 CMES, vol.81, no.2, pp.195-227, 2011

A Further Study on Using $\dot{\mathbf{x}}=\lambda[\alpha\mathbf{R}+\beta\mathbf{P}]$ ($\mathbf{P}=\mathbf{F}-\mathbf{R}(\mathbf{F}\cdot\mathbf{R})/\|\mathbf{R}\|^2$) and $\dot{\mathbf{x}}=\lambda[\alpha\mathbf{F}+\beta\mathbf{P}^*]$ ($\mathbf{P}^*=\mathbf{R}-\mathbf{F}(\mathbf{F}\cdot\mathbf{R})/\|\mathbf{F}\|^2$) in Iteratively Solving the Nonlinear System of Algebraic Equations $\mathbf{F}(\mathbf{x})=\mathbf{0}$

Chein-Shan Liu[1,2], **Hong-Hua Dai**[1] **and Satya N. Atluri**[1]

Abstract: In this continuation of a series of our earlier papers, we define a hypersurface $h(\mathbf{x},t)=0$ in terms of the unknown vector $\mathbf{x}$, and a monotonically increasing function $Q(t)$ of a time-like variable t, to solve a system of nonlinear algebraic equations $\mathbf{F}(\mathbf{x})=\mathbf{0}$. If $\mathbf{R}$ is a vector related to $\partial h/\partial\mathbf{x}$, we consider the evolution equation $\dot{\mathbf{x}}=\lambda[\alpha\mathbf{R}+\beta\mathbf{P}]$, where $\mathbf{P}=\mathbf{F}-\mathbf{R}(\mathbf{F}\cdot\mathbf{R})/\|\mathbf{R}\|^2$ such that $\mathbf{P}\cdot\mathbf{R}=0$; or $\dot{\mathbf{x}}=\lambda[\alpha\mathbf{F}+\beta\mathbf{P}^*]$, where $\mathbf{P}^*=\mathbf{R}-\mathbf{F}(\mathbf{F}\cdot\mathbf{R})/\|\mathbf{F}\|^2$ such that $\mathbf{P}^*\cdot\mathbf{F}=0$. From these evolution equations, we derive Optimal Iterative Algorithms (OIAs) with Optimal Descent Vectors (ODVs), abbreviated as ODV(R) and ODV(F), by deriving optimal values of α and β for fastest convergence. Several numerical examples illustrate that the present algorithms converge very fast. We also provide a solution of the nonlinear Duffing oscillator, by using a harmonic balance method and a post-conditioner, when very high-order harmonics are considered.

Keywords: Nonlinear algebraic equations, Optimal iterative algorithm (OIA), Optimal descent vector (ODV), Optimal vector driven algorithm (OVDA), Fictitious time integration method (FTIM), Residual-norm based algorithm (RNBA), Duffing equation, Post-conditioned harmonic balance method (PCHB)

1 Introduction

For solving a system of nonlinear algebraic equations (NAEs):

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

[1] Center for Aerospace Research & Education, University of California, Irvine
[2] Department of Civil Engineering, National Taiwan University, Taipei, Taiwan. E-mail: liucs@ntu.edu.tw

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{F} \in \mathbb{R}^n$, Liu and Atluri (2008a) first derived a system of nonlinear ODEs for $\mathbf{x}$ in terms of a time-like variable t:

$$\dot{\mathbf{x}} = -\frac{\nu}{q(t)}\mathbf{F}(\mathbf{x}), \tag{2}$$

where ν is a nonzero constant and $q(t)$ may in general be a monotonically increasing function of t. In their approach of the Fictitious Time Integration Method (FTIM), the term $\nu/q(t)$ plays a major role of a stabilized controller to help one obtain a solution even for a bad initial guess of solution, and speed up the convergence. Liu and Chang (2009) combined the FTIM with a nonstandard group preserving scheme [Liu (2001, 2005)] for solving a system of ill-posed linear equations. Ku, Yeih, Liu and Chi (2009) have employed a time-like function of $q(t) = (1+t)^m$, $0 < m \leq 1$ in Eq. (2), and a better performance was observed. After the work by Liu and Atluri (2008a), the FTIM had been applied to solve many engineering problems [Liu and Atluri (2008b, 2008c, 2009); Liu (2008a, 2008b, 2009a, 2009b, 2009c, 2009d, 2010); Chi, Yeih and Liu (2009); Ku, Yeih, Liu and Chi (2009); Chang and Liu (2009); Tsai, Liu and Yeih (2010)]. In spite of its success, the FTIM has only a local convergence, and one needs judiciously to determine the viscous damping coefficient ν.

Then, to remedy the shortcoming of the vector homotopy method as initiated by Davidenko (1953), Liu, Yeih, Kuo and Atluri (2009) defined a scalar homotopy function

$$h(\mathbf{x},t) = \frac{1}{2}[t\|\mathbf{F}(\mathbf{x})\|^2 - (1-t)\|\mathbf{x}\|^2], \tag{3}$$

and considered a "normality" relation for the evolution equation for $\dot{\mathbf{x}}$ in terms of t:

$$\dot{\mathbf{x}} = -\frac{\frac{\partial h}{\partial t}}{\|\frac{\partial h}{\partial \mathbf{x}}\|^2}\frac{\partial h}{\partial \mathbf{x}}, \tag{4}$$

where

$$\frac{\partial h}{\partial t} = \frac{1}{2}[\|\mathbf{F}(\mathbf{x})\|^2 + \|\mathbf{x}\|^2], \tag{5}$$

$$\frac{\partial h}{\partial \mathbf{x}} = t\mathbf{B}^{\mathrm{T}}\mathbf{F} - (1-t)\mathbf{x} = t\mathbf{R} - (1-t)\mathbf{x}. \tag{6}$$

Here $\partial h/\partial \mathbf{x}$ is a normal to the hyper-surface $h(\mathbf{x},t) = 0$, $\mathbf{B} := \partial \mathbf{F}/\partial \mathbf{x}$ is the Jacobian matrix, and we denote

$$\mathbf{R} := \mathbf{B}^{\mathrm{T}}\mathbf{F}, \tag{7}$$

where the superscript T denotes the transpose. Thus, $\mathbf{R}$ is a normal to the hyper-surface $h(\mathbf{x},t)=0$ at $t=1$. *The scalar homotopy method of Liu, Yeih, Kuo and Atluri (2009) thus made a crucial step in solving the NAEs by using a manifold-based method.*

Ku, Yeih and Liu (2010) modified Eq. (4) slightly, and defined

$$\dot{\mathbf{x}} = -\frac{\nu}{(1+t)^m}\frac{\|\mathbf{F}\|^2}{\|\mathbf{R}\|^2}\mathbf{R}. \tag{8}$$

In 2011, Liu and Atluri (2011a) introduced a modified definition for the hyper-surface:

$$h(\mathbf{x},t) = \frac{Q(t)}{2}\|\mathbf{F}(\mathbf{x})\|^2 - C = 0, \tag{9}$$

where $Q(t)>0$ is a monotonically increasing function of t, and C is a constant. In Liu and Atluri (2011a), the evolution equation for $\dot{\mathbf{x}}$ was taken to be

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)}\frac{\|\mathbf{F}\|^2}{\|\mathbf{R}\|^2}\mathbf{R}, \tag{10}$$

which is a generalization of Eq. (8). Integrating Eq. (10) as a system of nonlinear ODEs for $\mathbf{x}$ in terms of t, leads to an algorithm for finding the solution $\mathbf{x}$ of $\mathbf{F}(\mathbf{x})=\mathbf{0}$. Liu and Atluri (2011a) have pointed out the limitations of the above Residual-Norm Based Algorithm (RNBA), which converges very fast at the first many steps and then slows down to a plateau without further reducing the residual error of $\|\mathbf{F}\|$.

To further improve the convergence of the solution for $\mathbf{x}$, Liu and Atluri (2011b) used the same hyper-surface as in Eq. (9), but modified the evolution equation for $\dot{\mathbf{x}}$ as a "*non-normal*" relation, involving both $\mathbf{F}$ and $\mathbf{R}$:

$$\dot{\mathbf{x}} = \lambda\mathbf{u} = \lambda[\alpha\mathbf{F} + (1-\alpha)\mathbf{R}], \tag{11}$$

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

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)}\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}[\alpha\mathbf{F} + (1-\alpha)\mathbf{R}], \tag{12}$$

where

$$\begin{aligned}
&\mathbf{A} := \mathbf{B}\mathbf{B}^{\mathrm{T}},\\
&\mathbf{v} = \mathbf{B}\mathbf{u} = \mathbf{v}_1 + \alpha\mathbf{v}_2 = \mathbf{A}\mathbf{F} + \alpha(\mathbf{B}-\mathbf{A})\mathbf{F},\\
&\alpha = \frac{(\mathbf{v}_1\cdot\mathbf{F})(\mathbf{v}_1\cdot\mathbf{v}_2) - (\mathbf{v}_2\cdot\mathbf{F})\|\mathbf{v}_1\|^2}{(\mathbf{v}_2\cdot\mathbf{F})(\mathbf{v}_1\cdot\mathbf{v}_2) - (\mathbf{v}_1\cdot\mathbf{F})\|\mathbf{v}_2\|^2}.
\end{aligned} \tag{13}$$

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

$$\dot{\mathbf{x}} = \lambda [\alpha \mathbf{R} + \beta \mathbf{p}], \tag{14}$$

where

$$\mathbf{p} = \left[\mathbf{I}_n - \frac{\|\mathbf{R}\|^2}{\mathbf{R}^{\mathrm{T}} \mathbf{C} \mathbf{R}} \mathbf{C} \right] \mathbf{R}, \tag{15}$$

in which

$$\mathbf{C} = \mathbf{B}^{\mathrm{T}} \mathbf{B}, \tag{16}$$

such that, clearly, $\mathbf{p}$ is orthogonal to $\mathbf{R}$, i.e.,

$$\mathbf{R} \cdot \mathbf{p} = 0, \tag{17}$$

where a dot between two vectors signifies the inner product. Thus Liu, Dai and Atluri (2011) derived:

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}} (\alpha \mathbf{B}\mathbf{R} + \beta \mathbf{B}\mathbf{p})} [\alpha \mathbf{R} + \beta \mathbf{p}]. \tag{18}$$

Liu, Dai and Atluri (2011) derived an OIA with the optimized α and $\beta (= 1 - \alpha)$ to achieve a faster convergence for the iterative solution for $\mathbf{x}$.

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

$$\dot{\mathbf{x}} = -\frac{\dot{Q}(t)}{2Q(t)} \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}} (\alpha \mathbf{B}\mathbf{F} + \beta \mathbf{B}\mathbf{p}^*)} [\alpha \mathbf{F} + \beta \mathbf{p}^*], \tag{19}$$

where $\mathbf{p}^*$ is orthogonal to $\mathbf{F}$, i.e. $\mathbf{F} \cdot \mathbf{p}^* = 0$; and

$$\mathbf{p}^* = \left[\mathbf{I}_n - \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}} \mathbf{C} \mathbf{F}} \mathbf{C} \right] \mathbf{F}. \tag{20}$$

It was shown in Liu, Dai and Atluri (2011) that the OIAs based on Eqs. (18) and (19), namely the OIA/ODV[R] and OIA/ODV[F], had the fastest convergence and best accuracy as compared to any algorithms published in the previous literature by many other authors, as well the present authors themselves.

It can be seen that neither the vector $\mathbf{p}$ defined in Eq. (15) and which is normal to $\mathbf{R}$, nor the vector $\mathbf{p}^*$ defined in Eq. (20) and which is normal to $\mathbf{F}$, *are unique*. In

this paper we consider alternate vectors $\mathbf{P}$ and $\mathbf{P}^*$ which are also normal to $\mathbf{R}$ and $\mathbf{F}$, respectively, as follows:

$$\mathbf{P} := \mathbf{F} - \frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{R}\|^2}\mathbf{R}, \tag{21}$$

such that, clearly,

$$\mathbf{R}\cdot\mathbf{P} = 0, \tag{22}$$

and

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

such that, clearly,

$$\mathbf{F}\cdot\mathbf{P}^* = 0. \tag{24}$$

Using the relations as in Eqs. (21)-(24), we explore in this paper, the following evolution equations for $\dot{\mathbf{x}}$:

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

and

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

As before, we seek to optimize α and β, and seek purely iterative algorithms to solve for $\mathbf{x}$. We show that, with the algorithms proposed in this paper, we have been 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 $\mathbf{B} = \partial\mathbf{F}/\partial\mathbf{x}$.

The remaining portions of this paper are arranged as follows. In Section 2, we give detailed explanations of the related equations, where the concept of a *two-dimensional combination* of the residual vector $\mathbf{F}$ and the descent vector $\mathbf{R}$ with an orthogonality is introduced. Then, a genuine dynamics on the invariant-manifold is constructed in Section 3, resulting in an optimal iterative algorithm in terms of two weighting factors being optimized explicitly. The numerical examples are given in Section 4 to display some advantages of the present Optimal Iterative Algorithms (OIAs), as compared to the FTIM, the residual-norm based algorithm [Liu and Atluri (2011a)], OVDA, OIA/ODV[R], OIA/ODV[F] and the Newton method for some numerical examples. Finally, some conclusions are drawn in Section 5.

2 The residual-descent perpendicular vectors P and P*

Neither the vector **p** in Eq. (15) which is perpendicular to **R**, nor the vector $\mathbf{p}^*$ in Eq. (20) which is perpendicular to **F**, are unique. In our continuing exploration of the various choices for the descent vector in the evolution for $\dot{\mathbf{x}}$, we consider another vector **P**, which is also orthogonal to **R**:

$$\mathbf{P} := \mathbf{F} - \frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{R}\|^2}\mathbf{R}. \tag{27}$$

Clearly, $\mathbf{R}\cdot\mathbf{P} = 0$. Clearly, there are infinitely many such vectors **P** which are orthogonal to **R**. We now explore the following evolution equation:

$$\dot{\mathbf{x}} = \lambda\mathbf{u}, \tag{28}$$

where

$$\mathbf{u} = \alpha\mathbf{R} + \beta\mathbf{P} = \left[\alpha - \beta\frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{R}\|^2}\right]\mathbf{R} + \beta\mathbf{F}. \tag{29}$$

The above two parameters α and β are to be optimized in Section 3.1. Previously, Liu and Atluri (2011b) have constructed a powerful optimal vector driven algorithm (OVDA) to solve NAEs with the driving vector to be a linear combination of **F** and **R**: $\alpha\mathbf{F} + (1-\alpha)\mathbf{B}^{\mathrm{T}}\mathbf{F}$, which is a *one-dimensional combination* of **F** and **R**. Here we further extend this theory to a *two-dimensional combination* of **F** and **R** as shown by Eq. (29). The evolution equation (28) is not a simple variant of that explored in Liu, Dai and Atluri (2011) as shown in Eq. (14). The main differences between these two evolution equations lie on the difference of the subspace and the parameters α and β. When the former is evolving in a two-dimensional subspace spanned by **R** and **F**, the latter is evolving in a two-dimensional subspace generated from only **R** and its perpendicular vector $[\mathbf{I}_n - (\|\mathbf{R}\|^2/\mathbf{R}^{\mathrm{T}}\mathbf{C}\mathbf{R})\mathbf{C}]\mathbf{R}$.

We also consider, alternatively, an evolution equation

$$\dot{\mathbf{x}} = \lambda[\alpha\mathbf{F} + \beta\mathbf{P}^*], \tag{30}$$

where

$$\mathbf{P}^* = \mathbf{R} - \frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{F}\|^2}\mathbf{F}. \tag{31}$$

Clearly, $\mathbf{F}\cdot\mathbf{P}^* = 0$.

Taking the time differential of Eq. (9) with respect to t and considering $\mathbf{x} = \mathbf{x}(t)$, we can obtain

$$\frac{1}{2}\dot{Q}(t)\|\mathbf{F}(\mathbf{x})\|^2 + Q(t)\mathbf{R}\cdot\dot{\mathbf{x}} = 0. \tag{32}$$

Inserting Eq. (28) into Eq. (32) and using Eq. (7) we can solve λ, and inserting λ into Eq. (28) we can derive

$$\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{u}, \tag{33}$$

where

$$q(t) := \frac{\dot{Q}(t)}{2Q(t)}, \tag{34}$$

$$\mathbf{v} := \alpha\mathbf{v}_1 + \beta\mathbf{v}_2 = \mathbf{Bu} = \alpha\mathbf{BR} + \beta\mathbf{BP}. \tag{35}$$

Hence, in our algorithm if $Q(t)$ can be guaranteed to be a monotonically increasing function of t, from Eq. (9) we have an absolutely convergent property in solving the nonlinear equations system (1):

$$\|\mathbf{F}(\mathbf{x})\|^2 = \frac{C}{Q(t)}, \tag{36}$$

where

$$C = \|\mathbf{F}(\mathbf{x}_0)\|^2 \tag{37}$$

is determined by the initial value $\mathbf{x}_0$. We do not need to specify the function $Q(t)$ a priori, but $\sqrt{C/Q(t)}$ merely acts as a measure of the residual error of $\mathbf{F}$ in time. When t is increased to a large value, 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.

3 An optimal iterative algorithm

Let

$$s = \frac{Q(t)}{Q(t+\Delta t)} = \frac{\|\mathbf{F}(t+\Delta t)\|^2}{\|\mathbf{F}(t)\|^2} \tag{38}$$

be an important quantity in assessing the convergence of our algorithm for solving the system (1) of NAEs.

Following the same procedures as those explored by Liu, Dai and Atluri (2011) we can derive

$$s = 1 - \frac{1-\gamma^2}{a_0}, \tag{39}$$

where $0 \leq \gamma < 1$ is a relaxed parameter, and

$$a_0 := \frac{\|\mathbf{F}\|^2 \|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2} \geq 1, \tag{40}$$

by using the Cauchy-Schwarz inequality:

$$\mathbf{F}^{\mathrm{T}}\mathbf{v} \leq \|\mathbf{F}\| \|\mathbf{v}\|.$$

3.1 Optimizations of α and β

Then by inserting Eq. (40) for a_0 into Eq. (39) we can write s to be

$$s = 1 - \frac{(1-\gamma^2)(\mathbf{F}\cdot\mathbf{v})^2}{\|\mathbf{F}\|^2 \|\mathbf{v}\|^2}, \tag{41}$$

where $\mathbf{v}$ as defined by Eq. (35) includes the parameters α and β. By the minimization of

$$\min_{\alpha,\beta} s, \tag{42}$$

we let $\partial s/\partial\alpha = 0$ and $\partial s/\partial\beta = 0$, and again, following Liu, Dai and Atluri (2011) we can derive

$$\beta = \omega\alpha, \tag{43}$$

where

$$\omega = \frac{[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2]\cdot\mathbf{v}_1}{[\mathbf{v}_2, \mathbf{F}, \mathbf{v}_1]\cdot\mathbf{v}_2} \tag{44}$$

is expressed in terms of the Jordan algebra derived by Liu (2000a):

$$[\mathbf{a}, \mathbf{b}, \mathbf{c}] = (\mathbf{a}\cdot\mathbf{b})\mathbf{c} - (\mathbf{c}\cdot\mathbf{b})\mathbf{a}, \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n. \tag{45}$$

Usually, for $\mathbf{u}$ as shown by Eq. (29) we can require the coefficient in front of $\mathbf{R}$ to be equal to 1, if $\mathbf{R}$ plays the major role in the search vector for finding the solution for $\mathbf{x}$, such that, from

$$\alpha - \beta\frac{\mathbf{F}\cdot\mathbf{R}}{\|\mathbf{R}\|^2} = 1, \tag{46}$$

and Eq. (43) we can solve

$$\alpha = \frac{\|\mathbf{R}\|^2}{\|\mathbf{R}\|^2 - \omega\mathbf{F}\cdot\mathbf{R}}, \quad \beta = \frac{\omega\|\mathbf{R}\|^2}{\|\mathbf{R}\|^2 - \omega\mathbf{F}\cdot\mathbf{R}}. \tag{47}$$

3.2 An optimal iterative algorithm, with the present optimal descent vector

Thus, we can arrive at a purely iterative algorithm by discretizing Eq. (33) with the forward Euler method and using $q(t)\Delta t = (1-\gamma)/a_0$:
(i) Select a suitable value of γ in $0 \le \gamma < 1$, and assume an initial value of $\mathbf{x}_0$ and compute $\mathbf{F}_0 = \mathbf{F}(\mathbf{x}_0)$.
(ii) For $k = 0, 1, 2 \ldots$, we repeat the following computations:

$$
\begin{aligned}
\mathbf{R}_k &= \mathbf{B}_k^{\mathrm{T}} \mathbf{F}_k, \\
\mathbf{P}_k &= \mathbf{F}_k - \frac{\mathbf{R}_k \cdot \mathbf{F}_k}{\|\mathbf{R}_k\|^2} \mathbf{R}_k, \\
\mathbf{v}_1^k &= \mathbf{B}_k \mathbf{R}_k, \\
\mathbf{v}_2^k &= \mathbf{B}_k \mathbf{P}_k, \\
\omega_k &= \frac{[\mathbf{v}_1^k, \mathbf{F}_k, \mathbf{v}_2^k] \cdot \mathbf{v}_1^k}{[\mathbf{v}_2^k, \mathbf{F}_k, \mathbf{v}_1^k] \cdot \mathbf{v}_2^k}, \\
\alpha_k &= \frac{\|\mathbf{R}_k\|^2}{\|\mathbf{R}_k\|^2 - \omega_k \mathbf{F}_k \cdot \mathbf{R}_k}, \\
\beta_k &= \frac{\omega_k \|\mathbf{R}_k\|^2}{\|\mathbf{R}_k\|^2 - \omega_k \mathbf{F}_k \cdot \mathbf{R}_k}, \\
\mathbf{u}_k &= \alpha_k \mathbf{R}_k + \beta_k \mathbf{P}_k, \\
\mathbf{v}_k &= \alpha_k \mathbf{v}_1^k + \beta_k \mathbf{v}_2^k, \\
\mathbf{x}_{k+1} &= \mathbf{x}_k - (1-\gamma) \frac{\mathbf{F}_k \cdot \mathbf{v}_k}{\|\mathbf{v}_k\|^2} \mathbf{u}_k.
\end{aligned} \tag{48}
$$

If $\mathbf{x}_{k+1}$ converges according to a given stopping criterion $\|\mathbf{F}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii).

Sometimes we have another option to choose the driving vector in solving nonlinear algebraic equations. Here, instead of $\mathbf{R}$ we use $\mathbf{F}$ as a primary driving vector, and then use the $\mathbf{P}^*$ as defined by Eq. (31), which is orthogonal to $\mathbf{F}$. Hence, we also have another option of the optimal iterative algorithm:
(i) Select a suitable value of γ in $0 \le \gamma < 1$, and assume an initial value of $\mathbf{x}_0$ and compute $\mathbf{F}_0 = \mathbf{F}(\mathbf{x}_0)$.

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

$$\begin{aligned}
&\mathbf{R}_k = \mathbf{B}_k^{\mathrm{T}} \mathbf{F}_k, \\
&\mathbf{P}_k^* = \mathbf{R}_k - \frac{\mathbf{R}_k \cdot \mathbf{F}_k}{\|\mathbf{F}_k\|^2} \mathbf{F}_k, \\
&\mathbf{v}_1^k = \mathbf{B}_k \mathbf{F}_k, \\
&\mathbf{v}_2^k = \mathbf{B}_k \mathbf{P}_k^*, \\
&\omega_k = \frac{[\mathbf{v}_1^k, \mathbf{F}_k, \mathbf{v}_2^k] \cdot \mathbf{v}_1^k}{[\mathbf{v}_2^k, \mathbf{F}_k, \mathbf{v}_1^k] \cdot \mathbf{v}_2^k}, \\
&\alpha_k = \frac{\|\mathbf{F}_k\|^2}{\|\mathbf{F}_k\|^2 - \omega_k \mathbf{F}_k \cdot \mathbf{R}_k}, \\
&\beta_k = \frac{\omega_k \|\mathbf{F}_k\|^2}{\|\mathbf{F}_k\|^2 - \omega_k \mathbf{F}_k \cdot \mathbf{R}_k}, \\
&\mathbf{u}_k = \alpha_k \mathbf{F}_k + \beta_k \mathbf{P}_k^*, \\
&\mathbf{v}_k = \alpha_k \mathbf{v}_1^k + \beta_k \mathbf{v}_2^k, \\
&\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma) \frac{\mathbf{F}_k \cdot \mathbf{v}_k}{\|\mathbf{v}_k\|^2} \mathbf{u}_k.
\end{aligned} \tag{49}$$

If $\mathbf{x}_{k+1}$ converges according to a given stopping criterion $\|\mathbf{F}_{k+1}\| < \varepsilon$, then stop; otherwise, go to step (ii).

We call the algorithm in Eq. (48) the ODV(R), where $\mathbf{R}$ plays the role of a primary driving vector, and the algorithm in Eq. (49) is called the ODV(F), where $\mathbf{F}$ plays the role of a primary driving vector. Up to here, we have derived two novel algorithms endowed with a Jordan structure in Eq. (44) for computing the coefficient ω. While the relaxation parameter γ is chosen by the user, depending on the problem, the parameters α and β are precisely given in Eq. (48) for the algorithm ODV(R), and in Eq. (49) for the algorithm ODV(F).

4 Numerical examples

In this section we apply the new methods of ODV(F) and ODV(R) to solve some nonlinear ODEs and PDEs. In order to reveal the superior performance of the present algorithms, we compare some numerical results with those calculated by Liu, Dai and Atluri (2011), who proposed two algorithms with the designations of OIA/ODV[F] and OIA/ODV[R], and also in some cases, we will compare the present algorithms with the Newton method, the FTIM proposed by Liu and Atluri (2008a), the residual-norm based algorithm (RNBA) proposed by Liu and Atluri (2011a), and the optimal vector driven algorithm (OVDA) proposed by Liu and

Atluri (2011b).

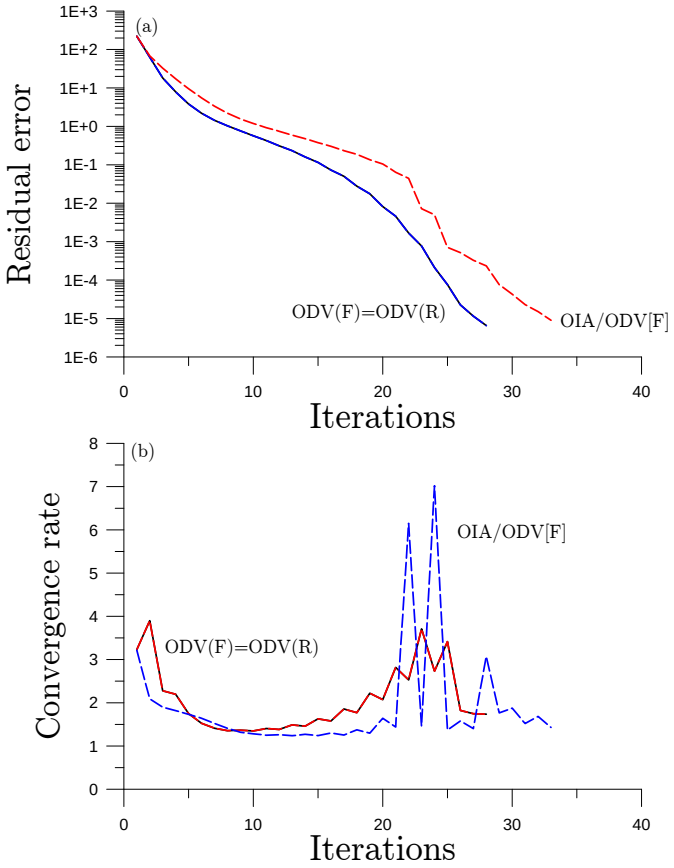

Figure 1: For example 1, comparing (a) the residual errors, and (b) the convergence rates of ODV(F) and ODV(R) with the OIA/ODV[F] in Liu, Dai and Atluri (2011).

4.1 Example 1

In this example we apply the present Eqs. (48) and (49), algorithms ODV(R) and ODV(F), to solve the following nonlinear boundary value problem:

$$u'' = \frac{3}{2}u^2, \tag{50}$$

$$u(0) = 4, \quad u(1) = 1. \tag{51}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{52}$$

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

$$F_i = \frac{1}{(\Delta x)^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2 = 0, \tag{53}$$

$$u_0 = 4, \;\; u_{n+1} = 1, \tag{54}$$

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

We fix $n = 9$ and $\varepsilon = 10^{-5}$. The parameter γ used both in ODV(F) and ODV(R) is 0.05. In Fig. 1(a) we compare the residual errors obtained by ODV(F) and ODV(R), while the convergence rates of ODV(F) and ODV(R) are shown in Fig. 1(b). They converge with 28 iterations, and the mximum numerical errors are both 4.7×10^{-3}. Very interestingly, these two algorithms lead to the same results for this example. The residual error calculated by Liu, Dai and Atluri (2011) with OIA/ODV[F] is also shown in Fig. 1 by the dashed line. When both ODV(F) and ODV(R) converge with 28 iterations, the OIA/ODV[F] converges with 33 iterations. As shown in Fig. 1(b), the most convergence rates of OIA/ODV[F] are smaller than that of ODV(F) and ODV(R). The convergence rate is evaluated by $1/\sqrt{s_{k+1}} = \|\mathbf{F}_k\|/\|\mathbf{F}_{k+1}\|$ at each iteration $k = 0, 1, 2, \ldots$.

4.2 Example 2

One famous mesh-less numerical method to solve the nonlinear PDE of elliptic type is the radial basis function (RBF) method, which expands the trial solution u by

$$u(x,y) = \sum_{k=1}^{n} a_k \phi_k, \tag{55}$$

where a_k are the expansion coefficients to be determined and ϕ_k is a set of RBFs, for example,

$$\begin{aligned}
&\phi_k = (r_k^2 + c^2)^{N-3/2}, \; N = 1, 2, \ldots, \\
&\phi_k = r_k^{2N} \ln r_k, \; N = 1, 2, \ldots, \\
&\phi_k = \exp\left(-\frac{r_k^2}{a^2}\right), \\
&\phi_k = (r_k^2 + c^2)^{N-3/2} \exp\left(-\frac{r_k^2}{a^2}\right), \; N = 1, 2, \ldots,
\end{aligned} \tag{56}$$

where the radius function r_k is given by $r_k = \sqrt{(x-x_k)^2+(y-y_k)^2}$, while (x_k, y_k), $k = 1, \ldots, n$ are called source points. The constants a and c are shape parameters. In the below we take the first set of ϕ_k with $N = 2$ as trial functions, which is known as a multi-quadric RBF [Golberg, Chen and Karur (1996); Cheng, Golberg, Kansa and Zammito (2003)].

In this example we apply the multi-quadric radial basis function to solve the following nonlinear PDE:

$$\Delta u = 4u^3(x^2+y^2+a^2), \tag{57}$$

where $a = 4$ was fixed. The domain is an irregular domain with

$$\rho(\theta) = (\sin 2\theta)^2 \exp(\sin\theta) + (\cos 2\theta)^2 \exp(\cos\theta). \tag{58}$$

The exact solution is given by

$$u(x,y) = \frac{-1}{x^2+y^2-a^2}, \tag{59}$$

which is singular on the circle with a radius a.

Inserting Eq. (55) into Eq. (57) and placing some field points inside the domain to satisfy the governing equation and some points on the boundary to satisfy the boundary condition we can derive n NAEs to determine the n coefficients a_k. The source points (x_k, y_k), $k = 1, \ldots, n$ are uniformly distributed on a contour given by $R_0 + \rho(\theta_k)$, where $\theta_k = 2k\pi/n$. Under the following parameters $R_0 = 0.5$, $c = 0.5$, $\gamma = 0.1$, and $\varepsilon = 10^{-2}$, in Fig. 2(a) we show the residual errors obtained by ODV(F) and ODV(R), of which the ODV(F) is convergent with 294 iterations, and the ODV(R) is convergent with 336 iterations. It can be seen that the residual-error curve decays very fast at the first few steps. In Fig. 2(b) we compare the convergence rates of ODV(F) and ODV(R). The numerical solution is quite accurate with the maximum error being 3.32×10^{-3} for ODV(F), and 2.82×10^{-3} for ODV(R).

4.3 Example 3

In this example we apply the ODV(F) and ODV(R) to solve the following boundary value problem of nonlinear elliptic equation:

$$\Delta u(x,y) + \omega^2 u(x,y) + \varepsilon_1 u^3(x,y) = p(x,y). \tag{60}$$

While the exact solution is

$$u(x,y) = \frac{-5}{6}(x^3+y^3) + 3(x^2y+xy^2), \tag{61}$$

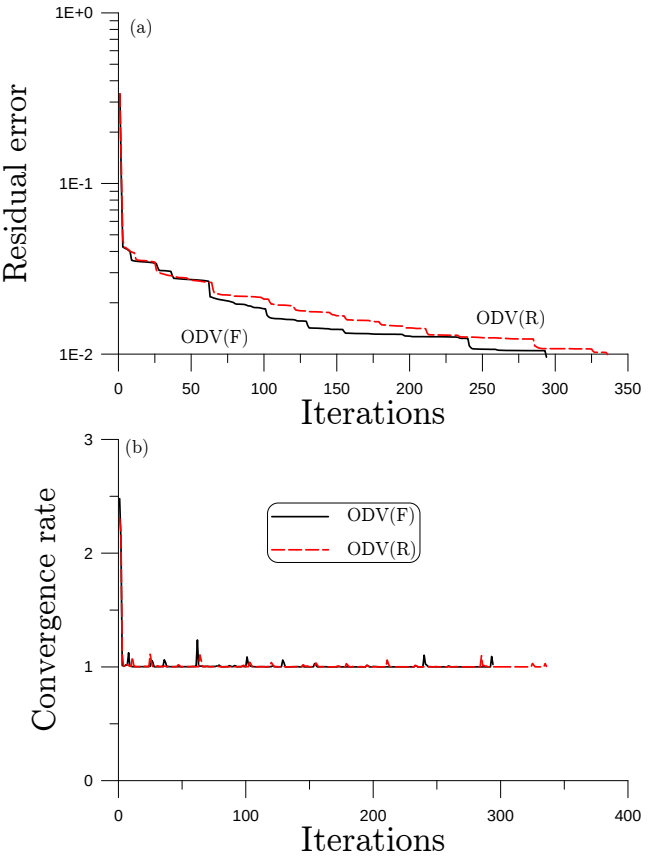

Figure 2: For example 2, comparing (a) the residual errors, and (b) the convergence rates of ODV(F) and ODV(R).

the exact p can be obtained by inserting the above u into Eq. (60).

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

$$\begin{aligned} F_{i,j} &= \frac{1}{(\Delta x)^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})+\frac{1}{(\Delta y)^2}(u_{i,j+1}-2u_{i,j}+u_{i,j-1}) \\ &+ \omega^2 u_{i,j}+\varepsilon_1 u_{i,j}^3-p_{i,j}=0. \end{aligned} \tag{62}$$

The boundary conditions can be obtained from the exact solution in Eq. (61). Here, $(x,y)\in[0,1]\times[0,1]$, $\Delta x=1/n_1$, and $\Delta y=1/n_2$.

Under the following parameters $n_1=n_2=13$, $\gamma=0.1$, $\varepsilon=10^{-3}$, $\omega=1$ and $\varepsilon_1=0.001$ we compute the solutions of the above system of NAEs by ODV(F) and

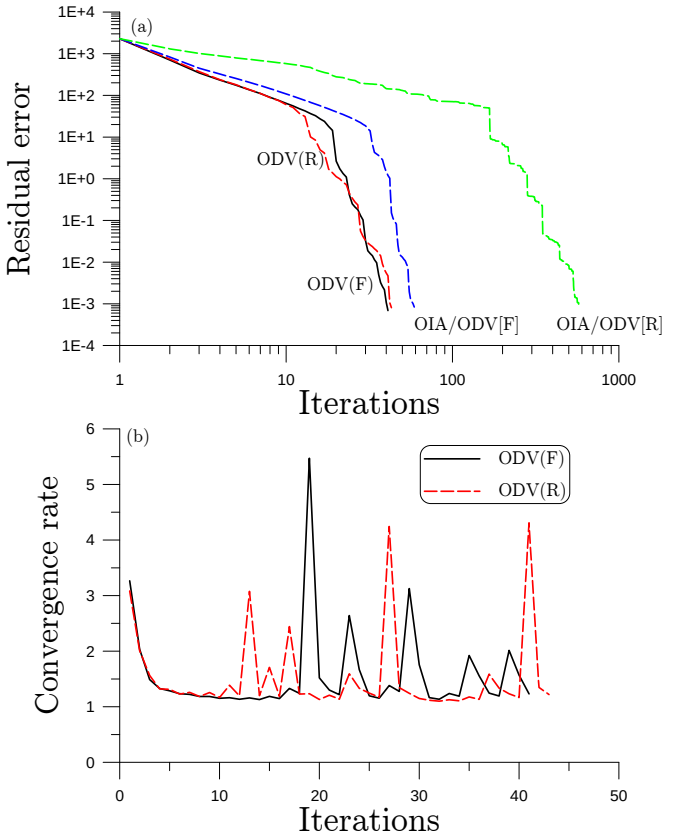

Figure 3: For example 3, comparing (a) the residual errors of ODV(F), ODV(R), OIA/ODV[F] and OIA/ODV[R], and (b) the convergence rates of ODV(F) and ODV(R).

ODV(R). In Fig. 3(a) we show the residual errors, of which the ODV(F) converges with only 41 iterations, and the ODV(R) requires 43 iterations. The reason for the fast convergence of ODV(F) and ODV(R) is shown in Fig. 3(b), where the convergence rate of ODV(R) is slightly lower than that of ODV(F). The maximum numerical error is about 5.2×10^{-6} for ODV(F) and ODV(R). Very accurate numerical results were obtained by the present ODV algorithms. The residual errors calculated by Liu, Dai and Atluri (2011) with OIA/ODV[F] and OIA/ODV[R] are also shown in Fig. 3. The numbers of iterations for ODV(F), ODV(R), OIA/ODV[F] and OIA/ODV[R] are, respectively, 41, 43, 59 and 575. Obviously, the ODV(F) and ODV(R) are faster than OIA/ODV[F], and much faster than OIA/ODV[R].

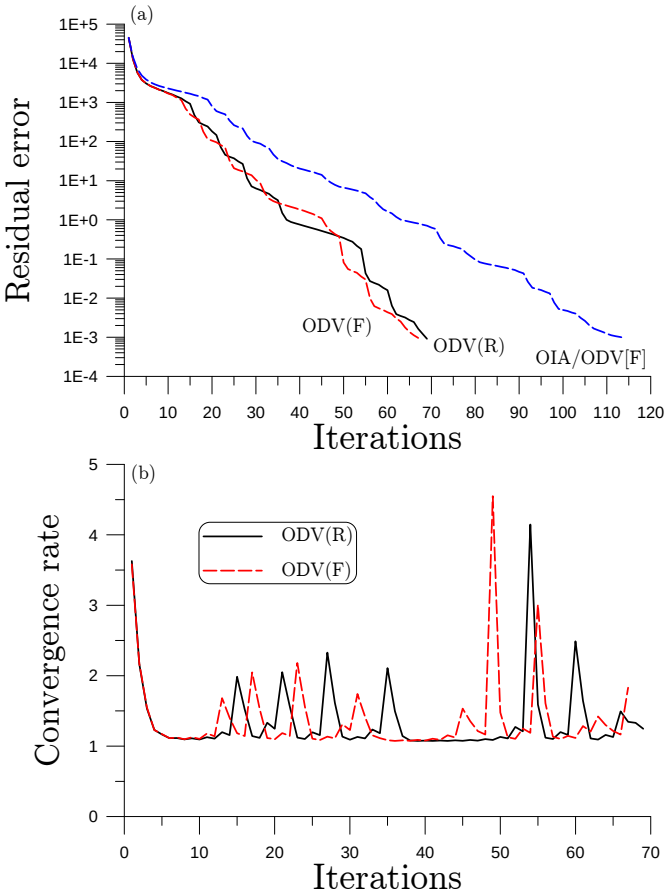

Figure 4: For example 4, comparing (a) the residual errors of ODV(F), ODV(R) and OIA/ODV[F], and (b) the convergence rates of ODV(F) and ODV(R).

4.4 Example 4

We consider a nonlinear heat conduction equation:

$$u_t = \alpha(x)u_{xx} + \alpha'(x)u_x + u^2 + h(x,t), \tag{63}$$

$$\alpha(x) = (x-3)^2, \;\; h(x,t) = -7(x-3)^2 e^{-t} - (x-3)^4 e^{-2t}, \tag{64}$$

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

By applying the ODV(F) and ODV(R) to solve the above equation in the domain

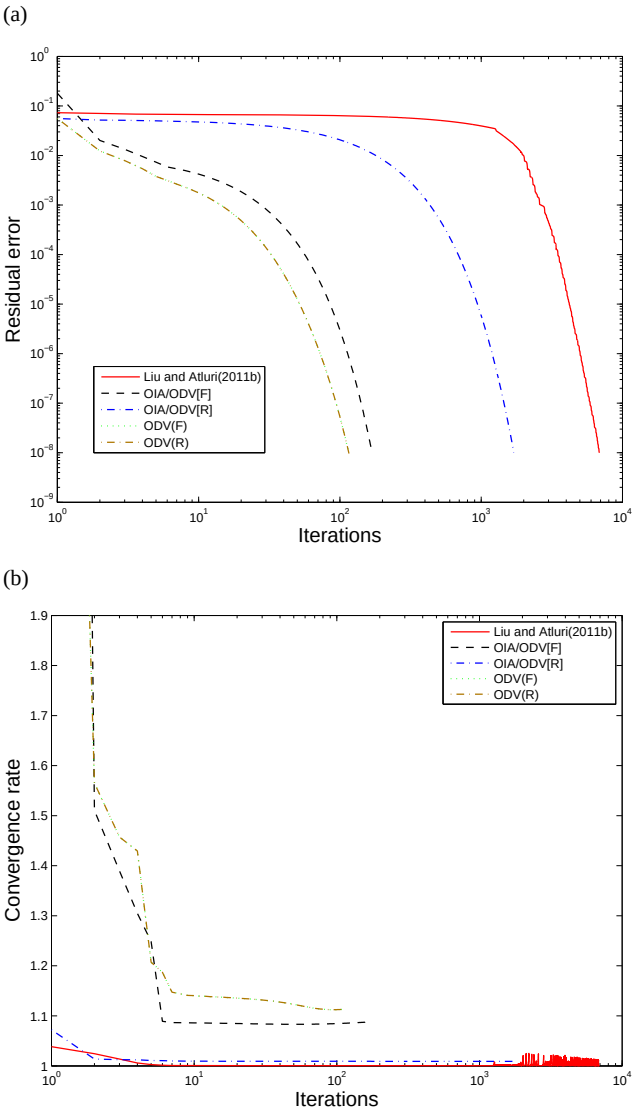

Figure 5: For example 5, with the NAEs from HB solved by ODV(F), ODV(R), OVDA [Liu and Atluri (2011b)], OIA/ODV[F] and OIA/ODV[R] [Liu, Dai and Atluri (2011)], comparing (a) the residual errors and (b) the convergence rates.

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. 4(a) we show the residual errors, which are convergent very fast with 69 iterations for ODV(R) with $\gamma = 0.1$, and 67 iterations for ODV(F) with $\gamma = 0.11$.

The convergence rates of ODV(F) and ODV(R) are compared in Fig. 4(b). The numerical results are quite accurate with the maximum error being 3.3×10^{-3}. The residual error calculated by Liu, Dai and Atluri (2011) with OIA/ODV[F] is also shown in Fig. 4. The numbers of iterations for ODV(F), ODV(R) and OIA/ODV[F] are, respectively, 67, 69 and 114. Obviously, the ODV(F) and ODV(R) are faster than OIA/ODV[F].

4.5 Example 5

In this example, we solve the widely investigated Duffing equation by applying the Harmonic Balance Method (HB). The non-dimensionalized Duffing equation is given as follows:

$$\ddot{x} + 2\xi\dot{x} + x + x^3 = F\sin\omega t, \tag{65}$$

where x is a non-dimensionalized displacement, ξ is a damping ratio, F is the amplitude of external force, and ω is the excitation frequency of external force. Traditionally, to employ the standard harmonic balance method (HB), the solution of x is sought in the form of a truncated Fourier series expansion:

$$x(t) = x_0 + \sum_{n=1}^{N} [x_{2n-1}\cos(n\omega t) + x_{2n}\sin(n\omega t)], \tag{66}$$

where N is the number of harmonics used in the truncated Fourier series, and x_n, $n = 0, 1, \ldots, 2N$ are the unknown coefficients to be determined in the HB method. We differentiate $x(t)$ with respect to t, leading to

$$\dot{x}(t) = \sum_{n=1}^{N} [-n\omega x_{2n-1}\sin(n\omega t) + n\omega x_{2n}\cos(n\omega t)], \tag{67}$$

$$\ddot{x}(t) = \sum_{n=1}^{N} [-(n\omega)^2 x_{2n-1}\cos(n\omega t) - (n\omega)^2 x_{2n}\sin(n\omega t)]. \tag{68}$$

The nonlinear term in Eq. (65) can also be expressed in terms of the truncated Fourier series with N harmonics kept:

$$x^3(t) = r_0 + \sum_{n=1}^{N} [r_{2n-1}\cos(n\omega t) + r_{2n}\sin(n\omega t)]. \tag{69}$$

Thus, considering the Fourier series expansion as well as the orthogonality of trigonometric functions, r_n, $n = 0, 1, \ldots, 2N$ are obtained by the following formu-

las:

$$r_0 = \frac{1}{2\pi}\int_0^{2\pi}\{x_0 + \sum_{n=1}^{N}[x_{2n-1}\cos(n\theta) + x_{2n}\sin(n\theta)]\}^3 d\theta, \tag{70}$$

$$r_{2n-1} = \frac{1}{\pi}\int_0^{2\pi}\{x_0 + \sum_{n=1}^{N}[x_{2n-1}\cos(n\theta) + x_{2n}\sin(n\theta)]\}^3 \cos(n\theta) d\theta, \tag{71}$$

$$r_{2n} = \frac{1}{\pi}\int_0^{2\pi}\{x_0 + \sum_{n=1}^{N}[x_{2n-1}\cos(n\theta) + x_{2n}\sin(n\theta)]\}^3 \sin(n\theta) d\theta. \tag{72}$$

Note that because of the orthogonality of trigonometric functions, $r_n,\ n = 0,1,\ldots,2N$ can be achieved without integration. Next, substituting Eqs. (66)-(69) into Eq. (65), and collecting the terms associated with each harmonic $\cos(n\theta)$, $\sin(n\theta)$, $n = 1,\ldots,N$, we finally obtain a system of NAEs in a vector form:

$$(\mathbf{A}^2 + 2\xi\mathbf{A} + \mathbf{I}_{2N+1})\mathbf{Q}_x + \mathbf{R}_x = F\mathbf{H}, \tag{73}$$

where

$$\mathbf{Q}_x = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2N} \end{bmatrix},\ \mathbf{R}_x = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{2N} \end{bmatrix},\ \mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{J}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{J}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{J}_N \end{bmatrix},\ \mathbf{J}_n = n\begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}. \tag{74}$$

One should note that $r_n,\ n = 0,1,\ldots,2N$ are analytically expressed in terms of the coefficients $x_n,\ n = 0,1,\ldots,2N$, which makes the HB not immediately ready for application. Later, we will introduce a post-conditioned harmonic balance method (PCHB). In the present example, we can solve the Duffing equation by employing the standard HB method with the help of Mathematica. In doing so, one has no difficulty to handle the symbolic operations to evaluate r_n, and a large number of harmonics can be taken into account. In the current case, we apply the HB

method with 8 harmonics to the Duffing equation (65), and then arrive at a system of NAEs in Eq. (73). The NAEs are to be solved by ODV(F), ODV(R) and OVDA algorithms. To start with, we set $\xi = 0.1$, $\omega = 2$, $F = 1.25$ and the initial values of x_n to be zeros. The stop criterion is taken as $\varepsilon = 10^{-8}$.

We compare the residual errors obtained by ODV(F), ODV(R) and OVDA [Liu and Atluri (2011b)] in Fig. 5(a). It shows that the ODV algorithms converge much faster than OVDA. Specifically, the iterations to achieve convergence in ODV(F), ODV(R) and OVDA algorithms are, respectively, 116, 116 and 6871. The convergence ratio between ODV methods and OVDA is an amazing 59.2. Furthermore, the convergence rates for these methods are also provided in Fig. 5(b), from which we can see that the rate of convergence of ODV(F) is the same as that of ODV(R), while the convergence rates of the ODV methods are much superior to OVDA. In this case, the peak amplitude is $A = 0.43355$. The numerical results of the coefficients $x_n,\ n = 0, 1, \ldots, 16$ are given in Table 1. It shows that all the coefficients of odd modes are zeros and the absolute values of odd modes decrease with the increasing mode numbery as expected.

Table 1: The coefficients of x_n obtained by solving the NAEs from the Harmonic Balance Method (HB)

x_n	value	x_n	value
x_0	0	x_9	-0.000000567563101
x_1	-0.059988154613779	x_{10}	-0.000000609434275
x_2	-0.428790543120576	x_{11}	0
x_3	0	x_{12}	0
x_4	0	x_{13}	0.000000001021059
x_5	0.000254872564318	x_{14}	0.000000000587338
x_6	0.000525550273254	x_{15}	0
x_7	0	x_{16}	0
x_8	0		

We also applied the numerical methods of OIA/ODV[F] and OIA/ODV[R] [Liu, Dai and Atluri (2011)] to solve this problem. The residual errors and the convergence rates of these methods are compared with the present ODV(F) and ODV(R) in Fig. 5. In Table 2 we summarize the numbers of iterations of these methods.

In summary, the HB method can simply, efficiently, and accurately solve problems with complex nonlinearity with the help of symbolic operation software, i.e. Mathematica, and Maple. However, if many harmonics are considered, Eq. (73) will become ill-conditioned, and the expressions for the nonlinear terms, Eqs. (70)-(72)

Table 2: By solving the NAEs from HB, comparing the numbers of iterations

Methods	Numbers of iterations
OVDA [Liu and Atluri (2011b)]	6871
OIA/ODV[F] [Liu, Dai and Atluri (2011)]	169
OIA/ODV[R] [Liu, Dai and Atluri (2011)]	1705
Present ODV(F)	116
Present ODV(R)	116

become much more complicated. In order to employ the HB method to a complex system with higher modes, we can apply the postconditioner as first developed by Liu, Yeih and Atluri (2009) to Eq. (73), which is obtained from a multi-scale Trefftz boundary-collocation method for solving the Laplace equation:

$$\tilde{\mathbf{Q}}_x = \mathbf{T}_R \mathbf{Q}_x, \tag{75}$$

where $\mathbf{T}_R$ is a postconditioner given by

$$\theta_j = \frac{2j\pi}{n}, \quad n = 2N+1, \ j = 0,1,\ldots,2N, \tag{76}$$

$$\mathbf{T}_R = \begin{bmatrix} 1 & \cos\theta_0 & \sin\theta_0 & \cdots & \cos(N\theta_0) & \sin(N\theta_0) \\ 1 & \cos\theta_1 & \sin\theta_1 & \cdots & \cos(N\theta_1) & \sin(N\theta_1) \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 1 & \cos\theta_{2N} & \sin\theta_{2N} & \cdots & \cos(N\theta_{2N}) & \sin(N\theta_{2N}) \end{bmatrix}. \tag{77}$$

The inverse of $\mathbf{T}_R$ is

$$\mathbf{T}_R^{-1} = \frac{2}{n} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \cdots & \frac{1}{2} & \frac{1}{2} \\ \cos\theta_0 & \cos\theta_1 & \cdots & \cos\theta_{2N-1} & \cos\theta_{2N} \\ \sin\theta_0 & \sin\theta_1 & \cdots & \sin\theta_{2N-1} & \sin\theta_{2N} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \cos(N\theta_0) & \cos(N\theta_1) & \cdots & \cos(N\theta_{2N-1}) & \cos(N\theta_{2N}) \\ \sin(N\theta_0) & \sin(N\theta_1) & \cdots & \sin(N\theta_{2N-1}) & \sin(N\theta_{2N}) \end{bmatrix} \tag{78}$$

Now, in addition the unknown $\mathbf{Q}_x$ in Eq. (73) we can define similarly

$$\tilde{\mathbf{R}}_x = \mathbf{T}_R \mathbf{R}_x, \quad \tilde{\mathbf{H}} = \mathbf{T}_R \mathbf{H}, \quad \mathbf{A} = \mathbf{T}_R^{-1} \mathbf{D} \mathbf{T}_R, \tag{79}$$

and thus rearrange Eq. (73) to:

$$
\begin{aligned}
&(\mathbf{A}^2+2\xi\mathbf{A}+\mathbf{I}_{2N+1})\mathbf{T}_R^{-1}\tilde{\mathbf{Q}}_x+\mathbf{T}_R^{-1}\tilde{\mathbf{R}}_x=F\mathbf{T}_R^{-1}\tilde{\mathbf{H}}_x,\\
&(\mathbf{T}_R^{-1}\mathbf{D}\mathbf{T}_R\mathbf{T}_R^{-1}\mathbf{D}\mathbf{T}_R+2\xi\mathbf{T}_R^{-1}\mathbf{D}\mathbf{T}_R+\mathbf{T}_R^{-1}\mathbf{T}_R)\mathbf{T}_R^{-1}\tilde{\mathbf{Q}}_x+\mathbf{T}_R^{-1}\tilde{\mathbf{R}}_x=F\mathbf{T}_R^{-1}\tilde{\mathbf{H}}_x,\\
&(\mathbf{T}_R^{-1}\mathbf{D}^2\mathbf{T}_R+2\xi\mathbf{T}_R^{-1}\mathbf{D}\mathbf{T}_R+\mathbf{T}_R^{-1}\mathbf{T}_R)\mathbf{T}_R^{-1}\tilde{\mathbf{Q}}_x+\mathbf{T}_R^{-1}\tilde{\mathbf{R}}_x=F\mathbf{T}_R^{-1}\tilde{\mathbf{H}}_x.
\end{aligned}
$$

Finally, by dropping out $\mathbf{T}_R^{-1}$ we can obtain

$$(\mathbf{D}^2+2\xi\mathbf{D}+\mathbf{I}_{2N+1})\tilde{\mathbf{Q}}_x+\tilde{\mathbf{R}}_x=F\tilde{\mathbf{H}}. \tag{80}$$

By using Eqs. (75), (79) and (77) it is interesting that

$$
\tilde{\mathbf{Q}}_x=\mathbf{T}_R\mathbf{Q}_x=\begin{bmatrix} x(\theta_0)\\ x(\theta_1)\\ \vdots\\ x(\theta_{2N})\end{bmatrix},\ \tilde{\mathbf{R}}_x=\mathbf{T}_R\mathbf{R}_x=\begin{bmatrix} x^3(\theta_0)\\ x^3(\theta_1)\\ \vdots\\ x^3(\theta_{2N})\end{bmatrix},\ \tilde{\mathbf{H}}=\mathbf{T}_R\mathbf{H}=\begin{bmatrix} \sin(\theta_0)\\ \sin(\theta_1)\\ \vdots\\ \sin(\theta_{2N})\end{bmatrix}. \tag{81}
$$

So now, we can solve for the unknown $x(\theta_k),\ k=0,1,\ldots,2N$ in Eq. (80), instead of the unknown $x_k,\ k=0,1,\ldots,2N$ in Eq. (73). Here, the $2N+1$ coefficients x_n are recast into the variables $x(\theta_n)$ which are selected at $2N+1$ equally spaced phase angle points over a period of oscillation by a constant Fourier transformation matrix. In the present HB method, namely the post-conditioned harmonic balance method (PCHB), the analytical expression for nonlinear term is no longer necessary. Thus, for a large number of harmonics, the PCHB can be implemented more easily. In this example, we apply the PCHB with 8 harmonics to the Duffing equation (65) by solving a system of NAEs in Eq. (80).

We compare the residual errors and convergence rates of ODV(R), ODV(F) and OVDA [Liu and Atluri (2011b)] in Figs. 6(a) and 6(b). It shows that the ODV(R) and ODV(F) converge up to roughly one hundred and fifty times faster than OVDA, and the numbers of iterations of ODV(R) and ODV(F) coincide just as we found in example 1. Specifically, the iterations of ODV(R), ODV(F) and OVDA are, respectively, 157, 157 and 23995. The ratio between the convergent speeds of ODV and OVDA is amazingly 152.8. In this case, the peak amplitude is also found to be $A=0.43355$.

The coefficients of $x(\theta_n)$ obtained by solving the NAEs from PCHB are listed in Table 3. It can be seen from Table 3 that the first nine values are negative and the last eight values are positive, which is as expected. Because the $2N+1$ values represent the displacements at $2N+1$ equally spaced angular phase points over a

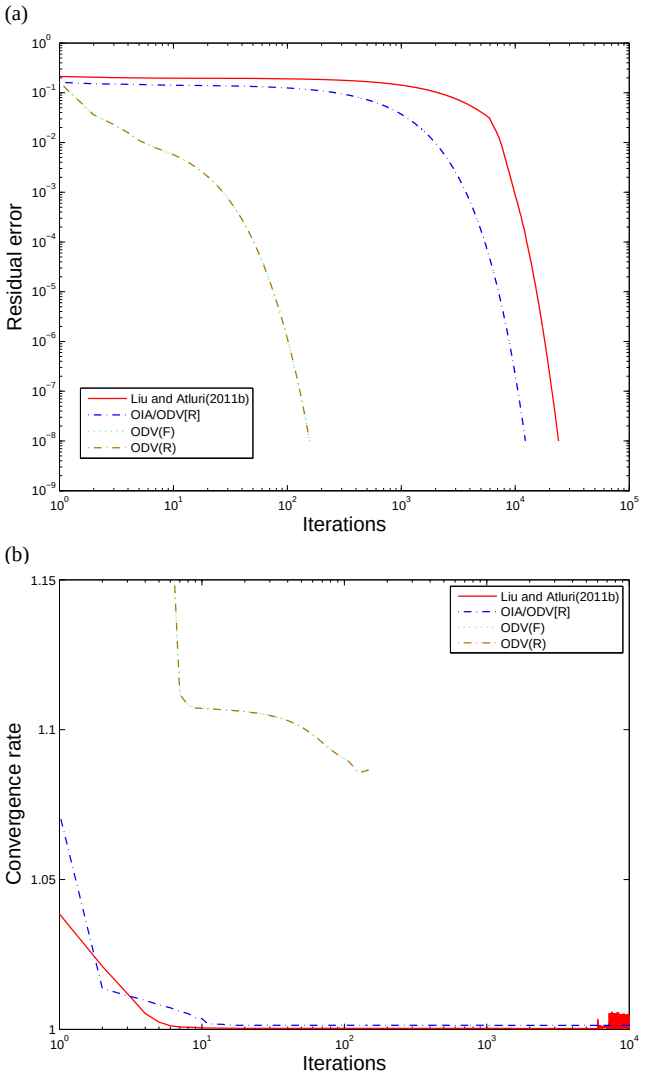

Figure 6: For example 5, with the NAEs from PCHB solved by ODV(F), ODV(R), OVDA [Liu and Atluri (2011b)], OIA/ODV[F] and OIA/ODV[R] [Liu, Dai and Atluri (2011)], comparing (a) the residual errors and (b) the convergence rates.

period. As in a period of oscillation, the displacements of the first half period and those of the second half period are expected to be having opposite values.

The coefficients of x_n obtained by solving the NAEs from PCHB are listed in Table 4. It should be noted that the values corresponding to odd modes are not exact

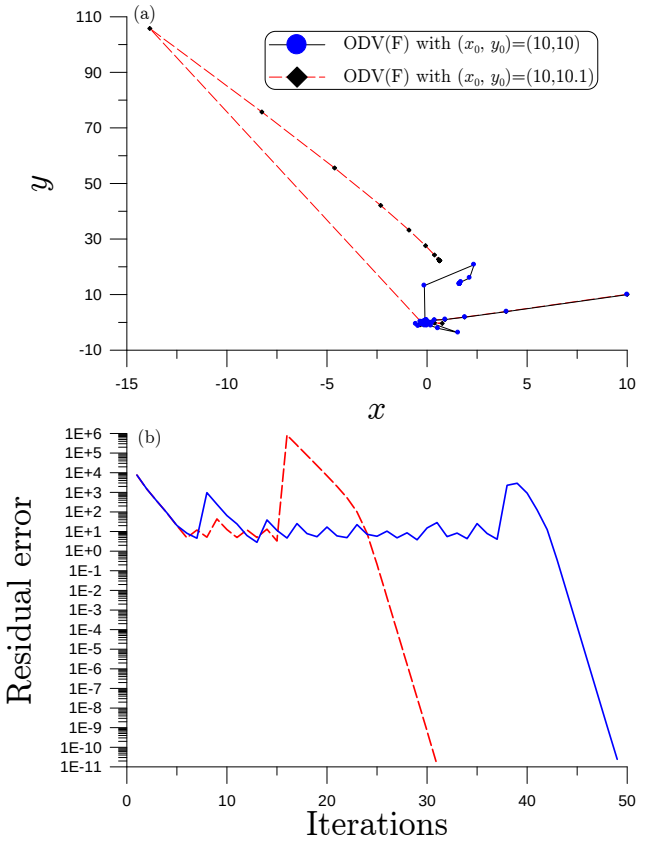

Figure 7: For example 6, solved by ODV(F), comparing (a) the solution paths, and (b) the residual errors for slightly different initial conditions.

zeros, which is different from the HB method. However, all these values are very small and almost close to zero. The reason for this is that in Eq. (66) the trial function of $x(t)$ is exactly equal to the Fourier series expansion on the right hand side only when N approaches infinity.

We also applied the numerical methods of OIA/ODV[F] and OIA/ODV[R] [Liu, Dai and Atluri (2011)] to solve this problem. The residual errors and the convergence rates of these methods are compared with the present ODV(F) and ODV(R) in Fig. 6. In Table 5 we summarize the numbers of iterations of these methods.

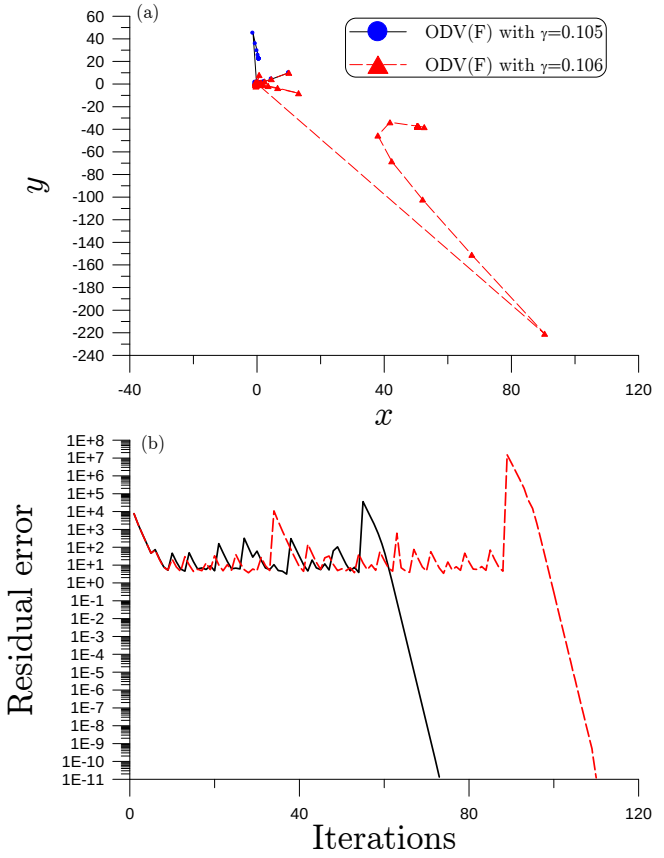

Figure 8: For example 6, solved by ODV(F), comparing (a) the solution paths, and (b) the residual errors for slightly different values of parameter γ.

4.6 Example 6

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

$$F_1(x,y) = x^3 - 3xy^2 + a_1(2x^2 + xy) + b_1y^2 + c_1x + a_2y = 0, \tag{82}$$

$$F_2(x,y) = 3x^2y - y^3 - a_1(4xy - y^2) + b_2x^2 + c_2 = 0, \tag{83}$$

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 (2008a) 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 (2008a) found three solutions by guessing three different initial values, and Liu,

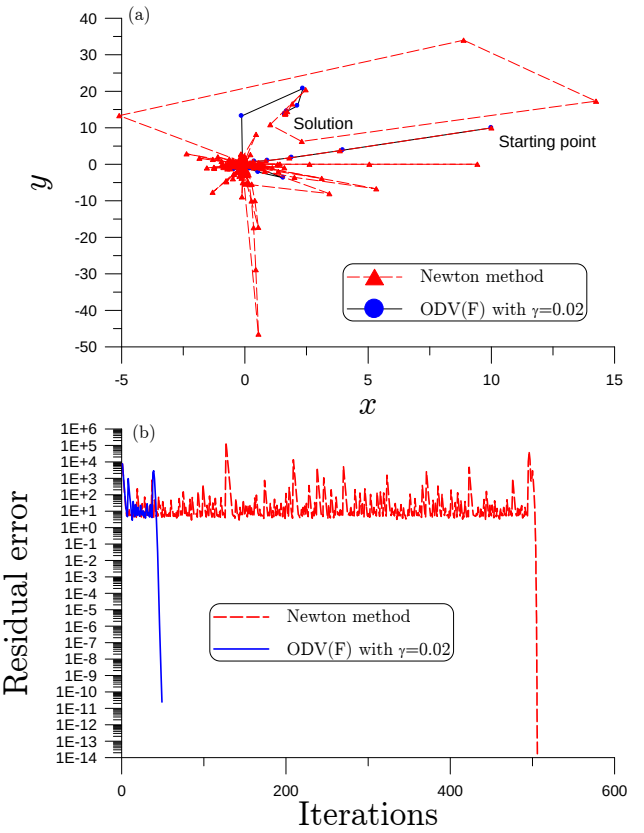

Figure 9: For example 6, solved by ODV(F) and the Newton method, comparing (a) the solution paths, and (b) the residual errors.

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 ODV(F) for four cases with (a) $\gamma = 0.105$, (b) $\gamma = 0.106$, (c) $\gamma = 0.02$, and (d) $\gamma = 0.05$ under a convergence criterion $\varepsilon = 10^{-10}$. The residual errors of (F_1, F_2) are all smaller than 10^{-10}. For case (a) we find the second root (x, y) = (0.6277425, 22.2444123) through 73 iterations. For case (b) we find the fourth root (x, y) = (50.46504, -37.2634179) through 110 iterations. For case (c) we find the fifth root (x, y) = (1.6359718, 13.8476653) through 49 iterations. For case (d) we find the first root (x, y) = (-50.3970755, -0.8042426) through 259 iterations. To find these solutions the FTIM [Liu and Atluri (2008a)] spent 792 iterations for the first root, 1341 iter-

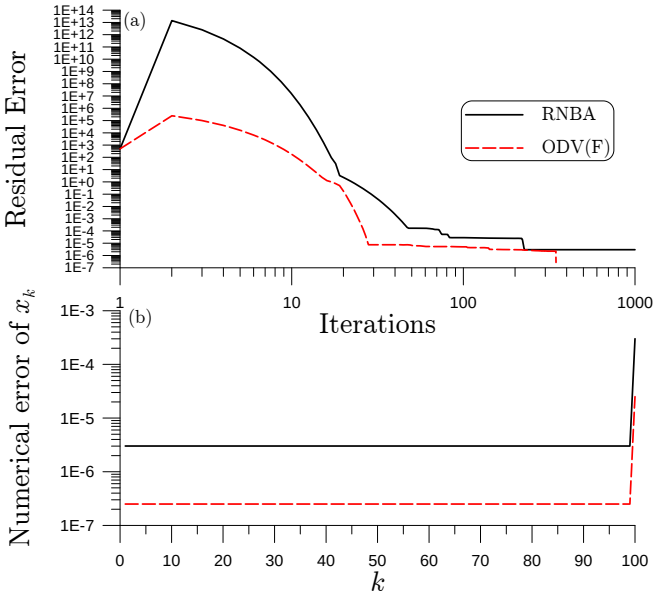

Figure 10: For example 7, solved by ODV(F) and the RNBA [Liu and Atluri (2011a)], comparing (a) the residual errors, and (b) the numerical errors.

Table 3: The unknown $x(\theta_n)$ obtained by solving the NAEs from a Post-Conditioned Harmonic Balance Method (PCHB)

$x(\theta_n)$	value	$x(\theta_n)$	value
$x(\theta_0)$	-0.059733847308209	$x(\theta_9)$	0.137264260203943
$x(\theta_1)$	-0.210250668263990	$x(\theta_{10})$	0.276231988329339
$x(\theta_2)$	-0.332939493461817	$x(\theta_{11})$	0.378380328434565
$x(\theta_3)$	-0.410923683734554	$x(\theta_{12})$	0.429381360255661
$x(\theta_4)$	-0.433072441477243	$x(\theta_{13})$	0.421862418920612
$x(\theta_5)$	-0.396170461505062	$x(\theta_{14})$	0.356943737674487
$x(\theta_6)$	-0.305603234068375	$x(\theta_{15})$	0.243969709017363
$x(\theta_7)$	-0.174180100211517	$x(\theta_{16})$	0.098603616046807
$x(\theta_8)$	-0.019763488665098		

ations for the fourth root, and 1474 iterations for the third root, while the OVDA [Liu and Atluri (2011b)] spent 68 iterations to find the fifth root. Obviously, the performance of present ODV(F) is better than these algorithms.

In Fig. 7 we compare the solution paths and residual errors for two cases with the

Table 4: The coefficients of x_n obtained by solving the NAEs from PCHB

x_n	value	x_n	value
x_0	0.000000000011120	x_9	-0.000000567584459
x_1	-0.059988153310220	x_{10}	-0.000000609447325
x_2	-0.428790541309004	x_{11}	0.000000000004109
x_3	-0.000000000000186	x_{12}	-0.000000000000276
x_4	0.000000000000063	x_{13}	0.000000001015925
x_5	0.000254872556411	x_{14}	0.000000000583850
x_6	0.000525550267339	x_{15}	-0.000000000000765
x_7	0.000000000000008	x_{16}	0.000000000000940
x_8	0.000000000000001		

Table 5: By solving the NAEs from PCHB, comparing the numbers of iterations

Methods	Numbers of iterations
OVDA [Liu and Atluri (2011b)]	23995
OIA/ODV[F] [Liu, Dai and Atluri (2011)]	Very large (omitted)
OIA/ODV[R] [Liu, Dai and Atluri (2011)]	12275
Present ODV(F)	157
Present ODV(R)	157

same $\gamma = 0.02$ but with a slightly different initial conditions with (a) $(x_0, y_0) = (10, 10)$ and (b) $(x_0, y_0) = (10, 10.1)$. Case (a) tends to the fifth root, but case (b) tends to the second root. In Fig. 8 we compare the solution paths and residual errors for two cases with the same initial condition $(x_0, y_0) = (10, 10)$ but with a slightly different (a) $\gamma = 0.105$ and (b) $\gamma = 0.106$. When case (a) tends to the second root, case (b) tends to the fourth root. The above results show that for this problem the ODV(F) is sensitive to initial condition and the value of the parameter γ. However, no matter what cases are considered, the present ODV(F) is always available to obtain one of the solutions.

In the last case we compare the Newton method with the ODV(F) with $\gamma = 0.02$. Starting from the same initial value of $(x_0, y_0) = (10, 10)$, and under the same convergence criterion 10^{-10}, when the Newton method converges with 506 iterations, it is amazingly that the ODV(F) only needs 49 iterations to obtain the same fifth root $(x, y) = (1.6359718, 13.8476653)$. We compare in Fig. 9(a) the solution paths and in Fig. 9(b) the residual errors of the above two methods, from which we can see that the solution path of the Newton method spends many steps around the zero point and is much irregular than the solution path generated by the ODV(F). From

the solution paths as shown in Fig. 9(a), and the residual errors as shown in Fig. 9(b) we can observe that the mechanism of both the Newton method and the ODV(F) 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 method is too long, which causes a slower convergence than ODV(F). So for this problem the ODV(F) is ten times faster than the Newton method.

Remark: In solving linear systems, van den Doel and Ascher (2011) have found that the fastest practical methods of the family of faster gradient descent methods in general generate the *chaotic dynamical systems*. Indeed, in an earlier time, Liu (2011) has developed a relaxed steepest descent method for solving linear systems, and found that the iterative dynamics can undergo a Hopf bifurcation with an intermittent behavior [Liu (2000b, 2007)] appeared in the residual-error descent curve. Similarly, Liu and Atluri (2011b) also found the intermittent behavior of the iterative dynamics by using the Optimal Vector Driven Algorithm (OVDA) to solve nonlinear systems. *The above sensitivity to initial condition and parameter value by using the ODV(F) also hints that the iterative dynamics generated by the present ODV(F) is chaotic. It is interesting that in order to achieve a faster convergence the iterative dynamics generated by the algorithm is usually chaotic.*

4.7 Example 7

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

$$F_i = x_i + \sum_{j=1}^{j=n} x_j - (n+1), \;\; i = 1,\ldots,n-1, \tag{84}$$

$$F_n = \prod_{j=1}^{j=n} x_j - 1, \tag{85}$$

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 method, 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 method diverged quite rapidly. Now, we apply our algorithm to this tough problem with $n = 100$. However, Liu and Atluri (2011a) using the residual-norm based algorithm (RNBA) can also solve this problem without any difficulty.

Under the convergence criterion $\varepsilon = 10^{-6}$ and with the initial guess $x_i = 0.5$ we solve this problem with $n = 100$ by using the RNBA, whose residual error and numerical error are shown in Fig. 10 by the solid lines. The accuracy is very good

with the maximum error being 3×10^{-4}. Under the same conditions we apply the ODV(F) to solve this problem with $\gamma = 0.1$. When the RNBA does not converge within 1000 iterations, the residual error for the ODV(F) as shown in Fig. 10(a) by the dashed line can converge with 347 iterations, and its numerical error as shown in Fig. 10(b) by the dashed line is much smaller than that of RNBA, with the maximum error being 2.5×10^{-5}.

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 ODV(F), when the residual vector **F** is a primary driving vector; and the ODV(R) when the descent vector **R** is a primary driving vector. The ODV methods *have a better computational efficiency and accuracy than other algorithms*, e.g., the FTIM, the RNBA, the Newton method, the OVDA and the OIA/ODV, in solving the nonlinear algebraic equations. We also applied the algorithms of ODV(F) and ODV(R) to solve the Duffing equation by using a harmonic balance method and a post-conditioned harmonic balance method. The computational efficiency was very good to treat such a highly nonlinear Duffing oscillator, using a large number of harmonics in the Harmonic Balance Method. Amazingly, the ratio between the convergence speeds of ODV and OVDA is 152.8. *Our earlier and present studies revealed that in order to achieve a faster convergence the iterative dynamics generated by the algorithm is essentially chaotic*.

Acknowledgement: This work was supported by the US Army Research Labs, under a Collaborative Research Agreement with UCI. Taiwan's National Science Council project NSC-100-2221-E-002-165-MY3 granted to the first author is also highly appreciated.

References

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.

Chang, C. W.; Liu, C.-S. (2009): A fictitious time integration method for backward advection-dispersion equation. *CMES: Computer Modeling in Engineering & Sciences*, vol. 51, pp. 261-276.

Cheng, A. H. D.; Golberg, M. A.; Kansa, E. J.; Zammito, G. (2003): Exponential convergence and *H-c* multiquadric collocation method for partial differential equations. *Numer. Meth. Part. Diff. Eqs.*, vol. 19, pp. 571-594.

Chi, C. C.; Yeih, W.; Liu, C.-S. (2009): A novel method for solving the Cauchy problem of Laplace equation using the fictitious time integration method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 47, pp. 167-190.

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

Golberg, M. A.; Chen, C. S.; Karur, S. R. (1996): Improved multiquadric approximation for partial differential equations. *Eng. Anal. Bound. Elem.*, vol. 18, pp. 9-17.

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. Nonlinear Sci. Numer. Simul.*, vol. 11, pp. 435-450.

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.

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

Liu, C.-S. (2000b): Intermittent transition to quasiperiodicity demonstrated via a circular differential equation. *Int. J. Non-Linear Mech.*, vol. 35, pp. 931-946.

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

Liu, C.-S. (2005): Nonstandard group-preserving schemes for very stiff ordinary differential equations. *CMES: Computer Modeling in Engineering & Sciences*, vol. 9, pp. 255-272.

Liu, C.-S. (2007): A study of type I intermittency of a circular differential equation under a discontinuous right-hand side. *J. Math. Anal. Appl.*, vol. 331, pp. 547-566.

Liu, C.-S. (2008a): 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. (2008b): A fictitious time integration method for two-dimensional quasi-

linear elliptic boundary value problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 33, pp. 179-198.

Liu, C.-S. (2009): 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 revision of relaxed steepest descent method from the dynamics on an invariant manifold. *CMES: Computer Modeling in Engineering & Sciences*, vol. 80, pp. 57-86.

Liu, C.-S.; Atluri, S. N. (2008a): 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. (2008b): A novel fictitious time integration method for solving the discretized inverse Sturm-Liouville problems, for specified eigenvalues. *CMES: Computer Modeling in Engineering & Sciences*, vol. 36, pp. 261-285.

Liu, C.-S.; Atluri, S. N. (2008c): A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to non-linear optimization. *CMES: Computer Modeling in Engineering & Sciences*, vol. 34, pp. 155-178.

Liu, C.-S.; Atluri, S. N. (2009): A fictitious time integration method for the numerical solution of the Fredholm integral equation and for numerical differentiation of noisy data, and its relation to the filter theory. *CMES: Computer Modeling in Engineering & Sciences*, vol. 41, pp. 243-261.

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 Sci. Tech.*, vol. 9, 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,* in press.

Liu, C.-S.; Kuo, C. L. (2011): A dynamical Tikhonov regularization method for

solving nonlinear ill-posed problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 76, pp. 109-132.

Liu, C.-S.; Yeih, W.; Atluri, S. N. (2009): On solving the ill-conditioned system $\mathbf{Ax} = \mathbf{b}$: general-purpose conditioners obtained from the boundary-collocation solution of the Laplace equation, using Trefftz expansions with multiple length scales. *CMES: Computer Modeling in Engineering & Sciences*, vol. 44, pp. 281-311.

Liu, C.-S.; Yeih, W.; Atluri, S. N. (2010): An enhanced fictitious time integration method for non-linear algebraic equations with multiple solutions: boundary layer, boundary value and eigenvalue problems. *CMES: Computer Modeling in Engineering & Sciences*, vol. 59, pp. 301-323.

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.

Tsai, C. C.; Liu, C.-S.; Yeih, W. (2010): Fictitious time integration method of fundamental solutions with Chebyshev polynomials for solving Poisson-type nonlinear PDEs. *CMES: Computer Modeling in Engineering & Sciences*, vol. 56, pp. 131-151.

van den Doel, K.; Ascher, U. (2011): The chaotic nature of faster gradient descent methods. *J. Sci. Comput.*, DOI 10.1007/s10915-011-9521-3.