Journal of
Marine Science and Technology

Volume 17 | Issue 3 Article 9

A MODIFIED NEWTON METHOD FOR SOLVING NON-LINEAR ALGEBRAIC EQUATIONS

Satya N. Atluri
Center for Aerospace Research & Education, University of California, Irvine.

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

Chung-Lun Kuo
Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.

Recommended Citation
Atluri, Satya N.; Liu, Chein-Shan; and Kuo, Chung-Lun (2009) "A MODIFIED NEWTON METHOD FOR SOLVING NON-LINEAR ALGEBRAIC EQUATIONS," *Journal of Marine Science and Technology*: Vol. 17: Iss. 3, Article 9.
DOI: 10.51400/2709-6998.1960
Available at: https://jmstt.ntou.edu.tw/journal/vol17/iss3/9

A MODIFIED NEWTON METHOD FOR SOLVING NON-LINEAR ALGEBRAIC EQUATIONS

Satya N. Atluri*, Chein-Shan Liu**, and Chung-Lun Kuo***

Key words: nonlinear algebraic equations, iterative method, ordinary differential equations, fictitious time integration method (FTIM), modified Newton method (MNM).

ABSTRACT

The Newton algorithm based on the "continuation" method may be written as being governed by the equation $\dot{x}_j(t) + B_{ij}^{-1}F_i(x_j) = 0$, where $F_i(x_j) = 0$, i, $j = 1, \ldots n$ are nonlinear algebraic equations (NAEs) to be solved, and $B_{ij} = \partial F_i/\partial x_j$ is the corresponding Jacobian matrix. It is known that the Newton's algorithm is quadratically convergent; however, it has some drawbacks, such as being sensitive to the initial guess of solution, and being expensive in the computation of the inverse of B_{ij} at each iterative step. How to preserve the convergence speed, and to remove the drawbacks is a very important issue in the solutions of NAEs. In this paper we discretize the above equation being written as $B_{ij}\dot{x}_j(t) + F_i(x_j) = 0$, by a backward difference scheme in a new time scale of $s = 1 - e^{-t}$, and an ODEs system is derived by introducing a fictitious time-like variable. The new algorithm is obtained by applying a numerical integration scheme to the resultant ODEs. The new algorithm does not need the inverse of B_{ij}, and is thus resulting in a significant reduction in computational time than the Newton's algorithm. A similar technique is also used to modify the homotopy method. Numerical examples given confirm that the modified Newton method is highly efficient, insensitive to the initial condition, to find the solutions with a very small the residual error.

I. INTRODUCTION

The numerical solution of nonlinear algebraic equations is one of the main aspects of computational mathematics. Usually it is hard to solve a large system of highly-nonlinear algebraic equations. Although a lot of priori research has been conducted in this area, we still lack an efficient and reliable algorithm to solve this difficult problem. 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 [48], Atluri and Zhu [8], Atluri [5], Atluri and Shen [7], and Atluri, Liu and Han [6] 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, such as those used by Liu [27-30] 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 simplical or piecewise linear method. The monographs by Allgower and Georg [2] and Deuflhard [18] are devoted to the continuation methods for solving NAEs.

The Newton's method and its improvements are extensively used nowadays; however, those algorithms fail if the initial guess of solution is improper. In general, it is difficult to choose a good initial condition for most large systems of NAEs. Thus, it is necessary to develop an efficient algorithm, which is insensitive to the initial guess of the solution, and which converges fast.

This paper is arranged as follows. In the next section we introduce an evolution from a discretized method to a continuous method, where an artificial time is introduced for writing the NAEs in the form of ODEs. In Section III the main algorithms are introduced, where the novel feature is a suitable combination of the fictitious time integration method (FTIM) with the Newton method. In Section IV we give some numerical examples to evaluate the new algorithms of the modified Newton method (MNM) and the modified homotopy method (MHM). Finally, we draw conclusions in Section V.

Author for correspondence: Chein-Shan Liu (e-mail: liucs@ntu.edu.tw).

**Center for Aerospace Research & Education, University of California, Irvine.*

***Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.*

****Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan, R.O.C.*

II. FROM DISCRETE TO CONTINUOUS METHODS

For the following algebraic equations:

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

the Newton method is given by

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

where we use $\mathbf{x}: = (x_1, \ldots, x_n)^{\mathrm{T}}$ and $\mathbf{F}: = (F_1, \ldots, F_n)^{\mathrm{T}}$ to represent the vectors, and $\mathbf{B}$ is an $n \times n$ Jacobian matrix with its ij-th component given by $B_{ij} = \partial F_i/\partial x_j$. Starting from an initial guess of solution by $\mathbf{x}_0$, Eq. (2) can be used to generate a sequence of $\mathbf{x}_k$, $k = 1, 2, \ldots$. When $\mathbf{x}_k$ are convergent under a specified convergent criterion, the solutions of (1) are obtained.

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, and the computational burden of $[\mathbf{B}(\mathbf{x}_k))]^{-1}$. Some quasi-Newton methods are developed to overcome these defects of the Newton method; see the discussions by Broyden [11], Dennis [15], Dennis and More [16, 17], and Spedicato and Huang [46].

Hirsch and Smale [19] and many others have derived a continuous Newton method governed by the following differential equation:

$$\dot{\mathbf{x}}(t) = -\mathbf{B}^{-1}(\mathbf{x})\mathbf{F}(\mathbf{x}), \tag{3}$$

$$\mathbf{x}(0) = \mathbf{a}, \tag{4}$$

where t is an artificial time, and $\mathbf{a}$ is an initial guess of $\mathbf{x}$. It can be seen that the ODEs in (3) are difficult to calculate, because they include an inverse matrix.

The corresponding dynamics of (3) has been studied by several researchers, such as, Alber [1], Boggs and Dennis [10], Smale [45], Chu [13], Maruster [41], and Ascher, Huang and van den Doel [3]. Presently, this artificial time embedding technique does not bring out any practically useful result pertaining to the Newton's algorithm. Below we will develop a new ODEs system, which is equivalent to (3). Then, a natural embedding technique from the NAEs into the ODEs as developed by Liu and Atluri [37] will be combined with a new continuous form of (3). Corresponding to the artificial embedding technique, which is not yet proven to be useful, our embedding technique by transforming the continuous form in (3) into a space, which is one-dimension higher, may find to be very useful.

III. MODIFIED METHODS

1. A Novel Technique

Liu and Atluri [37] have introduced a novel continuation method, by embedding the NAEs into a system of nonautonomous first-order ODEs. For the later requirement, we consider a single NAE:

$$F(x) = 0. \tag{5}$$

The above equation only has an independent variable x. We may transform it into a first-order ODE by introducing a fictitious time-like variable τ in the following transformation of variable from x to y:

$$y(\tau) = (1+\tau)x. \tag{6}$$

Here, τ is a variable which is independent of x; hence, $y' = dy/d\tau = x$. If $\nu \neq 0$, Eq. (5) is equivalent to

$$0 = -\nu F(x). \tag{7}$$

Adding the equation $y' = x$ to (7) we obtain

$$y' = x - \nu F(x). \tag{8}$$

By using (6) we can derive

$$y' = \frac{y}{1+\tau} - \nu F\left(\frac{y}{1+\tau}\right). \tag{9}$$

This is a first-order ODE for $y(\tau)$. The initial condition for the above equation is $y(0) = x$, which is however an unknown and requires a guess.

Multiplying (9) by an integrating factor of $1/(1 + \tau)$ we can obtain

$$\frac{d}{d\tau}\left(\frac{y}{1+\tau}\right) = -\frac{\nu}{1+\tau}F\left(\frac{y}{1+\tau}\right). \tag{10}$$

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

$$x' = -\frac{\nu}{1+\tau}F(x). \tag{11}$$

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

2. Modified Newton Method

When one applies the forward Euler method to (3) with a time stepsize equal to 1, Eq. (2) is obtained. If a suitable initial condition is chosen, when time increases to a large value, we may expect the sequence $\mathbf{x}_k$ to converge to a true solution. However, the Newton method is very time consuming in the calculation of $\mathbf{B}_k^{-1}$ and is not easy to choose a suitable initial condition.

First, we propose a variable transformation $s = 1 - e^{-t}$ and write (3) as

$$(1-s)\mathbf{B}(\mathbf{x})\mathbf{x}'(s) + \mathbf{F} = \mathbf{0}, \tag{12}$$

where the prime denotes the derivative of $\mathbf{x}(s)$ with respect to s. Now the interval of s is $s \in [0, 1)$, when $t \in [0, \infty)$. We divide the interval of [0, 1) into m subintervals with $\Delta s = 1/m$, and approximate the above equation by a backward finite difference:

$$(1-s_i)\mathbf{B}(x^i)\frac{x^i - x^{i-1}}{\Delta s} + \mathbf{F}(x^i) = \mathbf{0}, \quad i = 1, \ldots, m, \tag{13}$$

$$\mathbf{x}^0 = \mathbf{a}, \tag{14}$$

where $\mathbf{x}^i = \mathbf{x}(s_i)$ with $s_i = i\Delta s$, and now $\mathbf{x}^0 = \mathbf{a}$ is a boundary condition, instead of the initial condition in (3). Again, Eq. (13) is a coupled system of NAEs, with m vectorial-variables $\mathbf{x}^i$, $i = 1, \ldots, m$. When $\mathbf{x}^m$ is solved from (13), the solution of NAEs is found. Now, we can apply the technique in (11) to (13), obtaining

$$\frac{d\mathbf{x}^i}{d\tau} = -\frac{\nu}{1+\tau}\left[(1-s_i)\mathbf{B}(\mathbf{x}^i)\frac{\mathbf{x}^i - \mathbf{x}^{i-1}}{\Delta s} + \mathbf{F}(\mathbf{x}^i)\right], \quad i = 1, \ldots, m. \tag{15}$$

We find that the present formulation is insensitive to the condition of $\mathbf{x}^0 = \mathbf{a}$, because $\mathbf{a}$ is just a boundary value of the many ODEs in (15) with m-unknown vectors; hence, we may set $\mathbf{a} = \mathbf{0}$. It deserves to note that there are two advantages to transforming (3) into (12): first, the domain length of s is 1 such that we can use a small integer m to divide the whole interval into some subintervals by using $\Delta s = 1/m$, and second, we no longer need to use the inverse of $\mathbf{B}$. In (3), because we need to integrate the ODE along the t-direction, the inverse of $\mathbf{B}$ is required; however, in (15) we only integrate the ODEs along the τ-direction, and the inverse of $\mathbf{B}$ is not required any more. Eq. (15) is a new equation, which is a combination of the continuous form of the Newton's algorithm with the fictitious time integration form. We will use this equation to solve the NAEs.

It is interesting that when we take $m = 1$, $\Delta s = 1$, $s_1 = 1$, the following term

$$(1-s_1)\mathbf{B}(x^1)\frac{x^1 - x^0}{\Delta s}$$

drops out, and (15) is reduced to

$$\frac{d\mathbf{x}}{d\tau} = -\frac{\nu}{1+\tau}\mathbf{F}(\mathbf{x}), \tag{16}$$

where we replace $\mathbf{x}^1$ by $\mathbf{x}$. This equation has been used by Liu and Atluri [37] to solve the NAEs, and the new method is called a fictitious time integration method (FTIM). As reported by Liu and Atluri [37], when the technique of FTIM is used to solve a large system of NAEs, high performance can be achieved.

The above idea of introducing a fictitious time coordinate τ into the governing equation was first proposed by Liu [31] to treat an inverse Sturm-Liouville problem by transforming an ODE into a PDE. Then, Liu [32-34], and Liu, Chang, Chang and Chen [40] extended this idea to develop new methods for estimating parameters in the inverse vibration problems. More recently, Liu [35] has used the FTIM technique to solve the nonlinear complementarity problems, whose numerical results are very well. Then, Liu [36] used the FTIM to solve the boundary value problems of elliptic type partial differential equations. Liu and Atluri [38] also employed this technique of FTIM to solve the mixed-complementarity problems and optimization problems. Then, Liu and Atluri [39] using the technique of FTIM solved the inverse Sturm-Liouville problem, for specified eigenvalues.

3. Modified Homotopy Method

Davidenko [14] was the first who developed a new idea of homotopy method to solve (1) by numerically integrating

$$\dot{\mathbf{x}}(t) = -\mathbf{H}_{\mathbf{x}}^{-1}\mathbf{H}_t(\mathbf{x}, t), \tag{17}$$

$$\mathbf{x}(0) = \mathbf{a}, \tag{18}$$

where $\mathbf{H}$ is a homotopic vector function given by

$$\mathbf{H} = (1-t)(\mathbf{x} - \mathbf{a}) + t\mathbf{F}(\mathbf{x}), \tag{19}$$

and $\mathbf{H}_{\mathbf{x}}$ and $\mathbf{H}_t$ are respectively the partial derivatives of $\mathbf{H}$ with respect to $\mathbf{x}$ and t. The solution $\mathbf{x}(t)$ of (17) forms a homotopy path for $0 \le t \le 1$. One then solves a sequence of problems $\mathbf{H}(t) = \mathbf{0}$ for values of t increasing from 0 to 1, where for each such problem a good initial guess from previous steps is at hand. This powerful idea has been around for a while; see Watson, Sosonkina, Melville, Morgan and Walker [47] for a general package, Nocedal and Wright [43] for a discussion in the context of optimization, and Ascher, Mattheij and Russell [4] for boundary value ODEs. The homotopy theory was later refined by Kellogg, Li and Yorke [21], Chow, Mallet-Paret and Yorke [12], Li and Yorke [23], and Li [24]. For some highly complicated NAEs, a continuation approach of the homotopic method may yield the only practical route for a solution algorithm.

With the use of (19), the homotopic ODEs in (17) can be written as

$$\left[s\mathbf{B} + (1-s)\mathbf{I}_n\right]\mathbf{x}'(s) + \mathbf{a} - \mathbf{x} + \mathbf{F} = \mathbf{0}. \tag{20}$$

Here we use s to replace t in order to be consistent with the notation s used in (12).

Similarly, by a discretization of the above equation we can obtain a new algebraic equation:

$$\left[s_i\mathbf{B}(x^i)+(1-s_i)\mathbf{I}_n\right]\frac{x^i-x^{i-1}}{\Delta s}+\mathbf{a}-\mathbf{x}^i+\mathbf{F}(\mathbf{x}^i)=\mathbf{0},$$

$$i=1,\ldots,m, \tag{21}$$

$$\mathbf{x}^0=\mathbf{a}. \tag{22}$$

Again, applying the technique in (11) to (21) we can obtain

$$\frac{d\mathbf{x}^i}{d\tau}=-\frac{\nu}{1+\tau}\left[\left[s_i\mathbf{B}(\mathbf{x}^i)+(1-s_i)\mathbf{I}_n\right]\frac{\mathbf{x}^i-\mathbf{x}^{i-1}}{\Delta s}+\mathbf{a}-\mathbf{x}^i+\mathbf{F}(\mathbf{x}^i)\right],$$

$$i=1,\ldots,m. \tag{23}$$

We will integrate (15) and (23) by using the group preserving scheme introduced in the next section.

4. The GPS for Differential Equations System

We develop a stable group preserving scheme (GPS) as follows. We can write a vector form of ODEs by

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},t),\quad \mathbf{x}\in\mathbb{R}^n,\quad t>0. \tag{24}$$

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 [25] 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

$$\|\mathbf{x}\|=\sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{x}}=\sqrt{\mathbf{x}\cdot\mathbf{x}}, \tag{25}$$

where the dot between two *n*-dimensional vectors denotes their inner product. Taking the derivatives of both the sides of (25) with respect to *t*, we have

$$\frac{d\|\mathbf{x}\|}{dt}=\frac{\dot{\mathbf{x}}^{\mathrm{T}}\mathbf{x}}{\sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{x}}}. \tag{26}$$

Then, by using (24) and (25) we can derive

$$\frac{d\|\mathbf{x}\|}{dt}=\frac{\mathbf{f}^{\mathrm{T}}\mathbf{x}}{\|\mathbf{x}\|}. \tag{27}$$

It is interesting that (24) and (27) can be combined together into a simple matrix equation:

$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}\\ \|\mathbf{x}\|\end{bmatrix}=\begin{bmatrix}\mathbf{0}_{n\times n} & \dfrac{\mathbf{f}(\mathbf{x},t)}{\|\mathbf{x}\|}\\ \dfrac{\mathbf{f}^{\mathrm{T}}(\mathbf{x},t)}{\|\mathbf{x}\|} & 0\end{bmatrix}\begin{bmatrix}\mathbf{x}\\ \|\mathbf{x}\|\end{bmatrix}. \tag{28}$$

It is obvious that the first row in (28) is the same as the original equation (24), but the inclusion of the second row in (28) gives us a Minkowskian structure of the augmented state variables of $\mathbf{X}:=(\mathbf{x}^{\mathrm{T}},\|\mathbf{x}\|)^{\mathrm{T}}$, which satisfies the cone condition:

$$\mathbf{X}^{\mathrm{T}}\mathbf{g}\mathbf{X}=0, \tag{29}$$

where

$$\mathbf{g}=\begin{bmatrix}\mathbf{I}_n & \mathbf{0}_{n\times1}\\ \mathbf{0}_{1\times n} & -1\end{bmatrix} \tag{30}$$

is a Minkowski metric, and $\mathbf{I}_n$ is the identity matrix of order *n*. In terms of $(\mathbf{x},\|\mathbf{x}\|)$, Eq. (29) becomes

$$\mathbf{X}^{\mathrm{T}}\mathbf{g}\mathbf{X}=\mathbf{x}\cdot\mathbf{x}-\|\mathbf{x}\|^2=\|\mathbf{x}\|^2-\|\mathbf{x}\|^2=0. \tag{31}$$

It follows from the definition given in (25), and thus (29) is a natural result.

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

$$\dot{\mathbf{X}}=\mathbf{A}\mathbf{X} \tag{32}$$

with a constraint (29), where

$$\mathbf{A}:=\begin{bmatrix}\mathbf{0}_{n\times n} & \dfrac{\mathbf{f}(\mathbf{x},t)}{\|\mathbf{x}\|}\\ \dfrac{\mathbf{f}^{\mathrm{T}}(\mathbf{x},t)}{\|\mathbf{x}\|} & 0\end{bmatrix}, \tag{33}$$

satisfying

$$\mathbf{A}^{\mathrm{T}}\mathbf{g}+\mathbf{g}\mathbf{A}=\mathbf{0}, \tag{34}$$

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 G must exactly preserve the following properties:

$$\mathbf{G}^{\mathrm{T}}\mathbf{g}\mathbf{G}=\mathbf{g}, \tag{35}$$

Table 1. Comparison of MNM and MHM for Example 1.

Method	m	h	ν	ε	IN	(x, y)	(F_1, F_2)
MNM	5	0.15	2	10^{-5}	79	(1.618031, 1.618031)	$(-6.751 \times 10^{-6}, -6.751 \times 10^{-6})$
MNM	5	0.15	-2	10^{-5}	94	(–0.618031, –0.618031)	$(-6.9408 \times 10^{-6}, -6.9408 \times 10^{-6})$
MHM	14	0.7	0.1	10^{-4}	738	(1.617965, 1.617965)	$(-1.539 \times 10^{-4}, -1.539 \times 10^{-4})$
MHM	10	0.1	0.05	10^{-4}	3102	(–0.618, –0.618)	$(-8.704 \times 10^{-4}, -8.704 \times 10^{-4})$

$$\det \mathbf{G} = 1, \tag{36}$$

$$G_0^0 > 0, \tag{37}$$

where G_0^0 is the 00-th component of **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 [25]:

$$\mathbf{X}_{k+1} = \mathbf{G}(k)\mathbf{X}_k, \tag{38}$$

where $\mathbf{X}_k$ denotes the numerical value of **X** at t_k, and $\mathbf{G}(k) \in SO_o(n, 1)$ is the group value of **G** at t_k. If $\mathbf{G}(k)$ satisfies the properties in (35)-(37), then $\mathbf{X}_k$ satisfies the cone condition in (29).

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

$$\mathbf{G}(k) = \exp\left[h\mathbf{A}(k)\right] = \begin{bmatrix} \mathbf{I}_n + \dfrac{(a_k - 1)}{\|\mathbf{f}_k\|^2}\mathbf{f}_k\mathbf{f}_k^{\mathrm{T}} & \dfrac{b_k\mathbf{f}_k}{\|\mathbf{f}_k\|} \\ \dfrac{b_k\mathbf{f}_k}{\|\mathbf{f}_k\|} & a_k \end{bmatrix}, \tag{39}$$

where

$$a_k := \cosh\left(\frac{h\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right), \tag{40}$$

$$b_k := \sinh\left(\frac{h\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right). \tag{41}$$

Substituting (39) for $\mathbf{G}(k)$ into (38), we obtain

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

$$\|\mathbf{x}_{k+1}\| = a_k\|\mathbf{x}_k\| + \frac{b_k}{\|\mathbf{f}_k\|}\mathbf{f}_k \cdot \mathbf{x}_k, \tag{43}$$

where

$$\eta_k := \frac{b_k\|\mathbf{x}_k\|\|\mathbf{f}_k\| + (a_k - 1)\mathbf{f}_k \cdot \mathbf{x}_k}{\|\mathbf{f}_k\|^2}. \tag{44}$$

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

5. Numerical Procedure

Starting from an initial value of $\mathbf{x}^i(0)$, we may employ the above GPS to integrate (15) or (23) from $\tau = 0$ to a selected final time τ_f. In the numerical integration process we can check the residual norm by

$$\left(\sum_{i=1}^{n}[F_i(x^m)]^2\right)^{1/2} \le \varepsilon, \tag{45}$$

where ε is a given convergent criterion. If at a time $\tau_0 \le \tau_f$ the above criterion is satisfied, then the solution of $\mathbf{x}^i$ is obtained, and thus $\mathbf{x}^m$ gives the solution of (1).

IV. NUMERICAL TESTS

1. Example 1

We first consider two simple algebraic equations:

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

The roots are (–1, 0), (0, –1), $((1+\sqrt{5})/2,\ (1+\sqrt{5})/2)$ and $((1-\sqrt{5})/2,\ (1-\sqrt{5})/2)$.

In the computations by using the modified Newton method (MNM) and modified homotopy method (MHM) we require to specify the values of m, h used in the GPS, ν, ε, and some initial conditions; however, we let $\mathbf{a} = 0$. We calculate this example by MNM and MHM, of which the third root $((1+\sqrt{5})/2, (1+\sqrt{5})/2) \approx$ (1.618034, 1.618034) and the fourth root $((1-\sqrt{5})/2,\ (1-\sqrt{5})/2) \approx$ (–0.618034, –0.618034) are calculated, and the values of these parameters are recorded in Table 1, where IN is a shorthand of the iteration number spent in the calculation.

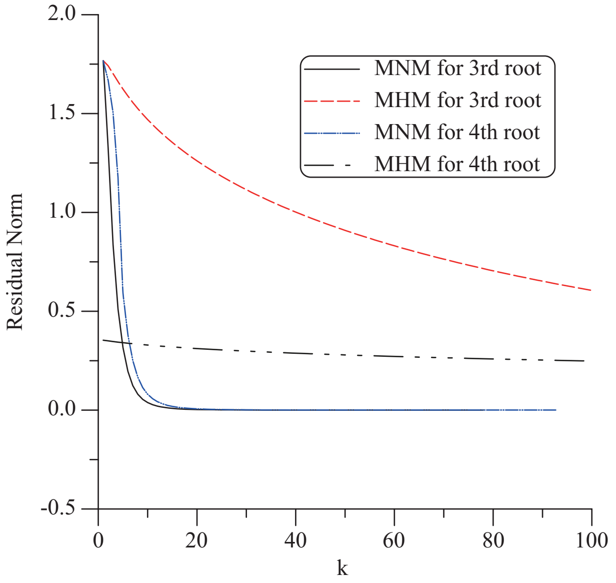

Fig. 1. Comparing the iterative residual norms of Example 1 by MNM and MHM.

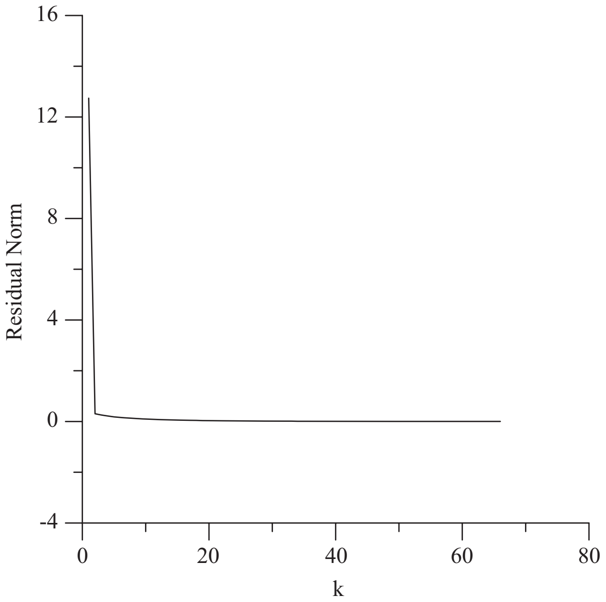

Fig. 2. The residual norm of Example 2.

In Fig. 1 we plot the variation of the residual norms for MNM and MHM with respect to the number of iteration, denoted by *k*. It can be seen that the MNM converges very fast, which is much fast than that of the MHM. It is interesting that the MNM can find two different roots by merely changing the sign of ν and using the same initial guess of $x_1^i = x_2^i = 0.5$. However, for the MHM this is not working, and we use different ν and different initial guess of $x_1^i = x_2^i = 0.5$ for the third root, and $x_1^i = x_2^i = -0.5$ for the fourth root.

2. Example 2

We study the following system of two algebraic equations [46]:

$$F_1(x,y) = x - y^2 = 0,$$
$$F_2(x,y) = (y-1)^2(y-2)^2 + (x-y^2)^2 = 0. \tag{47}$$

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

In this test of the MNM we take *m* = 1, *h* = 0.01, *ν* = 10. As shown in Fig. 2 the residual norm converges very fast with only 67 steps for satisfying the convergent criterion of $\varepsilon = 10^{-3}$. We get the solutions (*x*, *y*) = (1.0013, 1.0011) with the residual errors $(F_1, F_2) = (-9.66 \times 10^{-4}, 2.17 \times 10^{-6})$.

3. Example 3

Then we consider a system of two algebraic equations in two-variables [19]:

$$F_1(x,y) = x^3 - 3xy^2 + a_1(2x^2 + xy) + b_1 y^2 + c_1 x + a_2 y = 0,$$
$$F_2(x,y) = 3x^2 y - y^3 - a_1(4xy - y^2) + b_2 x^2 + c_2 = 0. \tag{48}$$

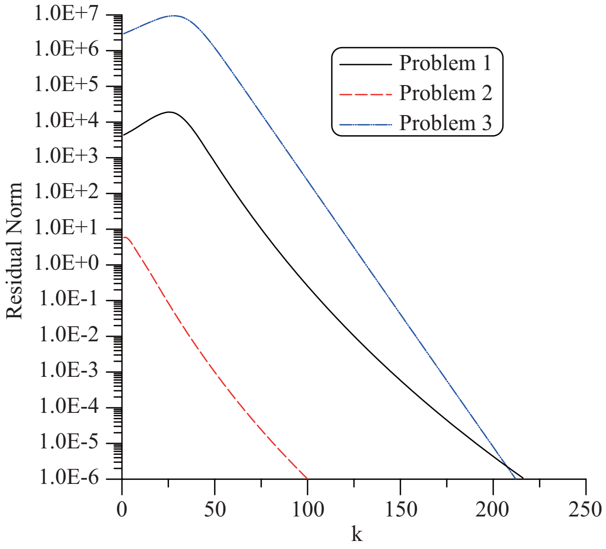

Fig. 3. The residual norms of Example 3.

The parameters used in this test are listed in Table 2. For these problems the initial guesses are respectively (*x*, *y*) = (–10, –1), (*x*, *y*) = (0.1, 0.1), and (*x*, *y*) = (–100, –0.1).

In Fig. 3 we display the residual errors for the above three problems. The third problem is hard to solve because there appears a much large coefficient a_1 than others. As reported by Hsu [20], he could not calculate the third problem by using the homotopic algorithm with a Gordon-Shampine integrator, the Li-Yorke algorithm with the Euler predictor and Newton corrector, and the Li-Yorke algorithm with the Euler predictor and quasi-Newton corrector.

Table 2. The parameters and results for Example 3.

	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)
(m, ν, h, ε)	$(2, 0.01, 0.01, 10^{-6})$	$(5, 1.5, 0.01, 10^{-6})$	$(2, 0.01, 0.0001, 10^{-6})$
IN	217	101	213
(x, y)	(–50.3970755, –0.8042426)	(0.134212, 0.811128)	(–400.0952897, –0.2000316)
(F_1, F_2)	$(9.916 \times 10^{-7}, 7.836 \times 10^{-9})$	$(-1.141 \times 10^{-7}, -8.913 \times 10^{-7})$	$(8.573 \times 10^{-7}, 2.328 \times 10^{-10})$

Hirsch and Smale [19] used the continuous Newton algorithm to calculate the above three problems. However, as point out by Liu and Atluri [37], the results obtained by Hirsch and Smale [19] are not accurate. Under this situation we may say that the present modified Newton method can offer more efficient and accurate solutions; and also the new MNM as compared with the FTIM reported by Liu and Atluri [37] for calculating this example is convergent fast than FTIM and can retain the same accuracy.

4. Example 4

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

$$
\begin{aligned}
&F_1(x, y, z) = x + y + z - 3 = 0,\\
&F_2(x, y, z) = xy + 2y^2 + 4z^2 - 7 = 0, \qquad (49)\\
&F_3(x, y, z) = x^8 + y^4 + z^9 - 3 = 0.
\end{aligned}
$$

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

In this test we take $m = 1$, $h = 0.01$, $\nu = 12$. As shown in Fig. 4 the residual norm converges very fast with 140 steps for satisfying the convergent criterion of $\varepsilon = 10^{-3}$. We get the solutions $(x, y, z) = (0.9996, 0.999, 1.0007)$ with the residual errors $(F_1, F_2, F_3) = (-6.046 \times 10^{-4}, 7.574 \times 10^{-4}, 3.005 \times 10^{-6})$.

5. Example 5

The following example is given by Roose, Kulla, Lomb and Meressoo [44]:

$$
\begin{aligned}
&F_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + \frac{1}{4}(x_{i+1} - x_{i-1})^2,\\
&x_0 = 0,\ x_{n+1} = 20.
\end{aligned} \qquad (50)
$$

Initial values are fixed to be $x_i = 1$, $i = 1, \ldots, n$.

For this case we use $m = 2$ and a large $\nu = -110$ to speed up the rate of convergence, which needs 749 steps with a time stepsize $h = 10^{-4}$ used in the GPS integrating method. When the convergent criterion is given by 10^{-6}, the residual error $\left(\sum_{i=1}^{10} F_i^2\right)^{1/2}$ of numerical solutions is about 9.81×10^{-7}. In Fig. 5 we plot the residual error with respect to k, and the numerical solutions of x_i, $i = 1, \ldots, 10$ are recorded in Table 3.

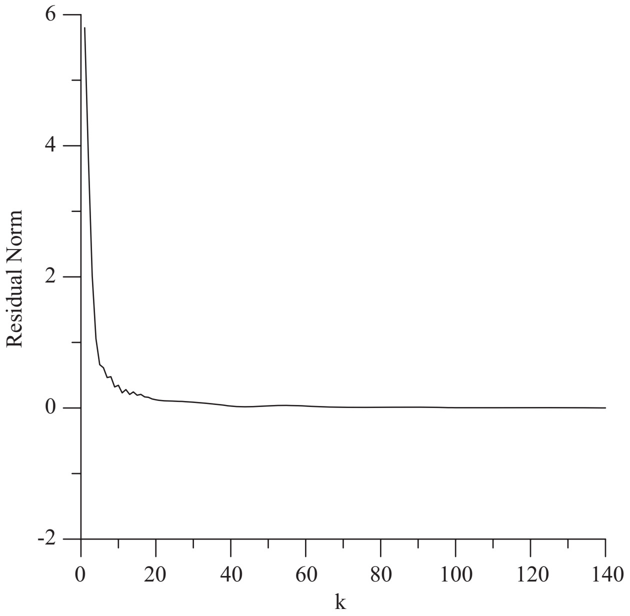

Fig. 4. The residual norm of Example 4.

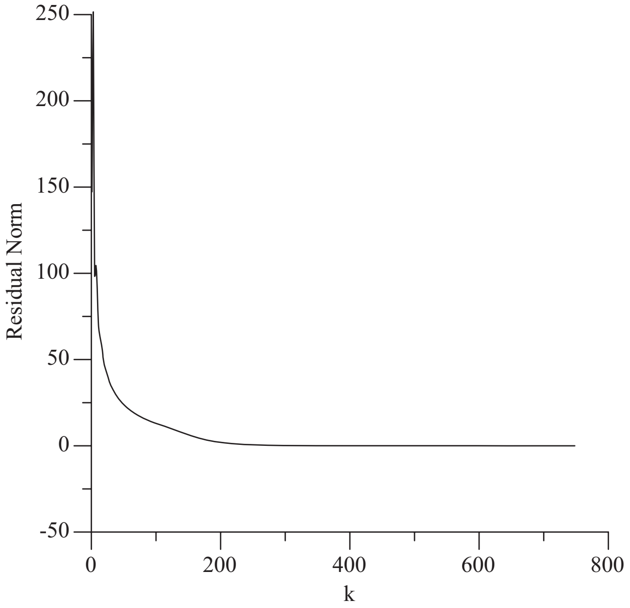

Fig. 5. The residual norm of Example 5.

Table 3. The numerical solutions of Example 5 with *n* = 10.

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

As compared with those reported by Spedicato and Huang [46] for the Newton-like methods, the present modified Newton method is more accurate and time saving, where the computational time is lesser than 0.1 sec by using a PC586.

6. Example 6

Then, we consider a similar test example given by Krzyworzcka [22]:

$$\begin{aligned} F_1 &= (3-5x_1)x_1 + 1 - 2x_2, \\ F_i &= (3-5x_i)x_i - x_{i-1} - 2x_{i+1}, \quad i = 2, \ldots, 9, \\ F_{10} &= (3-5x_{10})x_{10} + 1 - x_9. \end{aligned} \tag{51}$$

For this case we use $m = 1$ and a large $\nu = 20$ to speed up the rate of convergence, which needs only 55 steps with a time stepsize $h = 0.01$ used in the GPS integrating method. The initial values are fixed to be $x_i = -0.1$, $i = 1, \ldots, 10$. When the convergent criterion is given by 10^{-6}, the residual error $\left(\sum_{i=1}^{10} F_i^2\right)^{1/2}$ of numerical solutions is about 9.61×10^{-7}. In Fig. 6 we plot the residual error with respect to k, and the numerical solutions of x_i, $i = 1, \ldots, 10$. are recorded in Table 4. As reported by Mo, Liu and Wang [42] the Newton method cannot be applied for this example, and their solutions obtained by the conjugate direction particle swarm optimization method are different from the present solutions. For this example it may have multiple solutions, but Krzyworzcka [22] didn't give solution for this example. Obviously, our method converges faster than that in the above cited paper by Mo, Liu and Wang.

7. Example 7

In this example we apply the MNM to solve the following boundary value problem [26]:

$$\begin{aligned} u'' &= \frac{3}{2}u^2, \\ u(0) &= 4, \; u(1) = 1. \end{aligned} \tag{52}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{53}$$

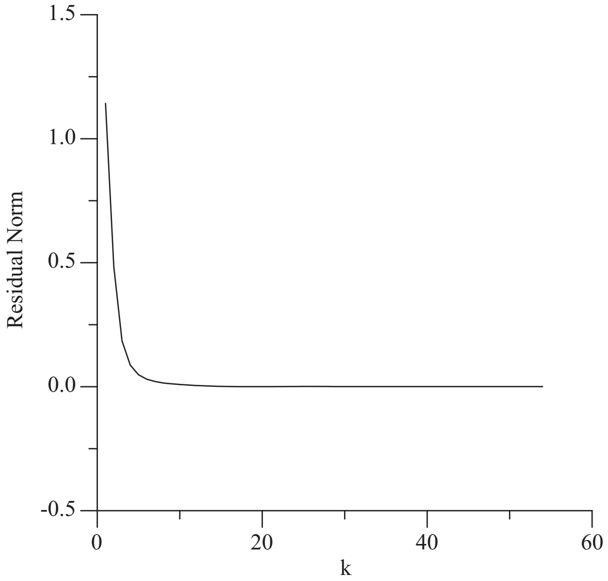

Fig. 6. The residual norm of Example 6.

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

$$\begin{aligned} F_i &= \frac{1}{(\Delta x)^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2, \\ u_0 &= 4, \; u_{n+1} = 1, \end{aligned} \tag{54}$$

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

Using the following parameters $m = 3$, $n = 20$, $h = 10^{-3}$, $\nu = -0.4$ and $\varepsilon = 10^{-4}$ we compute the roots of the above system. In Fig. 7(a) we plot the residual error with respect to k, and compare the numerical solution with exact solution in Fig. 7(b), which can be seen that the error as shown in Fig. 7(c) is very small in the order of 10^{-3}.

V. CONCLUSIONS

Since the work of Newton, iterative algorithms were developed by many researchers, extending to continuous type of systems by introducing an artificial time. The present paper transformed the continuous form $\dot{x}_j(t) + B_{ij}^{-1}F_i(x_j) = 0$ of the Newton's algorithm into another continuous form $(1-s)B_{ij}x'_j(s) + F_i(x_j) = 0$ through a new time variable of $s = 1 - e^{-t}$. A discretization of the above equation by a backward difference is performed, and an ODEs system is derived by introducing a fictitious time. The iterative algorithm, which was obtained by applying the GPS to the resultant ODEs, does not need the inverse of B_{ij}, and is computationally far more efficient than the Newton's algorithm. In doing so, we found that the modified Newton method not only can remove the drawbacks of the Newton's method, but also can preserve the quadratically convergent speed, as shown in the plots of the

Table 4. The numerical solutions of Example 6.

x_1	x_2	x_3	x_4	x_5
–0.2804040215	–0.1171721541	–0.06987970309	–0.05844167655	–0.06126148975
x_6	x_7	x_8	x_9	x_{10}
–0.07205400694	–0.09042982540	–0.1200616717	–0.1709146287	–0.2693706395

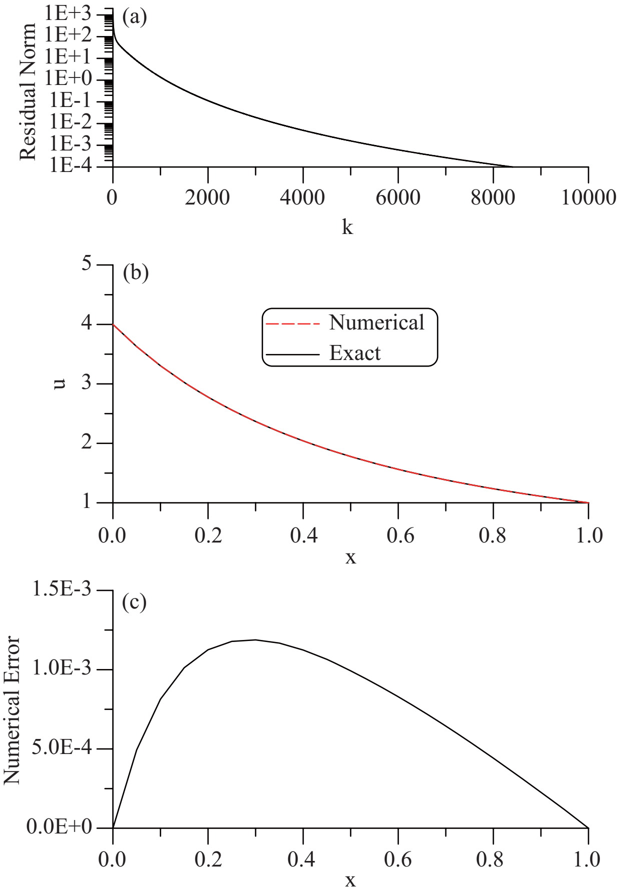

Fig. 7. Applying the MNM to a boundary value problem: (a) residual norm, (b) comparing numerical and exact solutions, and (b) displaying the numerical error.

residual norm vs. iteration number *k* for many examples examined in this paper. Numerical examples confirmed that the modified Newton method is highly efficient to find the true solutions with the residual errors being very small. The modified homotopy method is more complex than the modified Newton method; however, the accuracy and efficiency of the modified Newton method are much better than that of the modified homotopy method.

REFERENCES

1. Alber, Y. I., "Continuous processes of the Newton type," *Differential Equations*, Vol. 7, pp. 1461-1471 (1971).
2. Allgower, E. L. and Georg, K., *Numerical Continuation Methods: An Introduction*, Springer, New York (1990).
3. Ascher, U., Huang, H., and van den Doel, K., "Artificial time integration," *BIT*, Vol. 47, pp. 3-25 (2007).
4. Ascher, U., Mattheij, R., and Russell, R., *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia (1995).
5. Atluri, S. N., *Methods of Computer Modeling in Engineering and Sciences*, Tech. Science Press, 1400 pages (2002).
6. Atluri, S. N., Liu, H. T., and Han, Z. D., "Meshless local Petrov-Galerkin (MLPG) mixed collocation method for elasticity problems," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 14, pp. 141-152 (2006).
7. Atluri, S. N. and Shen, S., "The meshless local Petrov-Galerkin (MLPG) method: a simple & less-costly alternative to the finite and boundary element methods," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 3, pp. 11-51 (2002).
8. Atluri, S. N. and Zhu, T. L., "A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics," *Computational Mechanics*, Vol. 22, pp. 117-127 (1998).
9. Atluri, S. N. and Zhu, T. L., "A new meshless local Petrov-Galerkin (MLPG) approach to nonlinear problems in computer modeling and simulation," *Computer Modeling and Simulation in Engineering: CMSE*, Vol. 3, pp. 187-196 (1998).
10. Boggs, P. and Dennis, J. E., "A stability analysis for perturbed nonlinear analysis methods," *Mathematics of Computation*, Vol. 30, pp. 199-215 (1976).
11. Broyden, C. G., "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation*, Vol. 19, pp. 577-593 (1965).
12. Chow, S. N., Mallet-Paret, J., and Yorke, J. A., "Finging zeroes of maps: homotopy methods that are constructive with probability one," *Mathematics of Computation*, Vol. 32, pp. 887-889 (1978).
13. Chu, M. T., "On the continuous realization of iterative processes," *SIAM Review*, Vol. 30, pp. 375-387 (1988).
14. Davidenko, D., "On a new method of numerically integrating a system of nonlinear equations," *Doklady Akademii Nauk SSSR*, Vol. 88, pp. 601-604 (1953).
15. Dennis, J. E., "On the convergence of Broyden's method for nonlinear systems of equations," *Mathematics of Computation*, Vol. 25, pp. 559-567 (1971).
16. Dennis, J. E. and More, J. J., "A characterization of superlinear convergence and its application to quasi-Newton method," *Mathematics of Computation*, Vol. 28, pp. 549-560 (1974).
17. Dennis, J. E. and More, J. J., "Quasi-Newton methods, motivation and theory," *SIAM Review*, Vol. 19, pp. 46-89 (1977).
18. Deuflhard, P., *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Springer, New York (2004).
19. Hirsch, M. and Smale, S., "On algorithms for solving $f(x) = 0$," *Communications on Pure and Applied Mathematics*, Vol. 32, pp. 281-312 (1979).
20. Hsu, S. B., *The Numerical Methods for Nonlinear Simultaneous Equations*, Central Book Publisher, Taipei, Taiwan (1988).
21. Kellogg, R. B. T., Li, T. Y., and Yorke, J. A., "A constructive proof of the Brouwer fixed-point theorem and computational results," *SIAM Journal*

on Numerical Analysis: a Publication of the Society of Industrial and Applied Mathematics, Vol. 13, pp. 473-483 (1976).

22. Krzyworzcka, S., "Extension of the Lanczos and CGS methods to systems of non-linear equations," *Journal of Computational and Applied Mathematics*, Vol. 69, pp. 181-190 (1996).
23. Li, T. Y. and Yorke, J. A., "A simple reliable numerical algorithm for following homotopy paths," *Analysis and Computation of Fixed Points*, Robinson, S. M. ed., pp. 73-91, Academic Press, New York (1980).
24. Li, T. Y., "Numerical solution of multivariate polynomial systems by homotopy continuation methods," *Acta Numerica*, Vol. 6, pp. 399-436 (1997).
25. Liu, C.-S., "Cone of non-linear dynamical system and group preserving schemes," *International Journal of Non-Linear Mechanics*, Vol. 36, pp. 1047-1068 (2001).
26. Liu, C.-S., "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 (2006).
27. Liu, C.-S., "A modified Trefftz method for two-dimensional Laplace equation considering the domain's characteristic length," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 21, pp. 53-65 (2007).
28. Liu, C.-S., "A highly accurate solver for the mixed-boundary potential problem and singular problem in arbitrary plane domain," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 20, pp. 111-122 (2007).
29. Liu, C.-S., "An effectively modified direct Trefftz method for 2D potential problems considering the domain's characteristic length," *Engineering Analysis with Boundary Elements*, Vol. 31, pp.983-993 (2007).
30. Liu, C.-S., "A highly accurate collocation Trefftz method for solving the Laplace equation in the doubly-connected domains," *Numerical Methods for Partial Differential Equations*, Vol. 24, pp.179-192 (2008).
31. Liu, C.-S., "Solving an inverse Sturm-Liouville problem by a Lie-group method," *Boundary Value Problems*, Vol. 2008, Article ID 749865 (2008).
32. Liu, C.-S., "Identifying time-dependent damping and stiffness functions by a simple and yet accurate method," *Journal of Sound and Vibration*, Vol. 318, pp. 148-165 (2008).
33. Liu, C.-S., "A Lie-group shooting method for simultaneously estimating the time-dependent damping and stiffness coefficients," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 27, pp. 137-149 (2008).
34. Liu, C.-S., "A Lie-group shooting method estimating nonlinear restoring forces in mechanical systems," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 35, pp.157-180 (2008).
35. Liu, C.-S., "A time-marching algorithm for solving non-linear obstacle problems with the aid of an NCP-function," *CMC: Computers, Materials & Continua*, Vol. 8, pp.53-65 (2008).
36. Liu, C.-S., "A fictitious time integration method for two-dimensional quasilinear elliptic boundary value problems," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 33, pp. 179-198 (2008).
37. Liu, C.-S. and Atluri, S. N., "A novel time integration method for solving a large system of non-linear algebraic equations," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 31, pp. 71-83 (2008).
38. Liu, C.-S. and Atluri, S. N., "A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to non-linear optimization," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 34, pp. 155-178 (2008).
39. Liu, C.-S. and Atluri, S. N., "A novel fictitious time integration method for solving the discretized inverse Sturm-Liouville problems, for specified eigenvalues," *CMES: Computer Modeling in Engineering & Sciences*, Vol. 36, pp. 261-285 (2008).
40. Liu, C.-S., Chang, J. R., Chang, K. H., and Chen, Y. W., "Simultaneously estimating the time-dependent damping and stiffness coefficients with the aid of vibrational data," *CMC: Computers, Materials & Continua*, Vol. 7, pp. 97-107 (2008).
41. Maruster, S., "The stability of gradient-like methods," *Applied Mathematics and Computation*, Vol. 117, pp. 103-115 (2001).
42. Mo, Y., Liu, H., and Wang, Q., "Conjugate direction particle swarm optimization solving systems of nonlinear equations," *Computers & Mathematics with Applications*, Vol. 57, pp. 1877-1882 (2009).
43. Nocedal, J. and Wright, S., *Numerical Optimization,* Springer, New York (1999).
44. Roose, A., Kulla, V., Lomb, M., and Meressoo, T., "Test examples of systems of non-linear equations," Estonian Software and Computer Service Company, Tallin (1990).
45. Smale, S., "A convergent process of price adjustment and global Newton methods," *Journal of Mathematical Economics*, Vol. 3, pp. 107-120 (1976).
46. Spedicato, E. and Hunag, Z., "Numerical experience with Newton-like methods for nonlinear algebraic systems," *Computing*, Vol. 58, pp. 69-89 (1997).
47. Watson, L. T., Sosonkina, M., Melville, R. C., Morgan, A. P., and Walker, H. F., "A suite of FORTRAN 90 codes for globally convergent homotopy algorithms," *ACM Transactions on Mathematical Software*, Vol. 23, pp. 514-549 (1997).
48. Zhu, T., Zhang, J., and Atluri, S. N., "A meshless local boundary integral equation (LBIE) method for solving nonlinear problems," *Computational Mechanics*, Vol. 22, pp. 174-186 (1998).
49. Zhu, T., Zhang, J., and Atluri, S. N., "A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and non-linear boundary value problems," *Engineering Analysis with Boundary Elements*, Vol. 23, pp. 375-389 (1999).