 CMES, vol.81, no.4, pp.335-362, 2011

Iterative Solution of a System of Nonlinear Algebraic Equations $\mathbf{F}(\mathbf{x}) = 0$, Using $\dot{\mathbf{x}} = \lambda[\alpha\mathbf{R} + \beta\mathbf{P}]$ or $\dot{\mathbf{x}} = \lambda[\alpha\mathbf{F} + \beta\mathbf{P}^*]$ R is a Normal to a Hyper-Surface Function of F, P Normal to R, and $\mathbf{P}^*$ Normal to F

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

Abstract: To solve an ill- (or well-) conditioned system of Nonlinear Algebraic Equations (NAEs): $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, we define a scalar hyper-surface $h(\mathbf{x},t) = 0$ in terms of $\mathbf{x}$, and a monotonically increasing scalar function $Q(t)$ where t is a time-like variable. We define a vector $\mathbf{R}$ which is related to $\partial h/\partial \mathbf{x}$, and a vector $\mathbf{P}$ which is normal to $\mathbf{R}$. We define an Optimal Descent Vector (ODV): $\mathbf{u} = \alpha\mathbf{R} + \beta\mathbf{P}$ where α and β are optimized for fastest convergence. Using this ODV [$\dot{\mathbf{x}} = \lambda\mathbf{u}$], we derive an Optimal Iterative Algorithm (OIA) to solve $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. We also propose an alternative Optimal Descent Vector [$\mathbf{u} = \alpha\mathbf{F} + \beta\mathbf{P}^*$] where $\mathbf{P}^*$ is normal to $\mathbf{F}$. We demonstrate the superior performance of these two alternative OIAs with ODVs to solve NAEs arising out of the weak-solutions for several ODEs and PDEs. More importantly, we demonstrate the applicability of solving simply, most efficiently, and highly accurately, the Duffing equation, using 8 harmonics in the Harmonic Balance Method to find a weak-solution in time.

Keywords: Nonlinear algebraic equations, Perpendicular operator, Optimal Descent Vector (ODV), Optimal Iterative Algorithm (OIA), Optimal Iterative Algorithm with an Optimal Descent Vector (OIA/ODV), Invariant-manifold

1 Introduction and motivation

For the following nonlinear algebraic equations (NAEs):

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

we will propose numerical algorithms based on a space-time invariant manifold. For solving Eq. (1), the vector homotopy method, as initiated by Davidenko (1953),

[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

significantly enhances the convergence property from a local convergence to a global convergence. The vector homotopy method is based on the construction of a vector homotopy function $\mathbf{H}(\mathbf{x},\tau)=\tau\mathbf{F}(\mathbf{x})+(1-\tau)\mathbf{G}(\mathbf{x})$, which serves the objective of continuously transforming a vector function $\mathbf{G}(\mathbf{x})$, whose zero points are easily detected, into $\mathbf{F}(\mathbf{x})$ by introducing a homotopy parameter τ. The homotopy parameter τ can be treated as a time-like variable, such that $\mathbf{H}(\mathbf{x},\tau=0)=\mathbf{G}(\mathbf{x})$ and $\mathbf{H}(\mathbf{x},\tau=1)=\mathbf{F}(\mathbf{x})$. Hence we can construct a system of ODEs by keeping $\mathbf{H}$ to be a zero vector, whose path is named the homotopic path, and which can trace the zeros of $\mathbf{G}(\mathbf{x})$ to our desired solutions of $\mathbf{F}(\mathbf{x})=\mathbf{0}$ while the parameter τ reaches to 1. Among the various vector homotopy functions, the fixed point vector homotopy function, i.e. $\mathbf{G}(\mathbf{x})=\mathbf{x}-\mathbf{x}_0$, and the Newton vector homotopy function, i.e. $\mathbf{G}(\mathbf{x})=\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}_0)$, are simpler ones and are often used. The Newton vector homotopy function can be written as

$$\mathbf{H}(\mathbf{x},\tau)=\tau\mathbf{F}(\mathbf{x})+(1-\tau)[\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}_0)], \tag{2}$$

where $\mathbf{x}_0$ is a given initial value of $\mathbf{x}$ and $\tau\in[0,1]$. However, the resultant ODEs based-on the Newton vector homotopy function require an evaluation of the inverse of the Jacobian matrix $\partial\mathbf{F}/\partial\mathbf{x}$, which operation is computationally expensive.

To improve upon the slow convergence of the vector homotopy function method, Liu, Yeih, Kuo and Atluri (2009) have instead developed a scalar homotopy function method by converting the vector equation $\mathbf{F}=\mathbf{0}$ into a scalar equation $\|\mathbf{F}\|^2=0$ through

$$\mathbf{F}=\mathbf{0}\Leftrightarrow\|\mathbf{F}\|^2=0, \tag{3}$$

where $\|\mathbf{F}\|^2:=\sum_{i=1}^{n}F_i^2$, and defined a scalar homotopy function:

$$h(\mathbf{x},\tau)=\frac{\tau}{2}\|\mathbf{F}(\mathbf{x})\|^2+\frac{\tau-1}{2}\|\mathbf{x}-\mathbf{x}_0\|^2=0 \tag{4}$$

to derive the governing ODEs for $\dot{\mathbf{x}}$.

The scalar homotopy method retains the major advantage of the vector homotopy method as being globally convergent, and in addition, it does not involve the complicated computation of the inverse of the Jacobian matrix. The scalar homotopy method, however, requires a very small time step in order to keep $\mathbf{x}$ on the manifold (4), such that in order to reach $\tau=1$, the process of convergence is very slow. Later, Ku, Yeih and Liu (2010) modified Eq. (4) to the Newton scalar homotopy function by defining:

$$h(\mathbf{x},t)=\frac{Q(t)}{2}\|\mathbf{F}(\mathbf{x})\|^2-\frac{1}{2}\|\mathbf{F}(\mathbf{x}_0)\|^2=0, \tag{5}$$

where the function $Q(t) > 0$ satisfies $Q(0) = 1$, monotonically increasing, and $Q(\infty) = \infty$. Using this scalar homotopy function, Ku, Yeih and Liu (2010) could derive a faster convergent algorithm for solving nonlinear algebraic equations. As pointed out by Liu and Atluri (2011a), Eq. (5) indeed provides a *differentiable manifold*

$$Q(t)\|\mathbf{F}(\mathbf{x})\|^2 = C, \tag{6}$$

where $C = \|\mathbf{F}(\mathbf{x}_0)\|^2$ is a constant, to confine the solution path being retained on the manifold. *The idea behind this approach is that we can construct an ODEs system on the space-time manifold (6), and then upon getting a trace of the solution path solved from the ODEs on the manifold we can eventually obtain the solution of the nonlinear algebraic equations, because of*

$$\|\mathbf{F}\|^2 \to 0 \Leftrightarrow \mathbf{F} \to \mathbf{0}. \tag{7}$$

In the optimization theories, and in the computations involved there, some discussions about the Riemann manifold [Absil, Baker and Gallivan (2007); Adler, Dedieu, Margulies, Martens and Shub (2002); Baker, Absil and Gallivan (2008); Luenberger (1972); Smith (1994); Yang (2007)], were already presented. However, we provide herein quite a different approach for solving the nonlinear algebraic equations based on an *invariant-manifold*.

2 Evolution equation for $\dot{\mathbf{x}}$

2.1 *Evolution equation for* $\dot{\mathbf{x}}$ *in the direction* $\mathbf{R}$ *normal to* $h(\mathbf{x},t) = 0$*; and the operator* $\mathbf{D}$ *which generates a normal to* $\mathbf{R}$

For the system (1) of NAEs, we expect $h(\mathbf{x},t) = 0$ to be an invariant-manifold in the space-time domain $(\mathbf{x},t)$ for a dynamical system $h(\mathbf{x}(t),t) = 0$ to be specified further. When $Q > 0$, the manifold defined by Eq. (5) is continuous and differentiable, and thus the following differential operation makes sense.

Taking the time differential of Eq. (5) with respect to t and considering $\mathbf{x} = \mathbf{x}(t)$, we have

$$\frac{1}{2}\dot{Q}(t)\|\mathbf{F}(\mathbf{x})\|^2 + Q(t)\mathbf{R}\cdot\dot{\mathbf{x}} = 0, \tag{8}$$

where

$$\mathbf{R} := \mathbf{B}^{\mathrm{T}}\mathbf{F} \tag{9}$$

is a descent gradient vector[1] of the functional $\|\mathbf{F}\|^2/2$, and $\mathbf{B}$ is the Jacobian matrix with its ij-component given by $B_{ij} = \partial F_i/\partial x_j$.

The governing equation of $\dot{\mathbf{x}}$ cannot be uniquely determined by Eq. (8); however, we suppose that the following "normality" condition may be used:

$$\dot{\mathbf{x}} = -\lambda \frac{\partial h}{\partial \mathbf{x}} = -\lambda Q(t)\mathbf{R}, \tag{10}$$

where the preset multiplier λ is to be determined. Inserting Eq. (10) into Eq. (8) we can solve

$$\lambda = \frac{\dot{Q}(t)\|\mathbf{F}\|^2}{2Q^2(t)\|\mathbf{R}\|^2}. \tag{11}$$

Thus we obtain the evolution equation for $\mathbf{x}$:

$$\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{F}\|^2}{\|\mathbf{R}\|^2}\mathbf{R}, \tag{12}$$

where

$$q(t) := \frac{\dot{Q}(t)}{2Q(t)}. \tag{13}$$

Hence, we have a convergence property in solving the NAEs in Eq. (1):

$$\|\mathbf{F}(\mathbf{x})\|^2 = \frac{C}{Q(t)}. \tag{14}$$

When t is large enough the above equation will enforce the residual norm $\|\mathbf{F}(\mathbf{x})\|^2$ to tend to zero, and meanwhile the solution of Eq. (1) is obtained approximately. However, it is a great challenge to develop a suitable numerical integrator for solving Eq. (12), such that the orbit of $\mathbf{x}$ can really remain on the manifold $h(\mathbf{x},t) = 0$.

In order to keep $\mathbf{x}$ on the manifold in Eq. (14) we can consider the evolution of $\mathbf{F}$:

$$\dot{\mathbf{F}} = \mathbf{B}\dot{\mathbf{x}} = -q(t)\frac{\|\mathbf{F}\|^2}{\|\mathbf{B}^{\mathrm{T}}\mathbf{F}\|^2}\mathbf{A}\mathbf{F}, \tag{15}$$

where

$$\mathbf{A} := \mathbf{B}\mathbf{B}^{\mathrm{T}}. \tag{16}$$

[1] The vector $\mathbf{R}$ is normal to the hyper-surface $h(\mathbf{x},t) = 0$

Let

$$\mathbf{Y} = \sqrt{Q(t)}\mathbf{F}, \tag{17}$$

and from Eq. (14) we know that $\mathbf{Y} \in \mathbb{S}^{n-1}$ with a radius $\sqrt{C}$.

Now we derive the governing equation for $\dot{\mathbf{Y}}$. From Eq. (15) we can derive

$$\dot{\mathbf{Y}} = q(t)\left[\mathbf{I}_n - \frac{\|\mathbf{Y}\|^2}{\|\mathbf{B}^{\mathrm{T}}\mathbf{Y}\|^2}\mathbf{A}\right]\mathbf{Y}. \tag{18}$$

If we define the following operator:

$$\mathbf{D} = \mathbf{I}_n - \frac{\|\mathbf{Y}\|^2}{\|\mathbf{B}^{\mathrm{T}}\mathbf{Y}\|^2}\mathbf{A}, \tag{19}$$

we have a new dynamical system for $\mathbf{Y}$:

$$\dot{\mathbf{Y}} = q(t)\mathbf{D}\mathbf{Y}, \tag{20}$$

where $\mathbf{D}$ has the following properties:

$$\mathbf{D}^{\mathrm{T}} = \mathbf{D}, \;\; \mathbf{Y}^{\mathrm{T}}\mathbf{D}\mathbf{Y} = 0, \text{ or } \; \mathbf{Y}^{\mathrm{T}}\dot{\mathbf{Y}} = 0, \tag{21}$$

due to $\mathbf{A}^{\mathrm{T}} = \mathbf{A}$ and $\mathbf{Y}^{\mathrm{T}}\mathbf{A}\mathbf{Y} = \|\mathbf{B}^{\mathrm{T}}\mathbf{Y}\|^2$.

Usually, if the length of $\mathbf{Y}$ is preserved, the matrix $\mathbf{D}$ is skew-symmetric, but the present $\mathbf{D}$ is a negative-definite matrix. Indeed, it is a perpendicular operator, sending $\mathbf{Y}$ to a new vector which is perpendicular to $\mathbf{Y}$ itself. Hence, we have derived a nonlinear perpendicular operator

$$\mathbf{D} = \mathbf{I}_n - \frac{\|\mathbf{Y}\|^2}{\mathbf{Y}^{\mathrm{T}}\mathbf{A}\mathbf{Y}}\mathbf{A}, \tag{22}$$

where $\mathbf{A}$ is a positive-definite matrix, not necessarily a constant matrix. As an extension, any positive-definite matrix $\mathbf{A}$ not necessarily the $\mathbf{A}$ given by Eq. (16), can generate a nonlinear perpendicular operator $\mathbf{D}$ in Eq. (22).

2.2 A non-normal relation for the evolution

While employing Eq. (10), we have mentioned that the equation for $\dot{\mathbf{x}}$ is not unique. Indeed, we can add an extra term which is perpendicular to the normal $\mathbf{R}$ to the hyper-surface $h(\mathbf{x},t) = 0$,[2] and this does not influence the consistency condition

[2] Liu and Atluri (2011b) proposed $\mathbf{u} = \alpha\mathbf{F} + (1-\alpha)\mathbf{B}^{\mathrm{T}}\mathbf{F} = \alpha\mathbf{F} + (1-\alpha)\mathbf{R}$ with an optimal value for α. We show now the algorithms presented in the current paper are much better, and converge faster, than those in Liu and Atluri (2011b)

(8). Thus, we can construct a new evolution equation for $\dot{\mathbf{x}}$, as an extension of the residual-norm based algorithm [Liu and Atluri (2011a)]:

$$\dot{\mathbf{x}} = \lambda \mathbf{u}, \tag{23}$$

where

$$\mathbf{u} = \alpha \mathbf{R} + \beta \mathbf{P}, \tag{24}$$

in which $\mathbf{P}$ is a supplemental search vector being perpendicular to the descent vector $\mathbf{R}$,

$$\mathbf{P} = \mathbf{D}\mathbf{R} = \left[\mathbf{I}_n - \frac{\|\mathbf{R}\|^2}{\mathbf{R}^{\mathrm{T}}\mathbf{C}\mathbf{R}} \mathbf{C} \right] \mathbf{R}, \tag{25}$$

where

$$\mathbf{C} = \mathbf{B}^{\mathrm{T}}\mathbf{B}, \tag{26}$$

and the two parameters α and β are to be optimized, as will be in Section 3.3.

Inserting Eq. (23) into Eq. (8) we can derive

$$\dot{\mathbf{x}} = -q(t) \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}} \mathbf{u}, \tag{27}$$

where $q(t)$ is the same as Eq. (13), and

$$\mathbf{v} := \alpha \mathbf{v}_1 + \beta \mathbf{v}_2 = \mathbf{B}\mathbf{u} = \alpha \mathbf{B}\mathbf{R} + \beta \mathbf{B}\mathbf{P}. \tag{28}$$

We note that

$$\mathbf{F}^{\mathrm{T}}\mathbf{v} = \mathbf{R}^{\mathrm{T}}\mathbf{u} = \alpha \|\mathbf{R}\|^2 > 0, \tag{29}$$

due to $\mathbf{R}^{\mathrm{T}}\mathbf{P} = 0$ and $\alpha > 0$; see Section 3.3.

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 system (1) of NAEs:

$$\|\mathbf{F}(\mathbf{x})\|^2 = \frac{C}{Q(t)}, \tag{30}$$

where

$$C = \|\mathbf{F}(\mathbf{x}_0)\|^2 \tag{31}$$

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.[3] Hence, we impose in our algorithm that $Q(t) > 0$ is a monotonically increasing function of t. When t is increased to a large value, the above equation (30) will enforce the residual error $\|\mathbf{F}(\mathbf{x})\|$ to tend to zero, and meanwhile the solution of Eq. (1) is obtained approximately. However, it is still a challenge to develop a suitable numerical integrator to solve Eq. (27), such that the orbit of $\mathbf{x}$ can really remain on the invariant-manifold (30).

Notice that the dynamical system (27) is time-dependent and nonlinear, which is quite different from that of the so-called Dynamical Systems Method (DSM), which was previously developed by Ramm (2007, 2009), Hoang and Ramm (2008, 2010), and Sweilam, Nagy and Alnasr (2009) for solving linear equations, where the ODEs system is time-independent and linear. Here, we have to stress the importance of the concept of *invariant-manifold*; without the help from this concept we cannot derive a nonlinear ODEs system to depict the solution of $\mathbf{x}$. The effectiveness of the *invariant-manifold* will be explored further in the next section.

3 Dynamics on the invariant-manifold

3.1 Discretizing in time, yet keeping x *on the manifold*

Now we discretize the continuous time dynamics (27) into a discrete time dynamics by applying the forward Euler scheme:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - \eta \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{u}, \tag{32}$$

where

$$\eta = q(t)\Delta t \tag{33}$$

is a steplength. Correspondingly, $\mathbf{u}$ is a search direction weighted by a stepsize $\eta\|\mathbf{F}\|^2/\mathbf{F}\cdot\mathbf{v}$.

In order to keep $\mathbf{x}$ on the manifold (30), we can consider the evolution of $\mathbf{F}$ along the path $\mathbf{x}(t)$ by

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

[3] In Section 3.2 we can compute Q at each iteration step

Similarly, we use the forward Euler scheme to integrate Eq. (34):

$$\mathbf{F}(t+\Delta t) = \mathbf{F}(t) - \eta \frac{\|\mathbf{F}\|^2}{\mathbf{F}^{\mathrm{T}}\mathbf{v}}\mathbf{v}. \tag{35}$$

Taking the square-norms of both the sides of Eq. (35) and using Eq. (30) we can obtain

$$\frac{C}{Q(t+\Delta t)} = \frac{C}{Q(t)} - 2\eta \frac{C}{Q(t)} + \eta^2 \frac{C}{Q(t)} \frac{\|\mathbf{F}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2}\|\mathbf{v}\|^2. \tag{36}$$

Thus, after dividing by $C/Q(t)$ on both the sides, the following scalar equation is obtained:

$$a_0\eta^2 - 2\eta + 1 - \frac{Q(t)}{Q(t+\Delta t)} = 0, \tag{37}$$

where

$$a_0 := \frac{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2}. \tag{38}$$

As a result $h(\mathbf{x},t) = 0$, $t \in \{0,1,2,\ldots\}$ remains to be an invariant-manifold in the space-time domain $(\mathbf{x},t)$ for the discrete time dynamical system $h(\mathbf{x}(t),t) = 0$, which will be further explored in the following sections.

3.2 An iterative dynamics

Let

$$s = \frac{Q(t)}{Q(t+\Delta t)} = \frac{\|\mathbf{F}(t+\Delta t)\|^2}{\|\mathbf{F}(t)\|^2} \tag{39}$$

be a quantity in assessing the convergence property of our numerical algorithm for solving the system (1) of NAEs.

From Eqs. (37) and (39) it follows that

$$a_0\eta^2 - 2\eta + 1 - s = 0, \tag{40}$$

where

$$a_0 := \frac{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}{(\mathbf{F}^{\mathrm{T}}\mathbf{v})^2} \geq 1, \tag{41}$$

by using the Cauchy-Schwarz inequality:

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

From Eq. (40), we can take the solution of η to be

$$\eta = \frac{1-\sqrt{1-(1-s)a_0}}{a_0}, \text{ if } 1-(1-s)a_0 \ge 0. \tag{42}$$

Let

$$1-(1-s)a_0 = \gamma^2 \ge 0, \tag{43}$$

$$s = 1-\frac{1-\gamma^2}{a_0}, \tag{44}$$

such that the condition in Eq. (42) is automatically satisfied, and from Eq. (42) it follows that

$$\eta = \frac{1-\gamma}{a_0}. \tag{45}$$

From Eqs. (32), (38) and (45) we can obtain the following algorithm:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - (1-\gamma)\frac{\mathbf{F}^{\mathrm{T}}\mathbf{v}}{\|\mathbf{v}\|^2}\mathbf{u}, \tag{46}$$

where

$$0 \le \gamma < 1 \tag{47}$$

is a parameter.

When s is determined by Eq. (44), Q can be sequentially computed from

$$Q(t+\Delta t) = \frac{Q(t)}{s} = \frac{a_0 Q(t)}{a_0 - 1 + \gamma^2}, \tag{48}$$

by starting from $Q(0) = 1$.

Under conditions (41) and (47), and from Eqs. (39) and (44) we can prove that the new algorithm satisfies

$$\frac{\|\mathbf{F}(t+\Delta t)\|}{\|\mathbf{F}(t)\|} = \sqrt{s} < 1, \tag{49}$$

which means that the residual error is absolutely decreased. In other words, the convergence rate is always greater than 1:

$$\text{Convergence Rate} := \frac{\|\mathbf{F}(t)\|}{\|\mathbf{F}(t+\Delta t)\|} = \frac{1}{\sqrt{s}} > 1. \tag{50}$$

The property in Eq. (50) is meaningful, since it guarantees that the new algorithm is absolutely convergent to the true solution. *Smaller s implies Faster convergence*. Next we seek an optimal solution for the parameters α and β.

3.3 Optimal values of α and β

The algorithm (46) does not specify how to choose the parameters α and β, which are appeared in the vector $\mathbf{v}$. We can determine suitable α and β such that s defined by Eq. (44) is minimized with respect to α and β, because a smaller s will lead to a larger convergence rate as shown in Eq. (50).

Thus by inserting Eq. (38) for a_0 into Eq. (44) we can write s to be

$$s = 1 - \frac{(1-\gamma^2)(\mathbf{F}\cdot\mathbf{v})^2}{\|\mathbf{F}\|^2\|\mathbf{v}\|^2}, \tag{51}$$

where $\mathbf{v}$ as defined by Eq. (28) includes parameters α and β. By the minimization of

$$\min_{\alpha,\beta} s, \tag{52}$$

we let $\partial s/\partial\alpha = 0$ and $\partial s/\partial\beta = 0$, and derive

$$\mathbf{F}\cdot\mathbf{v}_1\|\mathbf{v}\|^2 - \mathbf{F}\cdot\mathbf{v}[\|\mathbf{v}_1\|^2\alpha + \mathbf{v}_1\cdot\mathbf{v}_2\beta] = 0, \tag{53}$$

$$\mathbf{F}\cdot\mathbf{v}_2\|\mathbf{v}\|^2 - \mathbf{F}\cdot\mathbf{v}[\|\mathbf{v}_2\|^2\beta + \mathbf{v}_1\cdot\mathbf{v}_2\alpha] = 0. \tag{54}$$

Basically, the above two simultaneous equations can be used to solve α and β; however, they are rather complex, and we cannot obtain the closed-form solutions of α and β from them at this moment.

By observation we can write Eqs. (53) and (54) into a matrix equation:

$$\begin{bmatrix} \mathbf{F}\cdot\mathbf{v}_1 & -[\|\mathbf{v}_1\|^2\alpha + \mathbf{v}_1\cdot\mathbf{v}_2\beta] \\ \mathbf{F}\cdot\mathbf{v}_2 & -[\|\mathbf{v}_2\|^2\beta + \mathbf{v}_1\cdot\mathbf{v}_2\alpha] \end{bmatrix} \begin{bmatrix} \|\mathbf{v}\|^2 \\ \mathbf{F}\cdot\mathbf{v} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{55}$$

In order to avoid the zeros of $\|\mathbf{v}\|^2$ and $\mathbf{F}\cdot\mathbf{v}$, the determinant of the above coefficient matrix must be zero, i.e.,

$$\begin{vmatrix} \mathbf{F}\cdot\mathbf{v}_1 & -[\|\mathbf{v}_1\|^2\alpha + \mathbf{v}_1\cdot\mathbf{v}_2\beta] \\ \mathbf{F}\cdot\mathbf{v}_2 & -[\|\mathbf{v}_2\|^2\beta + \mathbf{v}_1\cdot\mathbf{v}_2\alpha] \end{vmatrix} = 0, \tag{56}$$

which leads to a linear relation between β and α:

$$\beta = \omega\alpha, \tag{57}$$

where

$$\omega := \frac{\mathbf{v}_2 \cdot \mathbf{F}\|\mathbf{v}_1\|^2 - (\mathbf{v}_1 \cdot \mathbf{F})(\mathbf{v}_1 \cdot \mathbf{v}_2)}{\mathbf{v}_1 \cdot \mathbf{F}\|\mathbf{v}_2\|^2 - (\mathbf{v}_2 \cdot \mathbf{F})(\mathbf{v}_1 \cdot \mathbf{v}_2)}. \tag{58}$$

We find that by inserting Eq. (57) into Eqs. (53) and (54), they are automatically satisfied.

Remark 1: For the usual three-dimensional vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c} \in \mathbb{R}^3$, the following formula is famous:

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{a} \cdot \mathbf{c})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}. \tag{59}$$

Liu (2000) has developed a Jordan algebra by extending the above formula to vectors in n-dimension:

$$[\mathbf{a}, \mathbf{b}, \mathbf{c}] = (\mathbf{a} \cdot \mathbf{b})\mathbf{c} - (\mathbf{c} \cdot \mathbf{b})\mathbf{a}, \quad \mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n. \tag{60}$$

Thus the above coefficient ω can be expressed neatly in terms of the Jordan algebra:

$$\omega = \frac{[\mathbf{v}_1, \mathbf{F}, \mathbf{v}_2] \cdot \mathbf{v}_1}{[\mathbf{v}_2, \mathbf{F}, \mathbf{v}_1] \cdot \mathbf{v}_2}. \tag{61}$$

Usually, for $\mathbf{u}$ as shown by Eq. (24) we can impose the constraint that $\alpha + \beta = 1$ which is a coefficient before the vector $\mathbf{R}$. If $\alpha + \beta = 1$, from Eq. (57) we can see that

$$\alpha = \frac{1}{1+\omega}, \quad \beta = \frac{\omega}{1+\omega}. \tag{62}$$

Remark 2: We prove that

$$\alpha > 0, \quad \beta > 0. \tag{63}$$

By Eq. (28) we have

$$\mathbf{v}_2 \cdot \mathbf{F} = (\mathbf{BP}) \cdot \mathbf{F} = \mathbf{P}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{F} = \mathbf{P} \cdot \mathbf{R} = 0,$$

because $\mathbf{P}$ and $\mathbf{R}$ are perpendicular. Under this condition, from Eq. (58) we have

$$\omega := -\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_2\|^2} > 0, \tag{64}$$

where the inequality holds by Eqs. (28) and (25):

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = (\mathbf{BR}) \cdot (\mathbf{BP}) = \mathbf{R}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\mathbf{BDR} = \mathbf{R}^{\mathrm{T}}\mathbf{CDR} < 0,$$

because $\mathbf{C}$ is positive definite and $\mathbf{D}$ is negative definite. Thus, the statements in Eq. (63) follow from Eqs. (62) and (64).

Remark 3: Now we bestow the present numerical algorithm a geometrical meaning. From Eq. (41) we have

$$a_0 := \frac{1}{\cos^2\theta} \geq 1, \text{ because of } \mathbf{F} \cdot \mathbf{v} = \|\mathbf{F}\|\|\mathbf{v}\| \cos\theta > 0, \tag{65}$$

where θ is the intersection angle between $\mathbf{F}$ and $\mathbf{v}$. See the note in Eq. (29) for $\mathbf{F} \cdot \mathbf{v} > 0$.

From Eq. (46) it can be seen that the algorithm is scale-invariant, that is we can scale the search vector $\mathbf{u}$, but the algorithm is the same. Hence we can impose the following condition

$$\|\mathbf{v}\| = \|\mathbf{Bu}\| = 1, \tag{66}$$

by a suitable scaling of $\mathbf{u}$.

The minimization of s as carried out previously is equivalent to the minimization of a_0 due to Eq. (44):

$$\min_{\alpha,\beta} s \equiv \min_{\alpha,\beta} a_0. \tag{67}$$

This minimization is a search of the smallest angle θ between $\mathbf{F}$ and $\mathbf{v}$. Furthermore, by Eqs. (46), (41) and (66) we can derive

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) - (1-\gamma)\frac{\|\mathbf{F}\|}{\sqrt{a_0}\|\mathbf{v}\|}\mathbf{u} = \mathbf{x}(t) - (1-\gamma)\frac{\|\mathbf{F}\|}{\sqrt{a_0}}\mathbf{u}. \tag{68}$$

So the minimization of a_0 is further equivalent to the maximization of the stepsize:

$$\min_{\alpha,\beta} s \equiv \min_{\alpha,\beta} a_0 \equiv \max_{\alpha,\beta} \frac{\mathbf{F} \cdot \mathbf{v}}{\|\mathbf{v}\|^2}. \tag{69}$$

When an algorithm has a larger stepsize, it generally leads to a faster convergence. Eq. (68) also indicates that when a_0 tends to a large value, the algorithm will slow down.

3.4 An optimal iterative algorithm

Since the fictitious time variable is now discrete, $t \in \{0,1,2,\ldots\}$, we let $\mathbf{x}_k$ denote the numerical value of $\mathbf{x}$ at the k-th step. Thus, we arrive at a purely iterative algorithm by Eq. (46):

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma)\frac{\mathbf{F}_k^{\mathrm{T}}\mathbf{v}_k}{\|\mathbf{v}_k\|^2}\mathbf{u}_k. \tag{70}$$

As a consequence the following Optimal Iterative Algorithm with an Optimal Descent Vecotr (OIA/ODV) is available:

(i) Select γ in the range of $0 \le \gamma < 1$, give 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:

$$\begin{aligned}
&\mathbf{R}_k = \mathbf{B}_k^{\mathrm{T}}\mathbf{F}_k, && [\text{Normal to } h(\mathbf{x},t)=0]\\
&\mathbf{P}_k = \mathbf{R}_k - \frac{\|\mathbf{R}_k\|^2}{\mathbf{R}_k^{\mathrm{T}}\mathbf{C}_k\mathbf{R}_k}\mathbf{C}_k\mathbf{R}_k, && [\text{Normal to } \mathbf{R}_k]\\
&\mathbf{v}_1^k = \mathbf{B}_k\mathbf{R}_k,\\
&\mathbf{v}_2^k = \mathbf{B}_k\mathbf{P}_k,\\
&\alpha_k = \frac{1}{1+\omega_k},\\
&\beta_k = \frac{\omega_k}{1+\omega_k},\\
&\mathbf{u}_k = \alpha_k\mathbf{R}_k + \beta_k\mathbf{P}_k, && [\text{Optimal Descent Vector}]\\
&\mathbf{v}_k = \alpha_k\mathbf{v}_1^k + \beta_k\mathbf{v}_2^k,\\
&\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma)\frac{\mathbf{F}_k\cdot\mathbf{v}_k}{\|\mathbf{v}_k\|^2}\mathbf{u}_k.
\end{aligned} \tag{71}$$

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 above, ω_k is computed from Eq. (61) by inserting $\mathbf{F}_k$, $\mathbf{v}_1^k$ and $\mathbf{v}_2^k$.

In summary, we have derived a novel algorithm endowed with a Jordan structure in Eq. (61) for computing the coefficient ω. While the relaxation parameter γ is chosen by the user for the problem-dependence, the parameters α and β are exactly given by Eqs. (61) and (62).

Sometimes we can have another option to choose the optimal descent vector in solving nonlinear equations. Here, instead of $\mathbf{R}$ we use $\mathbf{F}$ as a primary descent vec-

tor, and then $\mathbf{P}$ is constructed from $\mathbf{P}=\mathbf{DF}$ by Eq. (25) with $\mathbf{R}$ replaced by $\mathbf{F}$. Thus, we have two options about the optimal descent vector. We call the above algorithm OIA/ODV[R][4] if $\mathbf{R}$ is used as a primary driving vector, and OIA/ODV[F] if $\mathbf{F}$ is used as a primary driving vector. So the OIA/ODV[F] reads as:

(i) Select γ in the range of $0 \le \gamma < 1$, give 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:

$$\begin{aligned}
&\mathbf{P}_k^* = \mathbf{F}_k - \frac{\|\mathbf{F}_k\|^2}{\mathbf{F}_k^{\mathrm{T}}\mathbf{C}_k\mathbf{F}_k}\mathbf{C}_k\mathbf{F}_k, \quad \text{[Normal to } \mathbf{F}_k\text{]}\\
&\mathbf{v}_1^k = \mathbf{B}_k\mathbf{F}_k,\\
&\mathbf{v}_2^k = \mathbf{B}_k\mathbf{P}_k^*,\\
&\alpha_k = \frac{1}{1+\omega_k},\\
&\beta_k = \frac{\omega_k}{1+\omega_k},\\
&\mathbf{u}_k = \alpha_k\mathbf{F}_k + \beta_k\mathbf{P}_k^*, \qquad \text{[Optimal Descent Vector]}\\
&\mathbf{v}_k = \alpha_k\mathbf{v}_1^k + \beta_k\mathbf{v}_2^k,\\
&\mathbf{x}_{k+1} = \mathbf{x}_k - (1-\gamma)\frac{\mathbf{F}_k\cdot\mathbf{v}_k}{\|\mathbf{v}_k\|^2}\mathbf{u}_k.
\end{aligned} \tag{72}$$

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 above, ω_k is computed from Eq. (61) by inserting $\mathbf{F}_k$, $\mathbf{v}_1^k$ and $\mathbf{v}_2^k$.

4 Numerical examples

In this section we apply the new methods of OIA/ODV[F] and OIA/ODV[R] to some nonlinear ODEs and PDEs.

[4] OIA [Optimal Iterative Algorithm]; ODV [Optimal Descent Vector]

4.1 Example 1

In this example we apply the new algorithm OIA/ODV[F] to solve the following nonlinear boundary value problem:

$$u'' = \frac{3}{2}u^2, \tag{73}$$

$$u(0) = 4, \;\; u(1) = 1. \tag{74}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{75}$$

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

$$F_i = \frac{1}{(\Delta x)^2}(u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2 = 0, \tag{76}$$

$$u_0 = 4, \;\; u_{n+1} = 1, \tag{77}$$

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

We fix $n = 9$ and $\varepsilon = 10^{-5}$. The parameter γ used in OIA/ODV[F] is 0.11. In Fig. 1 we compare the residual errors obtained by OIA/ODV[F] and that obtained by the numerical method of Liu and Atluri (2011b). When the former converges with 33 iterations, the latter converges with 28 iterations.

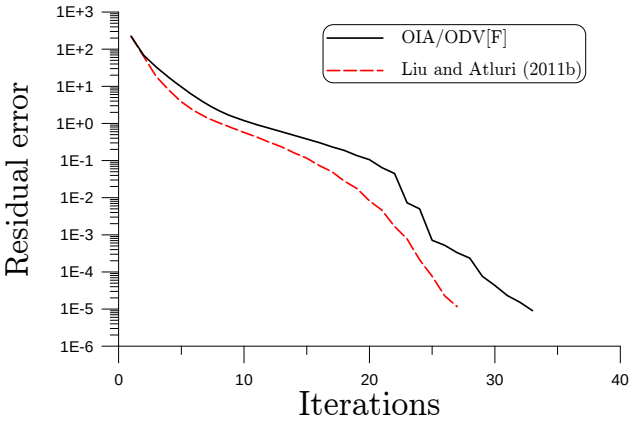

Figure 1: For example 1 comparing the residual erros for the numerical methods of OIA/ODV[F] and Liu and Atluri (2011b).

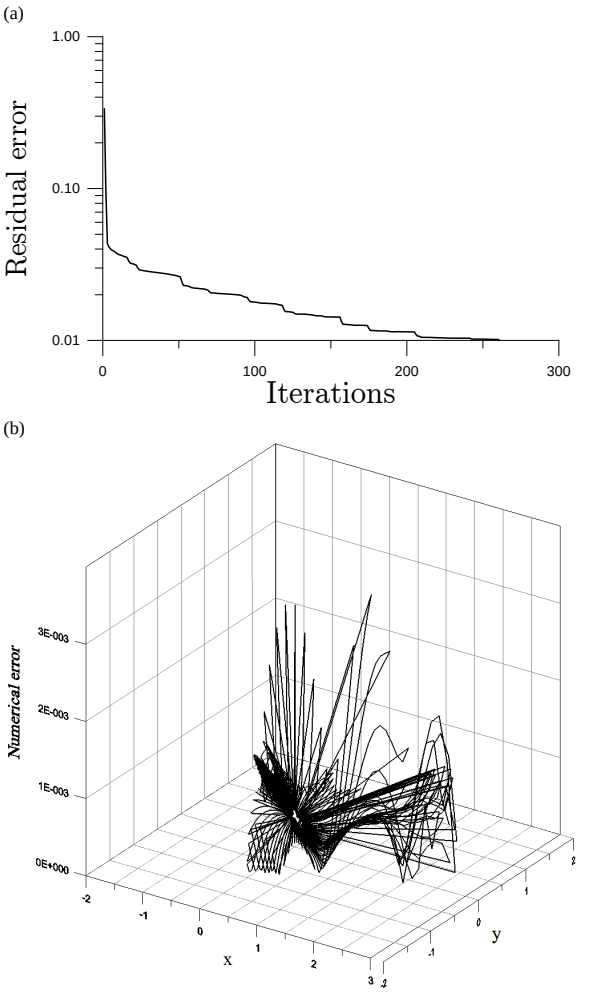

Figure 2: For example 2 solved by OIA/ODV[R]: (a) the residual error, and (b) the numerical error.

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

$$u(x,y) = \sum_{k=1}^{n} a_k \phi_k, \tag{78}$$

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

$$\begin{aligned}
&\phi_k = (r_k^2 + c^2)^{N-3/2},\ N = 1,2,\ldots,\\
&\phi_k = r_k^{2N} \ln r_k,\ N = 1,2,\ldots,\\
&\phi_k = \exp\left(-\frac{r_k^2}{a^2}\right),\\
&\phi_k = (r_k^2 + c^2)^{N-3/2} \exp\left(-\frac{r_k^2}{a^2}\right),\ N = 1,2,\ldots,
\end{aligned} \tag{79}$$

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 as trial functions, which is known as a multi-quadric RBF [Golberg, Chen and Karur (1996); Cheng, Golberg, Kansa and Zammito (2003)], with $N = 2$.

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

$$\Delta u = 4u^3(x^2 + y^2 + a^2), \tag{80}$$

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

$$\rho(\theta) = (\sin 2\theta)^2 \exp(\sin\theta) + (\cos 2\theta)^2 \exp(\cos\theta). \tag{81}$$

The exact solution is given by

$$u(x,y) = \frac{-1}{x^2 + y^2 - a^2}, \tag{82}$$

which is singular on the circle with a radius a.

Inserting Eq. (78) into Eq. (80) 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 error obtained by the OIA/ODV[R], which is convergent with 261 iterations. It can be seen that the residual-error curve decays very fast at the first few steps. The absolute error of numerical solution is plotted in Fig. 2(b), which is quite accurate with the maximum error being 3.91×10^{-3}.

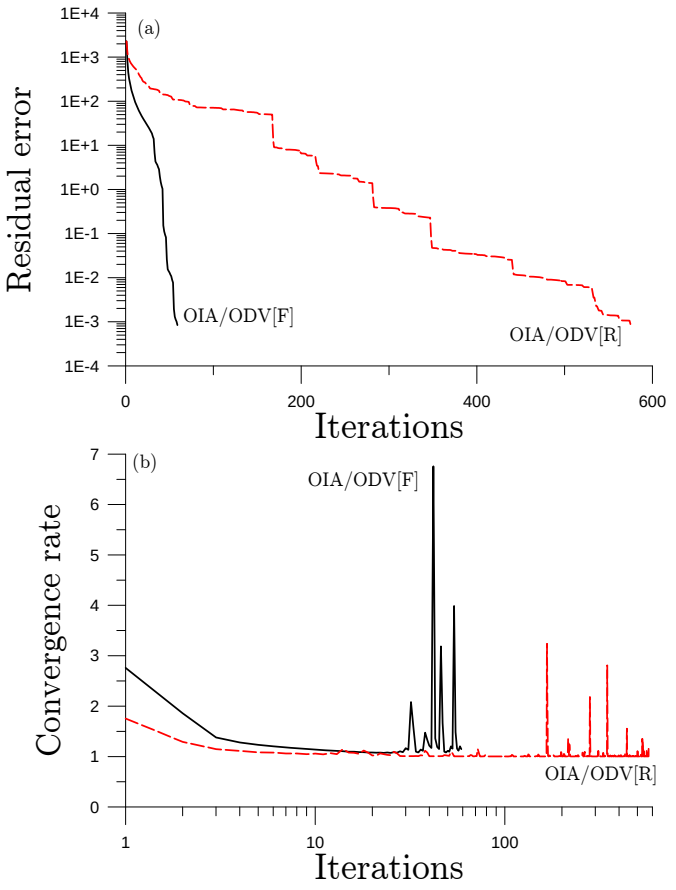

Figure 3: For example 3 solved by OIA/ODV[F] and OIA/ODV[R] comparing (a) the residual errors, and (b) the convergence rates.

4.3 Example 3

In this example we apply the OIA/ODV[F] and OIA/ODV[R] to solve the following boundary value problem of nonlinear elliptic equation

$$\Delta u(x,y)+\omega^2 u(x,y)+\varepsilon_1 u^3(x,y)=p(x,y). \tag{83}$$

While the exact solution is

$$u(x,y)=\frac{-5}{6}(x^3+y^3)+3(x^2y+xy^2), \tag{84}$$

the exact p can be obtained by inserting the above u into Eq. (83).

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

$$\begin{aligned} F_{i,j} &= \frac{1}{(\Delta x)^2}(u_{i+1,j}-2u_{i,j}+u_{i-1,j})+\frac{1}{(\Delta y)^2}(u_{i,j+1}-2u_{i,j}+u_{i,j-1}) \\ &+ \omega^2 u_{i,j}+\varepsilon_1 u_{i,j}^3-p_{i,j}=0. \end{aligned} \tag{85}$$

The boundary conditions can be obtained from the exact solution in Eq. (84).

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 OIA/ODV[F] and OIA/ODV[R]. In Fig. 3(a) we show the residual errors, of which the OIA/ODV[F] converges with only 59 iterations, but the OIA/ODV[R] requires 575 iterations. The reason for the faster convergence of OIA/ODV[F] than OIA/ODV[R] is shown in Fig. 3(b), where the convergence rate of OIA/ODV[R] is much smaller than that of OIA/ODV[F]. The maximum numerical error is about 5.7×10^{-6} for OIA/ODV[F] and is about 5.9×10^{-6} for OIA/ODV[R]. Very accurate numerical results were obtained by the present OIA/ODV algorithms.

4.4 Example 4

We consider a nonlinear heat conduction equation:

$$u_t=\alpha(x)u_{xx}+\alpha'(x)u_x+u^2+h(x,t), \tag{86}$$

$$\alpha(x)=(x-3)^2,\ \ h(x,t)=-7(x-3)^2e^{-t}-(x-3)^4e^{-2t}, \tag{87}$$

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

By applying the OIA/ODV[F] 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. 4(a) we show the residual errors, which can be seen convergence very fast with 114 iterations. The absolute errors of numerical solution are plotted in Fig. 4(b), which reveal accurate numerical result with the maximum error being 3.3×10^{-3}.

4.5 Example 5

We consider a nonlinear backward heat conduction equation:

$$u_t=k(x)u_{xx}+k'(x)u_x+u^2+H(x,t), \tag{88}$$

$$k(x)=(x-3)^2,\ \ H(x,t)=-7(x-3)^2e^{-t}-(x-3)^4e^{-2t}, \tag{89}$$

with a closed-form solution being $u(x,t)=(x-3)^2e^{-t}$. The boundary conditions and a final time condition are available from the above solution. It is known that

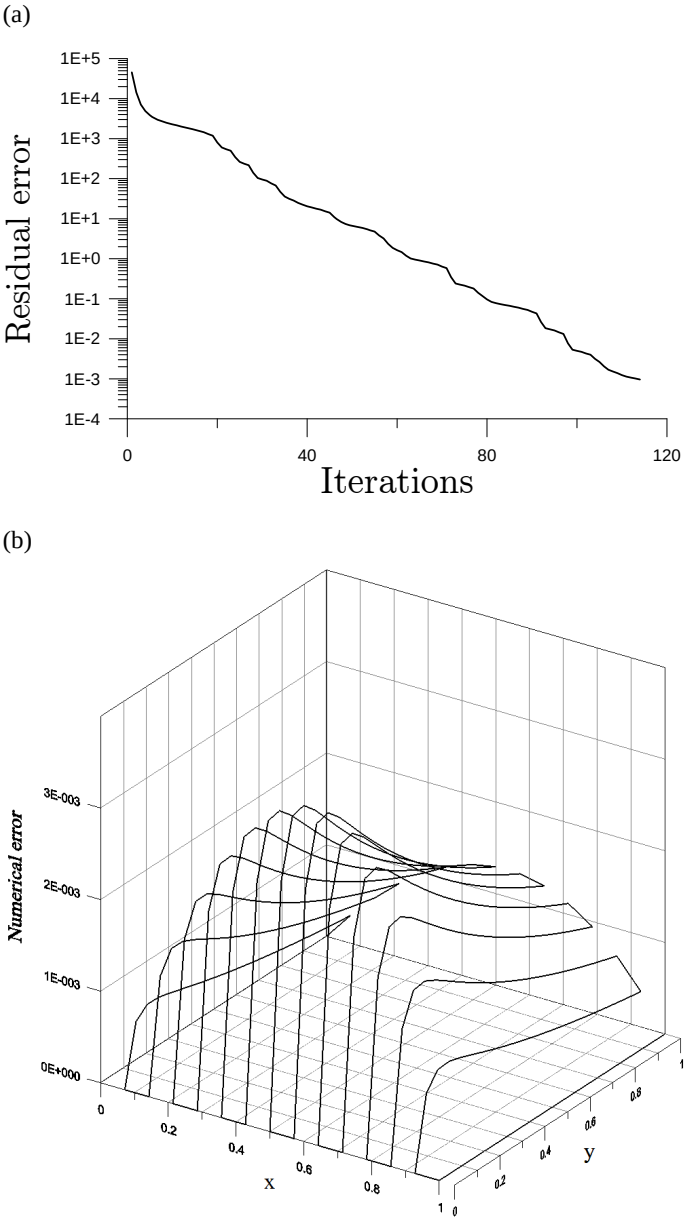

Figure 4: For example 4 solved by OIA/ODV[F]: (a) the residual error, and (b) the numerical error.

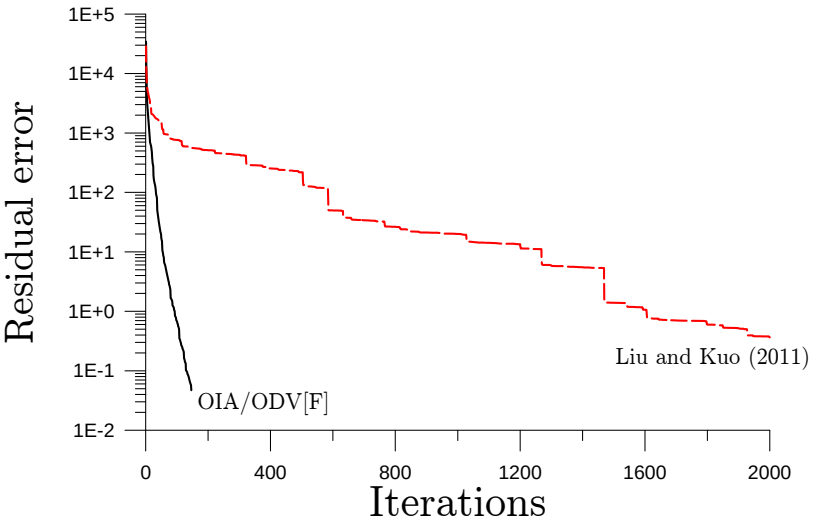

Figure 5: For example 5 comparing the residual errors by the numerical methods of OIA/ODV[F] and Liu and Kuo (2011).

the nonlinear backward heat conduction problem is highly ill-posed [Liu (2011)]. In order to test the stability of OIA/ODV we also add a relative noise in the final time data with an intensity $\sigma = 0.01$.

By applying the new algorithm of OIA/ODV[F] 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 Eq. (88):

$$k(x_i)\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2}+k'(x_i)\frac{u_{i+1,j}-u_{i-1,j}}{2\Delta x}+u_{i,j}^2+H(x_i,t_j)-\frac{u_{i,j+1}-u_{i,j}}{\Delta t}=0. \tag{90}$$

We compare the residual errors obtained by OIA/ODV[F] and that by the method of Liu and Kuo (2011) in Fig. 5. Obviously, the OIA/ODV[F] is convergent much fast than that of Liu and Kuo (2011). The OIA/ODV[F] can attain accurate solution with a maximum relative error being 1.64×10^{-3}, which is better than that computed by Liu and Atluri (2011b) and Liu and Kuo (2011).

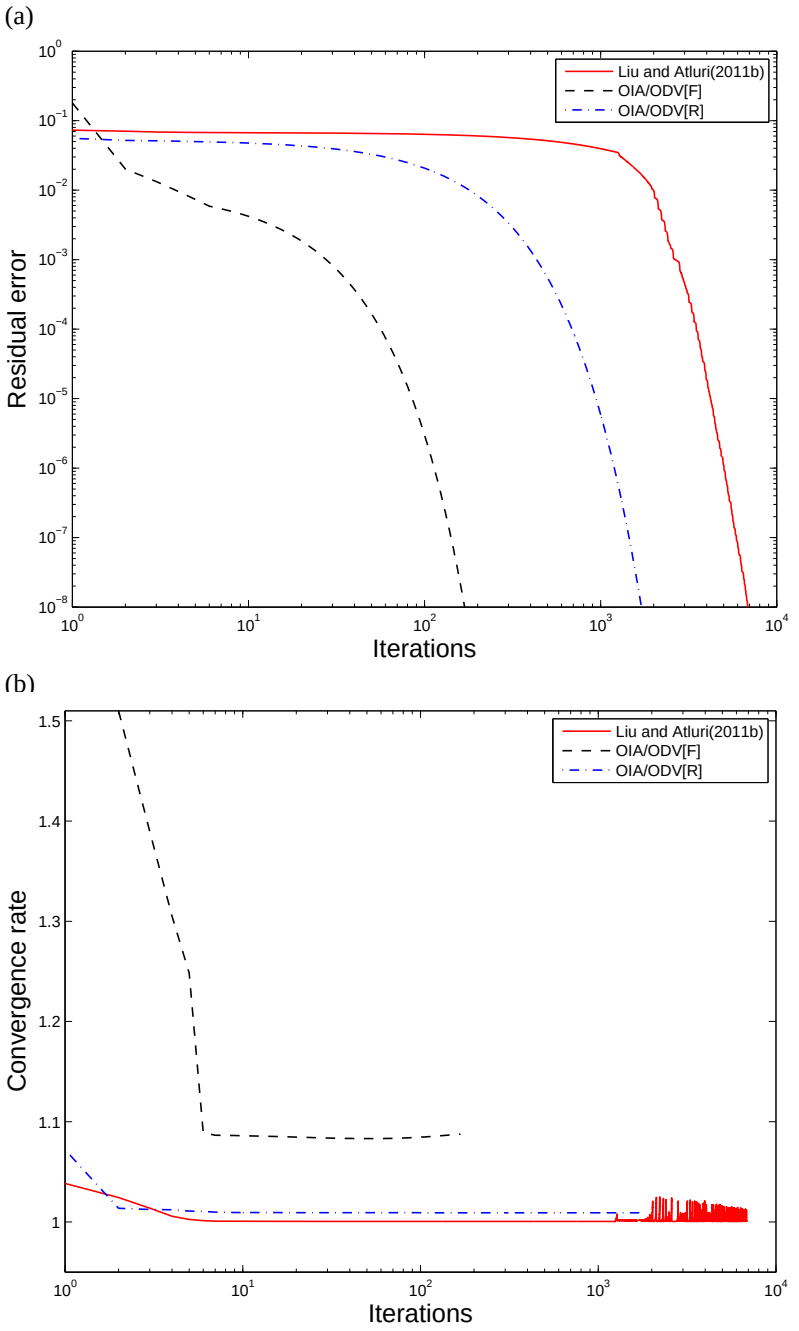

Figure 6: For example 6 solved by OIA/ODV[F], OIA/ODV[R] and OVDA [Liu and Atluri (2011b)] comparing (a) the residual errors and (b) the convergence rates.

4.6 Example 6

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

$$\ddot{x}+2\xi\dot{x}+x+x^3=F\sin\omega t, \tag{91}$$

where x is a non-dimensional displacement, ξ is a damping ratio, F is the amplitude of external force, and ω is the 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:

$$x(t)=x_0+\sum_{n=1}^{N}[x_{2n-1}\cos(n\omega t)+x_{2n}\sin(n\omega t)], \tag{92}$$

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. Differentiating $x(t)$ with respect to t, leads to

$$\dot{x}(t)=\sum_{n=1}^{N}[-n\omega x_{2n-1}\sin(n\omega t)+n\omega x_{2n}\cos(n\omega t)], \tag{93}$$

$$\ddot{x}(t)=\sum_{n=1}^{N}[-(n\omega)^2 x_{2n-1}\cos(n\omega t)-(n\omega)^2 x_{2n}\sin(n\omega t)]. \tag{94}$$

The nonlinear term in Eq. (91) can also be expressed in terms of the truncated Fourier series with N harmonics kept:

$$x^3(t)=r_0+\sum_{n=1}^{N}[r_{2n-1}\cos(n\omega t)+r_{2n}\sin(n\omega t)]. \tag{95}$$

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 formulas:

$$r_0=\frac{1}{2\pi}\int_0^{2\pi}\{x_0+\sum_{n=1}^{N}[x_{2n-1}\cos(n\theta)+x_{2n}\sin(n\theta)]\}^3 d\theta, \tag{96}$$

$$r_{2n-1}=\frac{1}{\pi}\int_0^{2\pi}\{x_0+\sum_{n=1}^{N}[x_{2n-1}\cos(n\theta)+x_{2n}\sin(n\theta)]\}^3\cos(n\theta)d\theta, \tag{97}$$

$$r_{2n}=\frac{1}{\pi}\int_0^{2\pi}\{x_0+\sum_{n=1}^{N}[x_{2n-1}\cos(n\theta)+x_{2n}\sin(n\theta)]\}^3\sin(n\theta)d\theta. \tag{98}$$

Note that because of the orthogonality of trigonometric functions, $r_n,\ n=0,1,\ldots,2N$ can be evaluated without numerical/analytical integration. Next, we substitute Eqs. (92)-(95) into Eq. (91), and collect 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:

$$(\mathbf{A}^2+2\xi\mathbf{A}+\mathbf{I}_{2N+1})\mathbf{Q}_x+\mathbf{R}_x=F\mathbf{H}, \tag{99}$$

where

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

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{J}_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{J}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{J}_N \end{bmatrix},\ \mathbf{J}_n=n\begin{bmatrix} 0 & \omega \\ -\omega & 0 \end{bmatrix}. \tag{100}$$

One should note that $r_n,\ n=0,1,\ldots,2N$ are analytically expressed by the coefficient variables, which makes the HB sometimes tedious for application. In order to deal with this problem easily, a time domain harmonic balance method (TDHB), sometime named HDHB, is proposed by Liu, Dowell and Hall (2006). In their study, the $2N+1$ coefficient variables x_n are recast into the time domain 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. However, the TDHB will cause spurious solutions, and the result obtained from TDHB is as accurate as that obtained from HB but requires twice the number of harmonic terms in analysis. Considering the above shortcomings, the TDHB is not a much better choice. In the present example, we can solve the Duffing equation by employing the standard HB method easily with the help of Mathematica. In doing so, one has no difficulty in handling the symbolic operations involved in evaluating r_n, and even a large number of harmonics can be easily solved, since the nonlinear algebraic equations (NAEs), i.e. Eq. (99), can be easily solved by the algorithm presented in this paper even when the order of the vector $\mathbf{Q}[2N+1]$ is very large [very large N]. It should

also be noted that the present algorithm for solving the NAEs is considerably simpler, better and faster than the well-known Newton or Newton-Raphson methods. In the current case, we apply the HB method with 8 harmonics to solve the Duffing equation (91), and then arrive at a system of 17 NAEs in Eq. (99). The NAEs are here solved by OIA/ODV[F], OIA/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 OIA/ODV[F], OIA/ODV[R] and OVDA [Liu and Atluri (2011b)] in Fig. 6(a). It shows that the OIA/ODV algorithms converge much faster than OVDA. Specifically, the iterations of OIA/ODV[F], OIA/ODV[R] and OVDA algorithms are 169, 1705 and 6871 respectively. The convergence ratio between OIA/ODV[F] and OVDA is as high as 40.7, which indicates a high efficiency of the present OIA/ODV[F] algorithm. Furthermore, the convergence rates for these methods are also provided in Fig. 6(b), from which we can see that the convergence rate of OIA/ODV[F] surpasses that of OIA/ODV[R], and is much superior to OVDA. In this case, the peak amplitude is $A = 0.43355$ which is a physically meaningful solution. We may come to a conclusion that although the TDHB is implemented without analytical operations of NAEs, it may cause non-physical solutions and lower accuracy. The HB method can efficiently, simply and accurately solve problems with complex nonlinearity with the help of symbolic operation software, i.e. Mathematica, Maple, and the present algorithms for solving NAEs. The physical nature of the solution for the Duffing's equation, when very higher-order harmonics are used in the Harmonic Balance Method (or the Galerkin method in the time variable) will be discussed elsewhere in a forth coming paper by the authors.

5 Conclusions

In the present paper, we have derived a purely iterative algorithm including a preset parameter γ $(0 \leq \gamma < 1)$:

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

where

$$\mathbf{u}_k = \alpha_k \mathbf{R}_k + \beta_k \mathbf{P}_k, \;\; \mathbf{v}_k = \alpha_k \mathbf{B}\mathbf{R}_k + \beta_k \mathbf{B}\mathbf{P}_k,$$

is a stepwise optimal search direction with $\mathbf{R}_k$ and $\mathbf{P}_k$ being two perpendicular vectors. This paper has revealed a methodology how the stepsize $\mathbf{F}_k \cdot \mathbf{v}_k / \|\mathbf{v}_k\|^2$ can

be maximized to accelerate the convergence speed in the numerical solution of NAEs and nonlinear ill-posed systems. These two algorithms were named the *Optimal Iterative Algorithm with Optimal Descent Vector (ODV)* with OIA/ODV[R] and OIA/ODV[F], which *have a better computational efficiency and accuracy than other algorithms* in solving either well-posed or ill-posed nonlinear algebraic equations. We also applied these algorithms to solve the Duffing equation by using a harmonic balance method. The computational efficiency was excellent to treat a highly nonlinear Duffing oscillator, even when a large number of harmonics were used in the Harmonic Balance Method.

Acknowledgement: This research at UCI was supported through a Collaborative Research Agreement by the US Army Research Labs 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

Absil, P.-A.; Baker, C. G.; Gallivan, K. A. (2007): Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, vol. 7, pp. 303-330.

Adler, R. L.; Dedieu, J.-P.; Margulies, J. Y.; Martens, M.; Shub, M. (2002): Newton's method on Riemannian manifolds and a geometric model for the human spine. *IMA J. Numer. Anal.*, vol. 22, pp. 359-390.

Baker, C. G.; Absil, P.-A.; Gallivan, K. A. (2008): Implicit trust-region methods on Riemannian manifolds. *IMA J. Numer. Anal.*, vol. 28, pp. 665-689.

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.

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.

Hoang, N. S.; Ramm, A. G. (2008): Solving ill-conditioned linear algebraic systems by the dynamical systems method (DSM). *J. Inverse Prob. Sci. Eng.*, vol. 16, pp. 617-630.

Hoang, N. S.; Ramm, A. G. (2010): Dynamical systems gradient method for solving ill-conditioned linear algebraic systems. *Acta Appl. Math.*, vol. 111, pp.

189-204.

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.

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. (2011): The method of fundamental solutions for solving the backward heat conduction problem with conditioning by a new post-conditioner. *Num. Heat Transfer, B: Fundamentals*, vol. 60, pp. 57-72.

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.; 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.; 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.

Liu, L.; Thomas, J. P.; Dowell, E. H.; Attar, P.; Hall, K. C. (2006): A comparison of classical and high dimension harmonic balance approaches for a Duffing oscillator. *J. Comput. Phys.*, vol. 215, pp. 298-320.

Luenberger, D. G. (1972): The gradient projection method along geodesics. *Management Sci.*, vol. 18, pp. 620-631.

Ramm, A. G. (2007): Dynamical System Methods for Solving Operator Equations. Elsevier, Amsterdam, Netherlands.

Ramm, A. G. (2009): Dynamical systems method for solving linear ill-posed problems. *Ann. Pol. Math.*, vol. 95, pp. 253-272.

Smith, S. T. (1994): Optimization techniques on Riemannian manifolds: Hamiltonian and gradient flows, algorithms and control. *Fields Inst. Commun.*, vol. 3, pp. 113-136.

Sweilam, N. H.; Nagy, A. M.; Alnasr, M. H. (2009): An efficient dynamical systems method for solving singularly perturbed integral equations with noise. *Comp.*

Math. Appl., vol. 58, pp. 1418-1424.

Yang, Y. (2007): Globally convergent optimization algorithms on Riemannian manifolds: Uniform framework for unconstrained and constrained optimization. *J. Optim. Theory Appl.*, vol. 132, pp. 245-265.