 CMES, vol.53, no.1, pp.47-71, 2009

A Scalar Homotopy Method for Solving an Over/Under-Determined System of Non-Linear Algebraic Equations

Chein-Shan Liu[1], Weichung Yeih[2], Chung-Lun Kuo[3] and Satya N. Atluri[4]

Abstract: Iterative algorithms for solving a system of nonlinear algebraic equations (NAEs): $F_i(x_j) = 0$, i, j= 1,...,n date back to the seminal work of Issac Newton. Nowadays a Newton-like algorithm is still the most popular one to solve the NAEs, due to the ease of its numerical implementation. However, this type of algorithm is sensitive to the initial guess of solution, and is expensive in terms of the computations of the Jacobian matrix $\partial F_i/\partial x_j$ and its inverse at each iterative step. In addition, the Newton-like methods restrict one to construct an iteration procedure for n-variables by using n-equations, which is not a necessary condition for the existence of a solution for underdetermined or overdetermined system of equations. In this paper, a natural system of first-order nonlinear Ordinary Differential Equations (ODEs) is derived from the given system of Nonlinear Algebraic Equations (NAEs), by introducing a scalar homotopy function gauging the total residual error of the system of equations. The iterative equations are obtained by numerically integrating the resultant ODEs, which does not need the inverse of $\partial F_i/\partial x_j$. The new method keeps the merit of homotopy method, such as the global convergence, but it does not involve the complicated computation of the inverse of the Jacobian matrix. Numerical examples given confirm that this Scalar Homotopy Method (SHM) is highly efficient to find the true solutions with residual errors being much smaller.

Keywords: Nonlinear algebraic equations, Iterative method, Ordinary differential equations, Scalar homotopy method (SHM)

---

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan. e-mail: liucs@ntu.edu.tw
[2] Department of Harbor & River Engineering and Computation and Simulation Center, National Taiwan Ocean University, Keelung, Taiwan
[3] Department of Systems Engineering and Naval Architecture, National Taiwan Ocean University, Keelung, Taiwan
[4] Center for Aerospace Research & Education, University of California, Irvine

1 Introduction

The numerical solution of (linear or nonlinear, well-conditioned or ill-conditioned, and underdetermined or overdetermined or ill-posed systems of) algebraic equations is one of the main aspects of computational mathematics. In many practical nonlinear engineering problems, methods such as the finite element method, boundary element method, finite volume method, the MLPG method (which leads to many different meshless methods), 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, 1999), Atluri and Zhu (1998), 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 a system of NAEs for a nonlinear problem. Boundary collocation methods, such as those used by Liu (2007a, 2007b, 2007c, 2008), for the modified Trefftz method of Laplace equation, also lead to a large system of well-conditioned linear algebraic equations.

Over the past twenty 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 (1990) and Deuflhard (2004) are devoted to the continuation methods for solving NAEs.

Here we consider the following system of nonlinear algebraic equations:

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

The Newton method for solving these equations is given algorithmically by

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

where we use $\mathbf{x} := (x_1,\cdots,x_n)^T$ and $\mathbf{F} := (F_1,\cdots,F_n)^T$ to represent the vectors, $\mathbf{B}$ is an $n\times n$ matrix with its *ij*-th component given by $\partial F_i/\partial x_j$, and $\mathbf{x}_{k+1}$ is the $(k+1)$-th iteration for $\mathbf{x}$.

Newton's method has a great advantage, in that it is quadratically convergent. However, it still has some drawbacks: the difficulty in guessing the starting values for $\mathbf{x}$, the computational burden of finding $[\mathbf{B}(\mathbf{x}_k)]^{-1}$, and $\mathbf{F}$ being required to be differentiable. Some quasi-Newton methods are hence developed to overcome some of 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. (1) by numerically integrating

$$\dot{\mathbf{x}}(t) = -\mathbf{H}_{\mathbf{x}}^{-1}\mathbf{H}_t(\mathbf{x},t), \tag{3}$$

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

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 was later refined by Kellogg, Li and Yorke (1976), Chow, Mallet-Paret and Yorke (1978), Li and Yorke (1980), and Li (1997). The homotopy method has many merits, such as its global convergence (i.e., one can obtain the solution for arbitrary initial guess), multiple roots searching (due to the fact that one homotopy path cannot intersect with another homotopy path); but it also suffers from its very slow convergence speed in comparison with other iteration methods.

Hirsch and Smale (1979) also derived a "continuous Newton method" governed by the following differential equation:

$$\dot{\mathbf{x}}(t) = -\mathbf{B}^{-1}(\mathbf{x})\mathbf{F}(\mathbf{x}), \tag{5}$$

$$\mathbf{x}(0) = \mathbf{a}, \tag{6}$$

where $\mathbf{a} \in \mathbf{R}^n$. It can be seen that the ODEs in Eqs. (3) and (5) are difficult to solve, because they all involve inverting a matrix. Atluri, Liu and Kuo (2009) proposed a modified Newton method for solving nonlinear algebraic equations avoiding the inverse of the Jacobin matrix. In addition, the number of equations and the number of unknowns should be equal. Numerically speaking, such a constraint makes the inverse of a matrix possible. However, it is not a necessary condition for the existence of solutions for a system of underdetermined or overdetermined system of equations.

To eliminate the need for inverting a matrix in the iteration procedure, the first-order ODE system such as

$$\dot{\mathbf{x}} = -\mathbf{F}(\mathbf{x}), \tag{7}$$

$$\mathbf{x}(0) = \mathbf{a} \tag{8}$$

has been used [Ramm (2007)]. However, iteration procedure in Eq. (7) is very sensitive to the initial guess and may have a very low convergence speed. Liu and Atluri (2008) have proposed another first-order nonlinear ODE system, as:

$$\dot{\mathbf{x}} = -\frac{\nu}{1+t}\mathbf{F}(\mathbf{x}), \tag{9}$$

or in general,

$$\dot{\mathbf{x}} = -\frac{\nu}{q(t)}\mathbf{F}(\mathbf{x}), \tag{10}$$

where ν is nonzero, and $q(t)$ may in general be a monotonically increasing function of t. In their approach, the term $\frac{\nu}{1+t}$ [or $\frac{\nu}{q(t)}$] plays the role of a controller to help one obtain a solution even for bad initial guesses, and speeds up the convergence. Liu and Chang (2009) combined the above formula with nonstandard group preserving scheme for solving a system of ill-posed linear equations and accurate results were obtained. Recently, Ku, Yeih, Liu and Chi (2009) employed a new time-like function $q(t) = (1+t)^m$, $0 < m \leq 1$ in Eq. (10), and better performance was observed.

The systems in Eqs. (7)-(10), while avoiding the calculation of an inverse of a matrix, all have the property of a local convergence as in the Newton method of Eqs. (5) and (6). From the above brief review, one can find that only the homotopy method can guarantee the global convergence.

Below we will develop a new system of ODEs, which are equivalent to the original equation (1). This new approach is similar to the vector homotopy method and we name it the scalar homotopy method (SHM). This new method keeps the spirit of the vector homotopy method, such that the global convergence can be guaranteed. On the other hand, this new approach does not need to calculate the inverse of a matrix as in the vector homotopy method. Thus, it saves a lot of computational efforts. Furthermore, the SHM can be used even when the number of equations is not equal to the number of unknowns, i.e., the system of equations can be over/under-determined.

2 A scalar homotopy method

2.1 Transformation of a system of NAEs into a system of nonlinear ODEs

First, we note that the statement,

$$b_i = 0,\ i = 1, \cdots, m, \tag{11}$$

is mathematically equivalent to

$$\sum_{i=1}^{m} b_i b_i = 0. \tag{12}$$

That is, Eq. (11) implies Eq. (12), and conversely, Eq. (12) implies Eq. (11).

As is commonly done in the vector homotopy theory, we consider an initial vector equation:

$$\mathbf{x}-\mathbf{a}=\mathbf{0}, \tag{13}$$

and a final vector equation:

$$\mathbf{F}(\mathbf{x})=\mathbf{0}, \tag{14}$$

which is our target to solve.

Now we introduce a scalar function:

$$h(\mathbf{x},t)=\frac{1}{2}\left[t\,\|\mathbf{F}(\mathbf{x})\|^2-(1-t)\,\|\mathbf{x}-\mathbf{a}\|^2\right]. \tag{15}$$

Corresponding to the above introduced vector homotopy function $\mathbf{H}$ = (1-t)($\mathbf{x}$-$\mathbf{a}$)+$t\mathbf{F}(\mathbf{x})$, the present $h(\mathbf{x}$, t) is a scalar homotopy function. When t = 0 we have

$$h(\mathbf{x},\ t=0)=0\rightarrow\|\mathbf{x}-\mathbf{a}\|^2=0\rightarrow x_i=a_i,\ i=1,\ldots,n. \tag{16}$$

Similarly, when t= 1 we have

$$h(\mathbf{x},\ t=1)=0\rightarrow\|\mathbf{F}(\mathbf{x})\|^2=0\rightarrow F_i=0,\ i=1,\ldots,m. \tag{17}$$

The last implication comes from Eqs. (11) and (12). It can be seen that the number of equations now is equal to m and the number of unknowns is equal to n and they do not need to be equal to each other. The homotopy theory basically aims to construct a path from the solution of a given auxiliary function to the solution of the desired function continuously. It means that for any $t\in[0,1]$ we have to solve the following equation:

$$h(\mathbf{x},t)=\frac{1}{2}\left[t\,\|\mathbf{F}(\mathbf{x})\|^2-(1-t)\,\|\mathbf{x}-\mathbf{a}\|^2\right]=0. \tag{18}$$

In order to guarantee that the path lies on the hyper-surface described by Eq. (18), the following consistency equation then can be derived:

$$\frac{\partial h}{\partial t}+\frac{\partial h}{\partial \mathbf{x}}\cdot\frac{d\mathbf{x}}{dt}=0. \tag{19}$$

In an analogy to the theory of plasticity, Eq. (18) may be considered to be the definition of a "yield-surface", and Eq. (19) is the condition of "consistency". For the vector homotopy function, the consistency equation can uniquely determine the

ODE for $\mathbf{x}$ as written in Eqs. (3) and (4). However, since the equation written in Eq. (19) is a scalar equation and it cannot determine the ODE for $\mathbf{x}$ unless one prescribes a certain form for $\frac{d\mathbf{x}}{dt}$. In plasticity theory, the normality condition, or equivalently the "stability" of the material "in the small" was originally derived by Drucker and Ilyushin, based on the inequality of the plastic work (that it should be ≥ 0). A similar motivation for the use of the normality condition for $\frac{d\mathbf{x}}{dt}$ is based on the stability of solutions. We note that the form of $\frac{d\mathbf{x}}{dt}$ needs to be parallel to the gradient of the above scalar homotopy function, such that the trajectory of $\mathbf{x}$ can be equivalent to seeking of $h(\mathbf{x}, t)=0$. We propose here to use

$$\frac{d\mathbf{x}}{dt} = -q\frac{\partial h}{\partial \mathbf{x}}, \tag{20}$$

in which the gradient of h is assigned to be the driving force to adjust $\mathbf{x}$. In an analogy to the theory of plasticity, Eq. (20) may be considered to be the "flow-rule", and we will introduce the similarity between them, later. It can be seen that

$$q = \frac{\frac{\partial h}{\partial t}}{\left\|\frac{\partial h}{\partial \mathbf{x}}\right\|^2} \tag{21}$$

by substituting Eq. (20) into Eq. (19), where

$$\frac{\partial h}{\partial t} = \frac{1}{2}\left[\|\mathbf{F}(\mathbf{x})\|^2 + \|\mathbf{x}-\mathbf{a}\|^2\right], \tag{22}$$

$$\frac{\partial h}{\partial \mathbf{x}} = t\mathbf{B}^{\mathrm{T}}\mathbf{F} - (1-t)(\mathbf{x}-\mathbf{a}). \tag{23}$$

$\mathbf{B} := \frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ is usually called the Jacobian matrix of the NAEs. Hence, we can write the nonlinear ODEs for $\mathbf{x}$ as:

$$\dot{\mathbf{x}} = -\frac{\frac{\partial h}{\partial t}}{\left\|\frac{\partial h}{\partial \mathbf{x}}\right\|^2}\frac{\partial h}{\partial \mathbf{x}}. \tag{24}$$

Now we will make an analogy to the plasticity theory. In the plasticity theory, we have the associated flow rule given by [Liu and Chang (2004)]

$$\dot{\mathbf{e}}^P = \dot{\lambda}\frac{\partial h}{\partial \mathbf{x}}, \tag{25}$$

where $\dot{\mathbf{e}}^P$ denotes the plastic strain rate, h is the yield function and $\mathbf{x}$ relates to the stress state. Then by assuming an unit elastic modulus, the evolution of stress $\mathbf{x}$ is governed by

$$\dot{\mathbf{x}} = \dot{\mathbf{e}} - \dot{\lambda}\frac{\partial h}{\partial \mathbf{x}}, \tag{26}$$

where $\dot{\mathbf{e}}$ denotes the total strain rate vector as an input into the elastic-plastic constitutive relation for the stress-rate. Inserting Eq. (26) into the consistency Eq. (19), we can solve for $\dot{\lambda}$ as:

$$\dot{\lambda} = \frac{\frac{\partial h}{\partial t} + \frac{\partial h}{\partial \mathbf{x}} \cdot \dot{\mathbf{e}}}{\left\| \frac{\partial h}{\partial \mathbf{x}} \right\|^2}. \tag{27}$$

The above procedure means that one hopes to keep the trajectory of stress state on the yield surface.

Thus we have the following nonlinear ODEs:

$$\dot{\mathbf{x}} = \dot{\mathbf{e}} - \frac{\frac{\partial h}{\partial t} + \frac{\partial h}{\partial \mathbf{x}} \cdot \dot{\mathbf{e}}}{\left\| \frac{\partial h}{\partial \mathbf{x}} \right\|^2}\frac{\partial h}{\partial \mathbf{x}}. \tag{28}$$

It can be seen that if one takes $\dot{\mathbf{e}} = \mathbf{0}$, Eq. (28) is the same as Eq. (24). Actually the ODEs in Eq. (24) and Eq. (28) both can be used as the governing equations for the scalar homotopy theory. One can compare the governing equations for the scalar homotopy theory, Eq. (24) or Eq. (28), and that for the vector homotopy theory, Eq. (3), and easily find that the scalar homotopy theory is much simpler than the vector homotopy theory. First, the scalar homotopy theory does not need to calculate the inverse of the Jacobian matrix at each iteration step. Second, the scalar homotopy theory does not require that the number of equations be equal to the number of unknowns.

2.2 A Group Preserving Scheme to Integrate the System of Nonlinear ODEs

We can write Eq. (24) or Eq. (28) as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \ \mathbf{x} \in \mathbf{R}^n, \ 0 < t \leq 1. \tag{29}$$

Liu (2001) has embedded the above system into an augmented differential system, and obtained the following group preserving scheme (GPS) to integrate Eq. (29):

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_k \mathbf{f}_k, \tag{30}$$

$$\|\mathbf{x}_{k+1}\| = a_k \|\mathbf{x}_k\| + b_k \frac{\mathbf{f}_k \cdot \mathbf{x}_k}{\|\mathbf{f}_k\|}, \tag{31}$$

where

$$a_k := \cosh\left(\frac{\Delta t \,\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right), \tag{32}$$

$$b_k := \sinh\left(\frac{\Delta t \,\|\mathbf{f}_k\|}{\|\mathbf{x}_k\|}\right), \tag{33}$$

$$\eta_k := \frac{b_k \,\|\mathbf{x}_k\| \,\|\mathbf{f}_k\| + (a_k - 1)\mathbf{f}_k \cdot \mathbf{x}_k}{\|\mathbf{f}_k\|^2}. \tag{34}$$

Starting from an initial value of $\mathbf{x}_0$ which can be guessed in a rather free way, we employ the above GPS to integrate Eq. (24) or Eq. (28) from $t = 0$ to a final time t_f= 1.

2.3 Restart

Although the homotopy theory guarantees that when t= 1 the solution for the desired equations can be solved, the numerical integration to keep the consistency Eq. (19) may not be easily carried out. Especially when the system of equations is highly nonlinear, the time step required is very small, such that to reach t= 1 may lead to numerous evolution steps. It is well known that the convergence speed for the homotopy method is awfully slow in comparison with other methods, such as the Newton's method. In the previous literature, the restart technique has been proposed to speed up the convergence of the vector homotopy method [Nazareth (2003)]. A similar idea can be used here for the scalar homotopy method. In the following, the restart method will be briefly introduced.

Instead of using a small time increment, one can choose an adequate time step which can make the goal of t= 1 being accomplished very fast. However, one can expect that the "solution" at t= 1 may be not the true solution at all, since one did not select a very small time increment to preserve the consistency equation. However, one can use the final value, even though it is wrong, to replace the initial guess and redo the integration again. The abovementioned procedure continues until the convergence criterion is reached. The convergence criterion is written as

$$\|\mathbf{F}(\mathbf{x})\|_K \leq \varepsilon, \tag{35}$$

where the subscript K represents the K-th restart procedures, and ε is a convergence criterion defined by the user.

The reason why we can use the restart method is briefly explained as follows. Owing to the global convergence property for the homotopy method, one can use any initial guess, and not be concerned that the solution cannot be obtained. This global convergence property makes the restart method workable, no matter which initial guess one uses, where the initial guess for the current procedure comes from the final solution of the last evolution. Other numerical methods such as the Newton's method have the local convergence, such that the restart technique sometimes may fail.

Using this technique, one can now set the integration steps from t= 0 to t= 1 to be a small number, for example, 5 steps. Although one may restart many times to reach the final solution, the total number of evolution time steps using the restart method is much less than that using very small time increments. We will demonstrate this claim in the next section.

3 Numerical examples

3.1 Example 1

The following equations are considered:

$$F_1(x,y) = x^2 - y - 1 = 0,\ F_2(x,y) = y^2 - x - 1 = 0. \tag{36}$$

There are four roots: (-1, 0), (0, -1), $\left(\left(1+\sqrt{5}\right)/2, \left(1+\sqrt{5}\right)/2\right) \approx$ (1.618034, 1.618034) and $\left(\left(1-\sqrt{5}\right)/2, \left(1-\sqrt{5}\right)/2\right) \approx$(-0.618034, -0.618034). The strain rate is $\dot{\mathbf{e}} = 10^{-16}\,\mathbf{1}$ for all examples, where $\mathbf{1} = (1,\ldots,1)^{\mathrm{T}}$. For the first root, we set the initial guess to be (-20,-2). The convergence criterion for this problem is ε=10^{-10}. If the restart method is not adopted, even when we set the time step size to be $\Delta t = 10^{-7}$(it means totally 10^7 evolution steps are used), the residual norm is about 10^{-1}, which is far from our convergence criterion. However, if one uses the time step size $\Delta t = 0.5$ and uses the restart technique, the solution of (-0.99999999991274, -0.00000000015972) is obtained and the residual of each equation is $(F_1,F_2) =$ (- 0.14797385539111, - 0.87260865200278) $\times 10^{-10}$. The total number of evolution steps is 444. In the following, we fixed our parameters as: time step size $\Delta t = 0.5$ and restart technique is adopted.

For the second root, we use the initial guess as (1,-5). After 338 iteration steps, the solution of (0.00000000019164, -1.00000000005814) is obtained. The residual of each equation is $(F_1,F_2) =$ (0.58138382996731, - 0.75364603446815) $\times 10^{-10}$.

For the third root, the initial guess is chosen as (5, 5). After 80 iteration steps, the solution of (1.61803398877212, 1.61803398877212) is obtained. The residual of each equation is $(F_1,F_2) =$ (0.49692916448407, 0.49692916448407) $\times 10^{-10}$.

For the fourth root, the initial guess is chosen as (-5,-2). After 566 iteration steps, the solution of (-0.61803398892723, -0.61803398859329) is obtained. The residual of each equation is $(F_1,F_2) =$ (0.62595484351391, - 0.16237899913563) $\times 10^{-10}$.

We solved this problem by the Fictitious Time Integration Method (FTIM) also, in order to compare the results with the present approach. However, it is found that the third and fourth roots cannot be obtained by the FTIM, unless the initial guess is very close to the exact solutions. It appears that some roots of NAEs may not be easily solved by FTIM if the initial guess cannot be chosen appropriately, i.e., the initial guess is not within the attracting zone of the root.

3.2 Example 2

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

$$F_1(x,y) = x - y^2 = 0,$$

$$F_2(x,y) = (y-1)^2(y-2)^2 + \left(x - y^2\right)^2 = 0. \tag{37}$$

The two real roots are (x, y) = (1, 1) and (x, y) = (4, 2). For this example, the convergence criterion is chosen as ε=10^{-7}.

For the first root, the initial guess is chosen as (0, 10). After 3424 steps, the solution of (1.00055782102710, 1.00027890021624) is obtained. The residual of each equation is $(F_1, F_2) =$ (- 0.57190704394472, 0.77741951246269) $\times 10^{-7}$.

For the second root, the initial guess is chosen as (3, 9). After 30904 steps, the solution of (3.99989872456104, 1.99997466649739) is obtained. The residual of each equation is $(F_1, F_2) =$ (0.57929709473825, 0.00641757193548) $\times 10^{-7}$.

The residual norm versus the evolution step number is illustrated in Fig. 1, and the locus of evolution for each root is plotted in Fig. 2. The trajectories as shown in Fig. 2 do not look like continuous curves because the restart method is used. Every time when one restarts, the evolution forces refresh from t= 0 and are totally different from the previous step. We also solved this example by the FTIM, but the residual norm is still 9.1869×10^{-6} after 100000 evolution steps. It can be seen that the present scalar homotopy method is much faster than the FTIM for this example. In comparison, the scalar homotopy method reaches a more accurate result than the FTIM for the same number of evolution steps.

3.3 Example 3

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

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

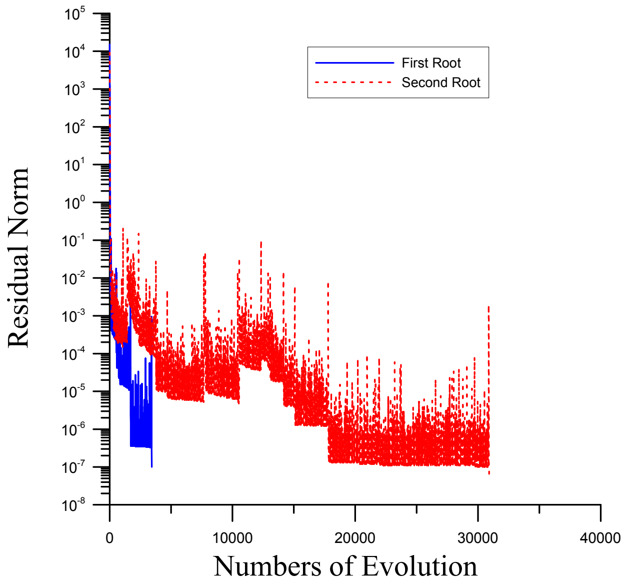

Figure 1: The residual norm for each root of Example 2.

$$F_2(x,y) = 3x^2y - y^3 - a_1\left(4xy - y^2\right) + b_2x^2 + c_2 = 0. \tag{38}$$

The parameters used and results obtained in this test are listed in Table 1. In Table 1, "IN" denotes the iteration number. The convergence criterion is $\varepsilon = 10^{-10}$ for problem 1 and problem 2, and $\varepsilon = 10^{-8}$ for problem 3. For problem 1, we find three solutions by using three different initial guesses. For problem 2, we find four solutions using four different initial guesses. For problem 3, we find three solutions using three different initial guesses. It should be emphasized that the third problem is hard to solve because there appears a much larger coefficient $a_1 = 200$ than others. As reported by Hsu (1988), he could not calculate the third problem by using the vector 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.

Hirsch and Smale (1979) used the continuous Newton algorithm to calculate the above three problems. However, as pointed out by Liu and Atluri (2008), the results obtained by Hirsch and Smale (1979) are not accurate. In the paper by Liu and Atluri (2008), only one root for problems 2 and 3 is presented. The currently proposed scalar homtopy method can find all roots for all three problems successfully, and multiple roots are reported. The convergence and accuracy are also better

Table 1: The parameters and results for Example 3.

Problem 1				
$(a_1, b_1, c_1, a_2, b_2, c_2)$	Initial guess	IN	Solution (x,y)	Residual $(F_1,F_2)\times 10^{-10}$
(25,1,2,3,4,5)	(10, 2)	322	(1.63597179958629, 13.84766532577993)	(0.19831247755064, -0.90331297997182)
	(0.5, 0.5)	586	(-50.39707550115868, -0.80424262327705)	(-0.40315306648608, -0.80035533756018)
	(0.5, 10)	906	(0.62774246874695, 22.24441227822409)	(0.34589220376802, 0.92033936027747)
Problem 2				
$(a_1, b_1, c_1, a_2, b_2, c_2)$	Initial guess	IN	Solution (x,y)	Residual $(F_1,F_2)\times 10^{-10}$
(25,-1,-2,-3,-4,-5)	(0, 2)	398	(0.13421210219977, 0.81112749271137)	(0.15566214983664, -0.76411765803641)
	(0, 10)	174	(-0.16363472338453, 0.23052874358429)	(-0.57356785987395, -0.35127456499140)
	(-1, 20)	132	(-0.52622363386433, 26.97330868866634)	(-0.04135358722124, -0.70625283399295)
	(-40, 0)	410	(-49.67626510484030, 0.79708118398900)	(0.72745365287119, -0.54569682106376)
Problem 3				
$(a_1, b_1, c_1, a_2, b_2, c_2)$	Initial guess	IN	Solution (x,y)	Residual $(F_1,F_2)\times 10^{-8}$
(200,1,2,3,1,2)	(0, 4)	468	(0.511596009556, 197.936304863638)	(0.02987690095324, -0.76710824359338)
	(-300, 4)	494	(-400.095289676515, -0.200031563605)	(0.04347108228941, -0.65192580223083)
	(10, 100)	164	(12.98635827024471, 89.10206184127932)	(0.18177956917498, 0.76774142598879)

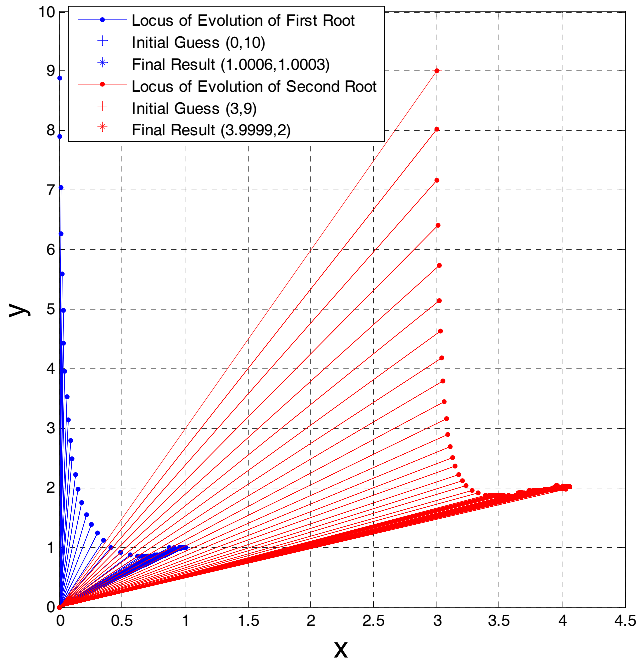

Figure 2: The locus of evolution for each root of Example 2.

than in Liu and Atluri (2008).

3.4 Example 4

In the following, we consider a system of three algebraic equations in three-variables:

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

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

$$F_3(x,y,z) = x^8+y^4+z^9-3 = 0. \tag{39}$$

Apparently, (x, y, z) = (1, 1, 1) is one of the roots. The system of equations in Eq. (39) is nonlinear and multiple roots are possible. In this example, the convergence criterion is chosen as $\varepsilon = 10^{-10}$. The initial guess is (x, y, z) = (0, 0.25, 0.5). After 1342 steps, the solution of (x, y, z) = (0.93054228413587, 1.21836693167919, 0.85109078422645) is obtained, which is one solution different from another solution of (x, y, z) = (1, 1, 1). The residual of each equation is

$$(F_1, F_2, F_3) = (\ 0.41511682979944,\ \text{-}\ 0.80120798884309,\ \text{-}\ 0.16624923659947) \times 10^{-10}.$$

The locus of evolution is plotted in Fig. 3, and the residual norm versus the evolution steps is shown in Fig. 4.

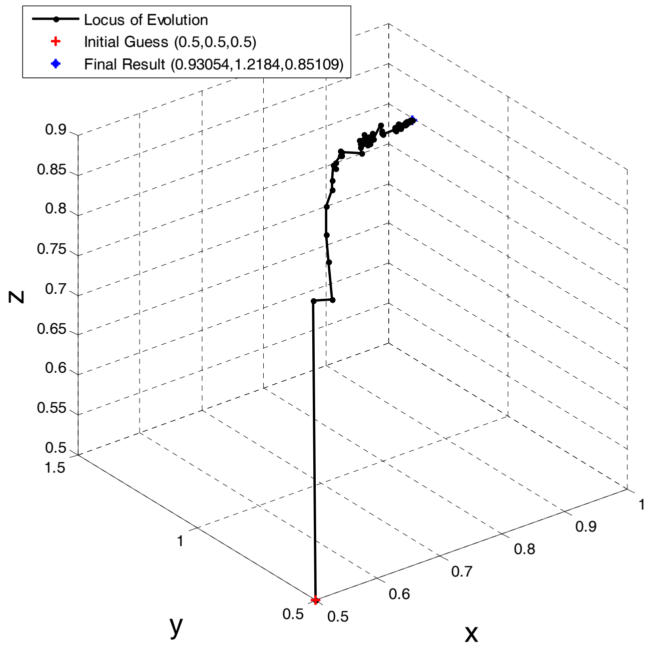

Figure 3: The locus of evolution of Example 4.

3.5 Example 5

The following example is given in Roose, Kulla, Lomb and Meressoo (1990):

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

$$x_0 = 0,\ x_{n+1} = 20. \tag{40}$$

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

x_1	x_2	x_3	x_4	x_5
3.0832	5.3831	7.3952	9.2397	10.969
x_6	x_7	x_8	x_9	x_{10}
12.612	14.186	15.705	17.176	18.606

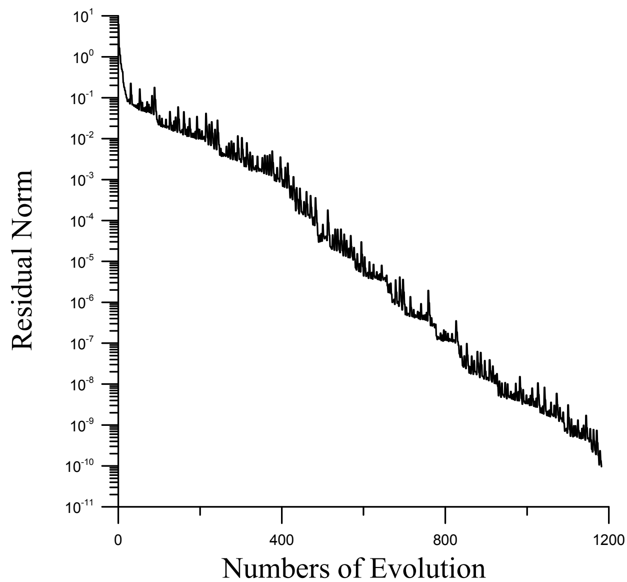

Figure 4: The residual norm for each root of Example 4.

Initial values are fixed to be x_i= 20, i= 1,...,n. The convergence criterion is chosen as $\varepsilon = 10^{-10}$. After 8768 steps, the solution is obtained. The parameter n is 10 and the result is shown in Table 2. As compared with those reported by Spedicato and Huang (1997) for the Newton-like methods, the present scalar homotopy method is more accurate and time saving, where the computational time is less than 0.1 sec by using a PC586.

3.6 Example 6

Then, we consider an example similar to the one given by Krzyworzcka (1996):

$$F_1 = (3-5x_1)x_1 + 1 - 2x_2,$$

$$F_i = (3-5x_i)x_i - x_{i-1} - 2x_{i+1},\ i = 2,\cdots,9,$$

$$F_{10} = (3-5x_{10})x_{10} + 1 - x_9. \tag{41}$$

The initial guess is fixed to be $x_i = -0.1,\ i = 1,..,10$. The convergence criterion is ε =10^{-10}. After 392 steps, the result obtained is reported in Table 3.

As reported by Mo, Liu and Wang (2009) 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

Table 3: The numerical solutions of Example 6.

x_1	x_2	x_3	x_4	x_5
-0.280404179177	-0.117172528010	-0.069880205750	-0.058442152525	-0.061261838916
x_6	x_7	x_8	x_9	x_{10}
-0.072054214387	-0.090429926667	-0.120061711893	-0.170914641178	-0.269370642228

may have multiple solutions, but Krzyworzcka (1996) did not give solution for this example. Obviously, our method converges faster than that by Mo, Liu and Wang (2009).

3.7 Example 7

In this example we apply the scalar homotopy method to solve the following boundary value problem, which is solved by Liu (2006) using the Lie group shooting method:

$$u'' = \frac{3}{2}u^2,$$

$$u(0) = 4,\ u(1) = 1. \tag{42}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{43}$$

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

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

$$u_0 = 4,\ u_{n+1} = 1, \tag{44}$$

where $\Delta x = \frac{1}{n+1}$ is the grid length. We select n= 25 and $\varepsilon = 10^{-10}$. The initial guess is generated randomly as shown in Fig. 5(a). After 32262 steps, the solution is obtained. It can be seen from Fig. 5(a) that the numerical solution perfectly coincides with the exact solution, and the maximum absolute error is about 7×10^{-4} as shown in Fig. 5(b).

3.8 Example 8

In this example, we consider the following equations:

$$F_1(x,y,z) = x^2 + y^2 + z^2 - 1 = 0,$$

$$F_2(x,y,z) = \frac{x^2}{4} + \frac{y^2}{4} + z^2 - 1 = 0. \tag{45}$$

It can be easily seen that this system has two equations in three variables and the solutions are (x, y, z) = (0, 0, 1) and (0, 0, -1). Although this system is very simple, the conventional Newton method and a vector homotopy method fail, since they all

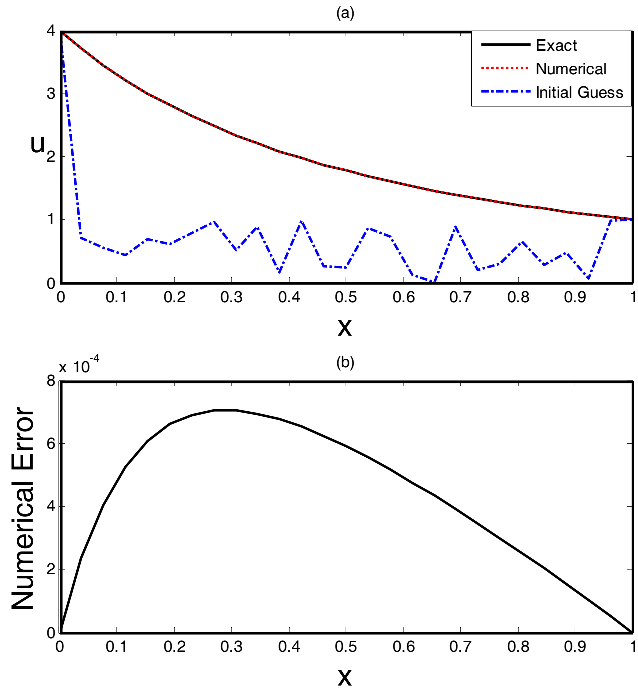

Figure 5: (a) The comparison of numerical and exact solutions and the initial guesses, and (b) the numerical error of Example 7.

require the number of equations should be equal to the number of unknowns. Using the scalar homotopy method proposed in this paper, this restriction is not necessary any more.

In this example, we use the convergence criterion as $\varepsilon = 10^{-6}$. For the first root, the initial guess is chosen as (5, 5, 5). After 17878 steps, the solution of (0.00097065383789, 0.00097065383789, 0.99999940340636) is obtained and the residual of each equation is $(F_1,F_2) = (6.9115081435811, -7.2210249524307) \times 10^{-7}$. For the second root, the initial guess is chosen as (-3, -4, -5). After 9490 steps, the solution of (-0.00066967890114, -0.00089290520789, -0.99999978109166) is obtained. The residual of each equation is

$$(F_1,F_2) = (8.0793290924142, -1.2637924640124) \times 10^{-7}.$$

The trajectories are plotted in Fig. 6.

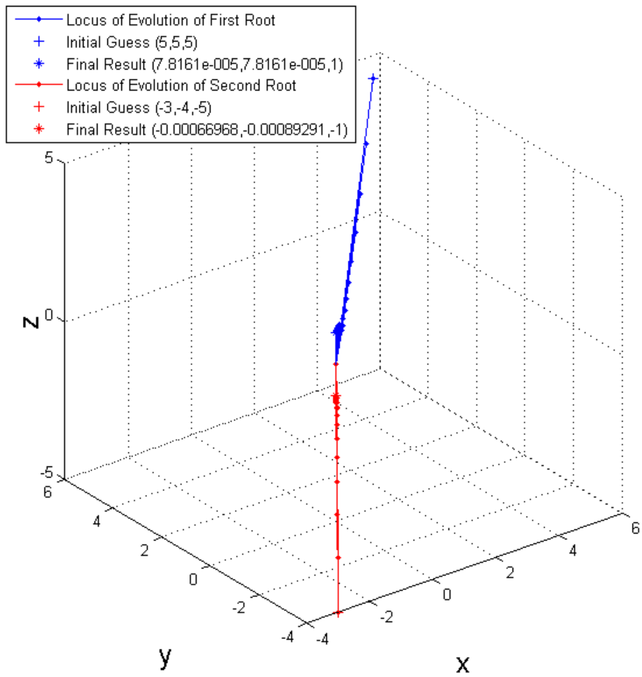

Figure 6: The locus of evolution for each root of Example 8.

3.9 Example 9

In this example, we demonstrate the performance of the scalar homotopy method while dealing with ill-posed problems. The nonlinear Fredhlom integral equation of the first-kind is considered. There is a limited literature dealing this problem, because the problem itself is highly ill-posed. In addition, the nonlinear behavior makes the amplification of errors more severe than in the linear system. The nonlinear integral equation we considered is written as

$$\int_0^1 x(s)\,x(t)\,dt = A\cos(\beta s)\,,\ A>0$$

where A and β are constants. We let A=1 and $\beta=3$ in the following example. In this example, we give data for $A\cos(\beta s)$ in the region $s\in[0,1]$ by equally dividing the region into 100 segments, that means a total of 101 data points are used. To solve the problem, we assume that the solution can be approximated by a polynomial

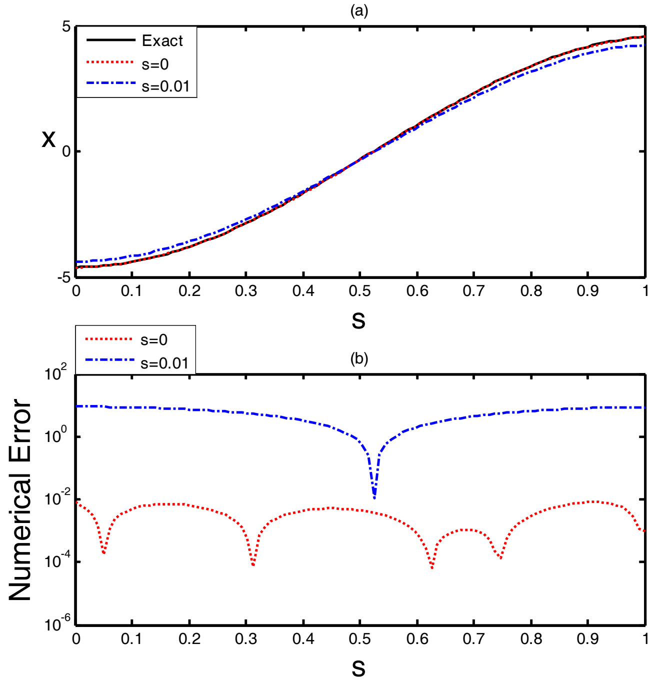

Figure 7: (a) The comparison of numerical and exact solutions, and (b) the numerical error of Example 9 for the first solution.

expansion as:

$$x(s) = c_0 + \sum_{k=1}^{p} c_k (s)^k$$

where p is the maximum power of polynomial expansion and c_k is the unknown coefficients. Two exact solutions exist: $x(s) = \pm\sqrt{\frac{A\beta}{\sin\beta}}\cos(\beta s) = \pm\sqrt{\frac{3}{\sin 3}}\cos(3s)$. (Polyanin and Manzhirov, 2007). We set p=10, it means that a total of 11 unknown coefficients need to be solved. It then can be seen here that the system is an over-determinated system. For a conventional numerical method such as the Newton's method, the vector homotopy method, and the fictitious time integration method (Liu and Atluri, 2008) all require that the number of equations should be equal to the number of unknowns. However, the scalar homotopy method does not have such an unnecessary constraint, and it can easily deal with the current situation. When we set the initial guess as: $c_0 = -1$, $c_k = 0$ when $k \neq 0$, the numerical results for 100,000 steps are illustrated in Fig. 7. It can be seen that the solution con-

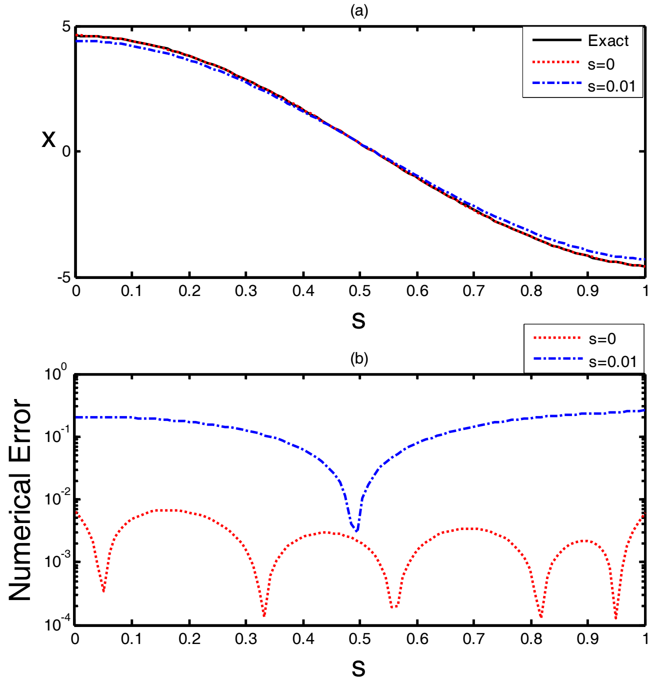

Figure 8: (a) The comparison of numerical and exact solutions, and (b) the numerical error of Example 9 for the second solution.

verges to $x(s) = -\sqrt{\frac{3}{\sin 3}}\cos(3s)$ and the scalar homotopy method can still have a reasonable result even when a random noise with the maximum absolute error of 0.01 is added in the data. When we set the initial guess as: $c_0 = 1$, $c_k = 0$ when $k \neq 0$, the numerical results for 100,000 steps are illustrated in Fig. 8. It can be seen that the solution now converges to $x(s) = \sqrt{\frac{3}{\sin 3}}\cos(3s)$. It is interesting that the current approach can deal with the highly ill-posed problem without using any regularization technique such as the Tikhonov's regularization method or the truncated singular value decomposition method. The current approach as well as the fictitious time integration method both do not need special treatments in dealing with ill-posed problems. However, the present scalar homotopy method does not have the constraint that the number of equations be equal to the number of unknowns, as in the fictitious time integration method, such that it is a more robust method in dealing with various kinds of systems of equations.

4 Conclusions

In this paper, a novel scalar homotopy method (SHM) is developed. In order to construct a system of first-order ODEs for the evolution of unknowns, a flow rule, in which the vector gradient of the scalar homotopy function is assigned as the evolution force, is introduced. The resulting dynamical system for the evolution of unknowns does not involve the inverse of matrix as that required by the Newton's method or vector homotopy method. Thus, the SHM can save a lot of computational time. The scalar homotopy method keeps the merits of the conventional vector homotopy method, such as the global convergence, but also suffers from its slow convergence speed like the vector homotopy method. In order to speed up the convergence, a "restart method" is introduced, which can work perfectly owing to the global convergence property of the homotopy method. Due to the mathematical structure of this scalar homotopy method, the constraint that requires the number of equations to be equal to the number of unknowns is no longer necessary. Nine examples were used to demonstrate the validity of the present method. Among these examples, several of them cannot be appropriately treated by using the conventional numerical methods, such as the vector homotopy method, the fictitious time integration method or the Newton method. The present method and the fictitious time integration method both have good convergence and accuracy, and are robust enough to solve the ill-posed problem under noisy data, but the fictitious time integration method cannot deal with the over/under-determined problem because of the limitation of its vector form. In one of the illustrated examples, the present method can find all solutions easily while the fictitious time integration method cannot find all of them, because the attracting zone of a root is too small such that initial guess is not easy to be chosen. In another example, the present method achieves better accuracy than the fictitious time integration method for the same evolution steps. Although we cannot draw a conclusion that the present method is better than the fictitious time integration method, it indeed shows some benefits, and further studies are necessary.

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.; Zhu, T. (1998): A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics. *Comp. Mech.*, vol. 22, pp. 117-127.

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.; Liu, C.-S.; Kuo C. L. (2009): A modified Newton method for solving non-linear algebraic equations. *J. Marine Sciences & Tech.*, vol. 17, pp. 238-247.

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.

Deuflhard, P. (2004): Newton Methods for Nonlinear 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. (1998): 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.

Krzyworzcka, S. (1996): Extension of the Lanczos and CGS methods to systems of non-linear equations. *J. Comp. Appl. Math.*, vol. 69, pp. 181-190.

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, pp. 173-190.

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. 53-65.

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. (2008): 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.; 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.; Chang, C. W. (2004): Lie group symmetry applied to the computation of convex plasticity constitutive equation. *CMES: Computer Modeling in Engineering & Sciences*, vol. 6, pp. 277-294.

Liu, C.-S.; Chang, C. W. (2009): Novel methods for solving severely ill-posed linear equations system. *J. Marine Sciences & Tech.*, vol. 17, pp. 216-227.

Mo, Y.; Liu, H.; Wang, Q. (2009): Conjugate direction particle swarm optimization solving systems of nonlinear equations. *Comp. Math. Appl.*, vol. 57, pp. 1877-1882.

Nazareth, J. L. (2003): Differentiable Optimization and Equation Solving. Springer-Verlag New York.

Polyanin, A. D.; Manzhirov, A. V. (2007): Handbook of Integral Equations, 2^{nd} edition, Chapman and Hall/CRC.

Ramm, A. G. (2007): Dynamical System Methods for Solving Operator Equations. Mathematics in Science and Engineering,Vol. 208 (series editor: Chu, C.K.), Elsevier, Amsterdam, Netherlands.

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.