 CMES, vol.76, no.2, pp.83-108, 2011

Dynamical Newton-Like Methods for Solving Ill-Conditioned Systems of Nonlinear Equations with Applications to Boundary Value Problems

Cheng-Yu Ku[1,2,3] Weichung Yeih[1,2] and Chein-Shan Liu[4]

Abstract: In this paper, a general dynamical method based on the construction of a scalar homotopy function to transform a vector function of Non-Linear Algebraic Equations (NAEs) into a time-dependent scalar function by introducing a fictitious time-like variable is proposed. With the introduction of a transformation matrix, the proposed general dynamical method can be transformed into several dynamical Newton-like methods including the Dynamical Newton Method (DNM), the Dynamical Jacobian-Inverse Free Method (DJIFM), and the Manifold-Based Exponentially Convergent Algorithm (MBECA). From the general dynamical method, we can also derive the conventional Newton method using a certain fictitious time-like function. The formulation presented in this paper demonstrates a variety of flexibility with the use of different transformation matrices to create other possible dynamical methods for solving NAEs. These three dynamical Newton-like methods are then adopted for the solution of ill-conditioned systems of nonlinear equations and applied to boundary value problems. Results reveal that taking advantages of the general dynamical method the proposed three dynamical Newton-like methods can improve the convergence and increase the numerical stability for solving NAEs, especially for the system of nonlinear problems involving ill-conditioned Jacobian or poor initial values which cause convergence problems.

Keywords: dynamical method, scalar homotopy function, fictitious time-like function, Newton's method, dynamical Jacobian-inverse free method, manifold-based exponentially convergent algorithm.

---

[1] Department of Harbor and River Engineering, National Taiwan Ocean University, Keelung.
[2] Computation and Simulation Center, National Taiwan Ocean University, Keelung.
[3] Corresponding Author, E-mail: chkst26@mail.ntou.edu.tw
[4] Department of Civil Engineering, National Taiwan University, Taipei

1 Introduction

Most physical systems are inherently nonlinear in nature. To deal with many practical nonlinear engineering problems, nonlinear problems are of interest to engineers, physicists and mathematicians. For solving nonlinear engineering problems, numerical methods including the finite element method, the boundary element method, the distinct element method, and the meshless method used in the computational mechanics [Atluri (2002)] usually need to solve a system of a non-linear algebraic equation system.

Over the past years, many contributions have been made towards the numerical solutions of Non-linear Algebraic Equations (NAEs). The iterative-based method, such as Newton's method, also known as the Newton–Raphson method, is perhaps the best known one for finding successively better approximations to the solutions of a real-valued non-linear system. Since it converges quadratically, Newton's method can often converge remarkably quickly if the initial guess is sufficiently close to the nonlinear solution. The conventional Newton-like algorithm is sensitive to the initial guess of solution, and it is very expensive in the computations of the Jacobian matrix and its inverse at each iterative step, especially for large scale nonlinear problems. Therefore, modifications of Newton's method, such as the arc-length methods or the Jacobian-Free Newton-Krylov method [Knoll and Keyes (2004); Lemieux et al. (2010)] have been extensively developed for this purpose.

Most of the methods for solving NAEs are based on the iterative scheme. Hirsch and Smale (1979) have derived a continuous Newton method governed by an Ordinary Differential Equation (ODE). Using the concept of the fictitious time, the procedure for solving NAEs is equivalent to solve an ODE. Recently, Liu and Atluri (2008) proposed a time integration method named the Fictitious Time Integration Method (FTIM). The FTIM was first used to solve a non-linear system of algebraic equations by introducing a fictitious time, such that it is a mathematically equivalent system in the augmented $n+1$-dimensional space as the original algebraic equation system is in the original n-dimensional space. The stationary point of these evolution equations is the solution for the original algebraic equations. Based on the dynamical algorithm, Liu (2008, 2009) introduced the use of the FTIM to solve two-dimensional quasi-linear elliptic boundary value problems and many other nonlinear problems. Ku, Yeih, Liu and Chi (2009) introduced the FTIM with a new time-like function to increase the speed of convergence.

In addition to the FTIM, the homotopy method [Ku, Yeih and Liu (2010)] can also be used to solve the NAEs using a similar fictitious time concept. To enhance the local convergence to a global convergence, the homotopy method represents a way to find a solution by constructing a new problem, simpler than the original one, and then gradually deforming this simpler problem into the original one by keeping track of the series of zeros that connect the solutions of the simpler problem to those of the original one, which is a harder one. The early practical application of homotopy-like methods to numerical solution of nonlinear equations is commonly attributed to Davidenko (1953). Recently, more references can be found in the application of homotopy methods. Liao (2004) employed the basic ideas of homotopy to propose a general method for nonlinear problems, namely the homotopy analysis method, and this method has been successfully applied to solve many types of nonlinear problems. He (2005) studied the homotopy method through a series of different non-linear ordinary differential equations. In many vector-based homotopy methods, each step involves computing the inverse of the Jacobian matrix which often raises the difficulty of divergence in certain circumstances; in such cases each step is as costly as a Newton step. Since a scalar-based homotopy method does not need to calculate the inverse of the Jacobian matrix and has a great numerical stability, it may be a better alternative for solving a system of NAEs.

In this paper, we introduce a general dynamical method which is based on the construction of a scalar homotopy function to transform a vector function of NAEs into a time-dependent scalar function by introducing a fictitious time-like variable. Three different dynamical Newton-like methods including the Dynamical Newton Method (DNM), the Dynamical Jacobian-Inverse Free Method (DJIFM) and the Manifold-Based Exponentially Convergent Algorithm (MBECA) are derived using different transformation matrices. The formulation presented in this paper demonstrates a variety of flexibility with the use of different transformation matrices to create other possible dynamical methods for solving NAEs. These dynamical Newton-like methods are then adopted for the solution of ill-conditioned systems of nonlinear equations and applied to boundary value problems. First of all, the general dynamical method is described as follows.

2 The General Dynamical Method

We consider the following NAEs:

$$F_i(x_1,\dots,x_n) = 0,\; i = 1,\dots,n. \tag{1}$$

Using $\mathrm{x} := (x_1,\dots,x_n)^T$ and $\mathrm{F} := (F_1,\dots,F_n)^T$, Eq. (1) can be written as $\mathrm{F}(\mathrm{x}) = 0$. Solving Eq. (1) by a first-order Taylor approximation, we can easily see that Newton's method for solving $\mathrm{F}(\mathrm{x}) = 0$ is given by

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

where B is an $n \times n$ Jacobian matrix with its ij-th component being given by $\partial F_i / \partial x_j$. Newton's method is only guaranteed to be a local convergence, if certain conditions are satisfied, and hence, depending on the type of the function and the initial guess of the solution, it may or may not converge. In addition, it is expensive in the computations of the Jacobian matrix and its inverse at each iterative step.

On the other hand, for solving the NAEs,

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

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

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

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

and the Newton homotopy function is

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

where x_0 is the given initial values and $\tau \in [0, 1]$. To conduct a scalar-based homotopy continuation method, we first convert the vector equation of $\mathrm{F} = 0$ to a scalar equation by noticing that

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

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

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

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

The scalar homotopy method retains the merits of the homotopy method, such as the global convergence, but it does not involve the complicated computation of the inverse of the Jacobian. The scalar homotopy method, however, needs a very small time step to reach the fictitious time, $\tau = 1$, which results in a slow convergence, in comparison with other methods. In this study, we propose a scalar homotopy algorithm based on the Newton homotopy function as described in Eq. (5), which can also be written as follows:

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

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

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

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

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

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

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

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

It is clear that the tracking of a solution path for the proposed scalar Newton homotopy function, as the homotopy parameter τ is gradually varied from 0 to 1, is 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, multiplying $Q(t)$ at both sides of Eq. (10) we have

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

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 make an analogy for the scalar homotopy method to the theory of plasticity. In their explanation, the above 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, $h(\mathrm{x},t)$, with respect to x and t can be written as

$$\frac{\partial h}{\partial t}=\frac{1}{2}\dot{Q}(t)\left\|\mathrm{F}(\mathrm{x})\right\|^{2}$$

and

$$\frac{\partial h}{\partial \mathrm{x}}=Q(t)\mathrm{B}^{\mathrm{T}}\mathrm{F}(\mathrm{x}), \tag{15}$$

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

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

Inserting Eqs. (15) and (16) into Eq. (14), we can derive

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

In Eq. (16), T is the transformation matrix which can be B^{-1}, the identity matrix, I, B^{T}, or any other square matrices. With the introduction of different transformation matrices, such as B^{-1}, I, or B^{T}, the proposed general dynamical method can be transformed into the DNM, the DJIFM and the MBECA, respectively.

Inserting Eq. (17) into Eq. (16), we have

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

The above equation is the general dynamical equation for solving non-linear algebraic equations. It is also found that in Eq. (18), we solve NAEs by introducing a fictitious time function, such that it is a mathematically equivalent system in the augmented $n+1$-dimensional space as the original algebraic equation system is in the original n-dimensional space. The fixed point of these evolution equations, which is the root for the original algebraic equation, is obtained by applying numerical integrations on the resultant ordinary differential equations.

3 The Fictitious Time-like Function

There are many ways to choose a suitable function of $Q(t)$. Based on the FTIM first proposed by Liu and Atluri (2008), the NAEs, $\mathrm{F}(\mathrm{x}) = 0$, can be embedded in a system of nonlinear ODEs: $\dot{x} = -\nu/q(\tau)\mathrm{F}(\mathrm{x})$ where τ is the fictitious time, $q(\tau)$ is a monotonically increasing function of τ. In their study, a simple time-like function of $q(\tau) = (1+\tau)$ was chosen. In addition to this original simple time-like function, Ku, Yeih, Liu, and Chi (2009) proposed a more general function such as $q(\tau) = (1+\tau)^m$. Based on a similar idea and replacing τ as t, we can let

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

Hence, we have

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

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

$$\dot{x} = \frac{-\nu}{2(1+t)^m}\frac{\|\mathrm{F}(\mathrm{x})\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x})\mathrm{B}\mathrm{T}\mathrm{F}(\mathrm{x})}\mathrm{T}\mathrm{F}(\mathrm{x}) \tag{21}$$

where m is a control parameter for speeding the convergence as discussed in Ku, Yeih, Liu and Chi (2009) and ν is a damping parameter introducing by Liu and Atluri (2008) for improving the convergence.

To satisfy the conditions that $Q(t) > 0$, $Q(0) = 1$, and $Q(t)$ is a monotonically increasing function of t, and $Q(\infty) = \infty$, another suitable function of $Q(t)$ can be easily found and written as

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

Inserting Eq. (22) into Eq. (19), we have

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

Again, inserting Eq. (23) into Eq. (18), we have

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

We can easily find that Eqs. (21) and (24) embed the fictitious time function in the evolution of the solution search. To deal with Eq. (21) and Eq. (24), we may employ a forward Euler scheme and obtain the following equations:

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

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

where h_t is the fictitious time step. In the above equations, it is found that the numerator and denominator of the fraction in Eqs. (25) and (26) are scalars if we adopt any one of the transformation matrices from B^{-1}, I, and B^{T}.

4 The Dynamical Newton Method (DNM)

To derive the DNM, we let the transformation matrix, T, be B^{-1}, and Eq. (18) can be written as

$$\dot{x} = -\frac{\dot{Q}(t)}{2Q(t)}\mathrm{B}^{-1}\mathrm{F}(\mathrm{x}). \tag{27}$$

Eq. (27) is similar to famous Newton's method. If we choose the fictitious time function as demonstrated in Eq. (20), we derive the DNM as

$$\dot{x} = \frac{-v}{2(1+t)^m}\mathrm{B}^{-1}\mathrm{F}(\mathrm{x}). \tag{28a}$$

Using the forward Euler scheme, we have

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \frac{h_t v}{2(1+t)^m}[\mathrm{B}(\mathrm{x}^k)]^{-1}\mathrm{F}(\mathrm{x}^k). \tag{28b}$$

If the fictitious time function $Q(t) = e^t$ is adopted, we derive the DNM as

$$\dot{x} = -\frac{1}{2}\mathrm{B}^{-1}\mathrm{F}(\mathrm{x}). \tag{29}$$

Using the forward Euler scheme and adopting $h_t = 2$, we have

$$\mathrm{x}^{k+1} = \mathrm{x}^k - [\mathrm{B}(\mathrm{x}^k)]^{-1}\mathrm{F}(\mathrm{x}^k). \tag{30}$$

Eq. (30) is identical to Newton's method. Newton's method is a simple iterative numerical method to approximate roots of equations. However, the DNM proposed in this study is more flexible than original Newton's method. It can be integrated with different fictitious time functions and can improve the convergence with the use of proper parameters in the fictitious time function.

5 The Dynamical Jacobian-Inverse Free Method (DJIFM)

To derive the DJIFM, we let the transformation matrix, T, be the identity matrix, I. Eq. (18) can be written as

$$\dot{x} = -\frac{\dot{Q}(t)}{2Q(t)}\frac{\|\mathrm{F}(\mathrm{x})\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x})\mathrm{B}\mathrm{F}(\mathrm{x})}\mathrm{F}(\mathrm{x}). \tag{31}$$

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

$$\dot{x} = -\frac{v}{2(1+t)^m}\frac{\|\mathrm{F}(\mathrm{x})\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x})\mathrm{B}\mathrm{F}(\mathrm{x})}\mathrm{F}(\mathrm{x}). \tag{32}$$

When Eq. (32) is similar to the FTIM, it is, however, without the fractional item in the original FTIM as shown in Eq. (33):

$$\dot{x} = -\frac{v}{(1+t)^m}\mathrm{F}(\mathrm{x}). \tag{33}$$

Using the forward Euler scheme, we have

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \frac{h_t v}{2(1+t)^m}\frac{\left\|\mathrm{F}(\mathrm{x}^k)\right\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x}^k)\mathrm{B}(\mathrm{x}^k)\mathrm{F}(\mathrm{x}^k)}\mathrm{F}(\mathrm{x}^k). \tag{34}$$

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

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \frac{h_t}{2}\frac{\left\|\mathrm{F}(\mathrm{x}^k)\right\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x}^k)\mathrm{B}(\mathrm{x}^k)\mathrm{F}(\mathrm{x}^k)}\mathrm{F}(\mathrm{x}^k). \tag{35}$$

In the above Eqs. (34) and (35), it is found that the numerator and denominator of the fraction in above are only scalars. Accordingly, we can avoid computing the inverse of the Jacobian matrix, and thus can improve the numerical stability. In addition, comparing Eq. (32) with Eq. (33), we can find that only the fraction item is added in Eq. (32). This fraction item can actually increase the convergence and is discussed in the later section.

6 The Manifold-Based Exponentially Convergent Algorithm (MBECA)

To derive the MBECA, we let the transformation matrix, T, be B^{T}. Eq. (18) can be written as

$$\dot{x} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathrm{F}(\mathrm{x})\|^2}{\mathrm{F}^{\mathrm{T}}(\mathrm{x})\mathrm{B}\mathrm{B}^{\mathrm{T}}\mathrm{F}(\mathrm{x})} \mathrm{B}^{\mathrm{T}}\mathrm{F}(\mathrm{x}). \tag{36}$$

The above equation can be expressed as

$$\dot{x} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathrm{F}(\mathrm{x})\|^2}{\left\|\mathrm{B}^{\mathrm{T}}\mathrm{F}(\mathrm{x})\right\|^2} \mathrm{B}^{\mathrm{T}}\mathrm{F}(\mathrm{x}). \tag{37}$$

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

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

Eq. (38) is the Manifold-Based Exponentially Convergent Algorithm (MBECA) which was original proposed by Ku, Yeih and Liu (2010). Using the forward Euler scheme, we have

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \frac{h_t v}{2(1+t)^m} \frac{\left\|\mathrm{F}(\mathrm{x}^k)\right\|^2}{\|[\mathrm{B}(\mathrm{x}^k)]^{\mathrm{T}}\mathrm{F}(\mathrm{x}^k)\|^2} [\mathrm{B}(\mathrm{x}^k)]^{\mathrm{T}}\mathrm{F}(\mathrm{x}^k). \tag{39}$$

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

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \frac{h_t}{2} \frac{\left\|\mathrm{F}(\mathrm{x}^k)\right\|^2}{\|[\mathrm{B}(\mathrm{x}^k)]^{\mathrm{T}}\mathrm{F}(\mathrm{x}^k)\|^2} [\mathrm{B}(\mathrm{x}^k)]^{\mathrm{T}}\mathrm{F}(\mathrm{x}^k). \tag{40}$$

In the above equations, it is found that the numerator and denominator of the fraction in Eqs. (39) and (40) are only scalars. Accordingly, we can avoid computing the inverse of the Jacobian matrix, and thus can improve the numerical stability.

7 Numerical illustrations

7.1 Example 1

In order to clarify the characteristics of the general dynamical methods, we first consider a simple scalar equation as

$$F(x) = x^2 - 1 = 0. \tag{41}$$

It is easy to know that the roots of Eq. (41) are 1 and -1. In this example, we compute the root of 1 for the above equation using the conventional Newton method, the DNM, the DJIFM and the MBECA as derived previously.

Figure 1 shows the speed of convergence to the exact solution of $x = 1$ for the conventional Newton method, the DNM, the DJIFM and the MBECA. The parameters used for each method are listed in Table 1. In this study, we use the Root Mean Square Norm (RMSN) as the stopping criterion for the convergence. Results obtained reveal that with proper use of the fictitious time-like function, all the DNM, the DJIFM and the MBECA can converge faster than the Newton method and reach the RMSN of 10^{-8} within 47 time steps.

Table 1: The parameters adopted for the analysis

	Method	m	h	ν	Initial guess
1	Newton's method	Not required			$x^0 = 1e-15$
2	DNM	0.01	1	2.5	$x^0 = 1e-15$
3	DJIFM	0.01	1	2.5	$x^0 = 1e-15$
4	MBECA	0.01	1	2.5	$x^0 = 1e-15$

7.2 Example 2

In the second example, we study the following system of two algebraic equations:

$$\begin{aligned} F_1(u,v) &= u^2 + v = 0, \\ F_2(u,v) &= -v^2 + 16 = 0. \end{aligned} \tag{42}$$

In this test, we compare the numerical stability of the general dynamical methods. The parameters used in this example for the DNM, the DJIFM and the MBECA are $m = 0.01$, $h = 1.0$, and $\nu = 2.5$. We start from an initial value of $(u,v) = (10^{-8},\ 0)$. Results obtained show that the conventional Newton method and the DNM diverge because the initial value causes an ill-conditioned Jacobian matrix. Since the DJIFM and the MBECA need not compute the inverse of the Jacobian matrix, both methods can obtain the solution within 100 fictitious time steps and reach the RMSN of only 10^{-8}. Figure 2 illustrates the comparison of the convergence for the DJIFM and the MBECA. It is interested that with the same parameters the DJIFM converges faster than the MBECA.

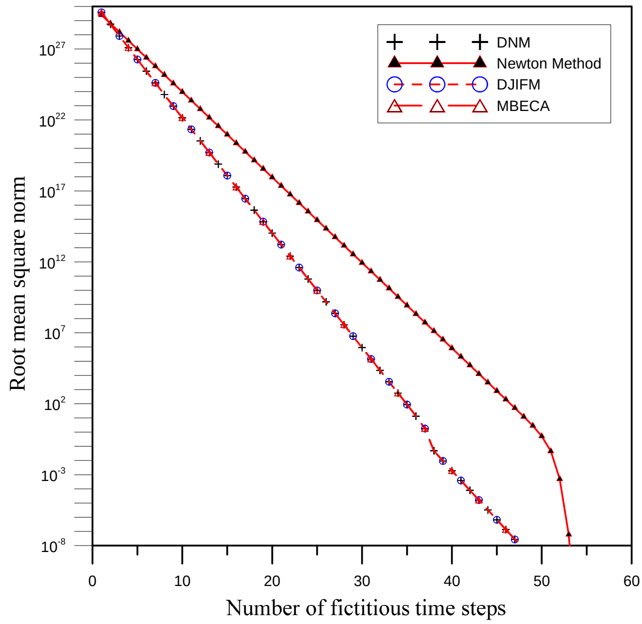

Figure 1: Convergence to the exact solution of $x = 1$ for Newton's method, the DNM, the DJIFM, and the MBECA.

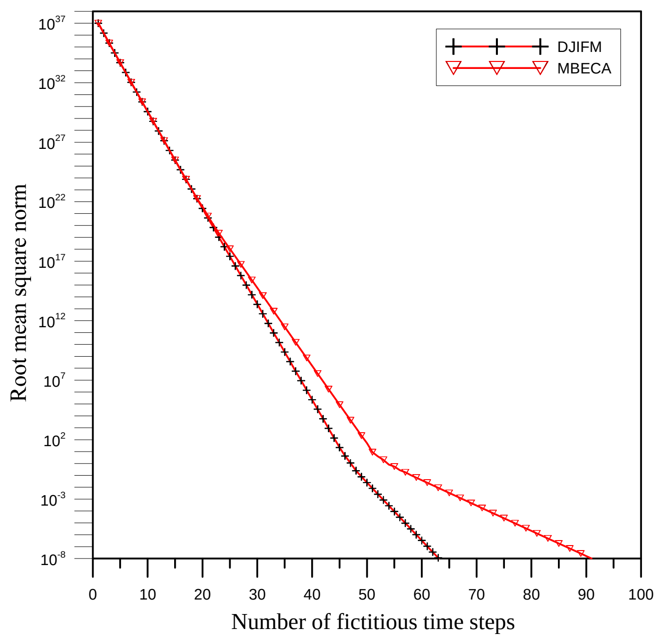

Figure 2: Comparison of the convergence for the DJIFM and the MBECA.

7.3 Example 3

In the third example, we study the following system of two NAEs:

$$\begin{aligned} F_1(x_1,x_2) &= x_1^2+x_2^2-2=0, \\ F_2(x_1,x_2) &= e^{(x_1-1)}+x_2^2-2=0, \end{aligned} \tag{43}$$

where the Jacobian is

$$\mathrm{B}=\begin{bmatrix} 2x_1 & 2x_2 \\ e^{(x_1-1)} & 2x_2 \end{bmatrix}. \tag{44}$$

This is an interesting example because the iteration for Newton's method stagnates with an initial value of $(x_1,x_2)=(3,\ 5)$, as illustrated by Kelly (2003). The solution search fails because the derivative of the target function, B, is nearly singular. In this test, we investigate this example again using the conventional Newton method, the DJIFM and the MBECA with the parameters of $m=0.01$, $h=1.0$, and $\nu=2.5$ and starting from the same initial value, $(x_1,x_2)=(3,\ 5)$. Figure 3 illustrates the comparison of the convergence for different methods. In our test, it is found that the Newton method is not able to converge. From the result, it is also found that the DJIFM converges much faster than the MBECA and both methods can reach to the solution with the RMSN in the order of 10^{-8}. Figure 4 shows that the DJIFM and the MBECA have different solution paths and converge to different solutions of $(x_1,x_2)=(-0.4777,-1.3311)$ and $(x_1,x_2)=(1.0,1.0)$, respectively.

It is interesting to note that the DNM is very similar to the conventional Newton method. However, the DNM can find the solution if proper dynamical parameters are adopted in the solution procedure. To demonstrate the flexibility of the DNM, we re-solve this example using the DNM and use the same initial value, $(x_1,x_2)=(3,\ 5)$. The solution procedure is that we use the parameters of $m=0.01$, $h=1.2$, and $\nu=-1.5$ at the first ten time steps and then change the parameter of $\nu=1.8$. Figure 5 illustrates the comparison of the convergence for the DNM and Newton's method. The solution path, as shown in Fig. 6, demonstrates that the DNM could have a different solution path comparing with the conventional Newton method. The solution, therefore, could be found to be $(x_1,x_2)=(-0.4777,-1.3311)$. On the other hand, we can find that the solution search fails at $x_1=3.51286$ using the Newton method. This result is similar to that obtained by Kelly (2003). This example reveals that the DNM has the advantages to obtain the solution which the solution search fails in the conventional Newton method.

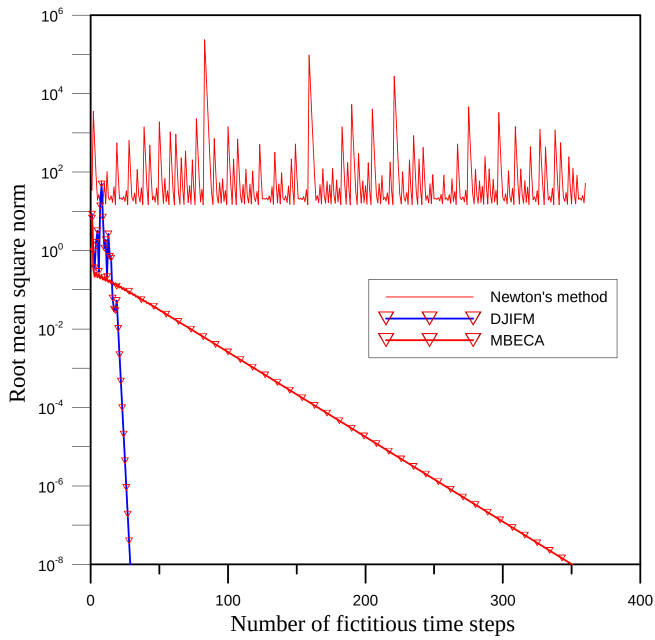

Figure 3: Comparison of the convergence for Newton's method, the DJIFM and the MBECA.

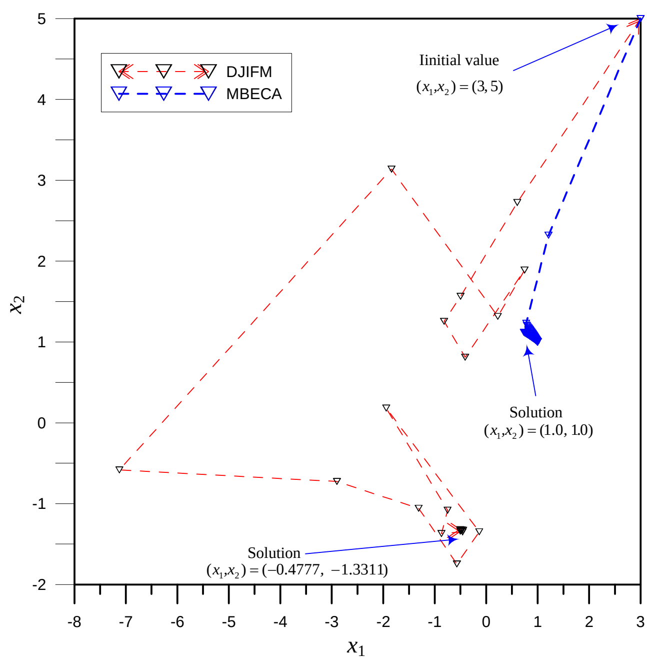

Figure 4: Solution paths of the DJIFM and the MBECA for solving Example 3.

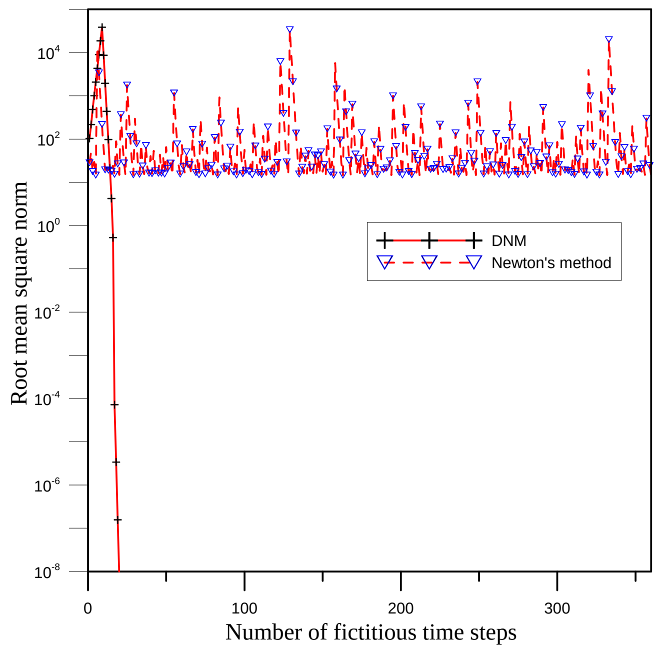

Figure 5: Comparison of the convergence for Newton's method, the DJIFM and the MBECA.

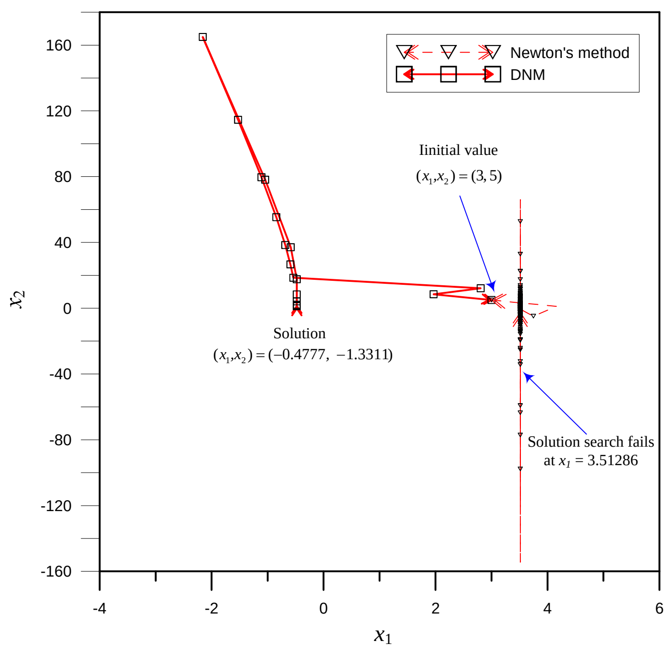

Figure 6: Solution path of the DNM for solving Example 3.

7.4 Example 4

This example is a system of three NAEs in three-variables as follows:

$$\begin{aligned}
&F_1(x,y,z) = x+y+z-3 = 0,\\
&F_2(x,y,z) = xy+2y^2+4z^2-7 = 0,\\
&F_3(x,y,z) = x^8+y^4+z^9-3 = 0.
\end{aligned} \tag{45}$$

In this example, the conventional Newton method, the DJIFM with the parameters of $m = 0.01$, $h = 1$, $\nu = 0.8$, and the MBECA with the parameters of $m = 0.01$, $h = 1$, $\nu = 2$ are adopted. We start from an initial value of $(x,\ y,\ z) = (0.0,\ 0.5,\ 0.6)$, the DJIFM and the MBECA can both converge to the solution $(x,\ y,\ z) = (1,1,1)$ and reach a residual in the order of 10^{-8}. However, with the same initial value of $(x,\ y,\ z) = (0.0,\ 0.5,\ 0.6)$, the Newton method is not able to converge to the solution. It is also found that the DJIFM converges to the solution much faster than the MBECA as shown in Fig. 7.

We re-investigate this example using the conventional Newton method, the DNM with the parameters of $m = 0.01$, $h = 1$, $\nu = 0.5$ and the MBECA with the parameters of $m = 0.01$, $h = 1$, $\nu = 1.88$. Starting from the initial value of $(x,\ y,\ z) = (0.01,\ 0.5,\ 0.6)$, it is found that the DNM and MBECA can converge and obtain the solution $(x,\ y,\ z) = (0.9892, 1.1360, 0.9077)$. Figure 8 demonstrates that the DNM converge a little faster than the MBECA to the solution and both methods reach the RMSN in the order of 10^{-8}.

7.5 Example 5

This example under investigation is a system of two NAEs in three-variables as follows:

$$\begin{aligned}
&F_1(x,y,z) = x^2+y^2+z^2-1 = 0,\\
&F_2(x,y,z) = x^2/4+y^2/4+z^2-1 = 0.
\end{aligned} \tag{46}$$

Since Eq. (46) has fewer equations than variables, it is also known as an underdetermined non-linear system. This underdetermined non-linear system has the number of unknowns and the number of equations unequal which often raises the difficulty of finding solutions by using the Newton method. As demonstrated in Eq. (28), Eq. (30) and Eq. (34), the DNM, the Newton method, and the DJIFM need the number of unknowns and the number of equations being equal in order to make the matrix multiplication. Accordingly, they are not capable to solve the problem of underdetermined non-linear system. However, from Eq. (38) the MBECA does not have the limitation as mentioned above. So, we start from an initial value of $(x,\ y,\ z) = (5, 10, 20)$, as expected only the MBECA can converge to the solution of $(x, y, z) = (0.000676, 0.000135, 0.999999)$ and reaches the RMSN in the order of 10^{-8}.

7.6 Example 6

Previous examples have demonstrated that the general dynamical method can be used to solve non-linear algebraic equations as well as the underdetermined non-linear system. This example is to illustrate that the proposed general dynamical method can also be used to solve ordinary differential equations. In this example we apply the general dynamical method to solve the following boundary value problem:

$$u'' = 3/2u^2, \tag{47}$$

The boundary conditions are $u(0) = 4,\ u(1) = 1$. Equation (47) has an exact solution as follows:

$$u(x) = \frac{4}{(1+x)^2}. \tag{48}$$

By introducing a finite difference discritization of u at the grid points, we can obtain

$$F_i = \frac{1}{\Delta x^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2, \tag{49}$$

with the boundary conditions of

$$u_0 = 4,\ u_{n+1} = 1, \tag{50}$$

where $\Delta x = \frac{1}{(n+1)}$.

Liu and Atluri (2011) has solved the boundary value problem using a residual norm based algorithm and FTIM. In this example, we first adopt the Newton method and the DJIFM and let the initial value of $u = -2/(3\Delta x^2)$. The parameters of $m = 0.01$, $h = 1.0$, $\nu = 1.5$ are used for the DJIFM. The number of grid point is $n = 9$. It is interesting that the Newton method can not converge to the solution due to the ill-conditioned Jacobian matrix occurred from the poor initial values. The DJIFM, however, can find the solution and takes about 200 time steps to converge to the solution with the RMSN in the order of 10^{-8}, as shown in Fig. 9. Figs. 10 and 11 demonstrate the comparison of the computed results with the exact solution and the numerical error, respectively.

In addition, we investigate the DJIFM and the FTIM to solve this boundary value problem with the parameters of $m = 0.01$, $h = 1.0$, $\nu = 1.5$ and $m = 0.01$, $h = 0.01$,

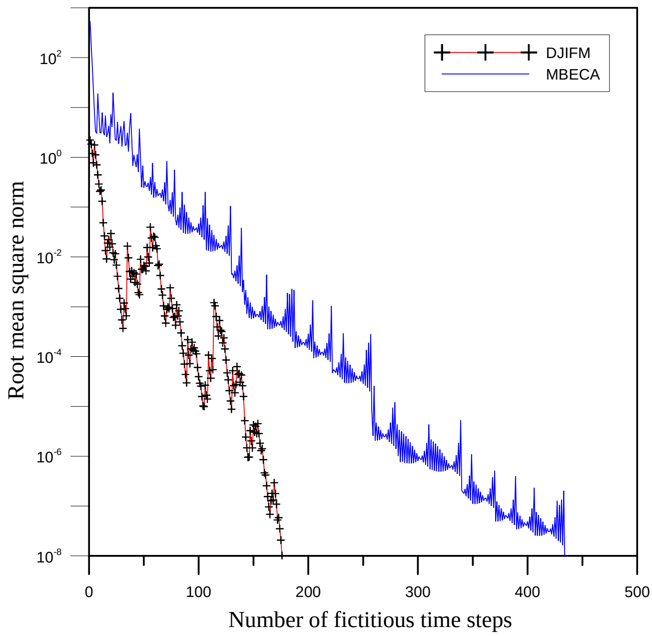

Figure 7: Comparison of the convergence for the DJIFM and the MBECA.

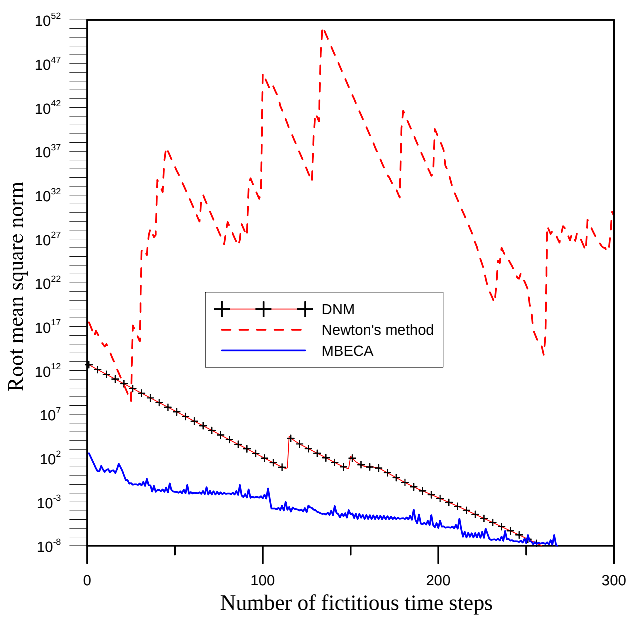

Figure 8: Comparison of the convergence for the DNM, Newton's method and the MBECA.

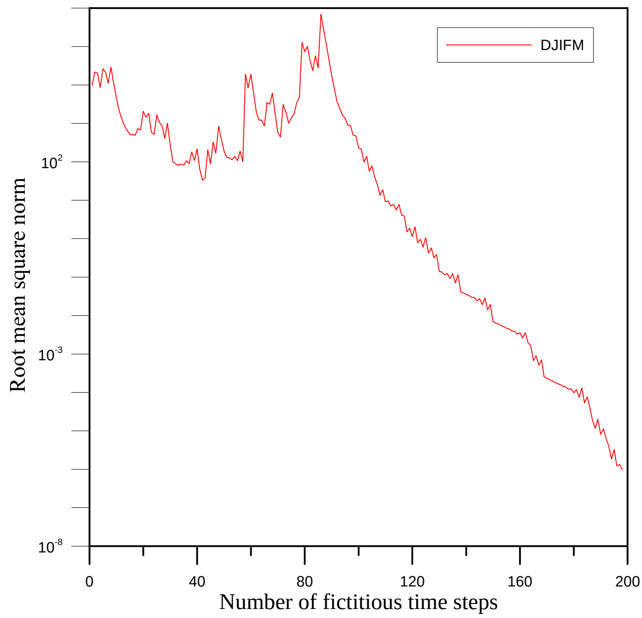

Figure 9: Root mean square norm of the DJIFM for solving Example 6.

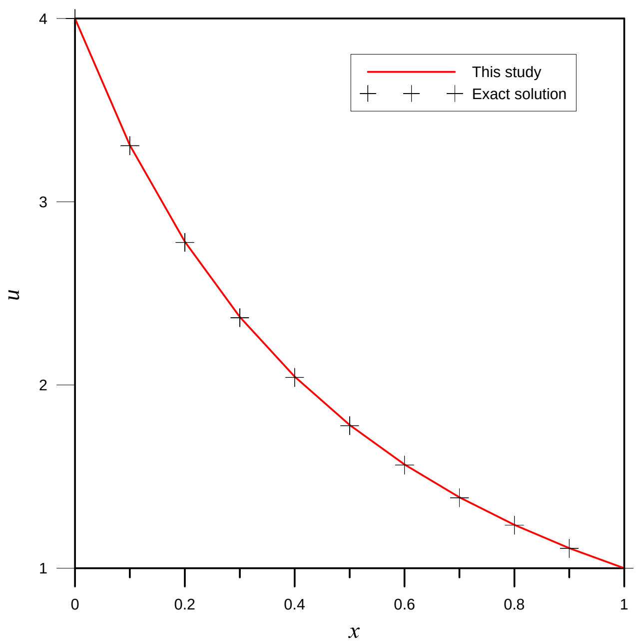

Figure 10: Comparison of results with the exact solution for Example 6.

$\nu = 0.0001$, respectively. To remain convergence, the small parameters have to use in the FTIM. The initial value of $u = -2/(3\Delta x^2)$ and grid point $n = 9$ are also the same in this example. Fig. 12 shows that both methods may converge to the solution. The DJIFM takes only 100 time steps to converge to the solution with the RMSN in the order of 10^{-6}. On the contrary, the FTIM can not converge within 5000 time steps. As a result, the DJIFM demonstrates a better performance than the FTIM in dealing with these ill-conditioned systems.

7.7 Example 7

The last example to be investigated is a groundwater flow equation. For 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 if a two-dimensional unconfined flow field is reduced to an one-dimensional horizontal flow field by the invocation of the Dupuit-Forcheimer theory [Strack (1989)]. This equation can be written as

$$\frac{K}{2}\frac{d^2h^2}{dx^2} + N = 0 \tag{51}$$

where K is the hydraulic conductivity and N is the infiltration rate which can be a function of position or a constant. In this example, we let $K = 2$ and $N = 0$. By introducing a finite difference discritization of h at the grid points, we can obtain

$$F_i = \frac{1}{\Delta x^2}(h_{i+1}^2 - 2h_i^2 + h_{i-1}^2) + N, \tag{52}$$

with the boundary conditions of $h_0 = 8$, $h_{n+1} = 2$ and $\Delta x = 1$. The number of grid point is $n = 50$. In this example we apply the conventional Newton method, the DJIFM and the MBECA to solve this groundwater flow equation using the same parameters of $m = 0.01$, $h = 1.0$, $\nu = 1.85$ and start from initial groundwater heads as shown in Fig. 13. The initial groundwater heads are set to zero at each odd number of grids of total 50 grid points. To examine the numerical stability, the initial groundwater heads at even number of total 50 grid points with noise are generated using random values from the normal distribution with the mean value of 10^{-8} and standard deviation of 10^{-8}.

Due to the ill-conditioned Jacobian matrix occurred from the poor initial values, the conventional Newton method can not converge to the solution. However, the DJIFM and MBECA can both converge to the solution with the RMSN in the order of 10^{-8} as shown in Fig. 14. From Fig. 14, we can also find that the DJIFM has a better performance than the MBECA for solving this example. Figs. 15 and 16 are the computed results and the error. Comparing with the exact solutions, good agreements are found.

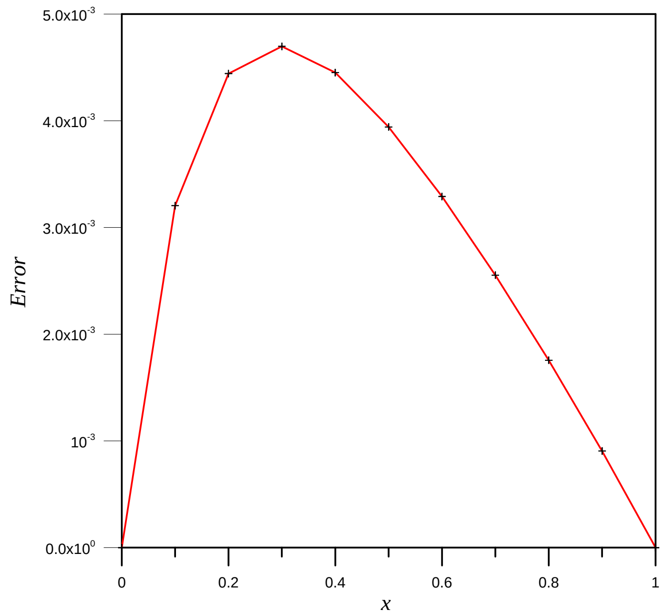

Figure 11: Numerical error of Example 6.

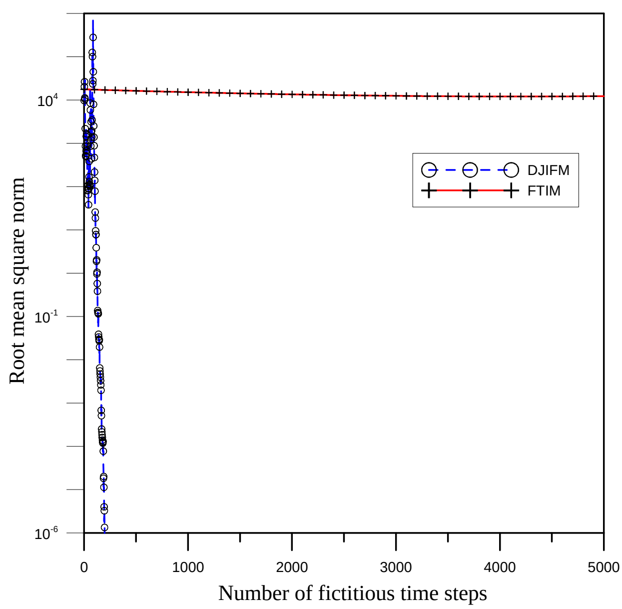

Figure 12: Comparison of the convergence for the DJIFM and the FTIM.

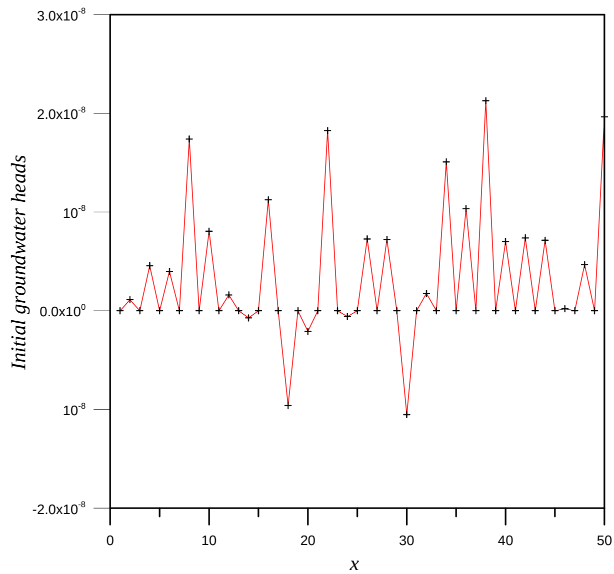

Figure 13: The initial groundwater heads for Example 7.

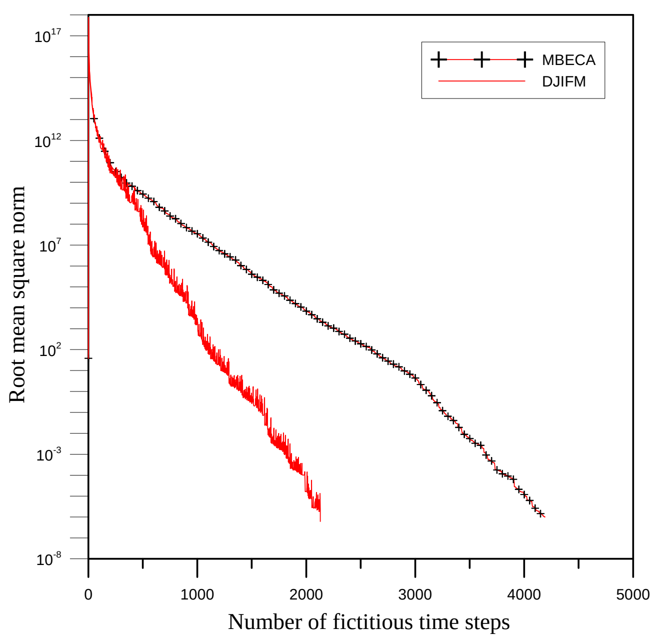

Figure 14: Comparison of the convergence for the DJIFM and the MBECA.

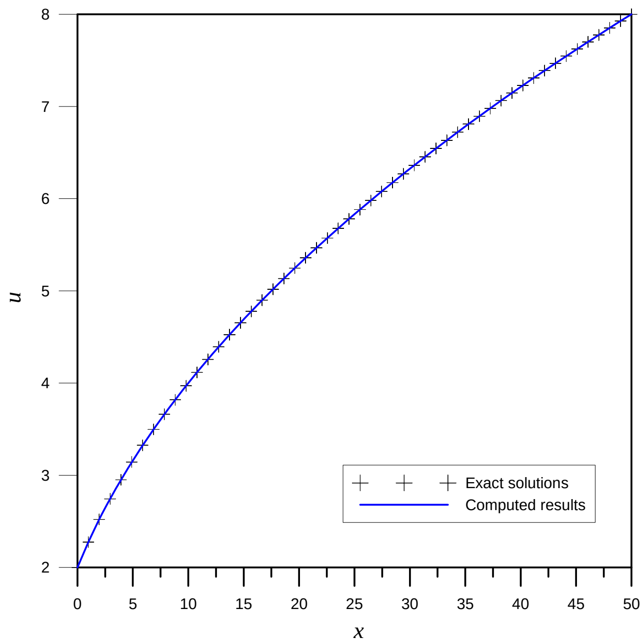

Figure 15: Comparison of results with the exact solution for Example 7.

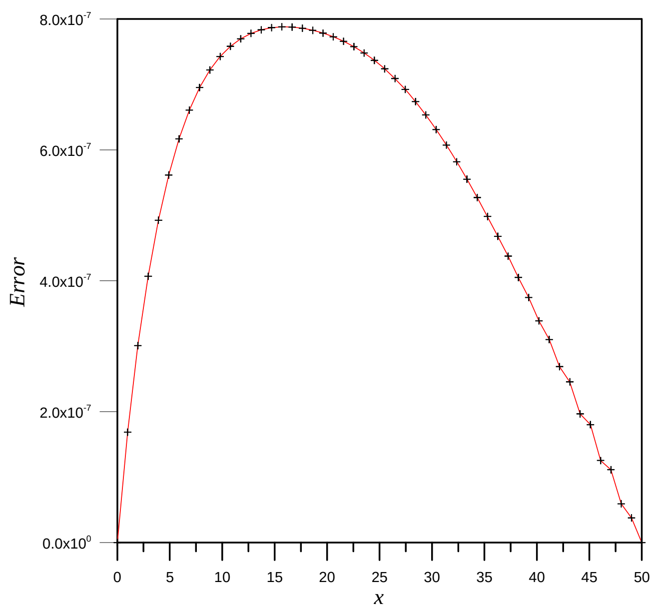

Figure 16: Numerical error of Example 7

8 Conclusions

In this paper, a general dynamical method based on the construction of a scalar homotopy function to transform a vector function of non-linear algebraic equations into a time-dependent scalar function by introducing a fictitious time-like variable is proposed. The important fundamental concepts and the construction of the dynamical method and three different dynamical Newton-like methods are clearly addressed. Several numerical illustrations for the solution of ill-conditioned systems of nonlinear equations are conducted. Findings from this study are drawn as follows.

The general dynamical method is based on the construction of a scalar homotopy function to transform a vector function into a time-dependent scalar function by introducing a fictitious time variable. With the novel formulation proposed in this study, the proposed methods are applied to the solution of ill-conditioned systems of nonlinear equations, i.e. systems having a "nearly singular" Jacobian at some iteration if only poor initial guesses of the solution are available.

Three different dynamical Newton-like methods including the Dynamical Newton Method (DNM), the Dynamical Jacobian-Free Method (DJIFM) and the Manifold-Based Exponentially Convergent Algorithm (MBECA) are derived using different transformation matrices. Illustration examples demonstrate that these different dynamical Newton-like methods can have better numerical stability than Newton's method.

The proposed new DJIFM and MBECA do not need to calculate the inverse of the Jacobian matrix which can retain the numerical stability in cases where the Jacobian matrix is close to zero. Accordingly, the proposed methods demonstrate a great potential for solving higher dimensions, non-linear algebraic equations, as well as for ill-conditioned systems of nonlinear equations.

Numerical experiments show that with the use of a proper control parameter ν, the proposed dynamical methods may converge even faster than Newton's method in certain circumstances for solving the scalar NAE. In addition, difficulties that arose previously in the numerical solution of ill-conditioned systems of nonlinear two-point boundary value problems from poor initial guesses can be removed by means of our proposed dynamical methods.

References

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

Knoll, D.A.; Keyes, D.E. (2004): Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, vol. 193 pp. 357-397.

Lemieux, J.F.; Tremblay, Bruno; Sedláek, Jan; Tupper, Paul; Thomas, Stephen; Huard, David; Auclair, Jean-Pierre (2010): Improving the numerical convergence of viscous-plastic sea ice models with the Jacobian-free Newton-Krylov method. *Journal of Computational Physics*, vol. 229, pp. 2840-2852.

Hirsch, M. W.; Smale, S. (1979): On Algorithms for Solving $f(x) = 0$, *Communications on Pure and Applied Mathematics*, vol. 32, Issue 3, pp. 281–312.

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. (2008): A fictitious time integration method for two-dimensional quasilinear elliptic boundary value problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 33, pp. 179-198.

Liu, C.-S. (2009): A fictitious time integration method for a quasilinear elliptic boundary value problem, defined in an arbitrary plane domain. *CMC: Computers, Materials & Continua*, vol. 11, pp. 15-32.

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.

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(6), pp. 435-450.

Davidenko, D. (1953): On a new method of numerically integrating a system of nonlinear equations. *Doklady Akad. Nauk SSSR*, vol. 88, pp. 601-604.

Liao, S. J. (2004): On the homotopy analysis method for nonlinear problems. *Appl. Math. Comput.*, vol. 47, pp. 49-513.

He, J. H. (2005): Homotopy perturbation method for bifurcation of nonlinear problems, *International Journal of Nonlinear Science and Numerical Simulation*, vol. 6, pp. 207-208.

Kelley, C. T. (2003): Solving Nonlinear Equations with Newton's Method number 1 in Fundamental Algorithms for Numerical Calculations, *SIAM*, Philadelphia.

Liu, C.-S.; Atluri, S. N. (2011): Simple "Residual-Norm" Based Algorithms, for the Solution of a Large System of Non-Linear Algebraic Equations, which Converge Faster than the Newton's Method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 71, no. 3, pp. 279-304.

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