



**ARTICLE**

# Nonlinear Algebraic Equations Solved by an Optimal Splitting-Linearizing Iterative Method

Chein-Shan Liu<sup>1</sup>, Essam R. El-Zahar<sup>2,3</sup> and Yung-Wei Chen<sup>4,\*</sup>

<sup>1</sup>Center of Excellence for Ocean Engineering, Center of Excellence for the Oceans, National Taiwan Ocean University, Keelung, 202301, Taiwan

<sup>2</sup>Department of Mathematics, College of Sciences and Humanities in Al-Kharj, Prince Sattam bin Abdulaziz University, Alkharj, 11942, Saudi Arabia

<sup>3</sup>Department of Basic Engineering Science, Faculty of Engineering, Menofia University, Shebin El-Kom, 32511, Egypt

<sup>4</sup>Department of Marine Engineering, National Taiwan Ocean University, Keelung, 202301, Taiwan

\*Corresponding Author: Yung-Wei Chen. Email: cyw0710@mail.ntou.edu.tw

Received: 26 January 2022 Accepted: 31 May 2022

## ABSTRACT

How to accelerate the convergence speed and avoid computing the inversion of a Jacobian matrix is important in the solution of nonlinear algebraic equations (NAEs). This paper develops an approach with a splitting-linearizing technique based on the nonlinear term to reduce the effect of the nonlinear terms. We decompose the nonlinear terms in the NAEs through a splitting parameter and then linearize the NAEs around the values at the previous step to a linear system. Through the maximal orthogonal projection concept, to minimize a merit function within a selected interval of splitting parameters, the optimal parameters can be quickly determined. In each step, a linear system is solved by the Gaussian elimination method, and the whole iteration procedure is convergent very fast. Several numerical tests show the high performance of the optimal split-linearization iterative method (OSLIM).

## KEYWORDS

Nonlinear algebraic equations; novel splitting-linearizing technique; iterative method; maximal projection; optimal splitting parameter

## 1 Introduction

Nonlinear partial differential equations (PDEs) frequently appear in many critical natural sciences and engineering technology problems. Mathematical models in physics, mechanics, and other disciplines fields can be applied to describe physical phenomena. Although some phenomena may be modeled by the linear type PDEs, the assumption of more complex conditions on the material property and geometric domain of the problem requires the use of nonlinear type PDEs. In most situations, solving nonlinear PDEs analytically and exactly is impossible. Because of the crucial role of the nonlinear problems, the researchers provided different numerical methods to obtain the solutions of nonlinear PDEs, which are properly transformed into the nonlinear algebraic equations (NAEs) by using numerically discretized methods. As mentioned in [1–4], nonlinear optimization problem,



nonlinear programming, nonlinear sloshing problem, and nonlinear complementarity problem (NCP) can be recast to the issues to solve NAEs, with the help of NCP-functions, for example, the Fischer-Burmeister NCP [5]. Indeed, the numerical solution of NAEs is one of the main problems in computational mathematics. Among the many numerical methods, Newton's method and its improvements have now been widely developed; however, if the initial guess of the solution is incorrect, these algorithms may fail. In general, it is difficult for most NAEs to choose a good initial condition. Therefore, it is necessary to develop a more effective algorithm that is less sensitive to the initial guess of the solution and converges quickly.

For the vector form of the NAEs:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \mathbf{x} := (x_1, \dots, x_n)^T \in \mathbb{R}^n, \mathbf{F} := (F_1, \dots, F_n)^T \in \mathbb{R}^n, \quad (1)$$

the Newton method is given by

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

where  $\mathbf{J}$  is an  $n \times n$  Jacobian matrix with its  $ij$ -th component given by  $J_{ij} = \partial F_i / \partial x_j$ . Starting from an initial guess  $\mathbf{x}_0$ , Eq. (2) with the computed  $\mathbf{J}^{-1}(\mathbf{x}_k)$  at each iteration can be used to generate a sequence of  $\mathbf{x}_k$ ,  $k = 1, 2, \dots$ . When  $\mathbf{x}_k$  is convergent under a specified convergence criterion, the solution of Eq. (1) is obtained.

The Newton method possesses a significant advantage in that it is quadratically convergent. However, the main drawbacks are sensitive to the initial point  $\mathbf{x}_0$  and need to compute  $\mathbf{J}^{-1}(\mathbf{x}_k)$  at each iteration step. Therefore, some quasi-Newton methods have been developed to overcome these shortcomings of the Newton method. For example, in order to eliminate the requirement of finding and inverting the Jacobian matrix in Newton's iterative algorithm, Liu et al. [5] proposed a derivative-free first-order system of nonlinear ordinary differential equations (ODE):

$$\dot{\mathbf{x}} = -\frac{\nu}{q(t)} \mathbf{F}(\mathbf{x}), \quad (3)$$

where  $\nu$  is a nonzero constant and  $q(t)$  may generally be a monotonically increasing function of  $t$ . The term  $\nu/q(t)$  plays a vital role in stabilizing the controller in their approach. Even if the initial guess of the solution is not proper, it can help obtain a stable solution and accelerate the convergent speed. Liu et al. [2,6,7] showed that the efficiency and performance of the fictitious time integration method (FTIM) combined with the group preserving scheme [8] to integrate the resultant ODEs in Eq. (3). Atluri et al. [9] proposed an improvement of Newton's method, which combined two methods of Newton's method and FTIM, without the inversion of  $J_{ij}$ . Related to the FTIM, some papers are employed to solve nonlinear problems [10,11]. To give a motivation for the present study, we begin with a system of linear algebraic equations (LAEs):

$$\mathbf{Ax} = \mathbf{b}, \quad (4)$$

where  $\mathbf{A}$  and  $\mathbf{b} \in \mathbb{R}^n$  are respectively a non-singular matrix and an input vector.

Let  $\mathbf{D} = \text{diag } \mathbf{A}$  and suppose that  $\mathbf{D}$  is non-singular. Setting

$$\mathbf{C} = \mathbf{D} - \mathbf{A} (\mathbf{A} = \mathbf{D} - \mathbf{C}), \quad (5)$$

the Jacobi iterative scheme for solving Eq. (4) is given by [12,13]

$$\mathbf{D}\mathbf{x}_{k+1} = \mathbf{C}\mathbf{x}_k + \mathbf{b}, \quad (6)$$

where  $\mathbf{x}_k$  denotes the  $k$ th step unknown variables  $\mathbf{x} \in \mathbb{R}^n$ . The splitting of  $\mathbf{A}$  to  $\mathbf{A} = \mathbf{D} - \mathbf{C}$  used in Eq. (6) was proposed by Jacobi to solve the linear algebraic equations (LAEs) iteratively. In general,  $\mathbf{A}$  can be uniquely decomposed into

$$\mathbf{A} = \mathbf{D} - \mathbf{U} - \mathbf{L}, \quad (7)$$

where  $\mathbf{U}$  and  $\mathbf{L}$  are a strict upper- and lower triangular matrix of  $\mathbf{A}$ , respectively.

According to Eqs. (4) and (7), the SOR iterative method can be derived as follows [14,15]:

$$(\mathbf{D} - w\mathbf{L}) \mathbf{x}_{k+1} = w\mathbf{b} + (1 - w) \mathbf{D}\mathbf{x}_k + w\mathbf{U}\mathbf{x}_k, \quad (8)$$

where  $w$  is a nonzero relaxation factor. When  $w = 1$ , it is known as the Gauss-Seidel iterative method. If  $D_{ii} \neq 0$ ,  $i = 1, 2, \dots, n$ , from Eq. (8),  $\mathbf{x}_{k+1}$  can be easily found by the forward substitution method. The main reason is  $\mathbf{D} - w\mathbf{L}$  a lower triangular matrix.

Nowadays, there have different splitting techniques to decompose  $\mathbf{A}$  as

$$\mathbf{A} = \mathbf{M} - \mathbf{N}, \quad (9)$$

where  $\mathbf{M}$  is a non-singular matrix. Inserting Eq. (9) into Eq. (4), moving  $\mathbf{N}\mathbf{x}$  to the right-hand side, taking the value of  $\mathbf{x}$  to be  $\mathbf{x}_k$  and on the left-hand side taking the value of  $\mathbf{x}$  to be  $\mathbf{x}_{k+1}$ . Then, the resulting iterative scheme expresses as follows:

$$\mathbf{M}\mathbf{x}_{k+1} = \mathbf{N}\mathbf{x}_k + \mathbf{b}, \quad (10)$$

$\mathbf{M}^{-1}\mathbf{N}$  is called an iteration matrix. The sufficient and necessary condition of the convergence for Eq. (10) is  $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$ , where  $\rho$  is the spectral radius, defined as the absolute value of the largest eigenvalue in the magnitude of  $\mathbf{M}^{-1}\mathbf{N}$ . When  $\rho(\mathbf{M}^{-1}\mathbf{N})$  gets smaller and smaller, the convergence speed is accelerated.

For the LAE, there are many discussions about the split method to determine the best relaxation factor  $w$  in SOR [16–20]. In contrast, few results discuss the relationship between the split method and NAEs. In this paper, the split method based on the nonlinear term is applied to reduce the influence of nonlinear terms and further avoid selecting the parameters by a trail and error method. Hence, this article will propose a novel splitting-linearizing technique to iteratively solve the NAE and develop a new algorithm for determining the optimal splitting parameters.

## 2 A Split Linearization Method

In this paper, a new splitting idea is implemented and applied to decompose Eq. (1), and we develop an iterative method to solve it. In the general case, Eq. (1) can be expressed as follows:

$$\mathbf{A}\mathbf{x} + \mathbf{B}(\mathbf{x}) \mathbf{x} = \mathbf{b}, \quad (11)$$

where  $\mathbf{A}$  and  $\mathbf{B}$  are both  $n \times n$  matrices and  $\mathbf{b}$  are constant  $n$ -vector. While  $\mathbf{A}$  is a constant matrix,  $\mathbf{B}$  is a matrix function of  $\mathbf{x}$ . When  $\mathbf{A}\mathbf{x}$  incorporates all linear terms in the NAEs,  $\mathbf{B}(\mathbf{x}) \mathbf{x}$  collects the nonlinear terms.

Then, we can decompose Eq. (11) to

$$\mathbf{A}\mathbf{x} + (1 - w) \mathbf{B}(\mathbf{x}) \mathbf{x} = \mathbf{b} - w\mathbf{B}(\mathbf{x}) \mathbf{x}. \quad (12)$$

The idea of the splitting technique in [21,22] has been proposed to solve a single nonlinear equation. Future, the nonlinear term  $\mathbf{B}(\mathbf{x}) \mathbf{x}$  in Eq. (11) is first written as  $(1 - w + w) \mathbf{B}(\mathbf{x}) \mathbf{x}$ . Then,

while the term  $(1 - w) \mathbf{B}(\mathbf{x}) \mathbf{x}$  is kept on the left-hand side, the other  $w \mathbf{B}(\mathbf{x}) \mathbf{x}$  is moved to the right-hand side, as shown in Eq. (12). Above decomposition is a new splitting application, using a splitting parameter to be determined below.

We give an initial guess  $\mathbf{x}^0$  to initialize the iteration. At the  $k$ th step of the iteration,  $\mathbf{x}^k$  is known, and Eq. (12) is linearized around  $\mathbf{x}^k$  to

$$\mathbf{A} \mathbf{x} + (1 - w) \mathbf{B}(\mathbf{x}^k) \mathbf{x} = \mathbf{b} - w \mathbf{B}(\mathbf{x}^k) \mathbf{x}^k. \quad (13)$$

For  $\mathbf{x}$ , it is a linear equations system, since the coefficients  $\mathbf{A}$  and  $\mathbf{B}(\mathbf{x}^k)$  are constant matrices. The right-hand side of Eq. (13) is a constant vector because  $\mathbf{B}(\mathbf{x}^k) \mathbf{x}^k$  is available from the previous step. A new linear system can be derived and applied to determine the new  $\mathbf{x}^{k+1}$  at the next step by

$$\mathbf{E}^k \mathbf{x}^{k+1} = \mathbf{e}^k, \quad (14)$$

where

$$\mathbf{E}^k := \mathbf{A} + (1 - w) \mathbf{B}(\mathbf{x}^k), \quad \mathbf{e}^k := \mathbf{b} - w \mathbf{B}(\mathbf{x}^k) \mathbf{x}^k. \quad (15)$$

The above process is a novel splitting-linearizing method for solving the NAEs.

In this study, the Gaussian elimination method solves Eq. (14) to determine  $\mathbf{x}^{k+1}$  at each iterative step. However, before solving Eq. (14), an unknown parameter  $w$  is given in Eq. (15). Instead of being given a constant value  $w$  in the iterative procedure of the proposed method, which is determined through some trial and error, we can formulate a new method to determine the optimal value of  $w_k$  at each step.

For a linear system

$$\mathbf{E} \mathbf{c} = \mathbf{b}, \quad (16)$$

to minimize the merit function has been derived and proved in [23–25] as follows:

$$\min \left\{ f_0 = \frac{\|\mathbf{b}\|^2 \|\mathbf{E} \mathbf{c}\|^2}{[\mathbf{b} \cdot (\mathbf{E} \mathbf{c})]^2} \right\}, \quad (17)$$

to obtain a fast descent direction  $\mathbf{c}$  for a given residual vector  $\mathbf{b}$  used in the iterative solution of Eq. (16).

Let

$$\mathbf{y} := \mathbf{E} \mathbf{c}. \quad (18)$$

The minimization in Eq. (17) is equivalent to maximize

$$\max_{\mathbf{y}} \left\{ \frac{(\mathbf{b} \cdot \mathbf{y})^2}{\|\mathbf{b}\|^2 \|\mathbf{y}\|^2} \right\}, \quad (19)$$

which is the orthogonal projection of  $\mathbf{b}/\|\mathbf{b}\|$  onto  $\mathbf{y}$ .

To optimize  $w$  in Eqs. (14) and (15),  $\mathbf{x}^{k+1}$  on the left-hand side, are expressed as  $\mathbf{x}^k$ . Then, a linear system at each step shows as follows:

$$(\mathbf{A}_1^k - w_k \mathbf{A}_2^k) \mathbf{x}^k = \mathbf{b}_1^k + w_k \mathbf{b}_2^k, \quad (20)$$

where

$$\mathbf{A}_1^k := \mathbf{A} + \mathbf{B}(\mathbf{x}^k), \quad \mathbf{A}_2^k := -\mathbf{B}(\mathbf{x}^k), \quad \mathbf{b}_1^k = \mathbf{b}, \quad \mathbf{b}_2^k = -\mathbf{B}(\mathbf{x}^k) \mathbf{x}^k. \quad (21)$$

Let

$$\mathbf{a}_1^k := \mathbf{A}_1^k \mathbf{x}^k, \mathbf{a}_2^k := \mathbf{A}_2^k \mathbf{x}^k, \tag{22}$$

and insert them into Eq. (17), we can derive

$$f_0 = \frac{f_1(w_k)f_2(w_k)}{f_3^2(w_k)}, \tag{23}$$

$$f_1(w_k) := \|\mathbf{a}_1^k + w_k \mathbf{a}_2^k\|^2 = \|\mathbf{a}_1^k\|^2 + 2w_k \mathbf{a}_1^k \cdot \mathbf{a}_2^k + w_k^2 \|\mathbf{a}_2^k\|^2, \tag{24}$$

$$f_2(w_k) := \|\mathbf{b}_1^k + w_k \mathbf{b}_2^k\|^2 = \|\mathbf{b}_1^k\|^2 + 2w_k \mathbf{b}_1^k \cdot \mathbf{b}_2^k + w_k^2 \|\mathbf{b}_2^k\|^2, \tag{25}$$

$$f_3(w_k) := (\mathbf{a}_1^k + w_k \mathbf{a}_2^k) \cdot (\mathbf{b}_1^k + w_k \mathbf{b}_2^k) = \mathbf{a}_1^k \cdot \mathbf{b}_1^k + w_k (\mathbf{a}_1^k \cdot \mathbf{b}_2^k + \mathbf{a}_2^k \cdot \mathbf{b}_1^k) + w_k^2 \mathbf{a}_2^k \cdot \mathbf{b}_2^k. \tag{26}$$

We can search the minimal value of  $w_k$  by

$$\min_{w_k \in [a_0, b_0]} \left\{ f_0 = \frac{f_1(w_k)f_2(w_k)}{f_3^2(w_k)} \right\}, \tag{27}$$

in a given interval  $[-a_0, +b_0]$  by taking  $w_k^j = a_0 + j(b_0 - a_0) / N_w, j = 1, 2, \dots, N_w$ , for example,  $a_0 = -1, b_0 = +1$  and  $N_w = 10$ .  $w_k^j$  can be used to adjust the search direction. Inserting  $w_k^j$  into Eqs. (24)–(26), we can compute  $f_0$  in Eq. (23), and then we take the optimal value of  $w_k^j$  to obtain the minimization of  $f_0$ . The optimal value of  $w_k$  in  $[a_0, b_0]$  can quickly be obtained. This new algorithm with the optimal value  $w_k$  is called the optimal split-linearization iterative method (OSLIM).

In the OSLIM, there is an interesting feature that by using the Cauchy-Schwarz inequality in Eq. (20),

$$\mathbf{b} \cdot (\mathbf{E}\mathbf{c}) \leq \|\mathbf{b}\| \|\mathbf{E}\mathbf{c}\|, \tag{28}$$

it readily leads to

$$f_0 \geq 1. \tag{29}$$

In general,  $f_0 > 1$  means that  $\mathbf{x}^k$  is not fully satisfied Eq. (11);  $f_0 = 1$  denotes that  $\mathbf{x}$  satisfies Eq. (11) and gets the convergent solution simultaneously.

The procedures of the OSLIM to solve the NAEs are summarized as follows:

- (i) Give  $n$  and nonlinear equations,  $a_0, b_0$  and  $N_w$ .
- (ii) Give an initial guess  $\mathbf{x}^0$ .
- (iii) For  $k = 0, 1, 2, \dots$ , we repeat the following computations:

$$\mathbf{A}_1^k := \mathbf{A} + \mathbf{B}(\mathbf{x}^k),$$

$$\mathbf{A}_2^k := -\mathbf{B}(\mathbf{x}^k),$$

$$\mathbf{b}_1^k = \mathbf{b},$$

$$\mathbf{b}_2^k = -\mathbf{B}(\mathbf{x}^k) \mathbf{x}^k,$$

$w_k$  from Eq. (27).

$$\mathbf{x}^{k+1} \text{ from } [\mathbf{A} + (1 - w_k) \mathbf{B}(\mathbf{x}^k)] \mathbf{x}^{k+1} = \mathbf{b} - w_k \mathbf{B}(\mathbf{x}^k) \mathbf{x}^k. \tag{30}$$

If  $\mathbf{x}^k$  converges  $\|\mathbf{x}^{k+1} - \mathbf{x}^k\| < \varepsilon$ , this iterative procedure stops; otherwise, go to step (iii) for the next iteration. Here,  $\varepsilon$  is a given convergence criterion and  $\|\mathbf{x}^{k+1} - \mathbf{x}^k\|$  named the residual norm. Also, the residual norm can be checked by

$$\left(\sum_{i=1}^n F_i^2(\mathbf{x}^k)\right)^{1/2} \leq \varepsilon. \quad (31)$$

To further assess the performance of OSLIM, we investigate it using the convergence order. For solving a scalar equation  $f(x) = 0$ , the numerically computed order of convergence (COC) is approximated by [22]

$$\text{COC} := \frac{\ln |(x_{n+1} - r) / (x_n - r)|}{\ln |(x_n - r) / (x_{n-1} - r)|}, \quad (32)$$

where  $r$  is a solution of  $f(x) = 0$  and the sequence  $x_n$  is generated from an iterative scheme. In the computation of COC, we store the values of  $x_n$  and take  $n \leq k_0 - 1$ , where  $k_0$  is the number of iterations for the convergence.

For solving the NAEs, we can generate the sequence  $\mathbf{x}^k$ ,  $k = 0, 1, \dots, k_0$  and store them in the program. We define

$$R(k) := \|\mathbf{x}^k - \mathbf{x}^e\|, \quad (33)$$

$$\text{COC} := \frac{\ln R(k_0 - 1) / R(k_0 - 2)}{\ln R(k_0 - 2) / R(k_0 - 3)}, \quad (34)$$

where  $k_0$  is the number of iterations for satisfying the convergence criterion,  $\mathbf{x}^e$  is exact solution, and  $R(k)$  is the error in terms of Euclidean norm at the  $k$ th step. If the values of  $\mathbf{x}^e$  are unknown, we take  $\mathbf{x}^e = \mathbf{x}^{k_0}$ .

### 3 Numerical Tests

#### 3.1 Test Case 1

We first consider [26]:

$$F_1(x_1, x_2) = x_1^2 + x_2^2 - 2 = 0,$$

$$F_2(x_1, x_2) = e^{x_1-1} + x_2^2 - 2 = 0. \quad (35)$$

Two roots are (1, 1) and (1, -1) obviously; however, there exist other roots.

As noted by Yeih et al. [26], the Newton method fails because the Jacobi matrix

$$\mathbf{J} = \begin{bmatrix} 2x_1 & x_2 \\ e^{x_1-1} & x_2 \end{bmatrix} \quad (36)$$

is singular at  $x_2 = 0$ . Although the scalar homotopy method [26] considered the optimal hybrid search directions, the order of the residual norm only satisfies  $10^{-4}$ . When solving this one by the residual-norm based algorithm (RNBA) [27], the residual norm satisfies the order of  $10^{-7}$ . Above iteration algorithms based on the evolution of the residual vector are not satisfied the given convergence criterion, and the third solution cannot be obtained by these algorithms.

Using the OSLIM to solve this problem, a translation is applied as follows:

$$y_1 = x_1 + d_0, y_2 = x_2,$$

and choose  $d_0$  sufficiently large to render  $y_1 > 0$  such that we have

$$\begin{aligned} -2d_0y_1 + y_1^2 + y_2^2 &= 2 - 2d_0^2, \\ e^{y_1-d_0-1} + y_2^2 &= 2. \end{aligned} \tag{37}$$

Eq. (37) can be written as

$$\begin{bmatrix} -2d_0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} y_1 & y_2 \\ \frac{e^{y_1-d_0-1}}{y_1} & y_2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2 - 2d_0^2 \\ 2 \end{bmatrix}, \tag{38}$$

of which

$$\mathbf{A} = \begin{bmatrix} -2d_0 & 0 \\ 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} y_1 & y_2 \\ \frac{e^{y_1-d_0-1}}{y_1} & y_2 \end{bmatrix}. \tag{39}$$

By using the OS LIM, we take  $d_0 = 1, a_0 = -1, b_0 = 0$  and  $N_w = 10$ , and starting from (1.5, 1.5) and through 34 iterations it converges under  $\varepsilon = 10^{-15}$  as shown in Fig. 1a. The third solution is obtained as  $(x_1, x_2) = (-0.4776700623, 1.331101541)$ , and the residuals are  $(|F_1|, |F_2|) = (4.44 \times 10^{-16}, 6.66 \times 10^{-16})$ . In Fig. 1b, the gradient direction is quickly found at the 5th step, and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . The algorithm adjusts the gradient direction to approach the solution to obtain a smaller step size. As shown in Fig. 1c, the values of  $w$  change drastically after 15 steps, and the change value of  $w$  is between  $[a_0, b_0] = [-1, 0]$ . For this example,  $\text{COC} = 1.2838$  is obtained, which indicates that the OS LIM converges faster than the iterative scheme with linear convergence speed.

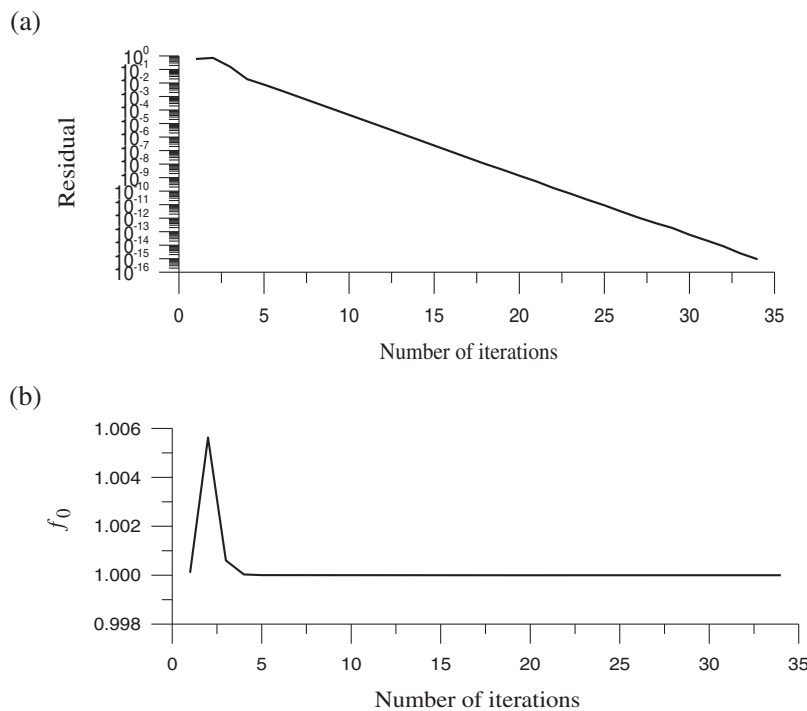
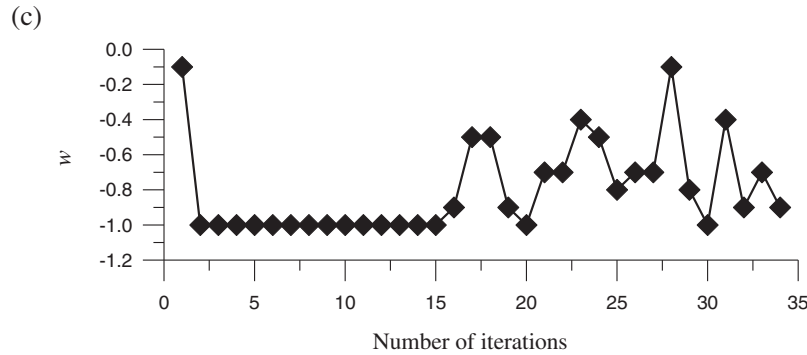


Figure 1: (Continued)



**Figure 1:** For test case 1, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

### 3.2 Test Case 2

Considering two-variable nonlinear equations [28] as follows:

$$F_1(x, y) = x^3 - 3xy^2 + a_1(2x^2 + xy) + b_1y^2 + c_1x + a_2y = 0, \quad (40)$$

$$F_2(x, y) = 3x^2y - y^3 - a_1(4xy - y^2) + b_2x^2 + c_2 = 0. \quad (41)$$

Hirsch et al. [28] used the continuous Newton algorithm to calculate this problem; however, the numerical results are inaccurate. Under the same situation, the FTIM [6], modified Newton method [9], and RNBA [27] have been applied and obtained three solutions from three different initial guessing values. In Liu et al. [29], this problem solved by the optimal vector drive algorithm (OVDA) can find five solutions. As stated in the literature [6], the attracting sets for several fixed points of problem are undefined. Therefore, various methods may give different solutions, even with the same initial value.

Eqs. (40) and (41) can be written as

$$\begin{bmatrix} c_1 & a_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x^2 - 3y^2 + 2a_1x + a_1y & b_1y \\ 3xy - 4a_1y + b_2x & -y^2 + a_1y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -c_2 \end{bmatrix}, \quad (42)$$

or written as

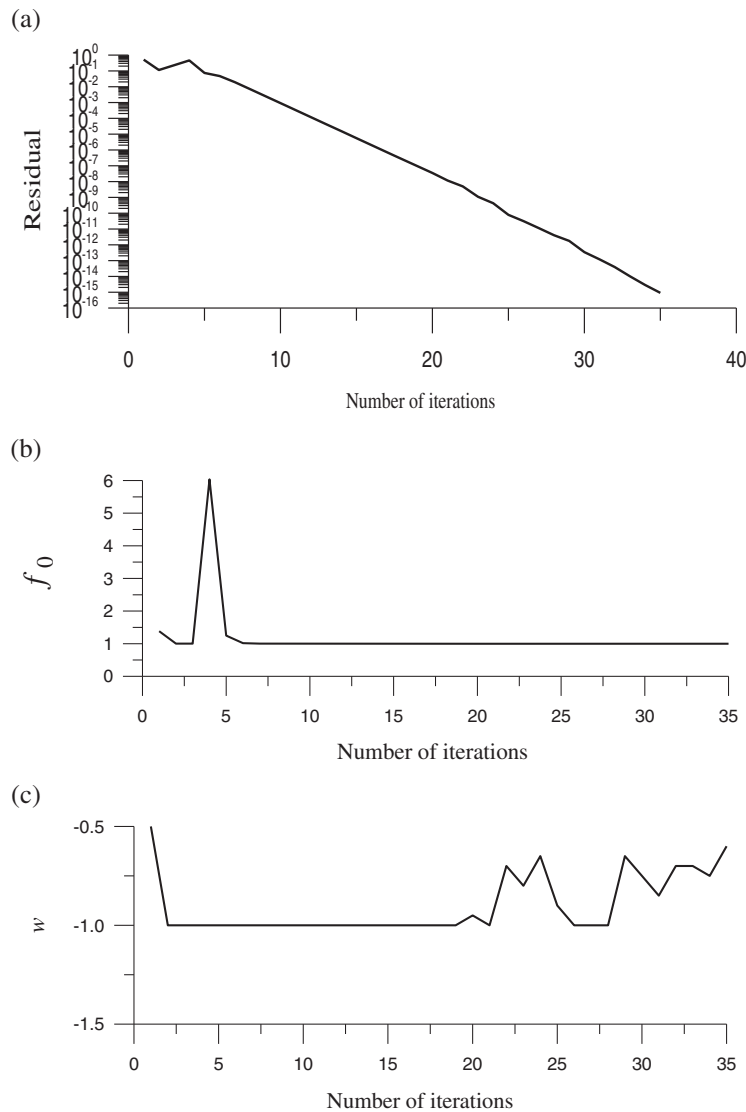
$$\begin{bmatrix} c_1 & a_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x^2 + 2a_1x & b_1y - 3xy + a_1x \\ b_2x & 3x^2 - y^2 + a_1y - 4a_1x \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ -c_2 \end{bmatrix}. \quad (43)$$

Eqs. (42) and (43) are of the form in Eq. (14), but with different  $\mathbf{B}$ . For the uniqueness of  $\mathbf{B}$ , for example, we can put the product term  $-3xy^2$  as  $(-3y^2)x$ , such that  $-3y^2$  in the brace is viewed as the coefficient which is inserted into the coefficient matrix  $\mathbf{B}$ , while  $x$  is put into the unknown vector. The principle is that the former variable  $x$  is preferred over the latter variable  $y$  placed in the unknown vector.

To compare with the result obtained by Atluri et al. [9], we take  $a_1 = 25$ ,  $b_1 = -1$ ,  $c_1 = -2$ ,  $a_2 = -3$ ,  $b_2 = -4$  and  $c_2 = -5$ . In [9], starting from  $(x_0, y_0) = (0.1, 0.1)$  and through 101 steps the solution  $(x, y) = (0.134212, 0.811128)$  is obtained with the residuals being  $(|F_1|, |F_2|) = (1.141 \times 10^{-7}, 8.913 \times 10^{-7})$ . By using the OSLIM, we take  $a_0 = -1$ ,  $b_0 = -0.5$  and  $N_w = 10$ , and it converges with 35 iterations under  $\varepsilon = 10^{-15}$  as shown in Fig. 2a. The solution  $(x, y) = (-0.16363472339, 0.23052874358)$  is obtained, and the residuals  $(|F_1|, |F_2|) = (2.22 \times 10^{-15}, 5.33 \times 10^{-15})$  are much smaller than that in [9]; moreover, the OSLIM converges faster than that obtained by Atluri et al. [9]. In Fig. 2b, the gradient



direction is quickly found at the 6th step, and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . The values of  $w$  change drastically after 19 steps, and the change value of  $w$  is between  $[a_0, b_0] = [-1, -0.5]$  as shown in Fig. 2c. For this example,  $COC = 1.0895$  is obtained.



**Figure 2:** For test case 2, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

### 3.3 Test Case 3

Considering a system of three algebraic equations with three variables:

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

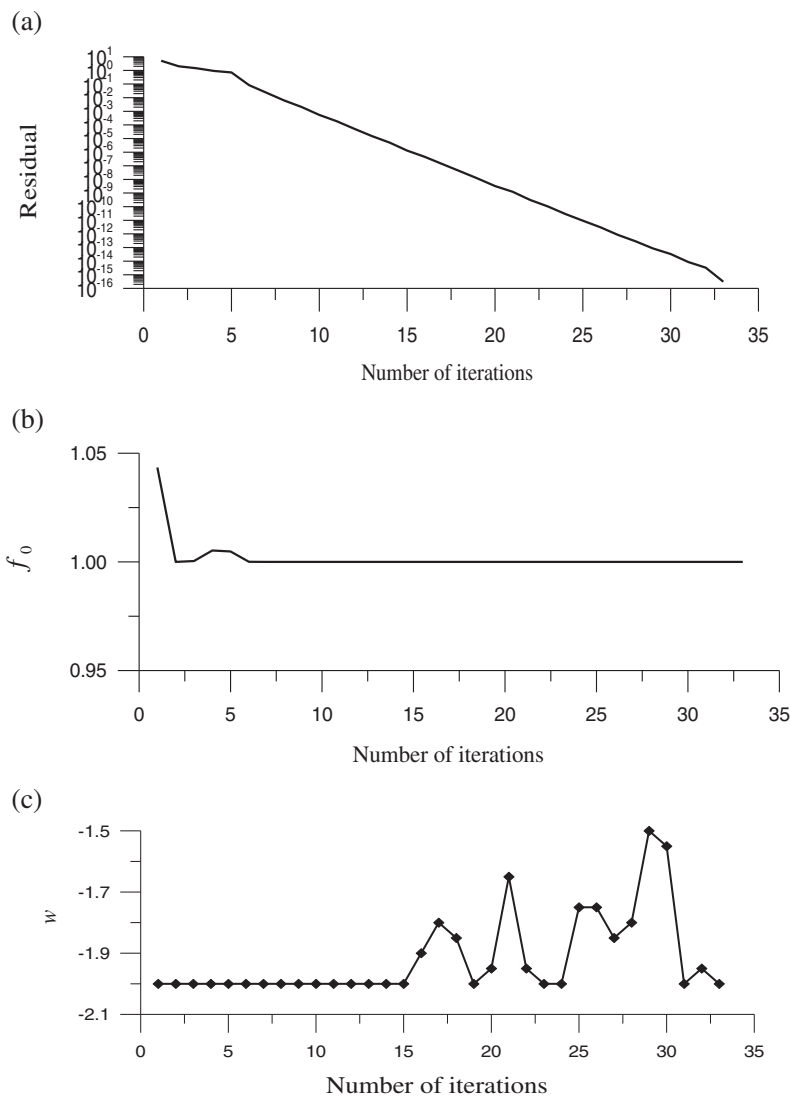
$$F_2 = xy + 2y^2 + 4z^2 - 7 = 0,$$

$$F_3 = x^8 + y^4 + z^9 - 3 = 0.$$

(44)

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

By using the OSLIM, we take  $a_0 = -2$ ,  $b_0 = -1.5$  and  $N_w = 10$ , and starting from  $(x_0, y_0, z_0) = (0.5, 0.5, 0.6)$ , through 33 iterations it converges under  $\varepsilon = 10^{-15}$  as shown in Fig. 3a. The solution  $(x, y, z) = (0.9305422841, 1.218366932, 0.8510907842)$  is obtained, and the residuals are  $(|F_1|, |F_2|, |F_3|) = (0, 3.553 \times 10^{-15}, 1.332 \times 10^{-15})$ . The OSLIM converges faster and more accurate than obtained by Atluri et al. [9], where 140 steps are required, and the residual errors are  $(|F_1|, |F_2|, |F_3|) = (6.046 \times 10^{-4}, 7.574 \times 10^{-4}, 3.005 \times 10^{-6})$ . In Fig. 3b, the gradient direction is quickly found at the 6th step, and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . When approaching solutions, the values of  $w$  of the algorithm adjusts the iterative step size after 15 steps, and the change value of  $w$  is between  $[a_0, b_0] = [-2, -1.5]$  as shown in Fig. 3c. For this example, COC = 1.0429 is obtained.



**Figure 3:** For test case 3, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

**3.4 Test Case 4**

Here, considering the following boundary value problem:

$$u'' = \frac{3}{2}u^2, u(0) = 4, u(1) = 1. \tag{45}$$

The exact solution is

$$u(x) = \frac{4}{(1+x)^2}. \tag{46}$$

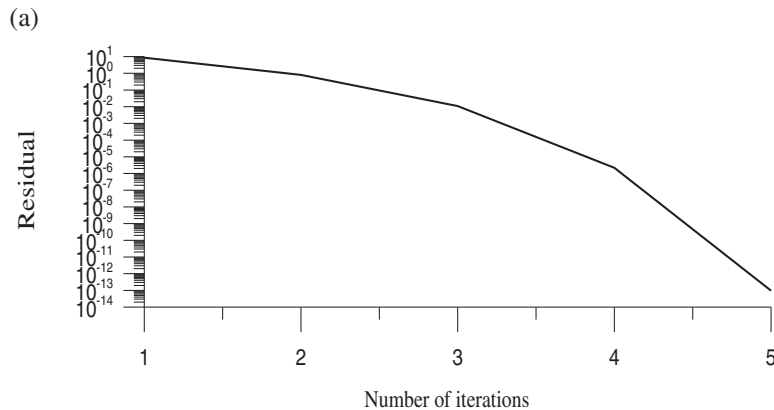
By using the finite difference discretisation of  $u$  at the grid points, we can obtain

$$F_i = \frac{1}{(\Delta x)^2} (u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i^2 = 0, i = 1, \dots, n, u_0 = 4, u_{n+1} = 1, \tag{47}$$

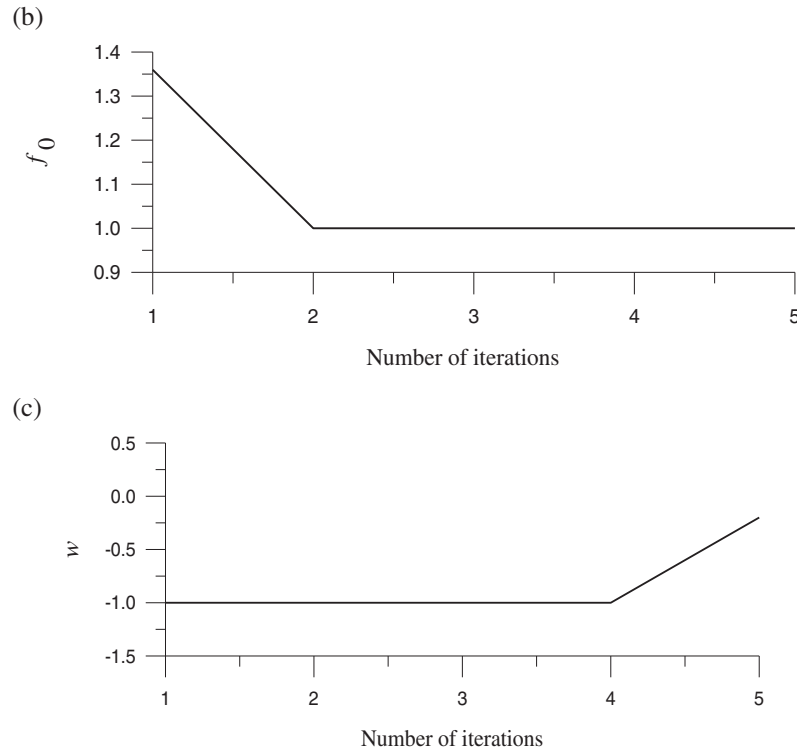
where  $\Delta x = 1/(n+1)$  is the element length. In order to conveniently describe the coefficient matrices of  $\mathbf{B}$  as Eq. (21),  $\mathbf{A}_1^k$ ,  $\mathbf{A}_2^k$ , and  $\mathbf{b}_2^k$  can be expressed as follows:

$$A_1^k = \frac{1}{(\Delta x)^2} (u_{i+1} - 2u_i + u_{i-1}) - \frac{3}{2}u_i, A_2^k = \frac{3}{2}u_i, b_2^k = \frac{3}{2}u_i^2, i = 1, \dots, n. \tag{48}$$

Setting  $n = 39$ ,  $a_0 = -1$ ,  $b_0 = 1$ ,  $\varepsilon = 10^{-10}$  and  $N_w = 10$ , the OSLIM converges within 5 iterations as shown in Fig. 4a. In Fig. 4b, the gradient direction is quickly found at the 2nd step when the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . When approaching solutions, the values of  $w$  of the algorithm adjusts the iterative step size after the 4th step, and the change value of  $w$  is between  $[a_0, b_0] = [-1, -0.25]$  as shown in Fig. 4c. The maximum error obtained by the OSLIM is  $2.98 \times 10^{-4}$ , which converges faster and is more accurate than Atluri et al. [9]. In [9], the iteration number is more than 8000 steps under the convergence criterion  $\varepsilon = 10^{-4}$ , and the errors are in the order of  $10^{-3}$ . COC = 1.9741 is obtained, which reveals that the OSLIM is almost quadratic convergence.



**Figure 4:** (Continued)



**Figure 4:** For test case 4, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

### 3.5 Test Case 5

Then, considering a test example given by Krzyworzcka [30] as follows:

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

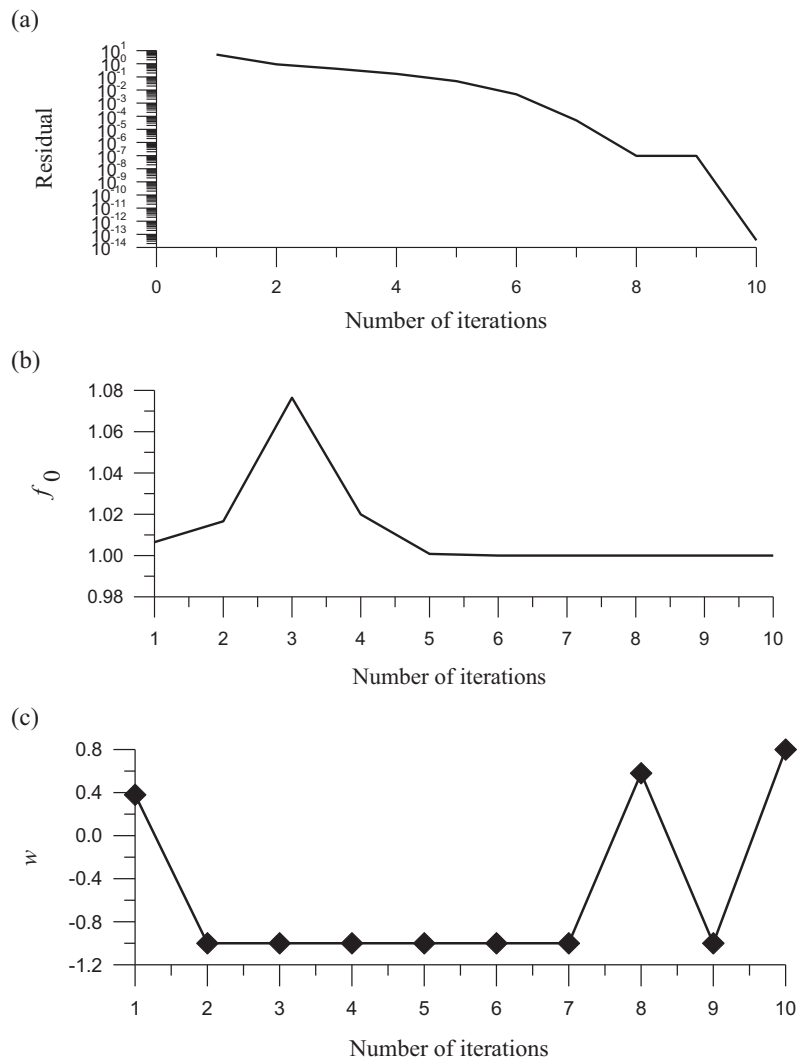
$$F_i = (3 - 5x_i)x_i - x_{i-1} - 2x_{i+1}, \quad i = 2, \dots, 9,$$

$$F_{10} = (3 - 5x_{10})x_{10} + 1 - x_9. \quad (49)$$

To convince describe the coefficient matrices of  $\mathbf{B}$  as Eq. (21),  $\mathbf{A}_1^k$ ,  $\mathbf{A}_2^k$ , and  $\mathbf{b}_2^k$  can be expressed as follows:

$$A_1^k = (-2x_{i+1} + 3x_i - x_{i-1}) - 5x_i, \quad A_2^k = 5x_i, \quad b_2^k = 5x_i^2, \quad i = 1, \dots, 10. \quad (50)$$

Setting  $a_0 = -1$ ,  $b_0 = 1$ ,  $\varepsilon = 10^{-10}$ ,  $N_w = 10$  and initial values  $x_i = 1$ ,  $i = 1, \dots, 10$ , the OSLIM converges within 10 iterations as shown in Fig. 5a. In Fig. 5b, the gradient direction is quickly found at the 5th step and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . When approaching solutions, the values of  $w$  of the algorithm adjusts the iterative step size after the 7th steps, and the change value of  $w$  is between  $[a_0, b_0] = [-1, 0.8]$  as shown in Fig. 5c. The calculation results of  $x_i$ ,  $i = 1, \dots, 10$ , are shown in Table 1. The OSLIM converges faster than that obtained by Atluri et al. [9], where 55 steps are required to satisfy a convergence criterion  $\varepsilon = 10^{-6}$ . COC = 1.9996 is obtained, which shows again that the OSLIM is quadratic convergence.



**Figure 5:** For test case 5, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

**Table 1:** The calculation results of test case 5

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
-0.2803136806	-0.1169099198	-0.0693778627	-0.0576450538	-0.0600860299
$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
-0.0703323454	-0.0878221002	-0.1158487807	-0.1634144710	-0.2539580395

### 3.6 Test Case 6

The following example is given by Roose et al. [31]:

$$F_i = 3x_i(x_{i+1} - 2x_i + x_{i-1}) + \frac{1}{4}(x_{i+1} - x_{i-1})^2, \quad i = 1, \dots, 19, u_0 = 0, u_{n+1} = 20. \quad (51)$$

The OSLIM can adopt two different formulations to solve Eq. (53). The two different formulations lead to different coefficient matrices of  $\mathbf{B}$ . First, we adopt

$$\begin{aligned} & 3(1-w)(x_2^k - 2x_1^k)x_1 + \frac{1-w}{4}x_2^kx_2 = -3w[x_2^k - 2x_1^k]x_1^k - \frac{w}{4}[x_2^k]^2, \\ & 3(1-w)(x_{i+1}^k - 2x_i^k + x_{i-1}^k)x_i + \frac{(1-w)}{4}[x_{i+1}^kx_{i+1} - x_{i-1}^kx_{i+1} - x_{i+1}^kx_{i-1} + x_{i-1}^kx_{i-1}] \\ & = -3wx_i^k[x_{i+1}^k - 2x_i^k + x_{i-1}^k] - \frac{w}{4}\left([x_{i+1}^k]^2 - 2x_{i-1}^kx_{i+1}^k + [x_{i-1}^k]^2\right), \quad i = 2, \dots, n-1, \\ & 60x_n - 10x_{n-1} + 3(1-w)(x_{n-1}^k - 2x_n^k)x_n + \frac{1-w}{4}x_{n-1}^kx_{n-1} \\ & = -100 - 3w[x_{n-1}^k - 2x_n^k]x_n^k - \frac{w}{4}[x_{n-1}^k]^2, \end{aligned} \quad (52)$$

To convince desdescribe coefficient matrices of  $\mathbf{B}$  as Eq. (21),  $\mathbf{A}_2^k$  can be expressed as follows:

$$\begin{aligned} A_2^k &= -3(x_2^k - 2x_1^k)x_1 - \frac{x_2^k}{4}x_2, \\ A_2^k &= -3(x_{i+1}^k - 2x_i^k + x_{i-1}^k)x_i - \frac{1}{4}[(x_{i+1}^k - x_{i-1}^k)x_{i+1} + (-x_{i+1}^k + x_{i-1}^k)x_{i-1}], \quad i = 2, \dots, n-1, \\ A_2^k &= -3(-2x_n^k + x_{n-1}^k)x_n - \frac{x_{n-1}^k}{4}x_{n-1}. \end{aligned} \quad (53)$$

where the number with a superscript  $k$  denotes the  $k$ th step value, and it is inserted in the coefficient matrix when we derive the linear system. We modify the term

$$(x_{i+1}^k - 2x_i^k + x_{i-1}^k)x_i$$

in the second line to

$$x_i^kx_{i+1} - 2x_i^kx_i + x_i^kx_{i-1},$$

such that we have a second formulation:

$$\begin{aligned} & 3(1-w)(x_2^k - 2x_1^k)x_1 + \frac{1-w}{4}x_2^kx_2 = -3w[x_2^k - 2x_1^k]x_1^k - \frac{w}{4}[x_2^k]^2, \\ & 3(1-w)[x_i^kx_{i+1} - 2x_i^kx_i + x_i^kx_{i-1}] + \frac{(1-w)}{4}[x_{i+1}^kx_{i+1} - x_{i-1}^kx_{i+1} - x_{i+1}^kx_{i-1} + x_{i-1}^kx_{i-1}] \\ & = -3wx_i^k[x_{i+1}^k - 2x_i^k + x_{i-1}^k] - \frac{w}{4}\left([x_{i+1}^k]^2 - 2x_{i-1}^kx_{i+1}^k + [x_{i-1}^k]^2\right), \quad i = 2, \dots, n-1, \\ & 60x_n - 10x_{n-1} + 3(1-w)(x_{n-1}^k - 2x_n^k)x_n + \frac{1-w}{4}x_{n-1}^kx_{n-1} \\ & = -100 - 3w[x_{n-1}^k - 2x_n^k]x_n^k - \frac{w}{4}[x_{n-1}^k]^2. \end{aligned} \quad (54)$$

To convince describe the coefficient matrices of  $\mathbf{B}$  as Eq. (21),  $A_2^k$  can be expressed as follows:

$$\begin{aligned}
 A_2^k &= -3(x_2^k - 2x_1^k)x_1 - \frac{x_2^k}{4}x_2, \\
 A_2^k &= 6x_i^k x_i - \left[ \frac{(x_{i+1}^k - x_{i-1}^k)}{4} + 3x_i^k \right] x_{i+1} - \left[ \frac{(-x_{i+1}^k + x_{i-1}^k)}{4} + 3x_i^k \right] x_{i-1}, i = 2, \dots, n - 1, \\
 A_2^k &= -3(-2x_n^k + x_{n-1}^k)x_n - \frac{x_{n-1}^k}{4}x_{n-1}.
 \end{aligned}
 \tag{55}$$

Setting  $n = 10$ ,  $a_0 = -0.1$ ,  $b_0 = 0.1$ ,  $\varepsilon = 10^{-10}$ ,  $N_w = 10$  and initial values  $x_i = x_i = 2i$ ,  $i = 1, \dots, 10$ , the first OSLIM with the first formulation converges within 14 iterations as shown in Fig. 6a. COC = 1.0651 is obtained. In Fig. 6b, the gradient direction is quickly found at the 4th step and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$ . In Fig. 6c, the value of  $w$  of the algorithm adjusts the iterative step size after the 8th step, and the change value of  $w$  is between  $[a_0, b_0] = [-0.1, 0.04]$  as shown in Fig. 6c. The OSLIM converges faster than that obtained by Atluri et al. [9], where over 700 steps are required under  $\varepsilon = 10^{-6}$ . The residual errors of  $|F_i|$  and the calculation results of  $x_i$ ,  $i = 1, \dots, 10$ , are listed in Tables 2 and 3, respectively.

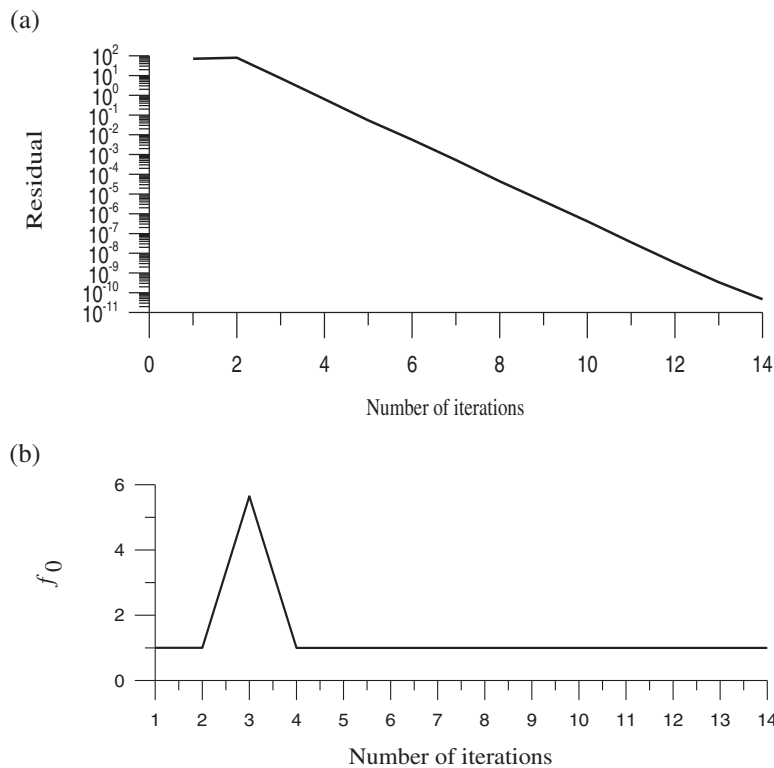
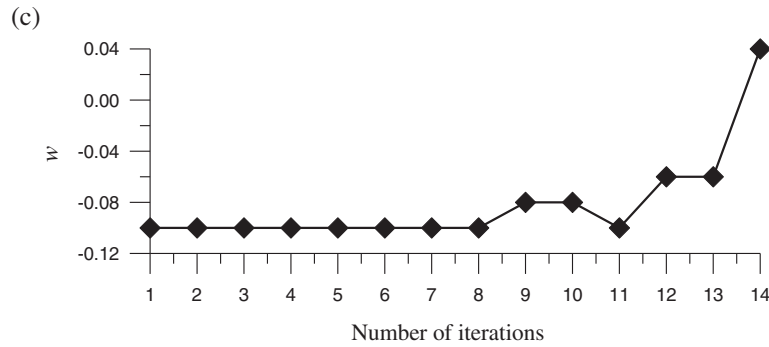


Figure 6: (Continued)



**Figure 6:** For test case 6, (a) display the residual plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

**Table 2:** The residuals of the calculation results of test case 7

$ F_1 $	$ F_2 $	$ F_3 $	$ F_4 $	$ F_5 $
$3.4598756 \times 10^{-21}$	$2.8717257 \times 10^{-19}$	$3.4456428 \times 10^{-19}$	$9.2828361 \times 10^{-17}$	$1.7105604 \times 10^{-15}$
$ F_6 $	$ F_7 $	$ F_8 $	$ F_9 $	$ F_{10} $
$6.9887644 \times 10^{-15}$	$5.7579812 \times 10^{-13}$	$1.3318470 \times 10^{-13}$	$1.6290491 \times 10^{-10}$	$6.6123107 \times 10^{-10}$

**Table 3:** The calculation results of test case 7

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$
$7.49745 \times 10^{-11}$	$1.01856 \times 10^{-9}$	$1.41193 \times 10^{-8}$	$1.96674 \times 10^{-7}$	$2.73944 \times 10^{-6}$
$x_6$	$x_7$	$x_8$	$x_9$	$x_{10}$
$3.81553 \times 10^{-5}$	$5.31434 \times 10^{-4}$	$7.40193 \times 10^{-3}$	0.1030955	1.4359354

Setting  $n = 10$ ,  $a_0 = -0.1$ ,  $b_0 = 0.1$ ,  $\varepsilon = 10^{-10}$ ,  $N_w = 100$  and initial values  $x_i = x_i = 2i$ ,  $i = 1, \dots, 10$ , the second OSLIM with the second formulation converges within 13 iterations as shown in Fig. 7a. However, the solutions are different from those obtained from the first OSLIM with the first formulation. When  $n$  is raised to 50, both OSLIM can find solutions quickly with 12 and 15 iterations, respectively. These two solutions are compared in Fig. 7b.

### 3.7 Test Case 7

Considering the following a two-dimensional nonlinear Bratu equation:

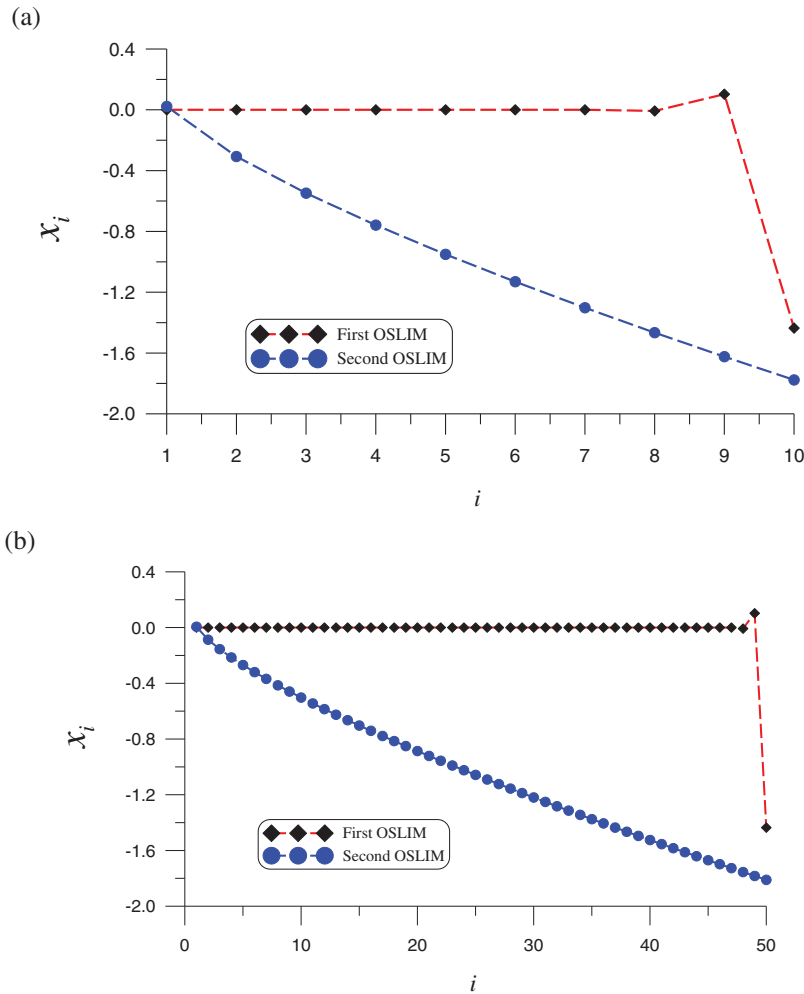
$$\Delta u(x, y) + \exp u(x, y) = S(x, y), \quad (56)$$

$$u(x, y) = \ln(x^2 + y^2 + 2), \quad (57)$$

and the domain is an ellipse with semi-axes lengths  $a = 1$  and  $b = 0.5$ . The Dirichlet boundary condition is imposed. Due to  $u > 0$ , we decompose it to

$$\Delta u(x, y) + (1 - w) \frac{\exp u^k(x, y)}{u^k(x, y)} u(x, y) = S(x, y) - w \exp u^k(x, y), \quad (58)$$





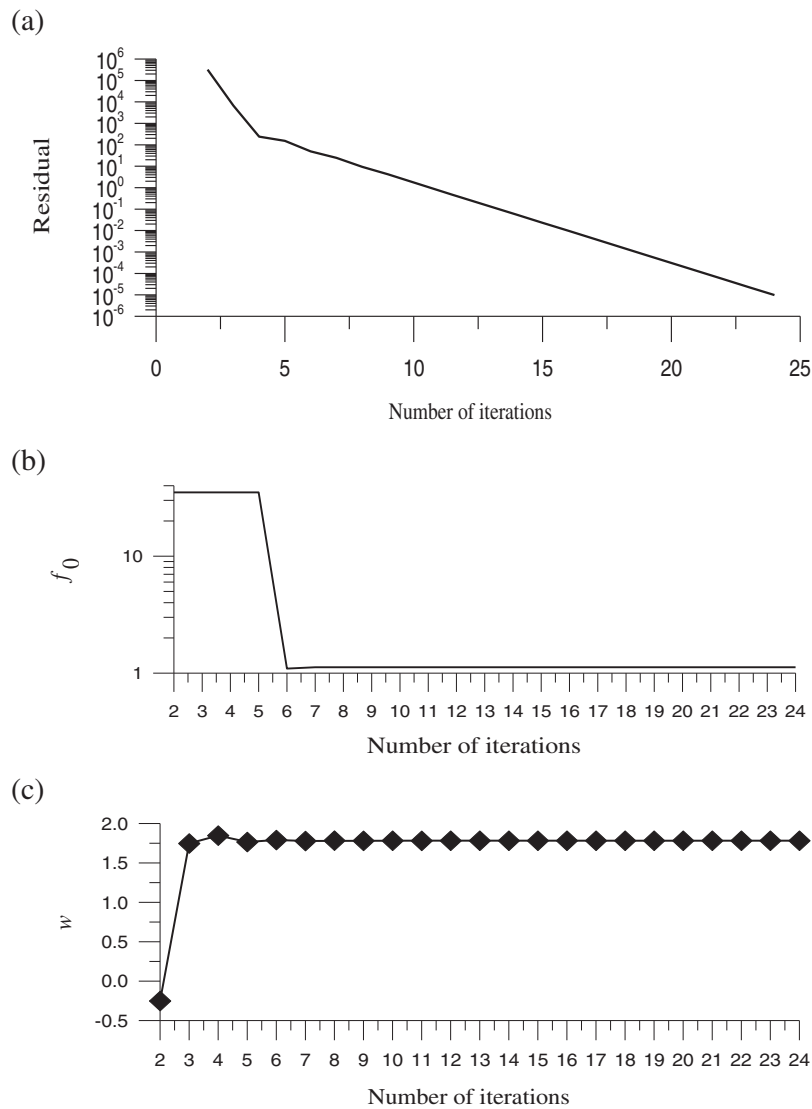
**Figure 7:** For test case 6, comparing two different solutions obtained by the first OSLIM and the second OSLIM: (a)  $n = 10$ , and (b)  $n = 50$

where  $u^k(x, y)$  is the value at the  $k$ th step, expanded by

$$u^k(x, y) = \sum_{i=0}^M \sum_{j=0}^i a_{ij}^k x^{i-j} y^j. \tag{59}$$

After collocating points inside the domain and on the boundary, simultaneously, to satisfy the Eq. (56) and the boundary condition equation, we can derive NAEs to determine the  $n = (M + 1)(M + 2)$  coefficients  $a_{ij}$  iteratively.

By using the OSLIM, we take  $n = 12 \times 13/2 = 78$ , and to determine  $w$  by using the Newton method, through 24 iterations it converges under  $\varepsilon = 10^{-5}$  as shown in Fig. 8a.  $\text{COC} = 1.4926$  is obtained. The gradient direction is quickly found at the 7th step, and then the value of  $f_0$  quickly tend to the minimum value  $f_0 = 1$  as shown in Fig. 8b. In order to satisfy  $f_0 = 1$ , we can find that the algorithm adjusts the search direction after the 3rd step and the change value of  $w$  is between  $[a_0, b_0] = [-0.25, 1.78]$  as shown in Fig. 8c. The maximum error obtained by the OSLIM is  $7.95 \times 10^{-6}$ . Interestingly, the OSLIM can be used to solve a highly nonlinear PDE with high accuracy.



**Figure 8:** For test case 7, (a) display the convergence plot, (b) the minimum value of the merit function, and (c) the optimized value of the splitting parameter

#### 4 Conclusions

When evaluating the performance of newly developed iterative solutions for solving nonlinear algebraic equations (NAE), there are two critical factors: accuracy and convergence speed. In this article, we first decomposed the nonlinear terms in the NAEs through a splitting parameter and then we linearized the NAEs around the values at the previous step and derived a system of linear algebraic equations (LAEs). Through the maximum orthogonal projection concept at each iterative step, we select the optimal value of the splitting parameter and use the Gaussian elimination method to solve the LAE. The features of the proposed method are summarized as follows: (a) The merit function is in terms of the coefficient matrix, right-side vector and the value of unknown vector at the previous step. (b) It is easy to minimize the search within a preferred interval through some operations. (c) The

current OSLIM is insensitive to the initial guess and without needing the inversion of the Jacobian matrix at each iterative step, which can save much of the computational time. We have successfully applied the OSLIM to solve the NAEs resulting from the nonlinear boundary value problem in one- and two-dimensional spatial spaces. Through the numerical test, the convergence order of the OSLIM is between linear and quadratic convergence.

**Funding Statement:** The corresponding author appreciates the financial support provided by the Ministry of Science and Technology, Taiwan, ROC under Contract No. MOST 110-2221-E-019-044.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Liu, C. S., Atluri, S. N. (2008). A fictitious time integration method (FTIM) for solving mixed complementarity problems with applications to non-linear optimization. *Computer Modeling in Engineering & Sciences*, 34(2), 155–178. DOI 10.3970/cmcs.2008.034.155.
2. Kwak, B. M., Kwak, J. Y. (2021). Binary NCP: A new approach for solving nonlinear complementarity problems. *Journal of Mechanical Science and Technology*, 35(3), 1161–1166. DOI 10.1007/s12206-021-0229-5.
3. Shih, C. F., Chen, Y. W., Chang, J. R., Soon, S. P. (2021). The equal-norm multiple-scale trefftz method for solving the nonlinear sloshing problem with baffles. *Computer Modeling in Engineering & Sciences*, 127(3), 993–1012. DOI 10.32604/cmcs.2021.012702.
4. Nurhidayat, I., Hao, Z., Hu, C. C., Chen, J. S. (2020). An ordinary differential equation approach for nonlinear programming and nonlinear complementarity problem. *International Journal of Industrial Optimization*, 1(1), 1–14. DOI 10.12928/ijio.v1i1.764.
5. Fischer, A. (1992). A special newton-type optimization method. *Optimization*, 24(3–4), 269–284. DOI 10.1080/02331939208843795.
6. Liu, C. S., Atluri, S. N. (2008). A novel time integration method for solving a large system of non-linear algebraic equation. *Computer Modeling in Engineering & Sciences*, 31(2), 71–83.15. DOI 10.3970/CMES.2008.031.071.
7. Liu, C. S. (2009). A fictitious time integration method for a quasilinear elliptic boundary value problem, defined in an arbitrary plane domain. *Computers, Materials & Continua*, 11(1), 15–32. DOI 10.3970/cmc.2009.011.015.
8. Liu, C. S. (2001). Cone of non-linear dynamical system and group preserving schemes. *International Journal of Non-Linear Mechanics*, 36, 1047–1068. DOI 10.1016/S0020-7462(00)00069-X.
9. Atluri, S. N., Liu, C. S., Kuo, C. L. (2009). A modified newton method for solving non-linear algebraic equations. *Journal of Marine Science and Technology*, 17(3), 238–247. DOI 10.51400/2709-6998.1960.
10. Jones, C., Tian, H. (2017). A time integration method of approximate fundamental solutions for nonlinear poisson-type boundary value problems. *Communications in Mathematical Sciences*, 15(3), 693–710. DOI 10.4310/CMS.2017.v15.n3.a6.
11. Hashemi, M. S., Mustafa, I., Parto-Haghighi, M., Mustafa, B. (2019). On numerical solution of the time-fractional diffusion-wave equation with the fictitious time integration method. *European Physical Journal Plus*, 134, 488. DOI 10.1140/epjp/i2019-12845-1.
12. Demmel, J. W. (1997). *Applied numerical linear algebra*. Philadelphia: SIAM.
13. Saad, Y. (2003). *Iterative methods for sparse linear systems*. Philadelphia: SIAM.
14. Quarteroni, A., Sacco, R., Saleri, F. (2000). *Numerical mathematics*. New York: Springer Science.
15. Hadjidimos, A. (2000). Successive overrelaxation (SOR) and related methods. *Journal of Computational and Applied Mathematics*, 123(1–2), 177–199. DOI 10.1016/S0377-0427(00)00403-9.

16. Bai, Z. Z., Chi, X. B. (2003). Asymptotically optimal successive overrelaxation method for systems of linear equations. *Journal of Computational Mathematics*, 21(5), 603–612.
17. Wen, R. P., Meng, G. Y., Wang, C. L. (2013). Quasi-Chebyshev accelerated iteration methods based on optimization for linear systems. *Computers & Mathematics with Applications*, 66(6), 934–942. DOI 10.1016/j.camwa.2013.06.016.
18. Meng, G. Y. (2014). A practical asymptotical optimal SOR method. *Applied Mathematics and Computation*, 242, 707–715. DOI 10.1016/j.amc.2014.06.034.
19. Miyatake, Y., Sogabe, T., Zhang, S. L. (2020). Adaptive SOR methods based on the wolfe conditions. *Numerical Algorithms*, 84, 117–132. DOI 10.1007/s11075-019-00748-0.
20. Yang S., G., K., M. (2009). The optimal relaxation parameter for the SOR method applied to the poisson equation in any space dimensions. *Applied Mathematics Letters*, 22(3), 325–331. DOI 10.1016/j.aml.2008.03.028.
21. Liu, C. S. (2020). A new splitting technique for solving nonlinear equations by an iterative scheme. *Journal of Mathematics Research*, 12(4), 40–48. DOI 10.5539/jmr.v12n4p40.
22. Liu, C. S., Hong, H. K., Lee, T. L. (2021). A splitting method to solve a single nonlinear equation with derivative-free iterative schemes. *Mathematics and Computers in Simulation*, 190, 837–847. DOI 10.1016/j.matcom.2021.06.019.
23. Liu, C. S. (2012). A globally optimal iterative algorithm to solve an ill-posed linear system. *Computer Modeling in Engineering & Sciences*, 84(4), 383–403. DOI 10.3970/cmcs.2012.084.383.
24. Liu, C. S. (2013). An optimal tri-vector iterative algorithm for solving ill-posed linear inverse problems. *Inverse Problems in Science and Engineering*, 21, 650–681. DOI 10.1080/17415977.2012.717077.
25. Liu, C. S. (2014). A globally optimal tri-vector method to solve an ill-posed linear system. *Journal of Computational and Applied Mathematics*, 260, 18–35. DOI 10.1016/j.cam.2013.09.017.
26. Yeih, W. C., Ku, C. Y., Liu, C. S., Chan, I. Y. (2013). A scalar homotopy method with optimal hybrid search directions for solving nonlinear algebraic equations. *Computer Modeling in Engineering & Sciences*, 90(4), 255–282. DOI 10.3970/cmcs.2013.090.255.
27. Chen, Y. W. (2014). A residual-norm based algorithm for solving determinate/indeterminate systems of nonlinear algebraic equations. *Journal of Marine Science and Technology*, 22(5), 583–597. DOI 10.6119/JM-ST-013-0912-1.
28. Hirsch, M., Smale, S. (1979). On algorithms for solving  $f(x) = 0$ . *Communications on Pure & Applied Mathematics*, 32(3), 281–312. DOI 10.1002/cpa.3160320302.
29. Liu, C. S., Atluri, S. N. (2011). 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  $\alpha$  in  $\dot{\mathbf{x}} = \lambda [\alpha \mathbf{F} + (1 - \alpha) \mathbf{B}^T \mathbf{F}]$ ;  $B_{ij} = \partial F_i / \partial x_j$ . *Computer Modeling in Engineering & Sciences*, 73(4), 395–431. DOI 10.3970/cmcs.2011.073.395.
30. Krzyworzcka, S. (1996). Extension of the lanczos and CGS methods to systems of nonlinear equations. *Journal of Computational and Applied Mathematics*, 69(1), 181–190. DOI 10.1016/0377-0427(95)00032-1.
31. Roose, A., Kulla, V., Lomb, M., Meresso, T. (1990). *Test examples of systems of non-linear equations*. Tallin: Estonian Software and Computer Service Company.