# A fast algorithm to solve systems of nonlinear equations ☆

Abdolreza Amiri[b], Alicia Cordero[a], Mohammad Taghi Darvishi[b], Juan R. Torregrosa[a,*]

[a]*Instituto de Matemáticas Multidisciplinar, Universitat Politècnica de València, 46022 València, Spain*
[b]*Department of Mathematics, Faculty of Science, Razi University, Kermanshah 67149, Iran*

**Abstract**

A new HSS-based algorithm for solving systems of nonlinear equations is presented and its semilocal convergence is proved. Spectral properties of the new method are investigated. Performance profile for the new scheme is computed and compared with HSS algorithm. Besides, by a numerical example in which a two-dimensional nonlinear convection-diffusion equation is solved, we compare the new method and the Newton-HSS method. Numerical results show that the new scheme solves the problem faster than the Newton-HSS scheme in terms of CPU-time and number of iterations. Moreover, the application of the new method is found to be fast, reliable, flexible, accurate, and has small CPU time.

*Keywords:* Nonlinear systems; iterative method; Newton method; Newton-HSS method; Newton-GPSS method; Jacobian free scheme

## 1. Introduction

Let us consider the following system of nonlinear equations $F(x) = 0$, where $F : D \subseteq \mathbb{C}^n \longrightarrow \mathbb{C}^n$ is a nonlinear differentiable function and $D$ is an open set. We suppose that Jacobian matrix $F'(x)$ is a positive definite, nonsymmetric and sparse matrix. Nonlinear systems arise in different areas of scientific computing and engineering computations, especially in the discretization of nonlinear partial differential equations, boundary value problems, integral equations, etc. There are cases where thousands of nonlinear equations on some independent variables must be solved effectively. Therefore, finding roots of systems of nonlinear equations has widespread applications in numerical and applied mathematics. The most common root-finding scheme for systems of nonlinear equations is the second order classical Newton's method, with iterative expression

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1}F(x^{(k)}), \quad k = 0, 1, \ldots$$

To avoid the computation of the inverse of the Jacobian matrix, previous expression is changed to

$$F'(x^{(k)})(x^{(k+1)} - x^{(k)}) = -F(x^{(k)}),$$

which is a linear system. Therefore, for obtaining the new iteration we can solve the following linear system:

$$F'(x^{(k)})s^{(k)} = -F(x^{(k)}), \tag{1}$$

whence $x^{(k+1)} = x^{(k)} + s^{(k)}$. Thus, for applying Newton's scheme, we must solve a linear system in each iteration.

Therefore, any step of Newton's method contains two kinds of iterations. The linear one to solve (1), which is called inner iteration, and the nonlinear iteration to compute $\{x^{(k)}\}$ that is called outer iteration. Some of the most famous inner iteration methods are Jacobi, Gauss-Seidel, successive overrelaxation (SOR), accelerated overrelaxation (AOR) and Krylov subspace methods. They are based on splitting of the coefficient matrix $A$ of the linear system as $A = M - N$. Conjugate gradient (CG) and GMRES methods are widely used as outer iteration schemes.

If we apply Krylov subspace method as the inner iteration, the method is called Newton-Krylov subspace method. Similarly, Newton-CG and Newton-GMRES are methods which use CG and GMRES as the outer iterations. In the last decade, some efforts were made to present efficient splitting of the coefficient matrix for solving the linear system of

☆This research was partially supported by Ministerio de Economía y Competitividad under grants MTM2014-52016-C2-2-P and Generalitat Valenciana PROMETEO/2016/089.
*Corresponding author
*Email addresses:* amiriabdolreza@ymail.com (Abdolreza Amiri), acordero@mat.upv.es (Alicia Cordero), darvishimt@yahoo.com (Mohammad Taghi Darvishi), jrtorre@mat.upv.es (Juan R. Torregrosa )

equations (1). In this regard, some HSS-based iteration algorithms were introduced to solve linear systems. Bai et al. [1] introduced the Hermitian/skew-Hermitian splitting (HSS) iteration scheme for non-Hermitian positive definite linear systems. Bertaccini et al. [2] studied the role of preconditioning for coercive problems for two-step iterative methods, based on the Hermitian/skew-Hermitian splitting of the coefficient matrix of a nonsymmetric linear system whose real part is coercive. Bai et al. in [3] established a class of preconditioned Hermitian/skew-Hermitian splitting iterative methods, for a positive semidefinite linear system when its coefficient matrix had a two-by-two block structure.

A class of lopsided Hermitian/skew-Hermitian (LHSS) methods was established to solve non-Hermitian and positive definite systems of linear equations in [4], which included a two-step LHSS iteration, its inexact version, and the inexact Hermitian/skew-Hermitian (ILHSS) iteration. Also, a preconditioned iterative method based on HSS preconditioned is used for weighted Toeplitz least squares problems in [5]. Li et al. [6] presented an asymmetric Hermitian/skew-Hermitian (AHSS) iterative method for large sparse non-Hermitian positive definite systems of linear equations. Li et al. [7] modified the HSS method and presented the Lopsided-HSS (LHSS) iterative method.

Discretization of certain nonlinear partial differential equations results in nonlinear systems of equations of the form [8, 9]

$$F(x) := Ax - \varphi(x) = 0, \tag{2}$$

where $Ax$ and $\varphi(x)$ are linear and nonlinear parts of (2), respectively. Besides, (2) is a weakly nonlinear system when the linear part is strongly dominant on the nonlinear part. In this case, based on the separability and strong dominance between the linear term and the nonlinear one, Bai and Yang [10] introduced the Picard-HSS and the nonlinear HSS-like iterative methods. To solve large sparse systems of weakly nonlinear equations, Zhu [11] proposed a class of modified iteration methods, Picard-LHSS and nonlinear LHSS-like algorithms, by using optimal parameters for asymmetric Hermitian and skew-Hermitian splitting iteration schemes.

Picard iterative method $Ax^{(k+1)} = \varphi(x^{(k)})$ is not a suitable scheme to solve weakly nonlinear system (2). There are many iterative methods to solve nonlinear systems, but most of them need to evaluate the Jacobian function in one or some points. Some authors tried to use HSS iterative method for inner iterations of Newton's procedure. Bai and Guo [12] applied Newton-HSS method to solve systems of nonlinear equations with positive definite Jacobian matrices. By making use of the HSS iteration as the inner solver for the Newton method, Bai et al. [13] proposed a class of Newton-HSS methods for solving large sparse systems of nonlinear equations with positive definite Jacobian matrices at the solution points.

Let us split the coefficient matrix $A$ of the linear system as $A = M - N$. With this splitting we have the following iterative scheme to solve $Ax = b$

$$Mx_l = Nx_{l-1} + b, \;\; l = 1, 2, \ldots.$$

If this scheme is used as inner iteration, we obtain the following inner/outer iteration scheme [1, 16]

$$\begin{aligned} &\text{for a given } \; x^{(0)} \\ &x^{(k+1)} = x^{(k)} - (T_k^{l_k-1} + \cdots + T_k^2 + T_k + I)M_k^{-1}F(x^{(k)}), \\ &T_k = M_k^{-1}N_k, \end{aligned} \tag{3}$$

where $l_k$ is the number of inner iteration steps, $M_k$ and $N_k$ are splitting parts of the matrix $F'(x^{(k)})$, that is, $F'(x^{(k)}) = M_k - N_k, \; k = 0, 1, \ldots.$

To solve nonlinear systems with Newton-HSS method, one uses HSS and Newton methods for inner and outer iterations, respectively. As we mentioned above, Newton's scheme needs to evaluate the Jacobian in one or some points which is too time consuming, so any method that reduces the number of evaluations of Jacobian is widely welcome. Here, we present a fast algorithm to solve the nonlinear system $F(x) = 0$. In our algorithm, HSS method is the inner iteration as well. But, for the outer iteration, we use an inexact version of the following third order Newton-like iterative method

$$x^{(k+1)} = x^{(k)} - [F'(x^{(k)})]^{-1}\left(F(x^{(k)}) + F(x_*^{(k+1)})\right) \tag{4}$$

where $x_*^{(k+1)}$ is a Newton step. This method is known as Traub's scheme and has order of convergence three [14].

In fact, this study presents a fast HSS-based algorithm to solve systems of nonlinear equations. In this algorithm, an inexact version of the third order Newton-like method (4) is applied as an outer iteration. We prove that this algorithm is convergent. Also, numerical results show that the new algorithm is faster than Newton-HSS scheme in terms of CPU-time and number of outer iterations. We also compare our algorithms with Picard-HSS and Nonlinear HSS-like introduced in [10], with good results.

The rest of the paper is organized as follows: In Section 2, the new algorithm is presented. Semilocal convergence of the new method is proven in Section 3. In Section 4, some computational tests are presented which confirm the theoretical results. Finally, some concluding remarks are shown in Section 5.

## 2. The design of the new method

Let us consider a system of linear equations of size $n \times n$, $Ax = b$. Suppose that $H$ and $S$ are the Hermitian and skew-Hermitian parts of $A$, respectively, that is, $A = H + S$ with $H = \frac{1}{2}(A + A^*)$ and $S = \frac{1}{2}(A - A^*)$, where $A^*$ denotes the conjugate transposed of $A$. Now, for an initial guess $x^{(0)} \in \mathbb{C}^n$, and positive constants $\alpha$ and $\mathrm{tol}$, HSS scheme presented in [1] computes $x^{(l)}$, for $l = 0, 1, 2, \ldots$, by

$$\begin{cases} (\alpha I + H)x^{(l+\frac{1}{2})} = (\alpha I - S)x^{(l)} + b, \\ (\alpha I + S)x^{(l+1)} = (\alpha I - H)x^{(l+\frac{1}{2})} + b, \end{cases} \tag{5}$$

where $I$ denotes the identity matrix of size $n \times n$. Stopping criterion for relations (5) is $\|b - Ax^{(l)}\| \leq \mathrm{tol}\|b - Ax^{(0)}\|$, for an initial guess $x^{(0)}$ and a given tolerance $\mathrm{tol}$.

If HSS method is used to solve the linear system obtained at each iteration of Picard method $Ax^{(k+1)} = \varphi(x^{(k)})$, it leads to the following inexact Picard iteration method, which is called Picard-HSS.

**The Picard-HSS Iteration Method.** [10] Let $\varphi : D \subset \mathbb{C}^n \to \mathbb{C}^n$ be a continuous function and $A \in \mathbb{C}^{n \times n}$ be a positive definite matrix. Suppose that $H$ and $S$ are Hermitian and skew-Hermitian parts of $A$, respectively, that is, $A = H + S$ with $H = \frac{1}{2}(A + A^*)$ and $S = \frac{1}{2}(A - A^*)$. Given an initial guess $x_0$ and a positive integer sequence $\{l_n\}_{n=0}^{\infty}$ and using the following iteration scheme to compute $x^{(n+1)}$ for $n = 0, 1, 2, \ldots$ until stopping criterion is satisfied,

1) Set $x_0^{(n)} := x^{(n)}$;

2) For $l = 0, 1, 2, \cdots, n-1$, solve the following linear systems to obtain $x^{(n+1)}$:

$$\begin{cases} (\alpha I + H)s_{l+\frac{1}{2}}^{(n)} = (\alpha I - S)s_l^{(n)} + \varphi(x^{(n)}) \\ (\alpha I + S)s_{l+1}^{(n)} = (\alpha I - H)s_{l+\frac{1}{2}}^{(n)} + \varphi(x^{(n)}) \end{cases}$$

where $\alpha$ is a positive constant and $I$ denotes the identity matrix.

3) Set $x^{(n+1)} := x_{l_n}^{(n)}$.

Determining the quantity of inner iterations $l_n$ at each step is problem dependent. Usually a modified form of Picard scheme called nonlinear Picard is used, which does not use the stopping criterion at each step of Picard iteration.

**The Nonlinear HSS-like Iteration Method.** [10] Given an initial guess $x_0 \in D \subset \mathbb{C}^n$, let us compute $x^{(n+1)}$ for $n = 0, 1, 2, \ldots$ by using the following iteration scheme until the stopping criterion is satisfied,

$$\begin{cases} (\alpha I + H)s^{(n+1)} = (\alpha I - S)s^{(n)} + \varphi(x^{(n+\frac{1}{2})}), \\ (\alpha I + S)s^{(n+\frac{1}{2})} = (\alpha I - H)s^{(n)} + \varphi(x^{(n)}) \end{cases},$$

where $\alpha$ is a positive constant and $I$ denotes the identity matrix.

To use the third order Traub's algorithm (4) as a root-finder of nonlinear systems, we must solve the following two systems:
For a given $x^{(0)}$, first we solve

$$F'(x^{(k)})d_1^{(k)} = -F(x^{(k)}) \tag{6}$$

and then we set $x_*^{(k+1)} = x^{(k)} + d_1^{(k)}$. Second, since from (4) we can write

$$\begin{aligned} -(F(x^{(k)}) + F(x_*^{(k+1)})) &= F'(x^{(k)})(x^{(k+1)} - x_*^{(k+1)} + x_*^{(k+1)} - x^{(k)}) \\ &= F'(x^{(k)})(d_2^{(k)} + d_1^{(k)}), \end{aligned}$$

hence, by using (6), we must solve the following system

$$F'(x^{(k)})d_2^{(k)} = -F(x_*^{(k+1)}), \tag{7}$$

whence $x^{(k+1)} = x_*^{(k+1)} + d_2^{(k)}$.

Now, we present the new HSS-based algorithm for approximating a solution of $F(x) = 0$. In this new method, HSS scheme is our inner iteration and the third-order Newton-like method is the outer iteration. This new scheme is denoted by INHSS. Algorithm 1 shows Traub's iterative algorithm presented by relations (6) and (7).

**Algorithm 1:** Inexact version of the third order Newton-like scheme (4)

**Input**: $x^{(0)}$ and tol.
For $k = 1, 2, \ldots$ until $\|F(x^{(k)})\| \leq \text{tol} \parallel F(x^{(0)}) \parallel$ do:
For $\eta_k \in [0, 1)$ find $\mathbf{d}_1^{(k)}$ such that

$$\parallel F(x^{(k)}) + F'(x^{(k)})d_1^{(k)} \parallel < \eta_k \parallel F(x^{(k)}) \parallel \tag{8}$$

Set $x_*^{(k+1)} = x^{(k)} + d_1^{(k)}$.
Find $d_2^{(k)}$ such that

$$\parallel F(x_*^{(k+1)}) + F'(x^{(k)})d_2^{(k)} \parallel < \eta_k \parallel F(x_*^{(k+1)}) \parallel \tag{9}$$

Set $x^{(k+1)} = x_*^{(k+1)} + d_2^{(k)}$.
End For

To present INHSS method for $l_k = 0, 1, 2, \ldots$ and $l'_k = 0, 1, 2, \ldots$, let us consider the following relations:

$$\begin{aligned} d^1_{k,l_k} &= (I - T_k^{l_k})(I - T_k)^{-1}M_k^{-1}F(z^{(k)}), \\ d^2_{k,l'_k} &= (I - T_k^{l'_k})(I - T_k)^{-1}M_k^{-1}F(z_*^{(k+1)}), \end{aligned} \tag{10}$$

where $\alpha$ is a positive constant, $T_k = T(\alpha; z^{(k)})$, $M_k = M(\alpha; z^{(k)})$, and

$$\begin{aligned} T(\alpha; z) &= M(\alpha; z)^{-1}N(\alpha; z), \\ M(\alpha; z) &= \frac{1}{2\alpha}(\alpha I + H(z))(\alpha I + S(z)), \\ N(\alpha; z) &= \frac{1}{2\alpha}(\alpha I - H(z))(\alpha I - S(z)), \end{aligned} \tag{11}$$

$H(z) = \frac{1}{2}(F'(z) + F'(z)^*)$ and $S(z) = \frac{1}{2}(F'(z) - F'(z)^*)$ are the Hermitian and skew-Hermitian parts of the Jacobian matrix $F'(z)$, respectively. So each iteration in INHSS method can be written as

$$z^{(k+1)} = z^{(k)} - (I - T_k^{l_k})F'(z^{(k)})^{-1}F(z^{(k)}) - (I - T_k^{l'_k})F'(z^{(k)})^{-1}F(z_*^{(k+1)}). \tag{12}$$

Algorithm 2 describes the steps of INHSS scheme. This method solves the nonlinear system $F(x) = 0$ with a positive definite Jacobian matrix. In Algorithms 1 and 2, $\eta_k$ denotes a constant in each step that is equal to $\eta$. The used norm in these algorithms is the Euclidean one.

In Algorithm 2, we firstly choose $d^1_{k,0} = 0$ and, for $l = 0, 1, 2, \ldots$ until relation (15) holds, we use HSS scheme to compute $d^1_{k,l_k}$ as follows

$$\begin{cases} (\alpha I + H(z^{(k)}))d^1_{k,l+\frac{1}{2}} &= (\alpha I - S(z^{(k)}))d^1_{k,l} - F(z^{(k)}), \\ (\alpha I + S(z^{(k)}))d^1_{k,l+1} &= (\alpha I - H(z^{(k)}))d^1_{k,l+\frac{1}{2}} - F(z^{(k)}), \end{cases} \tag{13}$$

where $l$ is the counter of the inner iterations, $d^1_{k,l_k}$ is the solution in $k$-th step of first outer iteration in INHSS scheme and $l_k$ is the number of HSS iterations which is necessary to satisfy (15). After obtaining a good approximation, we set $z_*^{(k+1)} = z^{(k)} + d^1_{k,l_k}$. Actually, $z_*^{(k+1)}$ is $(k+1)$-th intermediate approximation in the INHSS Algorithm. After this, similarly to the first outer iteration, we choose $d^2_{k,0} = 0$ and for $l' = 0, 1, 2, \ldots$ until (16) holds, we apply the HSS scheme as

$$\begin{cases} (\alpha I + H(z^{(k)}))d^2_{k,l'+\frac{1}{2}} &= (\alpha I - S(z^{(k)}))d^2_{k,l'} - F(z_*^{(k+1)}), \\ (\alpha I + S(z^{(k)}))d^2_{k,l'+1} &= (\alpha I - H(z^{(k)}))d^2_{k,l'+\frac{1}{2}} - F(z_*^{(k+1)}), \end{cases} \tag{14}$$

where $l'$ is the counter of inner iterations, $d^2_{k,l'_k}$ is the solution in $k-$th step of second outer iteration in INHSS scheme and $l'_k$ is the required number of HSS iterations to satisfy (16). Finally, we set $z^{(k+1)} = z_*^{(k+1)} + d^2_{k,l'_k}$, so $z^{(k+1)}$ is the $(k+1)$-th approximation which is obtained by the INHSS method.

---

**Algorithm 2:** INHSS Algorithm

---

**Input**: $z^{(0)}$, tol, $\alpha$ and positive integer sequences $\{l_k\}_{k=0}^{\infty}$, $\{l'_k\}_{k=0}^{\infty}$.
For $k = 1, 2, \ldots$ until $\|F(z^{(k)})\| \leq \text{tol} \parallel F(z^{(0)}) \parallel$ do:

**1** Set $d^1_{k,0} = 0$.

**2** For $l = 0, 1, 2, \ldots, l_k - 1$ until

$$\parallel F(z^{(k)}) + F'(z^{(k)})d^1_{k,l_k} \parallel < \eta_k \parallel F(z^{(k)}) \parallel \tag{15}$$

apply the HSS algorithm:

$$(\alpha I + H(z^{(k)}))d^1_{k,l+\frac{1}{2}} = (\alpha I - S(z^{(k)}))d^1_{k,l} - F(z^{(k)})$$
$$(\alpha I + S(z^{(k)}))d^1_{k,l+1} = (\alpha I - H(z^{(k)}))d^1_{k,l+\frac{1}{2}} - F(z^{(k)})$$

**3** Set $z_*^{(k+1)} = z^{(k)} + d^1_{k,l_k}$.

**4** Set $d^2_{k,0} = 0$.

**5** For $l' = 0, 1, 2, \ldots, l'_k - 1$ until

$$\parallel F(z_*^{(k+1)}) + F'(z^{(k)})d^2_{k,l'_k} \parallel < \eta_k \|F(z_*^{(k+1)})\| \tag{16}$$

apply the HSS algorithm:

$$(\alpha I + H(z^{(k)}))d^2_{k,l'+\frac{1}{2}} = (\alpha I - S(z^{(k)}))d^2_{k,l'} - F(z_*^{(k+1)})$$
$$(\alpha I + S(z^{(k)}))d^2_{k,l'+1} = (\alpha I - H(z^{(k)}))d^2_{k,l'+\frac{1}{2}} - F(z_*^{(k+1)})$$

**6** Set $z^{(k+1)} = z_*^{(k+1)} + d^2_{k,l'_k}$.
End For

---

To improve our method, we combine it with a Jacobian-free scheme. Knoll and Keyes [17] proposed a Newton-Krylov Jacobian free (JFNK) algorithm based on the generalized minimal residual method (GMRES). By applying a similar procedure based on Hermitian and skew-Hermitian splitting, we construct a Jacobian-free INHSS scheme. As applying of inexact version of the Newton method does not require the exact solution of the linear system in each iteration, we can approximate the Jacobian operator by

$$F'(x)v \approx \frac{F(x + \varepsilon v) - F(x - \varepsilon v)}{2\varepsilon}. \tag{17}$$

Equation (17) is a second order approximation to the Jacobian matrix $F'(x)$ acting on a vector $v$. If each individual component $J_{i,j}$ of the Jacobian matrix $F'$ is approximated as

$$J_{i,j} \approx \frac{F_i(v + \varepsilon e_j) - F_i(v - \varepsilon e_j)}{2\varepsilon},$$

where $e_j$ is the unit vector and $F_i$ is the coordinate function of $F$, so we can estimate the $j$-th column of the Jacobian matrix $F'$ as

$$\begin{bmatrix} J_{1,j} \\ J_{2,j} \\ \vdots \\ J_{n,j} \end{bmatrix} = \frac{F(x_0 + \varepsilon e_j) - F(x_0 - \varepsilon e_j)}{2\varepsilon}. \tag{18}$$

If one uses this approach, any column of $F'$ can be estimated without any direct computation of $F'$. This means that we have a Jacobian-free method.

## 3. Semilocal convergence of the INHSS method

In this section, we present the semilocal convergence of INHSS method. Guo and Duff [18] proved a Kantorovich-type convergence theorem for the Newton-HSS method. In this part we extend these results for INHSS scheme. First, we need some preliminaries and assumptions. Hence, to obtain convergence results for the INHSS method first consider the Kantorovich-type convergence Theorem for the Newton-HSS scheme as follows.

### *3.1. Assumptions*

Let $x^{(0)} \in \mathbb{C}^n$ and $F : D \subset \mathbb{C}^n \to \mathbb{C}^n$ be a $G$-differentiable function on an open set $N_0 \subset D$ on which $F'(x)$ is continuous and positive definite. Suppose that $F'(x) = H(x) + S(x)$, where $H(x) = \frac{1}{2}(F'(x) + F'(x)^*)$ and $S(x) = \frac{1}{2}(F'(x) - F'(x)^*)$ are the Hermitian and skew-Hermitian parts of the Jacobian matrix $F'(x)$, respectively. In addition, let us assume that the following conditions hold.

(C1) (**Bounded condition**) There exist positive constants $\beta$, $\gamma$ and $\delta$ such that

$$\max\{\|H(x^{(0)})\|, \|S(x^{(0)})\|\} \le \beta, \qquad \|F'(x^{(0)})^{-1}\| \le \gamma, \;\; \|F(x^{(0)})\| \le \delta. \tag{19}$$

(C2) (**Lipschitz condition**) There exist nonnegative constants $L_h$ and $L_s$ such that for all $x, y \in \mathcal{B}(x^{(0)}, r) \subset N_0$,

$$\begin{aligned} \|H(x) - H(y)\| &\le L_h \|x - y\|, \\ \|S(x) - S(y)\| &\le L_s \|x - y\|, \end{aligned} \tag{20}$$

where $\mathcal{B}(x, r) \equiv \{y : \| y - x \| < r\}$ shows an open ball with center $x$ and radius $r$.

From previous assumptions, $F'(x) = H(x) + S(x)$, $L = L_h + L_s$ and by applying Banach's Lemma, the next result holds.

**Lemma 1.** *[18] Under conditions (C1) and (C2), we have*

*1)* $\|F'(x) - F'(y)\| \le L\|x - y\|$,

*2)* $\|F'(x)\| \le L\|x - x^{(0)}\| + 2\beta$,

*3) If* $r \le 1/(\gamma L)$, *then* $F'(x)$ *is nonsingular and satisfies*

$$\|F'(x)^{-1}\| \le \frac{\gamma}{1 - \gamma L \|x - x^{(0)}\|}.$$

Therefore, the following semilocal convergence result is presented by Guo and Duff [18].

**Theorem 1.** *Let us assume that conditions (C1) and (C2) and Lemma 1 hold with the constants satisfying*

$$\delta\gamma^2 L \le \frac{1 - \eta}{2(1 + \eta^2)}, \tag{21}$$

*where* $\eta = \max_k\{\eta_k\} < 1$, $r = \min\{r_1, r_2\}$ *with*

$$\begin{aligned} r_1 &= \frac{\alpha + \beta}{L}\left(\sqrt{1 + \frac{2\alpha\tau\theta}{(2\gamma + \gamma\tau\theta)(\alpha + \beta)^2}} - 1\right), \\ r_2 &= \frac{b - \sqrt{b^2 - 2ac}}{a}, \\ a &= \frac{\gamma L(1 + \eta)}{1 + 2\gamma^2\delta L\eta}, \quad b = 1 - \eta, \quad c = 2\gamma\delta \end{aligned} \tag{22}$$

*and with $l_* = \mathrm{liminf}_{k\to\infty} l_k$ satisfying $l_* > \lfloor \ln(\eta)/\ln((\tau+1)\theta)\rfloor$, $\tau \in (0, (1-\theta)/\theta)$ and*

$$\theta \equiv \theta(\alpha; x^{(0)}) = \|T(\alpha; x^{(0)})\|, \tag{23}$$

*where $\lfloor u \rfloor$ shows the largest integer lower than or equal to $u$.*

*Then, the iteration sequence $\{x^{(k)}\}_{k=0}^{\infty}$ generated by NHSS algorithm is well-defined and converges to $x^*$, satisfying $F(x^*) = 0$.*

Now, we state and prove the extension of this result for INHSS method. We consider $r_2$ as in Theorem 1 and following the notations in [18], we introduce

$$\begin{aligned} &t_0 = 0, \\ &t_{k+1} = t_k - \frac{g(t_k)}{h(t_k)}, \quad k = 0, 1, \ldots \end{aligned} \tag{24}$$

where $g(t) = \frac{1}{2}at^2 - bt + c$ and $h(t) = at - 1$.

Authors in [18] proved that sequence (24) converges to $r_2$ monotone increasingly and $h(t_k) \le 0$. Therefore, we have $t_k < t_{k+1} < r_2$ and $t_k \to t_*(= r_2)$. The following result shows some properties of sequence $\{t_k\}$.

**Lemma 2.** *Sequence (24) for positive integers $m$ and $n$, $(m > n)$ satisfies the following relations:*

$$g(t_m) - g(t_n) - h(t_n)(t_m - t_n) = \frac{1}{2}a(t_m - t_n)^2 + \eta(t_m - t_n), \tag{25}$$

$$\frac{1}{-h(t_m)}(g(t_m) - g(t_n) - h(t_n)(t_m - t_n)) \le (t_{m+1} - t_{n+1}). \tag{26}$$

**Proof.** Since $g(t) = \frac{1}{2}at^2 - bt + c$, $h(t) = at - 1$ and $b = 1 - \eta$,

$$\begin{aligned} g(t_m) - g(t_n) - h(t_n)(t_m - t_n) &= \\ &= (\frac{1}{2}at_m^2 - bt_m + c) - (\frac{1}{2}at_n^2 - bt_n + c) - (at_n - 1)(t_m - t_n) \\ &= \frac{1}{2}a(t_m - t_n)^2 - b(t_m - t_n) + (t_m - t_n) = \frac{1}{2}a(t_m - t_n)^2 + \eta(t_m - t_n), \end{aligned}$$

hence relation (25) is obtained. To prove (26), we take into account that $\{t_k\}$ is an increasing sequence, $h(t)$ is an increasing function, $h(t_n) \le h(t_m)$ and

$$\begin{aligned} &\frac{1}{-h(t_m)}(g(t_m) - g(t_n) - h(t_n)(t_m - t_n)) \\ &\qquad \le (-\frac{g(t_m)}{h(t_m)} + t_m) + (\frac{g(t_n)}{h(t_n)} - t_n) = (t_{m+1} - t_{n+1}). \end{aligned}$$

This completes the proof. □

Now, from this lemma we present the following result to show the semilocal convergence of the INHSS method.

**Theorem 2.** *Let us suppose that the tolerance in INHSS algorithm is lower than $\frac{1}{8}\eta$, where $\eta$ is defined as in Theorem 1. Assume condition (C2) holds for constants defined in Theorem 1 and condition (C1) is changed by*

$$\max\{\|H(z^{(0)})\|, \|S(z^{(0)})\|\} \le \beta, \quad \|F'(z^{(0)})^{-1}\| \le \gamma', \quad \|F(z^{(0)})\| \le \frac{\delta}{4}, \tag{27}$$

*for an initial guess $z^{(0)}$. Moreover, $l_* = \min\{\mathrm{liminf}_{k\to\infty} l_k,\ \mathrm{liminf}_{k\to\infty} l'_k\}$, satisfying $l_* > \lfloor \frac{\ln\eta}{\ln(\tau+1)\theta} \rfloor$, $\tau \in \left(0, \dfrac{1-\theta}{\theta}\right)$ and*

$$\theta = \theta(\alpha; z^{(0)}) = \|T(\alpha; z^{(0)})\| < 1. \tag{28}$$

*Then, the iteration sequence $\{z^{(k)}\}_{k=0}^{\infty}$ generated by INHSS algorithm is well-defined and converges to $z^*$ satisfying $F(z^*) = 0$. Further, sequence $\{z^{(k)}\}_{k=0}^{\infty}$ hold the following relations*

$$\|z_*^{(1)} - z^{(0)}\| \le \frac{1}{4}(t_1 - t_0), \tag{29}$$

$$\|z_*^{(k)} - z^{(k-1)}\| \le \frac{1}{2^{k+3}}(t_{2k-1} - t_{k-1}), \qquad k = 2, 3, \ldots \tag{30}$$

*and also for $k = 1, 2, \ldots$, we have*

$$\|F(z_*^{(k)})\| \leq \frac{1}{2^{k+3}} \frac{1-\gamma L t_{2k-1}}{\gamma(1+\eta)}(t_{2k}-t_k), \tag{31}$$

$$\|z^{(k)} - z^{(k-1)}\| \leq \frac{1}{2^{k+2}}(t_{2k}-t_{k-1}), \tag{32}$$

$$\|F(z^{(k)})\| \leq \frac{1}{2^{k+2}} \frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+1}-t_k), \tag{33}$$

$$\|z^{(k)} - z^{(0)}\| \leq \frac{1}{2} r_2, \tag{34}$$

$$\|z_*^{(k)} - z^{(0)}\| \leq \frac{1}{4} r_2, \tag{35}$$

*where $\gamma = 4\gamma'$, $z_*^{(k+1)} = z^{(k)} - F'(z^{(k)})^{-1}F(z^{(k)})$, $r_2$ is defined as in Theorem 1 and $\{t_k\}$ is the sequence* (24).

**Proof.** Since $r = \min\{r_1, r_2\}$ and $F'(z) = M(\alpha; z) - N(\alpha; z)$, the proof of the relation

$$\|T(\alpha; z)\| < 1$$

is similar to the corresponding one of Theorem 1 for NHSS scheme in [18].

Now, we use mathematical induction to prove relations (30)-(33). For $k = 1$, from relations (10) and item 3 of Algorithm 2, we have

$$z_*^{(k+1)} - z^{(k)} = d_{k,l_k}^1 = (I - T_k^{l_k})(I - T_k)^{-1}M_k^{-1}F(z^{(k)}), \tag{36}$$

which from (11), changes to

$$z_*^{(k+1)} = z^{(k)} - (I - T_k^{l_k})F'(z^{(k)})^{-1}F(z^{(k)}), \tag{37}$$

setting $k = 0$ in (37) and by relations (10)-(11) we have

$$\begin{aligned} \|z_*^{(1)} - z^{(0)}\| &\leq \|F'(z^{(0)})^{-1}F(z^{(0)})\| + \|T_0^{l_*}F'(z^{(0)})^{-1}F(z^{(0)})\|, \\ &\leq \frac{\gamma}{4}(1+\theta^{l_*})\frac{\delta}{4} \leq \frac{1}{4}(2\gamma\delta) = \frac{1}{4}c = \frac{1}{4}(t_1 - t_0), \end{aligned}$$

hence, equation (29) is obtained. Also,

$$\|F(z^{(0)})\| \leq \frac{\delta}{4} \leq \frac{1}{4}\frac{2\delta}{1+\eta} = \frac{1}{4}\frac{1-\gamma L t_0}{\gamma(1+\eta)}(t_1 - t_0).$$

By using the integral mean-value Theorem and Lemma 1 for $x, y \in \mathcal{B}(z^{(0)}, r)$, we obtain

$$\begin{aligned} &\|F(x) - F(y) - F'(y)(x-y)\| \\ &\qquad = \|\int_0^1 F'(y + t(x-y))(x-y)dt - F'(y)(x-y)\| \\ &\qquad \leq \int_0^1 \|F'(y+t(x-y)) - F'(y)\|\|x-y\|dt \\ &\qquad \leq \int_0^1 Lt\|x-y\|^2 dt = \frac{L}{2}\|x-y\|^2. \end{aligned} \tag{38}$$

Now, from (38) and since $z^{(0)}, z_*^{(1)} \in \mathcal{B}(z^{(0)}, r)$ and also from (15),

$$\begin{aligned} \|F(z_*^{(1)})\| &\leq \|F(z_*^{(1)}) - F(z^{(0)}) - F'(z^{(0)})(z_*^{(1)} - z^{(0)})\| + \|F(z^{(0)}) + F'(z^{(0)})(z_*^{(1)} - z^{(0)})\| \\ &\leq \frac{L}{2}\|z_*^{(1)} - z^{(0)}\|^2 + \frac{\eta}{8}\|F(z^{(0)})\| \\ &\leq \frac{L}{2}(\frac{1}{4}(t_1 - t_0))^2 + \frac{\eta}{8}(\frac{1}{4}\frac{1-\gamma L t_0}{\gamma(1+\eta)})(t_1 - t_0), \\ &\leq \frac{1}{16}\left(\frac{L}{2}(t_1-t_0)^2 + \eta(\frac{1-\gamma L t_0}{\gamma(1+\eta)})(t_1 - t_0)\right), \end{aligned} \tag{39}$$

then we have

$$\begin{aligned}
\frac{\gamma(1+\eta)}{1-\gamma L t_1}\|F(z_*^{(1)})\| &\leq \frac{1}{16}\frac{\gamma(1+\eta)}{1-\gamma L t_1}\left(\frac{L}{2}(t_1-t_0)^2+\eta\frac{1-\gamma L t_0}{\gamma(1+\eta)}(t_1-t_0)\right)\\
&= \frac{1}{16}\left(\frac{1}{2}\frac{(1+\eta)\gamma L}{1-\gamma L t_1}(t_1-t_0)^2+\frac{\eta}{1-\gamma L t_1}(t_1-t_0)\right)\\
&\leq \frac{1}{16}\left(\frac{1}{2}\frac{a}{-h(t_1)}(t_1-t_0)^2+\frac{\eta}{-h(t_1)}(t_1-t_0)\right).
\end{aligned}$$

Since $\delta \leq \frac{1}{(2\gamma^2 L)}$, we have $1-\gamma L t_1 \geq -h(t_1)$ and from $t_k \geq t_1 = 2\gamma\delta$, so $\frac{(1+\eta)\gamma L}{(1-\gamma L t_1)} \leq \frac{a}{-h(t_1)}$, hence the last inequality is correct.

Now, from

$$g(t_1)-g(t_0)-h(t_0)(t_1-t_0) = \frac{1}{2}a(t_1-t_0)^2+\eta(t_1-t_0)$$

and (26), we obtain

$$\frac{\gamma(1+\eta)}{1-\gamma L t_1}\|F(z_*^{(1)})\| \leq \frac{1}{16}\frac{1}{-h(t_1)}\left(g(t_1)-g(t_0)-h(t_0)(t_1-t_0)\right) \leq \frac{1}{16}(t_2-t_1),$$

and, therefore,

$$\|F(z_*^{(1)})\| \leq \frac{1}{16}\frac{1-\gamma L t_1}{\gamma(1+\eta)}(t_2-t_1),$$

hence relation (31) holds for $k=1$. Also, we have

$$\begin{aligned}
\|z^{(1)}-z^{(0)}\| &\leq \|(I-T_0^{l_*})F'(z^{(0)})^{-1}F(z^{(0)})+(I-T_0^{l'_*})F'(z^{(0)})^{-1}F(z_*^{(1)})\|\\
&\leq \|F'(z^{(0)})^{-1}\|\left(\|(I-T_0^{l_*})\|\|F(z^{(0)})\|+\|(I-T_0^{l'_*})\|F(z_*^{(1)})\|\right)\\
&\leq (1+\eta)\frac{\gamma/4}{1-\gamma L t_0}\left(\frac{1}{4}\frac{1-\gamma L t_0}{\gamma(1+\eta)}(t_1-t_0)+\frac{1}{16}\frac{1-\gamma L t_1}{\gamma(1+\eta)}(t_2-t_1)\right)\\
&\leq \frac{1}{8}\left((t_1-t_0)+(t_2-t_1)\right) = \frac{1}{8}(t_2-t_0).
\end{aligned}$$

So, last inequality is correct since $1-\gamma L t_1 \leq 1-\gamma L t_0$. Thus, relation (32) holds for $k=1$.

Again, using (38) for $F(z^{(1)})$, since $z^{(1)}-z^{(0)} = d^1_{1,l_1}+d^2_{1,l'_1}$ and inequalities (15) and (16) hold, then

$$\begin{aligned}
\|F(z^{(1)})\| &\leq \|F(z^{(1)})-F(z^{(0)})-F'(z^{(0)})(z^{(1)}-z^{(0)})\|\\
&\quad +\|F(z^{(0)})+F(z_*^{(1)})+F'(z^{(0)})(z^{(1)}-z^{(0)})\|+\|F(z_*^{(1)})\|\\
&\leq \frac{L}{2}\|z^{(1)}-z^{(0)}\|^2+\|F(z^{(0)})+F'(z^{(0)})d^1_{1,l_1}\|\\
&\quad +\|F(z_*^{(1)})+F'(z^{(0)})d^2_{1,l'_1}\|+\|F(z_*^{(1)})\|\\
&\leq \frac{L}{2}\|z^{(1)}-z^{(0)}\|^2+\frac{\eta}{8}\|F(z^{(0)})\|+\frac{\eta}{8}\|F(z_*^{(1)})\|+\|F(z_*^{(1)})\|.
\end{aligned}$$

By multiplying both sides by $\frac{\gamma(1+\eta)}{1-\gamma L t_2}$ and by using upper bounds (31) and (39) for the third and last term, respectively, in the right hand side of the above inequality, we obtain

$$\begin{aligned}
&\frac{\gamma(1+\eta)}{1-\gamma L t_2}\|F(\mathbf{z}^{(1)})\|\\
&\leq \frac{\gamma(1+\eta)}{1-\gamma L t_2}\left(\frac{L}{2}(\frac{1}{64}(t_2-t_0)^2)+\frac{1}{32}\eta\frac{1-\gamma L t_0}{\gamma(1+\eta)}(t_1-t_0)+\frac{1}{16}\frac{\eta}{8}\frac{1-\gamma L t_1}{\gamma(1+\eta)}(t_2-t_1)\right)\\
&\quad +\frac{\gamma(1+\eta)}{1-\gamma L t_2}\left(\frac{L}{2}(\frac{1}{4}(t_1-t_0))^2+\frac{\eta}{8}(\frac{1}{4}\frac{1-\gamma L t_1}{\gamma(1+\eta)})(t_1-t_0)\right)\\
&\leq \frac{1}{8}\left(\frac{L}{2}\frac{\gamma(1+\eta)}{1-\gamma L t_2}(t_2-t_0)^2+\frac{\eta}{1-\gamma L t_2}(t_2-t_0)\right)\\
&\leq \frac{1}{8}\left(\frac{1}{2}\frac{a}{-h(t_2)}(t_2-t_0)^2+\frac{\eta}{-h(t_2)}(t_2-t_0)\right).
\end{aligned}$$

By similar calculations, we get

$$\frac{\gamma(1+\eta)}{1-\gamma L t_2}\|F(\mathbf{z}_1)\| \le \frac{1}{8}\left(\frac{1}{-h(t_2)}(g(t_2)-g(t_0)-h(t_0)(t_2-t_0))\right) \le \frac{1}{8}(t_3-t_1).$$

So relation (33) for $k=1$ is obtained.

Suppose that relations (30)-(33) hold for an arbitrary $k$, now we prove these relations for $k+1$. Since

$$\|z_*^{(k+1)}-z^{(k)}\| = \|F'(z^{(k)})^{-1}F(z^{(k)}) + T_k^{l_*}F'(z^{(k)})^{-1}F(z^{(k)})\|,$$

as

$$\begin{aligned}
\|z^{(k)}-z^{(k-1)}\| &\le \frac{1}{2^{k+2}}(t_{2k}-t_{k-1}) \le (t_{2k}-t_0) = \frac{1}{2^{k+2}}t_{2k},\\
\|z^{(k-1)}-z^{(k-2)}\| &\le \frac{1}{2^{k+1}}(t_{2k-2}-t_{k-2}) \le (t_{2k-2}-t_0) = \frac{1}{2^{k+1}}t_{2k},\\
&\vdots\\
\|z^{(1)}-z^{(0)}\| &\le \frac{1}{2^{2}}(t_2-t_0) \le \frac{1}{2^{k+1}}(t_{2k-2}-t_{k-2}) = \frac{1}{2^{2}}t_{2k},
\end{aligned}$$

thus

$$\|z^{(k)}-z^{(0)}\| \le t_{2k}.$$

Also, by Lemma 1, and since $r \le 1/(\gamma L) \le 1/(\frac{\gamma}{4}L)$, we have

$$\|F'(z^{(k)})^{-1}\| \le \frac{\gamma/4}{1-\gamma/4L\|z^{(k)}-z^{(0)}\|} \le \frac{\gamma/4}{1-\gamma L t_{2k}}$$

and

$$\begin{aligned}
&\|z_*^{(k+1)}-z^{(k)}\| \le (1+((\tau+1)\theta)^{l_*})\frac{\gamma/4}{1-\gamma L t_{2k}}\|F(z^{(k)})\|\\
&\le (1+\eta)\frac{\gamma/4}{1-\gamma L t_{2k}}\left(\frac{1}{2^{k+2}}\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+1}-t_k)\right) = \frac{1}{2^{(k+1)+3}}(t_{2k+1}-t_k).
\end{aligned}$$

This is relation (30) for $k+1$.

By using (38) and Lemma 1, as $z_*^{(k+1)}, z^{(k)} \in \mathcal{B}(z^{(0)}, r)$, we have

$$\begin{aligned}
\|F(z_*^{(k+1)})\| &\le \|F(z_*^{(k+1)}) - F(z^{(k)}) - F'(z^{(k)})(z_*^{(k+1)}-z^{(k)})\|\\
&\quad + \|F(z^{(k)}) + F'(z^{(k)})(z_*^{(k+1)}-z^{(k)})\|\\
&\le \frac{L}{2}\|z_*^{(k+1)}-z^{(k)}\|^2 + \frac{\eta}{8}\|F(z^{(k)})\|.
\end{aligned}$$

By using induction hypothesis, we get

$$\begin{aligned}
&\frac{\gamma(1+\eta)}{1-\gamma L t_{2k+1}}\|F(z_*^{(k+1)})\|\\
&\le \frac{\gamma(1+\eta)}{1-\gamma L t_{2k+1}}\left(\frac{L}{2}\left(\frac{1}{2^{(k+1)+3}}(t_{2k+1}-t_k)\right)^2 + \frac{\eta}{8}\left(\frac{1}{2^{k+2}}\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+1}-t_k)\right)\right)\\
&\le \frac{1}{2^{(k+1)+3}}\left(\frac{1}{2}\frac{a}{-h(t_{2k+1})}(t_{2k+1}-t_k)^2 + \frac{\eta}{-h(t_{2k+1})}(t_{2k+1}-t_k)\right).
\end{aligned}$$

With similar computations as for $k=1$, we obtain

$$\begin{aligned}
\frac{\gamma(1+\eta)}{1-\gamma L t_{2k+1}}\|F(z_*^{(k+1)})\| &\le \frac{1}{2^{(k+1)+3}}\frac{1}{-h(t_{2k+1})}\left(g(t_{2k+1})-g(t_k)-h(t_k)(t_{2k+1}-t_k)\right)\\
&\le \frac{1}{2^{(k+1)+3}}(t_{2k+2}-t_{k+1}),
\end{aligned}$$

which is relation (31) for $k+1$. Finally,

$$\begin{aligned}
&\|z^{(k+1)} - z^{(k)}\| = \|(I - T_k^{l_*})F'(z^{(k)})^{-1}F(z^{(k)}) + (I - T_k^{l_*})F'(z^{(k)})^{-1}F(z_*^{(k+1)})\| \\
&\leq (1+\eta)\frac{\gamma/4}{1-\gamma L t_{2k}}\left(\frac{1}{2^{k+2}}\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+1}-t_k) + \frac{1}{2^{(k+1)+3}}\frac{1-\gamma L t_{2k+1}}{\gamma(1+\eta)}(t_{2k+2}-t_{k+1})\right) \\
&\leq \frac{1}{2^{(k+1)+3}}(t_{2k+2}-t_k).
\end{aligned}$$

This is relation (32) for $k+1$.

To prove inequality (33), we use again (38) and Lemma 1

$$\begin{aligned}
\|F(z^{(k+1)})\| \leq\ & \|F(z^{(k+1)}) - F(z^{(k)}) - F'(z^{(k)})(z^{(k+1)} - z^{(k)})\| \\
& + \|F(z^{(k)}) + F(z_*^{(k+1)}) + F'(z^{(k)})(z^{(k+1)} - z^{(k)})\| + \|F(z_*^{(k+1)})\| \\
\leq\ & \frac{L}{2}\|z^{(k+1)} - z^{(k)}\|^2 + \|F(z^{(k)}) + F'(z^{(k)})d^1_{k,l_k}\| \\
& + \|F(z_*^{(k+1)}) + F'(z^{(k)})d^2_{k,l'_k}\| + \|F(z_*^{(k+1)})\| \\
\leq\ & \frac{L}{2}\|z^{(k+1)} - z^{(k)}\|^2 + \frac{\eta}{8}\|F(z^{(k)})\| + \frac{\eta}{8}\|F(z_*^{(k+1)})\| + \|F(z_*^{(k+1)})\|.
\end{aligned}$$

As $z^{(k+1)} - z^{(k)} = d^1_{k,l_k} + d^2_{k,l'_k}$, by using

$$\|F(z_*^{(k+1)})\| \leq \frac{L}{2}\|z_*^{(k+1)} - z^{(k)}\|^2 + \frac{\eta}{8}\|F(z^{(k)})\|,$$

and applying (38) yields

$$\begin{aligned}
\|F(z^{(k+1)})\| \leq\ & \frac{L}{2}\left(\|z^{(k+1)} - z^{(k)}\|^2 + \|z_*^{(k+1)} - z^{(k)}\|^2\right) \\
& + \left(\frac{\eta}{4}\|F(z^{(k)})\| + \frac{\eta}{8}\|F(z_*^{(k+1)})\|\right) + \|F(z_*^{(k+1)})\| \\
\leq\ & \frac{L}{2}\left((\frac{1}{2^{(k+1)+2}}(t_{2k+2}-t_k))^2 + (\frac{1}{2^{(k+1)+3}}(t_{2k+1}-t_{2k}))^2\right) \\
& + \frac{\eta}{4}\frac{1}{2^{k+2}}\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+1}-t_k) \\
& + \frac{\eta}{8}\left(\frac{1}{2^{(k+1)+3}}\frac{1-\gamma L t_{2k+1}}{\gamma(1+\eta)}(t_{2k+2}-t_{k+1})\right) \\
\leq\ & \frac{1}{2^{k+3}}\left(\frac{L}{2}(t_{2k+2}-t_k)^2 + \eta\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+2}-t_k)\right).
\end{aligned}$$

Again, we can write

$$\begin{aligned}
&\frac{\gamma(1+\eta)}{1-\gamma L t_{2k+2}}\|F(z^{(k+1)})\| \\
&\qquad \leq \frac{1}{2^{k+3}}\frac{\gamma(1+\eta)}{1-\gamma L t_{2k+2}}\left(\frac{L}{2}(t_{2k+2}-t_k)^2 + \eta\frac{1-\gamma L t_{2k}}{\gamma(1+\eta)}(t_{2k+2}-t_k)\right) \\
&\qquad \leq \frac{1}{2^{k+3}}\left(\frac{L}{2}\frac{(1+\eta)\gamma L}{1-\gamma L t_{2k+2}}(t_{2k+2}-t_k)^2 + \frac{\eta}{1-\gamma L t_{2k+2}}(t_{2k+2}-t_k)\right) \\
&\qquad \leq \frac{1}{2^{k+3}}\left(\frac{1}{2}\frac{a}{-h(t_{2k+2})}(t_{2k+2}-t_k)^2 + \frac{\eta}{-h(t_{2k+2})}(t_{2k+2}-t_k)\right).
\end{aligned}$$

Now, in a similar way as our previous computations, we have

$$g(t_{2k+2}) - g(t_k)) - h(t_k)(t_{2k+2}-t_k) = \frac{1}{2}a(t_{2k+2}-t_k)^2 + \eta(t_{2k+2}-t_k)$$

and so

$$\begin{aligned}
&\frac{\gamma(1+\eta)}{1-\gamma L t_{2k+2}}\|F(z^{(k+1)})\| \\
&\qquad \leq \frac{1}{2^{k+3}}\left(\frac{1}{-h(t_{2k+2})}(g(t_{2k+2}) - g(t_k)) - h(t_k)(t_{2k+2}-t_k)\right) \\
&\qquad \leq \frac{1}{2^{(k+1)+2}}(t_{2k+3}-t_{k+1}).
\end{aligned}$$

So, relation (33) for $k+1$ is obtained.
Note that if relations (29)-(33) hold, then

$$\begin{aligned}
&\|z^{(k)} - z^{(0)}\| \\
&\quad \leq \|z^{(k)} - z^{(k-1)}\| + \|z^{(k-1)} - z^{(k-2)}\| + \cdots + \|z^{(2)} - z^{(1)} + \|z^{(1)} - z^{(0)}\| \\
&\quad \leq \frac{1}{2^{k+2}}(t_{2k} - t_{k-1}) + \frac{1}{2^{k+1}}(t_{2k-2} - t_{k-2}) + \cdots + \frac{1}{8}(t_4 - t_1) + \frac{1}{4}(t_2 - t_0) \\
&\quad \leq \frac{1}{4}(\frac{1}{2^k}r_2 + \frac{1}{2^{k-1}}r_2 + \cdots + \frac{1}{2}r_2 + r_2).
\end{aligned}$$

By simplifying the last inequality, we get

$$\|z^{(k)} - z^{(0)}\| \leq \frac{1}{2}r_2,$$

by similar computations as for $\{z_*^{(k)}\}$, we obtain

$$\|z_*^{(k)} - z^{(0)}\| \leq \frac{1}{4}r_2$$

and these yield inequalities (34) and (35).
If $z \in \mathcal{B}(z^{(0)}, r)$ then, we have the following bounded for iterative matrix $T(\alpha; z)$

$$\|T(\alpha; z)\| \leq (\tau + 1)\theta < 1.$$

Since sequence $\{t_k\}$ converges to $t_* = r_2$, so the sequence $\{z^{(k)}\}$ converges to its limit, say $z^*$. Because $T(\alpha; z^*) < 1$, from (12), we have

$$F(z^*) = 0.$$

This completes the proof. □

## 4. Numerical results

Consider the following two-dimensional nonlinear convection-diffusion equation

$$\begin{aligned}
-(u_{xx} + u_{yy}) + q(u_x + u_y) = -e^u - \sin(1 + u_x + u_y), \quad & (x, y) \in \Omega \\
u(x, y) = 0, \quad & (x, y) \in \partial\Omega
\end{aligned}$$

where $\Omega = (0,1) \times (0,1)$, $\partial\Omega$ is its boundary and $q$ is a positive constant for measuring the magnitude of the convection term. Applying a five-point finite difference scheme to the diffusive term and the central difference scheme to the convective term, a system of nonlinear equations is obtained as

$$F(u) = Mu + h^2\psi(u), \tag{40}$$

where $h = \dfrac{1}{N+1}$ is the equidistant step-size with $N$ as a prescribed positive integer, $M = A_N \otimes I_N + A_N \otimes I_N$, $B = C_N \otimes C_N$ and $\psi(u) = \sin(1 + Bu) + \varphi(u)$, with tridiagonal matrices $A_N = \text{tridiag}(-1 - qh/2, 2, 1 + qh/2)$, $C_N = \text{tridiag}(-1/h, 0, 1/h)$ and $\varphi(u) = (e^{u_1}, e^{u_2}, ..., e^{u_n})^T$, $\otimes$ denotes the Kronecker product, $n = N \times N$ and $\sin(u)$ means $(\sin(u_1), \sin(u_2), \ldots, \sin(u_n))^T$.

The stopping criterion for the outer iteration in Jacobian free NHSS, Jacobian free INHSS, Nonlinear HSS-like and Picard-HSS are set to be $\dfrac{\| F(u^{(k)}) \|}{\| F(u^{(0)}) \|} \leq 10^{-11}$. The stopping criterion for the inner iteration in Newton-HSS and Picard-HSS iteration is

$$\|F(u^{(k)}) + F'(u^{(k)})s_{n_k}^{(k)}\| \leq \eta\|F(u^{(k)})\|. \tag{41}$$

$\{u^{(k)}\}$ is the sequence generated by NHSS and Picard-HSS method, $s_n^{(k)}$ is the $n$-th HSS inner iteration in the $k$-th step of NHSS and Picard-HSS method and $n_k$ is the number of HSS inner iterations which needs to satisfy relation (41). Also, stopping criterions for the inner iterations in INHSS algorithm are

$$\|F(v^{(k)}) + F'(v^{(k)})d_{k,\ell_k}^1\| \leq \eta\|F(v^{(k)})\|$$

and

$$\|F(v_*^{(k+1)}) + F'(v^{(k)})d^2_{k,\ell'_k}\| \leq \eta\|F(v_*^{(k+1)})\|,$$

where $v_*^{(k+1)} = v^{(k)} + d^1_{k,\ell_k}$ and $\{v^{(k)}\}$ is the sequence generated by INHSS scheme.

Numerical results for Jacobian free INHSS, Jacobian free NHSS, nonlinear HSS-like and Picard-HSS schemes in terms of total CPU-time (denoted by CPU), the outer and inner iteration steps (respectively denoted as $IT_{out}$ and $IT_{int}$) are presented in Tables 1 and 2. $IT$ shows the total inner iterations and $IT_{int}$ is the average of total inner iterations. $\|F(x^{(n)})\|$ denoted the norm of the function in the last iteration. Since Jacobian free INHSS algorithm contains two inner iterations, namely relations (15) and (16), hence we have reported these iterations by $IT1_{int}$ and $IT2_{int}$, so $IT1_{int}$ and $IT2_{int}$ are respectively the average of inner iterations (15) and (16) .

In Table 1 numerical results are listed for $q = 100, 1000, 2000$, different values of $N$, $\eta = 0.1$ and initial guess $u^{(0)} = \mathbf{e} = (1, 1, \ldots, 1)^T$. One can see that all the methods can perform the iterations, but for $q = 1000$ and $q = 2000$, Picard-HSS is not successful to solve this problem. Increasing $q$ give us an ill-posed matrix at each iteration, so Picard-HSS, nonlinear HSS-like, Jacobian free INHSS and Jacobian free NHSS need more inner iterations. Picard-HSS even for $q = 100$ needs many inner iterations.

Note that in the outer iterations one must compute the Jacobian matrix. Hence the most consumable time is in this part. So, by using Jacobian-free NHSS or Jacobian-free INHSS, the results can be improved impressively in terms of CPU-time. When parallel computing is applied to approximate the Jacobian matrix by divided difference (18), the time that is needed for computing this matrix, compare with the total time is negligible. Computing the real Jacobian matrix usually fails for large number $N$ (In our computations it fails for $N > 50$).

Table 2 shows the numerical results for $q = 1000$ and initial guess $u^{(0)} = 4.5\mathbf{e}$ and $u^{(0)} = 13\mathbf{e}$. For $u^{(0)} = 4.5\mathbf{e}$ that is relatively far from the solution (the real solution is near zero), only for $N = 80$ and $N = 100$ nonlinear HSS-like method was successful to solve the problem, and for initial guess $u^{(0)} = 13\mathbf{e}$ it couldn't perform the iterations at all. But Jacobian free INHSS and Jacobian free NHSS in all cases could easy solve the problem. Picard-HSS for both initial guess could not solve the problem.

These results show the efficiency of Jacobian free INHSS and Jacobian free NHSS with respect to Picard-HSS and nonlinear HSS-like especially when the problem is unstable or the initial guess is far away from the solution of the problem. Picard-HSS and nonlinear HSS-like are more suitable to solve weakly nonlinear problems but by starting from a very far point, the linear term is not strongly dominant over the nonlinear term at the starting point, so they cannot perform the iterations successfully. From these tables we can see the number of outer iterations in Jacobian free INHSS method are less than or equal to half of outer iterations in Jacobian free NHSS scheme. This shows that our new algorithm can reduce the number of computations of the Jacobian matrix as well, because the convergent rate of INHSS method is faster than that in the NHSS algorithm.

Determining the optimal value for parameter $\alpha$ that minimizes the number of iteration matrices is important, because it improves the convergence speed of these methods. However, determining the optimal value at each step of NHSS and INHSS schemes is impossible. Since they are stable methods, we only use the optimal $\alpha$ near the solution (for this example at point $u = \mathbf{0}$). The optimal $\alpha$ in this example, when the positive definite matrix $M$ is used in Picard-HSS and nonlinear HSS-like methods, is almost in all cases very near to the optimal $\alpha$ of matrix $F'(\mathbf{0})$ in NHSS and INHSS schemes. We investigate the spectral properties of HSS inner iterations in the following part.

### *4.1.* ***Spectral radius***

In this subsection, we investigate the value of spectral radius for different values of experimental optimal parameter $\alpha$ for NHSS, INHSS, Picard-HSS and nonlinear HSS-like methods.

In Figure 1 we show the spectral radius of the iteration matrix $T(\alpha)$ and its upper bound $\sigma(\alpha)$ for HSS algorithm. In [1] authors proved that spectral radius of HSS inner iteration is bounded by $\|T\| \leq \sigma(\alpha) \equiv \max_{\lambda_i \in \lambda(H)} |\frac{\alpha - \lambda_i}{\alpha + \lambda_i}| < 1$ and the minimum of this bound is obtained when $\alpha = \alpha^* = \sqrt{\lambda_{min(H)}\lambda_{max(H)}}$, being $\lambda_{min(H)}$ and $\lambda_{max(H)}$ the smallest and largest eigenvalues of Hermitian matrix $H$, respectively. In Table 3, we have written the optimal value of parameter $\alpha_{opt}$ (tested and optimal $\alpha$) which has been determined experimentally and calculating the spectral radii of the iteration matrix $T(\alpha)$, for HSS algorithm. We have used approximating value of the Jacobian matrix to obtain optimal $\alpha$. These results show that HSS algorithm always is a convergent method. In HSS scheme, when $q$ or $qh/2$ are small, $\sigma(\alpha)$ is close to $\rho(T(\alpha))$. So, when $q$ or $qh/2$ are small, $\alpha^*$ is close to $\alpha_{opt}$ and $\alpha^*$ can be a good estimation for $\alpha_{opt}$. But, when $q$ or $qh/2$ are large (the skew-Hermitian part is dominant), $\sigma(\alpha)$ deviates from $\rho(T(\alpha))$ very much so, to use $\alpha^*$ is not useful (see [1]). In nonlinear HSS-like and Picard-HSS schemes, $\alpha_{opt}$ is, in almost all cases, very close to the $\alpha_{opt}$ of NHSS and INHSS methods for (40). We observe in Table 3 that when the value of $N$ increases, the spectral radii decrease in some cases. On the opposite, increasing $q$ also increase the spectral radii.

Table 1: Numerical results for $\eta = 0.1$ and $u^{(0)} = \mathbf{e}$.

| **N** | | | **30** | **40** | **60** | **70** | **80** | **100** |
|---|---|---|---|---|---|---|---|---|
| $q = 100$ | Jacobian free NHSS | $\alpha_{opt}$ | 3.8 | 3.1 | 2.3 | 2.0 | 1.7 | 1.3 |
| | | CPU | 0.49 | 1.66 | 9.24 | 21.19 | 38.08 | 116.43 |
| | | $IT_{out}$ | 10 | 10 | 10 | 12 | 11 | 10 |
| | | $IT_{inn}$ | 6.1 | 7.2 | 9.6 | 10.18 | 10.33 | 14.6 |
| | | $IT$ | 61 | 72 | 96 | 112 | 124 | 146 |
| | | $\|F(x^{(n)})\|$ | 1.43e-10 | 1.35e-10 | 9.78e-11 | 1.54e-11 | 1.96e-11 | 2.86e-11 |
| | Jacobian free INHSS | $\alpha_{opt}$ | 3.8 | 3.1 | 2.3 | 2.0 | 1.7 | 1.3 |
| | | CPU | 0.39 | 1.34 | 8.05 | 15.87 | 31.92 | 86.87 |
| | | $IT_{out}$ | 5 | 5 | 5 | 5 | 5 | 5 |
| | | $IT1_{inn}$ | 7.80 | 10 | 12.20 | 12.8 | 13 | 12.60 |
| | | $IT2_{inn}$ | 4.40 | 4.8 | 8.05 | 10.4 | 12.8 | 17.80 |
| | | $IT$ | 61 | 74 | 102 | 116 | 129 | 152 |
| | | $\|F(x^{(n)})\|$ | 1.37e-10 | 3.08e-11 | 1.83e-08 | 2.79e-10 | 2.66e-10 | 3.40e-10 |
| | Nonlinear HSS-like | $\alpha_{opt}$ | 3.8 | 3.1 | 2.2 | 2.0 | 1.7 | 1.3 |
| | | CPU | 0.37 | 0.99 | 5.20 | 10.92 | 19.66 | 52.90 |
| | | $IT$ | 53 | 59 | 69 | 85 | 92 | 106 |
| | | $\|F(x^{(n)})\|$ | 1.22e-10 | 1.46e-10 | 1.51e-10 | 1.25e-10 | 1.22e-10 | 1.91e-10 |
| | Picard-HSS | $\alpha_{opt}$ | 3.8 | 3.1 | 2.3 | 2.0 | 1.7 | 1.3 |
| | | CPU | 1.58 | 5.97 | 39.16 | 82.73 | 173.08 | 474.92 |
| | | $IT_{out}$ | 12 | 12 | 12 | 12 | 12 | 12 |
| | | $IT_{inn}$ | 37.41 | 42.75 | 59.34 | 68.92 | 85.34 | 106 |
| | | $IT$ | 449 | 513 | 712 | 827 | 1024 | 1272 |
| | | $\|F(x^{(n)})\|$ | 6.57e-11 | 9.13e-11 | 1.45e-10 | 1.36e-10 | 1.68e-10 | 1.85e-10 |
| $q = 1000$ | Jacobian free NHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 1.40 | 2.70 | 10.24 | 20.20 | 36.77 | 96.65 |
| | | $IT_{out}$ | 11 | 11 | 11 | 11 | 11 | 11 |
| | | $IT_{inn}$ | 10.46 | 10.64 | 10.18 | 11.09 | 11.45 | 12 |
| | | $IT$ | 124 | 117 | 112 | 122 | 126 | 132 |
| | | $\|F(x^{(n)})\|$ | 5.76e-10 | 3.92e-10 | 2.40e-10 | 3.23e-10 | 2.55e-10 | 2.20e-10 |
| | Jacobian free INHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 0.73 | 1.80 | 8.50 | 14.03 | 28.03 | 56.07 |
| | | $IT_{out}$ | 5 | 5 | 5 | 5 | 5 | 5 |
| | | $IT1_{inn}$ | 12 | 12.40 | 12.20 | 12.60 | 13.20 | 13.40 |
| | | $IT2_{inn}$ | 11.20 | 10.80 | 10.20 | 10.40 | 10.20 | 10 |
| | | $IT$ | 116 | 116 | 112 | 115 | 117 | 117 |
| | | $\|F(x^{(n)})\|$ | 5.04e-09 | 5.90e-09 | 3.12e-09 | 3.27e-09 | 4.35e-09 | 3.19e-09 |
| | Nonlinear HSS-like | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 0.83 | 2.05 | 8.60 | 15.12 | 25.33 | 58.93 |
| | | $IT$ | 121 | 121 | 116 | 119 | 121 | 120 |
| | | $\|F(x^{(n)})\|$ | 1.48e-09 | 1.29e-09 | 1.25e-09 | 8.81e-10 | 9.32e-10 | 7.31e-10 |
| | Picard-HSS | - | - | - | - | - | - | - |
| $q = 2000$ | Jacobian free NHSS | $\alpha_{opt}$ | 26 | 23 | 12 | 11 | 10 | 8 |
| | | CPU | 1.01 | 3.04 | 13.96 | 25.83 | 41.91 | 113.47 |
| | | $IT_{out}$ | 11 | 11 | 11 | 11 | 11 | 11 |
| | | $IT_{inn}$ | 17 | 16.36 | 15.09 | 14.36 | 14.45 | 14.45 |
| | | $IT$ | 187 | 180 | 166 | 158 | 159 | 159 |
| | | $\|F(x^{(n)})\|$ | 1.48e-09 | 1.23e-09 | 9.48e-10 | 8.00e-10 | 1.03e-09 | 7.82e-10 |
| | Jacobian free INHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 0.82 | 2.68 | 11.71 | 17.61 | 34.78 | 79.15 |
| | | $IT_{out}$ | 5 | 5 | 5 | 5 | 5 | 5 |
| | | $IT1_{inn}$ | 17.60 | 17 | 15.80 | 15.60 | 15.60 | 16 |
| | | $IT2_{inn}$ | 16.20 | 16.20 | 15 | 14 | 14.20 | 13.80 |
| | | $IT$ | 169 | 166 | 154 | 148 | 149 | 149 |
| | | $\|F(x^{(n)})\|$ | 1.68e-08 | 1.37e-08 | 9.36e-09 | 9.86e-09 | 1.12e-08 | 3.19e-09 |
| | Nonlinear HSS-like | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 1.19 | 2.84 | 11.62 | 19.77 | 33.20 | 77.20 |
| | | $IT$ | 174 | 163 | 156 | 155 | 156 | 156 |
| | | $\|F(x^{(n)})\|$ | 3.14e-09 | 2.88e-09 | 2.07e-09 | 2.36e-09 | 1.88e-09 | 1.78e-09 |
| | Picard-HSS | - | - | - | - | - | - | - |

### 4.2. *Performance profile*

In the previous parts we have shown that NHSS and INHSS methods perform better than Picard-HSS and Nonlinear HSS-like ones. In this section, to analyze the performance of NHSS and INHSS schemes and comparing them more precisely, we apply the "performance profile " which is proposed in [19] as an evaluation tool, see also [10, 20]. It is

Table 2: Numerical results for $q = 1000$ and $\eta = 0.1$.

| N | | | 30 | 40 | 60 | 70 | 80 | 100 |
|---|---|---|---|---|---|---|---|---|
| $u^{(0)} = 4.5\mathbf{e}$ | Jacobian free NHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 0.82 | 2.53 | 11.82 | 21.19 | 36.64 | 87.41 |
| | | $IT_{out}$ | 11 | 11 | 11 | 11 | 11 | 11 |
| | | $IT_{inn}$ | 11.18 | 11.36 | 10.81 | 10.63 | 11.91 | 11.82 |
| | | $IT$ | 123 | 125 | 119 | 117 | 131 | 130 |
| | | $\|F(x^{(n)})\|$ | 2.85e-09 | 1.79e-09 | 1.77e-09 | 9.09e-10 | 1.08e-09 | 1.21e-09 |
| | Jacobian free INHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 0.69 | 1.75 | 7.50 | 14.07 | 23.85 | 52.73 |
| | | $IT_{out}$ | 5 | 5 | 5 | 5 | 5 | 5 |
| | | $IT1_{inn}$ | 11.6 | 12.4 | 12 | 12.8 | 15 | 13 |
| | | $IT2_{inn}$ | 11.4 | 11 | 10.50 | 10.6 | 11.4 | 11.4 |
| | | $IT$ | 115 | 117 | 111 | 117 | 124 | 122 |
| | | $\|F(x^{(n)})\|$ | 2.42e-08 | 1.77e-08 | 1.83e-08 | 5.17e-09 | 1.50e-08 | 2.34e-09 |
| | Nonlinear HSS-like | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | - | - | - | - | 25.57 | 58.69 |
| | | $IT$ | - | - | - | - | 121 | 126 |
| | | $\|F(x^{(n)})\|$ | - | - | - | - | 4.19e-09 | 3.29e-09 |
| | Picard-HSS | - | - | - | - | - | - | - |
| $u^{(0)} = 13\mathbf{e}$ | Jacobian free NHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 1.45 | 3.47 | 14.24 | 27.86 | 45.99 | 108 |
| | | $IT_{out}$ | 17 | 17 | 16 | 16 | 17 | 17 |
| | | $IT_{inn}$ | 9.82 | 8.76 | 8.50 | 8.43 | 8.29 | 8.12 |
| | | $IT$ | 167 | 149 | 136 | 135 | 141 | 138 |
| | | $\|F(x^{(n)})\|$ | 4.97e-08 | 2.17e-08 | 5.52e-08 | 4.03e-08 | 4.84e-09 | 6.83e-09 |
| | Jacobian free INHSS | $\alpha_{opt}$ | 18 | 16 | 9 | 8.5 | 8 | 5.5 |
| | | CPU | 1.26 | 2.96 | 12.82 | 23.14 | 38.93 | 62.04 |
| | | $IT_{out}$ | 10 | 10 | 10 | 10 | 10 | 10 |
| | | $IT1_{inn}$ | 10 | 8.9 | 8.1 | 8 | 7.9 | 7.6 |
| | | $IT2_{inn}$ | 9.7 | 7.8 | 7.1 | 6.7 | 6.5 | 6.3 |
| | | $IT$ | 197 | 167 | 152 | 147 | 144 | 139 |
| | | $\|F(x^{(n)})\|$ | 1.74e-07 | 1.47e-07 | 1.24e-07 | 8.75e-08 | 3.29e-08 | 1.41e-08 |
| | Nonlinear HSS-like | - | - | - | - | - | - | - |
| | Picard-HSS | - | - | - | - | - | - | - |

Table 3: Numerical results for optimal $\alpha$ in NHSS and INHSS methods.

| | N | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| $q = 100$ | $\alpha_{opt}$ | 4.3 | 3.8 | 3.1 | 2.6 | 2.2 | 20 | 1.7 | 1.5 | 1.3 |
| | $\rho(T(\alpha_{opt}))$ | 0.4887 | 0.4672 | 0.4687 | 0.4480 | 0.4733 | 0.5089 | 0.5395 | 0.5815 | 0.6131 |
| | $\alpha^*$ | 0.3554 | 0.1638 | 0.0938 | 0.0606 | 0.0424 | 0.0313 | 0.0241 | 0.0191 | 0.0155 |
| | $\rho(T(\alpha^*))$ | 0.8920 | 0.9480 | 0.9695 | 0.9800 | 0.9858 | 0.9894 | 0.9918 | 0.9935 | 0.9947 |
| | $\frac{qh}{2}$ | 2.3810 | 1.6129 | 1.2195 | 0.9804 | 0.8197 | 0.7042 | 0.6173 | 0.5495 | 0.4950 |
| | $\rho(T(\frac{qh}{2}))$ | 0.6003 | 0.6653 | 0.7127 | 0.7498 | 0.7799 | 0.8038 | 0.8229 | 0.8384 | 0.8513 |
| $q = 1000$ | $\alpha_{opt}$ | 22 | 18 | 16 | 13 | 9 | 8.5 | 8 | 6.5 | 5.5 |
| | $\rho(T(\alpha_{opt}))$ | 0.7596 | 0.7223 | 0.6925 | 0.6738 | 0.6617 | 0.6526 | 0.6470 | 0.6483 | 0.6467 |
| | $\alpha^*$ | 0.5962 | 0.4047 | 0.3062 | 0.2462 | 0.2059 | 0.1769 | 0.1551 | 0.1381 | 0.1244 |
| | $\rho(T(\alpha^*))$ | 0.8518 | 0.8971 | 0.9211 | 0.9360 | 0.9461 | 0.9535 | 0.9590 | 0.9634 | 0.9669 |
| | $\frac{qh}{2}$ | 23.8095 | 16.1290 | 12.1951 | 9.8039 | 8.1967 | 7.0423 | 6.1728 | 5.4945 | 4.9505 |
| | $\rho(T(\frac{qh}{2}))$ | 0.7608 | 0.7236 | 0.6974 | 0.6783 | 0.6674 | 0.6608 | 0.6574 | 0.6562 | 0.6560 |
| $q = 2000$ | $\alpha_{opt}$ | 30 | 26 | 23 | 17 | 12 | 11 | 10 | 9 | 8 |
| | $\rho(T(\alpha_{opt}))$ | 0.8216 | 0.7911 | 0.7662 | 0.7503 | 0.7419 | 0.7349 | 0.7297 | 0.7233 | 0.7213 |
| | $\alpha^*$ | 0.3554 | 0.1638 | 0.0938 | 0.0606 | 0.0424 | 0.0313 | 0.0241 | 0.0191 | 0.0155 |
| | $\rho(T(\alpha^*))$ | 0.9103 | 0.9579 | 0.9757 | 0.9842 | 0.9889 | 0.9918 | 0.9937 | 0.9950 | 0.9959 |
| | $\frac{qh}{2}$ | 47.6190 | 32.2581 | 24.3902 | 19.6078 | 16.3934 | 14.0845 | 12.3457 | 10.9890 | 9.9010 |
| | $\rho(T(\frac{qh}{2}))$ | 0.8452 | 0.7953 | 0.7674 | 0.7535 | 0.7770 | 0.7366 | 0.7345 | 0.7302 | 0.7279 |

briefly described as follows.

"Let us assume that we have a solver set $\mathcal{S}$ with $n_s$ solvers and a problem set $\mathcal{P}$ with $n_p$ problems. Let $\mu$ be a performance measure of solvers, e.g., the CPU time of a solver to solve a problem, and let $\mu_{p,s}$ be the measure result for the problem $p$

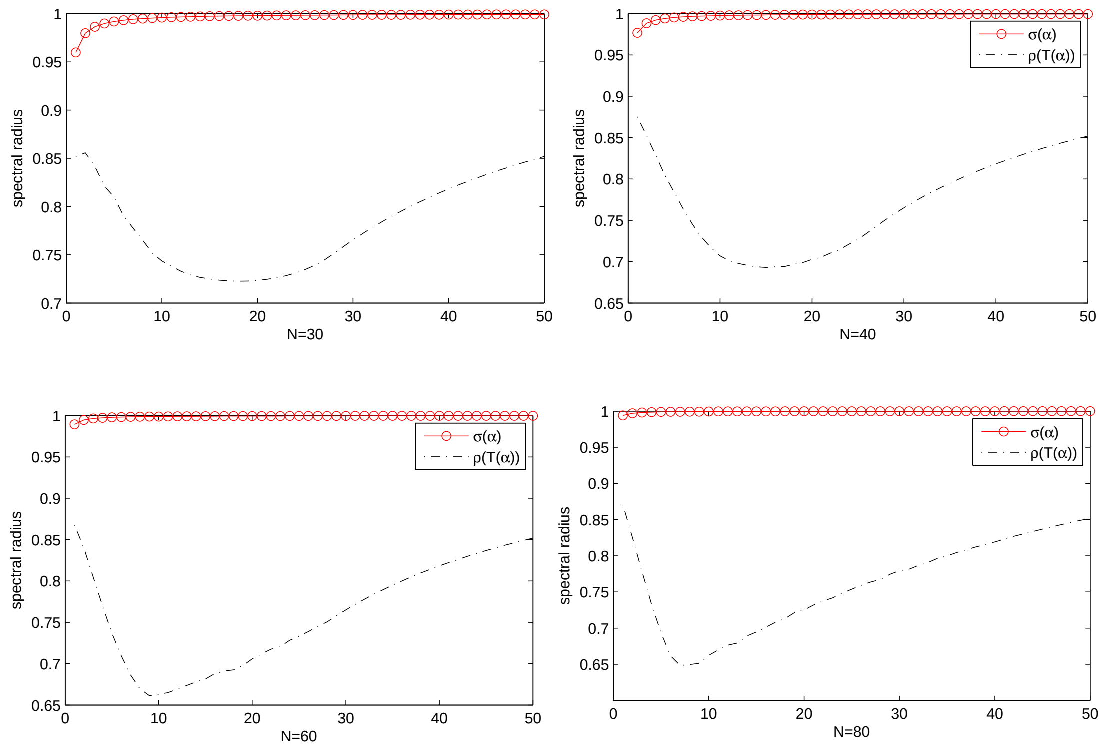


Figure 1: Plots of $\rho(T(\alpha))$ versus $\alpha$ with $q = 1000$ and different values of $N$ for HSS inner iterations.

when the solver $s$ is used. For each problem $p$, let

$$\mu_{p,min} = \min\{\mu_{p,s} \mid s \in \mathcal{S}, \text{and } p \text{ can be solved by } s\},$$

which is the best performance result for all solvers on problem $p$. Based on the profile ratio $\mu$, the performance ratio [20] is defined as

$$r_{p,s} = \begin{cases} \dfrac{\mu_{p,s}}{\mu_{p,min}}, & \text{if } p \text{ can be solved by } s, \\ r_\infty, & \text{if } p \text{ can not be solved by } s, \end{cases}$$

where $r_\infty > e^\xi$ is a given sufficiently large number, and

$$\xi \equiv \max\{\ln(r_{p,s}) | s \in \mathcal{S}, p \in \mathcal{P}, \text{and } p \text{ can be solved by } s\}. \tag{42}$$

Note that $r_{p,s}$ reflects the ratio of the performance of solver $s$ to the best performance on problem $p$. Now, the performance profile (see [20]) is defined as,

$$v_s(\tau) = \frac{|\Omega_s^\tau|}{n_p}, \quad \tau \in [0, \ln(r_\infty)], \tag{43}$$

where $\Omega_s^\tau = \{p \in \mathcal{P} | \ln(r_{p,s}) \leqslant \tau\}$ and $|\Omega_s^\tau|$ represents the number of elements contained in $\Omega_s^\tau$. Note that $v_s(\tau)$ denotes the probability for solver $s$ that a log-scale performance ratio is not greater than factor $\tau$. Besides,

(i) $v_s(0)$ represents the probability that solver $s$ can solve a problem with the best performance, and

(ii) $v_s(\xi)$ represents the probability that solver $s$ can solve a problem successfully. ”

In this study, the performance profiles based on the CPU time of NHSS and INHSS algorithms are shown in Figure 2. We have obtained these results in 50 tests by changing $N$ from 11 to 60 for both methods. From Figure 2 we can observe

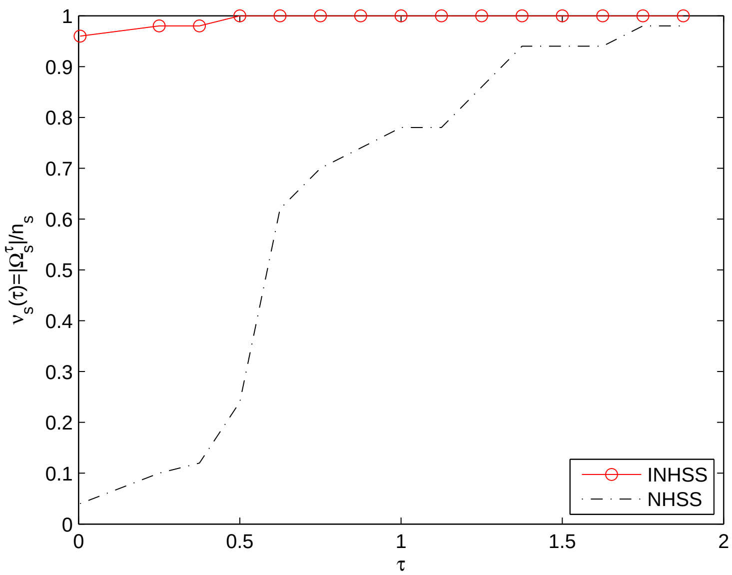


Figure 2: Performance profile based on CPU time for Jacobian free INHSS and NHSS algorithms ($\xi = 2$).

that the performance profile of INHSS algorithm is greater than NHSS one. Therefore, our new algorithm, that is, INHSS is the winner in the test. Observing the values of the performance profiles at point 0, we find that the probability that the INHSS method can give the best performance is nearly 0.96, while that of the NHSS is 0.04. By analyzing the highest parts of the two graphs in Figure 2, one see that INHSS method succeeds in solving about of the tests in the problem set, $100\%$, while that of the NHSS method is about $98\%$. This shows that INHSS scheme is more effective and robust than NHSS method.

### 5. Concluding Remarks

In this paper, a fast HSS-based algorithm has been proposed and applied to solve systems of nonlinear equations. The new scheme is an outer/inner iteration method. In our numerical example, the number of outer iterations in the new algorithm is less than or equal to half of outer iterations in the NHSS scheme. Therefore, our scheme in the sense of CPU-time and number of outer iterations is better than the NHSS method. As the computation of elements of Jacobian matrix is the most consumable part in any outer iteration step, hence reducing these computations can reduce the total CPU-time. Thus a Jacobian-free INHSS algorithm is presented which its application reduces the CPU-time. The obtained numerical results are compared with the existing numerical solutions. It is concluded that the presented algorithms, namely Jacobian-free INHSS, give better accuracy in comparison to Jacobian-free NHSS, Picard-HSS and nonlinear HSS-like methods. Also, spectral properties of our method are investigated. Finally, performance profiles of our algorithm is better than performance profiles of NHSS algorithm.
From these advantages, our algorithm might become a suitable method in finding the numerical solutions of nonlinear systems.

**Acknowledgement:** The authors would like to thank the anonymous reviewers for their comments and suggestions that have improved this manuscript.

### References

[1] Z.-Z. Bai, G.H. Golub, M.K. Ng, Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, SIAM J. Matrix Anal. Appl., 24 (2003) 603–626.

[2] D. Bertaccini, G.H. Golub, S.S. Capizzano, C.T. Possio, Preconditioned HSS methods for the solution of non-Hermitian positive definite linear systems and applications to the discrete convection-diffusion equation, Numer. Math., 99(3) (2005) 441-484.

[3] Z.-Z. Bai, G.H. Golub, J.-Y. Pan, Preconditioned Hermitian and skew-Hermitian splitting methods for non-Hermitian positive semidefinite linear systems, Numer. Math., 98(1)(2004) 1–32.

[4] Z.-Z. Bai, G.H. Golub, M.K. Ng, On successive-overrelaxation acceleration of the Hermitian and skew-Hermitian splitting iterations, Numer. Linear Algebra Appl., 14(4) (2007) 319–335.

[5] M. Benzi, M.K. Ng, Preconditioned iterative method for weighted Toeplitz least squares problems, SIAM J. Matrix Anal. Appl., 27 (2006) 1106–1124.

[6] L. Li, T.-Z. Huang, X.-P. Liu, Asymmetric Hermitian and skew-Hermitian splitting methods for positive definite linear systems, Comput. Math. Appl., 54 (2007) 147–159.

[7] L. Li, T.-Z. Huang, X.-P. Liu, Modified Hermitian and skew/Hermitian splitting methods for non-Hermitian positive-definite linear systems, Numer. Linear Algebra Appl., 14 (2007) 217–235.

[8] Z.-Z. Bai, A class of two-stage iterative methods for systems of weakly nonlinear equations, Numer. Algor., 14(4) (1997) 295–319.

[9] Z.-Z. Bai, On the convergence of parallel chaotic nonlinear multisplitting Newton-type methods, J. Comput. Appl. Math., 80(2) (1997) 317–334.

[10] Z.-Z. Bai, X. Yang, On HSS-based iteration methods for weakly nonlinear systems, Appl. Numer. Math., 59 (2009) 2923–2936.

[11] M.-Z. Zhu, Modified iteration based on the asymmetric HSS for weakly nonlinear systems, Comput. Anal. Appl., 15 (1) (2013) 188–195.

[12] Z.-Z. Bai, X.-P. Guo, The Newton-HSS methods for systems of nonlinear equations with positive-definite Jacobian matrices, J. Comput. Math., 28 (2010) 235–260.

[13] Z.-Z. Bai, G.H. Golub, M.K. Ng, On inexact Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, Linear Algebra Appl., 428 (2008) 413–440.

[14] J.F. Traub, Iterative Methods for the solution of equations, Prentice-Hall, Englewood Cliffs 1964.

[15] C.-X. Li, S.-L. Wu, A modified GHSS method for non-Hermitian positive definite linear systems, Japan J. Indust. Appl. Math., 29 (2012) 253–268.

[16] J.M. Ortega, W.C. Rheinboldt, Iterative solution of nonlinear equations in several variables, Academic Press, New York 1970.

[17] D.A. Knoll, D.A. Keyes, Jacobian-free Newton-Krylov methods, a survey of approaches and applications, J. Comput. Phys., 193 (2004) 357–397.

[18] X.-P. Guo, I.S. Duff, Semilocal and global convergence of the Newton-HSS method for systems of nonlinear equations, Linear Algebra Appl., 18 (2010) 299–315.

[19] E.D. Dolan, J.J. Moré, Benchmarking optimization software with performance profiles, Math. Program. Ser. A, 91 (2002) 201–213.

[20] H.-B. An, Z.-Z. Bai, A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations, Appl. Numer. Math., 57 (2007) 235–252.