Copyright © 2012 Tech Science Press CMC, vol.31, no.3, pp.173-200, 2012

Dynamical Newton-Like Methods with Adaptive Stepsize for Solving Nonlinear Algebraic Equations

Cheng-Yu Ku[1,2,3] **and Weichung Yeih**[1,2]

Abstract: In this paper, a dynamical Newton-like method with the adaptive stepsize 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 the fictitious time-like function, we derived the adaptive stepsize using the dynamics of the residual vector. Based on the proposed dynamical Newton-like method, we can also derive the dynamical Newton method (DNM) and the dynamical Jacobian-inverse free method (DJIFM) with the transformation matrix as the inverse of the Jacobian and the identity matrix, respectively. These two dynamical Newton-like methods are then adopted for the solution of NAEs. Numerical illustrations demonstrate that taking advantages of the dynamical Newton-like method with the adaptive stepsize the proposed two dynamical Newton-like methods can release limitations of the conventional Newton method such as root jumping, the divergence at inflection points, root oscillations, and the divergence of the root. Results reveal that with the use of the fictitious time-like function the proposed method presents exponential convergence. In addition, taking the advantages of the transformation matrix, the proposed method does not need to calculate the inverse of the Jacobian matrix and thus has great numerical stability.

Keywords: the scalar homotopy method, adaptive stepsize, Jacobian, dynamical Newton-like method, Newton's method.

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,

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

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 non-linear algebraic equations. Over the past years, many contributions have been made towards the numerical solutions of Non-linear Algebraic Equation (NAEs). Most of the methods for solving NAEs are based on the iteration scheme. The iteration-based method, such as Newton's method, also known as the Newton–Raphson method, is perhaps the most welll-known one for finding successively better approximations to the solutions of a real-valued non-linear system [Press et al. (2007)]. Since it converges quadratically, Newton's method can often converge remarkably quickly if the initial guess is sufficiently close to the solution. However, there are some limitations for Newton's method such as root jumping, the divergence at inflection points, oscillations, and the divergence of the root. 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 Jacobian-Free Newton-Krylov method [Knoll and Keyes (2004); Lemieux et al. (2010)] have been extensively developed for this purpose.

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. The stationary point of these evolution equations is the solution for the original algebraic equation. In addition to the FTIM, the homotopy method [Liao (1992); He (2003, 2005); Ku, Yeih and Liu (2010)] can also be used to solve the NAEs using the similar fictitious time concept. Later, the concept of the general dynamical method [Ku, Yeih and Liu (2011)] 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 was proposed. 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) were developed.

In this paper, we introduce a dynamical Newton-like method with the adaptive stepsize based on the general dynamical method proposed by Ku, Yeih and Liu in 2011. With the introduction of the fictitious time-like function, the adaptive stepsize is derived by using the dynamics of the residual vector. Several numerical illustrations including root jumping, the divergence at inflection points, solution oscillations, and the divergence of the root were conducted. The formulation of the proposed method is first described as the follows.

2 The General Dynamical Method

We consider the following NAEs:

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

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

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

where $\mathbf{B}$ is a $n\times n$ Jacobian matrix with its ij-th component being given by $\partial F_i/\partial x_j$. Newton's method can only guarantee the 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,

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

the homotopy method represents a way to enhance the convergence from the local convergence to the global convergence. All the homotopy methods are based on the construction of a vector function, $\mathbf{H}(\mathbf{x},\tau)$ which is called the homotopy function. The homotopy function serves 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 time-like fictitious 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 found while we also require that, once the parameter τ is equal to 1, then $\mathbf{H}(\mathbf{x},\tau)$ 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 ones that can be successfully applied to several different problems. The fixed point homotopy function can be written as

$$\mathbf{H}(\mathbf{x},\ \tau) = \tau\mathbf{F}(\mathbf{x}) + (1-\tau)[\mathbf{x} - \mathbf{x_0}] = \mathbf{0}, \tag{4}$$

and the Newton homotopy function is

$$\mathbf{H}(\mathbf{x},\ \tau) = \tau\mathbf{F}(\mathbf{x}) + (1-\tau)[\mathbf{F}(\mathbf{x}) - \mathbf{F}(\mathbf{x_0})] = \mathbf{0}, \tag{5}$$

where $\mathbf{x_0}$ 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 $\mathbf{F} = \mathbf{0}$ to a scalar equation by noticing that

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

where $\|\mathbf{F}\|^2 = F_1^2 + F_2^2 + \ldots + F_n^2$. Obviously, 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 $F_1 = F_2 = \ldots = F_n = 0$, and thus $\mathbf{F} = \mathbf{0}$.

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

$$h(\mathbf{x},\tau) = \frac{1}{2}\tau \|\mathbf{F}(\mathbf{x})\|^2 + \frac{1}{2}(\tau - 1)\|\mathbf{x} - \mathbf{x_0}\|^2 = 0. \tag{7}$$

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 matrix. 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:

$$\mathbf{H}(\mathbf{x},\tau) = \mathbf{F}(\mathbf{x}) + (\tau - 1)\mathbf{F}(\mathbf{x_0}) = \mathbf{0}. \tag{8}$$

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

$$h(\mathbf{x},\tau) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2 + \frac{1}{2}(\tau - 1)\|\mathbf{F}(\mathbf{x_0})\|^2 = 0. \tag{9}$$

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

$$h(\mathbf{x},t) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2 - \frac{1}{2}\frac{1}{Q(t)}\|\mathbf{F}(\mathbf{x_0})\|^2 = 0, \tag{10}$$

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

$$h(\mathbf{x},t=0) = \frac{1}{2}\|\mathbf{F}(\mathbf{x})\|^2 - \frac{1}{2}\|\mathbf{F}(\mathbf{x_0})\|^2 = 0 \ \Leftrightarrow \mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x_0}) \tag{11}$$

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

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 $h(\mathbf{x},t)=0$ is satisfied for any time greater than zero, multiplying $Q(t)$ at both sides of Eq. (10) we have

$$h(\mathbf{x},t) = \frac{1}{2}Q(t)\|\mathbf{F}(\mathbf{x})\|^2 - \frac{1}{2}\|\mathbf{F}(\mathbf{x_0})\|^2 = 0. \tag{13}$$

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:

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

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

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

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

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

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

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

In Eq. (16), $\mathbf{T}$ is the transformation matrix which can be $\mathbf{B}^{-1}$, the identity matrix, $\mathbf{I}$, $\mathbf{B}^{\mathbf{T}}$, or any other square matrices. With the introduction of different transformation matrices, such as$\mathbf{B}^{-1}$, $\mathbf{I}$, or $\mathbf{B}^{\mathbf{T}}$, 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

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

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, $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, can be embedded in a system of nonlinear ODEs: $\dot{\mathbf{x}} = -v/q(\tau)\mathbf{F}(\mathbf{x})$ 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

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

Hence, we have

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

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

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

where m is a control parameter for speeding the convergence as discussed in Ku, et al. (2009) and v 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

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

Inserting Eq. (22) into Eq. (19) and let $v = 1$, we have

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

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

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

We can easily find that Eq. (21) and Eq. (24) embeds 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:

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

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{\Delta t}{2}\frac{\left\|\mathbf{F}(\mathbf{x}^k)\right\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{x}^k)\mathbf{B}(\mathbf{x}^k)\mathbf{T}(\mathbf{x}^k)\mathbf{F}(\mathbf{x}^k)}\mathbf{T}(\mathbf{x}^k)\mathbf{F}(\mathbf{x}^k). \tag{26}$$

where Δt 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 $\mathbf{B}^{-\mathbf{1}}$, $\mathbf{I}$, and $\mathbf{B}^{\mathbf{T}}$. For simplicity, let $\mathbf{u} = \mathbf{TF}(\mathbf{x})$. Rewriting Eq. (24), we can obtain the evolution dynamics of $\mathbf{x}$ as:

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

By defining $\mathbf{v} = \mathbf{Bu}$, one can rewrite Eq. (27) as:

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{x})\mathbf{v}}\mathbf{u} \tag{28}$$

4 Dynamics of the Residual Vector

Let us take a look of the evolution of the residual vector $\mathbf{F}$, it can be written as:

$$\dot{\mathbf{F}}(\mathbf{x}(t)) = \mathbf{B}\dot{\mathbf{x}}. \tag{29}$$

Substituting Eq. (28) into Eq. (29), we then have:

$$\dot{\mathbf{F}}(\mathbf{x}(t)) = \frac{-\dot{Q}(t)}{2Q(t)}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{x})\mathbf{v}}\mathbf{v} \tag{30}$$

Using the forward Euler scheme, we can approximately express Eq. (30) as

$$\mathbf{F}(\mathbf{x}(t+\Delta t)) = \mathbf{F}(\mathbf{x}(t)) - \Delta t\frac{\dot{Q}(t)}{2Q(t)}\frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{T}(\mathbf{x})\mathbf{v}}\mathbf{v} \tag{31}$$

where Δt is the time increment. For simplicity, we let

$$\beta := \Delta t \frac{\dot{Q}(t)}{2Q(t)}. \tag{32}$$

Since we require that the evolution path of $\mathbf{x}$ should always remain on the space-time manifold, we then can obtain the following expressions from Eq. (10):

$$\|\mathbf{F}(\mathbf{x}(t))\|^2 = \frac{\|\mathbf{F}(\mathbf{x}_0)\|^2}{Q(t)}$$

and

$$\|\mathbf{F}(\mathbf{x}(t+\Delta t))\|^2 = \frac{\|\mathbf{F}(\mathbf{x}_0)\|^2}{Q(t+\Delta t)} \tag{33}$$

Taking the square norm of Eq. (33) for both sides and using Eqs. (31) and (32), we can obtain

$$\frac{\|F(x_0)\|^2}{Q(t+\Delta t)} = \frac{\|F(x_0)\|^2}{Q(t)} - 2\beta \frac{\|F(x_0)\|^2}{Q(t)} + \beta^2 \frac{\|F(x_0)\|^2}{Q(t)} \frac{\|F(x)\|^2}{(F^T(x)v)^2} \|v\|^2. \tag{34}$$

Rearranging the above equation, we can obtain an algebraic equation for β as

$$a_0\beta^2 - 2\beta + 1 - \frac{Q(t)}{Q(t+\Delta t)} = 0, \tag{35}$$

where

$$a_0 = \frac{\|\mathbf{F}(\mathbf{x})\|^2 \|\mathbf{v}\|^2}{(\mathbf{F}^T(\mathbf{x})\mathbf{v})^2} = \left\{ \frac{\|\mathbf{F}(\mathbf{x})\| \|\mathbf{v}\|}{(\mathbf{F}^T(\mathbf{x})\mathbf{v})} \right\}^2 = \left(\frac{1}{\cos\theta} \right)^2 \tag{36}$$

in which the angle θ denotes the angle between the residual vector $\mathbf{F}$ and the vector $\mathbf{v}$. From the Cauchy-Schwarz inequality, it can be easily verified that $a_0 \geq 1$. Now let us define:

$$s := \frac{Q(t)}{Q(t+\Delta t)} = \frac{\|\mathbf{F}(\mathbf{x}(t+\Delta t))\|^2}{\|\mathbf{F}(\mathbf{x}(t))\|^2}. \tag{37}$$

It can be found that this ratio s is the ratio between the square norm of the residual vector in the next state and the square norm of the residual vector in the current state. It is for sure that we hope $s \leq 1$, such that for each state the norm of the residual vector decreases. From Eq. (37), Eq. (35) now can be written as

$$a_0\beta^2 - 2\beta + 1 - s = 0. \tag{38}$$

Therefore, we have

$$s = a_0\beta^2 - 2\beta + 1. \tag{39}$$

Rewriting Eq. (39), we can obtain

$$\frac{1}{s} = \frac{1}{a_0\beta^2 - 2\beta + 1}. \tag{40}$$

Since s is the ratio between the square norm of the residual vector in the next state and the square norm of the residual vector in the current state. Rewriting Eq. (37) and using the forward Euler scheme, we have

$$s = \frac{Q(t)}{Q(t+\Delta t)} = \frac{Q(t)}{Q(t) + \dot{Q}(t)\Delta t}. \tag{41}$$

Eq. (41) can be rewritten as

$$\frac{1}{s} = \frac{Q(t) + \dot{Q}(t)\Delta t}{Q(t)} = 1 + \frac{\dot{Q}(t)}{Q(t)}\Delta t. \tag{42}$$

Inserting Eq. (32) into the above equation, we obtain

$$\frac{1}{s} = 1 + 2\beta. \tag{43}$$

Combining Eqs. (40) and (43), we have the following equation.

$$\frac{1}{a_0\beta^2 - 2\beta + 1} = 1 + 2\beta. \tag{44}$$

Rearranging Eq. (44), we obtain

$$\beta^2(2a_0\beta + (a_0 - 4)) = 0. \tag{45}$$

Accordingly, we have the exact solution of β as

$$\beta = \frac{4 - a_0}{2a_0}. \tag{46}$$

From Eq. (36), it is obvious that $a_0 \geq 1$. Again, we can easily find that $\beta \leq 1.5$. In addition, the fictitious time-like function should satisfy the conditions that $Q(t) > 0$, $Q(0) = 1$, and $Q(t)$ is a monotonically increasing function of t, and $Q(\infty) = \infty$. It is obvious to know that $\beta > 0$ from Eq. (32). From Eq. (46), it is found that $a_0 \leq 4$ while $\beta > 0$. Accordingly, we can obtain that the appropriate value of a_0 should be $1 \leq a_0 \leq 4$. That means if we hope the trajectory of the solution reamains on the manifold, the value of a_0 should satisfy the above restriction. In the following, we will discuss how to determine the adaptive stepsize.

A simple estimation of the stepsize is derived from Eq.(46) when $1 \leq a_0 \leq 4$ is satisfied. Using Eq.(32) and (46) together and using the fact $Q(t) = e^{\nu t}$, one can derive that $\Delta t = \frac{4-a_0}{\nu a_0}$. For the best choice of a_0, the transformation matrix $\mathbf{T}$ is $\mathbf{B}^{-1}$ and a_0=1. While that case is true and ν=1 we have Δt=3. However, the above-mentioned estimaion may overestimate the stepsize. The reasons come from that in Eq.(31) and Eq.(42) approximations using the forward Euler scheme are adopted. Then when we use the exact form of $\dot{Q}(t)$ in Eq.(32) may not be appropriate.

Using the Euler scheme, we have

$$Q(t+\Delta t) = Q(t) + \dot{Q}(t)\Delta t. \tag{47}$$

It means that $\dot{Q}(t) = \frac{Q(t+\Delta t)-Q(t)}{\Delta t}$.

Inserting the above equation into Eq. (32) and now using $Q(t) = e^{\nu t}$, we obtain

$$\beta = \frac{1}{2}(e^{\nu \Delta t} - 1). \tag{48}$$

Accordingly, the adaptive step size is

$$\Delta t = \frac{1}{\nu}\ln(2\beta + 1). \tag{49}$$

From Eq. (45), we can easily find the solution of Δt if a_0 and ν are given. Figure 1 shows the relationship of the stepsize and a_0. From Fig. 1, one can find that the maximum value of a_0 should be less than 4 because the stepsize is very close to zero while $a_0 = 4$. Beside, if $a_0 = 1$ and $\nu = 1$, the stepsize is 1.3863, which is much less than the first estimation, which is 3. To adopt smaller stepsize estimation as Eq.(49) states is more conservative and in the followings of this article, Eq.(49) is used to estimate the adaptive stepsize.

5 Numerical illustrations of the Dynamical Newton method

The dynamical Newton method (DNM) developed by Ku, Yeih, and Liu in 2011 can be described as follows.

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)}\mathbf{B}^{-1}\mathbf{F}(\mathbf{x}). \tag{50}$$

To derive the DNM, the transformation matrix $\mathbf{T}$ is set to be $\mathbf{B}^{-1}$ or $\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}$ and $\mathbf{v} = \mathbf{F}$. If the fictitious time function $Q(t) = e^{\nu t}$ is adopted, we derive the DNM as

$$\dot{\mathbf{x}} = -\frac{\nu}{2}\mathbf{B}^{-1}\mathbf{F}(\mathbf{x}). \tag{51}$$

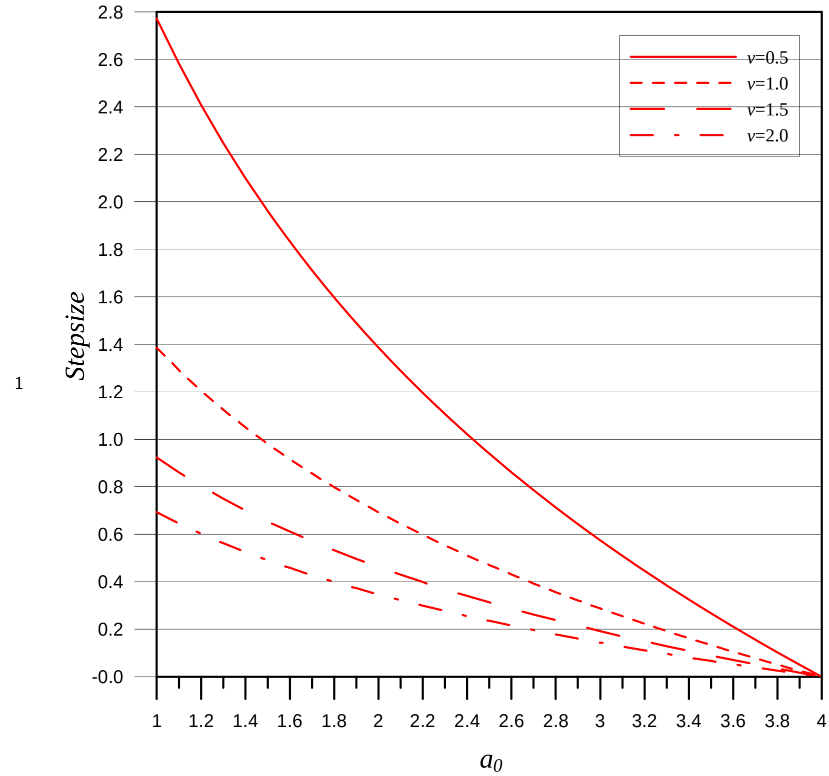

Figure 1: The a_0 versus the stepsize for different values of v.

Using the forward Euler scheme, we have

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{v\Delta t}{2}[\mathbf{B}(\mathbf{x}^k)]^{-1}\mathbf{F}(\mathbf{x}^k). \tag{52}$$

Eq. (30) is identical to Newton's method if we adopted $\Delta t = 2$ and let $v = 1$. Newton's method is a simple iterative numerical method to approximate roots of equations. However, there are some limitations such as root jumping, the divergence at inflection points, root oscillations, and the divergence of the root. From Eq. (36), we can easily find that the stepsize Δt in Eq. (30) can be directly determined from Eqs. (46) to (48) because the value of a_0 is always equal to one in the evolution for the DNM. Accordingly, we developed the DNM with the adaptive stepsize. The following examples demonstrate the advantages of using the adaptive stepsize of the DNM to avoid limitations mentioned above.

5.1 Example 5.1

We first consider a simple scalar equation as

$$F(x) = \sin x. \tag{53}$$

This example is to demonstrate the root jumping can be avoided by using the DNM with the adaptive stepsize. In this example where the function $F(x) = \sin x$ is oscillating and has a number of roots, one may choose an initial guess close to a root. However, the guesses may jump and converge to some other root. In this example, we first use the Newton method to solve Eq. (53). With the initial guess of $x = 2.4\pi$, the nearest root is $x = 2\pi$. However, the result obtained from Newton's method shows that the root of Eq. (53) converges to $x = 0$ instead of $x = 2\pi$. Using the DNM with the adaptive stepsize with the same initial value of $x = 2.4\pi$ and $v = 0.5$, very accurate solution of $x = 2\pi$ with the residual to the order of 10^{-6} can be obtained. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 2(a). The adaptive stepsize for this example is $\Delta t = 2.77$ and the fictitious time step is 12 for reaching the solution. The solution path of both methods is shown in Fig. 2(b).

5.2 Example 5.2

The second example is a simple nonlinear equation as

$$F(x) = (x-1)^3 + 0.512. \tag{54}$$

Selection of the initial guess or an iteration value of the root that is close to the inflection point of the function $F(x)$ may start diverging away from the root in the Newton method. This example is to demonstrate the divergence at inflection points can be reduced by using the DNM with the adaptive stepsize. In this example, we first use the Newton method to solve Eq. (54). With the same initial guess of $x = 5.1155$ and $v = 0.5$, we adopted the Newton method and the DNM with the adaptive stepsize of $\Delta t = 2.77$. The root mean square norm versus the fictitious time for both methods is shown in Fig. 3(a). It is found that there is a dramatic jump when the number of step is 6 in Newton's method. However, for our proposed method the jump of the root mean square norm is relatively small compared to Newton's method. Both methods can converge to the solution of $x = 0.2$. However, our proposed method converges to the solution in only 24 fictitious time steps. It is much faster than the Newton method in which 57 iteration steps are needed for the convergence. The solution paths for both methods are shown in Fig. 3(b) and Fig. 3(c).

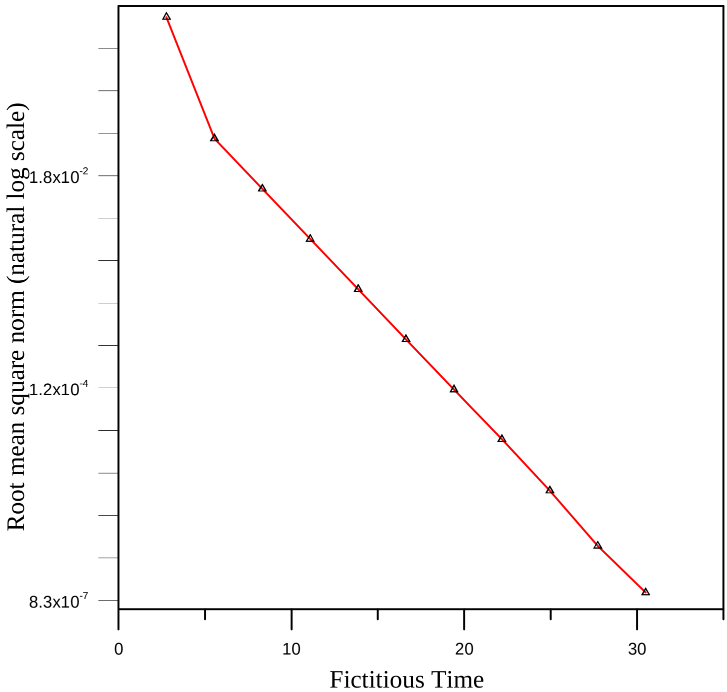

Figure 2(a): The exponential convergence for Example 5.1.

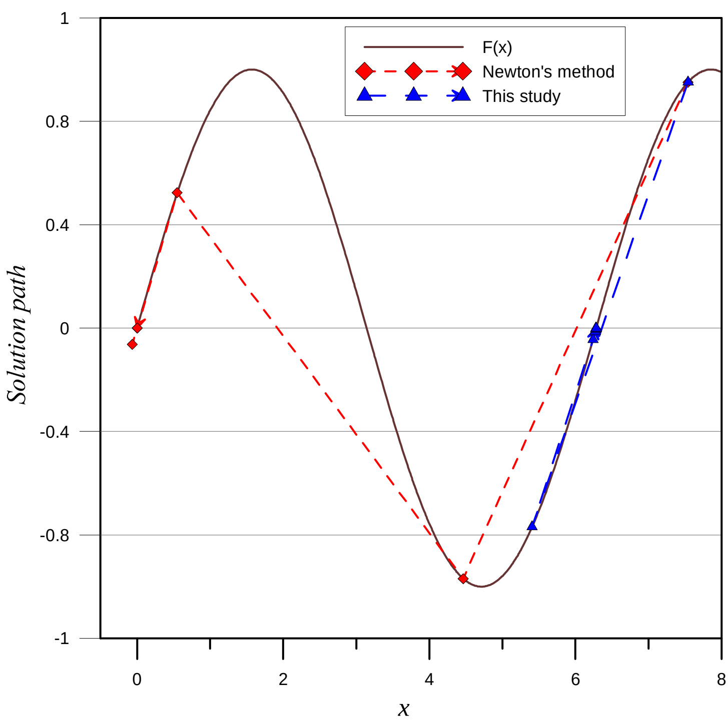

Figure 2(b): Comparison of the solution path for Example 5.1.

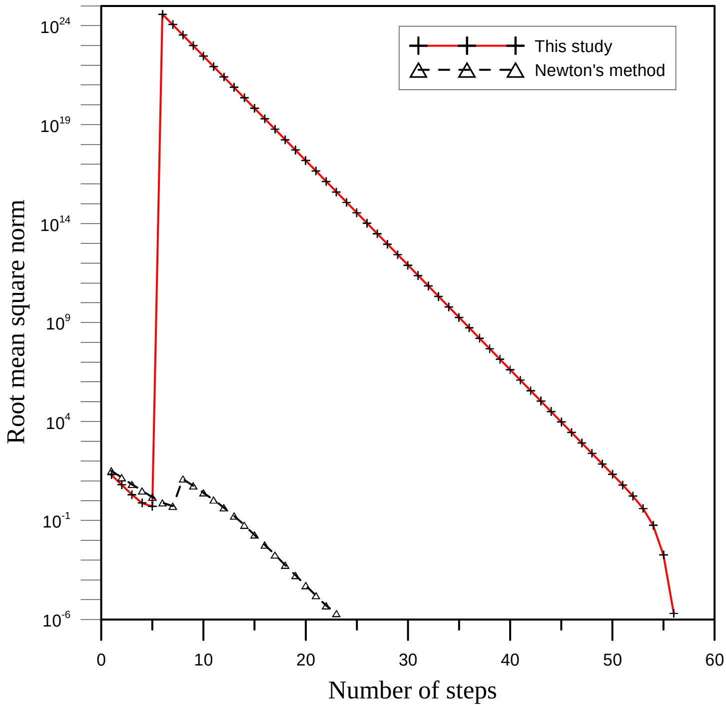

Figure 3(a): Comparison of the convergence for Example 5.2.

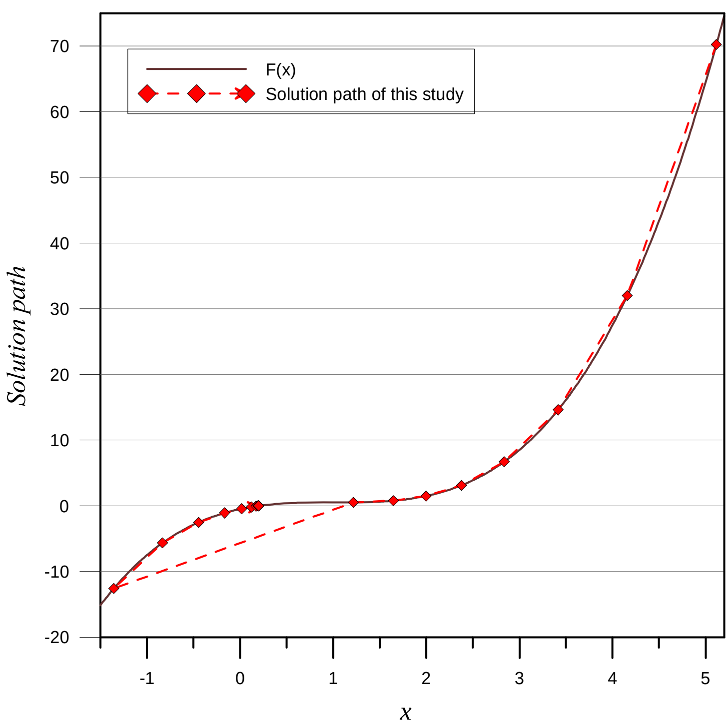

Figure 3(b): The solution path of this study for Example 5.2.

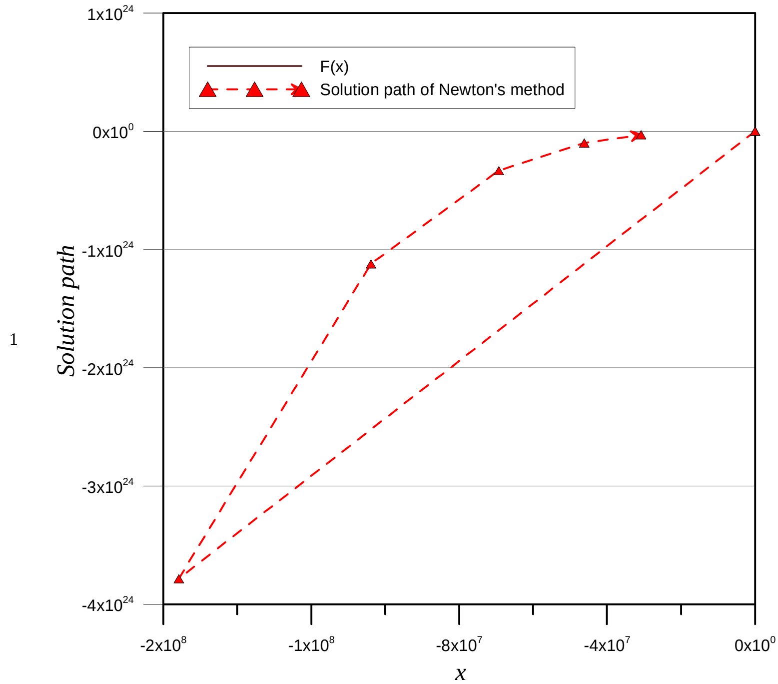

Figure 3(c): The solution path of Newton's method for Example 5.2.

5.3 *Example 5.3*

The third example is a simple nonlinear equation as

$$F(x) = x^4 + 4x^3 + 4x^2 - x - 1. \tag{55}$$

This example is to demonstrate the oscillations of finding the root can be avoided by using the DNM with the adaptive stepsize. Since $F(-1) > 0$ and $F(0) < 0$, it is known that there exists a solution in $(-1,\ 0)$ by the intermediate-value theorem. In this example, we use the Newton method and the DNM with the adaptive stepsize to solve Eq. (55). Results obtained from the Newton method shows that the root of Eq. (55) oscillate between –1 and 0 and consequently don't converge to the solution if the initial value of 0 is adopted. Using the DNM with the adaptive stepsize with the same initial value of 0 and $v = 0.5$, very accurate solution of $x = -0.4751$ with the residual to the order of 10^{-6} can be obtained. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 4(a). The adaptive stepsize for this example is $\Delta t = 2.77$ and the fictitious time step is 12 for reaching the solution. The solution path of both methods is shown in Fig. 4(b).

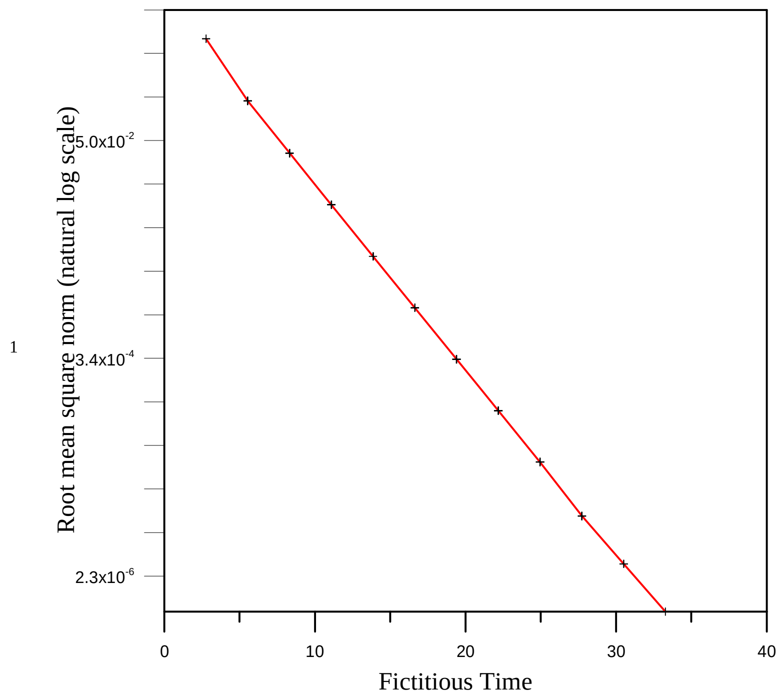

Figure 4(a): The exponential convergence for Example 5.3.

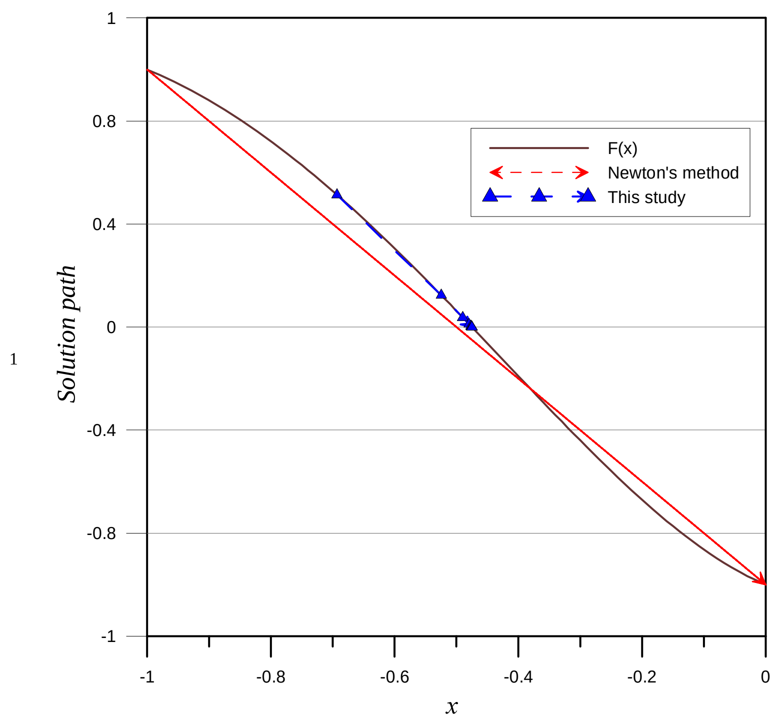

Figure 4(b): Comparison of the solution path for Example 5.3.

5.4 Example 5.4

This is an example of where Newton's method goes to the wrong solution. Consider the equation

$$F(x) = \frac{x}{1+x^2}. \tag{56}$$

This example is to demonstrate the divergence of finding the root can be avoided by using the DNM with the adaptive stepsize. Clearly, $x = 0$ is the solution to Eq. (56). In this example, we use the Newton method and the DNM with the adaptive stepsize to solve Eq. (56). Results obtained from the Newton method shows that instead of converging to the solution of $x = 0$, the root of Eq. (56) diverges to infinity if the initial value of $x = 0.6$ is adopted. Using the DNM with the adaptive stepsize with the same initial value of $x = 0$ and $v = 0.5$, very accurate solution of $x = -3.84 \times 10^{-7}$ with the residual to the order of 10^{-6} can be obtained. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 5(a). The adaptive stepsize for this example is $\Delta t = 2.77$ and the fictitious time step is 12 for reaching the solution. The solution path of both methods is shown in Fig. 5(b).

6 Numerical illustrations of the Dynamical Jacobian-Inverse Free Method

The previous section has demonstrated the advantages of using the DNM with the adaptive stepsize. However, for solving NAEs the computation of the inverse of the Jacobian matrix is needed which may have limitations in cases where the determinant of the Jacobian matrix is close to zero. The dynamical Jacobian-inverse free method (DJIFM) developed by Ku, Yeih, and Liu in 2011 does not need to calculate the inverse of the Jacobian matrix and has a great numerical stability. To derive the DJIFM, we let the transformation matrix, $\mathbf{T}$, be the identity matrix, $\mathbf{I}$ or $\mathbf{u} = \mathbf{IF}$(i.e. $\mathbf{v} = \mathbf{BF}$). We have

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}(\mathbf{x})\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{x})\mathbf{BF}(\mathbf{x})}\mathbf{F}(\mathbf{x}). \tag{57}$$

If we choose the fictitious time function $Q(t) = e^{vt}$, we derive the DJIFM as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{v\Delta t}{2} \frac{\|\mathbf{F}(\mathbf{x}^k)\|^2}{\mathbf{F}^{\mathbf{T}}(\mathbf{x}^k)\mathbf{B}(\mathbf{x}^k)\mathbf{F}(\mathbf{x}^k)}\mathbf{F}(\mathbf{x}^k). \tag{58}$$

In Eq. (58), 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 this study, the adaptive

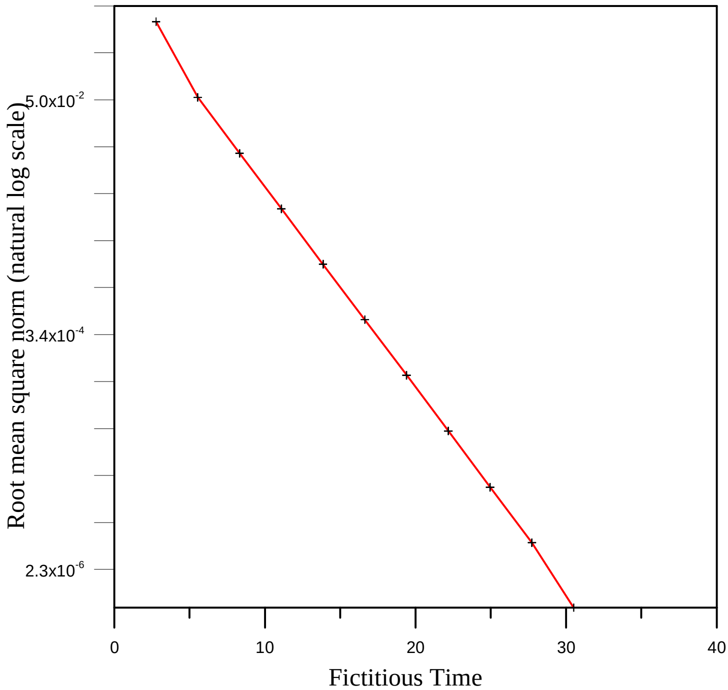

Figure 5(a): The exponential convergence for Example 5.4.

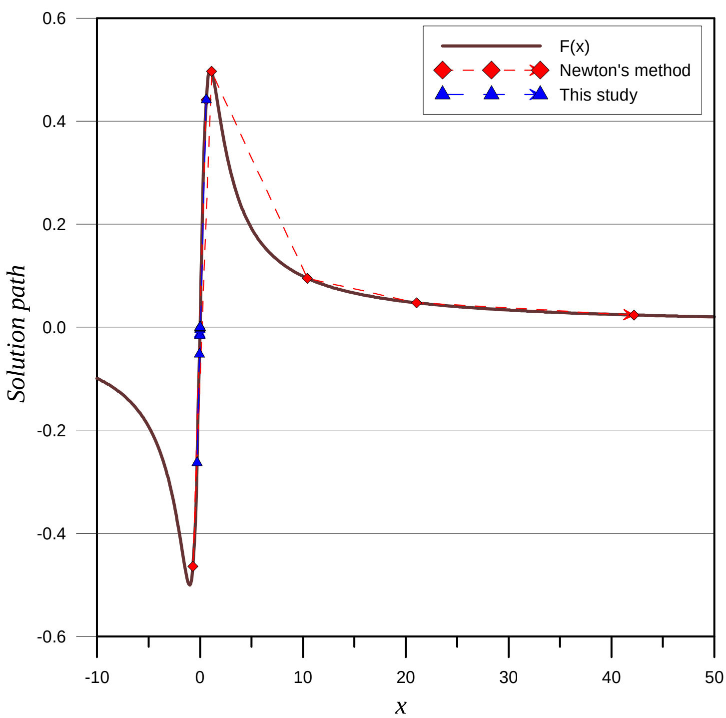

Figure 5(b): Comparison of the solution path for Example 5.4.

stepsize Δt in Eq. (36) can be determined from Eq. (46) for the DJIFM. To clarify the characteristics of the DJIFM with adaptive stepsize, several examples were conducted as follows.

6.1 Example 6.1

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

$$\begin{aligned} F_1(u,v) &= u^2 + v = 0, \\ F_2(u,v) &= -v^2 + 16 = 0. \end{aligned} \tag{59}$$

In this test, we compare the numerical stability of Newton's method and the DJIFM with the adaptive stepsize. We start from an initial value of $(u,v) = (1,\ 0)$. Results obtained show that the conventional Newton method and the DNM diverge because the initial value causes singular Jacobian matrix. Since the DJIFM with the adaptive stepsize need not compute the inverse the Jacobian matrix, with the same initial value of $(u,v) = (1,\ 0)$, and $v = 0.5$, very accurate solution of $(u,v) = (2,\ -4)$ with the residual to the order of 10^{-6} can be obtained. The fictitious time step is 12 for reaching the solution. We set the maximum value of a_0 is 3.97 to avoid the stepsize close to zero numerically. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 6(a). The a_0 versus the fictitious time step and the stepsize versus the fictitious time step are shown in Fig. 6(b) and Fig. 6(c), respectively.

6.2 Example 6.2

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

$$\begin{aligned} F_1(x_1,x_2) &= x_1^2 + x_2^2 - 2 = 0, \\ F_2(x_1,x_2) &= e^{(x_1-1)} + x_2^2 - 2 = 0, \end{aligned} \tag{60}$$

where the Jacobian matrix is

$$\mathbf{B} = \begin{bmatrix} 2x_1 & 2x_2 \\ e^{(x_1-1)} & 2x_2 \end{bmatrix}. \tag{61}$$

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 with the adaptive stepsize. Starting from the same initial value, $(x_1,x_2) = (3,\ 5)$, Newton's method does not converge to the solution. On the other hand, our proposed method converges to the solution of $(x_1,x_2) = (-0.4777, -1.3311)$ with

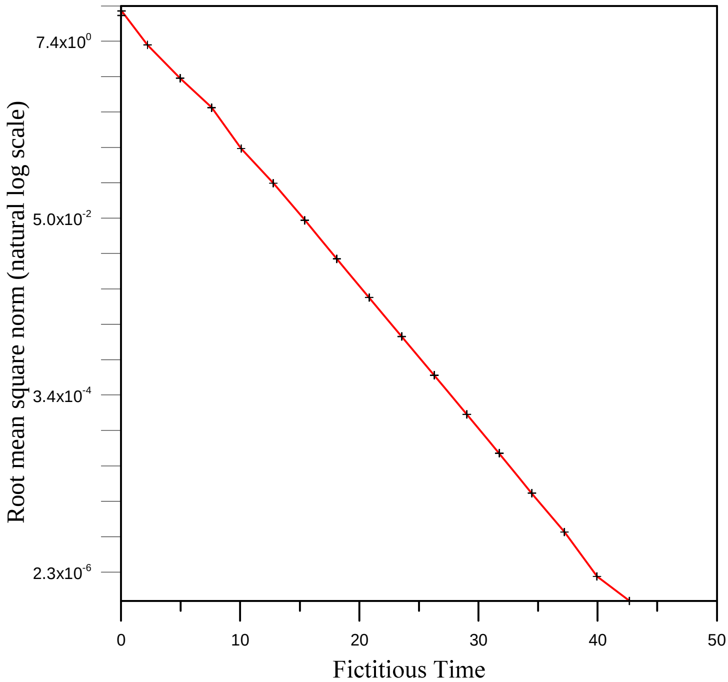

Figure 6(a): The exponential convergence rate for Example 6.1.

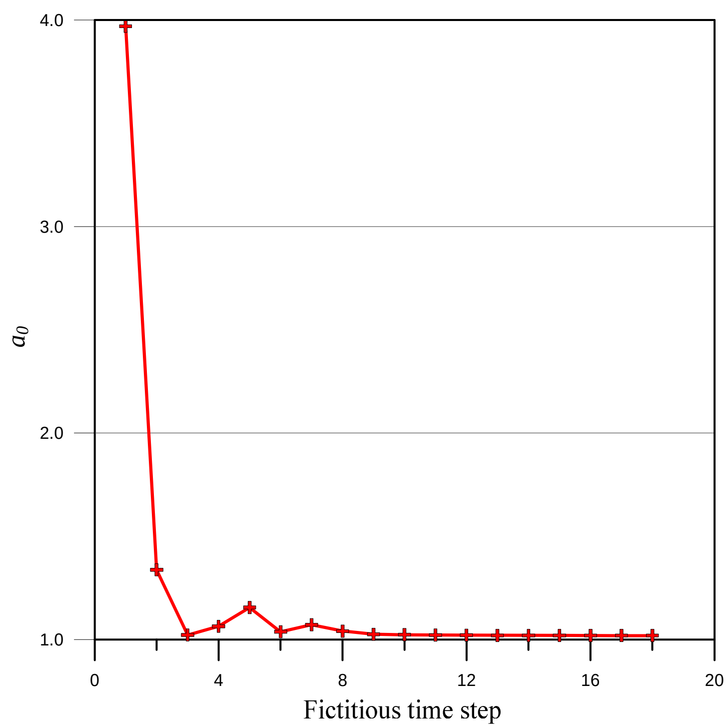

Figure 6(b): Evolution of a_0 for Example 6.1.

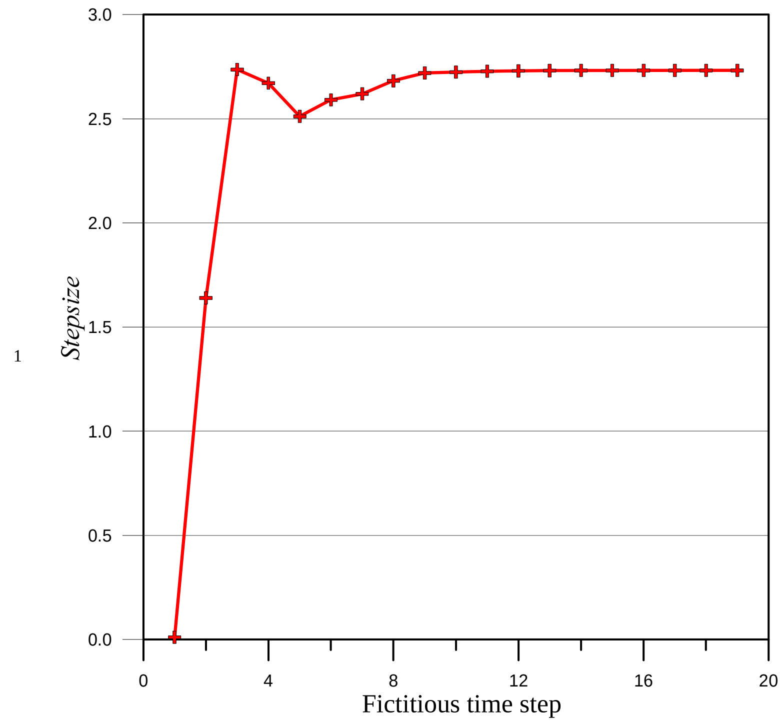

Figure 6(c): Evolution of the fictitious time stepsize for Example 6.1.

the root mean square norm in the order of 10^{-6}. The fictitious time step is 113 for reaching the solution. Again, starting from the same initial value, $(x_1, x_2) = (3, 1)$, Newton's method still does not converge to the solution. With the same initial value and $v = 1.0$, very accurate solution of $(x_1, x_2) = (1, 1)$ with the residual to the order of 10^{-6} can be obtained. The fictitious time step is 46 for reaching the solution. We set the maximum value of a_0 is 3.8 to avoid the stepsize close to zero numerically. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 7(a). The a_0 versus the fictitious time step and the stepsize versus the fictitious time step are shown in Fig. 7(b) and Fig. 7(c), respectively. This example reveals that the proposed method has the advantages to obtain the solution which the solution search fails in the conventional Newton method.

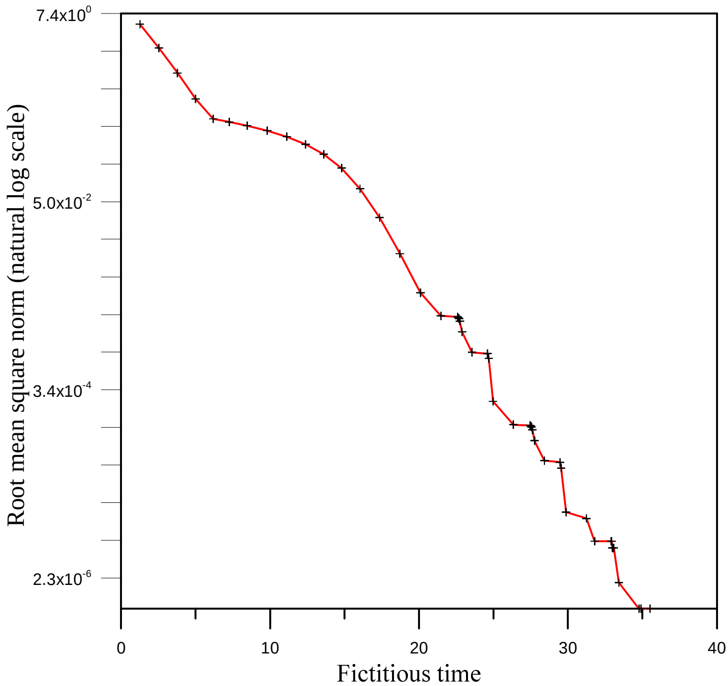

Figure 7(a): The exponential convergence rate for Example 6.2.

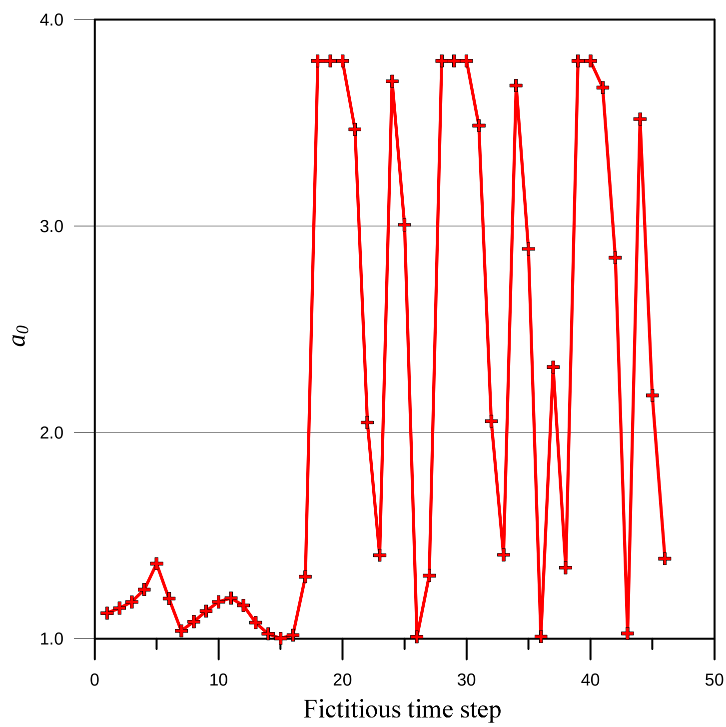

Figure 7(b): Evolution of a_0 for Example 6.2.

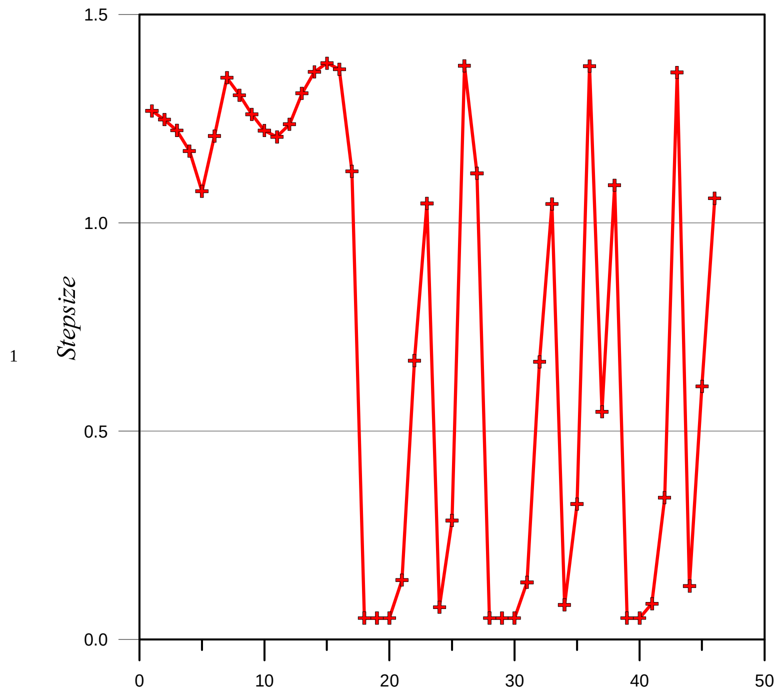

Figure 7(c): Evolution of the fictitious time stepsize for Example 6.2.

6.3 Example 6.3

The last example to be investigated is to solve the following boundary value problem.

$$u'' = 3/2u^2, \tag{62}$$

The boundary conditions are $u(0) = 4,\ u(1) = 1$. Equation (62) has an exact solution as follows.

$$u(x) = \frac{4}{(1+x)^2}. \tag{63}$$

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

$$F_i = \frac{1}{\Delta x^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2, \tag{64}$$

with the boundary conditions of

$$u_0 = 4,\ u_{n+1} = 1 \tag{65}$$

where $\Delta x = \frac{1}{(n+1)}$.
In this example, we adopt the proposed method and let the initial value of $u = 1$ at grid points and the number of the grid point $n = 19$ to solve Eq. (62). With $v = 1.0$, very accurate solution, shown in Fig. 8(a) with the residual to the order of 10^{-6} can be obtained. The fictitious time step is 35 for reaching the solution. We set the maximum value of a_0 is 3.8 to avoid the stepsize close to zero numerically. The root mean square norm versus the fictitious time presents the exponential convergence as shown in Fig. 8(b). The a_0 versus the fictitious time step and the stepsize versus the fictitious time step are shown in Fig. 8(c) and Fig. 8(d), respectively.

7 Conclusions

In this paper, a dynamical Newton-like method with adaptive stepsize 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 is proposed. The important fundamental concepts and the construction of the dynamical Newton-like method with adaptive stepsize are clearly addressed. Several numerical illustrations are conducted. Findings from this study are drawn as follows.

1. Taking advantages of the dynamical Newton-like method with adaptive stepsize, the proposed two dynamical Newton-like methods can release limitations of the conventional Newton method such as root jumping, the divergence at inflection points, root oscillations, and the divergence of the root.

2. The formulation derived in this study reveals that the conventional iterative scheme can be fully described by the proposed general dynamical method if certain fictitious time-like function and fictitious time step are adopted. In addition, the characteristics of the convergence for solving problems can be prescribed by the selected fictitious time function.

3. In this study, the exponential time-like function is adopted in the formulation for finding the adaptive stepsize. Results reveal that with the use of the fictitious time-like function the proposed method presents exponentially convergent. Other possible fictitious time-like functions to reach a fast convergence rate are suggested to study in the future.

Acknowledgement: This study was partially supported by the National Science Council under project grant NSC100-2628-E-019-054-MY3 and NSC100-3113-E-007-011. The author would like to thank the National Science Council for the generous financial support.

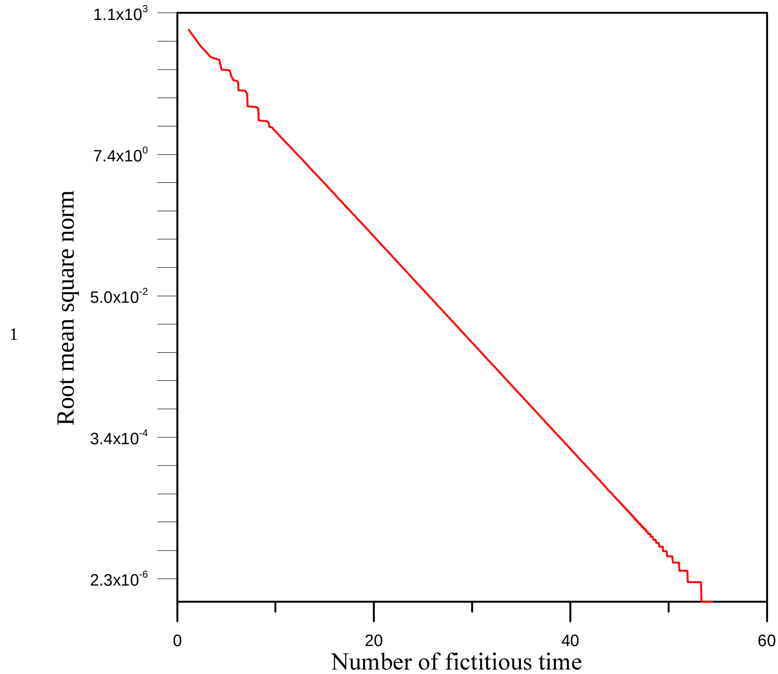

Figure 8(a): The exponential convergence rate for Example 6.3.

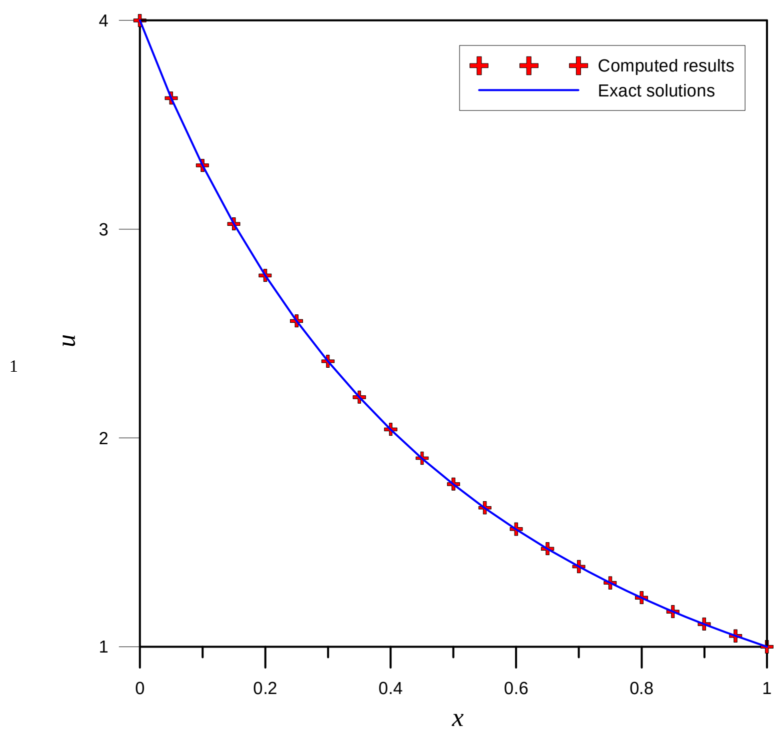

Figure 8(b): Comparison of the exact solution and the computed results for Example 6.3.

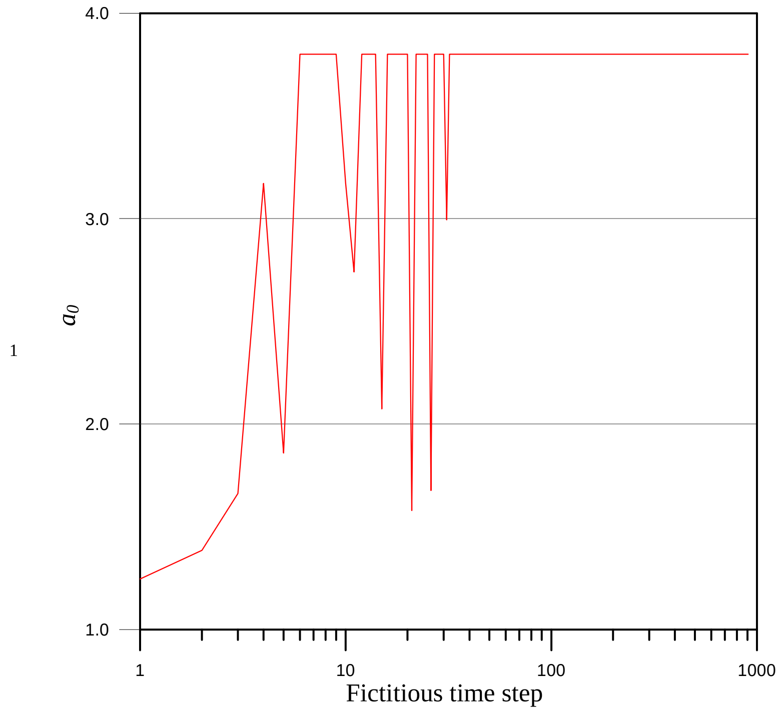

Figure 8(c): Evolution of a_0 for Example 6.3.

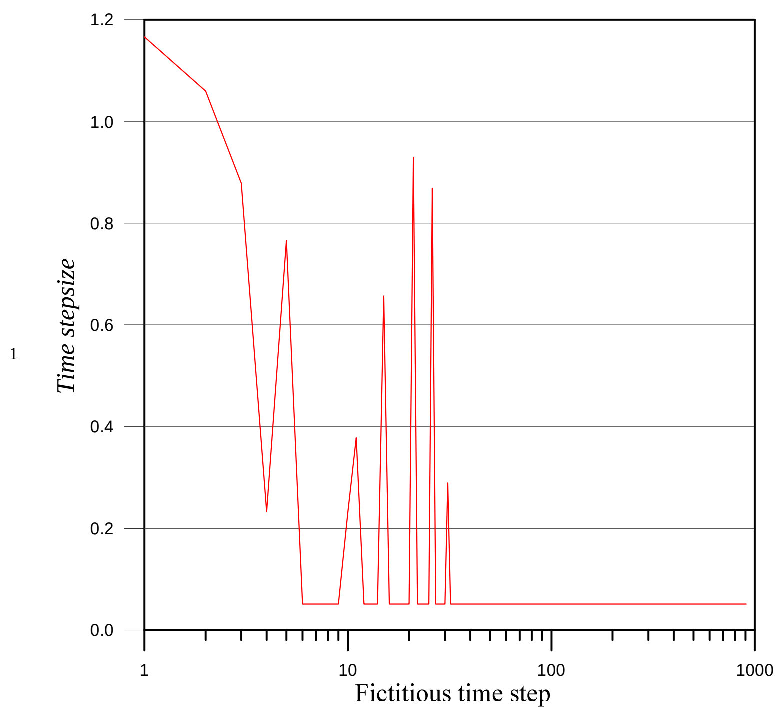

Figure 8(d): Evolution of the fictitious time stepsize for Example 6.3.

References

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

He, J. H. (2003): Homotopy perturbation methods: a new nonlinear analytical technique, *Applied Mathematics and Computation*, 135, pp. 73–79.

He, J. H. (2005): Homotopy perturbation method for bifurcation of nonlinear problems, *International Journal of Nonlinear Science and Numerical Simulation*, v6. 207-208.

Kelley, C. T. (2003): Solving Nonlinear Equations with Newton's Method. SIAM, Philadelphia, PA.

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

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*, 11(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, 2011.

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

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.

Lemieux, J.F.; Tremblay, Bruno; Sedlá?ek, 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*, 229, pp. 2840-2852.

Press, William H.; Teukolsky, Saul A.; Vetterling, William T.; Flannery, Brian

P. (2007): Numerical Recipes, The Art of Scientific Computing, Cambridge University Press.