

# A Novel Time Integration Method for Solving A Large System of Non-Linear Algebraic Equations

Chein-Shan Liu<sup>1</sup> and Satya N. Atluri<sup>2</sup>

**Abstract:** Iterative algorithms for solving a nonlinear system of algebraic equations of the type:  $F_i(x_j) = 0$ ,  $i, j = 1, \dots, n$  date back to the seminal work of Issac Newton. Nowadays a Newton-like algorithm is still the most popular one due to its easy numerical implementation. However, this type of algorithm is sensitive to the initial guess of the solution and is expensive in the computations of the Jacobian matrix  $\partial F_i / \partial x_j$  and its inverse at each iterative step. In a time-integration of a system of nonlinear Ordinary Differential Equations (ODEs) of the type  $B_{ij}\dot{x}_j + F_i = 0$  where  $B_{ij}$  are nonlinear functions of  $x_j$ , the methods which involve an inverse of the Jacobian matrix  $B_{ij} = \partial F_i / \partial x_j$  are called “Implicit”, while those that do not involve an inverse of  $\partial F_i / \partial x_j$  are called “Explicit”. In this paper a natural system of explicit ODEs is derived from the given system of nonlinear algebraic equations (NAEs), by introducing a fictitious time, such that it is a mathematically equivalent system in the  $n + 1$ -dimensional space as the original algebraic equations system is in the  $n$ -dimensional space. The iterative equations are obtained by applying numerical integrations on the resultant ODEs, which do not need the information of  $\partial F_i / \partial x_j$  and its inverse. The computational cost is thus greatly reduced. Numerical examples given confirm that this fictitious time integration method (FTIM) is highly efficient to find the true solutions with residual errors being much smaller. Also, the FTIM is used to study the attracting sets of fixed points, when multiple roots exist.

**Keyword:** Nonlinear algebraic equations, Iterative method, Ordinary differential equations, Fictitious time integration method (FTIM)

## 1 Introduction

Numerical solution of algebraic equations is one of the main aspects of computational mathematics. In many practical nonlinear engineering problems, the methods such as the finite element method, boundary element method, finite volume method, the meshless method, etc., eventually lead to a system of nonlinear algebraic equations (NAEs). Many numerical methods used in computational mechanics, as demonstrated by Zhu, Zhang and Atluri (1998), Atluri and Zhu (1998a), Atluri (2002), Atluri and Shen (2002), and Atluri, Liu and Han (2006) lead to the solution of a system of linear algebraic equations for a linear problem, and of an NAEs system for a nonlinear problem. Collocation methods, as those used by Liu (2007a, 2007b, 2007c, 2008a) for the modified Trefftz method of Laplace equation also need to solve a large system of algebraic equations.

Over the past forty years two important contributions have been made towards the numerical solutions of NAEs. One of the methods has been called the “predictor-corrector” or “pseudo-arclength continuation” method. This method has its historical roots in the embedding and incremental loading methods which have been successfully used for several decades by engineers to improve the convergence properties when an adequate starting value for an iterative method is not available. Another is the so-called simplicial or piecewise linear method. The monographs by Allgower and Georg (1990) and Deuffhard (2004) are devoted to the continuation methods for solving NAEs.

---

<sup>1</sup> Department of Mechanical and Mechatronic Engineering, Taiwan Ocean University, Keelung, Taiwan. E-mail: cslu@mail.ntou.edu.tw

<sup>2</sup> Center for Aerospace Research & Education, University of California, Irvine

In this paper we introduce a novel continuation method, by embedding the NAEs into a system of nonautonomous first order ODEs (FOODEs). To motivate the present approach, we consider a single NAE:

$$F(x) = 0. \quad (1)$$

In the above equation, we only have an independent variable  $x$ . We transform it into a FOODE by introducing a time-like or fictitious variable  $t$  into the following transformation of variable from  $x$  to  $y$ :

$$y(t) = (1+t)x. \quad (2)$$

Here,  $t$  is a variable which is independent of  $x$ ; hence,  $\dot{y} = dy/dt = x$ . If  $v \neq 0$ , Eq. (1) is equivalent to

$$0 = -vF(x). \quad (3)$$

Adding the equation  $\dot{y} = x$  to Eq. (3) we obtain:

$$\dot{y} = x - vF(x). \quad (4)$$

By using Eq. (2) we can derive

$$\dot{y} = \frac{y}{1+t} - vF\left(\frac{y}{1+t}\right). \quad (5)$$

This is a FOODE for  $y(t)$ . The initial condition for the above equation is  $y(0) = x$ , which is however an unknown and requires a guess.

We demonstrate the above idea by a simple algebraic equation:

$$F(x) = x - 1 = 0, \quad (6)$$

which has the solution  $x = 1$ .

From Eqs. (5) and (6) it follows that

$$\dot{y} = \frac{1-v}{1+t}y + v. \quad (7)$$

Suppose that  $y(0) = y_0$ , then the solution of Eq. (7) can be written as

$$y(t) = \frac{y_0 - 1}{(1+t)^{v-1}} + 1 + t. \quad (8)$$

If we choose  $v > 1$ , the above  $y(t)$  approaches  $1+t$  with a power of  $(1+t)^{1-v}$ . At this moment

of convergence, by  $x = y/(1+t)$  we can get the solution  $x = 1$  of Eq. (6). We note that  $x = 1$  is also the asymptotic of the following FOODE:

$$\dot{x} = -\frac{v}{1+t}(x-1) = -\frac{v}{1+t}F(x), \quad (9)$$

where  $\dot{x} = dx/dt$ . The solution of Eq. (9) is

$$x(t) = \frac{x_0 - 1}{(1+t)^v} + 1, \quad (10)$$

where  $x(t=0) = x_0$ . The solution  $x = 1$  is recovered very fast from  $x(t)$  in Eq. (10), when  $v > 0$  is a large number.

Multiplying Eq. (5) by an integrating factor of  $1/(1+t)$  we can obtain

$$\frac{d}{dt} \left( \frac{y}{1+t} \right) = -\frac{v}{1+t} F\left(\frac{y}{1+t}\right). \quad (11)$$

Further using  $y/(1+t) = x$ , leads to

$$\dot{x} = -\frac{v}{1+t} F(x). \quad (12)$$

The roots of  $F(x) = 0$  are fixed points of the above equation. We should stress that the factor  $-v/(1+t)$  before  $F(x)$  is important.

We may employ a forward Euler scheme on Eq. (12) by starting from a chosen initial condition  $x_0$ :

$$x_{k+1} = x_k - \frac{hv}{1+t_k} F(x_k), \quad (13)$$

where  $h$  is a time stepsize and  $x_k = x(t_k)$  is the value of  $x$  at the  $k$ -th discrete time  $t_k$ .

Suppose that  $t_k = k$  is an integer time with a time increment  $h = 1$ , then we have

$$x_{k+1} = x_k - \frac{v}{1+k} F(x_k). \quad (14)$$

This bears certain similarity with the famous Newton method for Eq. (1):

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)}. \quad (15)$$

But it can be seen that when there exists a danger in the Newton method of dividing by a zero

$F'(x_k)$ , the algorithm in Eq. (14) is always workable. If the function  $F$  is bounded, the algorithm (14) guarantees that the solution can be approached.

Now we turn to the discussions of the following algebraic equations:

$$F_i(x_1, \dots, x_n) = 0, \quad i = 1, \dots, n. \quad (16)$$

The Newton method for these equations is given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{B}(\mathbf{x}_k)]^{-1} \mathbf{F}(\mathbf{x}_k), \quad (17)$$

where we use  $\mathbf{x} := (x_1, \dots, x_n)^T$  and  $\mathbf{F} := (F_1, \dots, F_n)^T$  to represent the vectors, and  $\mathbf{B}$  is an  $n \times n$  matrix with its  $ij$ -th component given by  $\partial F_i / \partial x_j$ .

The Newton method has a great advantage that it is quadratically convergent. However, it still has some drawbacks of not being easy to guess the initial point, the computational burden of  $[\mathbf{B}(\mathbf{x}_k)]^{-1}$ , and  $\mathbf{F}$  being required to be differentiable. Some quasi-Newton methods are thus developed to overcome these defects of the Newton method; see the discussions by Broyden (1965), Dennis (1971), Dennis and More (1974, 1977), and Spedicato and Huang (1997).

Davidenko (1953) was the first who developed a new idea of homotopy method to solve Eq. (16) by numerically integrating

$$\dot{\mathbf{x}}(t) = -\mathbf{H}_{\mathbf{x}}^{-1} \mathbf{H}_t(\mathbf{x}, t), \quad (18)$$

$$\mathbf{x}(0) = \mathbf{a}, \quad (19)$$

where  $\mathbf{H}$  is a homotopic vector function, for example,  $\mathbf{H} = (1-t)(\mathbf{x} - \mathbf{a}) + t\mathbf{F}(\mathbf{x})$ , and  $\mathbf{H}_{\mathbf{x}}$  and  $\mathbf{H}_t$  are respectively the partial derivatives of  $\mathbf{H}$  with respect to  $\mathbf{x}$  and  $t$ . This theory is later refined by Kellogg, Li and Yorke (1976), Chow, Mallet-Paret and Yorke (1978), Li and Yorke (1980), and Li (1997). At the same time, Hirsch and Smale (1979) also derived a continuous Newton method governed by the following differential equation:

$$\dot{\mathbf{x}}(t) = -\mathbf{B}^{-1}(\mathbf{x})\mathbf{F}(\mathbf{x}), \quad (20)$$

$$\mathbf{x}(0) = \mathbf{a}. \quad (21)$$

It can be seen that the ODEs in Eqs. (18) and (20) are difficult to calculate, because they all include

an inverse matrix. Below we will develop a new ODEs system, which is equivalent to the original equation (16).

## 2 A fictitious time integration approach

### 2.1 Transformation into an ODEs system

First we propose the following variable transformation:

$$y_i(t) = (1+t)x_i, \quad i = 1, \dots, n, \quad (22)$$

and multiply Eq. (16) by a coefficient  $-v \neq 0$ :

$$0 = -vF_i(x_1, \dots, x_n). \quad (23)$$

Using Eq. (22) we have

$$0 = -vF_i\left(\frac{y_1}{1+t}, \dots, \frac{y_n}{1+t}\right). \quad (24)$$

Recalling that  $\dot{y}_i = x_i$  by Eq. (22), and adding it on both the sides of the above equation we obtain

$$\dot{y}_i = x_i - vF_i\left(\frac{y_1}{1+t}, \dots, \frac{y_n}{1+t}\right). \quad (25)$$

Then, by using  $x_i = y_i/(1+t)$ , we can change Eq. (16) into an ODEs system:

$$\dot{y}_i = \frac{y_i}{1+t} - vF_i\left(\frac{y_1}{1+t}, \dots, \frac{y_n}{1+t}\right). \quad (26)$$

Finally, multiplying each equation by the integrating factor  $1/(1+t)$  and using Eq. (22) again we obtain

$$\dot{x}_i = \frac{-v}{1+t}F_i(x_1, \dots, x_n), \quad i = 1, \dots, n. \quad (27)$$

It can be seen that this ODEs system is nonautonomous and is much simpler than those in Eqs. (18) and (20).

Furthermore, in terms of a logarithmic time scale

$$\tau = \ln(1+t), \quad (28)$$

Eq. (27) can be recast into a more elegant form:

$$\frac{dx_i}{d\tau} = -vF_i(x_1, \dots, x_n), \quad i = 1, \dots, n. \quad (29)$$

The above idea was first proposed by Liu (2008b) to treat an inverse Sturm-Liouville problem by

transforming an ODE into a PDE. Then, Liu and his coworkers [Liu (2008c, 2008d); Liu, Chang, Chang and Chen (2008)] extended this idea to develop new method for estimating parameters in the inverse vibration problems.

Eq. (22) is not the unique way to transform the algebraic equations (16) into the ODEs. We can adopt

$$y_i(t) = q(t)x_i, \quad i = 1, \dots, n, \quad (30)$$

and a similar derivation leads to

$$\dot{x}_i = \frac{-v}{q(t)} F_i(x_1, \dots, x_n), \quad i = 1, \dots, n. \quad (31)$$

The requirements on  $q(t)$  are that they be differentiable and  $q(0) = 1$ . A special case is  $q(t) = 1$  and  $v = -1$ , and then we have

$$\dot{x}_i = F_i(x_1, \dots, x_n). \quad (32)$$

Deuffhard (2004) has called the above equation a pseudo-transient continuation method. However, this equation is hard to work and usually leads to wrong solutions of  $F_i = 0$ .

From Eq. (29) we can understand that the so-called steady state must be considered in the logarithmic time scale  $\tau = \ln(1+t)$ , because this equation is no more a nonautonomous one as Eq. (27) is. In the logarithmic time scale, if the motion of  $x_i$  approaches a steady state, i.e.,  $dx_i/d\tau = 0$ , then the roots are reached. In this paper we focus on using Eq. (27) as our tool to compute the roots of algebraic equations. This is the most simple choice of  $q(t) = 1+t$  to meet the just mentioned requirements of  $q(t)$ . However, other choices are possible if they can provide better behavior than the present one.

## 2.2 GPS for differential equations system

As was done in Eq. (14), we may employ the Euler method for Eq. (27), and using  $h = 1$ , to obtain an iterative method to calculate the solutions of algebraic equations:

$$x_i^{k+1} = x_i^k - \frac{v}{1+k} F_i(x_1^k, \dots, x_n^k), \quad i = 1, \dots, n. \quad (33)$$

However, we find that this method is not so good, because sometimes  $h = 1$  may be too large to cause over-flow of the values of  $x_i$ .

Therefore we develop a more stable group preserving scheme (GPS) given as follows. Upon letting  $\mathbf{x} = (x_1, \dots, x_n)^T$ , and letting  $\mathbf{f}$  denote a vector with its  $i$ -th component being the right-hand side of Eq. (27) we can write Eq. (27) a vector form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) = \frac{-v}{1+t} \mathbf{F}(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n, \quad t > 0, \quad (34)$$

where  $n$  is the number of algebraic equations.

A GPS can preserve the internal symmetry group of the considered ODEs system. Although we do not know previously the symmetry group of differential equations system, Liu (2001) has embedded it into an augmented differential equations system, which concerns with not only the evolution of state variables themselves but also the evolution of the magnitude of the state variables vector. We note that

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}} = \sqrt{\mathbf{x} \cdot \mathbf{x}}, \quad (35)$$

where the dot between two  $n$ -dimensional vectors denotes their inner product. Taking the derivatives of both the sides of Eq. (35) with respect to  $t$ , we have

$$\frac{d\|\mathbf{x}\|}{dt} = \frac{\dot{\mathbf{x}}^T \mathbf{x}}{\sqrt{\mathbf{x}^T \mathbf{x}}}. \quad (36)$$

Then, by using Eqs. (34) and (35) we can derive

$$\frac{d\|\mathbf{x}\|}{dt} = \frac{\mathbf{f}^T \mathbf{x}}{\|\mathbf{x}\|}. \quad (37)$$

It is interesting that Eqs. (34) and (37) can be combined together into a simple matrix equation:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \|\mathbf{x}\| \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n \times n} & \frac{\mathbf{f}(\mathbf{x}, t)}{\|\mathbf{x}\|} \\ \frac{\mathbf{f}^T(\mathbf{x}, t)}{\|\mathbf{x}\|} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \|\mathbf{x}\| \end{bmatrix}. \quad (38)$$

It is obvious that the first row in Eq. (38) is the same as the original equation (34), but the inclusion of the second row in Eq. (38) gives us a Minkowskian structure of the augmented state variables of  $\mathbf{X} := (\mathbf{x}^T, \|\mathbf{x}\|)^T$ , which satisfies the cone condition:

$$\mathbf{X}^T \mathbf{g} \mathbf{X} = 0, \quad (39)$$

where

$$\mathbf{g} = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & -1 \end{bmatrix} \quad (40)$$

is a Minkowski metric, and  $\mathbf{I}_n$  is the identity matrix of order  $n$ . In terms of  $(\mathbf{x}, \|\mathbf{x}\|)$ , Eq. (39) becomes

$$\mathbf{X}^T \mathbf{g} \mathbf{X} = \mathbf{x} \cdot \mathbf{x} - \|\mathbf{x}\|^2 = \|\mathbf{x}\|^2 - \|\mathbf{x}\|^2 = 0. \quad (41)$$

It follows from the definition given in Eq. (35), and thus Eq. (39) is a natural result.

Consequently, we have an  $n+1$ -dimensional augmented differential equations system:

$$\dot{\mathbf{X}} = \mathbf{A} \mathbf{X} \quad (42)$$

with a constraint (39), where

$$\mathbf{A} := \begin{bmatrix} \mathbf{0}_{n \times n} & \frac{\mathbf{f}(\mathbf{x}, t)}{\|\mathbf{x}\|} \\ \frac{\mathbf{f}^T(\mathbf{x}, t)}{\|\mathbf{x}\|} & 0 \end{bmatrix}, \quad (43)$$

satisfying

$$\mathbf{A}^T \mathbf{g} + \mathbf{g} \mathbf{A} = \mathbf{0}, \quad (44)$$

is a Lie algebra  $so(n, 1)$  of the proper orthochronous Lorentz group  $SO_o(n, 1)$ . This fact prompts us to devise the GPS, whose discretized mapping  $\mathbf{G}$  must exactly preserve the following properties:

$$\mathbf{G}^T \mathbf{g} \mathbf{G} = \mathbf{g}, \quad (45)$$

$$\det \mathbf{G} = 1, \quad (46)$$

$$G_0^0 > 0, \quad (47)$$

where  $G_0^0$  is the 00-th component of  $\mathbf{G}$ .

Although the dimension of the new system is raised by one more, it has been shown that the new system permits a GPS given as follows [Liu (2001)]:

$$\mathbf{X}_{k+1} = \mathbf{G}(k) \mathbf{X}_k, \quad (48)$$

where  $\mathbf{X}_k$  denotes the numerical value of  $\mathbf{X}$  at  $t_k$ , and  $\mathbf{G}(k) \in SO_o(n, 1)$  is the group value of  $\mathbf{G}$  at  $t_k$ . If  $\mathbf{G}(k)$  satisfies the properties in Eqs. (45)-(47), then  $\mathbf{X}_k$  satisfies the cone condition in Eq. (39).

The Lie group can be generated from  $\mathbf{A} \in so(n, 1)$  by an exponential mapping,

$$\begin{aligned} \mathbf{G}(k) &= \exp[h\mathbf{A}(k)] \\ &= \begin{bmatrix} \mathbf{I}_n + \frac{(a_k-1)}{\|\mathbf{f}_k\|^2} \mathbf{f}_k \mathbf{f}_k^T & \frac{b_k \mathbf{f}_k}{\|\mathbf{f}_k\|} \\ \frac{b_k \mathbf{f}_k^T}{\|\mathbf{f}_k\|} & a_k \end{bmatrix}, \end{aligned} \quad (49)$$

where

$$a_k := \cosh\left(\frac{h\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right), \quad (50)$$

$$b_k := \sinh\left(\frac{h\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right). \quad (51)$$

Substituting Eq. (49) for  $\mathbf{G}(k)$  into Eq. (48), we obtain

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{f}_k, \quad (52)$$

$$\|\mathbf{x}_{k+1}\| = a_k \|\mathbf{x}_k\| + \frac{b_k}{\|\mathbf{f}_k\|} \mathbf{f}_k \cdot \mathbf{x}_k, \quad (53)$$

where

$$\eta_k := \frac{b_k \|\mathbf{x}_k\| \|\mathbf{f}_k\| + (a_k - 1) \mathbf{f}_k \cdot \mathbf{x}_k}{\|\mathbf{f}_k\|^2}. \quad (54)$$

The group properties are preserved in this scheme for all  $h > 0$ , and is called a group-preserving scheme.

### 2.3 Runge-Kutta method

We have derived a GPS in the last section, which is however a first-order numerical integration scheme. In order to increase the accuracy in the integration of Eq. (34) sometimes we may employ the fourth-order Runge-Kutta method (RK4) by the following process:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{h}{6} (\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4), \quad (55)$$

where

$$\mathbf{f}_1 = \mathbf{f}(\mathbf{x}_k, t_k),$$

$$\mathbf{f}_2 = \mathbf{f}(\mathbf{x}_k + h/2\mathbf{f}_1, t_k + h/2),$$

$$\mathbf{f}_3 = \mathbf{f}(\mathbf{x}_k + h/2\mathbf{f}_2, t_k + h/2),$$

$$\mathbf{f}_4 = \mathbf{f}(\mathbf{x}_k + h\mathbf{f}_3, t_k + h).$$

## 2.4 Numerical procedure

Starting from an initial value of  $\mathbf{x}(0)$  which can be guessed in a rather free way, we employ the above GPS or RK4 to integrate Eq. (34) from  $t = 0$  to a selected final time  $t_f$ . In the numerical integration process we check the convergence of  $x_i$  at the  $k$ - and  $k + 1$ -steps by

$$\sum_{i=1}^n (x_i^{k+1} - x_i^k)^2 \leq \varepsilon^2, \quad (56)$$

where  $\varepsilon$  is a given convergent criterion. If at a time  $t_0 \leq t_f$  the above criterion is satisfied, then the solution of  $x_i$  is obtained. In practice, if a suitable  $t_f$  is selected we find that the numerical solution is also approached very well to the true solution, even the above convergent criterion is not satisfied before the time  $t < t_f$ . A suitable coefficient  $\nu$  introduced in Eq. (27) can increase the stability of numerical integration, and speeds up the rate of convergence.

In particular we would emphasize that the present method is a new fictitious time integration method (FTIM), which can calculate the solution very stably and effectively. Below we give numerical examples to display some advantages of the present FTIM.

## 3 Numerical tests of FTIM by examples

In this section we will apply the new method of FTIM on both single, as well as multiple nonlinear algebraic equations, and sometimes compare it with the Newton method (NM).

Before embarking the numerical tests we use the following case to compare the FTIM and continuous Newton method:

$$x^2 - 1 = 0. \quad (57)$$

Of course, it has two roots  $x = -1$  and  $x = 1$ .

From Eq. (20) we have

$$\dot{x} = -\frac{x^2 - 1}{2x}. \quad (58)$$

Similarly, from Eq. (12) we have

$$\dot{x} = -\frac{\nu(x^2 - 1)}{1 + t}. \quad (59)$$

While the integral of the first equation (58) leads to

$$x^2 = 1 + (x_0^2 - 1)e^{-t}, \quad (60)$$

the integral of the second equation (59) leads to

$$\frac{x - 1}{x + 1} = \frac{x_0 - 1}{x_0 + 1}(1 + t)^{-2\nu}, \quad (61)$$

where  $x_0$  is an initial condition.

It is obvious that Eq. (60) quickly approaches  $x^2 = 1$  when  $t$  increases. However, because  $x^2$  is not a monotonic function, it cannot take the inverse of Eq. (60) to find  $x$ . Therefore, by applying a numerical integration method on Eq. (58) we need to decide which one of

$$x = \pm \sqrt{1 + (x_0^2 - 1)e^{-t}} \quad (62)$$

is selected.

Conversely, from Eq. (61) we have

$$x \rightarrow 1, \quad t \rightarrow \infty, \quad \text{if } \nu > 0, \quad (63)$$

$$x \rightarrow -1, \quad t \rightarrow \infty, \quad \text{if } \nu < 0. \quad (64)$$

The above convergence speed is dependent on the value of  $\nu$ . If we choose  $\nu > 0$  the FTIM will lead to a unique solution  $x = 1$ , no matter what  $x_0$  is selected; on the other hand, if we choose  $\nu < 0$  the FTIM will lead to a unique solution  $x = -1$ , no matter what  $x_0$  is selected. The convergence speed of FTIM is  $2\nu$  power of  $t$ , which is slower than the exponential convergence of Eq. (62), but its advantage is that we have a unique solution:  $x = 1$  if  $\nu > 0$ , and  $x = -1$  if  $\nu < 0$ . But the continuous NM cannot find these two solutions.

### 3.1 Example 1

We first consider a simple algebraic equation:

$$F(x) = x^3 - 3x^2 + 2x = 0. \quad (65)$$

The roots are 0, 1 and 2.

First we investigate the behavior in first 20 steps by tracing the paths in the plane of  $(x_k, x_{k+1})$ . As shown in Fig. 1(a), starting from  $x = -0.5$  the NM tends to the first root of  $x = 0$  very fast, while the FTIM with  $\nu = 1.5$  tends to the third root  $x = 2$

with an approximation value of 1.9996. It is interesting that when  $\nu = 1.6$  the FTIM tends to the second root of  $x = 1$  exactly, and when  $\nu = 1.7$  the FTIM tends to the first root of  $x = 0$  with a numerical value of  $-4.1 \times 10^{-7}$ .

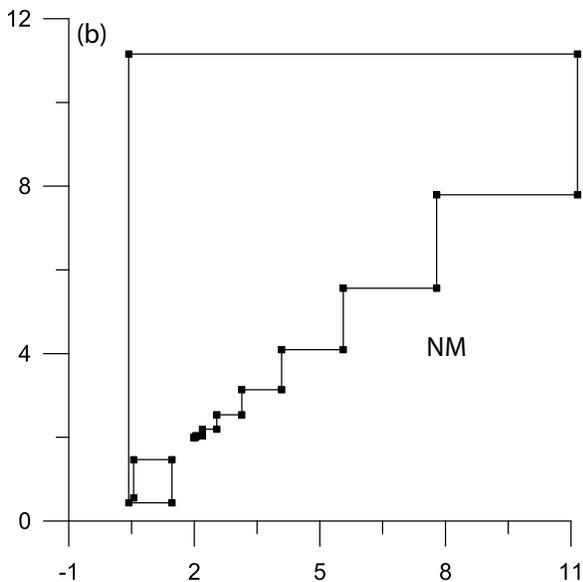
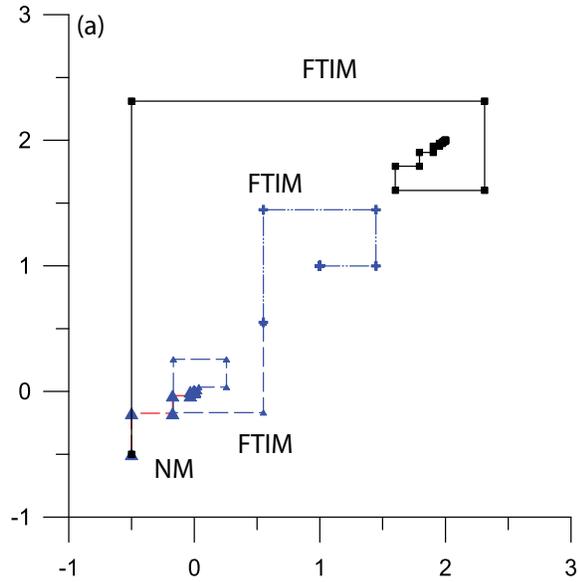


Figure 1: Comparing the iteration paths of Example 1 by FTIM and NM.

Next we start from  $x = 0.55$ . When the FTIM with  $\nu = -2.5$  tends to  $x = 1$  with a high accuracy of 0.999999992 and with  $\nu = 2$  tends to  $x = 0$  with a value of  $3.1 \times 10^{-7}$  as shown in Fig. 1(a), the NM

goes through a large range in the plane as shown separately in Fig. 1(b), and then tends to the third root of  $x = 2$ .

The above demonstration indicates that the solution by FTIM with a suitable choice of  $\nu$  can be very accurate even only through a few iterations. Between two roots with a same starting point the FTIM under different sign of  $\nu$  tend to different roots almost exactly.

### 3.2 Example 2

Then we consider a system of two algebraic equations in two-variables [Hirsch and Smale (1979)]:

$$F_1(x,y) = x^3 - 3xy^2 + a_1(2x^2 + xy) + b_1y^2 + c_1x + a_2y = 0, \quad (66)$$

$$F_2(x,y) = 3x^2y - y^3 - a_1(4xy - y^2) + b_2x^2 + c_2 = 0. \quad (67)$$

The parameters used in this test are listed in Table 1. For these problems the initial guesses are respectively  $(x,y) = (5,5)$ ,  $(x,y) = (0.25,0.1)$  and  $(x,y) = (-1,-1)$ .

For problem 1 there are other solutions given by  $(x,y) = (50.46504, -37.2634179)$ , and  $(x,y) = (36.045402, 36.80750808)$ . For the former solution the parameters we use are given by  $(\nu, h, \epsilon) = (0.1, 0.0001, 10^{-10})$ , and the initial point is  $(50, -30)$ . Through 1341 iterations the result is obtained. For the latter solution the parameters are given by  $(\nu, h, \epsilon) = (0.01, 0.01, 10^{-10})$ , and the initial point is  $(40, 20)$ . Through 1474 iterations, the result is obtained.

For a vision of the convergent paths we also plotted the orbits of  $(x,y)$  for the above three problems in Figs. 2(a), 2(b) and 2(c), respectively. In Fig. 2(a) the left-side corresponds to the first root, while the right-side is for the second root. It can be seen that the first fixed point is a node, while the second one is a focus; similarly, the third fixed point is a focus. There appears a zig-zag of the path for problem 2; however, it spends only 52 iterations to reach a highly accurate solution of the

	Problem 1	Problem 2	Problem 3
$(a_1, b_1, c_1, a_2, b_2, c_2)$	$(25, 1, 2, 3, 4, 5)$	$(25, -1, -2, -3, -4, -5)$	$(200, 1, 2, 3, 1, 2)$
$(\nu, h, \epsilon)$	$(0.1, 0.01, 10^{-10})$	$(1, 0.06, 10^{-11})$	$(0.02, 0.0001, 10^{-10})$
No. of Iterations	792	44	1274
$(x, y)$	$(-50.3970755, -0.8042426)$	$(0.134212, 0.811128)$	$(-400.0952897, -0.2000316)$
$(F_1, F_2)$	$(8.45 \times 10^{-7}, 6.67 \times 10^{-9})$	$(-7.77 \times 10^{-11}, -6.07 \times 10^{-10})$	$(4.26 \times 10^{-5}, 1.06 \times 10^{-8})$

Table 1: The parameters and results for Example 2

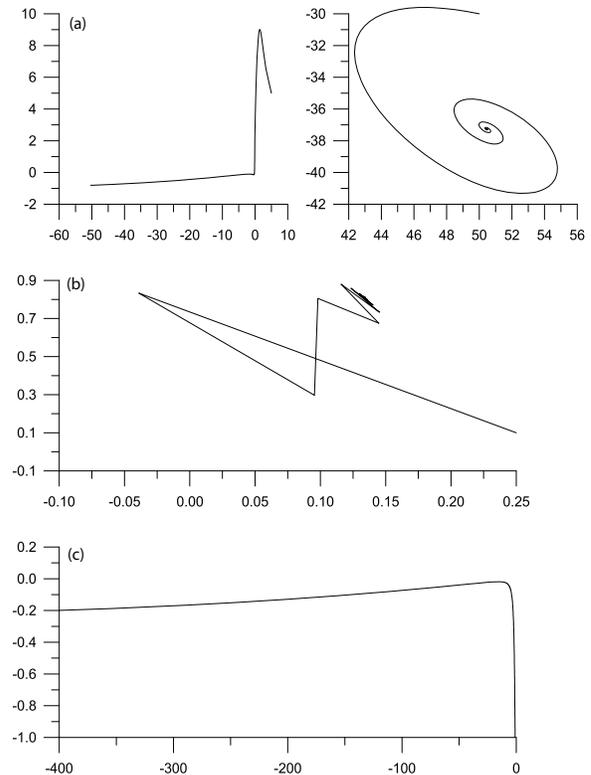


Figure 2: Two-dimensional solution orbits of Example 2 for three different problems.

roots with errors in the order of  $10^{-10}$ . The third problem is hard to solve because there appears a much large coefficient  $a_1 = 200$  than others. As reported by Hsu (1988), he could not calculate the third problem by using the homotopic algorithm with a Gordon-Shampine integrator, the Li-Yorke algorithm with Euler predictor and Newton corrector, and the Li-Yorke algorithm with Euler predictor and quasi-Newton corrector.

Hirsch and Smale (1979) used the continuous Newton algorithm to calculate the above three problems. For the first problem they obtained  $(x, y) = (36.0454, 36.8056)$ . However, inserting it into  $F_1$  and  $F_2$  we find that  $(F_1, F_2) = (13.315, 3.675)$ , which indicates that the result obtained by Hirsch and Smale (1979) is not an accurate root of Eqs. (66) and (67). For the second problem Hirsch and Smale (1979) obtained  $(x, y) = (39.0207, 38.2417)$ . Inserting it into  $F_1$  and  $F_2$  we find that  $(F_1, F_2) = (-0.339, -0.117)$ , which indicates that the result

obtained by Hirsch and Smale (1979) is not accurate. Finally, we check the third problem, of which Hirsch and Smale (1979) obtained  $(x, y) = (0.5115, 197.936)$ . Inserting it into  $F_1$  and  $F_2$  we find that  $(F_1, F_2) = (7.477, 26.964)$ , which indicates again that the result obtained by Hirsch and Smale (1979) is not an accurate root of Eqs. (66) and (67).

For problem 1 we have found three roots as shown above. It is interesting to investigate the attracting set of each fixed point in the plane of initial conditions of  $(x(0), y(0))$ . Starting from any initial condition in the domain of  $-60 < x(0) < 60$ ,  $-40 < y(0) < 40$  we apply the FTIM under a convergent criterion of  $\varepsilon = 10^{-7}$ , and with  $\nu = 0.01$  and  $h = 0.001$  to find its terminal location, and determines which attracting set it belongs by a small disk with a center on each fixed point. The most points in the left-side of Fig. 3 as shown by the diamonds are attracted by the fixed point  $(-50.3971, -0.8042)$ . At the right-side there are two fixed points  $(50.465, -37.263)$  and  $(36.045, 36.808)$ . The solid circular points as shown in Fig. 3 are the attracting set of the former one, while the square points are the attracting set of the latter one. This result reveals that the dynamics of the FTIM is rather simple, and it is convenient for us to choose suitable initial conditions to find the different roots. As we know the dynamics of Newton method is very complex, which usually makes the selection of initial condition being sensitive and difficult.

### 3.3 Example 3

Then we study the following system of two algebraic equations [Spedicato and Hunag (1997)]:

$$F_1(x, y) = x - y^2 = 0, \tag{68}$$

$$F_2(x, y) = (y - 1)^2(y - 2)^2 + (x - y^2)^2 = 0. \tag{69}$$

The two real roots are  $(x, y) = (1, 1)$  and  $(x, y) = (4, 2)$ .

In this test of the FTIM we study the attracting sets of these two fixed points. All nodes of a regular grid of 50 by 50 points with side length 5 in the region of  $[0, 5] \times [0, 5]$  are classified according to which fixed point is tended. In Fig. 4 the

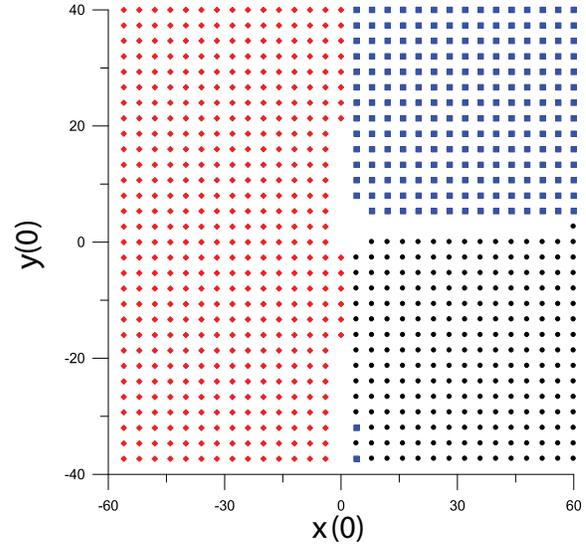


Figure 3: The attracting sets for three different fixed points of problem 1 in Example 2.

square points are those attracted by the the fixed point  $(x, y) = (1, 1)$ , while the triangular points are those attracted by the the fixed point  $(x, y) = (4, 2)$ . Under the convergent criterion  $\varepsilon = 10^{-4}$ , some points in the blank part of Fig. 4 are not convergent to any of the above fixed points.

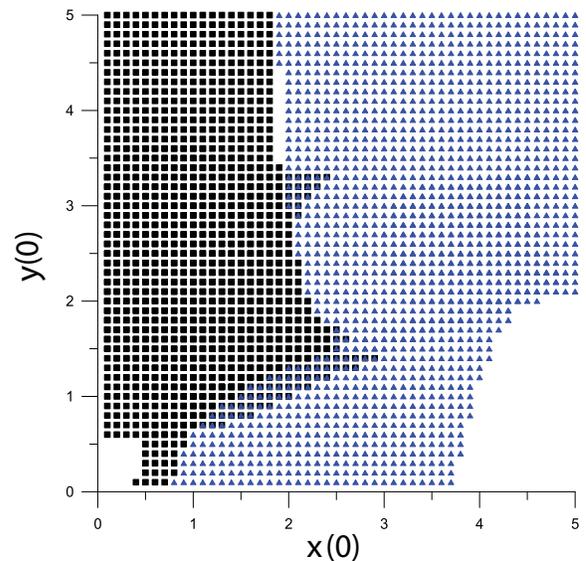


Figure 4: The attracting sets for two different fixed points of Example 3.

### 3.4 Example 4

We consider a system of three algebraic equations in three-variables:

$$F_1(x, y, z) = x + y + z - 3 = 0, \quad (70)$$

$$F_2(x, y, z) = xy + 2y^2 + 4z^2 - 7 = 0, \quad (71)$$

$$F_3(x, y, z) = x^8 + y^4 + z^9 - 3 = 0. \quad (72)$$

Obviously  $x = y = z = 1$  is the solution.

For this case we use a large  $\nu = 10$  to speed up the rate of convergence. In order to increase the accuracy we employ the RK4 method by using a small time stepsize  $h = 0.01$ , and a stringent convergent criterion with  $\varepsilon = 10^{-9}$  is used. Starting from an initial value of  $(x, y, z) = (0.5, 0.6, 0.6)$ , through 1264 iterations the orbit converges to the solution  $(x, y, z) = (1.000000037, 1.00000004, 0.999999955)$ , which is very near the true solution. The present method converges much faster than the above mentioned homotopic methods with the computational time smaller than 0.1 sec by using a PC586.

Because we do not use the steady-state concept in our formulation of the ODEs in Eq. (34), the final time spent by the present approach is not too long. For example, in this case the final time is  $t_f = 1264 \times 0.01 = 12.64$ .

### 3.5 Example 5

Now we consider a test example given by Roose, Kulla, Lomb and Meressoo (1990):

$$F_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + \frac{1}{4}(x_{i+1} - x_{i-1})^2, \quad (73)$$

$$x_0 = 0, \quad x_{n+1} = 20. \quad (74)$$

Initial value is fixed to be  $x_i = 10$ ,  $i = 1, \dots, n$  as that used by Spedicato and Huang (1997).

For this case we use a large  $\nu = -100$  to speed up the rate of convergence, which needs 2381 iterations with a time stepsize  $h = 0.0002$  for the RK4 method. When the convergent criterion is given by  $10^{-15}$ , the residual error  $(\sum_{i=1}^n F_i^2)^{1/2}$  of numerical solution is about  $1.72 \times 10^{-13}$ . This is equivalent to spending a time of  $2381 \times 0.0002 = 0.4762$  that the dynamical system of Eq. (34)

reaches the fixed point, which is recorded in Table 2.

As compared with the methods reported by Spedicato and Huang (1997) for the Newton-like methods, the present approach is more accurate and time saving, where the computational time is smaller than 0.1 sec by using a PC586. For  $n = 50$  the numerical solutions are plotted in Fig. 5(a), where the error of each  $F_k$  is shown in Fig. 5(b). The residual error is about  $5.83 \times 10^{-12}$ .

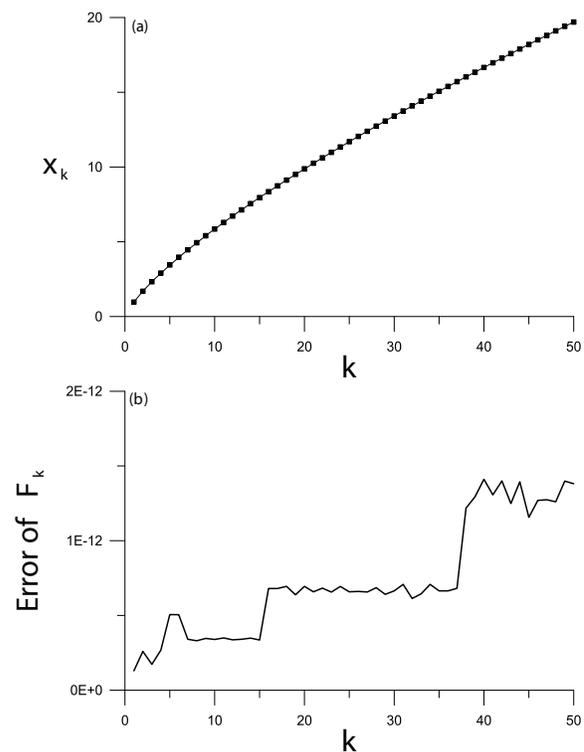


Figure 5: For Example 5 with  $n = 50$  the numerical solutions are plotted.

### 3.6 Example 6

In this example we apply the FTIM to solve the following boundary value problem [Liu (2006)]:

$$u'' = \frac{3}{2}u^2, \quad (75)$$

$$u(0) = 4, \quad u(1) = 1. \quad (76)$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \quad (77)$$

Table 2: The numerical solutions of Example 5 with  $n = 10$

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
3.083	5.383	7.395	9.240	10.969	12.612	14.186	15.705	17.176	18.606

By introducing a finite difference discretization of  $u$  at the grid points we can obtain

$$F_i = \frac{1}{(\Delta x)^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2, \quad (78)$$

$$u_0 = 4, \quad u_{n+1} = 1, \quad (79)$$

where  $\Delta x = 1/(n + 1)$  is the grid length.

Under the following parameters  $n = 49$ ,  $h = 0.001$ ,  $\nu = -0.3$  and  $\varepsilon = 10^{-10}$  we compute the roots of the above system, and compare them with the exact solutions in Fig. 6(a), which can be seen that the error as shown in Fig. 6(b) is very small in the order of  $10^{-4}$ .

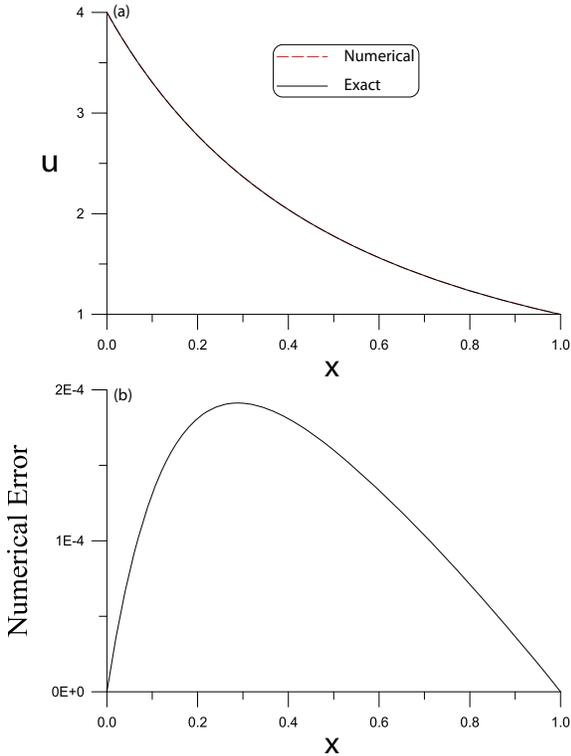


Figure 6: Applying the FTIM to a boundary value problem: (a) comparing numerical and exact solutions, and (b) displaying the numerical error.

### 3.7 Example 7

In this example we apply the FTIM to solve the following boundary value problem of nonlinear elliptic equation [Atluri and Zhu (1998a, 1998b); Zhu, Zhang and Atluri (1998, 1999)]:

$$\Delta u(x, y) + \omega^2 u(x, y) + \varepsilon u^3(x, y) = p(x, y). \quad (80)$$

The exact solution is

$$u(x, y) = \frac{-5}{6}(x^3 + y^3) + 3(x^2y + xy^2). \quad (81)$$

The exact  $p$  can be obtained by inserting the above  $u$  into Eq. (80).

By introducing a finite difference discretization of  $u$  at the grid points we can obtain

$$F_{i,j} = \frac{1}{(\Delta x)^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{(\Delta y)^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \omega^2 u_{i,j} + \varepsilon u_{i,j}^3 - p_{i,j}, \quad (82)$$

where  $\Delta x = 1/(n + 1)$  and  $\Delta y = 1/(n + 1)$  are grid lengths. The boundary constraints can be obtained from the exact solution in Eq. (81).

Under the following parameters  $n = 29 \times 29$ ,  $h = 0.0005$ ,  $\nu = -2$ ,  $\varepsilon = 10^{-5}$ ,  $\omega = 1$  and  $\varepsilon = 0.001$  we compute the roots of the above system, and compare them with the exact solutions. Starting from an initial value of  $u_{i,j} = -0.1$ , the FTIM converges within 5488 steps. At the point  $y_0 = 0.75$  the error of  $u$  was plotted with respect to  $x$  in Fig. 7 by the dashed line, of which the maximum error is about  $5.4 \times 10^{-6}$ . At the point  $x_0 = 0.5$  the error of  $u$  was plotted with respect to  $y$  in Fig. 7 by the solid line, of which the maximum error is about  $4.4 \times 10^{-6}$ . For this highly nonlinear problem the FTIM is effective and gives very small error of the numerical solution.

## 4 Conclusions

Since the work of Newton, iterative algorithms were developed by many researchers, extending

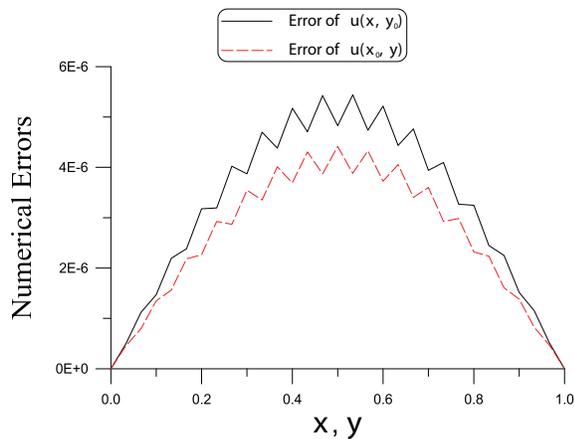


Figure 7: Applying the FTIM to a nonlinear elliptic boundary value problem, the errors are very small.

to continuous type by introducing an extra ad hoc artificial time. However, those ODEs are not intimately related to the original algebraic equations. The present paper very simply transforms the original nonlinear algebraic equations into an evolutionary system of equations by introducing a fictitious time, and had adding a coefficient to enhance the stability of numerical integration of the resulting ODEs and to speed up the convergence to the true roots. The main idea presented here is that the resulting system of ODEs is mathematically equivalent to the original equations, and no approximation is made. Hence, the present FTIM can work very effectively and accurately for the solution of nonlinear algebraic equations. Because no inverse of a matrix is required, the present method is very time efficient. Seven numerical examples were worked out, including the analysis of attracting sets and convergent paths. Some are compared with exact solutions revealing that high accuracy can be achieved by the FTIM. The new method is also applicable to the solutions of boundary value problems of elliptic type equation by discretizing them into high-dimensional nonlinear algebraic equations, revealing a high performance than other solvers.

## References

- Allgower, E. L.; Georg, K.** (1990): Numerical Continuation Methods: an Introduction. Springer, New York.
- Atluri, S. N.** (2002): Methods of Computer Modeling in Engineering and Sciences. Tech. Science Press, 1400 pages.
- Atluri, S. N.; Liu, H. T.; Han, Z. D.** (2006): Meshless local Petrov-Galerkin (MLPG) mixed collocation method for elasticity problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 14, pp. 141-152.
- Atluri, S. N.; Shen, S.** (2002): The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite and boundary element methods. *CMES: Computer Modeling in Engineering & Sciences*, vol. 3, pp. 11-51.
- Atluri, S. N.; Zhu, T. L.** (1998a): A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Comp. Mech.*, vol. 22, pp. 117-127.
- Atluri, S. N.; Zhu, T. L.** (1998b): A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation. *Comp. Model. Simul. Eng.*, vol. 3, pp. 187-196.
- Broyden, C. G.** (1965): A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, vol. 19, pp. 577-593.
- Chow, S. N.; Mallet-Paret, J.; Yorke, J. A.** (1978): Finding zeroes of maps: homotopy methods that are constructive with probability one. *Math. Comp.*, vol. 32, pp. 887-889.
- Davidenko, D.** (1953): On a new method of numerically integrating a system of nonlinear equations. *Doklady Akad. Nauk SSSR*, vol. 88, pp. 601-604.
- Dennis, J. E.** (1971): On the convergence of Broyden's method for nonlinear systems of equations. A class of methods for solving nonlinear simultaneous equations. *Math. Comp.*, vol. 25, pp. 559-567.
- Dennis, J. E.; More, J. J.** (1974): A characterization of superlinear convergence and its application to quasi-Newton method. *Math. Comp.*, vol. 28,

pp. 549-560.

**Dennis, J. E.; More, J. J.** (1977): Quasi-Newton methods, motivation and theory. *SIAM Rev.*, vol. 19, pp. 46-89.

**Deuffhard, P.** (2004): *Newton Methods for Non-linear Problems: Affine Invariance and Adaptive Algorithms*. Springer, New York.

**Hirsch, M.; Smale, S.** (1979): On algorithms for solving  $f(x) = 0$ . *Commun. Pure Appl. Math.*, vol. 32, pp. 281-312.

**Hsu, S. B.** (1988): *The Numerical Methods for Nonlinear Simultaneous Equations*. Central Book Publisher, Taipei, Taiwan.

**Kellogg, R. B. T.; Li, T. Y.; Yorke, J. A.** (1976): A constructive proof of the Brouwer fixed-point theorem and computational results. *SIAM J. Num. Anal.*, vol. 13, pp. 473-483.

**Li, T. Y.; Yorke, J. A.** (1980): A simple reliable numerical algorithm for following homotopy paths. In *Analysis and Computation of Fixed Points*, Robinson, S. M. ed., pp. 73-91, Academic Press, New York.

**Li, T. Y.** (1997): Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica*, vol. 6, pp. 399-436.

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

**Liu, C.-S.** (2006): The Lie-group shooting method for nonlinear two-point boundary value problems exhibiting multiple solutions. *CMES: Computer Modeling in Engineering & Sciences*, vol. 13, pp. 149-163.

**Liu, C.-S.** (2007a): A modified Trefftz method for two-dimensional Laplace equation considering the domain's characteristic length. *CMES: Computer Modeling in Engineering & Sciences*, vol. 21, pp. 53-66.

**Liu, C.-S.** (2007b): A highly accurate solver for the mixed-boundary potential problem and singular problem in arbitrary plane domain. *CMES: Computer Modeling in Engineering & Sciences*, vol. 20, pp. 111-122.

**Liu, C.-S.** (2007c): An effectively modified direct Trefftz method for 2D potential problems considering the domain's characteristic length. *Eng. Anal. Bound. Elem.*, vol. 31, pp. 983-993.

**Liu, C.-S.** (2008a): A highly accurate collocation Trefftz method for solving the Laplace equation in the doubly-connected domains. *Numer. Meth. Partial Diff. Eq.*, vol. 24, pp. 179-192.

**Liu, C.-S.** (2008b): Solving an inverse Sturm-Liouville problem by a Lie-group method. *Boundary Value Problems*, vol. 2008, Article ID 749865.

**Liu, C.-S.** (2008c): Identifying time-dependent damping and stiffness functions by a simple and yet accurate method. *J. Sound Vib.*, vol. 318, pp. 148-165.

**Liu, C.-S.** (2008d): A Lie-group shooting method for simultaneously estimating the time-dependent damping and stiffness coefficients. *CMES: Computer Modeling in Engineering & Sciences*, vol. 27, pp. 137-149.

**Liu, C.-S.; Chang, J. R.; Chang, K. H.; Chen, Y. W.** (2008): Simultaneously estimating the time-dependent damping and stiffness coefficients with the aid of vibrational data. *CMC: Computers, Materials & Continua*, vol. 7, pp. 97-107.

**Roose, A.; Kulla, V.; Lomb, M.; Meressoo, T.** (1990): *Test examples of systems of non-linear equations*. Estonian Software and Computer Service Company, Tallin 1990.

**Spedicato, E.; Hunag, Z.** (1997): Numerical experience with Newton-like methods for nonlinear algebraic systems. *Computing*, vol. 58, pp. 69-89.

**Zhu, T.; Zhang, J.; Atluri, S. N.** (1998): A meshless local boundary integral equation (LBIE) method for solving nonlinear problems. *Comp. Mech.*, vol. 22, pp. 174-186.

**Zhu, T.; Zhang, J.; Atluri, S. N.** (1999): A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and non-linear boundary value problems. *Eng. Anal. Bound. Elem.*, vol. 23, pp. 375-389.



