 CMES, vol.84, no.6, pp.575-601, 2012

A Globally Optimal Iterative Algorithm Using the Best Descent Vector $\dot{\mathbf{x}} = \lambda[\alpha_c \mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}]$, with the Critical Value α_c, for Solving a System of Nonlinear Algebraic Equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$

Chein-Shan Liu[1] **and Satya N. Atluri**[2]

Abstract: An iterative algorithm based on the concept of *best descent vector* $\mathbf{u}$ in $\dot{\mathbf{x}} = \lambda\mathbf{u}$ is proposed to solve a system of nonlinear algebraic equations (NAEs): $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. In terms of the residual vector $\mathbf{F}$ and a monotonically increasing positive function $Q(t)$ of a time-like variable t, we define a **future cone** in the Minkowski space, wherein the discrete dynamics of the proposed algorithm evolves. A new method to approximate the **best descent vector** is developed, and we find a critical value of the weighting parameter α_c in the best descent vector $\mathbf{u} = \alpha_c\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}$, where $\mathbf{B} = \partial\mathbf{F}/\partial\mathbf{x}$ is the Jacobian matrix. We can prove that such an algorithm leads to the largest convergence rate with the descent vector given by $\mathbf{u} = \alpha_c\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}$; hence we label the present algorithm as a **globally optimal iterative algorithm** (GOIA). Some numerical examples are used to validate the performance of the GOIA; a very fast convergence rate in finding the solution is observed.

Keywords: Nonlinear algebraic equations, Future cone, Optimal Iterative Algorithm (OIA), Globally Optimal Iterative Algorithm (GOIA)

1 Introduction

In this paper we develop a very powerful purely iterative method to solve a system of nonlinear algebraic equations (NAEs): $F_i(x_1,\ldots,x_n) = 0,\ i = 1,\ldots,n$, or in their vector-form:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \tag{1}$$

where $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^n$ is a given vector function to find the unknown vector $\mathbf{x} \in \mathbb{R}^n$.

[1] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan. E-mail: liucs@ntu.edu.tw

[2] Center for Aerospace Research & Education, University of California, Irvine

Hirsch and Smale (1979) have derived the so-called continuous Newton method, governed by the following first-order nonlinear ODEs in a time-like variable t:

$$\dot{\mathbf{x}}(t) = -\mathbf{B}^{-1}(\mathbf{x})\mathbf{F}(\mathbf{x}), \tag{2}$$

where $\mathbf{B}$ is the Jacobian matrix with its ij-th component being given by $B_{ij} = \partial F_i/\partial x_j$. A discretization of Eq. (2) leads to the Newton iterative algorithm:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1}\mathbf{F}_k, \tag{3}$$

where $\mathbf{x}_k$ is the k-th iterative value of the unknown vector $\mathbf{x}$, $\mathbf{F}_k = \mathbf{F}(\mathbf{x}_k)$ and $\mathbf{B}_k = \mathbf{B}(\mathbf{x}_k) \in \mathbb{R}^{n\times n}$. It can be seen that the Newton iterative algorithm in Eq. (3) is quite difficult to compute, because it involves an inverse matrix $\mathbf{B}_k^{-1}$. Usually the vector $\mathbf{B}_k^{-1}\mathbf{F}_k$ cannot be derived in a closed-form in practice, and moreover it is very difficult to be computed in a numerical scheme. However, in this paper we will raise the position of the descent vector

$$\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}, \quad \text{(best vector)} \tag{4}$$

as to be the **best descent direction** in searching the solution $\mathbf{x}$ of Eq. (1), and guide our theoretical development of novel optimal iterative algorithm to solve Eq. (1).

In order to eliminate the need for finding and inverting the Jacobian matrix in the Newton iteration procedure, Liu and Atluri (2008) have proposed an alternate first-order system of nonlinear ODEs:

$$\dot{\mathbf{x}} = -\frac{\nu}{q(t)}\mathbf{F}(\mathbf{x}), \tag{5}$$

where ν is a nonzero constant and $q(t)$ may in general be a monotonically increasing function of t. In their approach, the term $\nu/q(t)$ plays the major role of being a stabilizing controller to help one obtain a solution even for a bad initial guess of the solution, and speeds up the convergence. Liu and his coworkers [Liu (2008, 2009a, 2009b, 2009c); Liu and Chang (2009)] showed that high performance can be achieved by using the above fictitious time integration method (FTIM) together with the group-preserving scheme [Liu (2001)]. In spite of its success, the FTIM has only a local convergence. Atluri, Liu and Kuo (2009) have combined the above two methods, and proposed a modification of the Newton method, which does not need the inversion of B_{ij}.

To remedy the shortcoming of the vector homotopy method as initiated by Davidenko (1953), Liu, Yeih, Kuo and Atluri (2009) have proposed a scalar homotopy method (SHM) with the following evolution equation for $\mathbf{x}$:

$$\dot{\mathbf{x}} = -\frac{\frac{\partial h}{\partial t}}{\|\frac{\partial h}{\partial \mathbf{x}}\|^2}\frac{\partial h}{\partial \mathbf{x}}, \tag{6}$$

where

$$h(\mathbf{x},t) = \frac{1}{2}[t\|\mathbf{F}(\mathbf{x})\|^2 - (1-t)\|\mathbf{x}\|^2], \tag{7}$$

$$\frac{\partial h}{\partial t} = \frac{1}{2}[\|\mathbf{F}(\mathbf{x})\|^2 + \|\mathbf{x}\|^2], \tag{8}$$

$$\frac{\partial h}{\partial \mathbf{x}} = t\mathbf{B}^{\mathrm{T}}\mathbf{F} - (1-t)\mathbf{x}. \tag{9}$$

Ku, Yeih and Liu (2010) have combined this idea with an exponentially decaying scalar homotopy function, and developed a manifold-based exponentially convergent algorithm (MBECA):

$$\dot{\mathbf{x}} = -\frac{\nu}{(1+t)^m}\frac{\|\mathbf{F}\|^2}{\|\mathbf{B}^{\mathrm{T}}\mathbf{F}\|^2}\mathbf{B}^{\mathrm{T}}\mathbf{F}. \tag{10}$$

As pointed out by Liu and Atluri (2011a), two major drawbacks appeared in the MBECA: irregular bursts and flattened behavior appear in the residual-error curve such that it never satisfies a specified moderate convergence criterion.

For the development of stable and convergent numerical algorithms to solve NAEs, it is of utmost importance to build a framework to define the evolution equations on a suitable manifold. Liu, Yeih, Kuo and Atluri (2009) were the first to construct a manifold based on a scalar homotopy function, and a method based on the concept of "normality" and "consistency condition" is derived for solving the NAEs. Unfortunately, while such an algorithm is globally convergent, its convergence speed is terribly slow. Later, Ku, Yeih and Liu (2010) and Liu and Atluri (2011a) first constructed a more relevant manifold model directly based on the Euclidean-norm $\|\mathbf{F}\|$ of the residual vector $\mathbf{F}(\mathbf{x})$ in Eq. (1):

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

where $Q(t) > 0$ is a monotonically increasing function of t, and C is a constant. However, Liu and Atluri (2011a) pointed out the limitations of the $\|\mathbf{F}\|$-based with time-increment specified algorithms, and proposed three new algorithms which are purely iterative in nature.

To further improve the convergence of the solution for $\mathbf{x}$, Liu and Atluri (2011b) used the same hyper-surface as in Eq. (11), but modified the evolution equation for $\dot{\mathbf{x}}$ as a "**non-normal**" relation, involving both $\mathbf{F}$ and $\mathbf{R} := \mathbf{B}^{\mathrm{T}}\mathbf{F}$:

$$\dot{\mathbf{x}} = \lambda\mathbf{u} = \lambda[\alpha\mathbf{F} + (1-\alpha)\mathbf{R}], \tag{12}$$

where λ *is a preset multiplier determined by the "consistency condition"*, and α is a parameter. Liu and Atluri (2011b) proposed a way to optimize α in order

to achieve the best convergence. With the optimized value for α, Liu and Atluri (2011b) derived an optimal vector driven algorithm (OVDA) according to

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}} [\alpha\mathbf{F} + (1-\alpha)\mathbf{R}], \tag{13}$$

where

$$\begin{aligned} &\mathbf{A} := \mathbf{B}\mathbf{B}^{\mathrm{T}}, \\ &\mathbf{v} = \mathbf{B}\mathbf{u} = \mathbf{v}_1 + \alpha\mathbf{v}_2 = \mathbf{A}\mathbf{F} + \alpha(\mathbf{B}-\mathbf{A})\mathbf{F}, \\ &\alpha = \frac{[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2] \cdot \mathbf{v}_1}{[\mathbf{v}_2, \mathbf{F}, \mathbf{v}_1] \cdot \mathbf{v}_2}. \end{aligned} \tag{14}$$

Here $[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2] = \mathbf{v}_1 \cdot \mathbf{F}\mathbf{v}_2 - \mathbf{v}_2 \cdot \mathbf{F}\mathbf{v}_1$ is a Jordan algebra defined by Liu (2000).
Liu and Kuo (2011) have adopted a different strategy by using

$$\dot{\mathbf{x}} = \lambda\mathbf{u} = \lambda[\alpha\mathbf{x} + \mathbf{R}], \tag{15}$$

and minizing the following quantity

$$a_0 := \frac{\|\mathbf{F}\|^2 \|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2} \geq 1, \tag{16}$$

to obtain another optimal vector driven algorithm (OVDA) to solve Eq. (1).
In a continuing effort to accelerate the convergence of an optimal iterative algorithm (OIA) for finding the solution $\mathbf{x}$, Liu, Dai and Atluri (2011a) proposed another non-normal descent evolution equation for $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = \lambda[\alpha\mathbf{R} + \beta\mathbf{p}], \tag{17}$$

where

$$\mathbf{p} = \left[\mathbf{I}_n - \frac{\|\mathbf{R}\|^2}{\mathbf{R}^{\mathrm{T}}\mathbf{C}\mathbf{R}}\mathbf{C}\right]\mathbf{R}, \tag{18}$$

in which

$$\mathbf{C} = \mathbf{B}^{\mathrm{T}}\mathbf{B}, \tag{19}$$

such that, clearly, $\mathbf{p}$ is orthogonal to $\mathbf{R}$. Thus Liu, Dai and Atluri (2011a) derived:

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}(\alpha\mathbf{B}\mathbf{R} + \beta\mathbf{B}\mathbf{p})} [\alpha\mathbf{R} + \beta\mathbf{p}], \tag{20}$$

and optimized α and $\beta(=1-\alpha)$ to achieve a faster convergence optimal iterative algorithm (OIA) for the iterative solution for $\mathbf{x}$.

Liu, Dai and Atluri (2011a) also explored an alternative descent relation:

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}(\alpha\mathbf{B}\mathbf{F}+\beta\mathbf{B}\mathbf{p}^*)}[\alpha\mathbf{F}+\beta\mathbf{p}^*], \tag{21}$$

where $\mathbf{p}^*$ is orthogonal to $\mathbf{F}$,

$$\mathbf{p}^* = \left[\mathbf{I}_n - \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{C}\mathbf{F}}\mathbf{C}\right]\mathbf{F}. \tag{22}$$

It was shown by Liu, Dai and Atluri (2011a) that the OIAs based on Eqs. (20) and (21), namely the OIA/ODV[R] and OIA/ODV[F], had the fastest convergence and best accuracy as compared to any algorithm published in the previous literature.

It can be seen that neither the vector $\mathbf{p}$ defined in Eq. (18) and which is normal to $\mathbf{R}$, nor the vector $\mathbf{p}^*$ defined in Eq. (22) and which is normal to $\mathbf{F}$, are unique. Liu, Dai and Atluri (2011b) have considered alternate vectors $\mathbf{P}$ and $\mathbf{P}^*$ which are also normal to $\mathbf{R}$ and $\mathbf{F}$, respectively, as follows:

$$\mathbf{P} := \mathbf{F} - \frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{R}\|^2}\mathbf{R}, \tag{23}$$

$$\mathbf{P}^* := \mathbf{R} - \frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{F}\|^2}\mathbf{F}, \tag{24}$$

such that, clearly, $\mathbf{R}\cdot\mathbf{P}=0$ and $\mathbf{F}\cdot\mathbf{P}^*=0$. Using the relations as in Eqs. (23) and (24), Liu, Dai and Atluri (2011b) have adopted the following evolution equations for $\dot{\mathbf{x}}$:

$$\dot{\mathbf{x}} = \lambda[\alpha\mathbf{R}+\beta\mathbf{P}], \tag{25}$$

$$\dot{\mathbf{x}} = \lambda[\alpha\mathbf{F}+\beta\mathbf{P}^*]. \tag{26}$$

Liu, Dai and Atluri (2011b) showed that with the algorithms ODV(R) and ODV(F) proposed, they were able to achieve the fastest convergence, as well as the best accuracy, so far, for iteratively solving a system of nonlinear algebraic equations (NAEs): $\mathbf{F}(\mathbf{x})=\mathbf{0}$, without the need for inverting the Jacobian matrix $\mathbf{B}=\partial\mathbf{F}/\partial\mathbf{x}$.

Motivated by the Newton iterative algorithm, which led to $a_0=1$ by inserting Eq. (4) for $\mathbf{u}$ *and* $\mathbf{v}=\mathbf{B}\mathbf{u}=\mathbf{F}$ *into Eq. (16), in this paper we further develop a more powerful* **Globally Optimal Iterative Algorithm** *to solve the NAEs, which provides the global minimum of a_0 but without resorting on the inversion $\mathbf{B}^{-1}$ of the Jacobian matrix* $\mathbf{B}$.

The remainder of this paper is arranged as follows. In Section 2 we give a theoretical basis of the present algorithm. We start from a future cone defined in terms of the residual-norm and a monotonically increasing function $Q(t) > 0$, and arrive at a system of ODEs driven by a descent vector, which is a combination of the vectors $\mathbf{F}$ and $\mathbf{B}^{\mathrm{T}}\mathbf{F}$. Section 3 is devoted to deriving a scalar equation to keep the discretely iterative orbit on the future cone, and then we propose two new concepts of *optimal descent vector* and *best descent vector* to select the optimal and critical parameters α_o and α_c, which automatically have a convergent behavior of the residual-error curve. When the optimal descent vector leads to a local minimum of a_0, the best descent vector gives a global optimal iterative algorithm with the global minimum of a_0. In Section 4 we give several numerical examples to test the present algorithms with different weighting parameters α_o and α_c. Finally, the many advantages of the newly developed algorithms are emphasized in Section 5.

2 A future cone and a new evolution equation for $\dot{\mathbf{x}}$

For the nonlinear algebraic equations (NAEs) in Eq. (1), we can formulate a scalar Newton homotopy function:

$$h(\mathbf{x},t) = \frac{1}{2}Q(t)\|\mathbf{F}(\mathbf{x})\|^2 - \frac{1}{2}\|\mathbf{F}(\mathbf{x}_0)\|^2 = 0, \tag{27}$$

where we let $\mathbf{x}$ be a function of a fictitious time-like variable t, and its initial value is $\mathbf{x}(0) = \mathbf{x}_0$.

In terms of

$$\mathbf{X} = \begin{bmatrix} \frac{\mathbf{F}}{\|\mathbf{F}(\mathbf{x}_0)\|} \\ \frac{1}{\sqrt{Q(t)}} \end{bmatrix}, \tag{28}$$

Eq. (27) represents a cone:

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

in the Minkowski space $\mathbb{M}^{n+1}$, endowed with an indefinite Minkowski metric tensor:

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

where $\mathbf{I}_n$ is the $n \times n$ identity matrix. Because the last component $1/\sqrt{Q(t)}$ of $\mathbf{X}$ is positive, the cone in Eq. (29) is a future cone [Liu (2001)], which is schematically shown in Fig. 1.

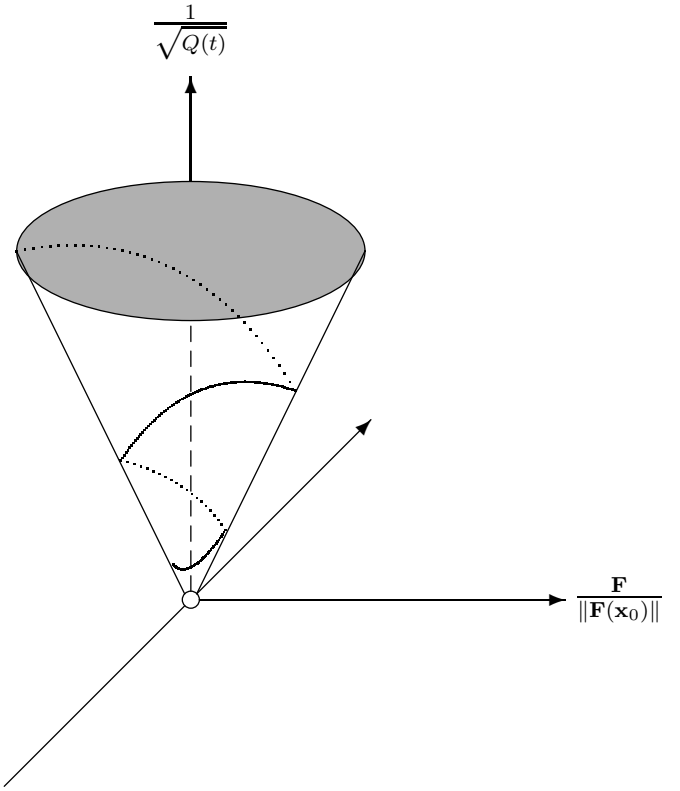

Figure 1: The construction of cone in the Minkowski space for solving nonlinear algebraic equations system signifies a conceptual breakthrough.

We expect $h(\mathbf{x},t)=0$ to be an invariant manifold in the space of $(\mathbf{x},t)$ for a dynamical system $h(\mathbf{x}(t),t)=0$ to be specified further. With the assumption of $Q>0$, the manifold defined by Eq. (27) is continuous, and thus the following operation of differential carried out on the manifold makes sense. As a consistency condition, by taking the time differential of Eq. (27) with respect to t and considering $\mathbf{x}=\mathbf{x}(t)$, we have

$$\frac{1}{2}\dot{Q}(t)\|\mathbf{F}(\mathbf{x})\|^2+Q(t)(\mathbf{B}^{\mathrm{T}}\mathbf{F})\cdot\dot{\mathbf{x}}=0. \tag{31}$$

We suppose that the evolution of $\mathbf{x}$ is driven by a vector $\mathbf{u}$:

$$\dot{\mathbf{x}}=\lambda\mathbf{u}, \tag{32}$$

where λ in general is a scalar function of t, and

$$\mathbf{u}=\alpha\mathbf{F}+\mathbf{B}^{\mathrm{T}}\mathbf{F} \tag{33}$$

is a suitable combination of the residual vector $\mathbf{F}$ with the gradient vector $\mathbf{B}^{\mathrm{T}}\mathbf{F}$, and α is a weighting parameter to be optimized below. As compared to Eqs. (5) and (10), the vector field in Eq. (32) is a compromise between $\mathbf{F}$ and $\mathbf{B}^{\mathrm{T}}\mathbf{F}$ as shown in Eq. (33).

Inserting Eq. (32) into Eq. (31) we can derive

$$\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{u}, \tag{34}$$

where

$$\mathbf{A} := \mathbf{B}\mathbf{B}^{\mathrm{T}}, \tag{35}$$

$$\mathbf{v} := \mathbf{B}\mathbf{u} = \mathbf{v}_1 + \alpha\mathbf{v}_2 = \mathbf{A}\mathbf{F} + \alpha\mathbf{B}\mathbf{F}, \tag{36}$$

$$q(t) := \frac{\dot{Q}(t)}{2Q(t)}. \tag{37}$$

Hence, in our algorithm if $Q(t)$ can be guaranteed to be a monotonically increasing function of t, we have an absolutely convergent property in solving the NAEs in Eq. (1):

$$\|\mathbf{F}(\mathbf{x})\|^2 = \frac{C}{Q(t)}, \tag{38}$$

where

$$C = \|\mathbf{F}(\mathbf{x}_0)\|^2 \tag{39}$$

is determined by the initial value $\mathbf{x}_0$. When t is large, the above equation will enforce the residual error $\|\mathbf{F}(\mathbf{x})\|$ to tend to zero, and meanwhile the solution of Eq. (1) is obtained approximately, as shown in Fig. 1 schematically.

3 Dynamics of the optimal iterative algorithm

3.1 Discretizing and keeping x *on the manifold*

Now we discretize the foregoing continuous time dynamics embodied in Eq. (34) into a discrete time dynamics:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \beta\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{u}, \tag{40}$$

where

$$\beta = q(t)\Delta t. \tag{41}$$

Eq. (40) is obtained from the ODEs in Eq. (34) by applying the Euler scheme.

In order to keep $\mathbf{x}$ on the manifold defined by Eq. (38), we can consider the evolution of $\mathbf{F}$ along the path $\mathbf{x}(t)$ by

$$\dot{\mathbf{F}} = \mathbf{B}\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{v}. \tag{42}$$

Similarly we use the Euler scheme to integrate Eq. (42), obtaining

$$\mathbf{F}(t+\Delta t) = \mathbf{F}(t) - \beta\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{v}, \tag{43}$$

of which taking the square-norms of both the sides and using Eq. (38) we can obtain

$$\frac{C}{Q(t+\Delta t)} = \frac{C}{Q(t)} - 2\beta\frac{C}{Q(t)} + \beta^2\frac{C}{Q(t)}\frac{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2}. \tag{44}$$

Thus, by dividing both the sides by $C/Q(t)$ we can derive the following scalar equation:

$$a_0\beta^2 - 2\beta + 1 - \frac{Q(t)}{Q(t+\Delta t)} = 0, \tag{45}$$

where

$$a_0 := \frac{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2} \geq 1, \tag{46}$$

by using the Cauchy-Schwarz inequality.

As a result $h(\mathbf{x},t) = 0$, $t \in \{0,1,2,\ldots\}$ remains to be an invariant manifold in the space of $(\mathbf{x},t)$ for the discrete time dynamical system $h(\mathbf{x}(t),t) = 0$, which will be explored further in the next two sections. Liu and Atluri (2011a) first derived the formulae (45) and (46) for a purely gradient-vector driven dynamical system.

3.2 A discrete dynamics

Let

$$s = \frac{Q(t)}{Q(t+\Delta t)} = \frac{\|\mathbf{F}(\mathbf{x}(t+\Delta t))\|^2}{\|\mathbf{F}(\mathbf{x}(t))\|^2}, \tag{47}$$

which is an important quantity to assess the convergence property of numerical algorithm for solving NAEs. If s can be guaranteed to be $s < 1$, then the residual error $\|\mathbf{F}\|$ will be decreased step-by-step in the iteration process.

From Eqs. (45) and (47) we can obtain

$$a_0\beta^2 - 2\beta + 1 - s = 0, \tag{48}$$

of which we can take the solution of β to be

$$\beta = \frac{1-\sqrt{1-(1-s)a_0}}{a_0}, \text{ if } 1-(1-s)a_0 \geq 0. \tag{49}$$

Let

$$1-(1-s)a_0 = \gamma^2 \geq 0, \tag{50}$$

$$s = 1 - \frac{1-\gamma^2}{a_0}; \tag{51}$$

and the condition $1-(1-s)a_0 \geq 0$ in Eq. (49) is automatically satisfied, leading to

$$\beta = \frac{1-\gamma}{a_0}. \tag{52}$$

From Eqs. (40), (46) and (52) we can obtain the following algorithm:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - (1-\gamma)\frac{\mathbf{F}^{\mathrm{T}}\mathbf{v}}{\|\mathbf{v}\|^2}\mathbf{u}, \tag{53}$$

where $0 \leq \gamma < 1$ is a parameter. Under the above condition we have

$$\text{Convergence Rate} := \frac{\|\mathbf{F}(t)\|}{\|\mathbf{F}(t+\Delta t)\|} = \frac{1}{\sqrt{s}} > 1. \tag{54}$$

This property is very important, since it guarantees the new algorithm to be absolutely convergent to the true solution.

3.3 The algorithm driven by the optimal descent vector $\mathbf{u} = \alpha_o\mathbf{F} + \mathbf{B}^T\mathbf{F}$

The algorithm (53) does not specify how to choose the parameter α. We can determine a suitable α such that s defined by Eq. (51) is minimized with respect to α, because a smaller s will lead to a faster convergence as shown in Eq. (54).

Thus by inserting Eq. (46) for a_0 into Eq. (51) we can write s to be

$$s = 1 - \frac{(1-\gamma^2)(\mathbf{F}\cdot\mathbf{v})^2}{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}, \tag{55}$$

where $\mathbf{v}$ as defined by Eq. (36) includes a parameter α. Let $\partial s/\partial\alpha = 0$, and through some algebraic operations we can solve α by

$$\alpha_o = \frac{[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2]\cdot\mathbf{v}_1}{[\mathbf{v}_2, \mathbf{F}, \mathbf{v}_1]\cdot\mathbf{v}_2}, \tag{56}$$

where

$$[\mathbf{a}, \mathbf{b}, \mathbf{c}] = (\mathbf{a}\cdot\mathbf{b})\mathbf{c} - (\mathbf{c}\cdot\mathbf{b})\mathbf{a}, \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n \tag{57}$$

is defined by Liu (2000) for a Jordan algebra. Inserting the above α_o into Eq. (33) we can obtain the optimal vector:

$$\mathbf{u} = \alpha_o \mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}. \quad (\textit{optimal descent vector}) \tag{58}$$

Hence, we can develop the following *optimal iterative algorithm* (OIA), involving an *optimal descent vector* (ODV), for solving the NAEs in Eq. (1):

(i) Select $0 \le \gamma < 1$, and give an initial $\mathbf{x}_0$.

(ii) For $k = 0, 1, 2\ldots$, we repeat the following computations:

$$\begin{aligned}
&\mathbf{v}_1^k = \mathbf{A}_k\mathbf{F}_k,\\
&\mathbf{v}_2^k = \mathbf{B}_k\mathbf{F}_k,\\
&\alpha_o^k = \frac{[\mathbf{v}_1^k, \mathbf{F}_k, \mathbf{v}_2^k]\cdot\mathbf{v}_1^k}{[\mathbf{v}_2^k, \mathbf{F}_k, \mathbf{v}_1^k]\cdot\mathbf{v}_2^k}, \quad (\text{optimal } \alpha_o^k)\\
&\mathbf{u}_k = \alpha_o^k\mathbf{F}_k + \mathbf{B}_k^{\mathrm{T}}\mathbf{F}_k, \quad (\text{optimal descent vector})\\
&\mathbf{v}_k = \mathbf{v}_1^k + \alpha_o^k\mathbf{v}_2^k,
\end{aligned} \tag{59}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma)\frac{\mathbf{F}_k\cdot\mathbf{v}_k}{\|\mathbf{v}_k\|^2}\mathbf{u}_k. \tag{60}$$

If $\mathbf{x}_{k+1}$ converges according to a given stopping criterion $\|\mathbf{F}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii). The above algorithm is named the OIA/ODV, which was first developed by Liu and Atluri (2011c) to solve linear algebraic equations.

3.4 A critical value for α

Previously, Liu and Atluri (2011b) have derived the optimal α in the descent vector $\mathbf{u} = \alpha\mathbf{F} + (1-\alpha)\mathbf{B}^{\mathrm{T}}\mathbf{F}$ by letting $\partial s/\partial\alpha = 0$. Also, in Section 3.3 we have used $\partial s/\partial\alpha = 0$ (or equivalently, $\partial a_0/\partial\alpha = 0$) to find the optimal value of α in the

descent vector $\mathbf{u} = \alpha\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}$. Usually, this value of α obtained from $\partial s/\partial\alpha = 0$ is not *the global minimum* of a_0 (or s), and instead of *a local minimum. Here, we try another approach and attempt to develop a more powerful selection principle of* α*, such that the value of* α *obtained is the global minimum of* a_0 *(or s)*; however, before that we give a remark about the best choice of the descent vector $\mathbf{u}$.

Remark 1: The best choice of $\mathbf{u}$ would be $\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}$, which by Eq. (36) leads to $\mathbf{v} = \mathbf{F}$, and by Eq. (46) further leads to the smallest value of $a_0 = 1$. If $\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}$ is realizable, we have $s = \gamma^2$ by Eq. (51), and thus from Eq. (54) we have an infinite convergence rate by letting $\gamma = 0$, which can find the solution with one step. In this regard $\mathbf{B}^{-1}\mathbf{F}$ is the *best vector* and is the *best descent direction* for a numerical algorithm used to solve Eq. (1). The Newton iterative algorithm in Eq. (3) is of this sort; however, it is very difficult to be realized in a numerical scheme because $\mathbf{B}^{-1}$ is hard to find.

Below, according to the following equivalent relation:

$$a_0 = 1 \quad \equiv \quad \mathbf{u} = \mathbf{B}^{-1}\mathbf{F}, \tag{61}$$

we introduce a new method to approximate the *best vector* in our framework of optimal iterative algorithm.

Motivated by the above remark about the best vector, which is very difficult to realize in practice, we can slightly relax the requirement of the value of a_0 to be 1. It means that we can relax the choice of $\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}$ by Eq. (61). Instead of, we can determine a suitable α such that a_0 defined by Eq. (46) takes a suitable small value $a_s > 1$. That is,

$$a_0 := \frac{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2} = a_s. \tag{62}$$

When a_s is near to 1, the convergence speed is very fast. Inserting Eq. (36) for $\mathbf{v}$ into the above equation, and through some elementary operations we can derive a quadratic equation to solve α:

$$e_1\alpha^2 + e_2\alpha + e_3 = 0, \tag{63}$$

where

$$e_1 := \|\mathbf{F}\|^2\|\mathbf{v}_2\|^2 - a_s(\mathbf{F}\cdot\mathbf{v}_2)^2, \tag{64}$$

$$e_2 := 2\|\mathbf{F}\|^2\mathbf{v}_1\cdot\mathbf{v}_2 - 2a_s\mathbf{F}\cdot\mathbf{v}_1\mathbf{F}\cdot\mathbf{v}_2, \tag{65}$$

$$e_3 := \|\mathbf{F}\|^2\|\mathbf{v}_1\|^2 - a_s(\mathbf{F}\cdot\mathbf{v}_1)^2. \tag{66}$$

If the following condition is satisfied

$$D := e_2^2 - 4e_1e_3 \geq 0, \tag{67}$$

then α has a **real solution**:

$$\alpha = \frac{\sqrt{D} - e_2}{2e_1}. \tag{68}$$

Inserting Eqs. (64)-(66) into the **critical equation**:

$$D = e_2^2 - 4e_1e_3 = 0, \tag{69}$$

we can derive an algebraic equation to determine which a_s is the lowest bound of Eq. (67), of which the equality holds. *In this lowest bound a_s is a critical value denoted by a_c*, and for all $a_s \geq a_c$ it can satisfy Eq. (67). From Eq. (69) through some operations, the critical value a_c can be solved as:

$$a_c = \frac{\|\mathbf{F}\|^2[\|\mathbf{v}_1\|^2\|\mathbf{v}_2\|^2 - (\mathbf{v}_1 \cdot \mathbf{v}_2)^2]}{\|[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2]\|^2}. \tag{70}$$

Then insert it for a_s into Eqs. (64) and (65) we can obtain a **critical value** α_c for α from Eq. (68):

$$\alpha_c = \frac{a_c\mathbf{F} \cdot \mathbf{v}_1\mathbf{F} \cdot \mathbf{v}_2 - \|\mathbf{F}\|^2\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{F}\|^2\|\mathbf{v}_2\|^2 - a_c(\mathbf{F} \cdot \mathbf{v}_2)^2}, \tag{71}$$

where $D = 0$ was used in view of Eq. (69).

Here we must stress that in the current frame of the descent vector $\mathbf{u} = \alpha\mathbf{F} + \mathbf{B}^T\mathbf{F}$, the above value α_c is the **best one**, *and the vector*

$$\mathbf{u} = \alpha_c\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F} \ \text{(best descent vector)} \tag{72}$$

is the **best descent vector**, *which is in practice the best approximation to the real best vector $\mathbf{u} = \mathbf{B}^{-1}\mathbf{F}$ used in the Newton iterative algorithm.* **Due to its criticality, if one attempts to find a better value than α_c, there would be no real solution of α**. Furthermore, the present **best descent vector** is also better than the **optimal descent vector** $\mathbf{u} = \alpha_o\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}$ derived in Eq. (58), as being a search direction of the solution for the NAEs in Eq. (1).

3.5 The present algorithm driven by the best descent vector $\mathbf{u} = \alpha_c\mathbf{F} + \mathbf{B}^T\mathbf{F}$, without inverting **B**

Then, we can derive the following *Globally Optimal Iterative Algorithm* (GOIA) to solve the NAEs in Eq. (1):

(i) Select $0 \leq \gamma < 1$, and give an initial guess of $\mathbf{x}_0$.

(ii) For $k = 0, 1, 2 \ldots$ we repeat the following computations:

$$\begin{aligned}
&\mathbf{v}_1^k = \mathbf{A}_k \mathbf{F}_k, \\
&\mathbf{v}_2^k = \mathbf{B}_k \mathbf{F}_k, \\
&a_c^k = \frac{\|\mathbf{v}_1^k\|^2 \|\mathbf{v}_2^k\|^2 - (\mathbf{v}_1^k \cdot \mathbf{v}_2^k)^2}{\|[\mathbf{v}_1^k, \mathbf{F}_k, \mathbf{v}_2^k]\|^2}, \\
&\alpha_c^k = \frac{a_c^k \mathbf{F}_k \cdot \mathbf{v}_1^k \mathbf{F}_k \cdot \mathbf{v}_2^k - \mathbf{v}_1^k \cdot \mathbf{v}_2^k}{\|\mathbf{v}_2^k\|^2 - a_c^k (\mathbf{F}_k \cdot \mathbf{v}_2^k)^2}, \ \text{(critical } \alpha_c^k\text{)} \qquad (73) \\
&\mathbf{u}_k = \alpha_c^k \mathbf{F}_k + \mathbf{B}_k^{\mathrm{T}} \mathbf{F}_k, \ \text{(best descent vector)} \\
&\mathbf{v}_k = \mathbf{v}_1^k + \alpha_c^k \mathbf{v}_2^k, \\
&\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma) \frac{\mathbf{F}_k \cdot \mathbf{v}_k}{\|\mathbf{v}_k\|^2} \mathbf{u}_k. \qquad (74)
\end{aligned}$$

If $\mathbf{x}_{k+1}$ converges according to a given stopping criterion $\|\mathbf{F}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii). In the above, we have eliminated the common term $\|\mathbf{F}_k\|^2$ in Eqs. (70) and (71).

Remark 2: In the iterative process it may happen that

$$\mathbf{F}_k \cdot \mathbf{v}_k = \mathbf{F}_k^{\mathrm{T}} [\mathbf{B}_k \mathbf{B}_k^{\mathrm{T}} + \alpha_c^k \mathbf{B}_k] \mathbf{F}_k = 0, \qquad (75)$$

such that the above iterative algorithm stagnates at a one point, i.e.,

$$\mathbf{x}_{k+1} = \mathbf{x}_k. \qquad (76)$$

There are two cases which may cause the above situation; one is $\mathbf{F}_k = \mathbf{0}$ and another is that the matrix $\mathbf{B}_k \mathbf{B}_k^{\mathrm{T}} + \alpha_c \mathbf{B}_k$ is singular at that point. In the first case of $\mathbf{F}_k = \mathbf{0}$ the solution is already obtained, and we do not need a special treatment of it. In the second case we can set $\alpha_c^k = 0$ when $\mathbf{F}_k \cdot \mathbf{v}_k$ is very small, say $\mathbf{F}_k \cdot \mathbf{v}_k \leq 10^{-15}$, and re-run this step; such that we have

$$\mathbf{F}_k \cdot \mathbf{v}_k = \mathbf{F}_k^{\mathrm{T}} \mathbf{B}_k \mathbf{B}_k^{\mathrm{T}} \mathbf{F}_k = \|\mathbf{B}_k^{\mathrm{T}} \mathbf{F}_k\|^2 > 0. \qquad (77)$$

This special treatment of the singularity of the matrix $\mathbf{B}_k \mathbf{B}_k^{\mathrm{T}} + \alpha_c^k \mathbf{B}_k$ can improve the stagnation of the iterative sequence of $\mathbf{x}_k$. However, in all the computations below we not yet face this situation.

In summary, we have derived a *thoroughly novel global algorithm* for solving the NAEs in Eq. (1). *In the present frame of future cone and the optimization in terms of the descent vector* $\mathbf{u} = \alpha \mathbf{F} + \mathbf{B}^T \mathbf{F}$*, the GOIA is the best one, which leads to the*

global minimum *of a_0 (or s), and hence the* **largest convergence rate**. While the parameter γ is chosen by the user and is problem-dependent, the parameter α_c^k is exactly given by Eq. (73). Up to here we have successfully derived a *drastically novel best descent vector algorithm*, which without the help from the formulae in Eqs. (46), (70) and (71) we cannot achieve such a wonderful result.

4 Numerical examples

In this section we apply the new method of GOIA to solve some nonlinear problems. In order to reveal the superior performance of the GOIA, we compare some numerical results with those calculated by the OIA/ODV, and also in some cases, we will compare the present algorithms with the Newton iterative algorithm.

4.1 Example 1

We revisit the following two-variable nonlinear equations [Hirsch and Smale (1979)]:

$$F_1(x,y) = x^3 - 3xy^2 + a_1(2x^2+xy) + b_1y^2 + c_1x + a_2y = 0, \tag{78}$$

$$F_2(x,y) = 3x^2y - y^3 - a_1(4xy - y^2) + b_2x^2 + c_2 = 0, \tag{79}$$

where $a_1 = 25$, $b_1 = 1$, $c_1 = 2$, $a_2 = 3$, $b_2 = 4$ and $c_2 = 5$.

This equation has been studied by Liu and Atluri (2008) by using the fictitious time integration method (FTIM), and then by Liu, Yeih and Atluri (2010) by using the multiple-solution fictitious time integration method (MSFTIM). Liu and Atluri (2008) found three solutions by guessing three different initial values, and Liu, Yeih and Atluri (2010) found four solutions. Liu and Atluri (2011b) applied the optimal vector driven algorithm (OVDA) to solve this problem, and they found the fifth solution.

Starting from an initial value of $(x_0,y_0) = (10,10)$ we solve this problem by using the OIA/ODV and GOIA with $\gamma = 0.25$ and under a convergence criterion $\varepsilon = 10^{-10}$. The residual errors of (F_1,F_2) are all smaller than 10^{-10}. It tends to the second root $(x,y) = (0.6277425, 22.2444123)$ through 98 iterations by the GOIA, and 135 iterations by the OIA/ODV.

Starting from the same initial value of $(x_0,y_0) = (10,10)$, and under the same convergence criterion 10^{-10}, the Newton iterative algorithm converges with 506 iterations to the fifth root $(x,y) = (1.6359718, 13.8476653)$. We compare in Fig. 2(a) the solution paths and in Fig. 2(b) the residual errors of the above three methods, from which we can see that the solution path of the Newton iterative algorithm spends

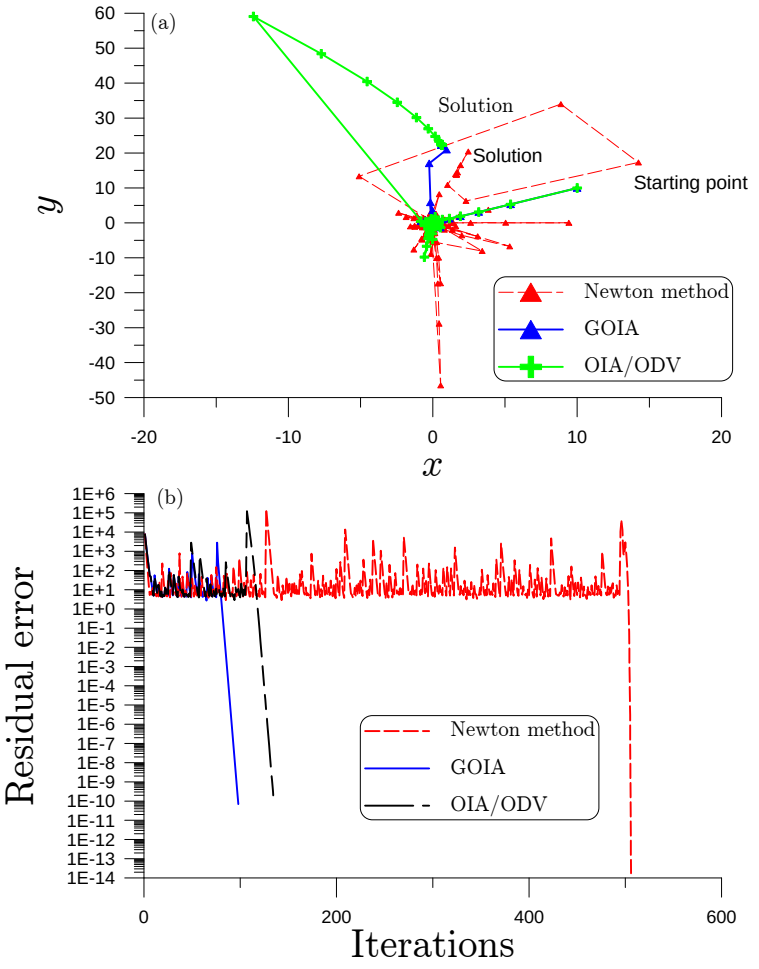

Figure 2: For example 1, solved by the GOIA, OIA/ODV and the Newton iterative, comparing (a) the solution paths, and (b) the residual errors.

many steps around the zero point and is much irregular than the solution path generated by the OIA/ODV and GOIA. From the solution paths as shown in Fig. 2(a), and the residual errors as shown in Fig. 2(b) we can observe that the mechanism to search solution has three stages: a mild convergence stage, an orientation adjusting stage where residual error appearing to be a plateau, and then following a fast convergence stage. It can be seen that the plateau for the Newton iterative algorithm is too long, which causes a slower convergence than the OIA/ODV and GOIA. So for this problem the GOIA is five times faster than the Newton iterative algorithm, and

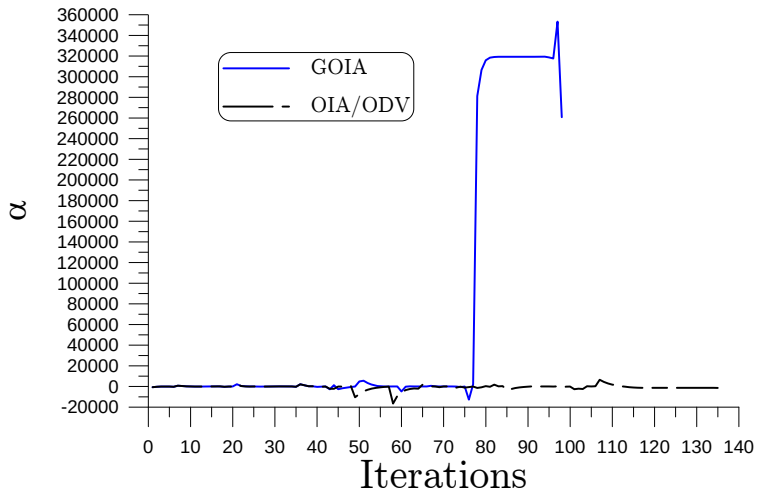

Figure 3: For example 1, solved by the GOIA and OIA/ODV comparing the weighting parameter.

the GOIA is faster than the OIA/ODV.

Both the OIA/ODV and GOIA have the same convergence rate 4 and the same value of $a_0 = 1$ for this problem, but their values of α are different. In Fig. 3 we compare the optimal α_o for the OIA/ODV and the critical α_c for the GOIA. It is amazingly, in the last few steps of the GOIA that the values of α are quite large, which show that in the last few steps the GOIA follows the Newton steps.

4.2 Example 2

We consider an almost linear Brown's problem [Brown (1973)]:

$$F_i = x_i + \sum_{j=1}^{n} x_j - (n+1), \quad i = 1,\ldots,n-1, \tag{80}$$

$$F_n = \prod_{j=1}^{n} x_j - 1, \tag{81}$$

with a closed-form solution $x_i = 1,\ i = 1,\ldots,n$.

As demonstarted by Han and Han (2010), Brown (1973) solved this problem with $n = 5$ by the Newton iterative algorithm, and gave an incorrectly converged solution $(-0.579, -0.579, -0.579, -0.579, 8.90)$. For $n = 10$ and 30, Brown (1973) found that the Newton iterative algorithm diverged quite rapidly. However, Liu and

Atluri (2011a) using the residual-norm based algorithm (RNBA) can also solve this problem without any difficulty. Furthermore, Liu, Dai and Atluri (2011b) using the optimal descent vector algorithm ODV(F) can solve this problem with $n = 100$,

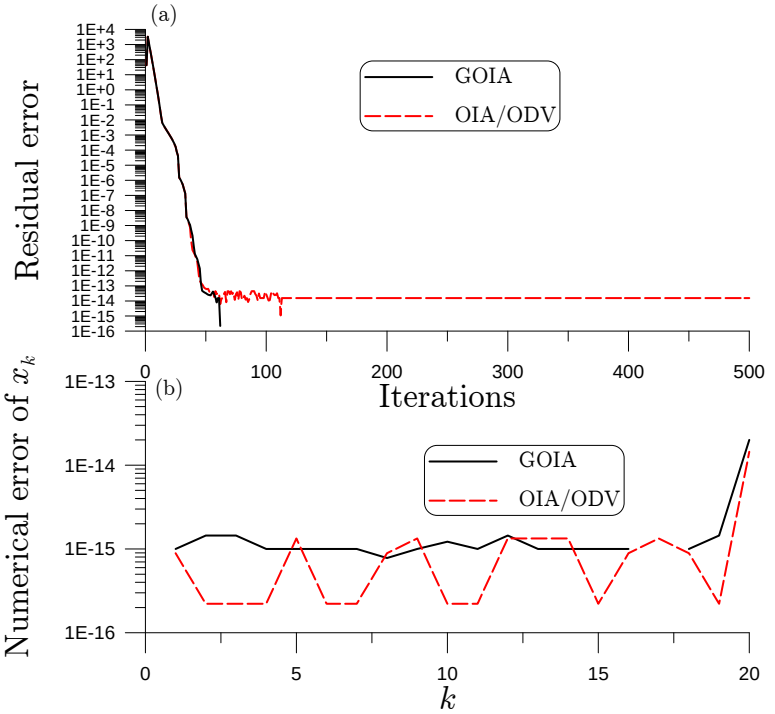

Figure 4: For example 2, solved by the GOIA and OIA/ODV, comparing (a) the residual errors, and (b) the numerical errors.

Here we fix $n = 20$ and $\gamma = 0.02$, and under a stringent convergence criterion $\varepsilon = 10^{-15}$ we test the performance of the OIA/ODV and GOIA. The residual errors and numerical errors are shown in Figs. 4(a) and 4(b), respectively. The accuracy is very good with the maximum error being 1.998×10^{-14} for the GOIA, and 1.44×10^{-14} for the OIA/ODV. When the OIA/ODV does not converge within 500 iterations, the GOIA as shown in Fig. 4(a) by the solid line of its residual error can converge with 62 iterations. As shown in Fig. 5, the GOIA has smaller a_0 than that of the OIA/ODV before the 62th iteration.

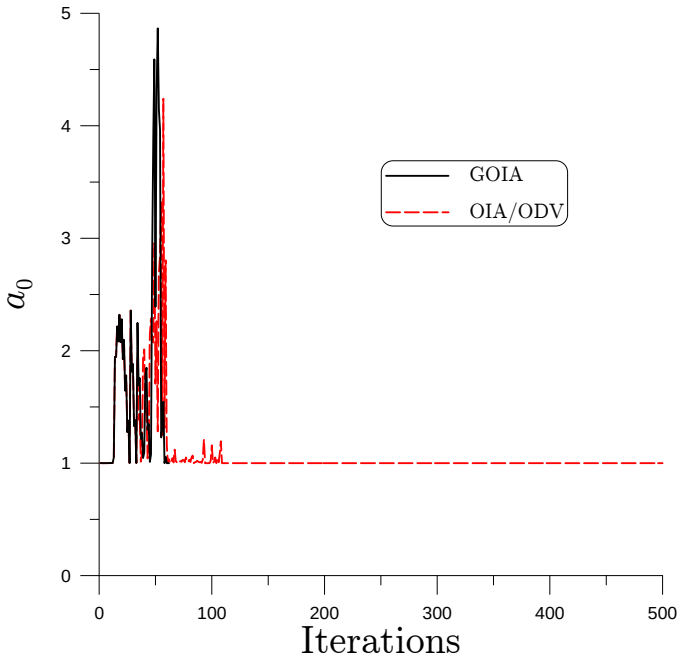

Figure 5: For example 2, solved by the GOIA and OIA/ODV comparing a_0.

4.3 Example 3

In this section we will apply the above algorithms to solve nonlinear ODE. To approach this problem by the polynomial interpolation, we begin with

$$p_m(x) = c_0 + \sum_{k=1}^{m} c_k x^k. \tag{82}$$

Now, the coefficient c_k is projected into two coefficients a_k and b_k to absorb more interpolation points; in the meanwhile, $\cos(k\theta_k)$ and $\sin(k\theta_k)$ are introduced to reduce the condition number of the coefficient matrix [Liu (2011)]. We suppose that

$$c_k = \frac{a_k \cos(k\theta_k)}{R_{2k}^k} + \frac{b_k \sin(k\theta_k)}{R_{2k+1}^k}, \tag{83}$$

and

$$\theta_k = \frac{2k\pi}{m}, \quad k = 1, \ldots, m. \tag{84}$$

The considered problem domain is $[a,b]$, and the interpolating points are:

$$a = x_0 < x_1 < x_2 < \ldots < x_{2m-1} < x_{2m} = b. \tag{85}$$

Substituting Eq. (83) into Eq. (82), we can obtain

$$p(x) = a_0 + \sum_{k=1}^{m} \left[a_k \left(\frac{x}{R_{2k}} \right)^k \cos(k\theta_k) + b_k \left(\frac{x}{R_{2k+1}} \right)^k \sin(k\theta_k) \right], \tag{86}$$

where we let $c_0 = a_0$. Here, a_k and b_k are unknown coefficients. In order to obtain them, we impose the following n interpolated conditions:

$$p_m(x_i) = y_i, \;\; i = 0, \ldots, n-1. \tag{87}$$

Thus, we obtain a linear equations system to determine a_k and b_k:

$$\begin{bmatrix} 1 & \frac{x_0\cos\theta_1}{R_2} & \frac{x_0\sin\theta_1}{R_3} & \ldots & \left(\frac{x_0}{R_{2m}}\right)^m \cos m\theta_m & \left(\frac{x_0}{R_{2m+1}}\right)^m \sin m\theta_m \\ 1 & \frac{x_1\cos\theta_1}{R_2} & \frac{x_1\sin\theta_1}{R_3} & \ldots & \left(\frac{x_1}{R_{2m}}\right)^m \cos m\theta_m & \left(\frac{x_1}{R_{2m+1}}\right)^m \sin m\theta_m \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{x_{2m-1}\cos\theta_1}{R_2} & \frac{x_{2m-1}\sin\theta_1}{R_3} & \ldots & \left(\frac{x_{2m-1}}{R_{2m}}\right)^m \cos m\theta_m & \left(\frac{x_{2m-1}}{R_{2m+1}}\right)^m \sin m\theta_m \\ 1 & \frac{x_{2m}\cos\theta_1}{R_2} & \frac{x_{2m}\sin\theta_1}{R_3} & \ldots & \left(\frac{x_{2m}}{R_{2m}}\right)^m \cos m\theta_m & \left(\frac{x_{2m}}{R_{2m+1}}\right)^m \sin m\theta_m \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ \vdots \\ a_m \\ b_m \end{bmatrix}$$

$$= \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_{2m-1} \\ y_{2m} \end{bmatrix}. \tag{88}$$

We note that the norm of the first column of the above coefficient matrix is $\sqrt{2m+1}$. According to the concept of equilibrated matrix, we can derive the optimal scales for the current interpolation with half order technique as

$$R_{2k} = \left(\frac{1}{2m+1} \sum_{j=0}^{2m} x_j^{2k} (\cos k\theta_k)^2 \right)^{1/(2k)}, \; k = 1, 2, \ldots, m, \tag{89}$$

$$R_{2k+1} = \left(\frac{1}{2m+1} \sum_{j=0}^{2m} x_j^{2k} (\sin k\theta_k)^2 \right)^{1/(2k)}, \; k = 1, 2, \ldots, m. \tag{90}$$

The improved method uses m order polynomial to interpolate $2m+1$ data nodes, while regular method [full order] can only interpolate $m+1$ data points.

Here we apply the above interpolation technique together with the nonlinear algebraic equations' solvers GOIA and OIA/ODV to solve the following nonlinear ODE:

$$u'' = \frac{3}{2}u^2, \tag{91}$$

$$u(0) = 4, \quad u(1) = 1. \tag{92}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{93}$$

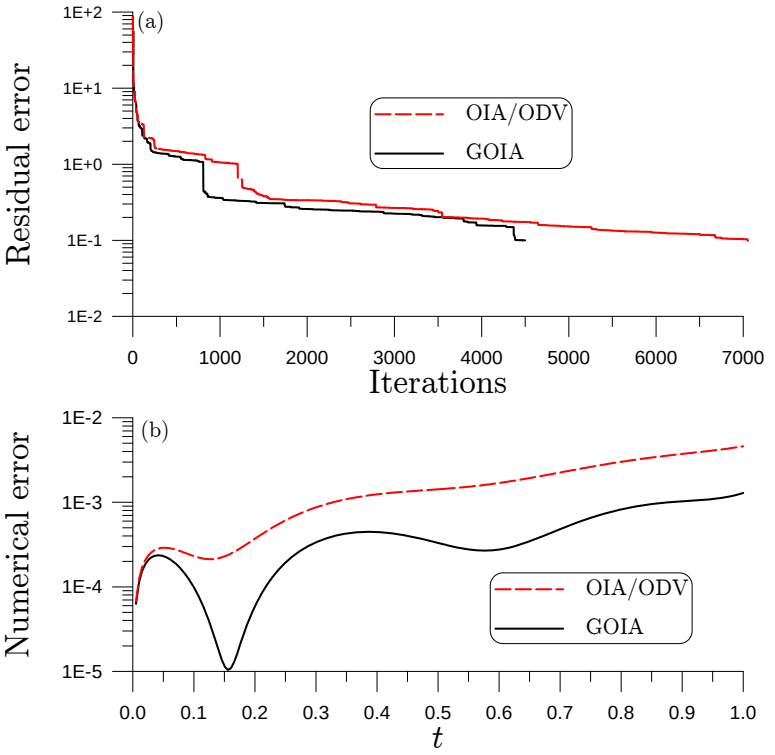

Figure 6: For example 3, solved by the GOIA and OIA/ODV, comparing (a) the residual errors, and (b) the numerical errors.

In the polynomial interpolation we take $m = 7$ and γ is fixed to be $\gamma = 0.1$. The residual errors are shown in Fig. 6(a), where the GOIA is convergent with 4499 iterations and the OIA/ODV is convergent with 7053 iterations, both under the convergence crtiterion with $\varepsilon = 0.1$. The numerical errors are compared in Fig. 6(b), of which we can see that both methods can lead to very accurate numerical solutions. The maximum error for the GOIA is about 1.29×10^{-3}, while that for the OIA/ODV is about 4.6×10^{-3}.

4.4 Example 4

We consider a nonlinear heat conduction equation:

$$u_t = \alpha(x)u_{xx} + \alpha'(x)u_x + u^2 + h(x,t), \tag{94}$$

$$\alpha(x) = (x-3)^2, \;\; h(x,t) = -7(x-3)^2 e^{-t} - (x-3)^4 e^{-2t}, \tag{95}$$

with a closed-form solution $u(x,t) = (x-3)^2 e^{-t}$.

We use the standard finite difference scheme to discretize Eq. (94). By applying the OIA/ODV and GOIA to solve the above equation in the domain of $0 \le x \le 1$ and $0 \le t \le 1$ we fix $\Delta x = 1/14$, $\Delta t = 1/20$, $\gamma = 0.1$ and $\varepsilon = 10^{-3}$. In Fig. 7(a) we show the residual errors, which are convergent very fast with 69 iterations for both methods with $\gamma = 0.1$ The numerical results are quite accurate with the maximum error being 3.3×10^{-3}. The weighting coefficients α of OIA/ODV and GOIA are compared in Fig. 7(b). For this problem the optimal vector and the best vector are coincident, which means that the local minimum of the OIA/ODV is also the global minimum of the GOIA.

4.5 Example 5

We consider Eqs. (94) and (95) again; however, we subject them to a final time condition with a closed-form solution being $u(x,t) = (x-3)^2 e^{-t}$. The boundary conditions and a final time condition are available from the above solution. It is known that the nonlinear backward heat conduction problem is highly ill-posed. In order to test the stability of GOIA we also add a relative noise in the final time data with an intensity $\sigma = 0.01$.

By applying the new algorithm to solve the above equation in the domain of $0 \le x \le 1$ and $0 \le t \le 1$ we fix $\Delta x = 1/n_1$ and $\Delta t = 1/n_2$, where $n_1 = 14$ and $n_2 = 10$ are numbers of nodal points used in a standard finite difference approximation of

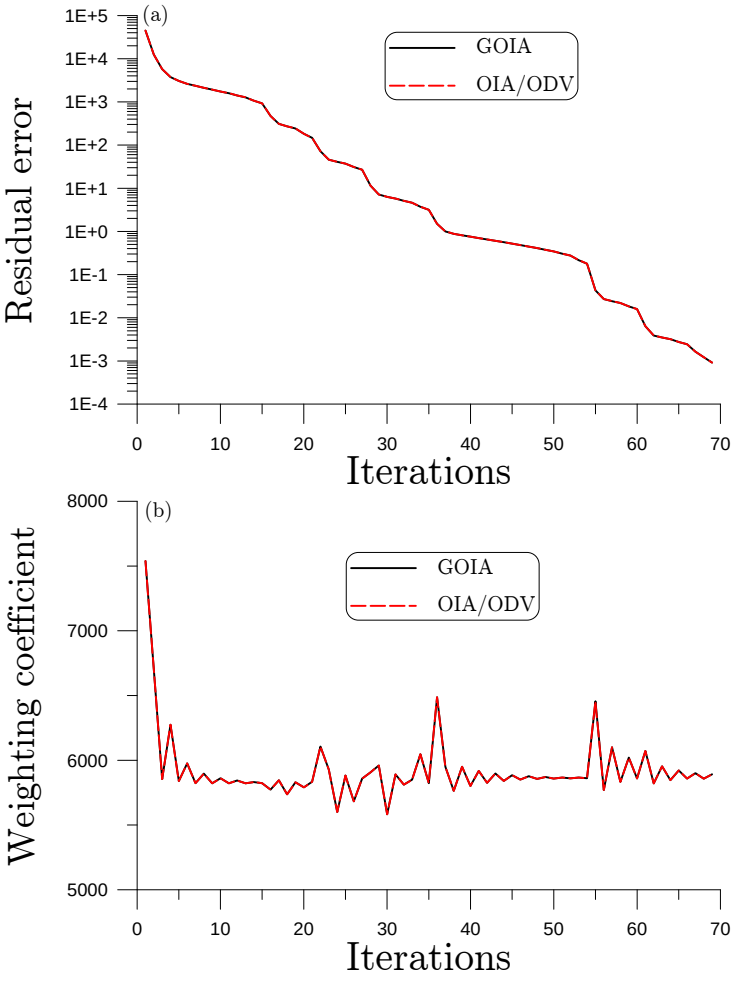

Figure 7: For example 4, solved by the GOIA and OIA/ODV, comparing (a) the residual errors, and (b) the weighting coefficients.

Eq. (94):

$$k(x_i)\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2}+k'(x_i)\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}+u_{i,j}^2+H(x_i,t_j)-\frac{u_{i,j+1}-u_{i,j}}{\Delta t}=0. \tag{96}$$

Under the convergence criterion $\varepsilon = 0.1$, the GOIA is convergent very fast with 49 iterations by using $\gamma = 0.2$ as shown in Fig. 8(a). As in Example 4, the OIA/ODV

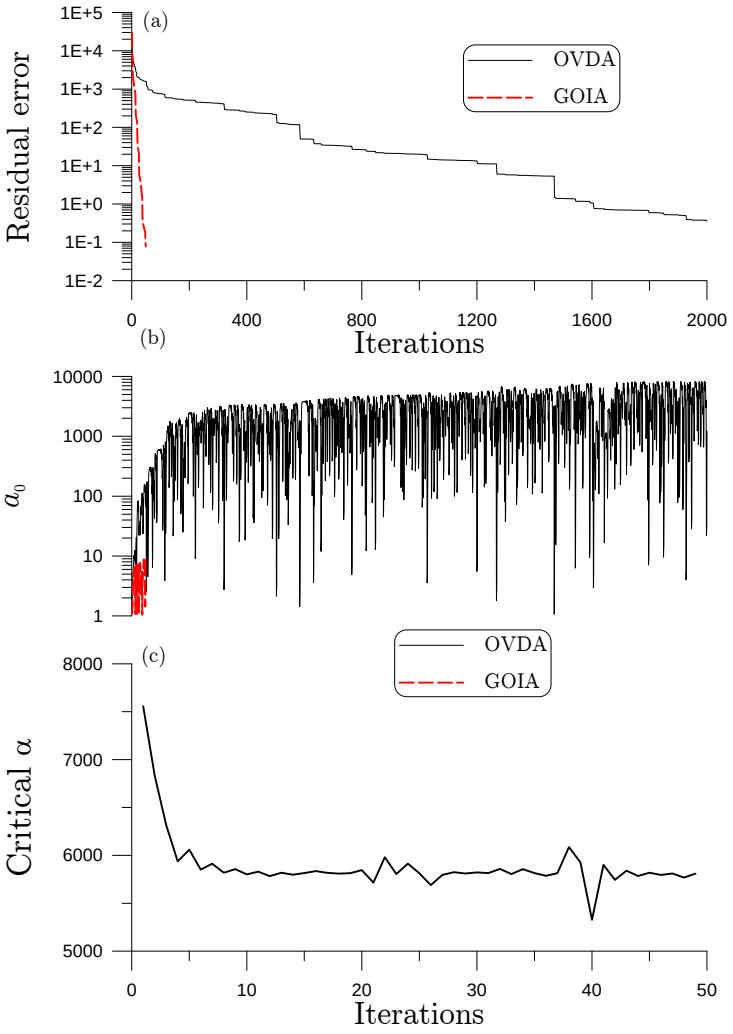

Figure 8: For example 5, solved by the OVDA and GOIA, comparing (a) the residual errors, and (b) a_0, and (c) the weighting coefficients of GOIA.

and the GOIA have the same performance for this problem. The residual error obtained by the GOIA is compared with that calculated by the OVDA [Liu and Kuo (2011)], which run 2000 steps without convergence. Both methods can attain accurate solutions with a maximum relative error being 1.785×10^{-3} for the OVDA, and 1.65×10^{-3} for the GOIA. Because a_0 defined in Eq. (46) is a very important factor of our new algorithm we compare it with the OVDA in Fig. 8(b), which we

can see that the value a_0 of GOIA is much smaller than that of the OVDA. The critical value of α for the GOIA is shown in Fig. 8(c). The present result is much better than that computed by Liu and Kuo (2011), and Liu and Atluri (2011b).

5 Conclusions

In the present paper, we have derived two *Optimal Iterative Algorithms* with *Optimal Descent Vectors* to accelerate the convergence speed in the numerical solution of NAEs. These two algorithms were named the Optimal Iterative Algorithm with Optimal Descent Vector (OIA/ODV), when the driving vector is the optimal descent vector; and the Globally Optimal Iterative Algorithm (GOIA) when the driving vector is the best descent vector. Both the OIA/ODV and the GOIA have good computational efficiency and accuracy. *However, for most cases the performance of the GOIA is better than the OIA/ODV*. Only two cases showed that the OIA/ODV and the GOIA have the same performance. *It was demonstrated that the critical value α_c of the weighting parameter in the best descent vector leads to the* **largest convergence rate** *with the descent vector given by* $\mathbf{u} = \alpha_c \mathbf{F} + \mathbf{B}^T\mathbf{F}$. **Due to its criticality, if one attempts to find a better value than α_c, there would be no real solution of α**. Hence, in the present frame of the future cone construction and under the class of the given descent vector $\mathbf{u} = \alpha\mathbf{F} + \mathbf{B}^{\mathrm{T}}\mathbf{F}$, the GOIA is a **globally optimal iterative algorithm** to solve the NAEs. In a near future we can extend the present theory to other iterative algorithms with different specifications of the descent vector.

Acknowledgement: The project NSC-100-2221-E-002-165-MY3 and the 2011 Outstanding Research Award from National Science Council of Taiwan, and Taiwan Research Front Awards 2011 from Thomson Reuters to the first author are highly appreciated. The second author acknowledges the support of the Army Research Labs under a collaborative research agreement with UCI, as well as by the WCU program through the National Research Foundation of Korea funded by the Ministry of Education, Science, and Technology [Grant No: R33-10049].

References

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

Brown, K. M. (1973): Computer oriented algorithms for solving systems of simultaneous nonlinear algebraic equations. In Numerical Solution of Systems of

Nonlinear Algebraic Equations, Byrne, G. D. and Hall C. A. Eds., pp. 281-348, Academic Press, New York.

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

Han, T.; Han Y. (2010): Solving large scale nonlinear equations by a new ODE numerical integration method. *Appl. Math.*, vol. 1, pp. 222-229.

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

Ku, C. Y.; Yeih, W.; Liu, C.-S. (2010): Solving non-linear algebraic equations by a scalar Newton-homotopy continuation method. *Int. J. Non-Linear Sci. Num. Simul.*, vol. 11, pp. 435-450.

Liu, C.-S. (2000): A Jordan algebra and dynamic system with associator as vector field. *Int. J. Non-Linear Mech.*, vol. 35, pp. 421-429.

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. (2008): 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.

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

Liu, C.-S. (2009b): A fictitious time integration method for the Burgers equation. *CMC: Computers, Materials & Continua*, vol. 9, pp. 229-252.

Liu, C.-S. (2009c): A fictitious time integration method for solving delay ordinary differential equations. *CMC: Computers, Materials & Continua*, vol. 10, pp. 97-116.

Liu, C.-S. (2011): A highly accurate multi-scale full/half-order polynomial interpolation. *CMC: Computers, Materials & Continua*, vol. 25, pp. 239-263.

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.; Atluri, S. N. (2011a): Simple "residual-norm" based algorithms, for the solution of a large system of non-linear algebraic equations, which converge faster than the Newton's method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 71, pp. 279-304.

Liu, C.-S.; Atluri, S. N. (2011b): An iterative algorithm for solving a system of

nonlinear algebraic equations, $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, using the system of ODEs with an optimum α in $\dot{\mathbf{x}} = \lambda[\alpha\mathbf{F} + (1-\alpha)\mathbf{B}^{\mathrm{T}}\mathbf{F}]$; $B_{ij} = \partial F_i/\partial x_j$. *CMES: Computer Modeling in Engineering & Sciences*, vol. 73, pp. 395-431.

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.

Liu, C.-S.; Dai, H. H.; Atluri, S. N. (2011): Iterative solution of a system of nonlinear algebraic equations $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, using $\dot{\mathbf{x}} = \lambda[\alpha\mathbf{R} + \beta\mathbf{P}]$ or $\lambda[\alpha\mathbf{F} + \beta\mathbf{P}^*]$, $\mathbf{R}$ is a normal to a hyper-surface function of $\mathbf{F}$, $\mathbf{P}$ normal to $\mathbf{R}$, and $\mathbf{P}^*$ normal to $\mathbf{F}$. *CMES: Computer Modeling in Engineering & Sciences*, vol. 81, pp. 335-362.

Liu, C.-S.; Yeih, W.; Kuo, C. L.; Atluri, S. N. (2009): A scalar homotopy method for solving an over/under-determined system of non-linear algebraic equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 53, pp. 47-71.