Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 9, No. 3, 2021, pp. 710-721
DOI:10.22034/cmde.2020.34989.1596

Constructing an efficient multi–step iterative scheme for nonlinear system of equations

Taher Lotfi*
Department of Mathematics, Hamedan Branch,
Islamic Azad University, Hamedan, Iran.
E-mail: lotfitaher@yahoo.com

Mohammad Momenzadeh
Department of Mathematics, Hamedan Branch,
Islamic Azad University, Hamedan, Iran.
E-mail: m.momenzadeh12@yahoo.com

Abstract The objective of this research is to propose a new multi-step method in tackling a system of nonlinear equations. The constructed iterative scheme achieves a higher rate of convergence whereas only one LU decomposition per cycle is required to proceed. This makes the efficiency index to be high as well in contrast to the existing solvers. The usefulness of the presented approach for tackling differential equations of nonlinear type with partial derivatives is also given.

Keywords. Iterative methods; high order; nonlinear systems; partial differential equations; efficiency index.

2010 Mathematics Subject Classification. 41A15; 65H10.

1. Introductory notes and literature

Let us take into account the following set of algebraic equations of nonlinear type:

$$G(\mathbf{x}) = 0, \tag{1.1}$$

at which G is defined by $G(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_n(\mathbf{x}))^T$ while $g_1(\mathbf{x})$, $g_2(\mathbf{x})$, …, $g_n(\mathbf{x})$ are the functions of coordinate, see e.g. [17]. For some background and recent studies in the field of iteration schemes in tackling nonlinear equations, an interested reader may consult the works and [3, 5, 8, 16, 19] the references cited therein.

Let us consider that $G(\mathbf{x})$ is a smooth function of $\mathbf{x}$ in $D \subseteq \mathbb{R}^n$ which is an open convex set. Let us now recall some of the pioneer iterative methods for tackling (1.1). The widely used method of Newton for solving (1.1) is given by [17]:

$$\begin{cases} G'(\mathbf{x}^{(l)})\mathbf{s}^{(l)} = -G(\mathbf{x}^{(l)}) \\ \mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} + \mathbf{s}^{(l)}, \quad l = 0, 1, 2, \cdots . \end{cases} \tag{1.2}$$

Received: 1 August 2019 ; Accepted: 9 March 2020.

∗ corresponding.

An advantage of (1.2) is the second order rate at the cost of tackling only one linear system of algebraic equations per cycle as long as the guess $\mathbf{x}^{(0)}$ is enough close to the exact root. In fact, to overcome some of the drawbacks of (1.2) and similar schemes like the derivative-free Steffensen's scheme, several works have been published.

An improvement of (1.2) was introduced by Traub possessing a cubical rate of convergence as comes next [25]:

$$\begin{cases} \mathbf{h}^{(l)} = \mathbf{x}^{(l)} - G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}), \\ \mathbf{x}^{(l+1)} = \mathbf{h}^{(l)} - G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{h}^{(l)}). \end{cases} \tag{1.3}$$

The Jarratt's iteration expression with quartic rate of local speed is defined by [4]:

$$\begin{cases} \mathbf{h}^{(l)} = \mathbf{x}^{(l)} - \frac{2}{3}G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}), \\ \mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \frac{1}{2}(3G'(\mathbf{h}^{(l)}) - G'(\mathbf{x}^{(l)}))^{-1} \\ \qquad\qquad \cdot(3G'(\mathbf{h}^{(l)}) + G'(\mathbf{x}^{(l)}))G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}). \end{cases} \tag{1.4}$$

Authors in [15] proposed the following scheme:

$$\begin{cases} \mathbf{v}^{(l)} &= G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}), \\ \mathbf{z}^{(l)} &= \mathbf{x}^{(l)} - \frac{2}{3}\mathbf{v}^{(l)}, \\ \mathbf{h}^{(l)} &= \mathbf{x}^{(l)} - [\frac{23}{8}I - 3G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{z}^{(l)}) \\ & \qquad + \frac{9}{8}(G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{z}^{(l)}))^2]\mathbf{v}^{(l)}, \\ \mathbf{x}^{(l+1)} &= \mathbf{h}^{(l)} - [\frac{5}{2}I - \frac{3}{2}G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{z}^{(l)})]G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{h}^{(l)}). \end{cases} \tag{1.5}$$

The purpose of this article is to propose a novel multi steps high order scheme to solve (1.1). Hence, we propose an eighth-order iterative scheme to compute the simple roots. The method is free of calculating the function's second Fréchet derivative. Note that some discussions on the usefulness of higher order iterative methods with and without memory can be found at [10, 11, 12].

The rest sections in this paper are given as follows. Section 2 contributed the derivation of the new scheme consisting of multi steps while Section 3 furnishes an analysis of error for its speed of convergence. Section 4 discusses the application of the proposed scheme with some discussion concerning the efficiency index in solving nonlinear partial differential equations (PDEs), [22, 23, 24]. The paper ends in Section 5 with some outlines for future works.

2. Construction of a scheme

Here, we propose our iteration scheme as an improved method over (1.2)-(1.5) as follows:

$$\begin{cases} \mathbf{y}^{(l)} &= \mathbf{x}^{(l)} - \frac{2}{3}G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}), \\ \mathbf{z}^{(l)} &= \mathbf{x}^{(l)} - [\frac{23}{8}I - 3G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{y}^{(l)}) \\ & \qquad + \frac{9}{8}(G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{y}^{(l)}))^2]G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{x}^{(l)}), \\ \mathbf{w}^{(l)} &= \mathbf{z}^{(l)} - [\frac{5}{2}I - \frac{3}{2}G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{y}^{(l)})]G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{z}^{(l)}), \\ \mathbf{x}^{(l+1)} &= \mathbf{w}^{(l)} - [\frac{5}{2}I - \frac{3}{2}G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{y}^{(l)})]G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{w}^{(l)}). \end{cases} \tag{2.1}$$

Note that in such schemes, one should avoid computing the inverse matrix. As such, linear systems using LU decompositions must be tackled. Now, if we consider

$G'(\mathbf{x}^{(l)}) = U^l$:

$$\begin{cases} U^l V^{(l)} = G(\mathbf{x}^{(l)}), \\ U^l M^{(l)} = G'(\mathbf{y}^{(l)}), \\ U^l W^{(l)} = G(\mathbf{z}^{(l)}), \\ U^l L^{(l)} = G(\mathbf{w}^{(l)}), \end{cases} \tag{2.2}$$

then the iterative expression (2.1) can be furnished in a more elegant way as follows:

$$\begin{cases} \mathbf{y}^{(l)} = \mathbf{x}^{(l)} - \frac{2}{3}V^{(l)}, \\ \mathbf{z}^{(l)} = \mathbf{x}^{(l)} - [\frac{23}{8}I - 3M^{(l)} + \frac{9}{8}(M^{(l)})^2]V^{(l)}, \\ \mathbf{w}^{(l)} = \mathbf{z}^{(l)} - [\frac{5}{2}I - \frac{3}{2}M^{(l)}]W^{(l)}, \\ \mathbf{x}^{(l+1)} = \mathbf{w}^{(l)} - [\frac{5}{2}I - \frac{3}{2}M^{(l)}]L^{(l)}. \end{cases} \tag{2.3}$$

To implement (2.3), one requires to tackle several linear systems of algebraic equation when the matrices U^l are dense and when they are sparse and of large size, some Krylov subspace methods could be taken into account to accelerate the process. Anyhow, the advantage in (2.3) is that the four systems of linear equation having the same system matrix.

As such, just one LU decomposition is enough, and then we can incorporate it to four various right hand side vectors and get the solution vector per sub-cycle of (2.3).

The convergence rate of the iteration expression (2.3) without memory must be pursued by applying n-D Taylor expansions. Accordingly, it is recalled that $e^{(l)} = \mathbf{x}^{(l)} - \mathbf{x}^*$ is the called the error at lth iterate and also [17, 20]:

$$e^{(l+1)} = Je^{(l)^p} + \mathcal{O}(e^{(l)^{p+1}}), \tag{2.4}$$

is named as the error equation, where J is a function of p-linear type. This means that $J \in L(\mathbb{R}^n, \mathbb{R}^n, \dots, \mathbb{R}^n)$ and p is the speed rate. Furthermore, we obtain:

$$e^{(l)^p} = (e^{(l)}, e^{(l)}, \dots, e^{(l)}). \tag{2.5}$$

3. Investigating the convergence rate

In this section, we give the following theoretic for the convergence speed of (2.3). Before going to the main theorem, we recall the n-D Taylor expansion as follows.

Assume that $G : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be sufficiently differentiable in terms of Fréchet in D. As in [6], the mth differentiation of G at $u \in \mathbb{R}^n$, $m \geq 1$, is the m-linear function

$$G^{(m)}(u) : \mathbb{R}^n \times \cdots \times \mathbb{R}^n \longrightarrow \mathbb{R}^n, \tag{3.1}$$

so that $G^{(m)}(u)(v_1, \dots, v_m) \in \mathbb{R}^n$. It is also famous that, for $\mathbf{x}^* + h \in \mathbb{R}^n$ locating in a neighborhood of a root $\mathbf{x}^*$ of (1.1), the expansion of Taylor could be written and we have [6]:

$$G(\mathbf{x}^* + h) = G'(\mathbf{x}^*)\left[h + \sum_{m=2}^{p-1} C_m h^m\right] + \mathcal{O}(h^p), \tag{3.2}$$

wherein

$$C_m = (1/m!)[G'(\mathbf{x}^*)]^{-1}G^{(m)}(\mathbf{x}^*),\ m \geq 2. \tag{3.3}$$

One finds $C_m h^m \in \mathbb{R}^n$, because $G^{(m)}(\mathbf{x}^*) \in L(\mathbb{R}^n \times \cdots \times \mathbb{R}^n, \mathbb{R}^n)$ and $[G'(\mathbf{x}^*)]^{-1} \in L(\mathbb{R}^n)$. Moreover, for G' we have:

$$G'(\mathbf{x}^* + h) = G'(\mathbf{x}^*)\left[I + \sum_{m=2}^{p-1} mC_m h^{m-1}\right] + \mathcal{O}(h^p), \tag{3.4}$$

where I is the matrix of identity. Herein, $mC_m h^{m-1} \in L(\mathbb{R}^n)$.

Theorem 3.1. *Assume that in (1.1) $G : D \subseteq \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is Fréchet differentiable sufficiently at any points of D at $\mathbf{x}^* \in \mathbb{R}^n$. Herein we take into account that $G(\mathbf{x}^*) = 0$. Furthermore assume that $G'(\mathbf{x})$ is nonsingular and continuous in $\mathbf{x}^*$. Hence, $\{\mathbf{x}^{(l)}\}_{l\geq 0}$ produced applying (2.3) tends to $\mathbf{x}^*$ with the rate of eight, while the equation of error satisfies*

$$e^{(l+1)} = -\frac{1}{9}(6C_2^2 - C_3)^2(45C_2^3 - 9C_3C_2 + C_4){e^{(l)}}^8 + \mathcal{O}({e^{(l)}}^9). \tag{3.5}$$

Proof. To prove the rate of converging, we apply (3.2) and (3.4) to get

$$G(\mathbf{x}^{(l)}) = G'(\mathbf{x}^*)\left[e^{(l)} + C_2{e^{(l)}}^2 + \cdots + C_8{e^{(l)}}^8\right] + \mathcal{O}({e^{(l)}}^9), \tag{3.6}$$

and

$$G'(\mathbf{x}^{(l)}) = G'(\mathbf{x}^*)\left[I + 2C_2e^{(l)} + \cdots + 8C_8{e^{(l)}}^7\right] + \mathcal{O}({e^{(l)}}^8), \tag{3.7}$$

where $C_l = (1/l!)[G'(\mathbf{x}^*)]^{-1}G^{(l)}(\mathbf{x}^*)$, $l = 2, 3, \ldots$, and $e^{(l)} = \mathbf{x}^{(l)} - \mathbf{x}^*$. Using (3.7), one obtains:

$$[G'(\mathbf{x}^{(l)})]^{-1} = \left[I + X_1e^{(l)} + X_2{e^{(l)}}^2 + X_3{e^{(l)}}^3 + \cdots\right][G'(\mathbf{x}^*)]^{-1} + \mathcal{O}({e^{(l)}}^8), \tag{3.8}$$

wherein $X_1 = -2C_2$, $X_2 = 4C_2^2 - 3C_3$, $X_3 = -8C_2^3 + 6C_2C_3 + 6C_3C_2 - 4C_4$, $\cdots$. Thus from (3.6)-(3.8), we obtain that

$$\mathbf{y}^{(l)} = \mathbf{x}^* + \frac{1}{3}e^{(l)} + \frac{2}{3}C_2{e^{(l)}}^2 \cdots + \mathcal{O}({e^{(l)}}^7). \tag{3.9}$$

For the matrix of Jacobian obtained from $G'(y^{(l)})$, one can similarly obtain

$$\begin{aligned} G'(\mathbf{y}^{(l)}) \quad &= \quad G'(\mathbf{x}^*)[I + 2C_2(\mathbf{x}^{(l)} - \mathbf{x}^*) + 3C_3(\mathbf{x}^{(l)} - \mathbf{x}^*)^2 \\ &\quad +4C_4(\mathbf{x}^{(l)} - \mathbf{x}^*)^3 5C_5(\mathbf{y}^{(l)} - \mathbf{x}^*)^4 + \cdots + \mathcal{O}({e^{(l)}}^7)]. \end{aligned} \tag{3.10}$$

Now, we obtain:

$$\mathbf{z}^{(l)} - \mathbf{x}^* \quad = \quad \left(5C_2^3 - C_3C_2\frac{1}{9} + C_4\right){e^{(l)}}^4 + \cdots + \mathcal{O}({e^{(l)}}^9). \tag{3.11}$$

Here, in order to avoid writing bulky formulas which will be simplified in the forthcoming sub steps of our proposed iterative scheme, we used "$\cdots$". This shows that similar expressions based on the multi-dimensional Taylor expansion are existed up to that order.

Now similarly, we have:

$$G(\mathbf{z}^{(l)}) = \frac{1}{9}(13C_2^3 - 9C_2C_3 + C_4)G'(\mathbf{x}^*){e^{(l)}}^4 - \frac{2}{27}(86C_2^4 - 144C_2C_3C_2$$

$$+27C_3^2 + 30C_4C_2 - 4C_5)G'(\mathbf{x}^*){e^{(l)}}^5 + \cdots + \mathcal{O}({e^{(l)}}^8). \tag{3.12}$$

Using (3.8)-(3.12) in the structure of the third sub step of (2.3), one derives:

$$[\frac{5}{2}I - \frac{3}{2}G'(\mathbf{x}^{(l)})^{-1}G'(\mathbf{y}^{(l)})]G'(\mathbf{x}^{(l)})^{-1}G(\mathbf{z}^{(l)}) = \left(5C_2^3 - C_3C_2 + \frac{C_4}{9}\right){e^{(l)}}^4$$
$$+\left(-36C_2^4 + 32C_2C_3C_2 - 2C_3^2 - \frac{20}{9}C_4C_2 + \frac{8}{27}C_5\right){e^{(l)}}^5$$
$$+(140C_2^5 - 251C_2^2C_3C_2 + 65C_3C_2C_3 + \frac{416}{9}C_2C_4C_2$$
$$-\frac{65}{9}C_4C_3 - \frac{10}{3}C_5C_2 + \frac{14}{27}C_6){e^{(l)}}^6 + \cdots + \mathcal{O}({e^{(l)}}^9). \tag{3.13}$$

Using (3.13) and further Taylor expansions at the fourth sub step of our method, we attain the final error equation of the presented scheme as in (3.5). This shows its eighth rate of convergence by employing only one LU decomposition per cycle to proceed. The proof is ended. □

3.1. **Comparing the indices of efficiency.** To achieve the high eighth rate of convergence in our presented scheme, we require to solve four linear systems per cycle but each having the same system matrix. Accordingly, just one LU factorization must be done and then act of the right hand side vectors. This would yield in an increased computational efficiency for the proposed solver since it avoids computing several matrix inverses per cycle.

To evaluate the index of efficiency for (2.3), we need to first define the number of functional evaluations, viz, the cost, for n-D functions and Jacobians as follows (without the index l):

- For $G(\mathbf{x})$, n evaluations,
- For $G(\mathbf{z})$, n evaluations,
- For the Jacobian $G'(\mathbf{x})$, n^2 evaluations,
- For the Jacobian $G'(\mathbf{y})$, n^2 evaluations, and similarly for other involved functional evaluations.

The efficiency index for iterative methods defined in the discussed context is given by [9]:

$$E = \mathfrak{p}^{\frac{1}{\mathcal{C}}}, \tag{3.14}$$

where $\mathcal{C}$ is the whole burden and $\mathfrak{p}$ is the speed in each loop by taking into account the number of function evaluations. Precisely, we also take into account that the cost for calculating each of the scalar functions is unit. And the cost for the computation of other operations is a factor of the unit cost.

Assume also that the number of matrix products, quotients, summations, and subtractions along with the cost of solving two triangular systems, viz, via flops are counted here. Reminding that the flops for computing the LU decomposition is $\frac{2n^3}{3}$, and to handle the 2 required triangular systems, the flops would be roughly $2n^2$. Whenever a matrix is in the right hand side, then the cost (flops) of the 2 triangular systems would increase to n^3.

The results in Table 1 show that for large n the efficiency index of our scheme overcome on its competitors. We do not include the results of other schemes since they are mainly inefficient whenever more than one LU decomposition should be computed per cycle.

Table 1. Comparing the efficiency indices of variants methods.

Iterative methods	(1.2)	(1.5)	(2.3)
No. of steps	1	3	4
Rate of convergence	2	6	8
No. of functional evaluations	$n+n^2$	$2n+2n^2$	$3n+2n^2$
The classical efficiency index	$2^{\frac{1}{n+n^2}}$	$6^{\frac{1}{2n+2n^2}}$	$8^{\frac{1}{3n+2n^2}}$
No. of LU factorizations	1	1	1
Cost for LU decompositions (based on flops)	$\frac{2n^3}{3}$	$\frac{2n^3}{3}$	$\frac{2n^3}{3}$
Cost for linear systems (based on flops)	$\frac{2n^3}{3}+2n^2$	$\frac{5n^3}{3}+4n^2$	$\frac{5n^3}{3}+6n^2$
Flops-like index of efficiency	$2^{\frac{1}{\frac{2n^3}{3}+3n^2+n}}$	$6^{\frac{1}{\frac{5n^3}{3}+6n^2+2n}}$	$8^{\frac{1}{\frac{5n^3}{3}+8n^2+3n}}$

4. Application in nonlinear PDEs

Here the objective is to reveal the application of our presented method for solving nonlinear PDEs arising frequently in important engineering problems.

The codes are done in Mathematica 11.0, [14]. The involved linear system of equations are handled by the following command:

```
LinearSolver[]
```

In addition, the computational tests are all done in the same environment having the following hardware specifications: 16.00 GB of RAM, Core(TM) i5–2430M CPU with Windows 7 Ultimate.

In the comparisons, we considered the quadratically convergent iterative method of Newton (1.2), the sixth-order expression of Montazeri et al. (1.5), and the presented eighth-order approach (2.3) for solving our system of nonlinear equations extracted from the numerical solution of nonlinear time dependent PDEs. Some other differential problems that might need a nonlinear algebraic solver such as the proposed ones, can be observed in [7, 13, 21].

When solving nonlinear problems like differential equations with partial derivatives in the nonlinear form [1, 2], this process will normally be yielded to tackle nonlinear systems of equations. Noting that efficient two or three steps methods for solving nonlinear systems can be found in [1, 2]. Apart from this point, normally computational procedures of lower accuracy in terms of the accuracy are required. Consequently, we approximate the solution employing the discretization of finite difference (FD) with the following stop termination:

$$||G(\cdot)||_2 < 10^{-8}. \tag{4.1}$$

Here, we take into consideration $u = u(y,t)$, as the true resolution of the nonlinear PDE. The numerical approximation is shown by:

$$w_{i,j} \simeq u(y_i, t_j), \tag{4.2}$$

at the location i, j on our uniform discretization points. Let us also denote M and N to be the steps' number for the space and time, and $m = M - 1$, $n = N - 1$, respectively.

Burgers' PDE or Bateman-Burgers PDE is a pioneer PDE happening in different fields of applied and computational mathematics, [27]. Burgers equations appear often as a simplification of a more complex and sophisticated model. Hence, it is usually thought as a toy model, viz, a tool that is applied to find out some of the behavior of a general procedure.

Experiment 4.1. *An important model of fluid flow is the Burgers' PDE alongside the side conditions of Dirichlet type with the coefficient of diffusion D as comes next:*

$$\begin{cases} u_t + uu_y = Du_{yy}, \\ u(y,0) = \frac{2D\beta\pi\sin(\pi y)}{\alpha+\beta\cos(\pi y)}, \quad 0 \leq y \leq 2, \\ u(0,t) = 0, \ t \geq 0, \\ u(1,t) = 0, \ t \geq 0. \end{cases} \tag{4.3}$$

To tackle this nonlinear problem, we employ the backward FD for the 1st derivative in time t:

$$u_t(y_i, t_j) \simeq \frac{w_{i,j} - w_{i,j-1}}{k}, \tag{4.4}$$

wherein the step size is k, and the FD formula of the central type for the other terms of the PDE, viz.,

$$u_y(y_i, t_j) \simeq \frac{w_{i+1,j} - w_{i-1,j}}{2h}, \tag{4.5}$$

and also

$$u_{yy}(y_i, t_j) \simeq \frac{w_{i+1,j} - 2w_{i,j} + w_{i-1,j}}{h^2}, \tag{4.6}$$

where we consider h to be the step size for discretization along y. We assume the following values for the involved parameters $\beta = 4$, $D = 0.05$, $T = 1$, and $\alpha = 5$.

When (4.3) is being solved, the procedure yields a system of nonlinear of equations possessing a large sparse matrix of Jacobian that are resolved and compared throughout the tested methods. The solution has been plotted in Figures 1-2. Table 2 demonstrates the comparison evidences in this case. Here, we have chosen $M = N = 21$, to attain a system of dimension 400×400, using the initial guess $\mathtt{y}_0 = \mathtt{Table}[0.6, \{\mathtt{i}, 1, \mathtt{m} * \mathtt{n}\}]$ written in the Mathematica with the precision of machine.

Another significant type of nonlinear PDEs is comprised of reaction-diffusion equations, [27]. This is investigated in the following experiment. It is also requisite to note that [18, chapter 8] the Experiments 4.2-4.2 do not admit an exact solution in general unless for special initial and boundary conditions, and thus the numerical results are not compared with any exact solution while the graphs of the solutions resemble totally with those in [18]. Furthermore, by having the spatial step size there is a clear convergence for the numerical values to specific values. However, since our aim is mainly focused on showing the applicability and usefulness of the proposed nonlinear algebraic solver for tackling discretization problems, we do not provide any comparison with other PDE solvers and we report the results of accuracies based on (4.1), i.e., based on the residual of the obtained nonlinear system of equations. Besides, the plots for (1.2), (1.5) and (2.3) are very much resemble to each other but (2.3) possess more accurate values due to its higher rate of convergence and efficiency.

Experiment 4.2. *Consider tackling the Fisher's equation with side conditions of the homogenous Neumann type:*

$$\begin{cases} u_t = Du_{yy} + u(1-u), \\ u(y,0) = \sin(\pi y), \quad 0 \le y \le 1, \\ u_y(0,t) = 0, \ \ t \ge 0, \\ u_y(1,t) = 0, \ \ t \ge 0. \end{cases} \tag{4.7}$$

Note that

$$f(u) = u(1-u), \tag{4.8}$$

indicating that $f'(u) = 1 - 2u$. In tackling (4.7) using the same discretizations as in Experiment 4.1, one obtains a set of nonlinear equations.

Here a challenge arises for the Neumann boundary conditions whereas two collections of novel nonlinear equations at the mesh nodes would be added into the system. This means that the following discretized equations have to be imposed:

$$u_y(0,t_j) \simeq \frac{-3w_{0,j} + 4w_{1,j} - w_{2,j}}{2h}, \tag{4.9}$$

and

$$u_y(1,t_j) \simeq \frac{-3w_{m-2,j} + 4w_{m-1,j} - w_{m,j}}{2h}. \tag{4.10}$$

The computational evidences for tackling this experiment are reported in Table 3, while its numerical resolution is shown in Figures 3-4. Here, we have selected $M = N = 23$, to attain a nonlinear system of dimension 528×528, which provides a large sparse Jacobian alongside the following initial guess

$$\mathtt{y}_0 = \mathtt{Table}[0., \{\mathtt{i}, 1, \mathtt{m} * \mathtt{n}\}]; \tag{4.11}$$

Table 2. Comparison evidences for Experiment 4.1.

Iterative methods	(1.2)	(1.5)	(2.3)
Iterates count	5	2	1
CPU time	4.45	3.85	3.02

Table 3. Comparison evidences for Experiment 4.2.

Iterative methods	(1.2)	(1.5)	(2.3)
Number of iterations	5	2	1
The elapsed time	6.23	3.67	3.25

Herein, the residual norm along with the iterates' counts in Mathematica 11.0 [26] are reported in Tables 2-3. The observations reveal that the proposed approach is not only better in terms of the elapsed computational time but also is fast and stable in calculating the computational solution of large scale nonlinear systems of algebraic equations extracted from the numerical solution of nonlinear PDEs in real problems via discretization techniques.

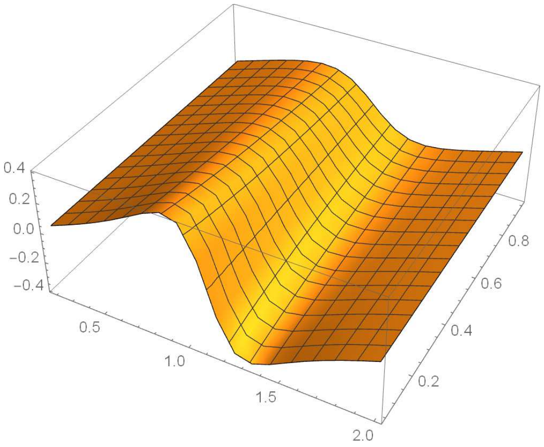

FIGURE 1. Numerical solution of nonlinear Burgers' PDE applying (2.3).

5. Summary

This work has contributed a higher order iteration scheme without memory as a multi step solver so as to find the simple zeros of nonlinear system of equations. Both real and complex solutions can be found upon the choice of a proper initial approximation.

The proposed approach reached an eighth order of convergence by solving only four linear systems per cycle via only one LU decomposition. This makes the implementation of the scheme quite straightforward and useful with a high computational efficiency index specially once the user needs to solve hard nonlinear systems arising from applications.

The scheme (2.3) is free from the computation of the 2nd Fréchet derivative and is of high order. The application of the proposed solver in real problems originated from solving nonlinear PDEs via discretization approaches was also pointed out showing a stable and fast behavior of this method.

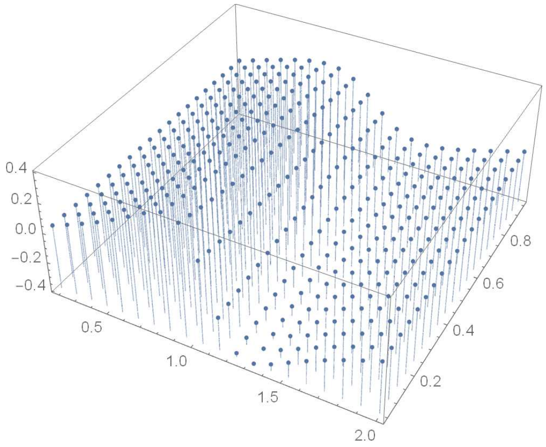

Figure 2. List point plot solution of nonlinear Burgers's PDE applying (2.3).

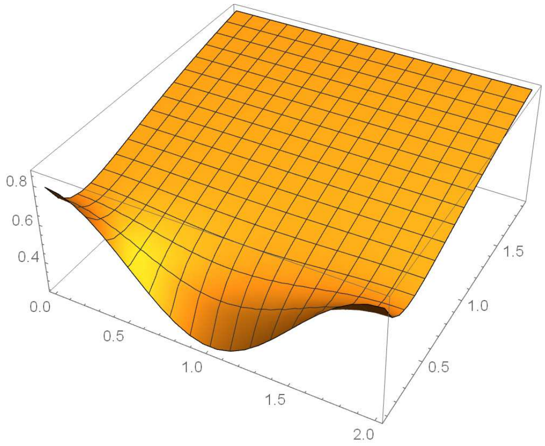

Figure 3. Numerical solution of nonlinear Fisher' PDE applying (2.3).

Acknowledgements

We are thankful to three anonymous referees whose comments and suggestions helped improve this work.

Conflicts of Interest

No conflict of interest is stated by the writers.

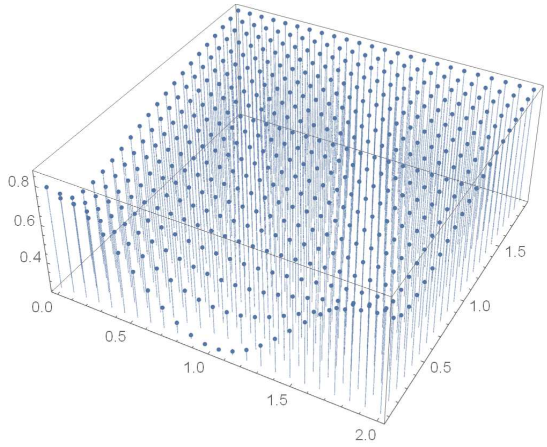

FIGURE 4. List point plot solution of nonlinear Fisher's PDE applying (2.3).

REFERENCES

[1] F. Ahmad, E. Tohidi, M. Zaka Ullah, and J. A. Carrasco, *Higher order multi–step Jarratt–like method for solving systems of nonlinear equations: Application to PDEs and ODEs*, Comput. Math. Appl. *70* (2015), 624–636.

[2] F. Ahmad, S. ur Réhman, M. Zaka Ullah, H. Moaiteq Aljahdali, S. Ahmad, A. S. Alshomrani, J. A. Carrasco, S. Ahmad, and S. Sivasankaran, *Frozen Jacobian multistep iterative method for solving nonlinear IVPs and BVPs*, Complexity, *2017* (2017), 1–30.

[3] F. Ahmad, T. A. Bhutta, U. Sohaib, M. Zaka Ullah, A. S. Alshomrani, S. Ahmad, and S. Ahmad, *A preconditioned iterative method for solving systems of nonlinear equations having unknown multiplicity*, Algorithms, *10* (2017), 1–9.

[4] D. K. R. Babajee, M. Z. Dauhoo, M. T. Darvishi, and A. Barati, *A note on the local convergence of iterative methods based on Adomian decomposition method and 3-node quadrature rule*, Appl. Math. Comput., *200* (2008), 452-458.

[5] R. Behl, E. Martínez, F. Cevallos, and D. Alarcón, *A higher order Chebyshev-Halley-type family of iterative methods for multiple roots*, Math., *7* (2019), 1-12.

[6] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, *A modified Newton-Jarratt's composition*, Numer. Algor., *55* (2010), 87-99.

[7] P. Darania, *Superconvergence analysis of multistep collocation method for delay Volterra integral equations*, Comput. Meth. Diff. Equ., *4* (2016), 205-216.

[8] I. G. Ivanov and T. Mateva, *Interval methods with fifth order of convergence for solving nonlinear scalar equations*, Axioms, *8* (2019), 1-11.

[9] M. Grau-Sánchez, À. Grau, and M. Noguera, *On the computational efficiency index and some iterative methods for solving systems of nonlinear equations, J. Comput. Appl. Math.*, *236* (2011), 1259-1266.

[10] F. W. Khdhr, F. Soleymani, R. K. Saeed, and A. Akgül, *An optimized Steffensen–type iterative method with memory associated with annuity calculation*, Euro. Phys. J. Plus, *134* (2019), Article ID: 146, 1–12.

[11] F. W. Khdhr, R. K. Saeed, and F. Soleymani, *Improving the computational efficiency of a variant of Steffensen's method for nonlinear equations*, Math., *7* (2019), Article ID: 306, 1–9.

[12] Y. I. Kim, R. Behl, and S. S. Motsa, *An optimal family of eighth-order iterative methods with an inverse interpolatory rational function error corrector for nonlinear equations*, Math. Model. Anal., *22* (2017), 321-336.

[13] C. Lin, C. W. Hsu, T. E. Simos, and Ch. Tsitouras, *Explicit, semi-symmetric, hybrid, six-step, eighth order methods for solving $Y'' = F(X;Y)$*, Appl. Comput. Math., *18* (2019), 296-304.

[14] S. Mangano, *Mathematica Cookbook*, O'Reilly Media, USA, 2010.

[15] H. Montazeri, F. Soleymani, S. Shateyi, and S. S. Motsa, *On a new method for computing the numerical solution of systems of nonlinear equations*, J. Appl. Math., *2012* (2012), Article ID: 751975, 15 pages.

[16] O. Ogbereyivwe and K. Obiajulu Muka, *Multistep quadrature based methods for nonlinear system of equations with singular Jacobian*, J. Appl. Math. Phys., *7* (2019), 702-725.

[17] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[18] T. Sauer, *Numerical Analysis*, Pearson, 2nd edition, USA, 2012.

[19] K. Shah and K. Rahmat, *Iterative scheme to a coupled system of highly nonlinear fractional order differential equations*, Computational Methods for Differential Equations, *3*(3) (2015), 163-176.

[20] J. R. Sharma, R. K. Guha, and R. Sharma, *An efficient fourth order weighted-Newton method for systems of nonlinear equations*, Numer. Algor., *62* (2013), 307–323.

[21] T. E. Simos and Ch. Tsitouras, *High phase lag-order, four-step methods for solving $y'' = f(x, y)$*, Appl. Comput. Math., *17* (2018), 307-316.

[22] F. Soleymani, M. Barfeie, and F. Khaksar Haghani, *Inverse multi-quadric RBF for computing the weights of FD method: Application to American options*, Commun. Nonlinear Sci. Numer. Simul., *64* (2018), 74-88.

[23] F. Soleymani, *Pricing multi-asset option problems: A Chebyshev pseudo-spectral method*, BIT, *59* (2019), 243-270.

[24] F. Soleymani and B. N. Saray, *Pricing the financial Heston–Hull–White model with arbitrary correlation factors via an adaptive FDM*, Comput. Math. Appl, *77* (2019), 1107–1123.

[25] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice Hall, New York, 1964.

[26] *Wolfram Research*, Inc., Mathematica, Version 11.0, Champaign, IL (2016).

[27] M. Zaka Ullah, F. Soleymani, and A. S. Al-Fhaid, *Numerical solution of nonlinear systems by a general class of iterative methods with application to nonlinear PDEs*, Numer. Algor, *67* (2014), 223–242.