Liu and Chi (2009)]. The fixed point of these evolution equations, which is the root of the original algebraic equation, is obtained by applying numerical integrations on the resultant ordinary differential equations, which do not require the derivatives of NAEs and their inverses. Based on a time marching algorithm, the general dynamical method [Ku, Yeih and Liu (2011)] was proposed using a scalar homotopy function to transform a vector function of LAEs or NAEs into a time-dependent scalar function by introducing a fictitious time-like variable, and this method demonstrated great numerical stability for solving LAEs or NAEs, particularly for systems involving ill-conditioned Jacobian or poor initial values that cause convergence problems. With the incorporation of a newly developed two-sided equilibrium algorithm, the solution of layered problems with extreme contrasts in physical properties, which are typically highly ill-conditioned, can be solved. The proposed method was then adopted for the solution of several highly ill-conditioned numerical examples, including the linear Hilbert matrix, linear Vandermonde matrix, and one-dimensional layered linear and nonlinear groundwater flow problems.

2 The Dynamical Jacobian-Inverse Free Method (DJIFM)

This section demonstrates the formulation of the dynamical Jacobian-inverse free method (DJIFM). First, we consider the following LAEs or NAEs:

$$F_i\left(x_1,\ldots,x_n\right)=0,\ i=1,\ldots,n. \tag{1}$$

Using $\mathbf{x}:=(\mathrm{x}_1,\ldots,\mathrm{x}_\mathrm{n})^\mathrm{T}$ and $\mathbf{F}:=(\mathrm{F}_1,\ldots,\mathrm{F}_\mathrm{n})^\mathrm{T}$, Eq. (1) can be written as $\mathbf{F}(\mathbf{x})=\mathbf{0}$. Solving Eq. (1) by using a first-order Taylor approximation, we can easily see that the Newton method for solving $\mathbf{F}(\mathbf{x})=\mathbf{0}$ is given by

$$\mathbf{x}^{\mathrm{k}+1}=\mathbf{x}^{\mathrm{k}}-[\mathbf{B}(\mathbf{x}^{\mathrm{k}})]^{-1}\mathbf{F}(\mathbf{x}^{\mathrm{k}}), \tag{2}$$

where B is an $n\times n$ Jacobian matrix with its ijth component being given by $\partial F_i/\partial x_j$. The Newton method is only guaranteed to a local convergence if certain conditions are satisfied; hence, depending on the type of function used and the initial guess of a solution, it might or might not converge. In addition, using the Newton method in the computations of the Jacobian matrix and its inverse at each iterative step is expensive. However, for solving LAEs or NAEs,

$$\mathbf{F}(\mathbf{x})=\mathbf{0}, \tag{3}$$

the homotopy method is a technique for enhancing the convergence from a local convergence to a global convergence. All the homotopy methods are based on the construction of a vector function, $\mathbf{H}(\mathbf{x},\boldsymbol{\tau})$, which is called the homotopy function. The homotopy function is the objective of continuously transforming a function $\mathbf{G}(\mathbf{x})$ into $\mathbf{F}(\mathbf{x})$ by introducing a homotopy parameter, τ. The homotopy parameter τ can be treated as a fictitious time-like variable, and the homotopy function can be any continuous function, such that $\mathbf{H}(\mathbf{x},\ 0)=\mathbf{G}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x},\ 1)=\mathbf{F}(\mathbf{x})$. Hence, we construct $\mathbf{H}(\mathbf{x},\ 0)$ in such a way that its zeros are easily determined, and also require that once the parameter $\boldsymbol{\tau}$ is equal to 1, $\mathbf{H}(\mathbf{x},\boldsymbol{\tau})$ coincides with the original function $\mathbf{F}(\mathbf{x})$.

Among the various homotopy functions that are generally used, the fixed point homotopy function (i.e., $\mathbf{G}(\mathbf{x}) = \mathbf{x}-\mathbf{x}_0$) and the Newton homotopy function (i.e., $\mathbf{G}(\mathbf{x}) = \mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}_0)$) are simple and powerful functions that can be successfully applied to various problems. The fixed point homotopy function can be written as

$$\mathbf{H}(\mathbf{x},\ \boldsymbol{\tau}) = \boldsymbol{\tau}\mathbf{F}(\mathbf{x}) + (1-\boldsymbol{\tau})[\mathbf{x}-\mathbf{x}_0] = 0, \tag{4}$$

and the Newton homotopy function is

$$\mathbf{H}(\mathbf{x},\ \boldsymbol{\tau}) = \boldsymbol{\tau}\mathbf{F}(\mathbf{x}) + (1-\boldsymbol{\tau})[\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}_0)] = 0, \tag{5}$$

where $\mathbf{x}_0$ is the given initial values and $\boldsymbol{\tau} \in [0,1]$. To conduct a scalar-based homotopy continuation method, we convert the vector equation of $\mathbf{F} = 0$ to a scalar equation by noticing that

$$\mathbf{F} = 0 \ \Leftrightarrow\ \|\mathbf{F}\|^2 = 0, \tag{6}$$

where $\|\mathbf{F}\|^2 = F_1^2 + F_2^2 + \ldots + F_n^2$. Clearly, the left-hand side implies the right-hand side. Conversely, by $\|\mathbf{F}\|^2 = F_1^2 + F_2^2 + \ldots + F_n^2 = 0$ we have $F_1 = F_2 = \ldots = F_n = 0$, thus $\mathbf{F} = 0$.

Based on the fixed point homotopy function, Liu, Yeih, Kuo, and Atluri (2009) developed a scalar homotopy function:

$$h(\mathbf{x},\boldsymbol{\tau}) = \frac{1}{2}\boldsymbol{\tau}\,\|\mathbf{F}(\mathbf{x})\|^2 + \frac{1}{2}(\boldsymbol{\tau}-1)\,\|\mathbf{x}-\mathbf{x}_0\|^2 = 0. \tag{7}$$

The scalar homotopy method retains the merits of the homotopy method, such as global convergence, but does not involve the complex computation of the inverse of the Jacobian. The scalar homotopy method, however, requires an extremely small time step to reach the fictitious time, $\boldsymbol{\tau} = 1$, which results in a slower convergence compared with other methods. In this paper, we propose a scalar homotopy algorithm based on the Newton homotopy function as described in Eq. (5), which can also be written as follows:

$$\mathbf{H}(\mathbf{x},\boldsymbol{\tau}) = \mathbf{F}(\mathbf{x}) + (\boldsymbol{\tau}-1)\mathbf{F}(\mathbf{x}_0) = 0. \tag{8}$$

Using Eq. (6), we can transform the vector equation into a fictitious time-dependent scalar function $h(\mathbf{x},\boldsymbol{\tau})$ as follows:

$$h(\mathbf{x},\boldsymbol{\tau}) = \frac{1}{2}\,\|\mathbf{F}(\mathbf{x})\|^2 + \frac{1}{2}(\boldsymbol{\tau}-1)\,\|\mathbf{F}(\mathbf{x}_0)\|^2 = 0. \tag{9}$$

Equation (9) holds for all $\boldsymbol{\tau} \in [0,1]$. To motivate this study, we first consider a fictitious time function $Q(t)$, where t is the fictitious time and $Q(t)$ must satisfy that $Q(t) > 0$, $Q(0) = 1$, and $Q(t)$ is a monotonically increasing function of t, and $Q(\infty) = \infty$. We then introduce the proposed fictitious time function $Q(t)$ into Eq. (9) and obtain

$$h(\mathbf{x},\mathrm{t}) = \frac{1}{2}\left\|\mathbf{F}(\mathbf{x})\right\|^2 - \frac{1}{2}\frac{1}{\mathrm{Q(t)}}\left\|\mathbf{F}(\mathbf{x}_0)\right\|^2 = 0, \tag{10}$$

Using the fictitious time function, $Q(t)$, when the fictitious time $t = 0$ and $t = \infty$, we can obtain

$$h(\mathbf{x},\mathrm{t}=0) = \frac{1}{2}\left\|\mathbf{F}(\mathbf{x})\right\|^2 - \frac{1}{2}\left\|\mathbf{F}(\mathbf{x}_0)\right\|^2 = 0 \;\Leftrightarrow \mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_0) \tag{11}$$

$$h(\mathbf{x},\mathrm{t}=\infty) = \frac{1}{2}\left\|\mathbf{F}(\mathbf{x})\right\|^2 = 0 \;\Leftrightarrow \mathbf{F}(\mathbf{x}) = 0. \tag{12}$$

As the homotopy parameter τ gradually varies from 0 to 1, the tracking of a solution path for the proposed scalar Newton homotopy function is clearly equivalent to the fictitious time varying from $t = 0$ to $t = \infty$.

If we assume that $h(\mathrm{x},t) = 0$ is satisfied for any time greater than zero, by multiplying $Q(t)$ at both sides of Eq. (10) we obtain

$$h(\mathbf{x},\mathrm{t}) = \frac{1}{2}\mathrm{Q(t)}\left\|\mathbf{F}(\mathbf{x})\right\|^2 - \frac{1}{2}\left\|\mathbf{F}(\mathbf{x}_0)\right\|^2 = 0. \tag{13}$$

Liu, Yeih, Kuo, and Atluri (2009) and Ku, Yeih, and Liu (2010) used the fixed point homotopy function and the Newton homotopy function, respectively, to construct an analogy of the scalar homotopy method to the theory of plasticity. In their explanation, the aforementioned assumption was equivalent to the stability in small for the plasticity theory. Considering the consistency condition, we derive from Eq. (13) that

$$\frac{dh}{dt} = \frac{\partial h}{\partial t} + \frac{\partial h}{\partial \mathrm{x}} \cdot \frac{d\mathrm{x}}{dt} = 0. \tag{14}$$

The derivatives of the scalar function, $h(\mathbf{x},\mathrm{t})$, with respect to $\mathbf{x}$ and t can be written as

$$\frac{\partial h}{\partial t} = \frac{1}{2}\dot{Q}(t)\left\|\mathbf{F}(\mathbf{x})\right\|^2 \quad \text{and} \quad \frac{\partial h}{\partial \mathrm{x}} = Q(t)\mathbf{B}^{\mathrm{T}}\mathbf{F}(\mathbf{x}) \tag{15}$$

Let $\dot{\mathbf{x}} = \frac{\mathrm{dx}}{\mathrm{dt}}$, and a possible solution of Eq. (14) for $\dot{\mathbf{x}}$ is

$$\dot{\mathbf{x}} = \lambda\mathbf{TF}. \tag{16}$$

By inserting Eq. (15) and Eq. (16) into Eq. (14), we can derive

$$\lambda = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x})\mathbf{BTF}(\mathbf{x})}. \tag{17}$$

In Eq. (16), $\mathbf{T}$ is the transformation matrix that can be $\mathbf{B}^{\text{-}1}$, the identity matrix $\mathbf{I}$, $\mathbf{B}^{\mathrm{T}}$, or any other square matrices. By inserting Eq. (16) into Eq. (15), we obtain

$$\dot{x} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathrm{F}(\mathrm{x})\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x})\mathrm{BTF}(\mathrm{x})}\mathrm{TF}(\mathrm{x}). \tag{18}$$

Equation (17) is the general dynamical equation used for solving LAEs or NAEs. Numerous methods can be used for selecting a suitable function of $Q(t)$. Based on the FTIM first proposed by Liu and Atluri (2008), the NAEs, $\mathbf{F}(\mathbf{x}) = 0$, can be embedded in a system of nonlinear ordinary differential equations: $\dot{\mathbf{x}} = -\mathrm{v}/\mathrm{q}(\tau)\mathbf{F}(\mathbf{x})$, where τ is the fictitious time, and $q(\tau)$ is a monotonically increasing function of τ. In their study, a simple time-like function of $q(\tau) = (1+\tau)$ was used. In addition to this original simple time-like function, Ku, Yeih, Liu, and Chi (2009) proposed a more general function, $q(\tau) = (1+\tau)^m$. Based on a similar idea and by replacing τ with t, we can let

$$\frac{\dot{Q}(t)}{Q(t)} = \frac{v}{(1+t)^m},\ 0 < m \leq 1. \tag{19}$$

Hence, we obtain

$$Q(t) = \exp\left[\frac{v}{1-m}[(1+t)^{1-m} - 1]\right]. \tag{20}$$

Inserting Eq. (19) into Eq. (18), we obtain

$$\dot{\mathbf{x}} = \frac{-\mathrm{v}}{2\,(1+\mathrm{t})^{\mathrm{m}}} \frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x})\mathbf{BTF}(\mathbf{x})}\mathbf{TF}(\mathbf{x}) \tag{21}$$

where m is a control parameter for hastening the convergence, as discussed in Ku, et al. (2009), and v is a damping parameter introduced by Liu and Atluri (2008) for improving the convergence. By satisfying the conditions that $Q(t) > 0$, $Q(0) = 1$, $Q(t)$ are monotonically increasing functions of t, and $Q(\infty) = \infty$, another suitable function of $Q(t)$ can be easily determined and written as

$$Q(t) = e^t. \tag{22}$$

By inserting Eq. (22) into Eq. (19), we obtain

$$\frac{\dot{Q}(t)}{Q(t)} = 1. \tag{23}$$

Again, by inserting Eq. (23) into Eq. (18), we obtain

$$\dot{\mathbf{x}} = -\frac{1}{2}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x})\mathbf{BTF}(\mathbf{x})}\mathbf{TF}(\mathbf{x}) \tag{24}$$

We can clearly observe that Eqs. (21) and (24) embed the fictitious time function in the evolution of the solution search. To calculate Eqs. (21) and (24), we can employ a forward Euler scheme and obtain the following equations:

$$\mathbf{x}^{\mathrm{k+1}} = \mathbf{x}^{\mathrm{k}} - \frac{\mathrm{h_t v}}{2(1+\mathrm{t})^{\mathrm{m}}}\frac{\|\mathbf{F}(\mathbf{x}^{\mathrm{k}})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x}^{\mathrm{k}})\mathbf{B}(\mathbf{x}^{\mathrm{k}})\mathrm{T}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}})}\mathbf{T}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}}). \tag{25}$$

$$\mathbf{x}^{\mathrm{k+1}} = \mathbf{x}^{\mathrm{k}} - \frac{\mathrm{h_t}}{2}\frac{\|\mathbf{F}(\mathbf{x}^{\mathrm{k}})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x}^{\mathrm{k}})\mathbf{B}(\mathbf{x}^{\mathrm{k}})\mathbf{T}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}})}\mathbf{T}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}}). \tag{26}$$

where h_t is the fictitious time step. In these equations, it is determined that the numerators and denominators of the fractions in Eqs. (25) and (26) are scalars if we adopt any one of the transformation matrices from $\mathbf{B}^{-1}$, $\mathbf{I}$, and $\mathbf{B}^{\mathrm{T}}$. To derive the DJIFM, we let the transformation matrix, T, be the identity matrix, $\mathbf{I}$; thus, Eq. (18) can be rewritten as

$$\dot{\mathbf{x}} = -\frac{\dot{\mathrm{Q}}(\mathrm{t})}{2\mathrm{Q}(\mathrm{t})}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x})\mathbf{BF}(\mathbf{x})}\mathbf{F}(\mathbf{x}). \tag{27}$$

If we select the fictitious time function as demonstrated in Eq. (20), we derive the DJIFM as

$$\dot{\mathbf{x}} = -\frac{\mathrm{v}}{2(1+\mathrm{t})^{\mathrm{m}}}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x})\mathbf{BF}(\mathbf{x})}\mathbf{F}(\mathbf{x}). \tag{28}$$

Equation (28) is similar to the FTIM; however, it does not have the fractional item in the original FTIM shown in Eq. (29).

$$\dot{\mathbf{x}} = -\frac{\mathrm{v}}{(1+\mathrm{t})^{\mathrm{m}}}\mathbf{F}(\mathbf{x}) \tag{29}$$

By using the forward Euler scheme, we obtain

$$\mathbf{x}^{\mathrm{k+1}} = \mathbf{x}^{\mathrm{k}} - \frac{\mathrm{h_t v}}{2(1+\mathrm{t})^{\mathrm{m}}}\frac{\|\mathbf{F}(\mathbf{x}^{\mathrm{k}})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x}^{\mathrm{k}})\mathbf{B}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}})}\mathbf{F}(\mathbf{x}^{\mathrm{k}}). \tag{30}$$

Similarly, using the fictitious time function $Q(t) = e^t$, we derive the DJIFM as

$$\mathbf{x}^{\mathrm{k+1}} = \mathbf{x}^{\mathrm{k}} - \frac{\mathrm{h_t}}{2}\frac{\|\mathbf{F}(\mathbf{x}^{\mathrm{k}})\|^2}{\mathbf{F}^{\mathrm{T}}(\mathbf{x}^{\mathrm{k}})\mathbf{B}(\mathbf{x}^{\mathrm{k}})\mathbf{F}(\mathbf{x}^{\mathrm{k}})}\mathbf{F}(\mathbf{x}^{\mathrm{k}}). \tag{31}$$

In Eqs. (30) and (31), it is determined that the numerators and denominators of the fractions are merely scalars. Accordingly, we can avoid computing the inverse of the Jacobian matrix, thus improving the numerical stability.

3 A Two-Sided Equilibration Algorithm

For a given system of ill-conditioned LAEs, $\mathbf{Ax} = \mathbf{b}$, the condition number can be used to evaluate whether a given nonsingular matrix $\mathbf{A}$ is ill-conditioned. The condition number of a square nonsingular matrix $\mathbf{A}$ is defined by

$$Cond(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}\|^{-1}. \tag{32}$$

where the matrix norm is the Frobenius norm. If the $\mathbf{A}$ matrix is singular, the condition number is infinite. The condition number of the matrix represents the ratio of the maximum relative stretching to the maximum relative shrinking that the matrix does to any nonzero vectors. Therefore, it is a measure of how close a matrix $\mathbf{A}$ is to being singular: a matrix with a large condition number is nearly singular, whereas a matrix with a condition number close to 1 is far from being singular.

For any matrix $\mathbf{A}$ and a scalar α, a crucial property exists:

$$Cond(\alpha\mathbf{A}) = \mathrm{Cond}(\mathbf{A}). \tag{33}$$

This property shows that it is possible to diminish $Cond(\mathbf{A})$ by multiplying every row and every column of the matrix $\mathbf{A}$ by a suitable set of scaling numbers. In this study, we adopted the two-sided equilibrium method proposed by Liu (2013) to determine the optimal set of scaling numbers for a given system of ill-conditioned LAEs, $\mathbf{Ax} = \mathbf{b}$. We first rewrite $\mathbf{Ax} = \mathbf{b}$ by using the matrix operation $\mathbf{QAPy} = \mathbf{Qb}$, where $\mathbf{x} = \mathbf{Py}$.

$\mathbf{QAPy} = \mathbf{Qb}$ can be written as $\mathbf{Cy} = \mathbf{B}$, where $\mathbf{C} = \mathbf{QAP}$ and $\mathbf{B} = \mathbf{Qb}$. The $\mathbf{Q}$ and $\mathbf{P}$ matrices represent the preconditioner and postconditioner, respectively. The purpose of the $\mathbf{Q}$ and $\mathbf{P}$ matrices is to reduce the illness of the $\mathbf{A}$ matrix in the original ill-conditioned system. The $\mathbf{P}$ matrix is assumed to be a diagonal matrix:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & P_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & P_{n-1} & 0 \\ 0 & \cdots & 0 & 0 & P_n \end{bmatrix} \tag{34}$$

To reduce the condition number, it is trying to find $P_k,\ k = 1,...,n$ and letting each column of the coefficient in matrix $\mathbf{C}$ have the same Euclidean norm. Therefore, we obtain

$$\sum_{j=1}^{n} C_{j1}^2 = \sum_{j=1}^{n} C_{j2}^2 = \cdots \sum_{j=1}^{n} C_{jn}^2. \tag{35}$$

where C_{ij} denotes the ij^{th} component of $\mathbf{C}$. Accordingly, the P_k can be solved by

$$P_k = \alpha \left(\frac{\sum\limits_{j=1}^{n} C_{j1}^2}{\sum\limits_{j=1}^{n} C_{jk}^2} \right)^{\frac{1}{2}}, \quad k = 2, \cdots, n. \tag{36}$$

The scalar α is an amplification factor that can be used to further reduce the condition number. Similarly, the $\mathbf{Q}$ preconditioner can be obtained by performing the same procedure. To determine the optimal $\mathbf{Q}$ and $\mathbf{P}$ conditioners, an iterative sequence is necessary. After we obtain the $\mathbf{Q}$ and $\mathbf{P}$ conditioners, we can easily convert the original system of ill-conditioned LAEs, $\mathbf{Ax} = \mathbf{b}$ to $\mathbf{Cy} = \mathbf{B}$. Therefore, Eq. (3) can be rewritten as

$$\mathbf{F}(\mathbf{y}) = \mathbf{Cy} - \mathbf{B} = 0 \tag{37}$$

By using the DJIFM, we obtain

$$\mathbf{y}^{\mathrm{k}+1} = \mathbf{y}^{\mathrm{k}} - \frac{\mathrm{h_t v}}{2\,(1+\mathrm{t})^{\mathrm{m}}} \frac{\left\|\mathbf{F}(\mathbf{y}^{\mathrm{k}})\right\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{y}^{\mathrm{k}})\mathbf{B}(\mathbf{y}^{\mathrm{k}})\mathbf{F}(\mathbf{y}^{\mathrm{k}})} \mathbf{F}(\mathbf{x}^{\mathrm{k}}). \tag{38}$$

Because the condition number of the $\mathbf{C}$ matrix has been reduced, the system of LAEs, $\mathbf{Cy} = \mathbf{B}$, can be solved, obtaining highly accurate results. Moreover, the limitation of convergence for solving the ill-conditioned LAEs can also be released. Finally, the unknowns of $\mathbf{x}$ can be obtained using $\mathbf{x} = \mathbf{Py}$. In order to validate the capability of the above proposed method for solving highly ill-conditioned problems, the examples for solving the Hilbert matrix and the Vandermonde matrix were conducted as follows.

3.1 The linear Hilbert problem

The Hilbert matrix is an example of a highly ill-conditioned matrix. The Hilbert matrix, introduced by Hilbert (1894), is a square matrix that has entries as unit fractions.

$$H_{ij} = \frac{1}{i+j-1}, \quad i = 1, \cdots, n \text{ and } j = 1, \cdots, n \tag{39}$$

A n by n Hilbert matrix can be written as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/4 & 1/5 \\ 1/2 & 1/3 & 1/4 & 1/5 & 1/6 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/7 \\ 1/4 & 1/5 & 1/6 & \ddots & \vdots \\ 1/5 & 1/6 & 1/7 & \cdots & \frac{1}{i+j-1} \end{bmatrix} \tag{40}$$

In this example, we considered a highly ill-conditioned linear system as follows:

$$\mathbf{Hx} = \mathbf{b}_0. \tag{41}$$

where $\mathbf{b}_0 = \mathrm{b}_0^{\mathrm{i}} = \sum_{\mathrm{j}=1}^{\mathrm{n}} \frac{1}{\mathrm{i}+\mathrm{j}-1}$ with the consideration of the solution of $x_j = 1$.

Conducting this test demonstrated the numerical stability of the DJIFM. In this example, the parameters used for the DJIFM were $m = 0.01$, $h_t = 2.0$, and $\nu = 1.0$. We started from an initial value of 0 for all x_i. The root mean square norm (RMSN) was set as the stopping criterion and it was 10^{-6} for all the methods in this example. We first solved the Hilbert matrix with $n = 500$, using the CGM, the DJIFM, the CGM with the two-sided equilibrium algorithm, and the DJIFM with the two-sided equilibrium algorithm. The condition number of the Hilbert matrix with $n = 500$ was 2.15×10^{20}. Using the two-sided equilibrium algorithm, the condition number of the Hilbert matrix with $n = 500$ reduced to 9.47×10^{19}. Unexpectedly, the condition number reduced by approximately only one order. However, it was determined that the accuracy can be improved from 10^{-2}–10^{-3} to 10^{-4}–10^{-5}, as shown in Fig. 1.

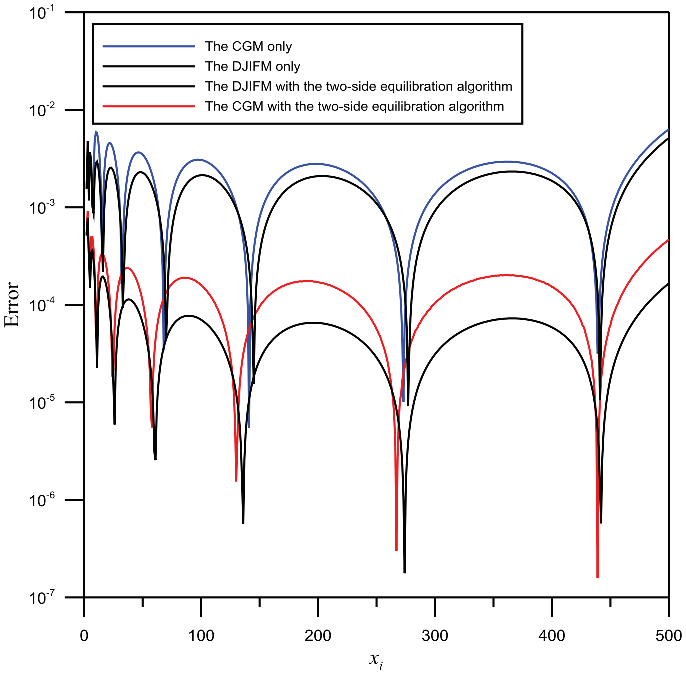

Figure 1: Comparison of error for solving the Hilbert matrix with $n = 500$

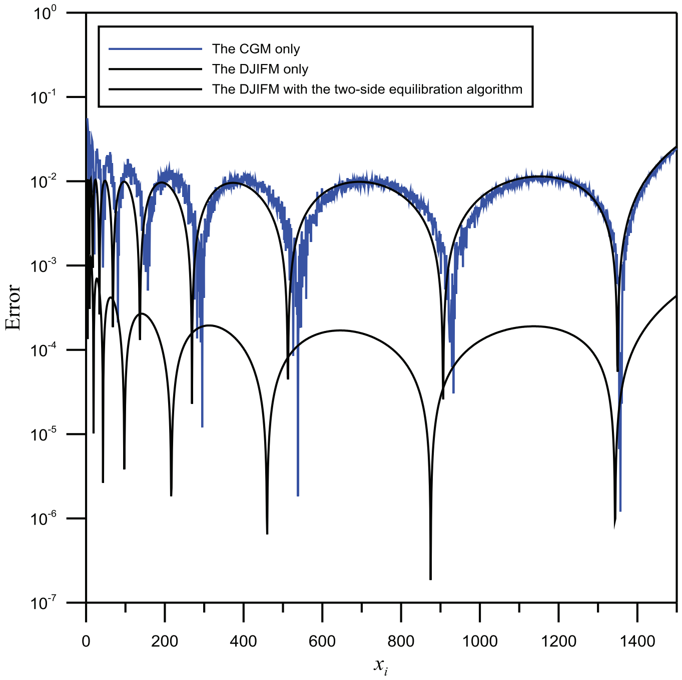

Figure 2: Comparison of error for solving the Hilbert matrix with $n = 1500$

We also solved the Hilbert matrix with $n = 1,500$ using the CGM, the DJIFM, the CGM with the two-sided equilibrium algorithm, and the DJIFM with the two-sided equilibrium algorithm. It is known that the illness of the Hilbert matrix increases when the value of n increases. The condition number of the Hilbert matrix with $n = 1,500$ was 5.13×10^{21}. Using the two-sided equilibrium algorithm, the condition number reduced to 7.58×10^{20}. Figure 2 shows that when the CGM and the DJIFM were implemented without using the two-sided equilibrium algorithm, the accuracy was only 10^{-2}. It was also determined that the CGM with the two-sided equilibrium algorithm is not able to converge within 10,000 iterations. An accuracy of 10^{-3} to 10^{-4} was obtained only by using the DJIFM with the two-sided equilibrium algorithm.

3.2 The linear Vandermonde problem

The Vandermonde matrix is a matrix where the first row is the first value evaluated at each of the n monomials, the second row is the second value evaluated at each of the n monomials, and so on. The Vandermonde matrix is another example of a highly ill-conditioned matrix. The Vandermonde matrix of order n has the

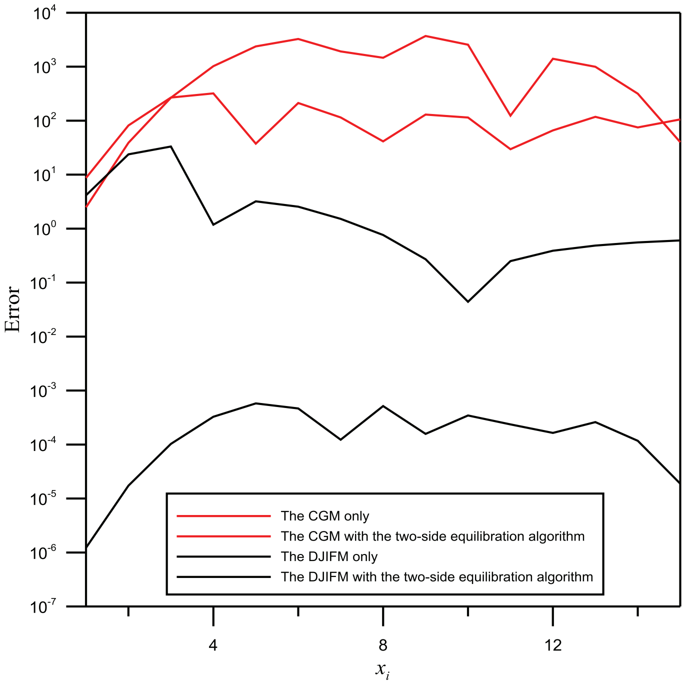

Figure 3: Comparison of error for solving the Vandermonde matrix.

following form:

$$\mathbf{V} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ 1 & x_3 & x_3^2 & \cdots & x_3^{n-1} \\ 1 & \vdots & \vdots & \cdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^{n-1} \end{bmatrix} \tag{42}$$

In this example, we considered a highly ill-conditioned linear system as follows:

$$\mathbf{Vy} = \mathbf{b}_0. \tag{43}$$

where $\mathbf{b}_0 = \mathrm{V} \times 1$, where 1 is the all-ones vector. Accordingly, the exact solution of vector y must be the vector of 1. To demonstrate the DJIFM's ability to handle highly ill-conditioned linear systems, we constructed the Vandermonde matrix by using the vector, **x**, which is the product of the Hilbert matrix and all-ones vector as

$$\mathbf{x} = \mathbf{H} \times 1 \quad \text{where} \quad \mathbf{x} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \end{bmatrix}^{\mathrm{T}}. \tag{44}$$

In the second example, the Vandermonde matrix shown in Eq. (43) is highly ill-conditioned. We solved the Vandermonde matrix with $n = 15$ using the CGM, the DJIFM, the CGM with the two-sided equilibrium algorithm, and the DJIFM with the two-sided equilibrium algorithm. The DJIFM parameters used in this example were $m = 0.01$, $h_t = 2.0$, and $\nu = 1.0$. We started from an initial value of 0 for all x_i. The RMSN was set as the stopping criterion and it was of 10^{-4} for all the methods in this example. The condition number of this example was 3.32×10^{19}. Using the two-sided equilibrium algorithm, the condition number reduced to 8.88×10^{12}. Figure 3 shows that the CGM is not able to converge, even when the two-sided equilibrium algorithm is incorporated. An accuracy of 10^{-3} to 10^{-5} was obtained only by using the DJIFM with the two-sided equilibrium algorithm.

4 Application Examples

4.1 Layered problems with an ill-conditioned system of LAEs

The above two examples in the previous section demonstrated that the DJIFM with the two-sided equilibrium algorithm can be used to handle highly ill-conditioned linear systems such as the Hilbert and Vandermonde matrices. The following example is a layered problem with an ill-conditioned system of LAEs. First, let us consider the flow of water through porous soils [Strack (1989)]. Considering a one-dimensional partial differential equation of flow, we have

$$\frac{\partial}{\partial x} k(x) \frac{\partial h}{\partial x} = 0 \tag{45}$$

where $k(x)$ is the hydraulic conductivity. This one-dimensional problem has a domain length of $0 \leq x \leq L_x$. In the geotechnical field, the flow of water through porous soils typically signifies seepage problems. The seepage problems in layered soils often consist of soil layers with the hydraulic conductivity coefficients of sandy and clayey soils. In this example, the coefficients of the hydraulic conductivity for two straight soil layers are $k(x) = ka$ within $0 \leq x < L_x/2$, and $k(x) = kb$ within $L_x/2 \leq x < L_x$. Equation (45) does not hold on an interface between soils of different hydraulic conductivity because the hydraulic conductivity and the hydraulic gradient are not continuous there. In the case of the horizontal interface shown in Fig. 4, the discharge velocity can only be defined on each side of the interface by introducing a finite difference discretization:

$$v_x^1 = k_a \frac{h_i - h_{i-1}}{\Delta x} \text{ and } v_x^2 = k_b \frac{h_{i+1} - h_i}{\Delta x} \tag{46}$$

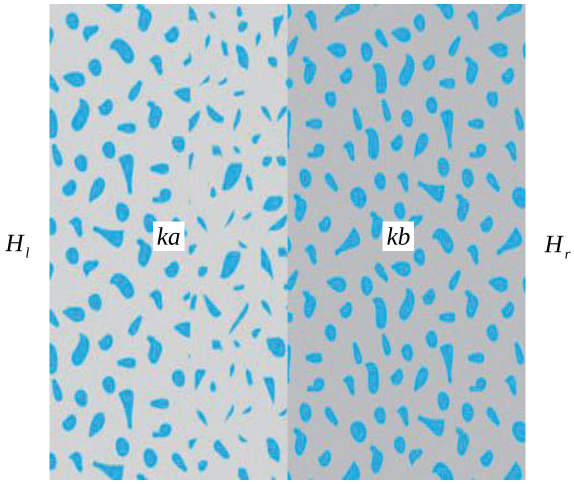

Figure 4: Configuration of the flow of water through two porous layered soils.

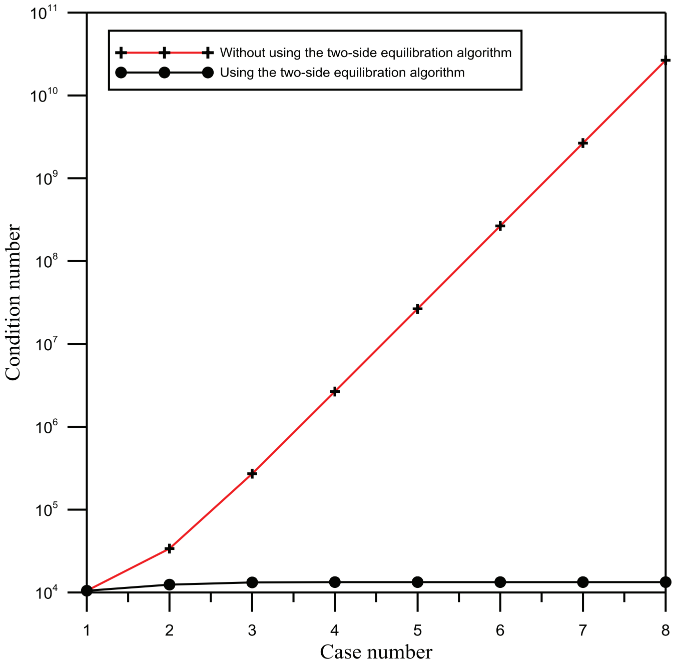

Figure 5: The condition number for different cases.

where v_x^1 and v_x^2 are the x components of the discharge velocities in medium of the hydraulic conductivities k_a and k_b, respectively. Because of the water flux conservation across the interface $v_x^1 = v_x^2$, Eq. (46) becomes

$$k_a \frac{h_i - h_{i-1}}{\Delta x} = k_b \frac{h_{i+1} - h_i}{\Delta x}. \tag{47}$$

Rewriting the previous equation, we have the following equation:

$$k_b h_{i+1} - (k_a + k_b) h_i + k_a h_{i-1} = 0 \tag{48}$$

Again, by introducing a finite difference discretization with $\Delta x = 1$ at grid points with Dirichlet boundary conditions in which the groundwater heads on the left and right boundaries are $h(0) = H_l$ and $h(L_x) = H_r$, respectively, the matrix form of Eq. (45) can be written as

$$\begin{bmatrix} -2k_a & k_a & 0 & \cdots & & 0 \\ k_a & \ddots & k_a & & & \vdots \\ 0 & k_a & -(k_a+k_b) & k_b & & \\ \vdots & & k_b & -2k_b & k_b & \\ & & & & \ddots & 0 \\ 0 & \cdots & & 0 & k_b & -2k_b \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ \\ h_n \end{bmatrix} = \begin{bmatrix} -k_a H_l \\ 0 \\ \vdots \\ 0 \\ -k_b H_r \end{bmatrix} \tag{49}$$

By introducing a finite difference discretization with $\Delta x = 1$ at 161 grid points, with Dirichlet boundary conditions in which the groundwater heads on the left and right boundaries are $h(x = 0) = 8$ and $h(x = 160) = 4$, respectively, Eq. (49) can be written as Ax = b. The typical hydraulic conductivity values of saturated soils are listed in Table 1 [Das (2010)]. From Table 1, radical changes in the hydraulic conductivity of different saturated soils can be determined. To demonstrate the illness of layered soils with extreme hydraulic conductivity contrast, we considered sand with a fixed hydraulic conductivity of 1 cm/s as the first layer of soil, and the second layer of soil was considered to gradually change from coarse sand to extremely fine clay. Accordingly, the second layer has the hydraulic conductivity from 1 cm/sec to 10^{-7} cm/s, as presented in Table 2. This dramatic change in hydraulic conductivity can cause an ill-conditioned system in LAEs. Figure 5 shows the condition number for Cases 1 to 8. Fig. 5 indicates that the two-sided equilibrium algorithm can effectively reduce the condition number and maintain it in the order of 10^{-4}.

Again, we solved this example by using the CGM, the DJIFM, the CGM with the two-sided equilibrium algorithm, and the DJIFM with the two-sided equilibrium algorithm. The DJIFM parameters used in this example were $m = 0.01$, $h_t = 2.0$,

Table 1: Typical values of hydraulic conductivity of saturated soils.

Soil type	Hydraulic conductivity (cm/sec)
Clean gravel	100∼1
Coarse sand	1∼0.01
Fine sand	0.01∼0.001
Silty sand	$10^{-3} \sim 10^{-5}$
Clay	$< 10^{-6}$

Table 2: The hydraulic conductivity of the flow of water through two porous layered soils.

Case number	1	2	3	4	5	6	7	8
ka	1	1	1	1	1	1	1	1
kb	1	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}

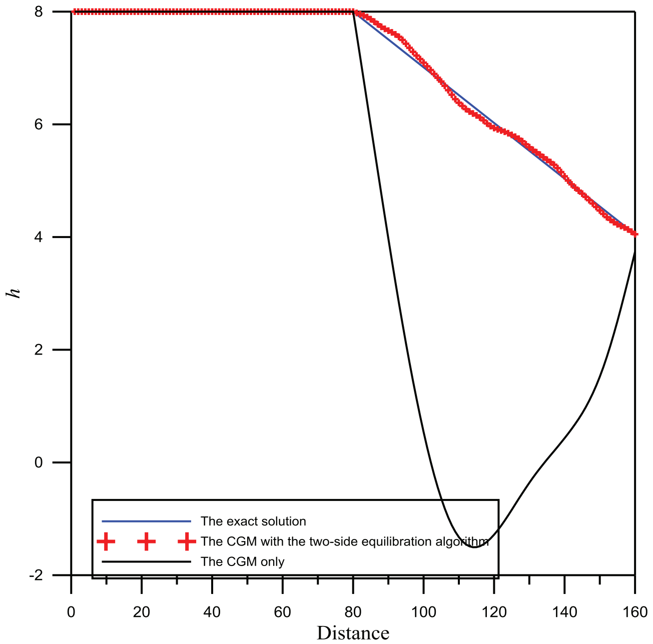

Figure 6: The solution of case 6 using the CGM with and without the two-side equilibration algorithm.

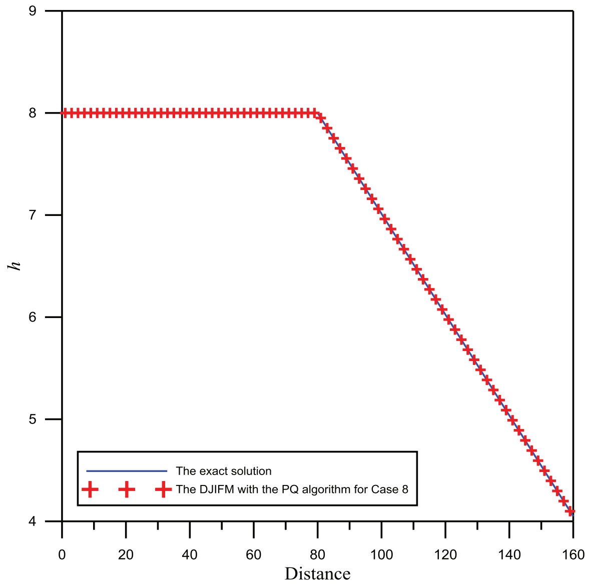

Figure 7: Comparison of the numerical and the exact solutions.

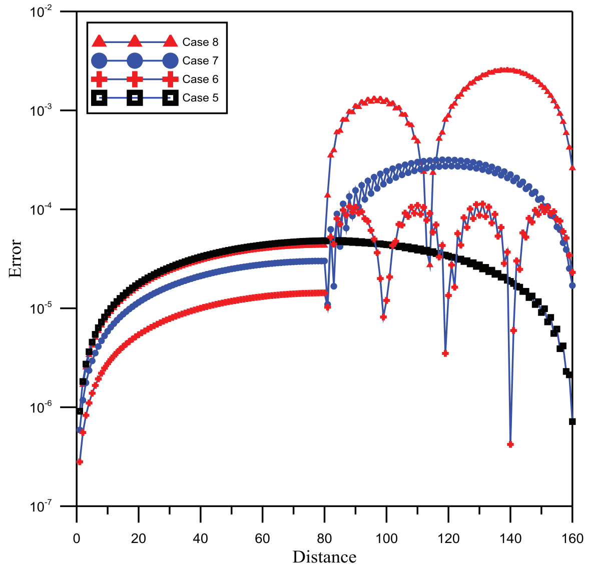

Figure 8: Error of case 5 to 8 using the DJIFM with the two-side equilibration algorithm.

and $\nu = 1.0$. In Cases 1 to 5, all the methods can be used to obtain the solution. Because the condition number increases as the case number increases (Fig. 5), it was determined that the CGM is not able to obtain the solution for Cases 6 to 8 because of the highly ill-conditioned system of LAEs. Figure 6 demonstrates that even using the two-sided equilibrium algorithm the CGM does not yield accurate results. By contrast, the DJIFM with the two-sided equilibrium algorithm can still be used to obtain a highly accurate solution for the most ill-conditioned case, as shown in Fig. 7. Figure 8 demonstrates the comparison of error with the exact solution. It was determined that by using the DJIFM with the two-sided equilibrium algorithm, an accuracy to the order of 10^{-3} to 10^{-6} can be attained, even in the case of layered soils with extreme hydraulic conductivity contrasts.

4.2 Layered problems with an ill-conditioned system of NAEs

The second example investigated was a groundwater flow equation. Regarding flow in unconfined systems bounded by a free surface, an approach pioneered by Dupuit (1863) and advanced by Forchheimer (1930) is often invoked. This nonlinear partial differential equation is often used when a two-dimensional unconfined flow field is reduced to a one-dimensional horizontal flow field by the invocation of the Dupuit-Forcheimer theory [Strack (1989)]. This equation can be written as

$$\frac{K}{2}\frac{d^2h^2}{dx^2} + N = 0 \tag{50}$$

where K is the hydraulic conductivity and N is the infiltration rate that can function as a position or a constant. In this example, we let $K = 2$ and $N = 0$. By introducing a finite difference discretization of h at the grid points, from Eq. (50) we can obtain

$$\mathbf{F} = \frac{1}{(\Delta \mathrm{x})^2}\left(\mathrm{h}_{\mathrm{i+1}}^2 - 2\mathrm{h}_{\mathrm{i}}^2 + \mathrm{h}_{\mathrm{i-1}}^2\right) + \mathrm{N}. \tag{51}$$

By introducing a finite difference discretization with $\Delta x = 1$ at 49 grid points with Dirichlet boundary conditions in which the groundwater heads on the left and right boundaries are $h(x = 0) = 8$ and $h(x = 48) = 2$, respectively, the exact solution as follows:

$$h(x) = \sqrt{-\frac{1}{2}Nx^2 + (-60/48 + 24N)x + 64}. \tag{52}$$

We first applied the DJIFM with the two-sided equilibrium algorithm to solve the homogenous one-dimensional horizontal flow problem presented in Eq. (50). The DJIFM parameters used in this example were $m = 0.01$, $h_t = 2.0$, and $\nu = 1.0$. Figure 9 demonstrates the computed groundwater head, using the DJIFM with the two-sided equilibrium algorithm. The error comparison is presented in Fig. 10. Good agreement was observed.

We then considered an anisotropic case in which the first layer of soil is sand with a fixed hydraulic conductivity of 1 cm/s, and the second layer of soil is clay with a fixed hydraulic conductivity of 10^{-6} cm/s. The condition number of this example was 2.53×10^8. Using the two-sided equilibrium algorithm, the condition number reduced to 1.73×10^3. According to the previous example, we learned that the DJIFM with the two-sided equilibrium algorithm can be used to solve this highly ill-conditioned system. Therefore, the DJIFM with the two-sided equilibrium algorithm was adopted in this example. The DJIFM parameters used in this example were $m = 0.01$, $h = 2.0$, and $\nu = 1.0$. In this example, an RMSN of 10^{-6} was set as the stopping criterion. Figure 11 demonstrates the computed groundwater head, using the DJIFM with the two-sided equilibrium algorithm. The RMSN versus the number of steps is shown in Fig. 12. It was determined that the proposed method reaches convergence with only 1,012 steps.

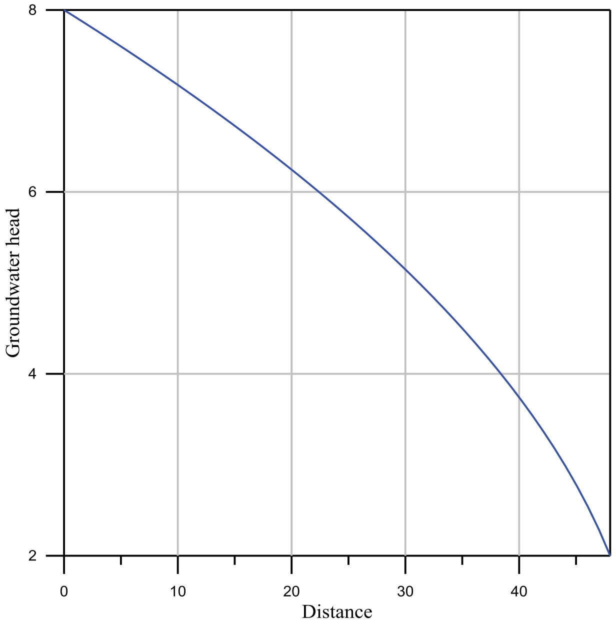

Figure 9: Computer groundwater heads using the DJIFM with the two-side equilibration algorithm.

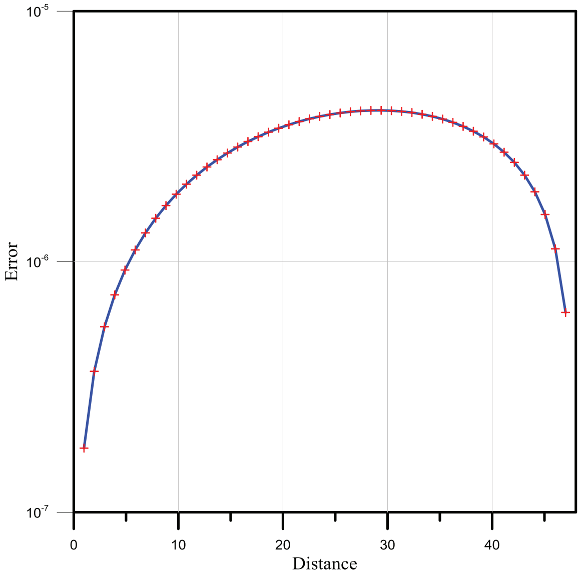

Figure 10: Comparison of the exact and numerical solutions.

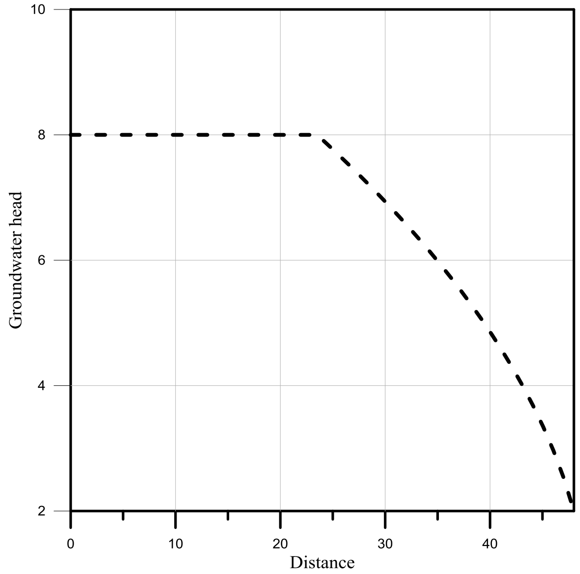

Figure 11: Computed result using the DJIFM with the two-side equilibration algorithm for a flow in unconfined systems with two different soil layers.

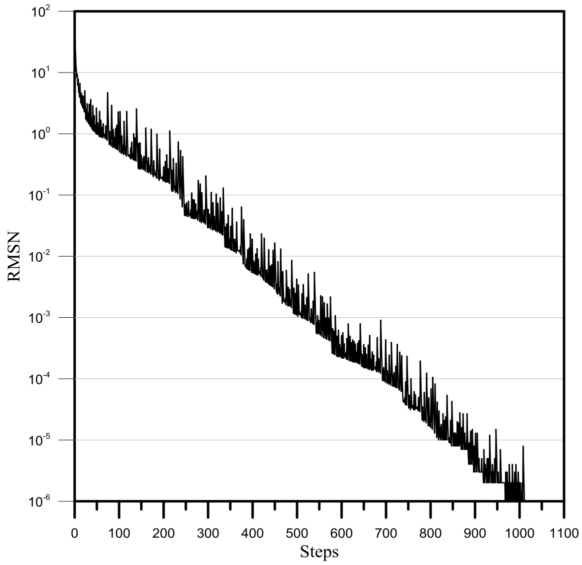

Figure 12: The root mean square norm versus the number of steps.

4.3 Two-dimensional layered problems with ill-conditioned system of LAEs

The last example to be investigated was a two-dimensional layered groundwater flow problem with an ill-conditioned system of LAEs. Because the example in 4.1 is only an one-dimensional example, we considered a two-dimensional partial differential equation of flow here.

$$\frac{\partial}{\partial x}k(x)\frac{\partial h}{\partial x}+\frac{\partial}{\partial y}k(y)\frac{\partial h}{\partial y}=0 \tag{53}$$

where $k(x)$ and $k(y)$ are hydraulic conductivities in x and y axes, respectively. The domain of this two-dimensional problem is $0 \leq x \leq L_x$ and $0 \leq y \leq L_y$ in x and y directions, respectively. Equation (53) does not hold on an interface between soils of different hydraulic conductivity because the hydraulic conductivity is not continuous on the interface. In this example, the coefficients of the hydraulic conductivity in x axis for two soil layers are $k(x) = ka$ within $0 \leq x < L_x/2$ and $k(x) = kb$ within $L_x/2 \leq x < L_x$. On the other hand, the coefficient of the hydraulic conductivity in y axis remains constant.

Using same procedure described in Example 4.1, we introduced a finite difference discretization using 51 by 51 grid points with $L_x = 10$ and $L_y = 10$. The Dirichlet boundary conditions were applied on the left and right side boundaries with the groundwater heads are $h(x=0) = 8$ and $h(x = L_x) = 2$, respectively. The no flow

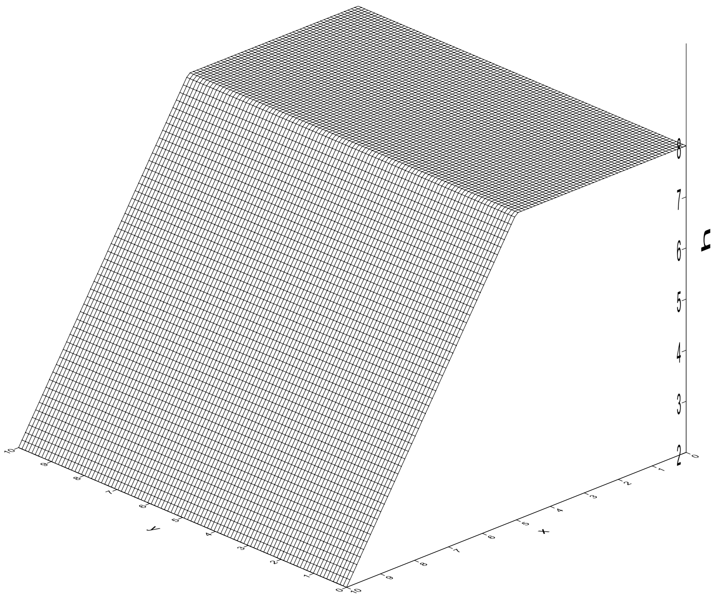

Figure 13: Computed results for a two-dimensional groundwater flow problem with two different soil layers.

Neumann boundary conditions were also applied on the top and bottom boundaries in which the fluxes of groundwater are $\partial h(y=0)/\partial y = 0$ and $\partial h(y=L_y)/\partial y = 0$, respectively. Figure 13 demonstrates the boundary conditions of this example. We considered the coefficients of the hydraulic conductivity in x axis for two soil layers . The first layer of soil is sand with a fixed hydraulic conductivity of $ka = 1$ cm/sec and the second layer of soil is very fine clay with a fixed hydraulic conductivity of $kb = 10^{-7}$ cm/sec. The hydraulic conductivity in y axis is $k(y) = 1$. In this example, an RMSN of 10^{-6} was set as the stopping criterion.

The finite difference method leads to a system of LAEs that can be written as $\mathbf{Ax} = \mathbf{b}$. In this example, we selected the fictitious time function $Q(t) = e^t$. Therefore, the DJIFM parameter used in this example was only $h_t = 1.0$. Because the above examples have demonstrated the advantages of the DJIFM with the two-sided equilibration algorithm for solving the layered problems with ill-conditioned system of LAEs, we solved this example only using the DJIFM with the two-sided equilibration algorithm. Figure 14 shows the computed groundwater head distribution for this two-dimensional problem. Figure 15 demonstrates the comparison of error on the profile $y = 5$ with the exact solution. It was determined that by using the DJIFM with the two-sided equilibrium algorithm, an accuracy to the order of

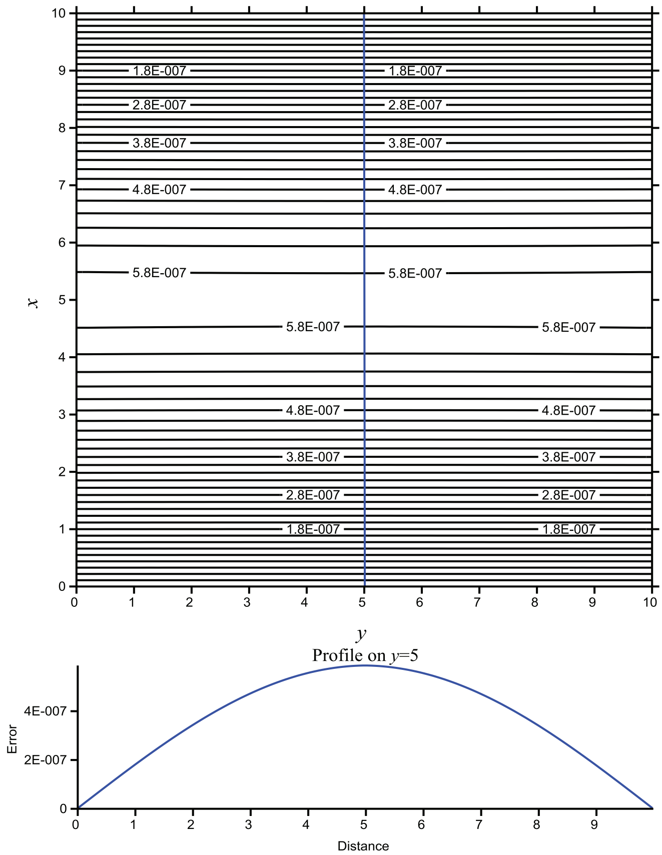

Figure 14: Comparison of the numerical and the exact solutions for a two-dimensional groundwater flow problem.

10^{-7} can be attained.

Similarly, the two-dimensional problem with three soil layers was conducted. The coefficients of the hydraulic conductivity in x axis for three soil layers are $k(x) = ka$ within $0 \leq x < L_x/3$, $k(x) = kb$ within $L_x/3 \leq x < 2L_x/3$, and $k(x) = kc$ within $2L_x/3 \leq x < L_x$. The coefficient of the hydraulic conductivity in y axis remains constant. All parameters used in this example were the same with the previous two soil layers example, except the coefficients of hydraulic conductivity in x axis are $ka = 1$ cm/sec, $kb = 10^{-7}$ cm/sec, and $kc = 10^{-2}$ cm/sec. Figure 16 shows the computed groundwater head distribution for this two-dimensional problem. This example presents that the proposed method can also be used to solve the multi-layered problems with extreme contracts in the physical property.

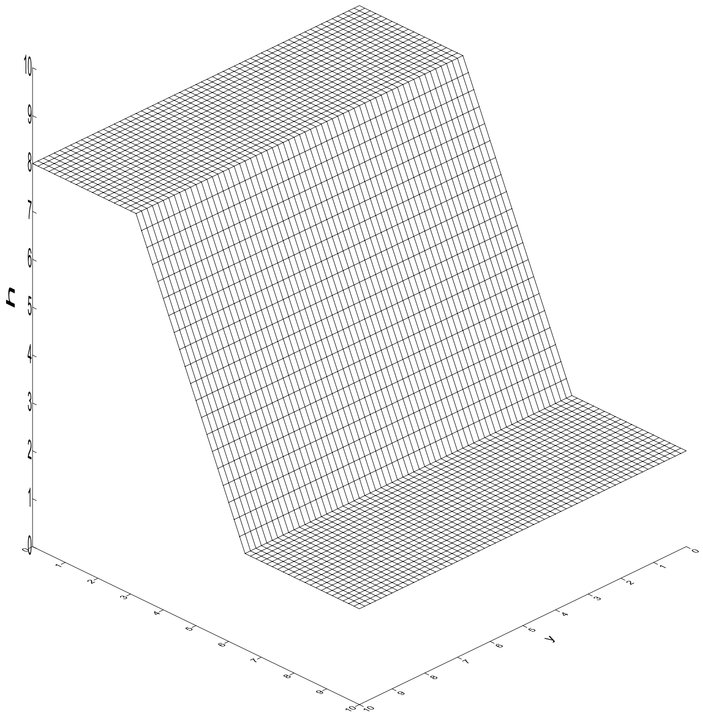

Figure 15: Computed results for a two-dimensional groundwater flow problem with three different soil layers.

5 Conclusion

This paper proposes a DJIFM with the incorporation of a two-sided equilibrium algorithm for solving ill-conditioned systems of linear equations with extreme contrasts in physical properties. The fundamental concepts and the construct of the proposed method are clearly addressed. The findings of this study are as follows:

1. Using the DJIFM with the incorporation of a two-sided equilibrium algorithm, the proposed method can effectively solve highly ill-conditioned systems of linear equations such as linear Hilbert matrix and linear Vandermonde matrix problems. By using the two-sided equilibrium algorithm, it is determined that the DJIFM presents better convergence characteristics than the CGM does for solving highly ill-conditioned systems of linear equations.

2. Layered problems with radical changes in material properties often present a great obstacle for most numerical methods to solve. In this study, the results

demonstrated that with the ease of numerical implementation, the DJIFM with the incorporation of a two-sided equilibrium algorithm can easily solve layered problems with extreme contrasts that are highly ill-posed and that have been difficult to solve in the past. The results of this study demonstrated that the proposed method can improve the convergence and increase the numerical stability for solving layered problems.

Acknowledgement: This study was partially supported by the National Science Council under project grant NSC100-2628-E-019-054-MY3 and NSC101-2218-E-019-001. The author would like to thank the National Science Council for the generous financial support.

References

Atluri, S. N. (2002): *Methods of Computer Modeling in Engineering and Sciences.* Tech. Science Press, 1400 pages.

Coomer, R. K.; Graham, I. G. (1996): Massively parallel methods for semiconductor device modelling. *Computing,* vol. 56, no. 1.

Das, Braja M. (2010): *Principles of Geotechnical Engineering,* 7th Edition, Cengage Learning, 2010, USA.

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.; Yeih, W.; Kuo, C.-L.; Atluri, S. N. (2009): A scalar homotopy method for solving an over/under-determined system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering & Sciences,* vol. 53, pp. 47-71.

Liu, C.-S.; Chang, Chih-Wen (2009): Novel Methods for Ill-Conditioned Linear Equations. *Journal of Marine Science and Technology,* vol. 17, no. 3, pp. 216-227.

Liu, C.-S.; Hong, Hong-Ki; Atluri Satya 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. (2012): Modifications of Steepest Descent Method and Conjugate Gradient Method Against Noise for Ill-posed Linear Systems . *Communications in Numerical Analysis,* vol. 2012, pp. 1-24.

Liu, C.-S. (2013a): An Optimal Multi-Vector Iterative Algorithm in a Krylov Subspace for Solving the Ill-Posed Linear Inverse Problems. *CMC: Computers, Materials & Continua*, vol. 33, no. 2, pp. 175-198.

Liu, C.-S. (2013b): A Two-Side Equilibration Method to Reduce the Condition Number of an Ill-Posed Linear System. *CMES: Computer Modeling in Engineering & Sciences,* vol. 91, pp. 17-42.

Liu, C.-S. (2014): A doubly optimized solution of linear equations system expressed in an affine Krylov subspace. *J. Comp. Appl. Math.,* vol. 260, pp. 375-394.
Ku, C.-Y.; Yeih, W.; Liu, C.-S.; Chi, C.-C. (2009): Applications of the fictitious time integration method using a new time-like function. *CMES: Computer Modeling in Engineering & Sciences,* vol. 43, no. 2, pp. 173-190.

Ku, C.-Y.; Yeih, W.; Liu, C.-S. (2010): Solving Non-Linear Algebraic Equations by a Scalar Newton-homotopy Continuation Method. *International Journal of Nonlinear Sciences and Numerical Simulation,* vol. 11, no. 6, pp. 435-450. **Ku, C.-Y.; Yeih, W.; Liu, C.-S.** (2011): Dynamical Newton-Like Methods for Solving Ill-Conditioned Systems of Nonlinear Equations with Applications to Boundary Value Problems. *CMES: Computer Modeling in Engineering & Sciences,* vol. 76, no. 2, pp. 83-108.

Strack, Otto D. (1989): Groundwater Mechanics, Prentice Hall Englewood Cliffs, New Jersey.

Vuik, C.; Segal, A.; Meijerinky, J. A. (1999): An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients. *Journal of Computational Physics,* vol. 152, pp. 385-403.

Vuik, C.; Segal, A.; Meijerinky, J. A.; Wilma, G. T. (2001): The Construction of Projection Vectors for a Deflated ICCG Method Applied to Problems with Extreme Contrasts in the Coefficients. **Journal of Computational Physics,** vol. 172, pp. 426-450.

Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian, P. (2007): *Numerical Recipes, The Art of Scientific Computing,* Cambridge University Press, Cambridge. UK, 2007.