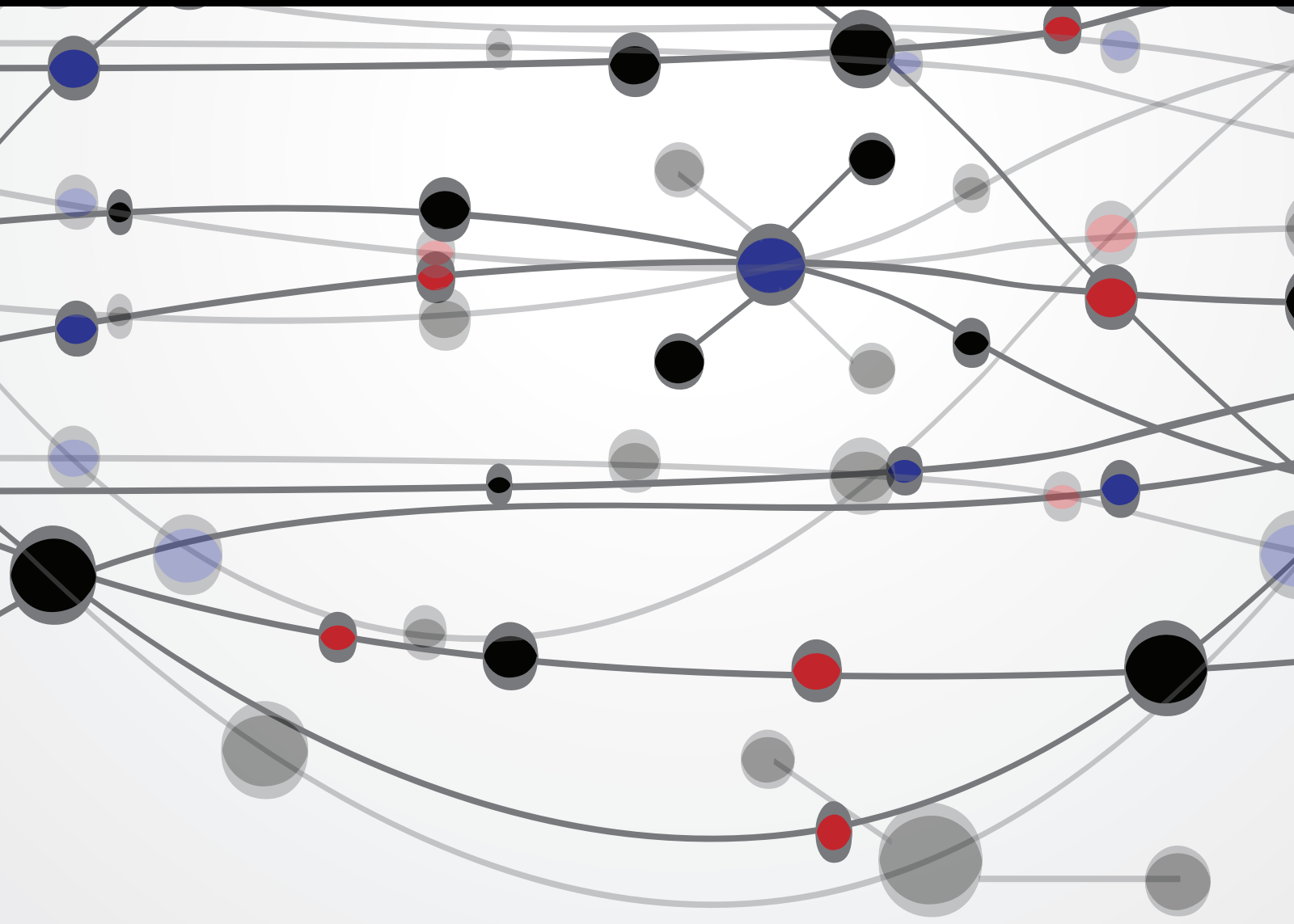
The Scientific World Journal

Recent Theories and Applications in Approximation Theory

Guest Editors: Fazlollah Soleymani, Predrag S. Stanimirovic, Juan R. Torregrosa, Hassan S. Nik, and Emran Tohidi



Recent Theories and Applications in Approximation Theory

The Scientific World Journal

Recent Theories and Applications in Approximation Theory

Guest Editors: Fazlollah Soleymani, Predrag S. Stanimirović, Juan R. Torregrosa, Hassan Saberi Nik, and Emran Tohidi

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 598279, 2 pages
http://dx.doi.org/10.1155/2015/598279

Editorial

Recent Theories and Applications in Approximation Theory

Fazlollah Soleymani,[1] Predrag S. Stanimirović,[2] Juan R. Torregrosa,[3] Hassan Saberi Nik,[4] and Emran Tohidi[5]

[1]*Department of Mathematics, Islamic Azad University, Zahedan Branch, Zahedan, Iran*
[2]*Faculty of Sciences and Mathematics, University of Niš, Visegradska 33, 18000 Niš, Serbia*
[3]*Instituto de Matemáticas Multidisciplinar, Universitat Politècnica de València, 46022 Valencia, Spain*
[4]*Department of Applied Mathematics, School of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran*
[5]*Department of Mathematics, Kosar University of Bojnord, P. O. Box 9415615458, Bojnord, Iran*

Correspondence should be addressed to Fazlollah Soleymani; fazlollah.soleymani@gmail.com

Received 1 March 2015; Accepted 1 March 2015

Approximation theory is a deep theoretical study of methods that use numerical approximation for the problems of mathematical analysis. In practical use, it is typically the application of computer simulation and other forms of computation to problems in various scientific disciplines. Recently, numerical algorithms in approximation theory have been a major thrust of research with numerous applications.

This special issue was opened in the middle of 2014 and closed in February of 2015. A number of selected submissions were accepted for publication after strict reviews, which furnished significant improvements in the topics of the special issue and its related applications. The guest editors of this special issue hope that the published results could provide outstanding viewpoints for further studies.

The fundamental aim of this special issue was to provide new trends in the field of approximation theory and related applications in mathematics. The authors were invited to submit original research articles to stimulate the continuing efforts in numerical approximation of mathematical problems and related theories. The special issue provided a forum for researchers and writers to communicate their state-of-the-art improvements and to propose their new findings on approximation theory.

The topics of the accepted papers cover the area from theory to real applications. Some new schemes and their corresponding convergence analysis have been discussed for some numerical problems. Furthermore, they have been equipped with several numerical tests with some applications.

Now, we have the pleasure to present, the selected papers for this special issue as follows.

S. S. Motsa et al. presented a novel scheme for solving higher order nonlinear evolution partial differential equations (NPDEs). Their discussed approach combines quasi-linearisation, the Chebyshev spectral collocation method, and bivariate Lagrange interpolation. They also showed that there is congruence between the numerical results and the exact solutions to a high order of accuracy.

The paper of M. Sharifi et al. presented interesting iterative methods including three steps for solving nonlinear equations. Their iterative approach possesses eighth-order of convergence which is optimal in the sense of Kung-Traub while it is also derivative-free. An integral equation has also been solved as an application-oriented experiment.

T. Lotfi et al., in their paper, investigate an optimal three-step method which has eighth-order of convergence. Then, they applied a self-accelerator parameter with Newtonian interpolation using the highest possible degree to improve the R-order of convergence as much as possible, that is, from 8 to 12 without any additional functional evaluations. This meant that a high computational efficiency index has been obtained for solving nonlinear equations.

H. S. Nik and P. Rebelo presented an application of pseudospectral method for solving the hyperchaotic complex systems. The proposed method, called the Multistage Spectral Relaxation Method (MSRM), was based on a technique of extending Gauss-Seidel type relaxation ideas to systems of nonlinear differential equations, while using the Chebyshev pseudospectral methods to solve the resulting system on a sequence of multiple intervals. Finally, it has been used to solve famous hyperchaotic complex systems such

as hyperchaotic complex Lorenz system and the complex permanent magnet synchronous motor.

Y. Zhao et al. in their paper proposed a semilocal convergence theorem for the inverse-free Jarratt method under new Holder conditions. In fact, a new error estimate has been attained. Finally, three examples were provided to show the application of their discussed theorem for numerical problem of solving nonlinear equations.

In the paper of R. Behl and S. S. Motsa, authors proposed some geometric variation of the fourth-order Ostrowski's method so as to obtain methods with eighth-order of convergence and four functional evaluations. In this way, their method is optimal in the sense of Kung-Traub and would be useful for a class of problems in approximation theory.

The authors in the paper titled "On a Derivative-Free Variant of King's Family with Memory" proposed derivative-free variants of the well-known King's family of methods for nonlinear equations. They designed the approximation of derivatives to be as accurate as possible so as to keep the rate of convergence four using the same number of function evaluations. Finally, an extension of the family as a method with memory possessing higher computational efficiency index has been attained.

The paper titled "On a Cubically Convergent Iterative Method for Matrix Sign" proposed an interesting interlink between solvers for nonlinear equations and their applications for computing matrix functions. In this paper, it was shown that the new scheme has global behavior with cubical rate of convergence. Finally, several examples were also included to show the applicability and efficiency of the proposed scheme and its reciprocal.

K. Muzhinji et al. presented the application of different smoothers and compared their effects in the overall performance of the multigrid solver. They studied the multigrid method with the following smoothers: distributed Gauss-Seidel, inexact Uzawa, preconditioned MINRES, and Braess-Sarazin type smoothers. Lastly, numerical results have been included to demonstrate the efficiency and robustness of the multigrid method and confirm the theoretical results.

Finally, B. Alkahtani presented the Homotopy Analysis Method (HAM) to obtain the analytical solutions of the general space-time fractional diffusion equation. The explicit solutions of the equations have been presented in the closed form by using initial conditions. Several examples were also discussed to confirm the method proposed in this paper.

Acknowledgments

The guest editors would like to express their gratitude to all of those who submitted papers for publication and to the many reviewers whose reports were essential for them.

Fazlollah Soleymani
Predrag S. Stanimirović
Juan R. Torregrosa
Hassan Saberi Nik
Emran Tohidi

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 460421, 12 pages
http://dx.doi.org/10.1155/2015/460421

Research Article

The Mixed Finite Element Multigrid Method for Stokes Equations

K. Muzhinji,[1] S. Shateyi,[1] and S. S. Motsa[2]

[1] *Department of Mathematics, University of Venda, Private Bag X5050, Thohoyandou 0950, South Africa*
[2] *Department of Mathematics, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 28 July 2014; Revised 8 September 2014; Accepted 8 September 2014

Academic Editor: Emran Tohidi

The stable finite element discretization of the Stokes problem produces a symmetric indefinite system of linear algebraic equations. A variety of iterative solvers have been proposed for such systems in an attempt to construct efficient, fast, and robust solution techniques. This paper investigates one of such iterative solvers, the geometric multigrid solver, to find the approximate solution of the indefinite systems. The main ingredient of the multigrid method is the choice of an appropriate smoothing strategy. This study considers the application of different smoothers and compares their effects in the overall performance of the multigrid solver. We study the multigrid method with the following smoothers: distributed Gauss Seidel, inexact Uzawa, preconditioned MINRES, and Braess-Sarazin type smoothers. A comparative study of the smoothers shows that the Braess-Sarazin smoothers enhance good performance of the multigrid method. We study the problem in a two-dimensional domain using stable Hood-Taylor Q_2-Q_1 pair of finite rectangular elements. We also give the main theoretical convergence results. We present the numerical results to demonstrate the efficiency and robustness of the multigrid method and confirm the theoretical results.

1. Introduction

This study considers the numerical solution of the large scale linear algebraic system arising from the discretization of the partial differential equations. The discretization is achieved by the finite element method. For positive definite linear systems, linked to the Poisson equations, the multigrid (MGM) methods are proven to be the most effective and fast methods [1, 2]. However it is more challenging for linear indefinite algebraic systems. In this paper we consider multigrid methods for solving linear indefinite algebraic system of equations arising from the mixed finite element discretization of the steady state Stokes problem:

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \Omega, \tag{1}$$

$$\operatorname{div} \mathbf{u} = \mathbf{0}, \quad \text{in } \Omega, \tag{2}$$

$$\mathbf{u} = 0, \quad \text{on } \partial\Omega, \tag{3}$$

where $\mathbf{u}$ is a velocity field, p represents pressure, and $\mathbf{f}$ is an external force field. The problem is considered with (1)–(3) defined on the domain $\Omega \subseteq \mathscr{R}^2$ with boundary $\partial\Omega$.

The main goal of this work is to construct and analyze numerical methods that produce an appropriate solution to the Stokes problem. The main thrust is to apply an iterative method, the multigrid method, to solve the linear system of equations that arise from the discretization of the Stokes equations. The MFEM applied to (1)–(3) with carefully chosen finite element spaces results in the algebraic system which must be solved. The velocity variable $\mathbf{u}$ together with the pressure variable p is the solutions of the system. We discretize the domain of the Stokes problem by the rectangular grids with a pair of conforming mixed finite element spaces that are inf-sup stable. In our experiment we use Hood-Taylor Q_2-Q_1 pair as used by [3]. The process produces a symmetric indefinite system of linear algebraic equations. In this paper we study an efficient solver for this system. This work on multigrid method has been motivated

by the need to effectively and efficiently solve large application problems. The multigrid method has been shown to be very efficient and successful in solving control problems [1, 2, 4–6] and elliptic partial differential equations [7–9] in an accurate and computationally efficient way. The multigrid method has been applied to problems discretized by the finite difference method and widely by finite element method [3, 8, 10–14]. The effectiveness of the multigrid method depends on the correct choice of the smoothers. Various smoothers have been suggested in literature weighted Jacobi, Gauss Seidel [11], Ilu [8], Vanka-type [9, 12, 13, 15], Braess-Sarazin-type ([13, 16–18]), Semi implicit method for pressure linked equations [15], SOR/Richardson [18], and inexact Uzawa [18]. It is the purpose of this study to apply the multigrid solver to the Stokes problem with the following iterative solvers as smoothers: Braess-Sarazin, inexact Uzawa, preconditioned MINRES, and the distributed Gauss Seidel. The inner solver of these smoothers can also be taken as the multigrid method for the definite subsystems. There is no work known where a comparative study is made on the effects of these four smoothers on the performance of the multigrid method for indefinite systems. The first step is to transform the continuous problem to the discrete system and apply the MFEM that produces the linear algebraic system on which the multigrid method is developed, analyzed, and finally numerically and computationally implemented.

The key features and ingredients of the multigrid method are smoothing and coarse grid correction that involves the intergrid transfers and a solution correction step. The main results of the work are the convergence of the multigrid method in calculating the velocity and pressure variables in an appropriate norm which is based on the smoothing and approximation properties [9, 18]. The rest of the paper is organized as follows. In Section 2 we give the discrete system of the Stokes problem by mixed finite element method. In Section 3 the iterative solution technique, the geometric method, and smoothers are outlined. The known theoretical convergence analysis results are also outlined. In Section 4 a numerical experimental and comparative analysis on the effects of smoothers on the performance multigrid method is presented and discussed and the conclusion is given.

2. The Stokes Discrete System

For the discretization of the Stokes equations in the domain Ω we need to transform the system (1)–(3) to the weak variational form. For the weak variational formulation of the Stokes equations we define the following solution and test spaces:

$$H^1(\Omega) := \left\{\mathbf{u} : \Omega \longrightarrow \mathscr{R} \mid \mathbf{u}, \nabla u \in L^2(\Omega)\right\},$$
$$H_0^1(\Omega) := \left\{\mathbf{v} : \mathbf{v} \in H^1 \mid \mathbf{v} = 0 \text{ on } \partial\Omega\right\}. \tag{4}$$

By multiplication of the first equation (1) with $\mathbf{v} \in H_0^1$ and the second equation (2) with $q \in L^2(\Omega)$, subsequently integrating over the domain Ω, applying the Gauss theorem, and incorporating the boundary condition (3), we obtain the variational form.

Find $\mathbf{u} \in H_0^1(\Omega)$ and $p \in L^2(\Omega)$ such that

$$a(\mathbf{u}, \mathbf{v}) - b(\mathbf{v}, p) = F(\mathbf{v}), \quad \forall \mathbf{v} \in H_0^1(\Omega),$$
$$b(\mathbf{u}, q) = 0, \quad \forall q \in L^2(\Omega), \tag{5}$$

where $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are continuous bilinear forms defined as

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}\, dx,$$
$$b(\mathbf{u}, q) = \int_\Omega (\operatorname{div}\mathbf{v})\, q\, dx, \tag{6}$$
$$F(\mathbf{v}) = \int_\Omega \mathbf{f} \cdot \mathbf{v}\, dx,$$

where $\nabla\mathbf{u} : \nabla\mathbf{v}$ represents a componentwise scalar product that is $\nabla u_x \cdot \nabla v_x + \nabla u_y \cdot \nabla v_y$ and $a : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathscr{R}$ and $b : H_0^1(\Omega) \times L^2(\Omega) \to \mathscr{R}$. The well-posedness follows from the coercivity of $a(\cdot,\cdot)$ in the Lax-Milgram theorem [19, 20] and partly from the inf-sup condition [7, 8, 16, 21–23]. Below is a sketch of the analysis of the existence uniqueness and stability of the solution $(\mathbf{u}, p) \in \mathbf{V} \times W = H_0^1(\Omega) \times L^2(\Omega)$ of mixed problem (5):

(i) the bilinear form $a(\cdot,\cdot)$ is bounded or continuous if

$$|a(\mathbf{u}, \mathbf{v})| \le \alpha \|\mathbf{u}\|_{\mathbf{V}} \|\mathbf{v}\|_{\mathbf{V}}, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{V},\ \alpha \in \mathscr{R}; \tag{7}$$

(ii) the bilinear form $a(\cdot,\cdot)$ is coercive on $V := H_0^1(\Omega)$; that is, there exists a positive constant α_1:

$$a(\mathbf{v}, \mathbf{v}) \ge \alpha_1 \|v\|_V^2,$$
$$\forall \mathbf{v} \in \mathbf{V} = \ker B = \left\{\mathbf{v} \in \mathbf{V} : b(\mathbf{v}, q) = 0\ \forall q \in W\right\}; \tag{8}$$

(iii) the bilinear form $a(\cdot,\cdot)$ is symmetric and nonnegative if

$$a(\mathbf{u}, \mathbf{v}) = a(\mathbf{v}, \mathbf{u}), \qquad a(\mathbf{v}, \mathbf{v}) \ge 0, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{V}; \tag{9}$$

(iv) the bilinear form $b(\cdot,\cdot)$ is bounded if

$$|b(\mathbf{u}, q)| \le \alpha_0 \|\mathbf{u}\|_{\mathbf{V}} \|q\|_W, \quad \forall \mathbf{u} \in \mathbf{V},\ q \in W,\ \alpha_0 \in \mathscr{R}; \tag{10}$$

(v) the bilinear form $b(\cdot,\cdot)$ satisfies the inf-sup condition; that is, there exists a constant β:

$$\inf_{0 \ne q \in W} \sup_{0 \ne \mathbf{v} \in \mathbf{V}} \frac{|b(\mathbf{v}, q)|}{\|\mathbf{v}\|_{\mathbf{V}} \|q\|_W} \ge \beta > 0. \tag{11}$$

For instance, in [22], it is shown that in our concrete case $b(\cdot,\cdot)$ fulfills the inf-sup condition; thus we can combine (i)–(v) to give the following theorem.

Theorem 1. *The variational problem* (5) *is uniquely solvable provided properties* (i)–(v) *are all satisfied.*

The proof relies on the closed range theorem and on the Lax-Milgram theorem. The details can be found in [22, 24].

2.1. The Mixed Finite Element Discretization. The mixed finite element discretization of the weak formulation of the Stokes equations produces a linear algebraic system of equations. The finite element method described here is based on [7, 16, 19, 21, 23, 25]. We will introduce the concept of mixed finite element methods. Details can be found in [7, 19–21, 25].

We assume that $\Omega \subseteq \mathscr{R}^2$. We define the finite dimensional spaces. Let W_h and $\mathbf{V}_h$ be subspaces of W and $\mathbf{V}$, respectively.

Now we can formulate a discrete version of problem (5).

Find a couple $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times W_h$ such that

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}_h) - b(\mathbf{v}_h, p_h) &= F(\mathbf{v}_h), \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \\ b(\mathbf{u}_h, q_h) &= 0, \quad \forall q_h \in W_h. \end{aligned} \tag{12}$$

The finite element discretization should satisfy the inf-sup condition. The following theorem shows that again the inf-sup condition is of major importance (for the proof we refer to [22]).

Theorem 2. *Assume that a is V_h-elliptic (with h independent ellipticity constant) and that there exists a constant $\beta > 0$ (independent of h) such that the discrete* inf-sup *condition*

$$\inf_{0 \neq q_h \in W_h} \sup_{0 \neq \mathbf{v}_h \in \mathbf{V}_h} \frac{|b(\mathbf{v}_h, q_h)|}{\|\mathbf{v}_h\|_{\mathbf{V}_h} \|q_h\|_{W_h}} \geq \beta > 0 \tag{13}$$

holds. Then the associated (discretized, steady state) Stokes problem has a unique solution $(\mathbf{u}_h, p_h)$, and there exists a constant β_1 such that

$$\begin{aligned} &\|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{V}} + \|p - p_h\|_W \\ &\quad \leq \beta_1 \left(\inf_{\mathbf{v} \in \mathbf{V}_h} \|\mathbf{u} - \mathbf{u}_h\|_{\mathbf{V}} \right) + \left(\inf_{\mathbf{q} \in \mathbf{W}_h} \|p - p_h\|_W \right). \end{aligned} \tag{14}$$

If the basis of W_h is given by $\{\psi_1, \ldots, \psi_m\}$ and of V_h is given by $\{\varphi_1, \ldots, \varphi_n\}$, then

$$\begin{aligned} \mathbf{u}_h &= \sum_{i=1}^{n_i} \mathbf{u}_i \cdot \varphi_i + \sum_{i=n_i+1}^{n_i+n_\partial} \mathbf{u}_i \cdot \varphi_i, \\ p_h &= \sum_{k=1}^{m} p_k \psi_k, \end{aligned} \tag{15}$$

where n_i is the number of inner nodes and n_∂ is the number of boundary nodes so the coefficients $\mathbf{u}_i : i = n_i + 1, \ldots, n_i + n_\partial$ interpolate the boundary data and $n = n_i + n_\partial$. The mixed finite element entails partitioning of the solution domain Ω into quadrilaterals; in our case that is $\Omega = \cup_i \tau_i$ we denote a set of rectangular (square) elements by $T_h = \{\tau_1, \tau_2, \tau_3, \ldots\}$ and on each element τ_i and we denote the space $P_k(\tau_i)$ of degree less than or equal to k. There are a variety of finite element pairs whose effectiveness is through stabilization [26]. In this work we are going to use Hoods-Taylor Q_2-Q_1 square finite elements which are known to be stable.

We specify

$$\begin{aligned} \mathbf{V}_h &:= \left\{ \mathbf{u}_h \in \mathbf{V} \middle| \mathbf{u}_h|_{\tau_i} \in P_2(\tau_i), \forall \text{ elements } \tau_i \right\}, \\ \mathbf{W}_h &:= \left\{ p_h \in \mathbf{W} \middle| p_h|_{\tau_i} \in P_1(\tau_i), \forall \text{ elements } \tau_i \right\}. \end{aligned} \tag{16}$$

An element $(\mathbf{u}_h, p_h) \in W_h \times V_h$ is uniquely determined by specifying d components of $\mathbf{u}_h$ on the nodes and on the midpoints of the edges of the elements and the values of p_h on the nodes of the elements. The mixed finite element method results in the coupled linear algebraic system which has to be solved by the appropriate solvers. The resulting system is

$$\begin{bmatrix} A_h & B_h^T \\ B_h & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \mathbf{f}_h \\ g_h \end{bmatrix}; \tag{17}$$

with A_h being a block Laplacian matrix and B_h being the divergence matrix whose entries are given by

$$\begin{aligned} A &= \left[a_{ij}\right], \quad a_{ij} = \int_\Omega \left(\nabla \phi_i : \nabla \phi_j\right)_{i,j=1,\ldots,n}, \\ B &= \left[b_{ki}\right], \quad b_{ki} = -\int_\Omega \left(\psi_k \nabla \cdot \phi_i\right)_{i=1,\ldots,n;\, k=1,\ldots,m}. \end{aligned} \tag{18}$$

The entries of the right hand side vector are

$$\begin{aligned} \mathbf{f} &= \left[\mathbf{f}_i\right], \quad \mathbf{f}_i = \int_{\partial\Omega} \mathbf{f} \cdot \phi_i - \sum_{i=n+1}^{n+n_\partial} \mathbf{u}_i \int_\Omega \left(\nabla \phi_i : \nabla \phi_j\right), \\ g &= \left[g_k\right], \quad g_k = \sum_{i=n+1}^{n+n_\partial} \mathbf{u}_i \int_\Omega \left(\psi_k \nabla \cdot \phi_i\right). \end{aligned} \tag{19}$$

The linear algebraic system can be represented as

$$\mathscr{M} x = b, \tag{20}$$

where $\mathscr{M} := \left[\begin{smallmatrix} A_h & B_h^T \\ B_h & O \end{smallmatrix}\right]$, $x := \left[\begin{smallmatrix} \mathbf{u}_h \\ p_h \end{smallmatrix}\right]$, and $b := \left[\begin{smallmatrix} \mathbf{f}_h \\ g_h \end{smallmatrix}\right]$.

The solution vectors $(\mathbf{u}_h, p_h)$ from (15) are the mixed finite element solution. The system (17)–(19) is called the discrete Stokes problem.

The discretization and assembling of matrices are done by the MATLAB implementation of the mixed finite element method [8]. A_h is stiffness matrix resulting from the discretization of the Laplacian. The resultant coefficient matrix is large, sparse, indefinite and the system must be solved iteratively, in this case by multigrid solvers. The multigrid solver is a well known fast solver for the elliptic partial differential equations [2, 5].

3. Multigrid Method

The main focus of this section is the construction of the multigrid solver to find the approximate solution of (20) at the finest mesh/discretization. Let $(\mathbf{V}_l \times W_l)$ be a sequence of subspaces of the finite dimensional subspaces W_h and V_h defined on sequence of grids $l \in \{0, 1, 2, 3, \ldots, l_{\max}\}$ with mesh sizes $h_0, h_1, h_2, \ldots, h_l$ with $h_{l+1} := (1/2)h_l$. We define a hierarchy/family of nested finite element subspaces for the velocity and pressure:

$$\begin{aligned} &(V_l \times W_l) \subset (V_{l+1} \times W_{l+1}) \subset (V_h \times W_h) \subset (V \times W) \\ &\qquad = H_0^1(\Omega) \times L^2(\Omega), \end{aligned} \tag{21}$$

where $(V_{l+1} \times W_{l+1})$ subspace which corresponds to Ω_{l+1} is the refinement of Ω_l with subspace $(V_l \times W_l)$ such that

$\Omega_l \subset \Omega_{l+1} \subset \Omega$. At the discrete level with the defined discrete spaces and bases, the linear algebraic system is defined by

$$\mathcal{M}_l x_l = b_l, \tag{22}$$

where $\mathcal{M}_l := \begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix}$, $x_l := [\mathbf{u}_l \quad p_l]$, and $b_l := \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix}$.

The main goal is to find the pair $x_l = (\mathbf{u}_l, p_l)$ of the discrete velocity and the discrete pressure variables at the finest level l.

Now we introduce the multigrid iteration for solving the discretized equation (22) on grid l. We define the multigrid algorithm at level l as $\mathrm{MGM}_l(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}}, \mathbf{u}_l^{\mathrm{old}}, p_l^{\mathrm{old}}, \mathbf{f}_l, g_l, m_1, m_2)$, where

(i) $(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}})$ is the output of velocity and pressure after one step of the multigrid algorithm at level l;

(ii) $\mathbf{u}_l^{\mathrm{old}}$ is the input velocity at level l;

(iii) p_l^{old} is the input pressure at level l;

(iv) $R_{\mathbf{u},l,l-1}$ and $R_{p,l,l-1}$ are restriction operators for velocity and pressure, respectively, from level l to level $l-1$;

(v) $P_{\mathbf{u},l-1,l}$ and $P_{p,l-1,l}$ are prolongation operators for velocity and pressure, respectively, from level $l-1$ to level l.

Algorithm 3 (multigrid algorithm).

$$\mathrm{MGM}_l\left(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}}, \mathbf{u}_l^{\mathrm{old}}, p_l^{\mathrm{old}}, \mathbf{f}_l, g_l, m_1, m_2\right) \tag{23}$$

if $l = 0$ (coarsest grid)

$$\begin{aligned} &\begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_l^{\mathrm{new}} \\ p_l^{\mathrm{new}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix} \\ &= \mathrm{MGM}_l\left(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}}, \mathbf{u}_l^{\mathrm{old}}, p_l^{\mathrm{old}}, \mathbf{f}_l, g_l, m_1, m_2\right) \end{aligned} \tag{24}$$

else $l > 0$ define $\mathrm{MGM}_l(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}}, \mathbf{u}_l^{\mathrm{old}}, p_l^{\mathrm{old}}, g_l, \mathbf{f}_l, m_1, m_2)$

(1) Pre-Smoothing: Smoothing operator $\mathcal{S}$ starting with, $(\mathbf{u}_l^{\mathrm{old}}, p_l^{\mathrm{old}})$ with m_1 smoothing steps, producing $(\widetilde{\mathbf{u}}_l, \widetilde{p}_l)$,

$$\begin{bmatrix} \widetilde{\mathbf{u}}_l \\ \widetilde{p}_l \end{bmatrix} = \begin{bmatrix} \mathbf{u}^{\mathrm{old}} \\ p^{\mathrm{old}} \end{bmatrix} - \mathcal{S}^{-1}\left(\begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_l^{\mathrm{old}} \\ p_l^{\mathrm{old}} \end{bmatrix} - \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix}\right) \tag{25}$$

(a) defect/residual

$$\begin{aligned} \mathbf{r}_l &= \mathbf{f}_l - \left(A_l \widetilde{\mathbf{u}}_l + B_l^T \widetilde{p}_l\right), \\ d_l &= g_l - B_l \widetilde{\mathbf{u}}_l \end{aligned} \tag{26}$$

(2) restrict the defect

$$\begin{aligned} \mathbf{r}_{l-1} &= R_{\mathbf{u},l,l-1} \mathbf{r}_l, \\ d_{l-1} &= R_{p,l,l-1} d_l, \end{aligned} \tag{27}$$

(3) approximate solution

$$\begin{bmatrix} A_{l-1} & B_{l-1}^T \\ B_{l-1} & O \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{v}}_{l-1} \\ \widetilde{q}_{l-1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_{l-1} \\ d_{l-1} \end{bmatrix} \tag{28}$$

(4) applying one/two iterations of MGM_{l-1} at the recursive call:

(a) apply μ steps of MGM_{l-1}

(b) Set $\mathbf{v}_{l-1}^0 = 0$

(c) Set $q_{l-1}^0 = 0$

(d) compute

for $\mu = 1 : 2$

$$\begin{aligned} &\left(\widetilde{\mathbf{v}}_{l-1}, \widetilde{q}_{l-1}\right) \\ &= \mathrm{MGM}_{l-1}\left(\widetilde{\mathbf{v}}_{l-1}, \widetilde{q}_{l-1}, \mathbf{v}_{l-1}^0, q_{l-1}^0, \mathbf{r}_{l-1}, d_{l-1}, m_1, m_2\right); \end{aligned} \tag{29}$$

end

(5) *Correction Step* define the new iterate:

$$\begin{aligned} \mathbf{u}_l^* &:= \widetilde{\mathbf{u}}_l - P_{\mathbf{u},l-1,l} \widetilde{\mathbf{v}}_{l-1}, \\ p_l^* &:= \widetilde{p}_l - P_{p,l-1,l} \widetilde{q}_{l-1}. \end{aligned} \tag{30}$$

Postsmoothing. Starting with $(\mathbf{u}_l^*, p_l^*)$ perform m_2 smoothing steps using smoothing operator $\mathcal{S}$ to produce $(\mathbf{u}_l^{\mathrm{new}}, p_l^{\mathrm{new}})$:

$$\begin{bmatrix} \mathbf{u}^{\mathrm{new}} \\ p^{\mathrm{new}} \end{bmatrix} = \begin{bmatrix} \widetilde{\mathbf{u}}_l^* \\ \widetilde{p}_l^* \end{bmatrix} - \mathcal{S}^{-1}\left(\begin{bmatrix} A_l & B_l^T \\ B_l & O \end{bmatrix} \begin{bmatrix} \widetilde{\mathbf{u}}_l^* \\ \widetilde{p}_l^* \end{bmatrix} - \begin{bmatrix} \mathbf{f}_l \\ g_l \end{bmatrix}\right). \tag{31}$$

The multigrid method described above belongs to a class of optimal order methods for solving linear systems emanating from the discretization techniques like the finite element method. Its known convergence speed does not deteriorate when the discretization is refined whereas classical iterative solvers slow down for the decreasing mesh size [1, 2, 5, 6]. The starting point of the multigrid concept is the observation that classical iteration methods have some smoothing properties. The operator $\mathcal{S}$ represents such methods; in this study it represents Braess-Sarazin, inexact Uzawa, distributed Gauss Seidel, and the preconditioned minimum residual method. These methods are characterized by poor/slow global convergence rates and for errors whose length scales are comparable to mesh sizes, they provide rapid damping leaving behind smooth, longer wave length errors. These smooth parts of the error are responsible for the poor convergence. A geometric multigrid method involves a hierarchy of meshes and related discretization. A quantity that is smooth on a certain grid can also be approximated on a coarser grid. Low frequency error components can be effectively reduced by a coarse grid correction procedure. Since the action of a smoothing iteration leaves only smooth error components, it is possible to represent them as the solution of an appropriate coarser system. Once this coarser problem is solved, the solution is interpolated back to the fine grid to correct the fine approximation for its low frequency errors.

The most essential ingredients of the multigrid method are the smoothing operator, for which using a wrong smoother will destroy the efficiency of the entire multigrid method, and the coarse grid correction which involves the prolongation and the restriction operators. In multigrid methods we have to transform information from one grid to another and for that purpose we use prolongations and restrictions operators. Restriction transfers values from fine grid to the next coarse grid. Prolongation transfers values from the coarse grid to the next fine grid.

Next we discuss the key components of the multigrid method.

(a) Intergrid transfer operators: the intergrid transfer operators are the restriction and prolongation between different grid levels. The restriction operator maps the residual from the finer grid to a coarser grid while the prolongation operator transfers vectors from coarse grid to fine grid. The restriction between levels l and $l-1$ is defined by

$$R_{(l,l-1)} := \begin{pmatrix} R_{(\mathbf{u},l,l-1)} & 0 \\ 0 & R_{(p,l,l-1)} \end{pmatrix}, \tag{32}$$

where the restriction operators $R_{(\mathbf{u},l,l-1)} : \mathscr{R}^{n_l} \to \mathscr{R}^{n_l-1}$ and $R_{(p,l,l-1)} : \mathscr{R}^{m_l} \to \mathscr{R}^{m_{l-1}}$ for velocity and pressure, respectively. The prolongation between levels $l-1$ and l is defined again as

$$P_{(l-1,l)} := \begin{pmatrix} P_{(\mathbf{u},l-1,l)} & 0 \\ 0 & P_{(p,l-1,1)} \end{pmatrix}, \tag{33}$$

where the prolongation operators $P_{(\mathbf{u},l-1,l)} : \mathscr{R}^{n_{l-1}} \to \mathscr{R}^{n_l}$ and $P_{(p,l-1,l)} : \mathscr{R}^{m_{l-1}} \to \mathscr{R}^{m_l}$ are representations of the following relations $\mathbf{V}_{l-1} \subset \mathbf{V}_l$ for the quadratic interpolation of the velocity (Q_2) and $W_{l-1} \subset W_l$ for the linear interpolation of the pressure (Q_1).

(b) Coarse grid correction: the other key ingredient of the multigrid method is the coarse grid correction. In the multigrid solution process we need to solve the problem at the finest define level $l = l_{\max}$. The problem is defined on the coarser grid levels and on the coarsest grid level the problem is solved exactly. There are very few situations in which a grid can be coarsened to the extent that it is not practical to solve the problem using a direct method but iteratively. In this work the iterative solver used as a smoother is applied to solve the problem at the coarsest level.

3.1. The Smoothers. The most crucial part is the proper choice of a smoothing technique. Usually, the well-known smoothing iterations for the scalar problems (damped Jacobi or Gauss-Seidel relaxation) are not appropriate for saddle point problems or are even not defined, for example, in saddle point systems like (22). There are natural ways to generalize scalar smoothing schemes to systems of PDEs. The smoothing process is the main ingredient of the multigrid method. The convergence of the multigrid method is influenced by the smoothing process [11, 14, 18]. We perform a number of iterations of an iterative solver to smooth the residual. The main goal is to compare the effectiveness of different iterative schemes as smoothers of the multigrid methods. On each level of a multigrid method, a system involving operator $\mathcal{S}$ has to be solved approximately. The smoother dumps out highly oscillating error modes of the systems. In this paper we consider the following smoothing process:

$$\begin{pmatrix} \mathbf{u}_l^{i+1} \\ p_l^{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_l^{i} \\ p_l^{i} \end{pmatrix} - \mathcal{S}_l^{-1} \left[\begin{pmatrix} A_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \mathbf{u}_l^{i} \\ p_l^{i} \end{pmatrix} - \begin{pmatrix} \mathbf{f}_l \\ 0 \end{pmatrix} \right]. \tag{34}$$

Several smoothers have been proposed and applied in literature. Brandt [4] advocates for the use of the distributed Gauss Seidel smoothing. The Vanka-type smoother is widely used with a coupled Gauss Seidel scheme [13, 14] that introduces the idea of transforming smoothers and combines with incomplete factorization to develop an efficient smoothing. John and Tobska [14] and Pernice [15] used the Braess-Sarazin-type smoother with the Schur complement schemes as smoothers which exhibit wonderful smoothing properties. The following algorithms describe the iterative schemes that are used as smoothers in this study.

3.1.1. Braess-Sarazin-Type Smoother. The Braess-Sarazin smoothers proposed in [17] and used in [13, 18] solve a large saddle point problem in each smoothing step. This Braess-Sarazin or SIMPLE-type iteration uses $\left(\begin{smallmatrix} \widehat{A} & B^T \\ B & O \end{smallmatrix}\right)$ as a smoother for the saddle point problem (22). The smoother as presented in [17] and generalised in [18] consisted of constant application of the smoothing iteration:

$$\begin{pmatrix} \mathbf{u}_l^{i+1} \\ p_l^{i+1} \end{pmatrix} = \begin{pmatrix} \mathbf{u}_l^{i} \\ p_l^{i} \end{pmatrix} - \begin{pmatrix} \widehat{A}_l & B_l^T \\ B_l & O \end{pmatrix}^{-1} \left[\begin{pmatrix} A_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \mathbf{u}_l^{i} \\ p_l^{i} \end{pmatrix} - \begin{pmatrix} \mathbf{f}_l \\ g_l \end{pmatrix} \right] \tag{35}$$

with $\widehat{A}_l = \alpha\, \mathrm{diag}(A_l)$ and $\alpha = 2$ given. The smoothing Braess-Sarazin iteration (35) solves the auxiliary problem

$$\begin{pmatrix} \alpha\widehat{A}_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \widehat{\mathbf{u}}_l^{i} \\ \widehat{p}_l^{i} \end{pmatrix} = \begin{pmatrix} \mathbf{r}_l^{i} \\ s_l^{i} \end{pmatrix} \tag{36}$$

with $\mathbf{r}_l^i = A_l \mathbf{u}_l^i + B_l^T p_l^i - \mathbf{f}_l$ and $s_l^i = B_l \mathbf{u}_l^i - g_l$. Inherent in the system system (36) is the problem of the auxiliary pressure variable $\widehat{p}_l$

$$\widehat{S}_l \widehat{p}_l = B_l \widehat{A}_l^{-1} \mathbf{r}_l^i - \alpha B_l \mathbf{u}_l^i. \tag{37}$$

This system is solved approximatively by an iterative solver. From the system we get $\widehat{p}_l$ approximately which can be used to approximately determine $\widehat{\mathbf{u}}_l$ from

$$\alpha \widehat{A}_l \widehat{\mathbf{u}}_l = \mathbf{r}_l^i - B_l^T p_l^i. \tag{38}$$

3.1.2. Inexact Uzawa Type Smoothers. The variant of the inexact Uzawa iteration used as a smoother is outlined.

Algorithm 4. (1) For $i = 1$: smoothing steps.

(2) Compute the residual $\mathbf{r}_i = \mathbf{f} - A\mathbf{u}_i - B^T p_i$.

(3) Compute the residual $s_i = g - B\mathbf{u}_i$.
(4) Solve $\widetilde{A}\mathbf{w}_i = \mathbf{r}_i$.
(5) Solve $\widetilde{S}d_i = B^T\mathbf{w}_i - s_i$.
(6) Solve $\widetilde{A}\mathbf{w}_i = \mathbf{r}_i - Bd_i$.
(7) Update the velocity and pressure

$$\begin{pmatrix}\mathbf{u}_{i+1}\\ p_{i+1}\end{pmatrix} = \begin{pmatrix}\mathbf{u}_i\\ p_i\end{pmatrix} + \begin{pmatrix}\mathbf{w}_i\\ d_i\end{pmatrix}; \tag{39}$$

End.

Step (6) in the outline may be rearranged as $\mathbf{w}_i := \mathbf{w}_i - \widetilde{A}^{-1}(Bd_i)$ with $\widetilde{A}^{-1}(Bd_i)$ obtained as a by-product of step (5). This saves the application of $\widetilde{A}^{-1}$ at the end of every outer iteration and hence improves the efficiency of the algorithm. The other variants of the inexact Uzawa method are analysed in [26–28].

3.1.3. The Distributed Gauss Seidel Type Smoothers (DGS). The standard smoothing iteration schemes like Jacobi and Gauss Seidel smoothers are not applicable to the system (22) because of the nature of the coefficient matrix; particularly the zero block in the diagonal hampers the smoothing process. The distributive smoother transforms the vital operators to the main diagonal and applied as a decoupled smoother. The DGS was introduced in [4] is related to a successive application of standard Gauss Seidel applied to the matrix operator $\mathcal{M}$ (22) and $\mathcal{G} = \begin{pmatrix} I_l & B_l^T \\ O & -B_lB_l^T \end{pmatrix}$ with $\mathcal{M}\mathcal{G} = \begin{pmatrix} A_l & B_l^T \\ B_l & B_lB_l^T \end{pmatrix}$. We solve the transformed residual equation

$$\begin{pmatrix}\widehat{A}_l & B_l^T\\ B_l & \widehat{A}_p\end{pmatrix}\begin{pmatrix}\mathbf{w}_i\\ q_i\end{pmatrix} = \begin{pmatrix}\mathbf{r}_\mathbf{u}\\ r_p\end{pmatrix} \tag{40}$$

with $\widehat{A}$ and $\widehat{A}_p$ being invertible approximations of A and $A_p := BB^T$, respectively. A single iteration with the update through a distributive matrix $\mathcal{G}$ is performed by the following algorithm.

Algorithm 5 ($[\mathbf{u}^{i+1}, p^{i+1}] \leftarrow \mathrm{DGS}(\mathbf{u}^i, p^i)$). (1) Smooth momentum equations

$$\mathbf{w} = \mathbf{u}^i + \widehat{A}^{-1}\left(\mathbf{f} - A\mathbf{u}^i - B^Tp^i\right). \tag{41}$$

(2) Smooth the transformed continuity equation

$$q = \widehat{A}_p^{-1}\left(g - B\mathbf{w}\right). \tag{42}$$

(3) Transform the correction back to the original variables

$$\begin{aligned}\mathbf{u}^{i+1} &= \mathbf{w} + B^Tq,\\ p^{i+1} &= p^i - BB^Tq.\end{aligned} \tag{43}$$

The DGS has been widely used as a smoother for the finite difference discretization. In this paper the DGS type smoothers are used for finite element discretization of the Stokes problem.

3.1.4. The Preconditioned Minimum Residual Smoother. The preconditioned minimum residual method is a Krylov subspace method for solving symmetric indefinite systems and uses popular block preconditioners. This method is used as a smoother for the multigrid method of the Stokes problem in this paper. For the Stokes equations, the classical block-diagonal preconditioner for MINRES method [8] is

$$P = \begin{pmatrix}\widehat{A} & 0\\ 0 & \widehat{S}\end{pmatrix} \tag{44}$$

with $\widehat{S} = B\widehat{A}^{-1}B^T$. The block preconditioning requires the solution of two systems of equations with matrices $\widehat{A}$ and $\widehat{S}$ at each MINRES iteration. If P^{-1} is computed exactly, the preconditioned Krylov methods converge in two or three steps [10]. For practical implementations, the Schur complement $\widehat{S}$ is replaced by the mass matrix M_p of the pressure space. For discontinuous pressure space, M_p is block diagonal and easy to invert. For continuous pressure space, say Q_1, the mass matrix M_p can be further replaced by its diagonal matrix [8].

3.2. Multigrid Convergence. The convergence analysis of the multigrid method relies on the two properties, namely, the approximation and the smoothing. The general convergence rates are independent of h (the mesh size), l is the level of discretization, and m_1 and m_2 are the number of pre- and postsmoothing iterations [1, 12, 18]. The results for the convergence of the multigrid method for the scalar elliptic problems cannot apply to the Stokes equations. We provide a snapshot of the available convergence results of the multigrid method for Stokes equations. The ideas presented in this paper are based on the work in [12, 16, 18]. An iteration of single multigrid step consists of a combination of smoothing step and a coarse grid correction step. We will consider the multigrid convergence with the Braess-Sarazin smoother with $\mathcal{S}_l$ being the iteration matrix of the smoother (34) and the $\mathcal{M}_l$ being the Stokes stiffness matrix in (22). The operator P_l and its adjoint R_l are intergrid transfer operators, prolongation, and the restriction, respectively. The convergence analysis of the multigrid method begins with the analysis of the two-grid method, with m_1 and m_2 the pre- and post-smoothing steps, respectively, applied to (22) results in the iteration matrix

$$L_l = \mathcal{S}_l^{m_1}\left(I_l - P_l\mathcal{M}_{l-1}^{-1}R_l\mathcal{M}_l\right)\mathcal{S}_l^{m_2}. \tag{45}$$

The key point on the analysis of the multigrid method is that the error can be split into two components. That is the one produced by the smoothing process and the one produced by the coarse grid correction. The coarse grid error consists of the low frequency components and the smoothing consists of the high frequency components of error. The ability to cope with the low frequency components is called the approximation property and with the second is called the smoothing property. For the analysis of the multigrid convergence [2, 29] used the framework based on the smoothing and approximation property. For analysis we

define the following norms, Euclidean norm by $\|\cdot\|$, and on $\mathcal{R}^{n_l+m_l}$ the following norm is applied:

$$\left\|\begin{pmatrix}\mathbf{u}_l\\ p_l\end{pmatrix}\right\|_h^2 := \|\mathbf{u}_l\|^2 + h_l^2\|p_l\|^2 = \left\|\Theta_l\begin{pmatrix}\mathbf{u}_l\\ p_l\end{pmatrix}\right\|_h^2 \quad \text{with } \Theta_l := \begin{pmatrix}I_{n_l} & O\\ O & I_{m_l}\end{pmatrix}. \tag{46}$$

Furthermore we introduce

$$\widehat{\mathcal{M}}_l := \Theta_l^{-1}\mathcal{M}_l\Theta_l^{-1}, \qquad \widehat{\mathcal{S}}_l := \Theta_l^{-1}\mathcal{S}_l\Theta_l^{-1}. \tag{47}$$

Using the norms defined above and taking $m_1 = m$ and $m_2 = 0$ above we obtain

$$\begin{aligned}\|L_l\|_h &= \left\|\Theta_l\left(\mathcal{M}_l^{-1} - P_l\mathcal{M}_{l-1}^{-1}R_l\right)\Theta_l\Theta_l^{-1}L_l\mathcal{S}_l^m\Theta_l^{-1}\right\| \\ &\le \left\|\Theta_l\left(\mathcal{M}_l^{-1} - P_l\mathcal{M}_{l-1}^{-1}R_l\right)\Theta_l\right\|\left\|\widehat{\mathcal{M}}_l\widehat{\mathcal{S}}_l\right\|.\end{aligned} \tag{48}$$

The theorems below state the two properties and the multigrid convergence. For detailed proof we refer to [12, 18].

Theorem 6 (approximation property). *Assume that Ω is such that the problem* (5) *is H^2-regular. Let $\mathcal{M}_l$ be the coefficient stiffness matrix and R_l and P_l the prolongation and the restriction operators. Then there exists a constant $C_{\mathcal{M}}$ independent of l and using h-scaling induced by $\mathcal{M}_l$ then*

$$\left\|\Theta_l\left(\mathcal{M}_l^{-1} - P_l\mathcal{M}_{l-1}^{-1}R_l\right)\Theta_l\right\|_h \le C_{\mathcal{M}}\left\|\mathcal{M}_l^{-1}\right\|_2, \tag{49}$$

where $C_{\mathcal{M}} = Ch^2$.

The smoothing property is dependent on the smoother used. It varies from one smoother to another. In this work we used the Braess-Sarazin in which we solve the system (37) exactly and sufficiently accurate inexact inner solver.

Theorem 7 (smoothing property). *Let $\mathcal{M}_l$ be the coefficient stiffness matrix and the smoothing operator $\mathcal{S}_l$. Then*

$$\left\|\widehat{\mathcal{M}}_l\widehat{\mathcal{S}}_l^m\right\| \le g(m)\left\|\mathcal{M}_l\right\|, \tag{50}$$

where $g(m) = ch_l^2/(m-1)$ for $m \ge 2$ and $g(m)$ is a decreasing function with $\lim_{m\to\infty} g(m) = 0$.

Combining the approximation property Theorem 6 with the smoothing property Theorem 7 produces a two-grid convergence result.

Theorem 8. *Assume that $m_2 = 0$ and that Ω is such that the problem* (5) *is H^2-regular. Then for the two-grid method the following holds:*

$$\left\|\mathcal{M}_l\right\|_h \le \frac{C_{\mathcal{M}}}{m-1}, \quad m \ge 2 \tag{51}$$

with a constant $C_{\mathcal{M}}$ independent of l and m.

Using this two-grid contraction number bound the multigrid W-cycle method convergence results can be derived using ideas in [1, 2].

4. Numerical Results

In this section we present the numerical solution of classical Stokes problem (1)–(3) using the solver presented above. The solver is denoted by MGM (Algorithm 3). We present the results of this method as outlined above to run the traditional test problem, the driven cavity flow problem [11, 12, 27, 28]. It is a model of the flow in a square cavity (the domain is $\Omega_\square$) with the top lid moving from left to right in our case the regularized cavity model $\{y = 1 : -1 \le x \le 1 \mid u_x = 1 - x^4\}$ [11]. The Dirichlet no-slip boundary condition is applied on the side and bottom boundaries. The mixed finite element method was used to discretize the cavity domain $\Omega = (-1, 1)^2$.

We pay particular attention to the computational performance of the multigrid method on the system (22) at different grid levels. We compare the effectiveness of different smoothing/relaxation methods in the performance of the multigrid method and different approximations for the preconditioners $\widehat{A}$ and $\widehat{S}$ of the smoothers. The following setup of the smoothers listed is considered.

(i) Distributed Gauss Seidel (DGS) smoother: we use one Gauss Seidel iteration for the evaluations of $\widehat{A}$ and one Gauss Seidel iteration for the computation of $\widetilde{A}_p$. The method becomes DGSMG.

(ii) Inexact Uzawa smoother (IUzawa): the two cases are considered for the evaluation of the preconditioners. Firstly, the approximation $\widehat{A} = \text{diag}(A)$ and one $\nu(1,1)$-cycle is used to approximate Schur compliment matrix $B\widehat{A}B^T$. The second case is to use one $\nu(1,1)$-cycle for both evaluations of $\widehat{A}$ and $B\widehat{A}B^T$. The method becomes IUZAWAMG.

(iii) Braess-Sarazin smoother (B-S): the two cases are conspired for the evaluation of the preconditioners. Firstly, the approximation $\widehat{A} = \text{diag}(A)$ and one $\nu(1,1)$-cycle is used to solve the approximate Schur compliment matrix $B\widehat{A}B^T$. The second case is to use one $\nu(1,1)$-cycle for both evaluations of $\widehat{A}$ and $B\widehat{A}B^T$. The method becomes B-SMG.

(iv) PMINRES smoother: the first case is to use diagonal preconditioner for $\widehat{A}$ and $\widehat{S}$ and the second case is one $\nu(1,1)$-cycle for committing the inversion of the Laplacian operator for velocity as one $\nu(1,1)$ cycle is used to approximate the Schur compliment using the pressure mass matrix to accelerate the MINRES. The method becomes PMINRESMG.

The comparison is made on the performance of the multigrid schemes with different smoothers (i)–(iv) and cases involving different approximations of preconditioners in terms of iterative counts and CPU time. The numerical treatment is given to the discrete Stokes problem which resulted from the mixed finite Hood-Taylor stable elements consisting of biquadratic elements for the velocities and bilinear elements for the pressure, on a uniform grid. Implementation of our algorithms was performed on a Windows 7 platform with 2.13 GHz speed intel dual core processor by using MATLAB 7.14

Table 1: Refinement levels and number of nodes (n_l: number of velocity unknowns (×2) and m_l : number of pressure unknowns).

Refinement level (l)	1	2	3	4	5	6	7	8
Mesh size (h_l)	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{1}{256}$
Velocity nodes (n_l)	9	25	81	289	1089	4425	16641	66049
Pressure nodes (m_l)	4	9	25	81	289	1089	4425	16641

Table 2: Number of iterations and CPU time for Braess-Sarazin (2diag(A), ν-cycle(1, 1)) multigrid V-cycle at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-V-cycle			
	$\nu(1,1)$ Iter(cpu time (sec))	$\nu(2,2)$ Iter(cpu time (sec))	$\nu(3,3)$ Iter(cpu time (sec))	$\nu(4,4)$ Iter(cpu time (sec))
$\frac{1}{16}$	21 (7.1$e-02$)	15 (7.3$e-02$)	9 (2.1$e-02$)	6 (8.6$e-02$)
$\frac{1}{32}$	22 (5.4$e-01$)	12 (6.4$e-01$)	10 (1.6$e-01$)	7 (7.7$e-01$)
$\frac{1}{64}$	22 (1.6$e-01$)	16 (5.14$e-01$)	11 (1.6)	7 (5.3$e-01$)
$\frac{1}{128}$	22 (1.64)	16 (2.58)	11 (2.17)	7 (4.45)

Table 3: Number of iterations and CPU time for Braess-Sarazin (2diag(A), $\nu(1,1)$) multigrid W-cycle at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-W-cycle			
	$\nu(1,1)$ Iter(cpu time (sec))	$\nu(2,2)$ Iter(cpu time (sec))	$\nu(3,3)$ Iter(cpu time (sec))	$\nu(4,4)$ Iter(cpu time (sec))
$\frac{1}{16}$	16 (8.3$e-02$)	11 (8.4$e-02$)	7 (3.5$e-02$)	5 (9$e-02$)
$\frac{1}{32}$	17 (6$e-01$)	12 (6$e-01$)	7 (1.2$e-01$)	5 (7.1$e-01$)
$\frac{1}{64}$	17 (5$e-01$)	13 (5.1$e-01$)	8 (1.89)	6 (5.6$e-01$)
$\frac{1}{128}$	16 (1.98)	13 (3.24)	8 (2.13)	6 (5.78)

programming language and the MATLAB built-in Minres functions are used for the smoother. For the discretization we start with a uniform square grid with $h_0 = 1/2$ and we apply regular refinements to this starting discretization to obtain the finest grid level. The discretized equations are solved using the multigrid iteration with the W-cycle and V-cycle and m_1 and m_2 being presmoothing and postsmoothing steps, respectively. The smoothers are determined by specifying approximations $\widehat{A}$ and $\widehat{S}$ as highlighted in (i)–(vi) and in all cases where one ν-cycle inner multigrid iteration is used with n_1 and n_2 being Gauss Seidel iteration steps of the presmoothing and the postsmoothing, respectively.

In this work we use the structured mesh and regular refinements. The finite element matrices on the rectangular grids are assembled and the meshes are generated by the MATLAB IFISS toolbox [3] in a hierarchy of grids which are produced by successive regular refinements. We need to choose the coarse mesh (the starting mesh), the finest mesh which corresponds to the maximum level of refinement on which the final approximate solution is considered. The assembled matrices are stored for each refinement level for the system (22). Table 1 shows an example of the refinement levels, we use the coarsest (starting) level to have 9 nodes for velocity and 4 nodes for pressure variables (level 1) but we start the computation at level 2.

Table 1 shows the refinement levels and the number of grid points (nodes) for each level.

The zero initial guess is chosen for all the tests. In all the tests the iterations are repeated until the tolerance $\|R_i\|/\|R_0\| < 10^{-6}$, where $R_i = \begin{pmatrix} \mathbf{f}_i \\ g_i \end{pmatrix} - \begin{pmatrix} A_l & B_l^T \\ B_l & O \end{pmatrix} \begin{pmatrix} \mathbf{u}_i \\ p_i \end{pmatrix}$ is satisfied. The schemes converge if the stopping criteria are satisfied. The results show the first case of the evaluation, the preconditioners of the smoothers $\widehat{A}$ and $\widehat{S}$ with $\widehat{A} = 2\,\mathrm{diag}(A)$, and for all cases evaluation of $\widehat{S}$ by one ν-cycle inner multigrid iteration with n_1 and n_2 being Gauss Seidel iteration of the presmoothing and postsmoothing steps, respectively.

Tables 2 and 3 show the number of iterations and computing time to demonstrate the effects of different V-cycle (1, 2, 3, and 4) and W-cycle (1, 2, 3, and 4) being

Table 4: Number of iterations and CPU time for multigrid (V-cycle) with different smoothers and smoother preconditioner approximations at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-V-cycle(3, 3)			
	DGS	IUzawa	Braess-Sarazin	PMINRES
	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))
$\frac{1}{16}$	18 ($8.6e-02$)	13 ($6.3e-02$)	9 ($2.1e-02$)	14 ($7.1e-02$)
$\frac{1}{32}$	19 ($7.4e-01$)	14 ($2.0e-01$)	10 ($1.6e-01$)	16 ($6.5e-01$)
$\frac{1}{64}$	19 (3.12)	14 (1.64)	11 (1.6)	16 (4.898)
$\frac{1}{128}$	19 (8.66)	14 (3.2)	11 (2.17)	16 (7.97)

Table 5: Number of iterations and CPU time for multigrid (W-cycle) with different smoothers and smoother preconditioner approximations at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-W-cycle(3, 3)			
	DGS	IUzawa	Braess-Sarazin	PMINRES
	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))
$\frac{1}{16}$	15 ($9.8e-02$)	9 ($7.1e-02$)	7 ($3.5e-02$)	15 ($8.5e-02$)
$\frac{1}{32}$	16 ($8.3e-01$)	10 ($3.4e-01$)	8 ($1.2e-01$)	14 ($5.1e-01$)
$\frac{1}{64}$	17 (5.76)	9 (2.56)	8 (1.89)	14 (6.43)
$\frac{1}{128}$	17 (9.87)	9 (4.32)	8 (2.13)	14 (8.55)

pre- and postsmoothing steps with Braess-Sarazin (B-S) smoother (diag(A), $\nu(1, 1)$). We compare the performance of the V-cycle and W-cycle multigrid iterations with various smoothing steps at different grid levels using one of the smoothers, Braess-Sarazin.

From Tables 2 and 3 we observe that the number of iterations decreases when the smoothing steps decrease and the CPU time increases as expected with the increase in smoothing steps.

Tables 4 and 5 show the numerical results obtained of the multigrid solver at different grid levels. The number of V-cycle and W-cycle multigrid iterations and CPU time are shown, respectively. All the results presented underline the efficiency of the multigrid solver to indefinite systems of equations. In both tables we present results of the four studied smoothers of the multigrid solver. In Tables 4 and 5 we choose the approximation of the smoother preconditioners as $\widehat{A} = 2\,\mathrm{diag}(A)$ and $\widehat{S} = \nu$-cycle$(1, 1)$ for Braess-Sarazin, IUzawa, and PMINRES. For the DGS we use the one Gauss Seidel for both A and A_p. We fix the number of smoothing steps to (3,3) for all the results in the tables.

Comparing the performance of the V-cycle and W-cycle multigrid solver, we observe that the smoothers have different effects on the performance of the multigrid solver. The multigrid solver is optimal and the iterations are bound for all the grid levels. In Tables 4 and 5 we compare all smoothers and we observe that the Braess-Sarazin smoother leads to faster convergence of the multigrid with fewer iterations and less CPU ahead of other smoothers. The inexact Uzawa did not disappoint in relaxing the error but the DGS and the PMINRES lead to more iterations and computing times. The other observation in Tables 4 and 5 is that the W-cycle converges in fewer iterations than the V-cycle though it has more computing times for all smoothers.

In Tables 6 and 7 we use different approximations for the preconditioner of the smoothers of the multigrid V-cycle and W-cycle, respectively. In applying the preconditioners, we approximate the preconditioner $\widehat{A}$ of the Laplacian stiffness and sparse matrix A and $\widehat{S}$ by a geometric multigrid $\nu(1, 1)$-cycle method (A_{mg}). The multigrid is a well-known fast solver for such systems. The multigrid solver is an inner iteration of the smoothers. The results in Tables 6 and 7 also show that the one iteration of the multigrid ν-cycle is a suitable approximation of the smoothers since the multigrid solver has improved iterations from the ones in Tables 4 and 5. In both tables the multigrid method is optimal in solving the indefinite systems and the number of iterations is bounded for all smoothers independent of the mesh size or grid level.

Table 8 shows the changes in the estimated a posteriori errors for regularized driven cavity flow using Q_2-Q_1 approximation for the flow: using the strategy built in IFISS [3, 8] that, for every element error, the local error estimation is given by the combination of the energy norm of the velocity error and the L_2 norm of the divergence error; that is,

$$\eta_T^2 := \left\|\nabla \mathbf{e}_T\right\|_T^2 + \left\|R_T\right\|_T^2, \tag{52}$$

Table 6: Number of iterations and CPU time of iterations for multigrid (V-cycle) with different smoothers and using one ν-cycle multigrid preconditioner approximation (for both $\widehat{A}$ and $\widehat{S}$) at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-V-cycle(3, 3)			
	DGS	IUzawa	Braess-Sarazin	PMINRES
	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))
$\frac{1}{16}$	14 ($4e-01$)	10 ($3.4e-02$)	6 ($1.3e-02$)	13 ($4.4e-02$)
$\frac{1}{32}$	15 ($2.1e-01$)	10 ($2.3e-01$)	7 ($1.5e-01$)	14 ($5.4e-01$)
$\frac{1}{64}$	16 (2.11)	11 (2.01)	8 (1.01)	14 ()
$\frac{1}{128}$	16 (4.56)	11 (3.67)	8 (2.02)	14 (9.01)

Table 7: Number of iterations and CPU time for multigrid (W-cycle) with different smoothers and using one ν-cycle multigrid preconditioner approximation (for both $\widehat{A}$ and $\widehat{S}$) at different levels of refinement, tolerance = 10^{-6}.

Levels	MG-W-cycle(3, 3)			
	DGS	IUzawa	Braess-Sarazin	PMINRES
	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))	Iter(cpu time (sec))
$\frac{1}{16}$	11 ($6.7e-01$)	8 ($5.6e-02$)	5 ($3.2e-02$)	13 ($5.8e-02$)
$\frac{1}{32}$	12 ($2.4e-01$)	9 ($3.2e-01$)	6 ($2.2e-01$)	14 ($7.3-01$)
$\frac{1}{64}$	13 (4.62)	9 (4.31)	6 (1.01)	15 (3.21)
$\frac{1}{128}$	13 (7.009)	9 (3.35)	6 (2.98)	15 (6.23)

Table 8: Changes in the $\|\nabla \cdot \mathbf{u}\|_\Omega$ estimated velocity divergence error using multigrid V-cycle. η: the global error estimator using different smoothers from one level to the other.

Levels	$\|\nabla \cdot \mathbf{u}\|_\Omega$		η	
	IUzawa	Braess-Sarazin	IUzawa	Braess-Sarazin
$\frac{1}{16}$	$3.2e-001$	$2.7e-002$	$7.97e-002$	$1.81e-001$
$\frac{1}{32}$	$1.24e-002$	$1.2e-002$	$3.08e-002$	$1.44e-001$
$\frac{1}{64}$	$5.7e-003$	$5.93e-003$	$1.15e-003$	$1.12e-002$
$\frac{1}{128}$	$3.18e-003$	$3.93e-003$	$4.8e-003$	$1.12e-002$

where $\mathbf{e}_T$ is the velocity error estimate and $R_T = \|\nabla \cdot \mathbf{u}\|_T$ and $\eta := \left(\sum_{T \in T_h} \eta_T^2\right)^{1/2}$ are the global error estimator, using different smoothers from one level to the other.

From Table 8 we note that the velocity divergence is clearly converging at a faster rate to $O(h^3)$, which means that the estimated global error η is increasingly dominated by the velocity error component as $h \to 0$.

Figure 1 shows the sample grid output at the levels 1/64 and 1/128, the sample velocity solution (exponential streamlines), and the pressure plot at the same level with the same smoother.

5. Conclusion

The purpose of this study was to explore the multigrid solver for the Stokes equations. We have introduced four smoother iterative methods for both multigrids V-cycle and W-cycle to solve the indefinite systems emanating for the mixed finite element discretization of the Stokes problem. We analyse the construction of the multigrid solver, construction of the smoothers, computation costs, and CPU time as an indicator of the performance of each smoother at all grid levels. Numerical experimental results are given for both V-cycle and W-cycle for the smoothers at different grid levels.

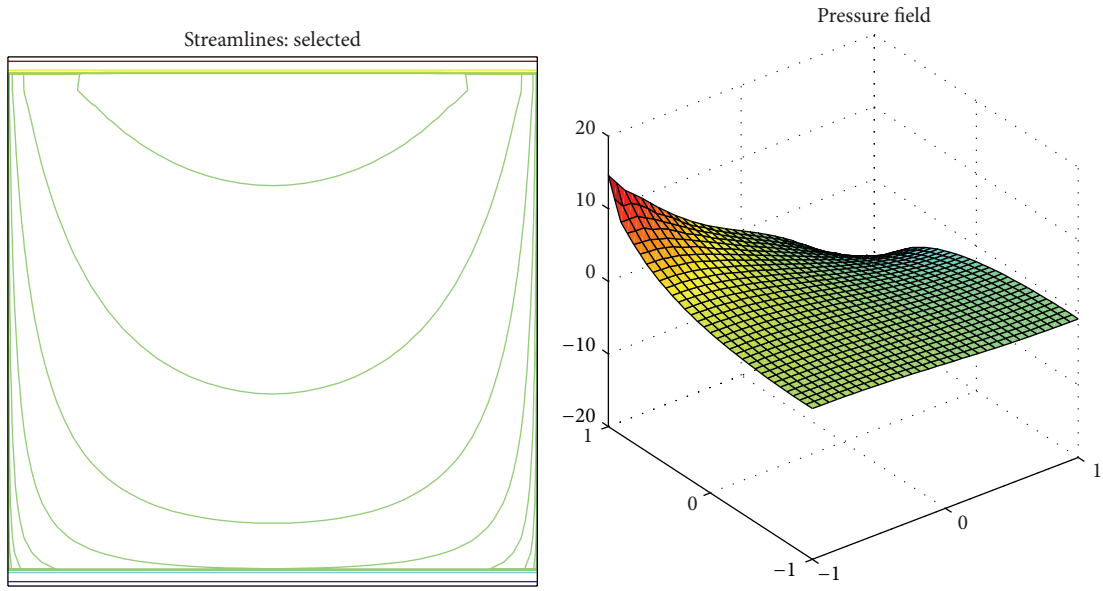


Figure 1: Velocity streamlines (a) and pressure plot (b) of the Stokes equation at level 5.

We have found out that for both cases and for all smoothers used in this study the multigrid solver is optimal and the number of iterations is bounded for all the grid levels. For the steady Stokes equations and the choices of the smoothers used the Braess-Sarazin like smoother became the best iteration to relax the error of the multigrid solver. All the numerical results show that the one v-cycle multigrid iteration is also a suitable preconditioner for the smoothers used.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] W. Hackbusch, *Multigrid Methods and Applications*, Springer, New York, NY, USA, 1985.

[2] W. Hackbusch, "Fast solution of elliptic control problems," *Journal of Optimization Theory and Applications*, vol. 31, no. 4, pp. 565–581, 1980.

[3] H. C. Elman, A. Ramage, and D. J. Silvester, "Algorithm 886: IFISS, a Matlab toolbox for modelling incompressible flow," *ACM Transactions on Mathematical Software*, vol. 33, no. 2, article 14, 18 pages, 2007.

[4] A. Brandt, *Multigrid Techniques: Giude with Applications to Fluid Dynamices*, vol. 85 of *GMD-studien*, 1984.

[5] W. Hackbusch, "On the fast solving of parabolic boundary control problems," *SIAM Journal on Control and Optimization*, vol. 17, no. 2, pp. 231–244, 1979.

[6] U. Trottenberg and C. S. Oosterleen, *Multigrid*, Academic Press, London, UK, 2001.

[7] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, New York, NY, USA, 2008.

[8] H. C. Elman, D. J. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, UK, 2005.

[9] A. Reusken, "Introduction to multigrid methods for elliptic boundary value problems," *Multiscale Simulation Methods in Molecula Sciences*, vol. 42, pp. 467–506, 2009.

[10] H. C. Elman and G. H. Golub, "Inexact and preconditioned Uzawa algorithms for saddle point problems," *SIAM Journal on Numerical Analysis*, vol. 31, no. 6, pp. 1645–1661, 1994.

[11] D. Silvester, H. C. Elman, D. Kay, and A. Wathen, "Efficient preconditioning of linearised Navier Stokes Equations," *Journal of Computational and Applied Mathematics*, 1999.

[12] M. Larin and A. Reusken, "A comparative study of efficient iterative solvers for generalized Stokes equations," *Numerical Linear Algebra with Applications*, vol. 15, no. 1, pp. 13–34, 2008.

[13] V. John, "A comparison of parallel solvers for the incompressible Navier-Stokes equations," *Computing and Visualization in Science*, vol. 1, no. 4, pp. 193–200, 1999.

[14] V. John and L. Tobska, "Numerical performance of smoothers in coupled multigrid method for parallel solution of incompressible Stokes Equations," *International Journal for Numerical Methods in Fluids*, vol. 33, pp. 453–473, 2000.

[15] M. Pernice, "A hybrid multigrid method for the steady-state incompressible Navier-Stokes equations," *Electronic Transactions on Numerical Analysis*, vol. 10, pp. 74–91, 2000.

[16] D. Braess, *Finite Elements*, Cambridge University Press, Cambridge, UK, 2007.

[17] D. Braess and R. Sarazin, "An efficient smoother for the Stokes problem," *Applied Numerical Mathematics*, vol. 23, no. 1, pp. 3–19, 1997.

[18] W. Zulehner, "A class of smoothers for saddle point problems," *Computing*, vol. 65, no. 3, pp. 227–246, 2000.

[19] P. G. Ciarlet, *The Finite Element Methods for Elliptic Problems*, North-Holland Publishing Company, 1978.

[20] M. S. Gockenbach, *Understanding and Implementing Finite Elements Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2006.

[21] A. Elakkad, A. Elkhalfi, and N. Guessous, "A mixed finite element method for Navier-Stokes equations," *Journal of Applied Mathematics & Informatics*, vol. 28, no. 5-6, pp. 1331–1345, 2010.

[22] V. Girault and P. Raviart, *A Finite Elements Approximation of the Navier-Stokes Equations*, Springer, Berlin, Germany, 1986.

[23] M. Gunzburger, *Finite Element Method for Viscous Incompressible Flows*, Academic Press, New York, NY, USA, 1989.

[24] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, 1991.

[25] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*, John Wiley and Sons, West Sussex, UK, 2003.

[26] C. H. Elman, "Multigrid and Krylov subspace methods for the discrete Stokes equations," UMIACS TR 74, 1994.

[27] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, "Analysis of the inexact Uzawa algorithm for saddle point problems," *SIAM Journal on Numerical Analysis*, vol. 34, no. 3, pp. 1072–1092, 1997.

[28] H. J. Bramble and J. E. Pasciak, "A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems," *Mathematics of Computation*, vol. 69, pp. 667–689, 1999.

[29] W. Zulehner, "Analysis of iterative methods for saddle point problems: a unified approach," *Mathematics of Computation*, vol. 71, no. 238, pp. 479–505, 2002.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 964257, 6 pages
http://dx.doi.org/10.1155/2015/964257

Research Article

On a Cubically Convergent Iterative Method for Matrix Sign

M. Sharifi,[1] S. Karimi Vanani,[1] F. Khaksar Haghani,[1] M. Arab,[1] and S. Shateyi[2]

[1]*Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran*
[2]*Department of Mathematics and Applied Mathematics, University of Venda, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 20 July 2014; Revised 29 August 2014; Accepted 30 August 2014

Academic Editor: Predrag S. Stanimirovic

We propose an iterative method for finding matrix sign function. It is shown that the scheme has global behavior with cubical rate of convergence. Examples are included to show the applicability and efficiency of the proposed scheme and its reciprocal.

1. Introduction

It is known that the function of sign in the scalar case is defined for any $z \in \mathbb{C}$ not on the imaginary axis by

$$\operatorname{sign}(z) = \begin{cases} 1, & \operatorname{Re}(z) > 0, \\ -1, & \operatorname{Re}(z) < 0. \end{cases} \tag{1}$$

An extension of (1) for the matrix case was given firstly by Roberts in [1]. This extended matrix function is of clear importance in several applications (see, e.g., [2] and the references therein).

Assume that $A \in \mathbb{C}^{n\times n}$ is a matrix with no eigenvalues on the imaginary axis. To define this matrix function formally, let

$$A = TJT^{-1} \tag{2}$$

be a Jordan canonical form arranged so that $J = \operatorname{diag}(J_1, J_2)$, where the eigenvalues of $J_1 \in \mathbb{C}^{p\times p}$ lie in the open left half-plane and those of $J_2 \in \mathbb{C}^{q\times q}$ lie in the open right half-plane; then

$$S = \operatorname{sign}(A) = T \begin{pmatrix} -I_p & 0 \\ 0 & I_q \end{pmatrix} T^{-1}, \tag{3}$$

where $p + q = n$. A simplified definition of the matrix sign function for Hermitian case (eigenvalues are all real) is

$$S = U \operatorname{diag}\left(\operatorname{sign}(\lambda_1), \ldots, \operatorname{sign}(\lambda_n)\right) U^*, \tag{4}$$

where

$$U^* A U = \operatorname{diag}(\lambda_1, \ldots, \lambda_n) \tag{5}$$

is a diagonalization of A.

The importance of computing S is also due to the fact that the sign function plays a fundamental role in iterative methods for matrix roots and the polar decomposition [3].

Note that although $\operatorname{sign}(A)$ is a square root of the identity matrix, it is not equal to I or $-I$ unless the spectrum of A lies entirely in the open right half-plane or open left half-plane, respectively. Hence, in general, $\operatorname{sign}(A)$ is a nonprimary square root of I.

In this paper, we focus on iterative methods for finding S. In fact, such methods are Newton-type schemes which are in essence fixed-point-type methods by producing a convergent sequence of matrices via applying a suitable initial matrix.

The most famous method of this class is the quadratic Newton method defined by

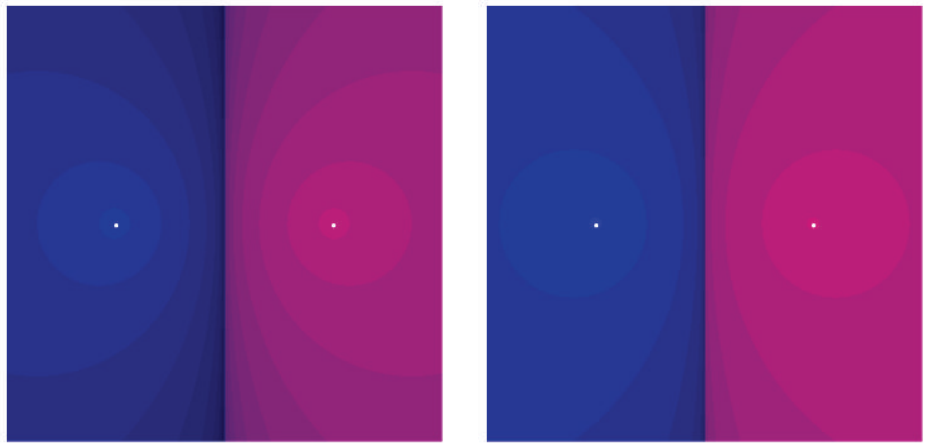
$$X_{k+1} = \frac{1}{2}\left(X_k + X_k^{-1}\right). \tag{6}$$

It should be remarked that iterative methods, such as (6), and the Newton-Schultz iteration

$$X_{k+1} = \frac{1}{2} X_k \left(3I - X_k^2\right) \tag{7}$$

or the cubically convergent Halley method

$$X_{k+1} = \left[I + 3X_k^2\right]\left[X_k\left(3I + X_k^2\right)\right]^{-1}, \tag{8}$$



(a) (b)

Figure 1: Attraction basins for (6) (a) and (8) (b) for the polynomial $g(x) = x^2 - 1$.

are all special cases of the Padé family proposed originally in [4]. The Padé approximation belongs to a broader category of rational approximations. Coincidentally, the best uniform approximation of the sign function on a pair of symmetric but disjoint intervals can be expressed as a rational function.

Note that although (7) does not possess a global convergence behavior, on state-of-the-art parallel computer architectures, matrix inversions scale less satisfactorily than matrix multiplications do, and subsequently (7) is useful in some problems. However, due to local convergence behavior, it is excluded from our numerical examples in this work.

The rest of this paper is organized as follows. In Section 2, we discuss how to construct a new iterative method for finding (3). It is also shown that the constructed method is convergent with cubical rate. It is noted that its reciprocal iteration obtained from our main method is also convergent. Numerical examples are furnished to show the higher numerical accuracy for the constructed solvers in Section 3. The paper ends in Section 4 with some concluding comments.

2. A New Method

The connection of matrix iteration methods with the sign function is not immediately obvious, but in fact such methods can be derived by applying a suitable root-finding method to the nonlinear matrix equation

$$X^2 = I \tag{9}$$

and when of course sign(A) is one solution of this equation (see for more [5]).

Here, we consider the following root-solver:

$$x_{k+1} = x_k - \frac{10 - 4L(x_k)}{10 - 9L(x_k)} \frac{f(x_k)}{f'(x_k)}, \tag{10}$$

with $L(x_k) = f''(x_k) f(x_k)/f'(x_k)^2$. In what follows, we observe that (10) possesses third order of convergence.

Theorem 1. *Let $\alpha \in D$ be a simple zero of a sufficiently differentiable function $f : D \subseteq \mathbb{C} \to \mathbb{C}$, which contains x_0 as an initial approximation. Then the iterative expression (10) satisfies*

$$e_{k+1} = \left(\frac{c_2^2}{5} - c_3 \right) e_k^3 + O\left(e_k^4\right), \tag{11}$$

where $c_j = f^{(j)}(\alpha)/j! f'(\alpha)$, $e_k = x_k - \alpha$.

Proof. The proof would be similar to the proofs given in [6]. □

Applying (10) on the matrix equation (9) will result in the following new matrix fixed-point-type iteration for finding (3):

$$X_{k+1} = \left(2I + 15X_k^2 + 3X_k^4\right)\left[9X_k + 11X_k^3\right]^{-1}, \tag{12}$$

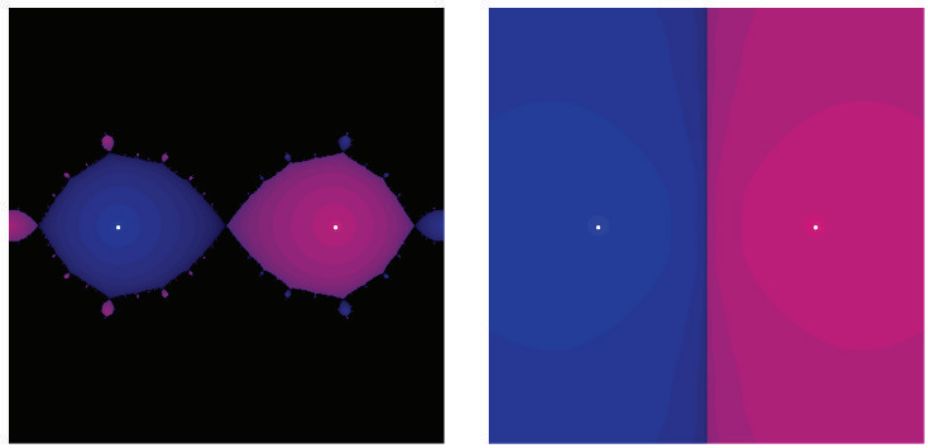
where $X_0 = A$. This is named PM1 from now on.

The proposed scheme (12) is not a member of Padé family [4]. Furthermore, applying (10) on the scalar equation $g(x) = x^2 - 1$ provides a global convergence in the complex plane (except the points lying on the imaginary axis). This global behavior, which is kept for matrix case, has been illustrated in Figure 1 by drawing the basins of attraction for (6) and (8). The attraction basins for (7) (local convergence) and (12) (global convergence) are also portrayed in Figure 2.

Theorem 2. *Let $A \in \mathbb{C}^{n\times n}$ have no pure imaginary eigenvalues. Then, the matrix sequence $\{X_k\}_{k=0}^{k=\infty}$ defined by (12) converges to S, choosing $X_0 = A$.*

Proof. We remark that all matrices, whether they are diagonalizable or not, have a Jordan normal form $A = TJT^{-1}$, where the matrix J consists of Jordan blocks. For this reason, let A have a Jordan canonical form arranged as

$$T^{-1}AT = \Lambda = \begin{bmatrix} C & 0 \\ 0 & N \end{bmatrix}, \tag{13}$$



(a) (b)

Figure 2: Attraction basins of (7) (a) and (12) (b) for the polynomial $g(x) = x^2 - 1$.

where T is a nonsingular matrix and C, N are square Jordan blocks corresponding to eigenvalues lying in $\mathbb{C}^-$ and $\mathbb{C}^+$, respectively. We have

$$\begin{aligned}&\operatorname{sign}(\Lambda)\\&= \operatorname{sign}\left(T^{-1}AT\right) = T^{-1}\operatorname{sign}(A)\,T\\&= \operatorname{diag}\left(\operatorname{sign}(\lambda_1),\ldots,\operatorname{sign}\left(\lambda_p\right),\operatorname{sign}\left(\lambda_{p+1}\right),\ldots,\operatorname{sign}(\lambda_n)\right).\end{aligned} \tag{14}$$

If we define $D_k = T^{-1}X_kT$, then, from the method (12), we obtain

$$D_{k+1} = \left(2I + 15D_k^2 + 3D_k^4\right)\left[9D_k + 11D_k^3\right]^{-1}. \tag{15}$$

Note that if D_0 is a diagonal matrix, then, based on an inductive proof, all successive D_k are diagonal too. From (15), it is enough to show that $\{D_k\}$ converges to $\operatorname{sign}(\Lambda)$. We remark that the case at which D_0 is not diagonal will be discussed later in the proof.

In the meantime, we can write (15) as n uncoupled scalar iterations to solve $g(x) = x^2 - 1 = 0$, given by

$$d_{k+1}^i = \left(2 + 15{d_k^i}^2 + 3{d_k^i}^4\right)\left[9d_k^i + 11{d_k^i}^3\right]^{-1}, \tag{16}$$

where $d_k^i = (D_k)_{i,i}$ and $1 \le i \le n$. From (15) and (16), it is enough to study the convergence of $\{d_k^i\}$ to $\operatorname{sign}(\lambda_i)$.

It is known that $\operatorname{sign}(\lambda_i) = s_i = \pm 1$. Thus, we attain

$$\frac{d_{k+1}^i - 1}{d_{k+1}^i + 1} = \frac{\left(-1 + d_k^i\right)^3\left(-2 + 3d_k^i\right)}{\left(1 + d_k^i\right)^3\left(2 + 3d_k^i\right)}. \tag{17}$$

Since $|d_0^i| = |\lambda_i| > 0$, we have

$$\lim_{k\to\infty}\left|\frac{d_{k+1}^i - 1}{d_{k+1}^i + 1}\right| = 0, \tag{18}$$

and $\lim_{k\to\infty}|d_k^i| = 1 = |\operatorname{sign}(\lambda_i)|$. This shows that $\{d_k^i\}$ is convergent.

In the convergence proof, D_0 may not be diagonal. Since the Jordan canonical form of some matrices may not be diagonal, thus, one cannot write (15) as n uncoupled scalar iterations (16). We comment that in this case our method is also convergent. To this goal, we must pursue the scalar relationship among the eigenvalues of the iterates for the studied rational matrix iteration.

In this case, the eigenvalues of X_k are mapped from the iterate k to the iterate $k + 1$ by the following relation:

$$\lambda_{k+1}^i = \left(2 + 15{\lambda_k^i}^2 + 3{\lambda_k^i}^4\right)\left[9\lambda_k^i + 11{\lambda_k^i}^3\right]^{-1}. \tag{19}$$

So, (19) clearly shows that the eigenvalues in the general case are convergent to ± 1; that is to say,

$$\lim_{k\to\infty}\left|\frac{\lambda_{k+1}^i - 1}{\lambda_{k+1}^i + 1}\right| = 0. \tag{20}$$

Consequently, we have

$$\lim_{k\to\infty} X_k = T\left(\lim_{k\to\infty} D_k\right)T^{-1} = T\operatorname{sign}(\Lambda)\;T^{-1} = \operatorname{sign}(A). \tag{21}$$

The proof is ended. □

Theorem 3. *Let $A \in \mathbb{C}^{n\times n}$ have no pure imaginary eigenvalues. Then the proposed method (12) converges cubically to the sign matrix S.*

Proof. Clearly, X_k are rational functions of A and, hence, like A, commute with S. On the other hand, we know that $S^2 = I$,

Table 1: Results of comparisons for Example 5 using $X_0 = A$.

Methods	NM	HM	PM1	PM2
IT	14	9	8	8
R_{k+1}	1.41584×10^{-249}	1.0266×10^{-299}	2.5679×10^{-298}	1.45091×10^{-337}
ρ	1.99077	3	3	3

Table 2: Results of comparisons for Example 6 using $X_0 = A$.

Methods	NM	HM	PM1	PM2
IT	10	7	6	6
R_{k+1}	5.7266×10^{-155}	5.80819×10^{-203}	8.38265×10^{-153}	1.55387×10^{-143}
ρ	2.00228	3.00001	3.00015	3

$S^{-1} = S$, $S^{2j} = I$, and $S^{2j+1} = S$, $j \geq 1$. Using the replacement $B_k = 9X_k + 11X_k^3$, we have

$$\begin{aligned} X_{k+1} - S &= \left(2I + 15X_k^2 + 3X_k^4\right) B_k^{-1} - S \\ &= \left(2I + 15X_k^2 + 3X_k^4 - SB_k\right) B_k^{-1} \\ &= \left(2I + 15X_k^2 + 3X_k^4 - 9SX_k - 11SX_k^3\right) B_k^{-1} \\ &= -\left(-2S - 15SX_k^2 - 3SX_k^4 + 9X_k + 11X_k^3\right) \\ &\quad \times S^{-1} B_k^{-1} \\ &= \left(X_k - S\right)^3 \left(2I - 3SX_k\right) S^{-1} B_k^{-1}. \end{aligned} \tag{22}$$

Now, using any matrix norm from both sides of (22), we attain

$$\left\|X_{k+1} - S\right\| \leq \left(\left\|B_k^{-1}\right\| \left\|S^{-1}\right\| \left\|2I - 3SX_k\right\|\right) \left\|X_k - S\right\|^3. \tag{23}$$

This reveals the cubical rate of convergence for the new method (12). The proof is complete. □

It should be remarked that the reciprocal iteration obtained from (12) is also convergent to the sign matrix (3) as follows:

$$X_{k+1} = \left(9X_k + 11X_k^3\right) \left[2I + 15X_k^2 + 3X_k^4\right]^{-1}, \tag{24}$$

where $X_0 = A$. This is named PM2. Similar convergence results as the ones given in Theorems 2-3 hold for (24).

A scaling approach to accelerate the beginning phase of convergence is normally necessary since the convergence rate cannot be seen in the initial iterates. Such an idea was discussed fully in [7] for Newton's method. An effective way to enhance the initial speed of convergence is to scale the iterates prior to each iteration; that is, X_k is replaced by $\mu_k X_k$. Subsequently, we can present the accelerated forms of our proposed methods as follows:

$$\begin{gathered} X_0 = A, \\ \mu_k = \text{is the scaling parameter computed by (27)}, \\ X_{k+1} = \left(2I + 15\mu_k^2 X_k^2 + 3\mu_k^4 X_k^4\right) \left[9\mu_k X_k + 11\mu_k^3 X_k^3\right]^{-1}, \end{gathered} \tag{25}$$

or

$$\begin{gathered} X_0 = A, \\ \mu_k = \text{is the scaling parameter computed by (27)}, \\ X_{k+1} = \left(9\mu_k X_k + 11\mu_k^3 X_k^3\right) \left[2I + 15\mu_k^2 X_k^2 + 3\mu_k^4 X_k^4\right]^{-1}, \end{gathered} \tag{26}$$

$$\mu_k = \begin{cases} \sqrt{\dfrac{\left\|X_k^{-1}\right\|}{\left\|X_k\right\|}}, & \text{(norm scaling)}, \\ \sqrt{\dfrac{\rho\left(X_k^{-1}\right)}{\rho\left(X_k\right)}}, & \text{(spectral scaling)}, \\ \sqrt{\left|\det\left(X_k\right)\right|^{-1/n}}, & \text{(determinantal scaling)}, \end{cases} \tag{27}$$

where $\lim_{k\to\infty} \mu_k = 1$ and $\lim_{k\to\infty} X_k = S$. The different scaling factors for μ_k in (27) are borrowed from Newton's method. For this reason it is important to show the behavior of the accelerator methods (25)-(26) and this will be done in the next section.

3. Numerical Examples

In this section, the results of comparisons in terms of number of iterations and the residual norms have been reported for various matrix iterations. We compare PM1 and PM2 with (6) denoted by NM and (8) denoted by HM. The programming package Mathematica [8] is used throughout this section. In Tables 1 and 2, IT stands for the number of iterates.

Note that the computational order of convergence for matrix iterations in finding S can be estimated by [9]

$$\rho = \frac{\log\left(\left\|X_{k+1}^2 - I\right\| / \left\|X_k^2 - I\right\|\right)}{\log\left(\left\|X_k^2 - I\right\| / \left\|X_{k-1}^2 - I\right\|\right)}, \tag{28}$$

where X_{k-1}, X_k, and X_{k+1} are the last three approximations.

Example 4. In this example, we compare the methods for the following 500×500 complex matrix:

```
n = 500; SeedRandom[123];
A = RandomComplex[{-100 - I, 100 + I},{n,n}];
```

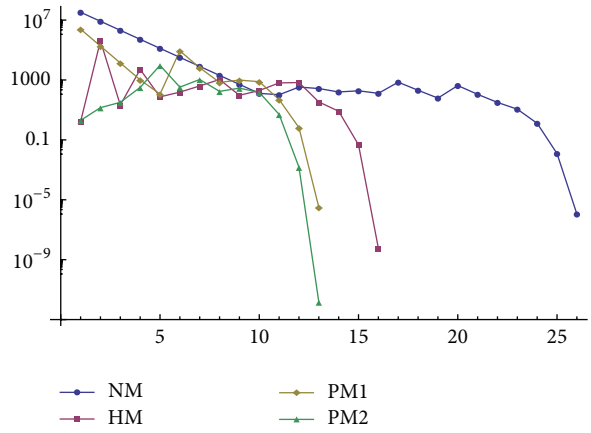


Figure 3: Convergence history versus number of iterations for different methods in Example 4.

We apply here double precision arithmetic with the stop termination $R_{k+1} = \|X_{k+1}^2 - I\|_\infty \leq 10^{-5}$. Results are given in Figure 3.

Example 5 (academic test). We compute the matrix sign for the following complex test problem:

$$A = \begin{pmatrix} 0 & 10 & i & 7+i \\ 7 & -5 & 6 & -5 \\ 0 & 60 & -2 & 9 \\ 0 & 5 & 9 & i \end{pmatrix}, \tag{29}$$

where

$$S = \begin{pmatrix} 0.882671 + 0.0118589i & 0.461061 - 0.0519363i & -0.167387 + 0.0215728i & 0.168184 - 0.0194164i \\ 0.219355 + 0.00464485i & 0.136809 - 0.00840032i & 0.313995 - 0.00196855i & -0.314977 - 0.00219388i \\ -0.566306 - 0.0184534i & 2.22878 + 0.0471091i & 0.189109 - 0.00416224i & 0.813305 + 0.0149399i \\ 0.145285 + 0.00157401i & -0.57165 + 0.000347003i & 0.207909 - 0.00345322i & 0.791412 + 0.000703638i \end{pmatrix}. \tag{30}$$

We apply here 600-digit fixed point arithmetic in our calculations with the stop termination $R_{k+1} = \|X_{k+1}^2 - I\|_\infty \leq 10^{-150}$. The results for this example are illustrated in Table 1. We report the COCs in l_∞.

Iterative schemes PM1 and PM2 are evidently believed to be more favorable than the other compared methods due to their fewer number of iterations and acceptable accuracy. Hence, the proposed methods with properly chosen initial matrix X_0 can be helpful in finding the sign of a nonsingular complex matrix.

Example 6. Here we rerun Example 5 using the scaling approaches (27) with the stop termination $R_{k+1} = \|X_{k+1}^2 - I\|_\infty \leq 10^{-100}$. The results for this example are illustrated in Table 2. We used the determinantal scaling for all compared methods. The numerical results uphold the theoretical discussions of Section 2.

A price paid for the high order convergence is the increased amount of matrix multiplications and inversions. This is a typical consequence. However the most important advantage of the presented methods in contrast to the methods of the same orders, such as (8), is their larger attraction basins. This superiority basically allows the new methods to converge to a required tolerance in one lower iteration than their same order methods. Hence, studying the thorough computational efficiency index of the proposed methods may not be an easy task and it must be pursued experimentally. In an experimental manner, if the costs of one matrix-matrix product and one matrix inversion are unity and 1.5 of unity, respectively, then we have the following efficiency indices for different methods: $E_{(6)} = 2^{1/(14(1)+14(1.5))} \simeq 1.020$, $E_{(8)} = 3^{1/(9(3)+9(1.5))} \simeq 1.027$, and $E_{(12)} = 3^{1/(8(4)+8(1.5))} \simeq 1.025$. Note that for Newton's method we have one matrix-matrix product per cycle due to the computation of stopping criterion. Other similar computations for efficiency indices for different examples show similar behaviors to the above mentioned one.

4. Summary

Matrix functions are used in many areas of linear algebra and arise in numerous applications in science and engineering. The function of a matrix can be defined in several ways, of which the following three are generally the most useful: Jordan canonical form, polynomial interpolation, and finally Cauchy integral.

In this paper, we have focus on iterative methods for this purpose. Hence, a third order nonlinear equation solver has been employed for constructing a new method for S. It was shown that the convergence is global via attraction basins in the complex plane and the rate of convergence is cubic. Furthermore, PM2 as the reciprocal of the method PM1 with the same convergence properties was proposed. The acceleration of PM1 and PM2 via scaling was also illustrated simply.

Finally some numerical examples in both double and multiple precisions were performed to show the efficiency of PM1 and PM2. Further researches must be forced to extend the obtained iterations for computing polar decompositions in future studies.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the referees for their helpful corrections and suggestions.

References

[1] J. D. Roberts, "Linear model reduction and solution of the algebraic Riccati equation by use of the sign function," *International Journal of Control*, vol. 32, no. 4, pp. 677–687, 1980.

[2] C. S. Kenney and A. J. Laub, "The matrix sign function," *IEEE Transactions on Automatic Control*, vol. 40, no. 8, pp. 1330–1348, 1995.

[3] N. J. Higham, *Functions of Matrices: Theory and Computation*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2008.

[4] C. Kenney and A. J. Laub, "Rational iterative methods for the matrix sign function," *SIAM Journal on Matrix Analysis and Applications*, vol. 12, no. 2, pp. 273–291, 1991.

[5] F. Soleymani, P. S. Stanimirović, S. Shateyi, and F. K. Haghani, "Approximating the matrix sign function using a novel iterative method," *Abstract and Applied Analysis*, vol. 2014, Article ID 105301, 9 pages, 2014.

[6] F. Soleymani, "Some high-order iterative methods for finding all the real zeros," *Thai Journal of Mathematics*, vol. 12, no. 2, pp. 313–327, 2014.

[7] C. Kenney and A. J. Laub, "On scaling Newton's method for polar decomposition and the matrix sign function," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 3, pp. 698–706, 1992.

[8] S. Wagon, *Mathematica in Action*, Springer, New York, NY, USA, 3rd edition, 2010.

[9] F. Soleymani, E. Tohidi, S. Shateyi, and F. Haghani, "Some matrix iterations for computing matrix sign function," *Journal of Applied Mathematics*, vol. 2014, Article ID 425654, 9 pages, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 614612, 11 pages
http://dx.doi.org/10.1155/2015/614612

Research Article

Geometric Construction of Eighth-Order Optimal Families of Ostrowski's Method

Ramandeep Behl and S. S. Motsa

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to S. S. Motsa; sandilemotsa@gmail.com

Received 15 July 2014; Accepted 28 August 2014

Academic Editor: Juan R. Torregrosa

Based on well-known fourth-order Ostrowski's method, we proposed many new interesting optimal families of eighth-order multipoint methods without memory for obtaining simple roots. Its geometric construction consists in approximating f_n' at z_n in such a way that its average with the known tangent slopes f_n' at x_n and y_n is the same as the known weighted average of secant slopes and then we apply weight function approach. The adaptation of this strategy increases the convergence order of Ostrowski's method from four to eight and its efficiency index from 1.587 to 1.682. Finally, a number of numerical examples are also proposed to illustrate their accuracy by comparing them with the new existing optimal eighth-order methods available in the literature. It is found that they are very useful in high precision computations. Further, it is also noted that larger basins of attraction belong to our methods although the other methods are slow and have darker basins while some of the methods are too sensitive upon the choice of the initial value.

1. Introduction

Multipoint iterative methods for solving nonlinear equation,

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

have drawn a considerable attention in the first decade of the 21st century, which led to the construction of many methods of this type. These methods are primarily introduced with the aim to achieve as high as possible order of convergence using a fixed number of function evaluations. However, multipoint methods do not use higher order derivatives and have great practical importance since they overcome the theoretical limitations of one-point methods regarding their convergence order and computational efficiency.

As the order of an iterative method increases, so does the number of functional evaluations per step. The efficiency index [1] gives a measure of the balance between those quantities, according to the formula $p^{1/d}$, where p is the order of convergence of the method and d is the number of functional evaluations per step. According to the Kung-Traub conjecture [2], the order of convergence of any multipoint method cannot exceed the bound 2^{n-1}, called the optimal order. Thus, the optimal order for a method with three functional evaluations per step would be four. The well-known King's family of methods [3] is an example of fourth order multipoint methods requiring three functional evaluations per full iteration, which is given by

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= x_n - \frac{\{f(x_n)\}^2 + (\beta - 1) f(x_n) f(y_n) + \beta\{f(y_n)\}^2}{f'(x_n)\left[f(x_n) + (\beta - 2) f(y_n)\right]}, \\ &\text{where } \beta \in \mathbb{R}. \end{aligned} \tag{2}$$

For $\beta = 0$, one can easily get the well-known Ostrowski's method. From practical point of view, King's family [3] and Ostrowski's method [1, 4] are one of the most efficient multipoint fourth-order methods known to date because they have simple body structures and do not require the computation of

a second-order derivative. They have efficiency index equal to 1.5874, which is very competitive.

In recent years, based on the King's method and Ostrowski's method, some higher order iterative methods have been proposed and analyzed for solving nonlinear equations. J. R. Sharma and R. Sharma [5] proposed a family of Ostrowski's method with eighth-order convergence, which is given by

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(x_n) f(y_n)}{f'(x_n)(f(x_n) - 2f(y_n))}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} H(\mu_n), \end{aligned} \tag{3}$$

where $\mu = f(y_n)/f(x_n)$ and $H(t)$ represents a real-valued function with $H(0) = 1$, $H'(0) = 2$, and $|H''(0)| < \infty$. We will refer to this method as SSM_8.

Liu and Wang [6] have also presented another eighth-order family of Ostrowski's method, requiring three-function and one-derivative evaluation per iteration:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(x_n) f(y_n)}{f'(x_n)(f(x_n) - 2f(y_n))}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \left[\left(\frac{(f(x_n) - f(y_n))}{(f(x_n) - 2f(y_n))} \right)^2 \right. \\ &\qquad + \frac{f(z_n)}{(f(y_n) - af(z_n))} \\ &\qquad \left. + \frac{4f(z_n)}{f(x_n) + bf(z_n)} \right], \end{aligned} \tag{4}$$

where a and b are two free disposable parameters. We will refer to this method as LWM_8.

Soleymani et al. [7] also proposed eighth-order variant of Ostrowski's method, which is given by

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(x_n) f(y_n)}{f'(x_n)(f(x_n) - 2f(y_n))}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \frac{f(x_n)}{(f(x_n) - 2f(y_n))} \\ &\quad \times \left[1 + 2\frac{f(z_n)}{f(x_n)} + \frac{f(z_n)}{f(y_n)} + \left(\frac{f(y_n)}{f(x_n)} \right)^2 \right. \\ &\qquad \left. + 2\left(\frac{f(y_n)}{f(x_n)} \right)^3 - \frac{f(z_n)}{f'(x_n)} + \left(\frac{f(y_n)}{f'(x_n)} \right)^3 \right]. \end{aligned} \tag{5}$$

We will refer to this method as SM_8.

The main goal of this paper is to develop a general class of very efficient three-point methods for solving nonlinear equations. Here, we derived several new optimal families of eighth-order Ostrowski's method by taking the arithmetic mean of three slopes and then applying weight function approach. In terms of computational cost, they require four functional evaluations per iteration. Thus, the new family adds only one evaluation of the function at another point other than Ostrowski's method and order increases from four to eight. This property of the new methods provides a new example of multipoint methods without memory having optimal order of convergence. The efficiency of the methods is tested on a number of numerical examples.

2. Development of Optimal Eighth-Order Families of Ostrowski's Method

Newton's method is probably the best known and most widely used one-point iterative method for solving nonlinear equation (1). It converges quadratically to a simple root and linearly to a multiple root. Its geometric construction consists in considering the straight line

$$y = ax + b, \tag{6}$$

then determining the unknowns a and b by imposing the tangency conditions:

$$y(x_n) = f(x_n), \qquad y'(x_n) = f'(x_n), \tag{7}$$

and thereby obtaining the tangent line

$$y(x) - f(x_n) = f'(x_n)(x - x_n), \tag{8}$$

to the graph of $f(x)$ at $(x_n, f(x_n))$.

The point of intersection of this tangent line with x-axis gives the celebrated Newton's method

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \geq 0. \tag{9}$$

The convergence order and computational efficiency of the one-point iterative methods are lower than multipoint iterative methods [8] because multipoint iterative methods can overcome theoretical limits of one-point methods concerning the convergence order and computational efficiency. In recent years, many multipoint iterative methods have been proposed that improve the local convergence order of the classical Newton's method. In 1973, King [3] had considered the following fourth-order iteration scheme:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ x_{n+1} &= y_n - \frac{f(y_n)}{f'(y_n)}, \quad n = 0, 1, 2, \ldots. \end{aligned} \tag{10}$$

But according to the Kung-Traub conjecture [2], the above scheme (10) is not an optimal method because it has fourth-order convergence and requires four functional evaluations

per full iteration. However, King [3] had reduced the number of function evaluations by using some suitable approximation of $f'(y_n)$. In fact, King had taken the approximation of $f'(y_n)$ in such a way that its average with the known tangent slopes f'_n at x_n and y_n is the same as the known secant slopes; that is,

$$\frac{f'(y_n)+f'(x_n)}{2}=\frac{f(x_n)-f(y_n)}{x_n-y_n}. \tag{11}$$

After solving (11), one can get the following value of $f'(y_n)$ as

$$f'(y_n)=\frac{f'(x_n)(f(x_n)-2f(y_n))}{f(x_n)}. \tag{12}$$

Using this value in scheme (10), we get

$$\begin{aligned} y_n &= x_n-\frac{f(x_n)}{f'(x_n)},\\ x_{n+1} &= y_n-\frac{f(x_n)f(y_n)}{f'(x_n)(f(x_n)-2f(y_n))}, \quad n=0,1,2,3,\ldots. \end{aligned} \tag{13}$$

This is well-known Ostrowski's method [1, 4]. It is very interesting to note that, by adding one evaluation of the function at another point iterated by Newton's method, the order of convergence increases from two to four and is free from the second-order derivative.

Now, we intend to derive the new optimal eighth-order family of Ostrowski's method. For this, we consider a three-step iteration scheme with existing Ostrowski's method as follows:

$$\begin{aligned} y_n &= x_n-\frac{f(x_n)}{f'(x_n)},\\ z_n &= y_n-\frac{f(x_n)f(y_n)}{f(x_n)-2f(y_n)},\\ x_{n+1} &= z_n-\frac{f(z_n)}{f'(z_n)}. \end{aligned} \tag{14}$$

Again the above method is not optimal according to the Kung-Traub conjecture [2], because it has eighth-order convergence and requires five functional evaluations per full iteration. However, we can reduce the number of function evaluations by using some suitable approximation of $f'(z_n)$. In fact, we will take the approximation of $f'(z_n)$ similar to King's approximation in such a way that its average with the known slopes f'_n at x_n, y_n, and z_n is the same as the known weighted average of secant slopes:

$$\begin{aligned} &\frac{f'(y_n)+f'(x_n)+f'(z_n)}{3}\\ &\quad=\frac{1}{3}\left[2\left(\frac{f(x_n)-f(y_n)}{x_n-y_n}\right)+\frac{f(z_n)-f(y_n)}{z_n-y_n}\right]. \end{aligned} \tag{15}$$

After solving (15), we get

$$f'(z_n)=\frac{f'(x_n)(f(x_n)-2f(y_n))(f(y_n)-f(z_n))}{f(x_n)f(y_n)}. \tag{16}$$

Using this value of $f'(z_n)$ in scheme (14), we get

$$\begin{aligned} y_n &= x_n-\frac{f(x_n)}{f'(x_n)},\\ z_n &= y_n-\frac{f(x_n)f(y_n)}{f'(x_n)(f(x_n)-2f(y_n))},\\ x_{n+1} &= z_n-\frac{f(x_n)f(y_n)f(z_n)}{f'(x_n)(f(x_n)-2f(y_n))(f(y_n)-f(z_n))}. \end{aligned} \tag{17}$$

This is a new sixth-order Ostrowski's method. It satisfies the following error equation:

$$e_{n+1}=\left(c_2^5-c_2^3c_3\right)e_n^6+O\left(e_n^7\right), \tag{18}$$

where $e_n=x_n-r$ and $c_k=(1/k!)(f^{(k)}(r)/f'(r))$, $k=2,3,\ldots$.

Again, the above method is not optimal according to the Kung-Traub conjecture [2]. Therefore, to improve the order of convergence of this method, we will now make use of weight function approach to build our optimal families of this iterative method by a simple change in its third step. Therefore, we consider

$$\begin{aligned} y_n &= x_n-\frac{f(x_n)}{f'(x_n)},\\ z_n &= y_n-\frac{f(x_n)f(y_n)}{f'(x_n)(f(x_n)-2f(y_n))},\\ x_{n+1} &= z_n-\frac{f(x_n)f(y_n)f(z_n)}{f'(x_n)(f(x_n)-2f(y_n))(f(y_n)-f(z_n))}\\ &\quad\times Q(u,v), \end{aligned} \tag{19}$$

where $u=f(z_n)/f(x_n)$, $v=f(y_n)/f(x_n)$, and $Q(u,v)$ is a two variable real-valued weight function such that its order of convergence reaches at the optimal level eight without using any more functional evaluations. Theorem 1 indicates that under what conditions on the weight function in (19) the order of convergence will reach at the optimal level eight.

3. Order of Convergence

Theorem 1. *Let a sufficiently smooth function $f: D\subseteq\mathbb{R}\to\mathbb{R}$ have a simple zero r in the open interval D. Let $Q(u,v)$ be a two-variable real-valued differentiable function. If an initial approximation x_0 is sufficiently close to the required root r of a function f, then the convergence order of the family of*

three-point methods (19) *is equal to eight when it satisfies the following conditions:*

$$Q_{00} = 1, \quad Q_{10} = 2, \quad Q_{01} = 0, \quad Q_{02} = 2, \quad Q_{03} = 12, \tag{20}$$

where $Q_{ij} = (1/i!\,j!)(\partial Q(u,v)/\partial u^i v^j)|_{(0,0)}$, *for* $i = 0, 1, 2, 3$ *and* $j = 0, 1, 2, 3$.

It satisfies the following error equation:

$$e_{n+1} = -c_2^2\left(c_2^2 - c_3\right) \times \left(\left(-7 + Q_{11}\right)c_2^3 - \left(-4 + Q_{11}\right)c_2c_3 - c_4\right)e^8 + O[e]^9, \tag{21}$$

where e_n *and* c_k *are already defined in* (18).

Proof. Let $x = r$ be a simple zero of $f(x)$. Expanding $f(x_n)$ and $f'(x_n)$ about $x = r$ by the Taylor's series expansion, we have

$$\begin{aligned} f(x_n) &= f'(r) \times \left(e_n + c_2e_n^2 + c_3e_n^3 + c_4e_n^4 + c_5e_n^5 + c_6e_n^6 + c_7e_n^7 + c_8e_n^8\right) + O\left(e_n^9\right), \\ f'(x_n) &= f'(r) \times \left(1 + 2c_2e_n + 3c_3e_n^2 + 4c_4e_n^3 + 5c_5e_n^4 + 6c_6e_n^5 + 7c_7e_n^6 + 8c_8e_n^7\right) + O\left(e_n^9\right), \end{aligned} \tag{22}$$

respectively.

From (22), we have

$$\frac{f(x_n)}{f'(x_n)} = e_n - c_2e_n^2 + 2\left(c_2^2 - c_3\right)e_n^3 + \left(-4c_2^3 + 7c_2c_3 - 3c_4\right)e_n^4 + O\left(e_n^5\right), \tag{23}$$

and in combination with the Taylor series expansion of $f(x_n - (f(x_n)/f'(x_n)))$ about $x = r$, we have

$$f(y_n) = f\left(x_n - \frac{f(x_n)}{f'(x_n)}\right) = f'(r) \times \left[c_2e_n^2 + \left(-2c_2^2 + 2c_3\right)e_n^3 + \left(5c_2^3 - 7c_2c_3 + 3c_4\right)e_n^4 - 2\left(6c_2^4 - 12c_2^2c_3 + 3c_3^2 + 5c_2c_4 - 2c_5\right)e_n^5\right] + O\left(e_n^6\right). \tag{24}$$

Therefore, we have

$$\begin{aligned} \frac{f(x_n)}{f(x_n) - 2f(y_n)} &= 1 + 2c_2e_n + \left(-2c_2^2 + 4c_3\right)e_n^2 + \left(-4c_2c_3 + 6c_4\right)e_n^3 + \left(4c_2^4 - 6c_2^2c_3 - 4c_2c_4 + 8c_5\right)e_n^4 \\ &\quad - 2\left(4c_2^5 - 14c_2^3c_3 + 5c_2^2c_4 - 2c_3c_4 + c_2\left(9c_3^2 + 2c_5\right) - 5c_6\right)e_n^5 + O\left(e_n^6\right), \\ u_n = \frac{f(y_n)}{f'(x_n)} &= c_2e_n^2 + \left(-4c_2^2 + 2c_3\right)e_n^3 + \left(13c_2^3 - 14c_2c_3 + 3c_4\right)e_n^4 \\ &\quad - 2\left(19c_2^4 - 32c_2^2c_3 + 6c_3^2 + 10c_2c_4 - 2c_5\right)e_n^5 + O\left(e_n^6\right). \end{aligned} \tag{25}$$

From (25), we have

$$\begin{aligned} z_n &= y_n - \frac{f(x_n)}{f'(x_n)}\frac{f(y_n)}{f(x_n) - 2f(y_n)}, \\ &= \left(c_2^3 - c_2c_3\right)e_n^4 - 2\left(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4\right)e_n^5 + \left(10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2\left(6c_3^2 - c_5\right)\right) \times e_n^6 + O\left(e_n^7\right). \end{aligned} \tag{26}$$

Now, expanding $f(z_n)$ about r, we get

$$f(z_n) = f'(r) \times \left[\left(c_2^3 - c_2c_3\right)e_n^4 - 2\left(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4\right)e_n^5 + \left(10c_2^5 - 30c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2\left(6c_3^2 - c_5\right)\right)e_n^6\right] + O\left(e_n^7\right). \tag{27}$$

Furthermore, we have

$$\begin{aligned} \frac{f(x_n)f(y_n)f(z_n)}{f'(x_n)\left(f(y_n) - f(z_n)\right)\left(f(x_n) - 2f(y_n)\right)} &= \left(c_2^3 - c_2c_3\right)e_n^4 - 2\left(2c_2^4 - 4c_2^2c_3 + c_3^2 + c_2c_4\right)e_n^5 \\ &\quad + \left(9c_2^5 - 29c_2^3c_3 + 12c_2^2c_4 - 7c_3c_4 + 3c_2\left(6c_3^2 - c_5\right)\right)e_n^6 + O\left(e_n^7\right), \end{aligned} \tag{28}$$

$$
\begin{aligned}
u_n &= \frac{f(z_n)}{f(x_n)} \\
&= \left(c_2^3 - c_2c_3\right)e_n^3 + \left(-5c_2^4 + 9c_2^2c_3 - 2c_3^2 - 2c_2c_4\right)e_n^4 \\
&\quad + \left(15c_2^5 - 40c_2^3c_3 + 14c_2^2c_4 - 7c_3c_4 + 3c_2\left(7c_3^2 - c_5\right)\right) \\
&\quad \times e_n^5 + O\left(e_n^6\right), \\
v_n &= \frac{f(y_n)}{f(x_n)} \\
&= c_2e_n + \left(-3c_2^2 + 2c_3\right)e_n^2 + \left(8c_2^3 - 10c_2c_3 + 3c_4\right)e_n^3 \\
&\quad + \left(-20c_2^4 + 37c_2^2c_3 - 8c_3^2 - 14c_2c_4 + 4c_5\right)e_n^4 \\
&\quad + O\left(e_n^5\right).
\end{aligned} \tag{29}
$$

Since it is clear from (29) that u_n and v_n are of order e_n^3 and e_n respectively, therefore, we can expand weight function $Q(u, v)$ in the neighborhood of origin by Taylor series expansion up to third-order terms as follows:

$$
\begin{aligned}
Q(u,v) &= Q_{00} + Q_{10}u + Q_{01}v \\
&\quad + \frac{1}{2}\left(Q_{20}u^2 + 2Q_{11}uv + Q_{02}v^2\right) \\
&\quad + \frac{1}{6}\left(Q_{30}u^3 + 3Q_{21}u^2v + 3Q_{12}uv^2 + Q_{03}v^3\right).
\end{aligned} \tag{30}
$$

Using (28), (29), and (30) in scheme (19), we have the following error equation size:

$$
\begin{aligned}
&e_{n+1} \\
&= z_n - \frac{f(x_n)\,f(y_n)\,f(z_n)}{f'(x_n)\left(f(x_n) - 2f(y_n)\right)\left(f(y_n) - f(z_n)\right)} \\
&\times Q(u,v) = -\left(-1 + Q_{00}\right)c_2\left(c_2^2 - c_3\right)e_n^4 \\
&+ \left(\left(-4 + 4Q_{00} - Q_{01}\right)c_2^4\right. \\
&\quad + \left(8 - 8Q_{00} + Q_{01}\right)c_2^2c_3 \\
&\quad \left. + 2\left(-1 + Q_{00}\right)c_3^2 + 2\left(-1 + Q_{00}\right)c_2c_4\right)e_n^5 \\
&+ \left(\left(10 - 9Q_{00} - \frac{Q_{02}}{2} + 7Q_{01}\right)c_2^5\right. \\
&\quad + \frac{1}{2}\left(-60 + 58Q_{00} + Q_{02} - 26Q_{01}\right)c_2^3c_3 \\
&\quad + 2\left(6 - 6Q_{00} + Q_{01}\right)c_2^2c_4 + 7\left(-1 + Q_{00}\right)c_3c_4 \\
&\quad + c_2\left(\left(18 - 18Q_{00} + 4Q_{01}\right)c_3^2\right. \\
&\quad \left.\left. + 3\left(-1 + Q_{00}\right)c_5\right)\right)e_n^6 \\
&+ \left(\left(-20 + 14Q_{00} + 5Q_{02} - \frac{Q_{03}}{6} - Q_{10} - 29Q_{01}\right)c_2^6\right. \\
&\quad - \frac{1}{6}\left(-480 + 408Q_{00} + 54Q_{02} - Q_{03} - 12Q_{10}\right. \\
&\quad \left.-474Q_{01}\right)c_2^4c_3 \\
&\quad + \left(-40 + 38Q_{00} + Q_{02} - 21Q_{01}\right)c_2^3c_4 \\
&\quad + c_2^2\left(\left(-80 + 76Q_{00} + 3Q_{02} - Q_{10} - 50Q_{01}\right)c_3^2\right. \\
&\quad \left. + \left(16 - 16Q_{00} + 3Q_{01}\right)c_5\right) \\
&\quad + 2\left(\left(6 - 6Q_{00} + 2Q_{01}\right)c_3^3 + 3\left(-1 + Q_{00}\right)c_4^2\right. \\
&\quad \left. + 5\left(-1 + Q_{00}\right)c_3c_5\right) \\
&\quad + c_2\left(2\left(26 - 26Q_{00} + 7Q_{01}\right)c_3c_4\right. \\
&\quad \left.\left. + 4\left(-1 + Q_{00}\right)c_6\right)\right)e_n^7 \\
&+ \frac{1}{6}\left(\left(216 - 90Q_{00} - 174Q_{02} + 13Q_{03} + 54Q_{10}\right.\right. \\
&\quad \left.-6Q_{11} + 558Q_{01}\right)c_2^7 \\
&\quad + \left(-1068 + 690Q_{00} + 450Q_{02} - 23Q_{03}\right. \\
&\quad \left.-156Q_{10} + 12Q_{11} - 2004Q_{01}\right)c_2^5c_3 \\
&\quad + 2\left(303 - 246Q_{00} - 45Q_{02} + Q_{03}\right. \\
&\quad \left.+12Q_{10} + 348Q_{01}\right)c_2^4c_4 \\
&\quad + c_2^3\left(-2\left(-756 + 624Q_{00} + 147Q_{02} - 4Q_{03}\right.\right. \\
&\quad \left.-63Q_{10} + 3Q_{11} - 987Q_{01}\right)c_3^2 \\
&\quad \left.+3\left(-102 + 96Q_{00} + 3Q_{02} - 58Q_{01}\right)c_5\right) \\
&\quad + 3c_2^2\left(\left(-418 + 390Q_{00} + 21Q_{02}\right.\right. \\
&\quad \left.- 8Q_{10} - 310Q_{01}\right)c_3c_4 \\
&\quad \left.+ 8\left(5 - 5Q_{00} + Q_{01}\right)c_6\right) \\
&\quad - 6\left(10\left(-5 + 5Q_{00} - 2Q_{01}\right)c_3^2c_4\right. \\
&\quad \left.- 17\left(-1 + Q_{00}\right)c_4c_5\right. \\
&\quad \left.- 13\left(-1 + Q_{00}\right)c_3c_6\right) \\
&\quad + 6c_2\left(\left(-91 + 87Q_{00} + 6Q_{02}\right.\right. \\
&\quad \left.\left.-4Q_{10} - 76Q_{01}\right)c_3^3\right.
\end{aligned}
$$

$$+\left(37-37Q_{00}+12Q_{01}\right)c_4^2 +4\left(17-17Q_{00}+5Q_{01}\right)c_3c_5 +5\left(-1+Q_{00}\right)c_7\right)e_n^8+O\left(e_n^9\right). \tag{31}$$

For obtaining an iterative method of order eight, the coefficients of e_n^4, e_n^5, e_n^6, and e_n^7 in the error equation (31) must be zero simultaneously. After simplifications, we have the following equations involving Q_{00}, Q_{10}, Q_{01}, Q_{02}, and Q_{03},

$$\begin{gathered}
\left(-1+Q_{00}\right)=0,\\
\left(-4+4Q_{00}-Q_{01}\right)=0,\\
\left(8-8Q_{00}+Q_{01}\right)=0,\\
\left(10-9Q_{00}-\frac{Q_{02}}{2}+7Q_{01}\right)=0,\\
\left(-60+58Q_{00}+Q_{02}-26Q_{01}\right)=0,\\
\left(6-6Q_{00}+Q_{01}\right)=0,\\
\left(18-18Q_{00}+4Q_{01}\right)=0,\\
\left(-20+14Q_{00}+5Q_{02}-\frac{Q_{03}}{6}-Q_{10}-29Q_{01}\right)=0,\\
\left(-480+408Q_{00}+54Q_{02}-Q_{03}-12Q_{10}-474Q_{01}\right)=0,\\
\left(-40+38Q_{00}+Q_{02}-21Q_{01}\right)=0,\\
\left(-80+76Q_{00}+3Q_{02}-Q_{10}-50Q_{01}\right)=0,\\
\left(16-16Q_{00}+3Q_{01}\right)=0,\\
\left(6-6Q_{00}+2Q_{01}\right)=0,\\
\left(26-26Q_{00}+7Q_{01}\right)=0.
\end{gathered} \tag{32}$$

After simplifying (32), we have the following conditions on the weight function:

$$\begin{gathered}
Q_{00}=1,\qquad Q_{10}=2,\qquad Q_{01}=0,\\
Q_{02}=2,\qquad Q_{03}=12.
\end{gathered} \tag{33}$$

Finally, we get the following error equation:

$$\begin{aligned}
e_{n+1}=&-c_2^2\left(c_2^2-c_3\right)\\
&\times\left(\left(-7+Q_{11}\right)c_2^3-\left(-4+Q_{11}\right)c_2c_3-c_4\right)e_n^8\\
&+O\left(e_n^9\right).
\end{aligned} \tag{34}$$

This reveals that the three-step class of Ostrowski's method (19) reaches the optimal order of convergence eight by using only four functional evaluations per full iteration. This completes the proof of the Theorem 1. □

4. Special Cases

In this section, we will consider some particular cases of the proposed scheme (19) depending upon the weight function $Q(u, v)$ as follows.

Case 1. Let us consider the following weight function:

$$Q(u,v)=(au+1)v^2+2u+2v^3+1. \tag{35}$$

It can be easily seen that the abovementioned weight function $Q(u, v)$ satisfies all the conditions of Theorem 1. Therefore, we obtain a new optimal family of eighth-order methods given by

$$\begin{aligned}
y_n&=x_n-\frac{f(x_n)}{f'(x_n)},\\
z_n&=y_n-\frac{f(x_n)f(y_n)}{f'(x_n)\left(f(x_n)-2f(y_n)\right)},\\
x_{n+1}&=z_n-\Big(f(y_n)f(z_n)\\
&\quad\times\left[2\{f(y_n)\}^3+\{f(y_n)\}^2\right.\\
&\quad\times\left(f(x_n)+af(z_n)\right)\\
&\quad\left.+\{f(x_n)\}^2\left(f(x_n)+2f(z_n)\right)\right]\Big)\\
&\quad\times\Big(f'(x_n)\{f(x_n)\}^2\left(f(x_n)-2f(y_n)\right)\\
&\quad\times\left(f(y_n)-f(z_n)\right)\Big)^{1/2}.
\end{aligned} \tag{36}$$

Case 2. Let us consider the following weight function:

$$Q(u,v)=1-\frac{v}{2}+\frac{4bu+bv}{2b-2u-4bv+4uv}. \tag{37}$$

It can be easily seen that the abovementioned weight function $Q(u, v)$ satisfies all the conditions of Theorem 1. Therefore, we obtain a new optimal family of eighth-order method given by

$$\begin{aligned}
y_n&=x_n-\frac{f(x_n)}{f'(x_n)},\\
z_n&=y_n-\frac{f(x_n)f(y_n)}{f'(x_n)\left(f(x_n)-2f(y_n)\right)},\\
x_{n+1}&=z_n-\Bigg(f(x_n)f(y_n)f(z_n)\\
&\quad\times\Bigg(1-\frac{f(y_n)}{2f(x_n)}\\
&\quad-\left(bf(x_n)\left(f(y_n)+4f(z_n)\right)\right)\\
&\quad\times\left(2\left(2f(y_n)-f(x_n)\right)\right.\\
&\quad\left.\times\left(bf(x_n)-f(z_n)\right)\right)^{1/2}\Bigg)\Bigg)\\
&\quad\times\Big(f'(x_n)\left(-2f(y_n)+f(x_n)\right)\\
&\quad\times\left(f(y_n)-f(z_n)\right)\Big)^{1/2}.
\end{aligned} \tag{38}$$

Case 3. Let us consider the following weight function:

$$Q(u,v)=\frac{3}{4}-\frac{v}{2}+\frac{b+8bu}{4b-8bv-4uv+8uv^2}. \tag{39}$$

Table 1: Test problems.

$f(x)$	r	$[a, b]$
$f_1(x) = x^5 + x^4 + 4x^2 - 15$	1.34742809896830498150671538071482 12	[1.1, 1.6]
$f_2(x) = \sin x - \frac{x}{3}$	2.27886266007582831269995110456188 86	[2.0, 3.0]
$f_3(x) = e^{x^2+7x-30} - 1$	3.00000000000000000000000000000000000	[2.90, 3.30]
$f_4(x) = xe^{-x}$	0.00000000000000000000000000000000000	[−1.0, 0.5]
$f_5(x) = e^{-x} + \cos x$	1.74613953040801241765070308895378 02	[1.2, 2.5]
$f_6(x) = 10xe^{-x^2} - 1$	1.67963061042844994067492033883797 03	[1.5, 1.8]

Table 2: Comparison of different eighth-order iterative methods with the same total number of functional evaluations (TNFE = 12).

$f(x)$	Initial guess	SSM_8	LM_8	TM_8	SM_8	$MOM1_8$ $b = -\frac{1}{2}$	$MOM2_8$ $b = -\frac{1}{4}$
1	1.1	$3.0e-308$	$4.94e-242$	$1.1e-156$	$1.5e-223$	$7.6e-319$	$2.3e-314$
	1.6	$1.9e-340$	$2.2e-297$	$1.9e-233$	$6.7e-297$	$1.2e-417$	$4.1e-429$
2	2.0	$9.83e-386$	$3.8e-331$	$3.2e-288$	$1.7e-339$	$1.1e-437$	$1.2e-431$
	2.5	$9.8e-506$	$3.6e-462$	$5.5e-436$	$1.3e-468$	$7.7e-556$	$7.8e-557$
3	2.92	$1.4e-173$	$2.8e-92$	D	$1.9e-42$	$3.0e-177$	$1.2e-163$
	3.20	$1.1e-68$	$2.3e-52$	$5.6e-27$	$2.1e-51$	$8.0e-77$	$3.0e-77$
	3.30	$1.8e-28$	$3.8e-20$	$1.1e-8$	$8.3e-20$	$1.5e-34$	$3.3e-35$
4	−0.5	$3.0e-222$	$1.5e-188$	$1.5e-139$	$1.4e-193$	$1.1e-249$	$7.8e-252$
	0.5	$3.5e-123$	$3.5e-38$	D	$1.1e-3$	$1.5e-165$	$3.7e-112$
5	1.2	$2.2e-516$	$6.30e-444$	$1.3e-441$	$1.3e-456$	$3.0e-527$	$1.7e-527$
	2.5	$6.0e-222$	$4.5e-157$	$1.6e-156$	$7.3e-165$	$4.5e-266$	$2.1e-232$
6	1.5	$4.9e-399$	$1.5e-351$	$9.4e-316$	$1.2e-369$	$1.4e-477$	$9.2e-491$
	1.8	$4.6e-445$	$3.8e-385$	$1.0e-325$	$2.1e-394$	$5.5e-463$	$2.2e-461$

It can be easily seen that the abovementioned weight function $Q(u, v)$ satisfies all the conditions of Theorem 1. Therefore, we obtain a new optimal family of eighth-order method given by

$$y_n = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$z_n = y_n - \frac{f(x_n) f(y_n)}{f'(x_n)(f(x_n) - 2f(y_n))},$$

$$\begin{aligned} x_{n+1} &= z_n - (f(y_n) f(z_n)) \\ &\quad \times \left(4f'(x_n)(-2f(y_n) + f(x_n))^2 (f(y_n) - f(z_n)) \right. \\ &\qquad \left. \times \left(-b\{f(x_n)\}^2 + f(y_n) f(z_n)\right)\right)^{1/2} \\ &\quad \times \left[f(y_n)\left\{4\{f(y_n)\}^2 - 8f(y_n) f(x_n) + 3\{f(x_n)\}^2\right\}\right. \\ &\qquad \times f(z_n) - 4b\{f(x_n)\}^2 \\ &\qquad \times \left\{\{f(y_n)\}^2 - 2f(y_n) f(x_n)\right. \\ &\qquad\quad \left.\left. + f(x_n)(f(x_n) + 2f(z_n))\right\}\right]. \end{aligned} \tag{40}$$

It is straightforward to see that per step all the proposed family of methods require four functional evaluation, namely, $f(x_n)$, $f(y_n)f(z_n)$, and $f'(x_n)$. In order to obtain an assessment of the efficiency of our proposed methods, one will make use of efficiency index [1]. For newly proposed eighth-order three-point methods, one finds $p = 8$ and $d = 4$ to get $E = \sqrt[4]{8} \approx 1.682$ which is better than $E = \sqrt{2} \approx 1.414$, the efficiency index of Newton's method. Further, by choosing different kinds of weight functions one can develope several new optimal families of eight-order multipoint methods.

5. Numerical Experiments

In this section, we will check the effectiveness of the new optimal methods. We employ the present methods (38) (for $b = -1/2$) and (40) (for $b = -1/4$) denoted by $MOM1_8$ and $MOM2_8$, respectively, to solve nonlinear equations given in Table 1. We compare them with J. R. Sharma and R. Sharma method (SSM_8), Liu and Wang method (LWM_8), Thukral method [9] (TM_8), and Soleymani method (SM_8), respectively. For better comparisons of our proposed methods, we have given two comparison tables in each example: one is corresponding to absolute error value of given nonlinear functions (with the same total number of functional evaluations = 12) and the other is with respect to number of iterations taken by each method to obtain the root correct up to 35 significant digits in Tables 2 and 3, respectively. All

Table 3: Comparison of different eighth-order iterative methods with respect to number of iterations.

$f(x)$	Initial guess	SSM_8	LM_8	TM_8	SM_8	$MOM1_8$ $b=-\frac{1}{2}$	$MOM2_8$ $b=-\frac{1}{4}$
1	1.1	3	4	4	4	3	3
	1.6	3	3	4	3	3	3
2	2.0	3	3	3	3	3	3
	2.5	3	3	3	3	3	3
3	2.92	4	4	D	4	4	4
	3.30	5	5	5	5	4	4
4	−0.5	4	4	4	4	4	4
	0.5	4	4	D	6	4	4
5	1.2	3	3	3	3	3	3
	2.5	3	4	4	4	4	4
6	1.5	3	3	3	3	3	3
	1.8	3	3	3	3	3	3

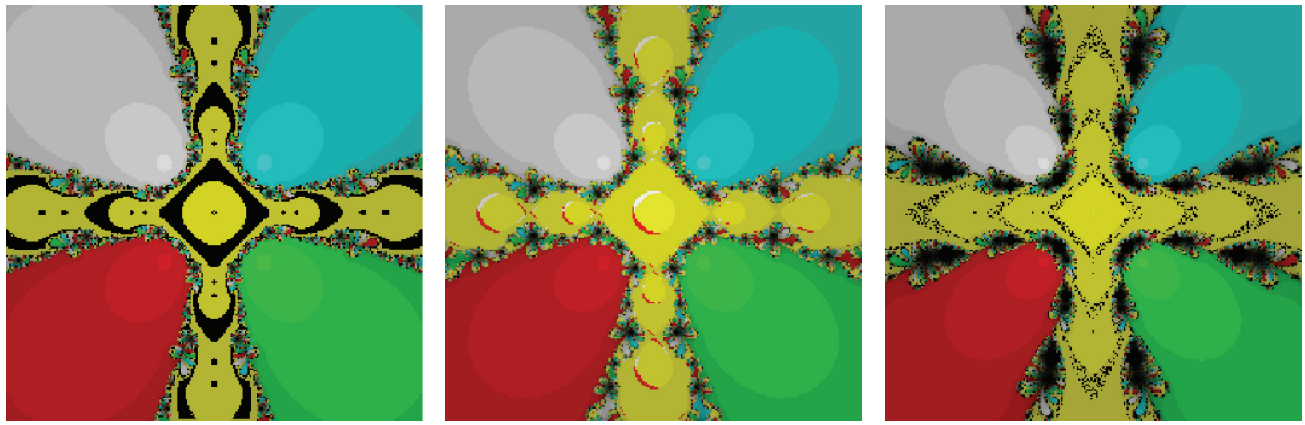


Figure 1: The basins of attraction for SSM_8, LW_8, and TM_8, respectively, in problem 1.

computations have been performed using the programming package *Mathematica* 9 with multiple precision arithmetic. We use $\epsilon = 10^{-34}$ as a tolerance error. The following stopping criteria are used for computer programs:

(i) $|x_{n+1} - x_n| < \epsilon$,

(ii) $|f(x_{n+1})| < \epsilon$.

6. Attractor Basins in the Complex Plane

We here investigate the comparison of the attained multiple root finders in the complex plane using basins of attraction. It is known that the corresponding fractal of an iterative root-finding method is a boundary set in the complex plane, which is characterized by the iterative method applied to a fixed polynomial $p(z) \in \mathbb{C}$; see, for example, [10, 11]. The aim herein is to use basin of attraction as another way for comparing the iteration algorithms.

From the dynamical point of view, we consider a rectangle $D = [-3, 3] \times [-3, 3] \in \mathbb{C}$ with a 400×400 grid, and we assign a color to each point $z_0 \in D$ according to the multiple root at which the corresponding iterative method starting from z_0 converges, and we mark the point as black if the method does not converge. In this section, we consider the stopping criterion for convergence to be less than 10^{-4} wherein the maximum number of full cycles for each method is considered to be 200. In this way, we distinguish the attraction basins by their colors for different methods.

Test Problem 1. Let $p_1(z) = (z^5 + z)$, having simple zeros $\{-0.707107 - 0.707107i, -0.707107 + 0.707107i, 0, 0.707107 - 0.707107i, 0.707107 + 0.707107i\}$. It is straight forward to see from Figures 1 and 2 that our methods, namely, OM_8^1 and OM_8^2, contain lesser number of divergent points in comparison to the methods, namely, SSM_8, LW_8, and TM_8. Further, our methods have also less chaotic behavior than other methods, namely, LW_8^1 and SM_8^2.

Test Problem 2. Let $p_2(z) = (z^4 - 1)$, having simple zeros $\{-1, -i, i, 1\}$. It is straight forward to see from Figures 3 and 4 that our method, namely, OM_8^1 and OM_8^2, performed better and larger basins of attraction as compared to the other methods, namely, SSM_8, LW_8, TM_8, and SM_8. Further, our

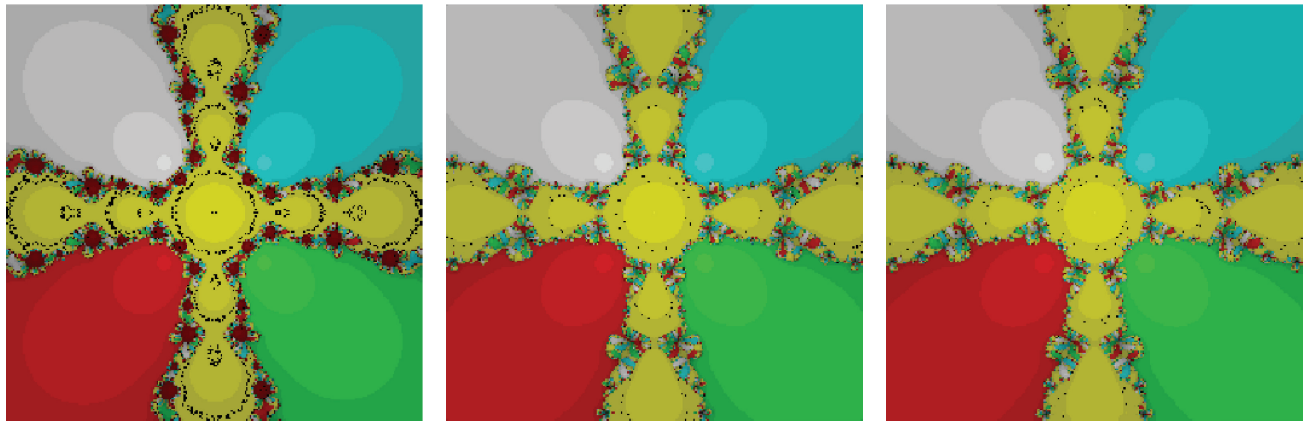


Figure 2: The basins of attraction for SM_8, OM_8^1, and OM_8^2, respectively, in problem 1.

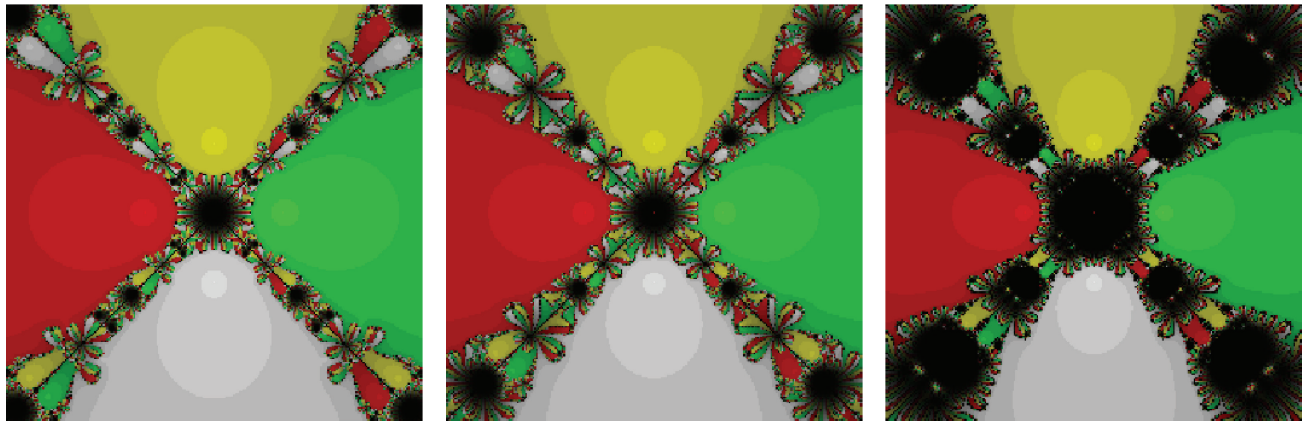


Figure 3: The basins of attraction for SSM_8, LW_8, and TM_8, respectively, in problem 2.

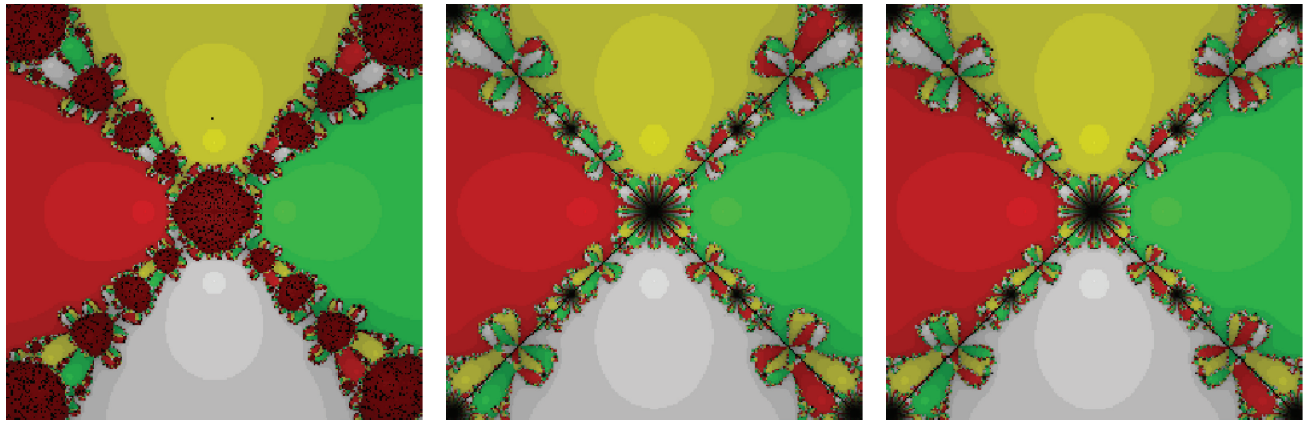


Figure 4: The basins of attraction for SM_8, OM_8^1, and OM_8^2, respectively, in problem 2.

methods have lesser number of divergent points and less chaotic behavior in comparison with other methods.

Test Problem 3. Let $p_3(z) = (z^3 + 2z - 1)$, having simple zeros $\{-0.226699 - 1.46771i,\ -0.226699 + 1.46771i,\ 0.453398\}$. It is straight forward to see from Figures 5 and 6 that our method, namely, OM_8^1 and OM_8^2, performed better and larger basins of attraction as compared to the other methods, namely, SSM_8, LW_8, TM_8, and SM_8. Further, our methods have less number of divergent points as compared to method TM_8. Note that our methods have also less chaotic behavior as compared to method SM_8.

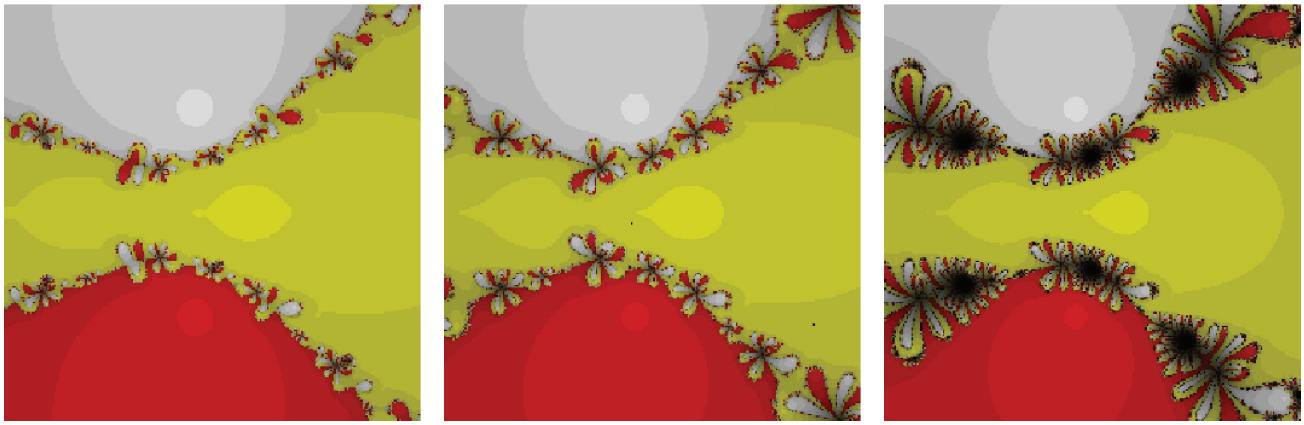


Figure 5: The basins of attraction for SSM_8, LW_8, and TM_8, respectively, in problem 3.

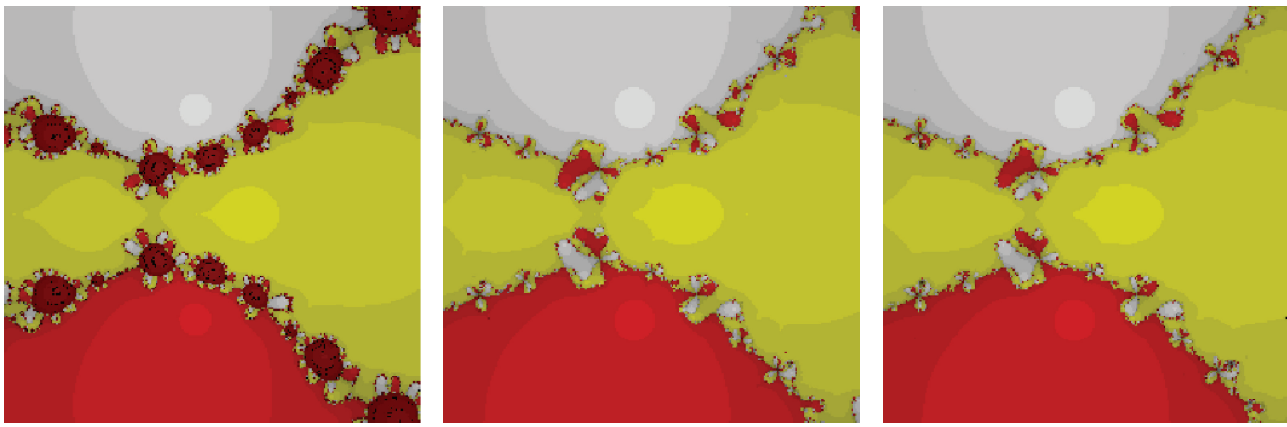


Figure 6: The basins of attraction for SM_8, OM_8^1, and OM_8^2, respectively, in problem 3.

7. Conclusions

In this paper, we have obtained a simple and elegant families of Ostrowski's method with optimal order of convergence eight by using an additional evaluation of function at the point iterated by Ostrowski's method. Its geometric construction consists in approximating f_n' at z_n in such a way that its average with the known tangent slopes f_n' at x_n and y_n is the same as the known weighted average of secant slopes and then we apply weight function approach. Further, we can also obtain many new optimal families of eighth-order Ostrowski's method by considering different kinds of weight functions which satisfy the conditions mentioned in Theorem 1. Each member of the proposed family requires three evaluations of the function f and one of its first-order derivative f' per full step and therefore has efficiency index better than fourth-order convergent Ostrowski's method. The superiority of present methods is also corroborated by numerical results displayed in Table 2. Our proposed iterative methods are compared in their efficiency and performance to various other multipoint methods, and it is observed that our proposed methods are efficient and perform better than existing methods available in the literature. Based on Figures 1–6, we conclude that larger basins of attraction belong to our methods, namely, OM_8^1 and OM_8^2, although the other methods are slow and have darker basins while some of the methods are too sensitive upon the choice of the initial value. Further, this idea can also be extended for the case of King's family.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. M. Ostrowski, *Solutions of Equations and System of Equations*, Academic Press, New York, NY, USA, 1960.

[2] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the ACM*, vol. 21, no. 4, pp. 643–651, 1974.

[3] R. F. King, "A family of fourth order methods for nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 876–879, 1973.

[4] V. Kanwar, R. Behl, and K. K. Sharma, "Simply constructed family of a Ostrowski's method with optimal order of convergence,"

Computers & Mathematics with Applications, vol. 62, no. 11, pp. 4021–4027, 2011.

[5] J. R. Sharma and R. Sharma, "A new family of modified Ostrowski's methods with accelerated eighth order convergence," *Numerical Algorithms*, vol. 54, no. 4, pp. 445–458, 2010.

[6] L. Liu and X. Wang, "Eighth-order methods with high efficiency index for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3449–3454, 2010.

[7] F. Soleymani, S. Karimi Vanani, M. Khan, and M. Sharifi, "Some modifications of King's family with optimal eighth order of convergence," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 1373–1380, 2012.

[8] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall Series in Automatic Computation, Prentice-Hall, Englewood Cliffs, NJ, USA, 1964.

[9] R. Thukral, "A new eighth-order iterative method for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 1, pp. 222–229, 2010.

[10] M. Scott, B. Neta, and C. Chun, "Basin attractors for various methods," *Applied Mathematics and Computation*, vol. 218, no. 6, pp. 2584–2599, 2011.

[11] B. Neta, M. Scott, and C. Chun, "Basins of attraction for several methods to find simple roots of nonlinear equations," *Applied Mathematics and Computation*, vol. 218, no. 21, pp. 10548–10556, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 346571, 9 pages
http://dx.doi.org/10.1155/2015/346571

Research Article

Semilocal Convergence Theorem for the Inverse-Free Jarratt Method under New Hölder Conditions

Yueqing Zhao,[1] Rongfei Lin,[1] Zdenek Šmarda,[2] Yasir Khan,[3] Jinbiao Chen,[1] and Qingbiao Wu[3]

[1]*Department of Mathematics, Taizhou University, Linhai, Zhejiang 317000, China*
[2]*Department of Mathematics, Brno University of Technology, 616 00 Brno, Czech Republic*
[3]*Department of Mathematics, Zhejiang University, Hangzhou, Zhejiang 310027, China*

Correspondence should be addressed to Yasir Khan; yasirmath@yahoo.com

Received 14 July 2014; Accepted 21 August 2014

Academic Editor: Fazlollah Soleymani

Under the new Hölder conditions, we consider the convergence analysis of the inverse-free Jarratt method in Banach space which is used to solve the nonlinear operator equation. We establish a new semilocal convergence theorem for the inverse-free Jarratt method and present an error estimate. Finally, three examples are provided to show the application of the theorem.

1. Introduction

We consider the following boundary value problem:

$$\begin{aligned} x'' &= -\lambda G(x), \\ x(a) &= x_a, \qquad x(b) = x_b. \end{aligned} \tag{1}$$

Those are equivalent to the following nonlinear integral equation (see [1, 2]):

$$x(s) = \alpha(s) + \lambda \int_a^b k(s,t)\, G(x(t))\, dt, \tag{2}$$

where $\alpha(s) = (1/(b-a))(x_a(b-s) + x_b(s-a))$ and $G : \Omega \subset C[a,b] \to C[a,b]$ is a twice Fréchet-differentiable operator. $C[a,b]$ is the set of all continuous functions in $[a,b]$; $k(s,t)$ is the Green function:

$$k(s,t) = \begin{cases} \dfrac{(b-s)(t-a)}{b-a}, & t \le s, \\ \dfrac{(s-a)(b-t)}{b-a}, & s \le t. \end{cases} \tag{3}$$

Instead of (2), we can try to solve a nonlinear operator equation $F(s) = 0$, where

$$\begin{aligned} F &: \Omega \subset C[a,b] \longrightarrow C[a,b], \\ F(x)(s) &= x(s) - \alpha(s) - \lambda \int_a^b k(s,t)\, G(x(t))\, dt. \end{aligned} \tag{4}$$

Solving the nonlinear operator equation is an important issue in the engineering and technology field as these kinds of problems appear in many real-world applications. Economics [3], chemistry [4], and physics [5–8] are some of the examples of the scientific and engineering technology areas applied to solve these type of equations. In this study, we consider to establish a new semilocal convergence theorem of the Jarratt method in Banach space which is used to solve the nonlinear operator equation

$$F(x) = 0, \tag{5}$$

where F is defined on an open convex Ω of a Banach space X with values in a Banach space Y.

There are a lot of methods of finding a solution of equation $F(x) = 0$. Particularly iterative methods are often used to solve this problem (see [1, 2, 9, 10]). If we use the famous Newton method, we can proceed as

$$x_{n+1} = x_n - F'(x_n)^{-1} F(x_n), \quad (n \ge 0)\ (x_0 \in \Omega). \tag{6}$$

Under a reasonable hypothesis, Newton's method is the second-order convergence.

To improve the convergence order, many modified methods have been presented. The famous Halley's method and the supper-Halley method are the third-order convergence.

References [11–22] give the convergence analysis for these methods. Now, we consider the following Jarratt method (see [23–25]):

$$y_n = x_n - F'(x_n)^{-1} F(x_n),$$
$$H(x_n, y_n) = \frac{3}{2} F'(x_n)^{-1} \left[F'\left(x_n + \frac{2}{3}(y_n - x_n) \right) - F'(x_n) \right],$$
$$x_{n+1} = y_n - \frac{1}{2} H(x_n, y_n) \left[I - H(x_n, y_n) \right] (y_n - x_n). \tag{7}$$

In this paper, we discuss the convergence of (7) for solving nonlinear operator equations in Banach spaces and establish a new semilocal convergence theorem under the following condition (see [20, 21]):

$$\left\| F'''(x) - F'''(y) \right\| \le \omega(\|x - y\|), \tag{8}$$

where $\omega : [0, +\infty) \to R$ is a nondecreasing continuous function. Finally, the corresponding error estimate is also given.

2. Main Results

In the section, we establish a new semilocal convergence theorem and present the error estimate. Denote $B(x, r) = \{y \in X \mid \|y - x\| < r\}$ and $\overline{B(x, r)} = \{y \in X \mid \|y - x\| \le r\}$. Suppose that X and Y are the Banach spaces, Ω is an open convex of the Banach space X, and $F : \Omega \subset X \to Y$ has continuous Fréchet derivative of the third-order. $F'(x_0)^{-1}$ exists, for some $x_0 \in \Omega$, and F satisfies

$$\begin{aligned}
&(\mathrm{A1}) \left\| y_0 - x_0 \right\| = \left\| F'(x_0)^{-1} F(x_0) \right\| \le \eta;\\
&(\mathrm{A2}) \left\| F'(x_0)^{-1} F''(x) \right\| \le M, \quad x \in \Omega,\ M \ge 0;\\
&(\mathrm{A3}) \left\| F'(x_0)^{-1} F'''(x) \right\| \le N, \quad x \in \Omega,\ N \ge 0;\\
&(\mathrm{A4}) \left\| F'(x_0)^{-1} \left[F'''(x) - F'''(y) \right] \right\| \le \omega(\|x - y\|), \quad x, y \in \Omega.
\end{aligned} \tag{9}$$

(A5) $\omega(z)$ is a nondecreasing continuous real function for $z > 0$ such that $\omega(0) \ge 0$, and there exists a positive real number $p \in (0, 1]$ such that $\omega(tz) \le t^p \omega(z)$ for $t \in [0, 1]$ and $z \in [0, +\infty)$.

(A6) Denote $A = \int_0^1 \int_0^1 t(1-t)(st)^p \, ds \, dt = (1/(p+1)(p+2)(p+3))$, $B = (1/3) \int_0^1 \int_0^1 (2st/3)^p t \, ds \, dt = (2^p/3^{p+1}(p+1)(p+2))$. Let $a_0 = M\eta$, $b_0 = N\eta^2$, $c_0 = \eta^2 \omega(\eta)$, $a_{n+1} = a_n f^2(a_n) g(a_n, b_n, c_n)$, $b_{n+1} = b_n f^3(a_n)\, g^2(a_n, b_n, c_n)$, $c_{n+1} = f^{3+p}(a_n) g^{2+p}(a_n, b_n, c_n)$, where

$$f(x) = \frac{2}{2 - 2x - x^2 - x^3},$$
$$g(x, y, z) = \frac{5x^3 + 2x^4 + x^5}{8} + \frac{xy}{12} + (A + B) z. \tag{10}$$

First, we get some lemmas.

Lemma 1. *Suppose that $f(x)$, $g(x, y, z)$ are given by (10). Then*

$\forall x \in (0, 1/2)$, $f(x)$ is increasing and $f(x) > 1$;

$\forall x \in (0, 1/2)$, $y > 0$, $g(x, y, z)$ is increasing;

$\forall \gamma \in (0, 1)$, $x \in (0, 1/2)$, $p > 0$, $f(\gamma x) < f(x)$ and $g(\gamma x, \gamma^2 y, \gamma^{2+p} z) < \gamma^{2+p} g(x, y, z)$.

Lemma 2. *Suppose that $f(x)$, $g(x, y, z)$ are given by (10). If*

$$a_0 \in \left(0, \frac{1}{2}\right), \quad f^2(a_0)\, g(a_0, b_0, c_0) < 1, \tag{11}$$

then

(i) *the sequences $\{a_n\}$, $\{b_n\}$, $\{c_n\}$ are nonnegative and decreasing;*

(ii) $(1 + (a_n/2)(1 + a_n))a_n < 1$, $\forall n \ge 0$.

Proof. (i) When $n = 1$,

$$\begin{aligned}
0 \le a_1 &= a_0 f^2(a_0)\, g(a_0, b_0, c_0) \le a_0,\\
0 \le b_1 &= b_0 f^3(a_0)\, g^2(a_0, b_0, c_0) \le b_0,\\
0 \le c_1 &= c_0 f^{3+p}(a_0)\, g^{2+p}(a_0, b_0, c_0) \le c_0.
\end{aligned} \tag{12}$$

Suppose $a_j \le a_{j-1}, b_j \le b_{j-1}$ for $j = 1, 2, \ldots, n$. By Lemma 1, f and g are increasing; then

$$\begin{aligned}
a_{n+1} &= a_n f^2(a_n)\, g(a_n, b_n, c_n) \le a_n f^2(a_0)\, g(a_0, b_0, c_0) \le a_n,\\
b_{n+1} &= b_n f^3(a_n)\, g^2(a_n, b_n, c_n) \le b_n f^3(a_0)\, g^2(a_0, b_0, c_0) \le b_n,\\
c_{n+1} &= c_n f^{3+p}(a_n)\, g^{2+p}(a_n, b_n, c_n)\\
&\le c_n \left[f^2(a_0)\, g(a_0, b_0, c_0) \right]^{2+p} \le c_n.
\end{aligned} \tag{13}$$

(ii) By (i), $\{a_n\}$ is decreasing and $a_0 \in (0, 1/2)$. So, for all $n \ge 0$,

$$\left(1 + \frac{a_n}{2}(1 + a_n)\right) a_n \le \left(1 + \frac{a_0}{2}(1 + a_0)\right) a_0 < 1. \tag{14}$$

This completes the proof of Lemma 2. □

Lemma 3. *Suppose that the conditions of Lemma 2 hold. Denote $\gamma = a_1/a_0 = f^2(a_0) g(a_0, b_0, c_0) < 1$. Then*

(i) $a_n \le \gamma^{(3+p)^{n-1}} a_{n-1} \le \gamma^{(((3+p)^n - 1)/(2+p))} a_0$, $b_n \le (\gamma^{(3+p)^{n-1}})^2 b_{n-1} \le (\gamma^{(((3+p)^n - 1)/(2+p))})^2 b_0$, $c_n \le (\gamma^{(3+p)^{n-1}})^{2+p} c_{n-1} \le \gamma^{(3+p)^n - 1} c_0$ $\forall n \ge 1$;

(ii) $f(a_n) g(a_n, b_n, c_n) \le \gamma^{(3+p)^n - 1} f(a_0) g(a_0, b_0, c_0) = (\gamma^{(3+p)^n} / f(a_0))$, $\forall n \ge 1$.

Proof. First, by induction, we prove that (i) holds. Because $a_1 = \gamma a_0$ and $f(a_0) > 1$, we have

$$\begin{aligned}
b_1 &= b_0 f^3(a_0)\, g^2(a_0, b_0, c_0) \le \gamma^2 b_0,\\
c_1 &= c_0 f^{3+p}(a_0)\, g^{2+p}(a_0, b_0, c_0) \le \gamma^{2+p} c_0.
\end{aligned} \tag{15}$$

Suppose that (i) holds for $n \geq 1$. Then we get

$$\begin{aligned}
a_{n+1} &= a_n f^2(a_n)\, g(a_n, b_n, c_n) \\
&\leq \gamma^{(3+p)^{n-1}} a_{n-1} f^2\left(\gamma^{(3+p)^{n-1}} a_{n-1}\right) \\
&\quad \times g\left(\gamma^{(3+p)^{n-1}} a_{n-1}, \left(\gamma^{(3+p)^{n-1}}\right)^2 b_{n-1}, \left(\gamma^{(3+p)^{n-1}}\right)^{2+p} c_{n-1}\right) \\
&\leq \gamma^{(3+p)^{n-1}} a_{n-1} f^2(a_{n-1}) \left(\gamma^{(3+p)^{n-1}}\right)^{2+p} g(a_{n-1}, b_{n-1}, c_{n-1}) \\
&= \gamma^{(3+p)^n} a_{n-1} f^2(a_{n-1})\, g(a_{n-1}, b_{n-1}, c_{n-1}) = \gamma^{(3+p)^n} a_n, \\
a_{n+1} &\leq \gamma^{(3+p)^n} a_n \leq \gamma^{(3+p)^n} \gamma^{(3+p)^{n-1}} a_{n-1} \\
&\leq \cdots \leq \gamma^{(3+p)^n} \gamma^{(3+p)^{n-1}} \cdots \gamma^{(3+p)^0} a_0 \\
&= \gamma^{(((3+p)^{n+1}-1)/(2+p))} a_0, \\
b_{n+1} &= b_n f^3(a_n)\, g^2(a_n, b_n, c_n) \leq b_n \left(\frac{a_{n+1}}{a_n}\right)^2 \leq \left(\gamma^{(3+p)^n}\right)^2 b_n \\
&\leq \cdots \leq \left(\gamma^{(3+p)^n}\right)^2 \left(\gamma^{(3+p)^{n-1}}\right)^2 \cdots \left(\gamma^{(3+p)^0}\right)^2 b_0 \\
&= \left(\gamma^{(((3+p)^{n+1}-1)/(2+p))}\right)^2 b_0, \\
c_{n+1} &= c_n f^{3+p}(a_n)\, g^{2+p}(a_n, b_n, c_n) \\
&\leq c_n \left[f^2(a_n)\, g(a_n, b_n, c_n)\right]^{2+p} = c_n \left(\frac{a_{n+1}}{a_n}\right)^{2+p} \\
&\leq \left(\gamma^{(3+p)^n}\right)^{2+p} c_n \leq \cdots \leq \gamma^{(3+p)^{n+1}-1} c_0
\end{aligned} \tag{16}$$

and from (ii) we get

$$\begin{aligned}
&f(a_n)\, g(a_n, b_n, c_n) \\
&\quad \leq f\left(\gamma^{(((3+p)^n-1)/(2+p))} a_0\right) \\
&\qquad \times g\left(\gamma^{(((3+p)^n-1)/(2+p))} a_0, \left(\gamma^{(((3+p)^n-1)/(2+p))}\right)^2 b_0, \gamma^{3^n-1} c_0\right) \\
&\quad \leq \gamma^{(3+p)^n-1} f(a_0)\, g(a_0, b_0, c_0) = \frac{\gamma^{(3+p)^n}}{f(a_0)}, \quad n \geq 1.
\end{aligned} \tag{17}$$

This completes the proof of Lemma 3. □

Lemma 4. *Suppose that X and Y are Banach spaces, Ω is an open convex of the Banach space X, $F : \Omega \subset X \to Y$ has continuous Fréchet derivative of the second-order, and the sequences $\{x_n\}$, $\{y_n\}$ are generated by (7). Then, for all natural numbers $n \geq 0$, the following formula holds:*

$$\begin{aligned}
&F(x_{n+1}) \\
&= \int_0^1 F''(y_n + t(x_{n+1} - y_n))(1-t)\, dt\, (x_{n+1} - y_n)^2 \\
&\quad + \left[\int_0^1 F''(x_n + t(y_n - x_n))(1-t)\, dt \right. \\
&\qquad \left. - \frac{1}{2}\int_0^1 F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right) dt\right](y_n - x_n)^2 \\
&\quad - \frac{1}{2}\int_0^1 \left[F''(x_n + t(y_n - x_n)) \right. \\
&\qquad \left. - F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right)\right] dt \\
&\qquad \times (y_n - x_n) H(x_n, y_n)(y_n - x_n) \\
&\quad + \frac{1}{2}\int_0^1 F''(x_n + t(y_n - x_n))\, dt \\
&\qquad \times (y_n - x_n) H(x_n, y_n) H(x_n, y_n)(y_n - x_n).
\end{aligned} \tag{18}$$

Proof. Consider

$$\begin{aligned}
&F(y_n) \\
&\quad = F(y_n) - F(x_n) - F'(x_n)(y_n - x_n) \\
&\quad = \int_0^1 F''(x_n + t(y_n - x_n))(1-t)\, dt\, (y_n - x_n)^2, \\
&F'(y_n)(x_{n+1} - y_n) \\
&\quad = -\frac{1}{2}\left[F'(y_n) - F'(x_n)\right] H(x_n, y_n) \\
&\qquad \times \left[I - H(x_n, y_n)\right](y_n - x_n) \\
&\qquad - \frac{1}{2}F'(x_n) H(x_n, y_n)\left[I - H(x_n, y_n)\right](y_n - x_n) \\
&\quad = -\frac{1}{2}\int_0^1 F''(x_n + t(y_n - x_n))\, dt \\
&\qquad \times (y_n - x_n) H(x_n, y_n)(y_n - x_n) \\
&\qquad + \frac{1}{2}\int_0^1 F''(x_n + t(y_n - x_n))\, dt \\
&\qquad \times (y_n - x_n) H(x_n, y_n) H(x_n, y_n)(y_n - x_n) \\
&\qquad - \frac{1}{2}\int_0^1 F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right) dt\, (y_n - x_n)^2 \\
&\qquad + \frac{1}{2}\int_0^1 F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right) dt \\
&\qquad \times (y_n - x_n) H(x_n, y_n)(y_n - x_n),
\end{aligned}$$

$$
\begin{aligned}
&F(x_{n+1}) \\
&\quad = F(x_{n+1}) - F(y_n) - F'(y_n)(x_{n+1} - y_n) \\
&\qquad + F(y_n) + F'(y_n)(x_{n+1} - y_n) \\
&\quad = \int_0^1 F''(y_n + t(x_{n+1} - y_n))(1-t)\,dt\,(x_{n+1} - y_n)^2 \\
&\qquad + \left[\int_0^1 F''(x_n + t(y_n - x_n))(1-t)\,dt \right. \\
&\qquad\quad \left. -\frac{1}{2}\int_0^1 F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right)dt\right](y_n - x_n)^2 \\
&\qquad - \frac{1}{2}\int_0^1 \left[F''(x_n + t(y_n - x_n)) \right. \\
&\qquad\quad \left. - F''\left(x_n + \frac{2}{3}t(y_n - x_n)\right)\right]dt \\
&\qquad\quad \times (y_n - x_n) H(x_n, y_n)(y_n - x_n) \\
&\qquad + \frac{1}{2}\int_0^1 F''(x_n + t(y_n - x_n))\,dt \\
&\qquad\quad \times (y_n - x_n) H(x_n, y_n) H(x_n, y_n)(y_n - x_n).
\end{aligned} \tag{19}
$$

This completes the proof of Lemma 4.

By (A1)–(A6), (10), and (11), if $a_0 < 1/2$, then

$$
\begin{aligned}
\|H(x_0, y_0)\| &\le M\|y_0 - x_0\| \\
&= M\left\|F'(x_0)^{-1}F'(x_0)\right\|\|y_0 - x_0\| \le a_0, \\
\|x_1 - y_0\| &\le \frac{1}{2}\|H(x_0, y_0)\|\|I - H(x_0, y_0)\|\|y_0 - x_0\| \\
&\le \frac{a_0}{2}(1 + a_0)\|y_0 - x_0\|, \\
\|x_1 - x_0\| &\le \|x_1 - y_0\| + \|y_0 - x_0\| \\
&\le \left[1 + \frac{a_0}{2}(1 + a_0)\right]\|y_0 - x_0\| < R\eta,
\end{aligned} \tag{20}
$$

where $R = [1+(a_0/2)(1+a_0)](1/(1-f(a_0)g(a_0, b_0, c_0)))$; hence, $x_1, y_0 \in S(x_0, R\eta)$. Consider

$$
\begin{aligned}
&\left\|F'(x_0)^{-1}F'(x_1) - I\right\| \\
&\quad \le M\|x_1 - x_0\| \le \left[1 + \frac{a_0}{2}(1 + a_0)\right]a_0 < 1.
\end{aligned} \tag{21}
$$

By Banach lemma, $F'(x_1)^{-1}$ exists, and

$$
\left\|F'(x_1)^{-1}F'(x_0)\right\| \le f(a_0) = f(a_0)\left\|F'(x_0)^{-1}F'(x_0)\right\|. \tag{22}
$$

□

By Lemma 4, we have

$$
\begin{aligned}
&\Big\|F'(x_0)^{-1} \\
&\quad \times \int_0^1 F''(x_0 + t(y_0 - x_0))(1-t)\,dt \\
&\qquad - \frac{1}{2}F'(x_0)^{-1} \\
&\quad \times \int_0^1 F''\left(x_0 + \frac{2}{3}t(y_0 - x_0)\right)dt\Big\| \\
&= \Big\| F'(x_0)^{-1}\int_0^1 F''\left[(x_0 + t(y_0 - x_0)) \right. \\
&\qquad\qquad \left. - F''(x_0)\right](1-t)\,dt \\
&\qquad - \frac{1}{2}F'(x_0)^{-1} \\
&\qquad \times \int_0^1 \left[F''\left(x_0 + \frac{2}{3}t(y_0 - x_0)\right) - F''(x_0)\right]dt\Big\| \\
&= \Big\|F'(x_0)^{-1}\iint_0^1 F'''\left[(x_0 + st(y_0 - x_0)) \right. \\
&\qquad\qquad \left. - F'''(x_0)\right]ds \\
&\qquad\qquad \times t(1-t)\,dt\,(y_0 - x_0) \\
&\qquad - \frac{1}{3}F'(x_0)^{-1} \\
&\qquad \times \iint_0^1 \left[F'''\left(x_0 + \frac{2}{3}st(y_0 - x_0)\right) \right. \\
&\qquad\qquad \left. - F'''(x_0)\right]ds\,t\,dt\,(y_0 - x_0)\Big\| \\
&\le (A + B)\,\omega(\eta)\|(y_0 - x_0)\|,
\end{aligned}
$$

$$
\begin{aligned}
&\left\|F'(x_0)^{-1}F(x_1)\right\| \\
&\quad \le \frac{M}{2}\|x_1 - y_0\|^2 + \frac{N}{12}a_0\|y_0 - x_0\|^3 \\
&\qquad + \frac{M}{2}a_0^2\|y_0 - x_0\|^2 + (A + B)\,\omega(\eta)\|y_0 - x_0\|^3, \\
&\|y_1 - x_1\| \\
&\quad \le \left\|F'(x_1)^{-1}F'(x_0)\right\|\left\|F'(x_0)^{-1}F(x_1)\right\| \\
&\quad \le f(a_0)\,g(a_0, b_0, c_0)\|y_0 - x_0\|.
\end{aligned} \tag{23}
$$

Hence,

$$
\begin{aligned}
\|H(x_1, y_1)\| &\le M\left\|F'(x_1)^{-1}F'(x_0)\right\|\|y_1 - x_1\| \\
&\le Mf^2(a_0)\,g(a_0, b_0, c_0)\|y_0 - x_0\| = a_1,
\end{aligned}
$$

$$
\begin{aligned}
&N\left\|F'(x_1)^{-1}F'(x_0)\right\|\left\|y_1-x_1\right\|^2\\
&\quad\le Nf^3(a_0)g^2(a_0,b_0,c_0)\left\|y_0-x_0\right\|^2=b_1,\\
&\left\|F'(x_1)^{-1}F'(x_0)\right\|\omega\left(\left\|y_1-x_1\right\|\right)\left\|y_1-x_1\right\|^2\\
&\quad\le f^{3+p}(a_0)g^{2+p}(a_0,b_0,c_0)\omega(\eta)\left\|y_0-x_0\right\|^2=c_1.
\end{aligned}
\tag{24}
$$

Hence,

$$
\begin{aligned}
\left\|x_2-y_1\right\|&\le\frac{1}{2}a_1(1+a_1)\left\|y_1-x_1\right\|,\\
\left\|x_2-x_1\right\|&\le\left\|x_2-y_1\right\|+\left\|y_1-x_1\right\|\\
&\le\left(1+\frac{1}{2}a_1(1+a_1)\right)\left\|y_1-x_1\right\|,\\
\left\|x_2-x_0\right\|&\le\left\|x_2-x_1\right\|+\left\|x_1-x_0\right\|\\
&\le\left[1+\frac{a_0}{2}(1+a_0)\right]\left[f(a_0)g(a_0,b_0,c_0)+1\right]\\
&\quad\times\left\|y_0-x_0\right\|<R\eta.
\end{aligned}
\tag{25}
$$

By

$$
\begin{aligned}
&\left\|F'(x_1)^{-1}F'(x_2)-I\right\|\le M\left\|F'(x_1)^{-1}F'(x_0)\right\|\left\|x_2-x_1\right\|\\
&\quad\le a_1\left[1+\frac{a_1}{2}(1+a_1)\right]<1,
\end{aligned}
\tag{26}
$$

hence $F'(x_2)^{-1}F'(x_0)$ exists, and $\|F'(x_2)^{-1}F'(x_0)\| \le f(a_1)\|F'(x_1)^{-1}F'(x_0)\|$. By induction, we can prove that the following Lemma 5 holds.

Lemma 5. *Under the hypotheses of Lemma 2, the following items are true for all $n\ge 1$:*

(I) $F'(x_n)^{-1}F'(x_0)$ *exists and* $\|F'(x_n)^{-1}F'(x_0)\| \le f(a_{n-1})\|F'(x_{n-1})^{-1}F'(x_0)\|$;

(II) $\|y_n-x_n\|\le f(a_{n-1})g(a_{n-1},b_{n-1},c_{n-1})\|y_{n-1}-x_{n-1}\|$;

(III) $H(x_n,y_n)\le M\|F'(x_n)^{-1}F'(x_0)\|\|y_n-x_n\|\le a_n$;

(IV) $N\|F'(x_n)^{-1}F'(x_0)\|\|y_n-x_n\|^2\le b_n$;

(V) $\|F'(x_n)^{-1}F'(x_0)\|\omega(\|y_n-x_n\|)\|y_n-x_n\|^2\le c_n$;

(VI) $\|x_{n+1}-y_n\|\le(a_n/2)(1+a_n)\|y_n-x_n\|$;

(VII) $\|x_{n+1}-x_n\|\le[1+(a_n/2)(1+a_n)]\|y_n-x_n\|$;

(VIII) $\|x_{n+1}-x_0\|\le R\eta$, *where* $R=[1+(a_0/2)(1+a_0)](1/(1-f(a_0)g(a_0,b_0,c_0)))$.

Theorem 6. *Let X and Y be two Banach spaces and $F:\Omega\subset X\to Y$ has continuous Fréchet derivative of the third-order on a nonempty open convex Ω. One supposes that $\Gamma_0=F'(x_0)^{-1}\in L(Y,X)$ exists for some $x_0\in\Omega$ and conditions (A1)–(A6) and (II) hold. If $\overline{S(x_0,R\eta)}\subset\Omega$, then the sequence $\{x_n\}$ generated by (7) is well defined and converges to a unique solution x^* of (2) in $S(x_0,(2/M)-R\eta)\cap\Omega$. Furthermore, the following error estimate is obtained:*

$$
\begin{aligned}
&\left\|x^*-x_n\right\|\\
&\quad\le\left[1+\frac{\gamma^{(((3+p)^n-1)/(2+p))}a_0}{2}\left(1+\gamma^{(((3+p)^n-1)/(2+p))}a_0\right)\right]\\
&\qquad\times\frac{1}{1-\gamma^{(3+p)^n}\Delta}\gamma^{(((3+p)^n-1)/(2+p))}\Delta^n\eta,
\end{aligned}
\tag{27}
$$

where $\gamma=f^2(a_0)g(a_0,b_0,c_0)=a_1/a_0$ and $\Delta=1/f(a_0)$, $R=(1+(a_0/2)(1+a_0))(1/(1-\gamma\Delta))$.

Proof. Firstly, we prove that the sequence $\{x_n\}$ is a Cauchy one. From (II) and by Lemma 3, we have

$$
\begin{aligned}
\left\|y_n-x_n\right\|&\le f(a_{n-1})g(a_{n-1},b_{n-1},c_{n-1})\left\|y_{n-1}-x_{n-1}\right\|\\
&\le\cdots\le\left(\prod_{i=0}^{n-1}f(a_i)g(a_i,b_i,c_i)\right)\eta\\
&\le\left(\prod_{i=0}^{n-1}\gamma^{(3+p)^i}\Delta\right)\eta=\gamma^{(((3+p)^n-1)/(2+p))}\Delta^n\eta.
\end{aligned}
\tag{28}
$$

For $n\ge 0$, $m\ge 1$,

$$
\begin{aligned}
&\left\|x_{n+m}-x_n\right\|\\
&\quad\le\left\|x_{n+m}-x_{n+m-1}\right\|+\left\|x_{n+m-1}-x_{n+m-2}\right\|\\
&\qquad+\cdots+\left\|x_{n+1}-x_n\right\|\\
&\quad\le\left[1+\frac{a_n}{2}(1+a_n)\right]\\
&\qquad\times\left(\left\|y_{n+m-1}-x_{n+m-1}\right\|+\cdots+\left\|y_{n+1}-x_n\right\|\right)\\
&\quad\le\left[1+\frac{a_n}{2}(1+a_n)\right]\\
&\qquad\times\left(\gamma^{(((3+p)^{n+m-1}-1)/(2+p))}\Delta^{n+m-1}\right.\\
&\qquad\qquad\left.+\cdots+\gamma^{(((3+p)^n-1)/(2+p))}\Delta^n\right)\eta\\
&\quad=\left[1+\frac{a_n}{2}(1+a_n)\right]\gamma^{(((3+p)^n-1)/(2+p))}\Delta^n\eta\\
&\qquad\times\left(\gamma^{((3+p)^n[(3+p)^{m-1}-1]/(2+p))}\Delta^{m-1}+\cdots+1\right).
\end{aligned}
\tag{29}
$$

By the Bernoulli inequality $(1+x)^k-1>kx$, so $(3+p)^k-1>k(2+p)$. Hence, we have

$$
\begin{aligned}
&\left\|x_{n+m}-x_n\right\|\\
&\quad<\left[1+\frac{a_n}{2}(1+a_n)\right]\frac{1}{1-\gamma^{(3+p)^n}\Delta}\gamma^{(((3+p)^n-1)/(2+p))}\Delta^n\eta.
\end{aligned}
\tag{30}
$$

Hence, $\{x_n\}$ is a Cauchy sequence and $x^* = \lim_{n\to\infty} x_n$. Obviously, $x_m \in B(x_0, R\eta)$, for all $m \geq 1$, as if $n = 0$ in (30); we obtain

$$\|x_m - x_0\| < \left(1 + \frac{a_0}{2}(1 + a_0)\right)\frac{1}{1 - \gamma\Delta}\eta = R\eta. \tag{31}$$

Following a similar procedure, we have $y_n \in B(x_0, R\eta)$, for all $n \geq 0$.

Now, let $n \to \infty$ in (28). It follows that $\|F'(x_n)^{-1} F(x_n)\| \to 0$. Besides $\|F(x_n)\| \to 0$, since $\|F(x_n)\| \leq \|F'(x_n)\| \|F'(x_n)^{-1}F(x_n)\|$ and $\{\|F'(x_n)\|\}$ is a bounded sequence, therefore $F(x^*) = 0$ by the continuity of F in $\overline{S(x_0, R\eta)}$.

By letting $m \to \infty$ in (30), we obtain error estimate (28).

To show uniqueness, let us assume that there exists a second solution y^* of (2) in $S(x_0, (2/M) - R\eta) \cap \Omega$. Then

$$\begin{aligned}
&\int_0^1 \left\|F'(x_0)^{-1}\left[F'(x^* + t(y^* - x^*)) - F'(x_0)\right]\right\| dt \\
&\leq M\int_0^1 \left\|x^* + t(y^* - x^*) - x_0\right\| dt \\
&\leq M\int_0^1 \left[(1-t)\left\|x^* - x_0\right\| + t\left\|y^* - x_0\right\|\right] dt \\
&< \frac{M}{2}\left(R\eta + \frac{2}{M} - R\eta\right) = 1.
\end{aligned} \tag{32}$$

By Banach lemma, we can obtain that the inverse of the linear operator $\int_0^1 F'(x^* + t(y^* - x^*))dt$ exists and

$$\int_0^1 F'(x^* + t(y^* - x^*))\, dt\, (y^* - x^*) = F(y^*) - F(x^*) = 0. \tag{33}$$

We get that $x^* = y^*$.

This completes the proof of Theorem 6. □

3. Application

In this section, we apply the convergence theorem and show three numerical examples.

Example 1. Consider the root of the equation $F(x) = x^{10/3} + x^{7/2} - x - 1 = 0$ on $x \in (0, +\infty)$. Then, we easily get that

$$F'''(x) = \frac{280}{27}x^{1/3} + \frac{105}{8}x^{1/2} \tag{34}$$

does not satisfy (K, p) Hölder condition

$$\left\|F'''(x) - F'''(y)\right\| \leq K\|x - y\|^p \tag{35}$$

because, for all $p \in (0, 1]$,

$$\sup_{x,y\in(0,+\infty)} \frac{(280/27)|x - y|^{1/3} + (105/8)|x - y|^{1/2}}{|x - y|^p} = +\infty. \tag{36}$$

Let

$$\omega(z) = \frac{280}{27}z^{1/3} + \frac{105}{8}z^{1/2}, \quad z > 0; \tag{37}$$

then $\omega(tz) \leq t^{1/3}\omega(z)$ for $t \in [0, 1]$ and $z \in [0, +\infty)$;

$$\left\|F'''(x) - F'''(y)\right\| \leq \omega\|x - y\|. \tag{38}$$

Let us consider a particular case of (2) from the operator given by the following nonlinear integral equation of mixed Hammerstein type (see [26]):

$$x(s) = \alpha(s) - \sum_{i=1}^{m}\int_a^b k(s,t)\,\varphi_i(x(t))\,dt, \tag{39}$$

where $-\infty < a < b < +\infty$, u, φ_i, for $i = 1, 2, \ldots, m$, are known functions and x is a solution to be determined. If φ''' is (L_i, p_i) Hölder continuous in Ω, for $i = 1, 2, \ldots, m$, the corresponding operator $F : \Omega \subseteq C[a, b] \to C[a, b]$,

$$\begin{aligned}
[F(x)](s) = x(s) + \sum_{i=1}^{m}\int_a^b k(s,t)\,\varphi_i(x(t))\,dt - \alpha(s), \\
s \in [a, b],
\end{aligned} \tag{40}$$

does not satisfy (K, p) Hölder condition; for instance, the max-norm is considered. In this case,

$$\begin{aligned}
\left\|F'''(x) - F'''(y)\right\| \leq \sum_{i=1}^{m} L_i\|x - y\|^{p_i}, \\
L_i > 0, \quad p_i \in (0, 1], \quad x, y \in \Omega.
\end{aligned} \tag{41}$$

To solve this type of equations, we can consider

$$\left\|F'''(x) - F'''(y)\right\| \leq \omega(\|x - y\|), \quad x, y \in \Omega, \tag{42}$$

where $\omega(z) = \Sigma_{i=1}^{m} L_i z^{p_i}$ satisfy $\omega(tz) \leq t^q\omega(z)$, where $q = \min\{p_i, p_2, \ldots, p_m\}$.

Remark 7. Observe that if F''' is Lipschitz continuous in Ω, we can choose $\omega(z) = Kz$, $K > 0$, so that Jarratt's method is of R-order, at least four order. If F''' is (L, p) Hölder continuous in Ω, then we can choose $\omega(z) = Lz^p$, $L < 0$, $p \in (0, 1]$, and Jarratt's method is of R-order, at least $3 + p$.

Example 2. Consider the case as follows:

$$\begin{aligned}
x(s) = 1 + \frac{1}{32}\int_0^1 k(s,t)\,x(t)^{16/5}\,dt \\
+ \frac{1}{30}\int_0^1 k(s,t)\,x(t)^{10/3}\,dt,
\end{aligned} \tag{43}$$

where the space is $X = C[0, 1]$ with the norm

$$\begin{aligned}
\|x\| = \max_{0\leq s\leq 1}|x(s)|, \\
k(s,t) = \begin{cases} t(1-s), & t \leq s, \\ s(1-t), & s \leq t. \end{cases}
\end{aligned} \tag{44}$$

This equation arises in the theory of the radiative transfer and neutron transport and in the kinetic theory of gasses. Let us define the operator F on X by

$$F(x) = x(s) - \frac{1}{32}\int_0^1 k(s,t)\,x(t)^{16/5}\,dt - \frac{1}{30}\int_0^1 k(s,t)\,x(t)^{10/3}\,dt - 1. \tag{45}$$

The first, the second, and the third derivatives of F are defined by

$$\begin{aligned}
F'(x)\,u(s) &= u(s) - \frac{1}{10}\int_0^1 k(s,t)\,x(t)^{11/5}\,u(t)\,dt - \frac{1}{9}\int_0^1 k(s,t)\,x(t)^{7/3}\,u(t)\,dt, \quad u \in X,\\
F''(x)(uv)(s) &= -\frac{11}{50}\int_0^1 k(s,t)\,x(t)^{6/5}\,u(t)\,v(t)\,dt - \frac{7}{27}\int_0^1 k(s,t)\,x(t)^{\frac{4}{3}}\,u(t)\,v(t)\,dt, \quad u \in X,\\
F'''(x)(uvw)(s) &= -\frac{66}{250}\int_0^1 k(s,t)\,x(t)^{1/5}\,u(t)\,v(t)\,w(t)\,dt - \frac{28}{81}\int_0^1 k(s,t)\,x(t)^{1/3}\,u(t)\,v(t)\,w(t)\,dt,
\end{aligned} \tag{46}$$

and we have

$$\begin{aligned}
&\left\|\left[F'''(x) - F'''(y)\right]uvw\right\|\\
&\quad\le \frac{66}{250}\max_{s\in[0,1]}\\
&\qquad\times\int_0^1 k(s,t)\left|\left(x(t)^{1/5} - y(t)^{1/5}\right)u(t)\,v(t)\,w(t)\right|dt\\
&\qquad+\frac{28}{81}\max_{s\in[0,1]}\\
&\qquad\times\int_0^1 k(s,t)\left|\left(x(t)^{1/3} - y(t)^{1/3}\right)u(t)\,v(t)\,w(t)\right|dt\\
&\quad\le \frac{66}{250}\times\frac{1}{8}\left\|x-y\right\|^{1/5}\|uvw\|\\
&\qquad+\frac{28}{81}\times\frac{1}{8}\left\|x-y\right\|^{1/3}\|uvw\|.
\end{aligned} \tag{47}$$

To apply Theorem 6, we choose $x_0 = x_0(s) = 1$ and we look for a domain in the form

$$\Omega = B(1,2) \subseteq C([0,1]). \tag{48}$$

In this case, we have

$$\left\|I - F'(x_0)\right\| \le \frac{19}{720} < 1 \tag{49}$$

and from the Banach lemma, we obtain

$$\begin{aligned}
&\left\|F'(x_0)^{-1}\right\| \le \frac{720}{701},\\
&\left\|F'(x_0)^{-1}F(x_0)\right\| \le \frac{720}{701}\times\frac{1}{8}\left(\frac{1}{32}+\frac{1}{30}\right) = \eta = \frac{93}{11216},\\
&M = 0.148766\cdots, \qquad N = 0.0948511\cdots,\\
&\omega(z) = \frac{33}{100z^{1/5}} + \frac{7}{162z^{1/3}}, \qquad p = \frac{1}{5}.
\end{aligned} \tag{50}$$

Then $a_0 = M\eta = 0.00123353 < 1/2$, $b_0 = 6.52127\times10^{-6}$, $c_0 = 1.47132\times10^{-6}$, $\gamma = f^2(a_0)g(a_0,b_0,c_0) = 3.48167\times10^{-7} < 1$, $\Delta = 0.998766\cdots$, and $R = 1.00062\cdots$. This means that the hypothesis of Theorem 6 is satisfied. Then, the error bound becomes

$$\begin{aligned}
&\left\|x^* - x_n\right\|\\
&\quad\le\left[1 + \frac{\gamma^{(((3.2)^n-1)/2.2)}a_0}{2}\left(1+\gamma^{(((3.2)^n-1)/2.2)}a_0\right)\right]\\
&\qquad\times\frac{1}{1-\gamma^{(3.2)^n}\Delta}\gamma^{(((3.2)^n-1)/2.2)}\Delta^n\eta.
\end{aligned} \tag{51}$$

For $n = 1, 2, 3, 4$, we get

$$\begin{aligned}
\left\|x_1 - x^*\right\| &\le 4.28944\times10^{-10},\\
\left\|x_2 - x^*\right\| &\le 5.76451\times10^{-16},\\
\left\|x_3 - x^*\right\| &\le 6.63209\times10^{-23},\\
\left\|x_4 - x^*\right\| &\le 2.86064\times10^{-32}.
\end{aligned} \tag{52}$$

Example 3. Let us consider the system of equations $F(u,v) = 0$, where

$$F(u,v) = \left(u^{7/2} - uv - v^{10/3} + 1, u^{7/2} + uv - v^{10/3} - 1\right)^T. \tag{53}$$

Then, we have

$$F'(u,v) = \begin{pmatrix} \frac{7}{2}u^{5/2} - v & -\frac{10}{3}v^{7/3} - u \\ \frac{7}{2}u^{5/2} + v & -\frac{10}{3}v^{7/3} + u \end{pmatrix},$$

$$F'(u,v)^{-1} = \frac{1}{(14/2)\,u^{7/2} + (20/3)\,v^{10/3}} \times \begin{pmatrix} -\frac{10}{3}v\frac{7}{3} + u & \frac{10}{3}v\frac{7}{3} + u \\ -\frac{7}{2}u\frac{5}{2} - v & \frac{7}{2}u\frac{5}{2} - v, \end{pmatrix},$$

$$F''(u,v) = \begin{pmatrix} \frac{35}{4}u^{3/2} & -1 \\ -1 & -\frac{70}{9}v^{4/3} \end{pmatrix}, \quad \begin{pmatrix} \frac{35}{4}u^{3/2} & 1 \\ 1 & -\frac{70}{9}v^{4/3}. \end{pmatrix}$$

$$F'''(u,v)(s,t)^3 = \begin{pmatrix} \frac{105}{8}u^{1/2} & \frac{280}{27}v^{1/3} \\ \frac{105}{8}u^{1/2} & \frac{280}{27}v^{1/3} \end{pmatrix} \begin{pmatrix} s^3 \\ t^3 \end{pmatrix}. \tag{54}$$

Now, we choose $x_0 = (u_0, v_0) = (1.5, 1.5)$ and $\Omega = \{x \mid \|x - x_0\| \le 1.5\}$. We take the max-norm in R^2 and the norm $\|A\| = \max\{|a_{11}|+|a_{12}|, |a_{21}|+|a_{22}|\}$ for $A = \left(\begin{smallmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{smallmatrix}\right)$. We define the norm of a bilinear operator B on R^2 by

$$\|B\| = \sup_{\|u\|=1} \max_i \sum_{j=1}^{2} \left| \sum_{k=1}^{2} b_i^{jk} u_k \right|, \tag{55}$$

where $u = (u_1, u_2)^T$ and $B = \left(\begin{smallmatrix} b_1^{11} & b_1^{12} \\ b_1^{21} & b_1^{22} \\ b_2^{11} & b_2^{12} \\ b_2^{21} & b_2^{22} \end{smallmatrix}\right)$.

Then, we get the following results: $\eta = \|F'(x_0)^{-1}F(x_0)\| = 0.09598\cdots$, $M = 9.20456\cdots$, $N = 10.7635\cdots$, and $p = 1/3$.

We get that the hypotheses of Theorem 6 are satisfied.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

The authors have made the same contribution. All authors read and approved the final paper.

Acknowledgments

This work is supported by the National Basic Research 973 Program of China (no. 2011JB105001), National Natural Science Foundation of China (Grant no. 11371320), Zhejiang Natural Science Foundation (Grant no. LZ14A010002), the Foundation of Science and Technology Department (Grant no. 2013C31084) of Zhejiang Province, and Scientific Research Fund of Zhejiang Provincial Education Department (nos. Y201431077 and Y201329420) and by the Grant FEKT-S-11-2-921 of Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic.

References

[1] J. A. Ezquerro, D. González, and M. A. Hernández, "A modification of the classic conditions of Newton-Kantorovich for Newton's method," *Mathematical and Computer Modelling*, vol. 57, no. 3-4, pp. 584–594, 2013.

[2] J. A. Ezquerro, D. González, and M. A. Hernández, "A variant of the Newton-Kantorovich theorem for nonlinear integral equations of mixed Hammerstein type," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9536–9546, 2012.

[3] I. K. Argyros, "On the solution of undetermined systems of nonlinear equations in Euclidean spaces," *Pure Mathematics and Applications*, vol. 4, no. 2, pp. 199–209, 1993.

[4] A. Holstad, "Numerical solution of nonlinear equations in chemical speciation calculations," *Computational Geosciences*, vol. 3, no. 3-4, pp. 229–257, 1999.

[5] K. Kowalski and K. Jankowski, "Towards complete solutions to systems of nonlinear equations of many-electron theories," *Physical Review Letters*, vol. 81, no. 6, pp. 1195–1198, 1998.

[6] B. M. Barbashov, V. V. Nesterenko, and A. M. Chervyakov, "General solutions of nonlinear equations in the geometric theory of the relativistic string," *Communications in Mathematical Physics*, vol. 84, no. 4, pp. 471–481, 1982.

[7] S. R. Bickham, S. A. Kiselev, and A. J. Sievers, "Stationary and moving intrinsic localized modes in one-dimensional monatomic lattices with cubic and quartic anharmonicity," *Physical Review B*, vol. 47, no. 21, pp. 14206–14211, 1993.

[8] G. Yuan and X. Lu, "A new backtracking inexact BFGS method for symmetric nonlinear equations," *Computers & Mathematics with Applications*, vol. 55, no. 1, pp. 116–129, 2008.

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

[10] R. Lin, Y. Zhao, Z. Smarda, Y. Khan, and Q. Wu, "Newton-Kantorovich and SMAle uniform type convergence theorem for a deformed Newton method in Banach spaces," *Abstract and Applied Analysis*, vol. 2013, Article ID 923898, 8 pages, 2013.

[11] L. Kantorovich, "On Newton method," *Trudy Matematicheskogo Instituta imeni VA Steklova*, vol. 28, pp. 104–144, 1949 (Russian).

[12] X. H. Wang and C. Li, "Convergence of Newton's method and uniqueness of the solution of equations in Banach spaces II," *Acta Mathematica Sinica*, vol. 19, no. 2, pp. 405–412, 2003.

[13] I. K. Argyros, "On the Newton-Kantorovich hypothesis for solving equations," *Journal of Computational and Applied Mathematics*, vol. 169, no. 2, pp. 315–332, 2004.

[14] I. K. Argyros, "On the comparison of a weak variant of the Newton-Kantorovich and Miranda theorems," *Journal of Computational and Applied Mathematics*, vol. 166, no. 2, pp. 585–589, 2004.

[15] M. A. Hernandez, "A modification of the classical Kantorovich conditions for Newton's method," *Journal of Computational and Applied Mathematics*, vol. 137, no. 1, pp. 201–205, 2001.

[16] S. Amat, S. Busquier, D. El kebir, and J. Molina, "A fast Chebyshev's method for quadratic equations," *Applied Mathematics and Computation*, vol. 148, no. 2, pp. 461–474, 2004.

[17] J. M. Gutierrez and M. A. Hernandez, "An acceleration of Newton's method: super-Halley method," *Applied Mathematics and Computation*, vol. 117, no. 2-3, pp. 223–239, 2001.

[18] J. A. Ezquerro and M. A. Hernández, "On the *R*-order of the Halley method," *Journal of Mathematical Analysis and Applications*, vol. 303, no. 2, pp. 591–601, 2005.

[19] J. A. Ezquerro and M. A. Hernández, "Fourth-order iterations for solving Hammerstein integral equations," *Applied Numerical Mathematics*, vol. 59, no. 6, pp. 1149–1158, 2009.

[20] J. A. Ezquerro and M. A. Hernández, "Halley's method for operators with unbounded second derivative," *Applied Numerical Mathematics*, vol. 57, no. 3, pp. 354–360, 2007.

[21] J. A. Ezquerro, D. Gonzalez, and M. A. Hernandez, "A variant of the Newton-Kantorovich theorem for nonlinear integral equations of mixed Hammerstein type," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9536–9546, 2012.

[22] R. Lin, Y. Zhao, Q. Wu, and J. Hu, "Convergence theorem for a family of new modified Halley's method in Banach space," *Journal of Applied Mathematics*, vol. 2014, Article ID 468694, 8 pages, 2014.

[23] I. K. Argyros, D. Chen, and Q. Qian, "The Jarratt method in Banach space setting," *Journal of Computational and Applied Mathematics*, vol. 51, no. 1, pp. 103–106, 1994.

[24] V. Kanwar, S. Kumar, and R. Behl, "Several new families of Jarratt's method for solving systems of nonlinear equations," *Applications and Applied Mathematics*, vol. 8, no. 2, pp. 701–716, 2013.

[25] I. K. Argyros, "A new convergence theorem for the Jarratt method in Banach space," *Computers & Mathematics with Applications*, vol. 36, no. 8, pp. 13–18, 1998.

[26] M. Ganesh and M. C. Joshi, "Numerical solvability of Hammerstein integral equations of mixed type," *IMA Journal of Numerical Analysis*, vol. 11, no. 1, pp. 21–31, 1991.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 514075, 5 pages
http://dx.doi.org/10.1155/2015/514075

Research Article

On a Derivative-Free Variant of King's Family with Memory

M. Sharifi,[1] S. Karimi Vanani,[1] F. Khaksar Haghani,[1] M. Arab,[1] and S. Shateyi[2]

[1] *Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran*
[2] *Department of Mathematics and Applied Mathematics, University of Venda, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 16 July 2014; Revised 29 August 2014; Accepted 1 September 2014

Academic Editor: Juan R. Torregrosa

The aim of this paper is to construct a method with memory according to King's family of methods without memory for nonlinear equations. It is proved that the proposed method possesses higher R-order of convergence using the same number of functional evaluations as King's family. Numerical experiments are given to illustrate the performance of the constructed scheme.

1. Introduction

Many problems arising in diverse disciplines of mathematical sciences can be described by a nonlinear equation of the following form (see, e.g., [1]):

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

where $f : D \subseteq \mathbb{R} \to \mathbb{R}$ is a sufficiently differentiable function in a neighborhood D of a simple zero α of (1). If we are interested in approximating the root α, we can do it by means of an iterative fixed-point method in the following form:

$$x_{k+1} = \psi(x_k), \quad k \geq 0, \tag{2}$$

provided that the starting point x_0 is given.

In this work, we are concerned with the fixed-point methods that generate sequences presumably convergent to the true solution of a given single smooth equation. These schemes can be divided into one-point and multipoint schemes. We remark that the one-point methods can possess high order by using higher derivatives of the function, which is expensive from a computational point of view. On the other hand, the multipoint methods are allowing the user not to waste information that had already been used. This approach provides the construction of efficient iterative root-finding methods [2].

In such circumstance, special attention is devoted to multipoint methods with memory that use already computed information to considerably increase convergence rate without additional computational costs. This would be the focus of this paper.

Traub in [2] proposed the following method with memory (TM):

$$w_k = x_k + \beta_k f(x_k), \quad \beta_k = -\frac{1}{f[x_k, x_{k-1}]}, \quad k = 0, 1, 2, \ldots,$$
$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \tag{3}$$

with the order of convergence $1 + \sqrt{2}$.

The iterative methods with memory can improve the order of convergence of the without memory method without any additional functional calculations, and this results in a higher computational efficiency index. We remark that it is assumed that an initial approximation x_0 close enough to the sought simple zero and β_0 are given for iterative methods of type (3).

Recently, authors in [3] designed an approach to make derivative-free families with low complexity out of optimal methods. In fact, they conjectured that every time that one applies the approximation of the derivative $f'(x_n) \approx f[x_n, w_n]$, with $w_n = x_n + \beta f(x_n)^l$, on an optimal order $2q$, we will need $l \geq q$ for preserving the order of convergence.

For instance, choosing the well-known optimal two-step family of King (KM) [4],

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \ldots,$$
$$x_{k+1} = y_k - \frac{f(y_k)}{f'(x_k)} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2) f(y_k)}, \quad \gamma \in \mathbb{R}, \tag{4}$$

and the conjecture of Cordero-Torregrosa, one may propose the following method (DKM):

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{FD}}, \quad k = 0, 1, 2, \ldots,$$
$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{FD}} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2) f(y_k)}, \quad \gamma \in \mathbb{R}, \tag{5}$$

wherein

$$\mathfrak{FD} = \frac{f(x_k) - f(w_k)}{x_k - w_k}, \quad w_k = x_k + \beta f(x_k)^2, \quad \beta \in \mathbb{R} \setminus \{0\}. \tag{6}$$

In this work, we propose a two-step method with memory possessing a high efficiency index according to the well-known family of King's methods (5).

Our inspiration and motivation for constructing a higher-order method are linked in a direct manner with the fundamental concept of numerical analysis that any numerical method should give as accurate as possible output results with minimal computational cost. To state the matter differently, it is necessary to pursue methods of higher computational efficiency.

For more background concerning this topic, one may refer to [5, 6].

The paper is organized as follows. In Section 2, the aim of this paper is presented by contributing an iterative method with memory based on (5) for solving nonlinear equations. The proposed scheme is an extension over (4) and has a simple structure with an increased computational efficiency. In Section 3, we compare the theoretical results by applying the definition of efficiency index and further supports are furnished whereas numerical reports are stated. Some concluding remarks will be drawn in Section 4 to end the paper.

2. A New Method with Memory

In this section, we propose the following iterative method with memory based on (5):

$$y_k = x_k - \frac{f(x_k)}{\mathfrak{FD}}, \quad w_k = x_k + \beta_k f(x_k)^2, \quad k = 0, 1, 2, \ldots,$$
$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{FD}} \frac{f(x_k) + \gamma f(y_k)}{f(x_k) + (\gamma - 2) f(y_k)}, \quad \gamma = -\frac{1}{2}, \tag{7}$$

wherein the self-accelerating parameter is β_k. The error equation of (5) is ($\gamma = -1/2$)

$$e_{k+1} = -c_2 \left(f'(\alpha)^2 \beta c_2 + c_3\right) e_k^4 + O\left(e_k^5\right), \tag{8}$$

where $c_j = (1/j!)(f^{(j)}(\alpha)/f'(\alpha))$. We now must find a way so as to vanish the asymptotic error constant $\eta = -c_2(f'(\alpha)^2 \beta c_2 + c_3)$.

Toward this goal, one can increase the R-order by considering the following substitution:

$$\beta = -\frac{c_3}{f'(\alpha)^2 c_2}. \tag{9}$$

Since the zero is not known, relation (9) cannot be used in its exact form and we must approximate it recursively. This builds a variant with memory for King's family by using

$$\beta_k \approx -\frac{\overline{c}_3}{\overline{f}'(\alpha)^2 \overline{c}_2}, \tag{10}$$

where $\overline{c}_j \approx c_j$. Now if we consider $N_3(t)$ to be Newton's interpolation polynomial of third degree set through four available approximations $x_k, x_{k-1}, y_{k-1}, w_{k-1}$ at the end of each cycle, we can propose the following new method with memory:

$$\beta_k = -\frac{N_3'''(x_k)}{3N_3'(x_k)^2 N_3''(x_k)},$$
$$y_k = x_k - \frac{f(x_k)}{\mathfrak{FD}}, \quad w_k = x_k + \beta_k f(x_k)^2, \quad k = 0, 1, 2, \ldots,$$
$$x_{k+1} = y_k - \frac{f(y_k)}{\mathfrak{FD}} \frac{f(x_k) - 1/2 f(y_k)}{f(x_k) - 5/2 f(y_k)}. \tag{11}$$

Note that, for example, we have the following formulation for the interpolating polynomial:

$$N_3'(x_k) = \left[\frac{d}{dt} N_3(t)\right]_{t = x_k}$$
$$= f[x_k, x_{k-1}] + f[x_k, x_{k-1}, y_{k-1}](x_k - x_{k-1})$$
$$+ f[x_k, x_{k-1}, y_{k-1}, w_{k-1}](x_k - x_{k-1})(x_k - y_{k-1}). \tag{12}$$

Acceleration in convergence for (11) is based on the use of a variation of one free nonzero parameter in each iterative step. This parameter is calculated using information from the current and previous iteration(s) so that the developed method may be regarded as method with memory according to Traub's classification [2].

We are at the time to write about the theoretical aspects of our proposed solver (11).

Theorem 1. *Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α. If an initial approximation x_0 is sufficiently close to α, then, the R-order of convergence of the two-step method (11) with memory is at least 4.23607.*

Proof. Let $\{x_k\}$ be a sequence of approximations generated by an iterative method. The error relations with the self-accelerating parameter $\beta = \beta_k$ for (11) are in what follows:

$$\widehat{e}_k = w_k - \alpha \sim c_{k,1} e_k, \tag{13}$$

$$\widetilde{e}_k = y_k - \alpha \sim c_{k,2} e_k^2, \tag{14}$$

$$e_{k+1} = x_{k+1} - \alpha \sim c_{k,4} e_k^4. \tag{15}$$

Using a symbolic computations, we attain that

$$-c_2 \left(f'(\alpha)^2 \beta c_2 + c_3 \right) \sim e_{k-1}. \tag{16}$$

Substituting the value of $-c_2(f'(\alpha)^2 \beta c_2 + c_3)$ from (16) in (15), one may obtain

$$e_{k+1} \sim c_{k,4} e_{k-1} e_k^4. \tag{17}$$

Note that in general we know that the error equation should read $e_{k+1} \sim A e_k^p$, where A and p are to be determined. Hence, one has $e_k \sim A e_{k-1}^p$, and subsequently

$$e_{k-1} \sim A^{-1/p} e_k^{1/p}. \tag{18}$$

Thus, it is easy to obtain

$$e_k^p \sim A^{-1/p} C e_k^{4+1/p}, \tag{19}$$

wherein C is a constant. This results in

$$p = 4 + \frac{1}{p}, \tag{20}$$

with two solutions $\{-0.236068, 4.23607\}$. Clearly the value for $p = 4.23607$ is acceptable and would be the convergence R-order of method (11) with memory. The proof is complete. □

The increase of R-order is attained without any (new) additional function calculations so that the novel method with memory possesses a high computational efficiency index. This technique is an extension over scheme (5) to increase the R-order from 4 to 4.23607.

The accelerating method (11) is new, simple, and useful, providing considerable improvement of convergence rate without any additional function evaluations in contrast to the optimal two-step methods without memory.

We also remark that an alternative form of our proposed method with memory could be deduced using backward finite difference formula at the beginning of the first substep and a minor modification in the accelerators; that is to say, we have the following alternative method with memory possessing 4.23607 as its R-order (APM) as well:

$$\begin{aligned} \beta_k &= \frac{N_3'''(x_k)}{3N_3'(x_k)^2 N_3''(x_k)}, \\ y_k &= x_k - \frac{f(x_k)}{\mathfrak{FD}}, \qquad w_k = x_k - \beta_k f(x_k)^2, \\ x_{k+1} &= y_k - \frac{f(y_k)}{\mathfrak{FD}} \frac{f(x_k) - 1/2 f(y_k)}{f(x_k) - 5/2 f(y_k)}. \end{aligned} \tag{21}$$

Theorem 2. *Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α. If an initial approximation x_0 is sufficiently close to α, then, the R-order of convergence of the two-step method (21) with memory is at least 4.23607.*

Proof. The proof of this theorem is similar to Theorem 1. It is hence omitted. □

3. Numerical Computations

Computational efficiency of different iterative methods with and without memory can be measured in a prosperous manner by applying the definition of efficiency index. For an iterative method with convergence (R-)order r that requires θ functional evaluations, the efficiency index (also named computational efficiency) is calculated by Ostrowski-Traub's formula [2]:

$$E = r^{1/\theta}. \tag{22}$$

According to this, we find

$$\begin{aligned} E(\mathrm{SM}) \approx 1.4142 < E(3) \approx 1.5737 = E(4) \approx 1.5874 \\ = E(5) \approx 1.5874 < E(11) \approx 1.6180, \end{aligned} \tag{23}$$

where SM is the quadratically convergent method of Steffensen without memory [7].

It should be remarked that Džunić in [8] designed an efficient one-step Steffensen-type method with memory possessing $(1/2)(3+\sqrt{17})$ R-order of convergence as follows:

$$\begin{aligned} w_k &= x_k + \beta_k f(x_k), \\ \beta_k &= -\frac{1}{N_2'(x_k)}, \qquad p_k = -\frac{N_3''(w_k)}{2N_3'(w_k)}, \\ x_{k+1} &= x_k - \frac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \end{aligned} \tag{24}$$

Table 1: Results of comparisons for Example 3 and to find $\alpha = 2$.

Methods	$\lvert f(x_1)\rvert$	$\lvert f(x_2)\rvert$	$\lvert f(x_3)\rvert$	$\lvert f(x_4)\rvert$	coc
KM	18.577	64890.	7.2226×10^{10}	3.2493×10^{9}	—
OM	4.7484	0.0023129	1.3928×10^{-16}	1.8313×10^{-69}	4.00000
DKM	0.53362	5.3207×10^{-7}	5.2711×10^{-31}	5.0774×10^{-127}	4.00000
PM	0.53362	1.9202×10^{-6}	3.6106×10^{-30}	1.6392×10^{-130}	4.22928

Table 2: Results of comparisons for Example 4.

Methods	$\lvert f(x_1)\rvert$	$\lvert f(x_2)\rvert$	$\lvert f(x_3)\rvert$	$\lvert f(x_4)\rvert$	coc
KM	2.0873	0.0095650	7.7971×10^{-12}	3.4597×10^{-48}	4.00000
OM	0.81344	0.0010884	1.5476×10^{-15}	6.3280×10^{-63}	4.00000
DKM	2.1909	0.013379	2.9909×10^{-11}	7.5008×10^{-46}	4.00000
PM	2.1909	0.0011772	7.0556×10^{-16}	8.4197×10^{-68}	4.23539
APM	1.9861	0.00089226	2.3251×10^{-16}	7.5243×10^{-70}	4.23526

and Cordero et al. in [9] presented a two-step biparametric Steffensen-type iterative method with memory possessing seventh R-order of convergence:

$$\begin{aligned} w_k &= x_k + \beta_k f(x_k), \qquad \beta_k = -\frac{1}{N_3'(x_k)}, \\ p_k &= -\frac{N_4''(w_k)}{2N_4'(w_k)}, \\ y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \\ x_{k+1} &= y_k - \frac{f(y_k)}{f[x_k, y_k] + f[w_k, x_k, y_k](y_k - x_k)}. \end{aligned} \tag{25}$$

Note that our main aim was to develop King's family in terms of efficiencies index and was not to achieve the highest possible efficiency index.

Although these methods possess higher computational efficiency indices than our proposed method (11), we exclude them from numerical comparisons since our method is not a Steffensen-type method and it is a Newton-type method with memory. For more refer to [10].

Now, we apply and compare the behavior of different methods for finding the simple zeros of some different nonlinear test functions in the programming package Mathematica [11] using multiple precision arithmetic to clearly reveal the high R-order of PM and APM. We compare methods with the same number of functional evaluations per cycle.

We notice that, by applying any root solver with local convergence, a special attention must be paid to the choice of initial approximations. If initial values are sufficiently close to the sought roots, then the expected (theoretical) convergence speed is obtainable in practice; otherwise, the iterative methods show slower convergence, especially at the beginning of the iterative process.

In this section, the computational order of convergence (coc) has been computed by

$$\text{coc} = \frac{\ln\left|f(x_k)/f(x_{k-1})\right|}{\ln\left|f(x_{k-1})/f(x_{k-2})\right|}. \tag{26}$$

The calculated value coc estimates the theoretical order of convergence well when pathological behavior of the iterative method (i.e., slow convergence at the beginning of the implemented iterative method, oscillating behavior of approximations, etc.) does not exist.

Here the results of comparisons for the test functions are given by applying 1000 fixed floating point arithmetic using the stop termination $\lvert f(x_k)\rvert \le 10^{-100}$.

Example 3. We consider the following nonlinear test function in the interval $D = [1.5, 2.5]$:

$$f(x) = (x - 2\tan(x))\left(x^3 - 8\right), \tag{27}$$

using the initial approximation $x_0 = 1.7$. The results are provided in Table 1.

In this section, we have used $\beta_0 = 0.0001$ whenever required. Furthermore, for DKM we considered $\gamma = -1/2$.

Example 4. We compare the behavior of different methods for finding the complex solution of the following nonlinear equation:

$$g(x) = (-1 + 2I) + \frac{1}{x} + x + \sin(x), \tag{28}$$

using the initial approximation $x_0 = 1 - 3I$ where $\alpha = 0.28860\cdots - 1.24220\cdots I$. The results for this test are given in Table 2.

It is evident from Tables 1 and 2 that approximations to the roots possess great accuracy when the proposed method with memory is applied. Results of the fourth iterate in Tables 1 and 2 are given only for demonstration of convergence speed of the tested methods and in most cases they are not required for practical problems at present.

We also incorporated and applied the developed methods with memory (11) and (21) for different test examples and obtained results with the same behavior as above. Hence, we could mention that the theoretical results are upheld by numerical experiments and thus the new method is good with a high computational efficiency index.

4. Summary

In this paper, we have proposed a new two-step Steffensen-type iterative method with memory for solving nonlinear scalar equations. Using one self-correcting parameter calculated by Newton interpolatory polynomial, the R-order of convergence of the constructed method was increased from 4 to 4.23607 without any additional calculations.

The new method was compared in performance and computational efficiency with some existing methods by numerical examples. We have observed that the computational efficiency index of the presented method with memory is better than those of other existing two-step King-type methods.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

The authors have made the same contribution. All authors read and approved the final paper.

References

[1] A. S. Al-Fhaid, S. Shateyi, M. Z. Ullah, and F. Soleymani, "A matrix iteration for finding Drazin inverse with ninth-order convergence," *Abstract and Applied Analysis*, vol. 2014, Article ID 137486, 7 pages, 2014.

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

[3] A. Cordero and J. R. Torregrosa, "Low-complexity root-finding iteration functions with no derivatives of any order of convergence," *Journal of Computational and Applied Mathematics*, 2014.

[4] R. F. King, "A family of fourth order methods for nonlinear equations," *SIAM Journal on Numerical Analysis*, vol. 10, pp. 876–879, 1973.

[5] N. Huang and C. Ma, "Convergence analysis and numerical study of a fixed-point iterative method for solving systems of nonlinear equations," *The Scientific World Journal*, vol. 2014, Article ID 789459, 10 pages, 2014.

[6] J. P. Jaiswal, "Some class of third- and fourth-order iterative methods for solving nonlinear equations," *Journal of Applied Mathematics*, vol. 2014, Article ID 817656, 17 pages, 2014.

[7] J. F. Steffensen, "Remarks on iteration," *Skandinavisk Aktuarietidskrift*, vol. 16, pp. 64–72, 1933.

[8] J. Džunić, "On efficient two-parameter methods for solving nonlinear equations," *Numerical Algorithms*, vol. 63, no. 3, pp. 549–569, 2013.

[9] A. Cordero, T. Lotfi, P. Bakhtiari, and J. R. Torregrosa, "An efficient two-parametric family with memory for nonlinear equations," *Numerical Algorithms*, 2014.

[10] X. Wang and T. Zhang, "A new family of Newton-type iterative methods with and without memory for solving nonlinear equations," *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.

[11] S. Wagon, *Mathematica in Action*, Springer, New York, NY, USA, 3rd edition, 2010.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 943293, 10 pages
http://dx.doi.org/10.1155/2014/943293

Research Article

Multistage Spectral Relaxation Method for Solving the Hyperchaotic Complex Systems

Hassan Saberi Nik[1] and Paulo Rebelo[2]

[1] *Department of Mathematics, Islamic Azad University, Mashhad Branch, Mashhad, Iran*
[2] *Departamento de Matemática, Universidade da Beira Interior, 6201-001 Covilhã, Portugal*

Correspondence should be addressed to Hassan Saberi Nik; saberi_hssn@yahoo.com

Received 4 August 2014; Accepted 12 September 2014; Published 16 October 2014

Academic Editor: Fazlollah Soleymani

We present a pseudospectral method application for solving the hyperchaotic complex systems. The proposed method, called the multistage spectral relaxation method (MSRM) is based on a technique of extending Gauss-Seidel type relaxation ideas to systems of nonlinear differential equations and using the Chebyshev pseudospectral methods to solve the resulting system on a sequence of multiple intervals. In this new application, the MSRM is used to solve famous hyperchaotic complex systems such as hyperchaotic complex Lorenz system and the complex permanent magnet synchronous motor. We compare this approach to the Runge-Kutta based ode45 solver to show that the MSRM gives accurate results.

1. Introduction

Chaos theory studies the behaviour of dynamical systems that are highly sensitive to initial conditions and have complex and highly unpredictable profiles [1, 2]. Chaotic systems can be observed in a wide variety of applications. In 1982, the complex Lorenz equations were proposed by Fowler et al. [3], which extended nonlinear systems into complex space. After that, some research works in this field have been achieved [4–9]. With in-depth study of complex nonlinear systems, a variety of physical phenomena could be described by the chaotic or hyperchaotic complex systems, for instance, the detuned laser systems and the amplitudes of electromagnetic fields.

The nature of complex chaotic systems precludes the possibility of obtaining closed form analytical solutions of the underlying governing equations. Thus, approximate-analytical methods, which are implemented on a sequence of multiple intervals to increase their radius of convergence, are often used to solve IVPs modelling chaotic systems. Examples of multistage methods that have been developed recently to solve IVPs for chaotic and nonchaotic systems include the multistage homotopy analysis method [10], piecewise homotopy perturbation methods [11, 12], multistage variational iteration method [13], and multistage differential transformation method [14]. Other multistage methods which use numerical integration techniques have also been proposed such as the piecewise spectral homotopy analysis method [15–17] which uses a spectral collocation method to perform the integration process. Accurate solutions of highly chaotic and hyperchaotic systems require resolution over many small intervals. Thus, seeking analytical solutions over the numerous intervals may be impractical or computationally expensive if the solution is sought over very long intervals.

In this paper, we propose a piecewise or multistage spectral relaxation method (MSRM) for solving the hyperchaotic complex systems as an accurate and robust alternative to recent multistage methods. The proposed MSRM was developed using the Gauss-Seidel idea of decoupling systems of equations and using Chebyshev pseudospectral methods to solve the resulting decoupled system on a sequence of multiple intervals. The spectral relaxation method (SRM) was recently proposed in [18, 19].

The rest of the paper is organized as follows. In Section 2, we give a brief description of the proposed MSRM algorithm.

In Section 3, we present the numerical implementation of the MSRM on two examples of hyperchaotic complex systems. Finally, the conclusion is given in Section 4.

2. Multistage Spectral Relaxation Method

In this section, we give a brief description of the numerical method of solution used to solve the nonlinear hyperchaotic complex. We employ the multistage spectral relaxation method (MSRM) proposed in [19]. The MSRM algorithm is based on a Gauss-Seidel type of relaxation that decouples and linearises the system and the use of spectral collocation method to solve the linearised equations in a sequential manner. For compactness, we express the system of m nonlinear first order differential equations in the form

$$\begin{aligned}\dot{x}_r(t) &= \sum_{k=1}^{m}\alpha_{r,k}x_k(t)\\ &\quad + f_r\left[x_1(t), x_2(t), \ldots, x_{r-1}(t),\right.\\ &\qquad \left. x_{r+1}(t), \ldots, x_m(t)\right],\end{aligned} \tag{1}$$

subject to the initial conditions

$$x_r(0) = x_r^*, \quad r = 1, 2, \ldots, m, \tag{2}$$

where x_r are the unknown variables and x_r^* are the corresponding initial conditions, $\alpha_{r,k}$ are known constant input parameters and f_r is the nonlinear component of the rth equation and the dot denotes differentiation with respect to time t.

The scheme computes the solution of (1) in a sequence of equal subintervals that makes the entire interval. We define the interval of integration as $\Omega = [0, T]$ and divide it into a sequence of nonoverlapping subintervals $\Omega_i = [t_{i-1}, t_i]$ $(i = 1, 2, 3, \ldots, f)$, where $t_0 = 0$ and $t_f = T$. We denote the solution of (1) in the first subinterval $[t_0, t_1]$ as $x_r^1(t)$ and the solutions in the subsequent subintervals $[t_{i-1}, t_i]$ $(i = 2, 3, \ldots, f)$ as $x_r^i(t)$. For obtaining the solution in the first interval $[t_0, t_1]$, (2) is used as the initial condition. By using the continuity condition between neighbouring subintervals the obtained solution in the interval $[t_0, t_1]$ is used to obtain the initial condition for the next subinterval $[t_1, t_2]$. This is applied over the f successive subintervals; that is, the obtained solution for each subinterval $[t_{i-1}, t_i]$ is used to obtain the initial condition for the next subinterval $[t_i, t_{i+1}]$ $(i = 1, 2, \ldots, f-1)$. Thus, in each interval $[t_{i-1}, t_i]$ we must solve

$$\begin{aligned}\dot{x}_r^i &= \alpha_{r,r}x_r^i + (1-\delta_{rs})\sum_{k=1}^{m}\alpha_{r,k}x_k^i\\ &\quad + f_r\left[x_1^i, \ldots, x_{r-1}^i, x_{r+1}^i, \ldots, x_n^i\right],\end{aligned} \tag{3}$$

subject to

$$x_r^i(t_{i-1}) = x_r^{i-1}(t_{i-1}), \tag{4}$$

where δ_{rs} is the Kronecker delta. As mentioned earlier, the main idea behind the MSRM scheme is decoupling the system of nonlinear IVPs using the Gauss-Seidel idea of decoupling systems of algebraic equations. The proposed MSRM iteration scheme for the solution in the interval $\Omega_i = [t_{i-1}, t_i]$ is given as

$$\begin{aligned}\dot{x}_{1,s+1}^i - \alpha_{1,1}x_{1,s+1}^i &= \alpha_{1,2}x_{2,s}^i + \alpha_{1,3}x_{3,s}^i\\ &\quad + \cdots + \alpha_{1,n}x_{n,s}^i + f_1\left[x_{1,s}^i, \ldots, x_{n,s}^i\right],\\ \dot{x}_{2,s+1}^i - \alpha_{2,2}x_{2,s+1}^i &= \alpha_{2,1}x_{1,s+1}^i + \alpha_{2,3}x_{3,s}^i\\ &\quad + \cdots + \alpha_{2,n}x_{n,s}^i\\ &\quad + f_2\left[x_{1,s+1}^i, x_{2,s}^i, \ldots, x_{n,s}^i\right],\\ &\vdots\\ \dot{x}_{m,s+1}^i - \alpha_{m,m}x_{m,s+1}^i &= \alpha_{m,1}x_{1,s+1}^i + \cdots + \alpha_{m,m-1}x_{m-1,s+1}^i\\ &\quad + f_m\left[x_{1,s+1}^i, \ldots, x_{m-1,s+1}^i, x_{m,s}^i\right],\end{aligned} \tag{5}$$

subject to the initial conditions

$$x_{r,s+1}^i(t_{i-1}) = x_r^{i-1}(t_{i-1}), \quad r = 1, 2, \ldots, m, \tag{6}$$

where $x_{r,s}$ is the estimate of the solution after s iterations. A suitable initial guess to start the iteration scheme (5) is one that satisfies the initial condition (6). A convenient choice of initial guess that was found to work in the numerical experiments considered in this work is

$$x_{r,0}^i(t) = \begin{cases} x_r^* & \text{if } i = 1,\\ x_r^{i-1}(t_{i-1}) & \text{if } 2 \le i \le f.\end{cases} \tag{7}$$

The Chebyshev spectral method is used to solve (5) on each interval $[t_{i-1}, t_i]$. First, the region $[t_{i-1}, t_i]$ is transformed to the interval $[-1, 1]$ on which the spectral method is defined by using the linear transformation,

$$t = \frac{(t_i - t_{i-1})\tau}{2} + \frac{(t_i + t_{i-1})}{2}, \tag{8}$$

in each interval $[t_{i-1}, t_i]$ for $i = 1, \ldots, f$. We then discretize the interval $[t_{i-1}, t_i]$ using the Chebyshev-Gauss-Lobatto collocation points [20]:

$$\tau_j^i = \cos\left(\frac{\pi j}{N}\right), \quad j = 1, 2, \ldots, N, \tag{9}$$

which are the extrema of the Nth order Chebyshev polynomial:

$$T_N(\tau) = \cos\left(N\cos^{-1}\tau\right). \tag{10}$$

The Chebyshev spectral collocation method is based on the idea of introducing a differentiation matrix D which is used to approximate the derivatives of the unknown variables $x_{r,s+1}^i(t)$ at the collocation points as the matrix vector product

$$\left.\frac{dx_{r,s+1}^i}{dt}\right|_{t=t_j} = \sum_{k=0}^{N}\mathbf{D}_{jk}x_{r,s+1}^i = \mathbf{D}\mathbf{X}_{r,s+1}^i, \quad j = 1, 2, \ldots, N, \tag{11}$$

Table 1: Numerical comparison between MSRM and ode45 for the hyperchaotic complex Lorenz system.

t	$x_1(t)$		$x_2(t)$		$x_3(t)$	
	MSRM	ode45	MSRM	ode45	MSRM	ode45
2	−2.91138	−2.91138	21.73155	21.73155	−3.24491	−3.24491
4	−3.63001	−3.63001	6.52144	6.52144	−6.30884	−6.30884
6	2.80571	2.80571	−2.77638	−2.77638	−2.37099	−2.37099
8	0.01134	0.01134	2.09585	2.09585	−0.14880	−0.14880
10	−0.80219	−0.80219	16.48559	16.48560	−0.06690	−0.06690

where $\mathbf{D} = 2D/(t_i - t_{i-1})$ and $\mathbf{X}^i_{r,s+1} = [x^i_{r,s+1}(\tau^i_0), x^i_{r,s+1}(\tau^i_1), \ldots, x^i_{r,s+1}(\tau^i_N)]$ are the vector functions at the collocation points τ^i_j.

Applying the Chebyshev spectral collocation method in (5) gives

$$\mathbf{A}_r \mathbf{X}^i_{r,s+1} = \mathbf{B}^i_r, \qquad \mathbf{X}^i_{r,s+1}\left(\tau^{i-1}_N\right) = \mathbf{X}^{i-1}_r\left(\tau^{i-1}_N\right), \quad r = 1, 2, \ldots, m, \tag{12}$$

with

$$\begin{aligned}
\mathbf{A}_r &= \mathbf{D} - \alpha_{r,r}\mathbf{I},\\
\mathbf{B}^i_1 &= \alpha_{1,2}\mathbf{X}^i_{2,s}\\
&\quad + \cdots + \alpha_{1,n}\mathbf{X}^i_{n,s} + f_1\left[\mathbf{X}^i_{1,s}, \ldots, \mathbf{X}^i_{m,s}\right],\\
\mathbf{B}^i_2 &= \alpha_{2,1}\mathbf{X}^i_{1,s+1} + \alpha_{2,3}\mathbf{X}^i_{3,s}\\
&\quad + \cdots + \alpha_{2,m}\mathbf{X}^i_{m,s} + f_2\left[\mathbf{X}^i_{1,s+1}, \mathbf{X}^i_{2,s}, \ldots, \mathbf{X}^i_{m,s}\right],\\
&\vdots\\
\mathbf{B}^i_m &= \alpha_{m,1}\mathbf{X}^i_{1,s+1} + \alpha_{m,2}\mathbf{X}^i_{2,s+1} + \cdots + \alpha_{m,m-1}\mathbf{X}^i_{m-1,s+1}\\
&\quad + f_m\left[\mathbf{X}^i_{1,s+1}, \ldots, \mathbf{X}^i_{m-1,s+1}, \mathbf{X}^i_{m,s}\right],
\end{aligned} \tag{13}$$

where $\mathbf{I}$ is an identity matrix of order $N + 1$. Thus, starting from the initial approximation (7), the recurrence formula

$$\mathbf{X}^i_{r,s+1} = \mathbf{A}^{-1}_r \mathbf{B}^i_r, \quad r = 1, 2, \ldots, m \tag{14}$$

can be used to obtain the solution $x^i_r(t)$ in the interval $[t_{i-1}, t_i]$. The solution approximating $x_r(t)$ in the entire interval $[t_0, t_F]$ is given by

$$x_r(t) = \begin{cases} x^1_r(t), & t \in [t_0, t_1]\\ x^2_r(t), & t \in [t_1, t_2]\\ \vdots & \\ x^F_r(t), & t \in \left[t_{f-1}, t_f\right]. \end{cases} \tag{15}$$

3. Numerical Examples

In this section, we consider two examples which demonstrate the efficiency and accuracy of the proposed method. In particular, we use the MSRM algorithm as an appropriate tool for solving nonlinear IVPs; we apply the method to two complex nonlinear chaotic systems.

Example 1. The hyperchaotic complex Lorenz system can be described as

$$\begin{aligned}
\dot{z}_1 &= a_1(z_2 - z_1) + jz_4,\\
\dot{z}_2 &= a_2 z_1 - z_2 - z_1 z_3 + jz_4,\\
\dot{z}_3 &= \frac{1}{2}\left(z_1\overline{z}_2 + \overline{z}_1 z_2\right) - a_3 z_3,\\
\dot{z}_4 &= \frac{1}{2}\left(z_1\overline{z}_2 + \overline{z}_1 z_2\right) - a_4 z_4,
\end{aligned} \tag{16}$$

where $z_1 = x_1 + jx_2$, $z_2 = x_3 + jx_4$, $z_3 = x_5$, $z_4 = x_6$, $j = \sqrt{-1}$, $\overline{z}_1$ and $\overline{z}_2$ are the conjugates of z_1 and z_2. When the parameters are chosen as $a_1 = 15$, $a_2 = 36$, $a_3 = 4.5$, and $a_4 = 12$, the system (16) is hyperchaotic [21].

Replacing the complex variables in system (16) with real and imaginary number variables, one can get an equivalent system as follows:

$$\begin{aligned}
\dot{x}_1 &= a_1(x_3 - x_1),\\
\dot{x}_2 &= a_1(x_4 - x_2) + x_6,\\
\dot{x}_3 &= a_2 x_1 - x_3 - x_1 x_5,\\
\dot{x}_4 &= a_2 x_2 - x_4 - x_2 x_5 + x_6,\\
\dot{x}_5 &= x_1 x_3 + x_2 x_4 - a_3 x_5,\\
\dot{x}_6 &= x_1 x_3 + x_2 x_4 - a_4 x_6.
\end{aligned} \tag{17}$$

For (17), the parameters $\alpha_{r,k}$ and f_r are defined as

$$\begin{aligned}
&\alpha_{1,1} = -a_1, \quad \alpha_{1,3} = a_1, \quad \alpha_{2,2} = -a_1,\\
&\alpha_{2,4} = a_1, \quad \alpha_{2,6} = 1,\\
&\alpha_{3,1} = a_2, \quad \alpha_{3,3} = -1, \quad \alpha_{4,2} = a_2,\\
&\alpha_{4,4} = -1, \quad \alpha_{4,6} = 1,\\
&\alpha_{5,5} = -a_3, \quad \alpha_{6,6} = -a_4, \quad f_3 = -x_1 x_5,\\
&f_4 = -x_2 x_5, \quad f_5 = f_6 = x_1 x_3 + x_2 x_4,
\end{aligned} \tag{18}$$

with all other $\alpha_{r,k}$ and $f_r = 0$ for $r, k = 1, 2, \ldots, 6$.

Through numerical experimentation, it was determined that $N = 6$ collocation points and 5 iterations of the MSRM scheme at each interval were sufficient to give accurate results in each $[t_{i-1}, t_i]$ interval. Tables 1 and 2 show a comparison of the solutions of the hyperchaotic complex Lorenz system

Table 2: Numerical comparison between MSRM and ode45 for the hyperchaotic complex Lorenz system.

t	$x_4(t)$		$x_5(t)$		$x_6(t)$	
	MSRM	ode45	MSRM	ode45	MSRM	ode45
2	23.96851	23.96851	44.32071	44.32071	26.54682	26.54682
4	11.30830	11.30830	14.68007	14.68007	3.25221	3.25221
6	4.65208	4.65208	39.34559	39.34559	12.99055	12.99055
8	−4.99685	−4.99685	33.79560	33.79560	8.02232	8.02232
10	1.98179	1.98179	50.59739	50.59740	24.48234	24.48234

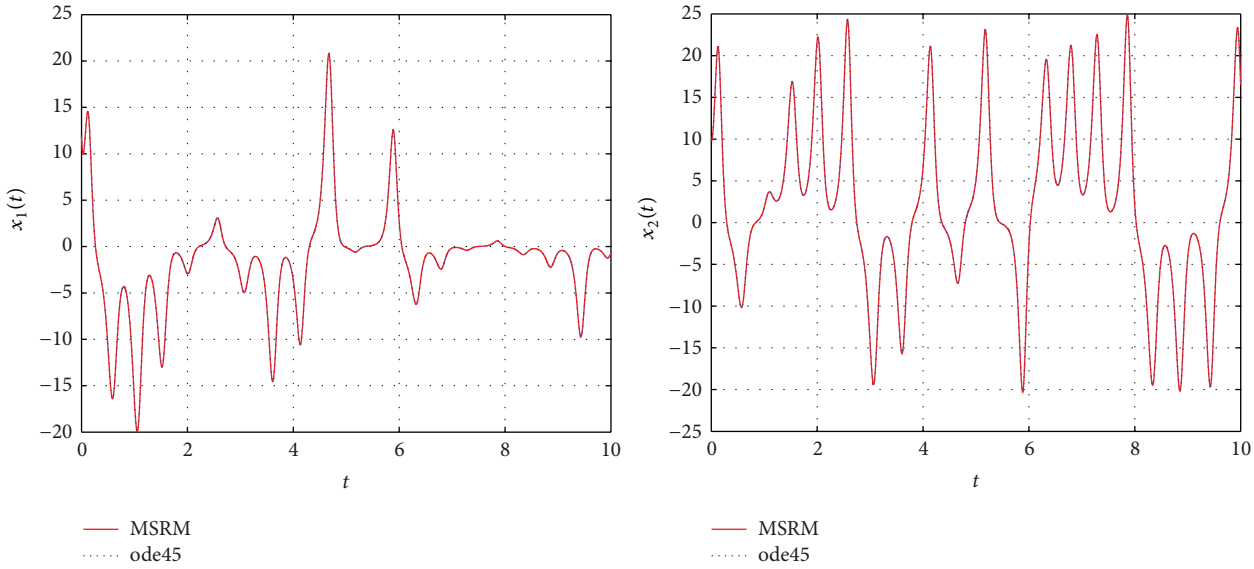


Figure 1: Comparison between the MSRM and ode45 results for the hyperchaotic complex Lorenz system.

computed by the MSRM and ode45. In Figures 1, 2, and 3, the MSRM graphical results are also compared with ode45 and good agreement is observed. The MRSM phase portraits in Figures 4 and 5 were also found to be exactly the same as those computed using ode45. This shows that the proposed MSRM is a valid tool for solving the hyperchaotic complex Lorenz system.

Example 2. State equations of a permanent magnet synchronous motor system in a field-oriented rotor can be described as follows [22, 23]:

$$\begin{aligned}
\frac{di_d}{dt} &= \frac{-R_1 i_d + \omega L_q i_q + u_d}{L_d},\\
\frac{di_q}{dt} &= \frac{R_1 i_q + \omega L_d i_q + u_q - \omega \Psi_r}{L_q},\\
\frac{d\omega}{dt} &= \frac{n_q \Psi_r i_d + n_p \left(L_d - L_q\right) i_d i_q - T_L - \beta\omega}{J},
\end{aligned} \tag{19}$$

where i_d, i_q, and ω are the state variables which represent currents and motor angular frequency, respectively; u_d and u_q are the direct-axis stator and quadrature-axis stator voltage components, respectively; J is the polar moment of inertia; T_L is the external load torque; β is the viscous damping coefficient; R_1 is the stator winding resistance; L_d and L_q are the direct-axis stator inductors and quadrature-axis stator inductors, respectively; Ψ_r is the permanent magnet flux; and n_p is the number of pole-pairs; the parameters L_d, L_q, J, T_L, R_1, Ψ_r, β are all positive.

When the air gap is even, and the motor has no load or power outage, the dimensionless equations of a permanent magnet synchronous motor system can be depicted as

$$\begin{aligned}
\dot{z}_1 &= a\left(z_2 - z_1\right),\\
\dot{z}_2 &= b z_1 - z_2 - z_1 z_3,\\
\dot{z}_3 &= z_1 z_2 - z_3,
\end{aligned} \tag{20}$$

where a, b are both positive parameters. If the current in the system (19) is plural and the variables z_1, z_2 in the system (20) are complex numbers, by changing cross coupled terms z_1 and z_2 to conjugate form, Wang and Zhang got a complex

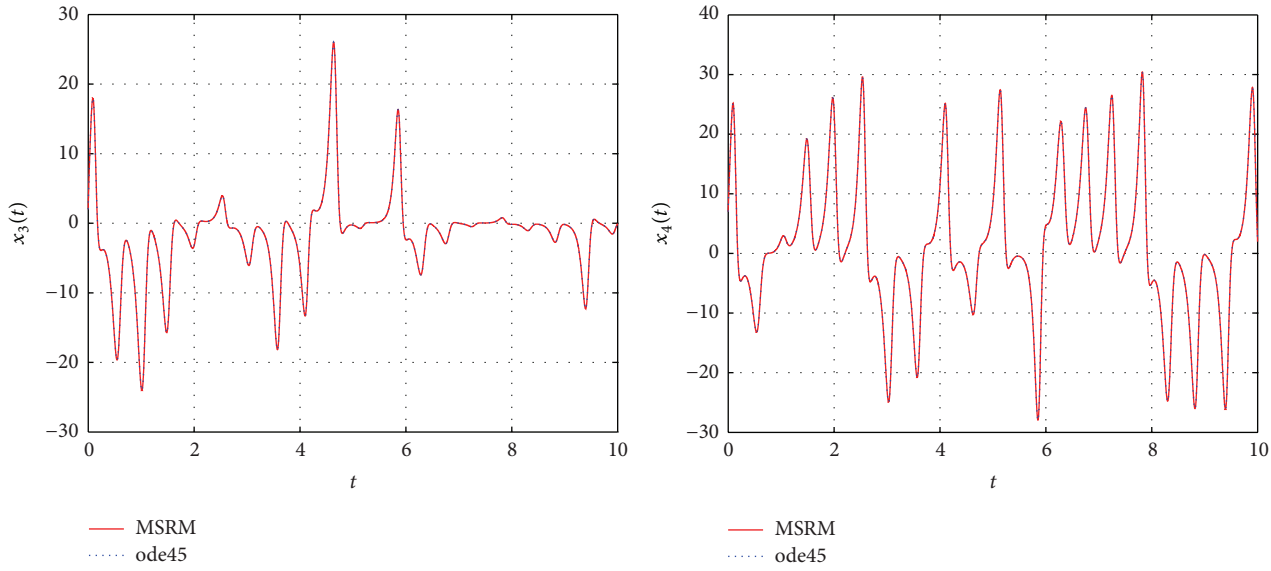


Figure 2: Comparison between the MSRM and ode45 results for the hyperchaotic complex Lorenz system.

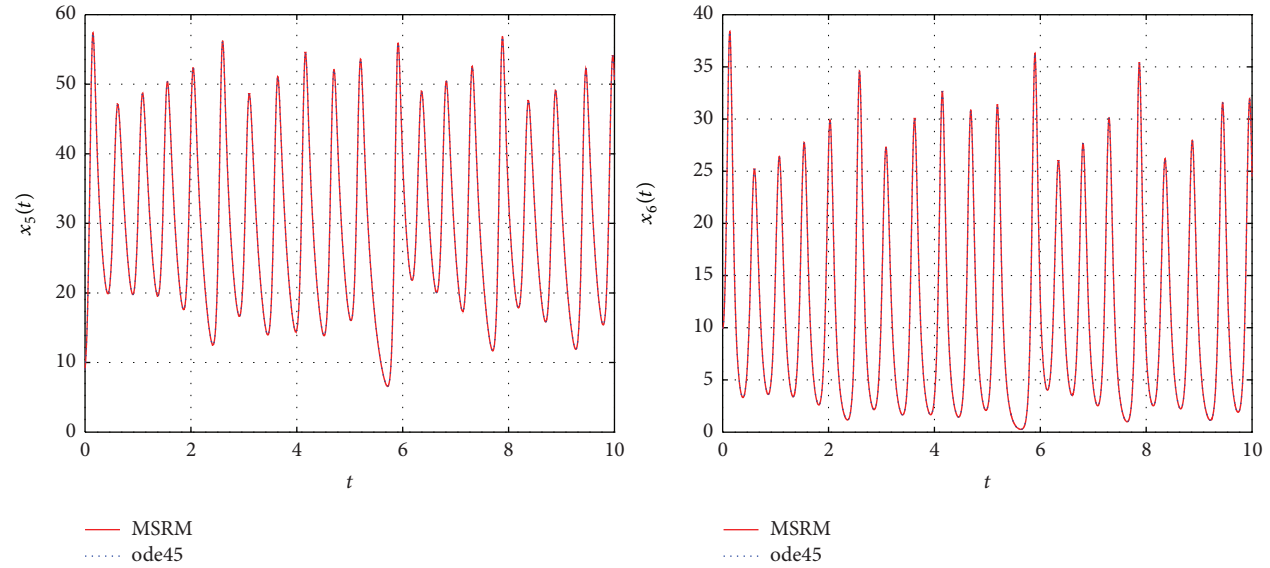


Figure 3: Comparison between the MSRM and ode45 results for the hyperchaotic complex Lorenz system.

permanent magnet synchronous motor system as follows [24]:

$$\begin{aligned}\dot{z}_1 &= a\left(z_2 - z_1\right),\\ \dot{z}_2 &= bz_1 - z_2 - z_1 z_3,\\ \dot{z}_3 &= \frac{1}{2}\left(z_1\overline{z}_2 + \overline{z}_1 z_2\right) - z_3,\end{aligned} \tag{21}$$

where $z_1 = x_1 + jx_2$, $z_2 = x_3 + jx_4$, $z_3 = x_5$, $j = \sqrt{-1}$, $\overline{z}_1$ and $\overline{z}_2$ are the conjugates of z_1 and z_2. Replacing the complex variables in system (21) with real and imaginary number variables, Wang and Zhang got an equivalent system as follows (see [24]):

$$\begin{aligned}\dot{x}_1 &= a\left(x_3 - x_1\right),\\ \dot{x}_2 &= a\left(x_4 - x_2\right),\\ \dot{x}_3 &= bx_1 - x_3 - x_1 x_5,\\ \dot{x}_4 &= bx_2 - x_4 - x_2 x_5,\\ \dot{x}_5 &= x_1 x_3 + x_2 x_4 - x_5,\end{aligned} \tag{22}$$

where a, b are positive parameters determining the chaotic behaviors and bifurcations of system (22). When the parameters satisfy $1 \le a \le 11$, $10 \le b \le 20$, there is one positive

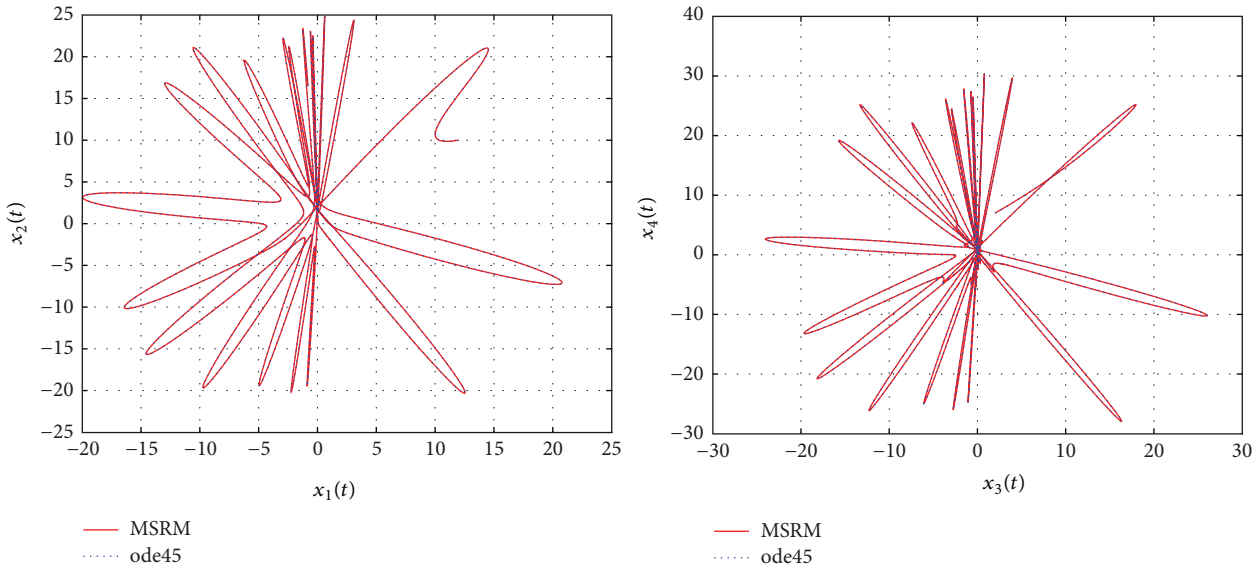


Figure 4: Phase portraits of the hyperchaotic complex Lorenz system.

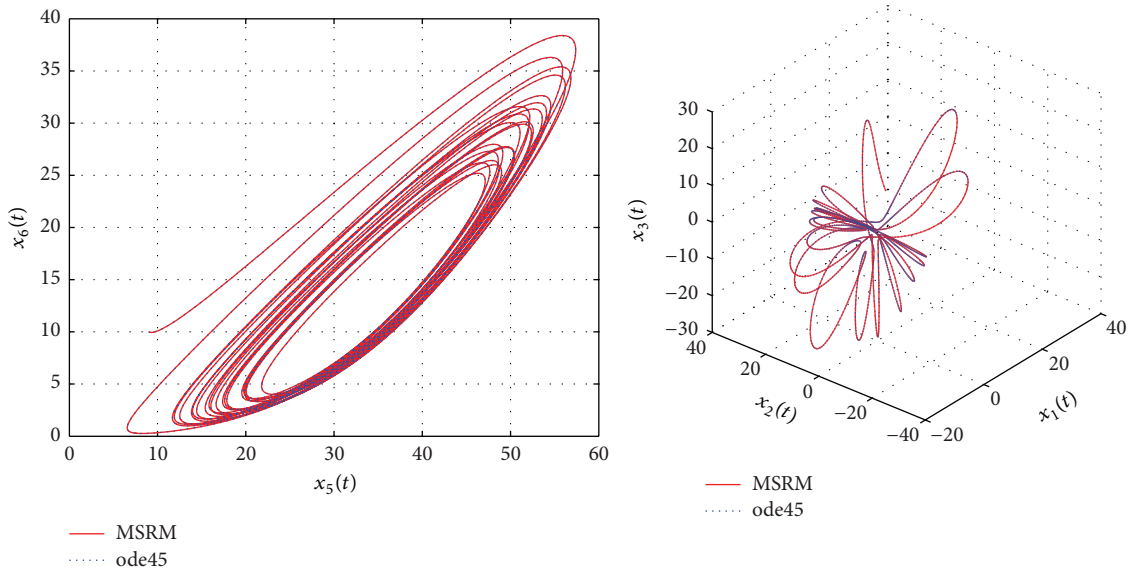


Figure 5: Phase portraits of the hyperchaotic complex Lorenz system.

Lyapunov exponent, two Lyapunov exponents of zero, and two negative Lyapunov exponents for system (22), which means system (22) is chaotic [24]. The values of parameters and initial values are $a = 11$, $b = 20$, and $x_1(0) = 1$, $x_2(0) = 2$, $x_3(0) = 3$, $x_4(0) = 4$, $x_5(0) = 5$.

For (21), the parameters $\alpha_{r,k}$ and f_r are defined as

$$\alpha_{1,1} = -a, \quad \alpha_{1,3} = a, \quad \alpha_{2,2} = -a,$$
$$\alpha_{2,4} = a, \quad \alpha_{3,1} = b, \quad \alpha_{3,3} = -1,$$
$$\alpha_{4,2} = b, \quad \alpha_{4,4} = -1, \quad \alpha_{5,5} = -1,$$
$$f_3 = -x_1x_5, \quad f_4 = -x_2x_5, \quad f_5 = x_1x_3 + x_2x_4, \tag{23}$$

with all other $\alpha_{r,k}$ and $f_r = 0$ for $r, k = 1, 2, \ldots, 5$.

The results obtained were compared to those from the MATLAB inbuilt solver, ode45. The ode45 solver integrates a system of ordinary differential equations using explicit 4th and 5th Runge-Kutta formula. Tables 3 and 4 show

Table 3: Numerical comparison between MSRM and ode45 for the complex permanent magnet synchronous motor.

t	$x_1(t)$		$x_2(t)$		$x_3(t)$	
	MSRM	ode45	MSRM	ode45	MSRM	ode45
3	−3.85711	−3.85711	−5.66683	−5.66683	−5.20445	−5.20445
10	−0.33729	−0.33729	−0.49554	−0.49554	−0.49104	−0.49104
17	0.12630	0.12631	0.18555	0.18557	0.15550	0.15551
24	0.05091	0.05105	0.07480	0.07501	0.19500	0.19518
31	−2.55034	−2.54878	−3.74694	−3.74465	−0.79819	−0.79326
38	−3.93154	−3.73551	−5.77619	−5.48818	−5.33693	−5.20595

Table 4: Numerical comparison between MSRM and ode45 for the complex permanent magnet synchronous motor.

t	$x_4(t)$		$x_5(t)$	
	MSRM	ode45	MSRM	ode45
3	−7.64635	−7.64635	15.05932	15.05932
10	−0.72144	−0.72143	10.73663	10.73663
17	0.22846	0.22848	14.25582	14.25583
24	0.28649	0.28675	19.33844	19.33921
31	−1.17270	−1.16545	25.34856	25.35739
38	−7.84098	−7.64855	14.98250	14.03140

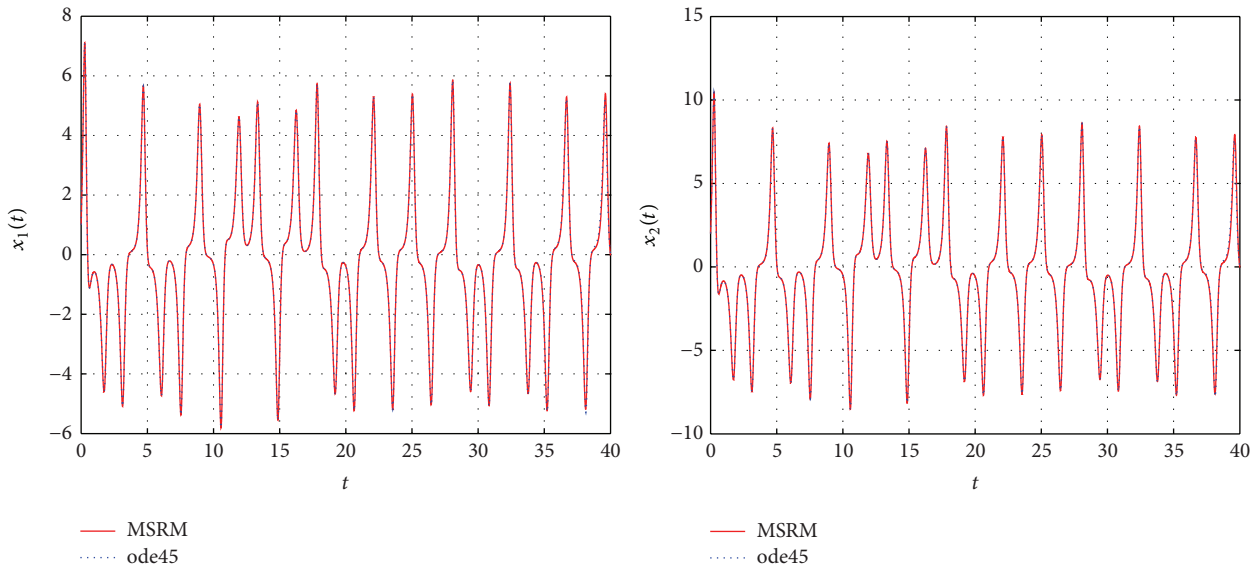


Figure 6: Comparison between the MSRM and ode45 results for the complex permanent magnet synchronous motor.

a comparison of the solutions of the complex permanent magnet synchronous motor computed by the MSRM and ode45. In Figures 6, 7, and 8, the MSRM graphical results are also compared with ode45 and good agreement is observed. The MRSM phase portraits in Figures 9 and 10 were also found to be exactly the same as those computed using ode45. This shows that the proposed MSRM is a valid tool for solving the complex permanent magnet synchronous motor.

4. Conclusion

In this paper, we have applied a spectral method called the multistage spectral relaxation method (MSRM) for the solutions of hyperchaotic complex systems. The proposed MSRM was developed using the Gauss-Seidel idea of decoupling systems of equations and using Chebyshev pseudospectral methods to solve the resulting decoupled system on a sequence of multiple intervals. The proposed MSRM was

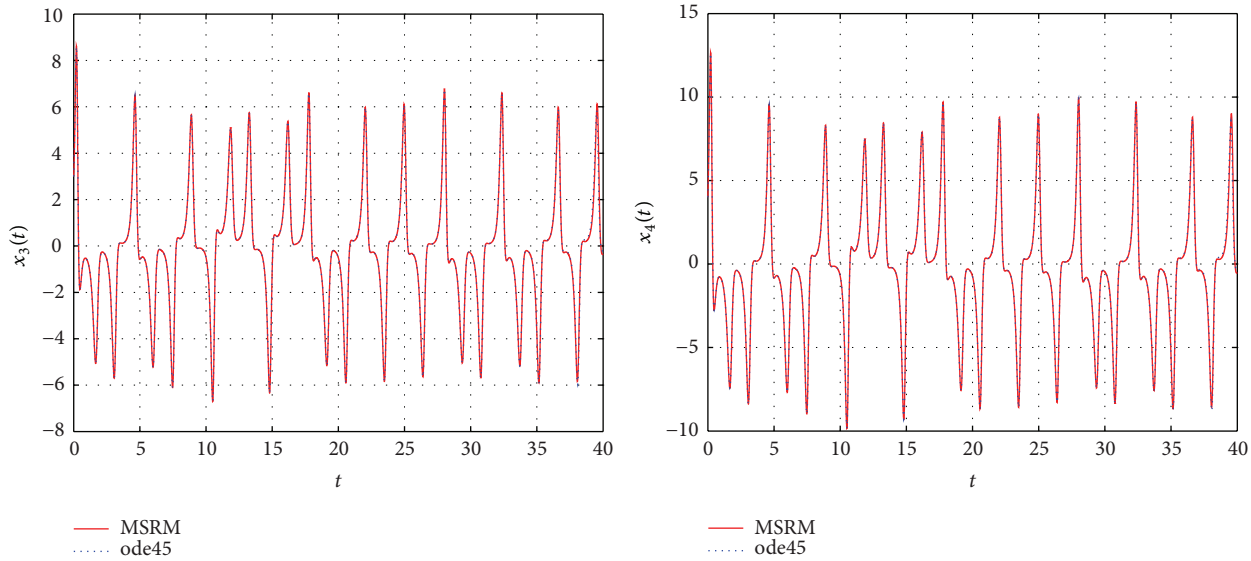


Figure 7: Comparison between the MSRM and ode45 results for the complex permanent magnet synchronous motor.

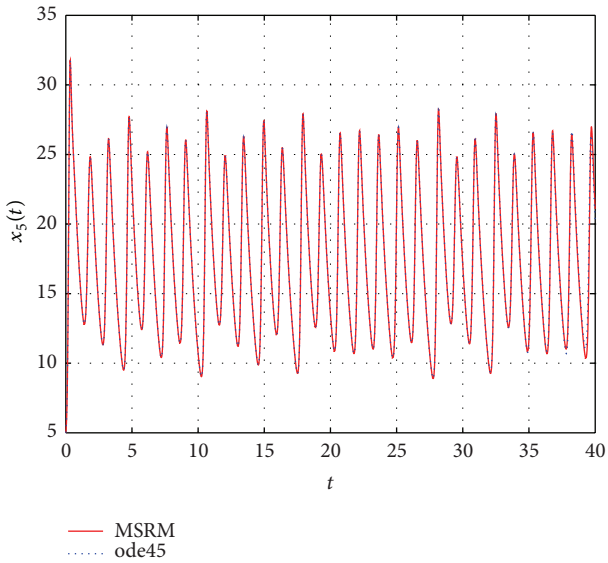


Figure 8: Comparison between the MSRM and ode45 results for the complex permanent magnet synchronous motor.

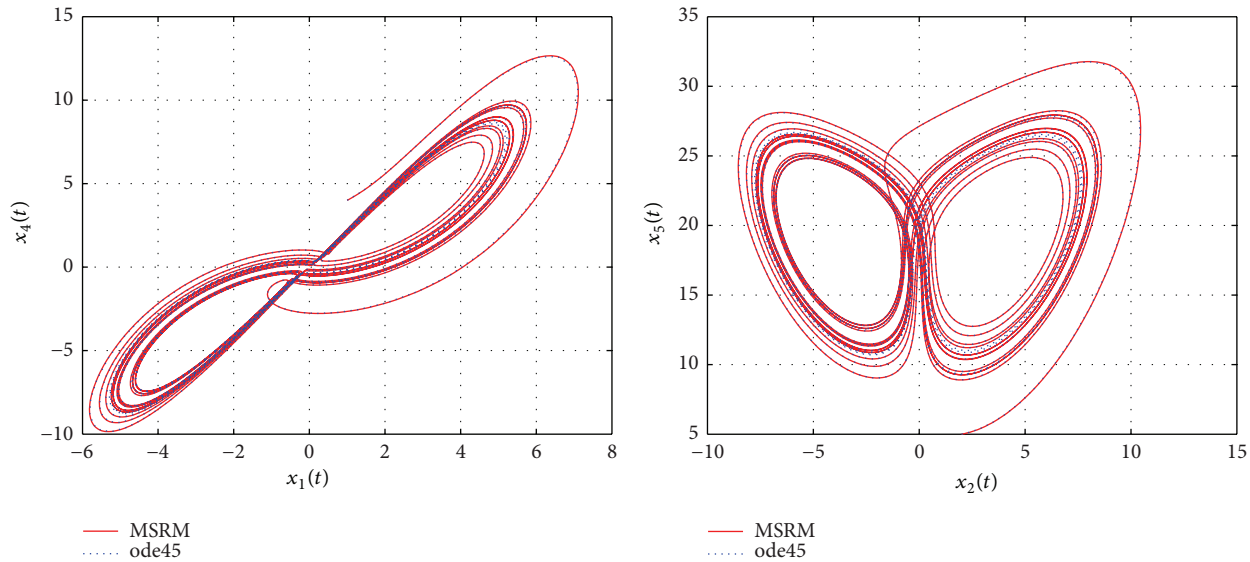


Figure 9: Phase portraits of the complex permanent magnet synchronous motor.

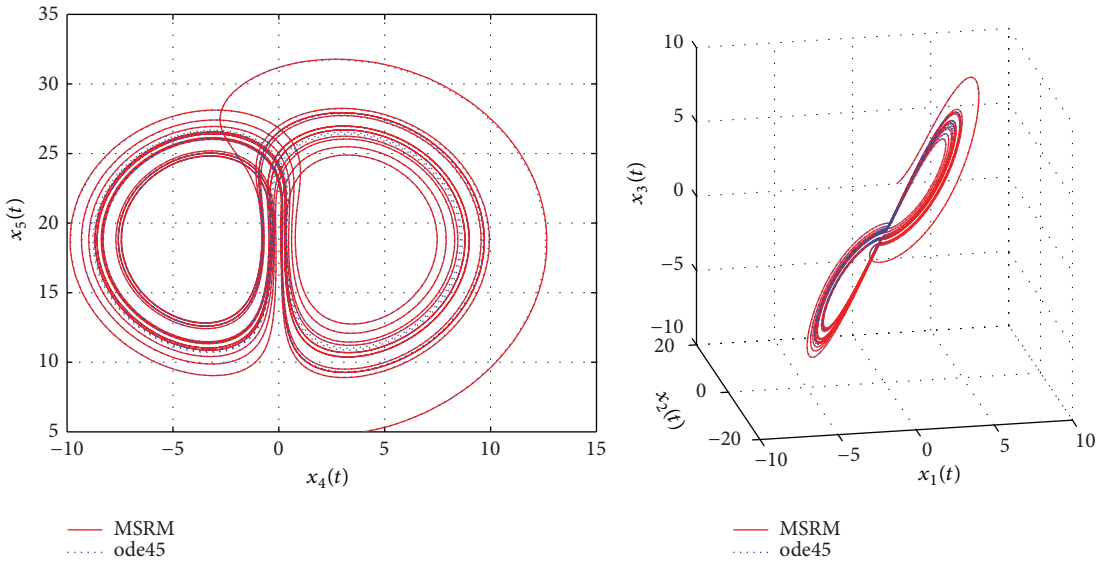


FIGURE 10: Phase portraits of the complex permanent magnet synchronous motor.

used to solve the hyperchaotic complex Lorenz system and complex permanent magnet synchronous motor. The accuracy and validity of the proposed method was tested against Matlab Runge-Kutta based inbuilt solvers and against previously published results.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] G. Chen and T. Ueta, "Yet another chaotic attractor," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 9, no. 7, pp. 1465–1466, 1999.

[2] Z. Wei, R. Wang, and A. Liu, "A new finding of the existence of hidden hyperchaotic attractors with no equilibria," *Mathematics and Computers in Simulation*, vol. 100, pp. 13–23, 2014.

[3] A. C. Fowler, J. D. Gibbon, and M. J. McGuinness, "The complex Lorenz equations," *Physica D: Nonlinear Phenomena*, vol. 4, no. 2, pp. 139–163, 1981/82.

[4] A. G. Vladimirov, V. Y. Toronov, and V. L. Derbov, "Properties of the phase space and bifurcations in the complex Lorenz model," *Technical Physics*, vol. 43–48, pp. 877–884, 1998.

[5] A. Rauh, L. Hannibal, and N. B. Abraham, "Global stability properties of the complex Lorenz model," *Physica D: Nonlinear Phenomena*, vol. 99, no. 1, pp. 45–58, 1996.

[6] R. L. Lang, "A stochastic complex model with random imaginary noise," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 561–565, 2010.

[7] G. M. Mahmoud, M. E. Ahmed, and E. E. Mahmoud, "Analysis of hyperchaotic complex Lorenz systems," *International Journal of Modern Physics C*, vol. 19, no. 10, pp. 1477–1494, 2008.

[8] G. M. Mahmoud, E. E. Mahmoud, and M. E. Ahmed, "On the hyperchaotic complex Lü system," *Nonlinear Dynamics*, vol. 58, no. 4, pp. 725–738, 2009.

[9] G. M. Mahmoud, M. A. Al-Kashif, and S. A. Aly, "Basic properties and chaotic synchronization of complex Lorenz system," *International Journal of Modern Physics C*, vol. 18, no. 2, pp. 253–265, 2007.

[10] A. K. Alomari, M. S. Noorani, and R. Nazar, "Explicit series solutions of some linear and nonlinear Schrodinger equations via the homotopy analysis method," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 4, pp. 1196–1207, 2009.

[11] S. Effati, H. Saberi Nik, and M. Shirazian, "An improvement to the homotopy perturbation method for solving the Hamilton-Jacobi-Bellman equation," *IMA Journal of Mathematical Control and Information*, vol. 30, no. 4, pp. 487–506, 2013.

[12] M. S. Chowdhury and I. Hashim, "Application of multistage homotopy-perturbation method for the solutions of the Chen system," *Nonlinear Analysis: Real World Applications*, vol. 10, no. 1, pp. 381–391, 2009.

[13] S. M. Goh, M. S. Noorani, and I. Hashim, "On solving the chaotic Chen system: a new time marching design for the variational iteration method using Adomian's polynomial," *Numerical Algorithms*, vol. 54, no. 2, pp. 245–260, 2010.

[14] A. Gökdoğan, M. Merdan, and A. Yildirim, "The modified algorithm for the differential transform method to solution of Genesio systems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, no. 1, pp. 45–51, 2012.

[15] H. Saberi Nik, S. Effati, S. S. Motsa, and M. Shirazian, "Spectral homotopy analysis method and its convergence for solving a class of nonlinear optimal control problems," *Numerical Algorithms*, vol. 65, no. 1, pp. 171–194, 2014.

[16] H. Saberi Nik, S. Effati, S. S. Motsa, and S. Shateyi, "A new piecewise-spectral homotopy analysis method for solving chaotic systems of initial value problems," *Mathematical Problems in Engineering*, vol. 2013, Article ID 583193, 13 pages, 2013.

[17] S. Effati, H. Saberi Nik, and A. Jajarmi, "Hyperchaos control of the hyperchaotic Chen system by optimal control design," *Nonlinear Dynamics*, vol. 73, no. 1-2, pp. 499–508, 2013.

[18] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "Solving hyperchaotic systems using the spectral relaxation method," *Abstract and Applied Analysis*, vol. 2012, Article ID 203461, 18 pages, 2012.

[19] S. S. Motsa, P. Dlamini, and M. Khumalo, "A new multistage spectral relaxation method for solving chaotic initial value systems," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 265–283, 2013.

[20] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.

[21] C. Luo and X. Wang, "Hybrid modified function projective synchronization of two different dimensional complex nonlinear systems with parameters identification," *Journal of the Franklin Institute*, vol. 350, no. 9, pp. 2646–2663, 2013.

[22] J. Yu, B. Chen, H. Yu, and J. Gao, "Adaptive fuzzy tracking control for the chaotic permanent magnet synchronous motor drive system via backstepping," *Nonlinear Analysis: Real World Applications*, vol. 12, no. 1, pp. 671–681, 2011.

[23] F. Zhang, C. Mu, X. Wang, I. Ahmed, and Y. Shu, "Solution bounds of a new complex PMSM system," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1041–1051, 2013.

[24] X. Wang and H. Zhang, "Backstepping-based lag synchronization of a complex permanent magnet synchronous motor system," *Chinese Physics B*, vol. 22, no. 4, Article ID 048902, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 134673, 9 pages
http://dx.doi.org/10.1155/2014/134673

Research Article

On a New Three-Step Class of Methods and Its Acceleration for Nonlinear Equations

T. Lotfi,[1] K. Mahdiani,[1] Z. Noori,[1] F. Khaksar Haghani,[2] and S. Shateyi[3]

[1] *Department of Mathematics, Islamic Azad University, Hamedan Branch, Hamedan, Iran*
[2] *Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran*
[3] *Department of Mathematics and Applied Mathematics, School of Mathematical and Natural Sciences, University of Venda, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 20 July 2014; Accepted 16 August 2014; Published 3 September 2014

Academic Editor: Hassan Saberi Nik

A class of derivative-free methods without memory for approximating a simple zero of a nonlinear equation is presented. The proposed class uses four function evaluations per iteration with convergence order eight. Therefore, it is an optimal three-step scheme without memory based on Kung-Traub conjecture. Moreover, the proposed class has an accelerator parameter with the property that it can increase the convergence rate from eight to twelve without any new functional evaluations. Thus, we construct a with memory method that increases considerably efficiency index from $8^{1/4} \approx 1.681$ to $12^{1/4} \approx 1.861$. Illustrations are also included to support the underlying theory.

1. Introduction

The first attempts for classifying iterative root-finding methods were done by Traub [1]. He divided iterative methods for finding zeros of a function into two sets: one-point and multipoint methods. There is a fact about how to create a new method. As Traub investigated in his book [1], and Kung and Traub mentioned in [2], construction of one-point methods is not a useful task. In other words, to construct an optimal one-point method with convergence order n, we need n functional evaluations, while for construction an optimal method without memory having convergence order 2^n; only $n + 1$ function evaluations are required.

To be more precise, constructing an optimal one-point method with eighth-order convergence needs eight function evaluations, while constructing an optimal three-point method without memory having the same convergence order requires four functional evaluations. As a result, many researchers have paid much attention to construct optimal multipoint iterations without memory based on the unproved conjecture due to Kung and Traub: any multipoint iteration without memory using $n + 1$ function evaluations can reach the optimal order 2^n.

This work follows two main goals: frst developing a new optimal three-step derivative-free class of methods without memory and second developing the proposed class to methods with memory. In this way, it reaches convergence order 12 without any new functional evaluations. Because of the derivative-free property of the proposed class, it can be used for finding zeros of not only smooth functions but also nonsmooth ones. Moreover, as we pointed out above we can reach the convergence order 12 using the same functional evaluations (to three-step without memory iterations) and, therefore, increasing 50% convergence order is the other aspect and contribution of this work.

Note that in most test problems for nonlinear equations computing derivatives are an easy exercise. However, for some practical problems computing the derivative might be a cumbersome task and we have to relay on methods free of derivatives. For further reading on this topic, one may refer to [3–6].

The paper is organized as follows. First, a new without memory family of optimal order eight, consuming four function evaluations per iteration, is proposed by using two weight functions in Section 2. A different way to compute the order of convergence for iterative methods that use divided

differences instead of derivatives is presented in Section 3, when we derive a method with memory. The significant increase of convergence speed is achieved without additional function evaluations, which is the main advantage of such methods. Section 4 is devoted to numerical results connected to the order of the methods with and without memory. And finally, concluding remarks will be drawn in Section 5.

2. Construction of a New Three-Step Class

Let the scalar function $f : D \subset \mathbb{R} \to \mathbb{R}$ and $f(\alpha) = 0 \neq f'(\alpha) = c_1$. In other words, α is a simple zero of $f(x) = 0$. In this section, we start with the three-step scheme:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f'(x_k)}, \\ z_k &= y_k - \frac{f(y_k)}{f'(y_k)}, \\ x_{k+1} &= z_k - \frac{f(z_k)}{f'(z_k)}, \end{aligned} \tag{1}$$

where $k = 0, 1, \ldots$, is the iteration indicator. The order of convergence for (1) is eight but its computational efficiency is low. We substitute derivatives in all three steps by suitable approximations that use available data; thus we introduce the following approximations:

$$\begin{aligned} f'(x_k) &\approx f[x_k, w_k], \\ &\text{where } w_k = x_k + \beta f(x_k), \\ f'(y_k) &\approx \frac{f[y_k, w_k]}{H(u_k, v_k)}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \ v_k = \frac{f(y_k)}{f(w_k)}, \\ f'(z_k) &\approx \frac{f[y_k, z_k] + f[w_k, y_k, z_k](z_k - y_k)}{W(s_k)}, \\ s_k &= \frac{f(z_k)}{f(x_k)}, \end{aligned} \tag{2}$$

in the first, second, and third steps of (1), where H and W are weight functions. The following iterative family of three-step methods is obtained:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta f(x_k), \\ &\beta \in R, \\ z_k &= y_k - H(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \\ v_k &= \frac{f(y_k)}{f(w_k)}, \\ x_{k+1} &= z_k - W(s_k) \\ &\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\ s_k &= \frac{f(z_k)}{f(x_k)}. \end{aligned} \tag{3}$$

In the following theorem, we state suitable conditions for deriving a new optimal class without memory according to the Kung and Traub conjecture [2] (also known as K-T hypothesis).

Theorem 1. *Let $f : D \subset \mathbb{R} \to \mathbb{R}$ be a scalar function which has a simple root α in the open interval D, and also the initial approximation x_0 is sufficiently close to the simple zero. Then, the three-step iterative method* (3) *has eighth-order under the conditions $W(0) = W'(0) = 1$, $H(0,0) = H_u(0,0) = H_{uu}(0,0) = 1$, $H_v(0,0) = H_{vv}(0,0) = 0$, and $H_{uv}(0,0) = 2$ and satisfies the following error equation:*

$$\begin{aligned} e_{k+1} = &\left(1 + \beta f'(\alpha)\right)^4 c_2^2 c_3 \\ &\times \left(\left(9 + \beta f'(\alpha)\left(7 + \beta f'(\alpha)\right)\right)c_2^3\right. \\ &\left. + 2c_2c_3 - c_4\right)e_k^8 + O\left(e_k^9\right). \end{aligned} \tag{4}$$

Proof. By using Taylor's expansion of $f(x)$ around α and taking into account that $f(\alpha) = 0$, we obtain

$$\begin{aligned} f(x) = f'(\alpha)\left(e_k + c_2e_k^2 + c_3e_k^3 + c_4e_k^4\right. \\ \left. + c_5e_k^5 + c_6e_k^6 + c_7e_k^7 + c_8e_k^8 + O\left(e_k^9\right)\right), \end{aligned} \tag{5}$$

where $e_k = x_k - \alpha$, $e_{k,w} = w_k - \alpha$, $e_{k,y} = y_k - \alpha$, $e_{k,z} = z_k - \alpha$, and $c_k = f^{(k)}(\alpha)/k!f'(\alpha)$. Therefore,

$$\begin{aligned} e_{k,w} = &\left(1 + \beta f'(\alpha)\right)e_k + \beta f'(\alpha)c_2e_k^2 + \beta f'(\alpha)c_3e_k^3 \\ &+ \beta f'(\alpha)c_4e_k^4 + \beta f'(\alpha)c_5e_k^5 + \beta f'(\alpha)c_6e_k^6 \\ &+ \beta f'(\alpha)c_7e_k^7 + \beta f'(\alpha)c_8e_k^8 + O\left(e_k^9\right), \end{aligned} \tag{6}$$

$$\begin{aligned} f(w_k) = &f'(\alpha)\left(1 + f'(\alpha)\beta\right)e_k \\ &+ f'(\alpha)\left(1 + f'(\alpha)\beta\left(3 + f'(\alpha)\beta\right)\right)c_2e_k^2 \\ &+ \cdots + O\left(e_k^9\right). \end{aligned} \tag{7}$$

Note that we used $\cdots$ in order to avoid writing further terms of the Taylor expansions due to symbolic computations. By using (5) and (7), we obtain

$$\begin{aligned} f[x_k, w_k] &= f'(\alpha)\left(1+\left(2+\beta f'(\alpha)\right)\right)c_2 e_k \\ &\quad + f'(\alpha)\left(\beta f'(\alpha)\, c_2^2 + \left(3+\beta f'(\alpha)\right.\right. \\ &\quad \left.\left.\times\left(3+\beta f'(\alpha)\right)\right)c_3\right)e_k^2 \\ &\quad + \cdots + O\left(e_k^8\right). \end{aligned} \tag{8}$$

Dividing (5) by (8) gives us

$$\begin{aligned} \frac{f(x_k)}{f[x_k, w_k]} &= e_k - \left(1+\beta f'(\alpha)\right)c_2 e_k^2 \\ &\quad + \left(\left(2+\beta f'(\alpha)\left(2+\beta f'(\alpha)\right)\right)c_2^2\right. \\ &\quad \left. - \left(1+\beta f'(\alpha)\right)\left(2+\beta f'(\alpha)\right)c_3\right)e_k^3 \\ &\quad + \cdots + O\left(e_k^9\right). \end{aligned} \tag{9}$$

And thus,

$$e_{k,y} = -\left(1+\beta f'(\alpha)\right)^2 c_2 c_3 e_k^4 + \cdots + O\left(e_k^9\right). \tag{10}$$

Subsequently, we have

$$\begin{aligned} f[y_k, w_k] &= f'(\alpha) + f'(\alpha)\left(1+\beta f'(\alpha)\right)c_2 e_k \\ &\quad + f'(\alpha)\left(\left(1+2\beta f'(\alpha)\right)c_2^2\right. \\ &\quad \left. + \left(1+\beta f'(\alpha)\right)^2 c_3\right)e_k^2 \\ &\quad + \cdots + O\left(e_k^8\right). \end{aligned} \tag{11}$$

Thus,

$$\begin{aligned} h(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]} &= \left(1+\beta f'(\alpha)\right)c_2 e_k^2 \\ &\quad + \left(-\left(2+\beta f'(\alpha)\left(2+\beta f'(\alpha)\right)\right)c_2^2\right. \\ &\quad \left. + \left(1+\beta f'(\alpha)\right)\left(2+\beta f'(\alpha)\right)c_3\right) \\ &\quad \times e_k^3 + \cdots + O\left(e_k^9\right), \\ f[w_k, y_k, z_k] &= f'(\alpha)c_2 + f'(\alpha)\left(1+\beta f'(\alpha)\right)c_3 e_k \\ &\quad + f'(\alpha)\left(\left(1+2\beta f'(\alpha)\right)c_2 c_3\right. \\ &\quad \left. + \left(1+\beta f'(\alpha)\right)^2 c_4\right)e_k^2 \\ &\quad + \cdots + O\left(e_k^9\right). \end{aligned} \tag{12}$$

Therefore, we attain

$$\begin{aligned} &\frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\ &\quad = -\left(1+\beta f'(\alpha)\right)^2 c_2 c_3 e_k^4 + \cdots + O\left(e_k^9\right). \end{aligned} \tag{13}$$

Finally, according to the above analysis, the general error equation is given by

$$\begin{aligned} e_{k+1} &= e_{k,z} - W(s_k)\frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\ &= \left(1+\beta f'(\alpha)\right)^4 c_2^2 c_3 \\ &\quad \times \left(\left(9+\beta f'(\alpha)\left(7+\beta f'(\alpha)\right)\right)c_2^3\right. \\ &\quad \left. + 2c_2 c_3 - c_4\right)e_k^8 + O\left(e_k^9\right), \end{aligned} \tag{14}$$

so that the proof of the theorem is finished. □

We provide some specific weight functions that satisfy the conditions of Theorem 1 as follows:

$$\begin{aligned} H_1(u_k, v_k) &= 1 + u_k + 2u_k v_k + u_k^2, \\ H_2(u_k, v_k) &= \frac{1}{1 - u_k - 2u_k v_k}, \\ W_1(s_k) &= \cos(s_k) + \sin(s_k), \\ W_2(s_k) &= \frac{1}{1 - s_k}, \\ W_3(s_k) &= 1 + s_k, \\ W_4(s_k) &= e^{s_k}. \end{aligned} \tag{15}$$

We consider these weight functions in, without and with memory methods, (3) and (18) in the forthcoming sections.

There are some measures for comparing various iterative techniques. Traub [1] introduced the informational efficiency and efficiency index, which can be expressed in terms of the order (r) of the method and the number of function (and derivative) evaluations (ρ). In fact, the efficiency index (or computational efficiency) is given by $E = r^{1/\rho}$.

Clearly, the efficiency index of the proposed optimal class of method is $8^{1/4} \approx 1.682$ which is optimal in the sense of K-T hypothesis and is higher than two- or one-step methods without memory.

It is worth emphasizing that the maximal order of convergence is not the only goal in constructing root-finding methods and, consequently, the ultimate measure of efficiency of the designed method. Complexity of the formulae involved, often called combinatorial cost, makes another important parameter, which should be taken into account. Hence, we wish to construct a new method with memory possessing a high order 12 requiring only 4 functional evaluations (just like (3)).

In the next section, we will modify the proposed method and introduce a new method. We use an accelerator parameter to increase the order of convergence significantly.

3. Construction of a Method with Memory

Error equation (14) indicates that the order of convergence for class (3) is equal to eight. This section is concerned with extracting an efficiency with memory method from (3) since its error equation contains the parameter β which can be approximated in such a way that increases the local order of convergence.

We set $\beta = \beta_k$ as the iteration proceeds by the formula $\beta_k = -1/\overline{f'}(\alpha)$ for $k = 1, 2, \ldots$, where $\overline{f'}(\alpha)$ is an approximation of $f'(\alpha)$. We have a method via the following forms of β_k:

$$\beta_k = -\frac{1}{\overline{f'}(\alpha)} = -\frac{1}{N_4'(x_k)}. \tag{16}$$

The key idea that provides the order acceleration lies in a special form of the error relation and a convenient choice of a free parameter. We define a self-accelerating parameter, which is calculated during the iterative process using Newton's interpolating polynomial.

Hence, we consider Newton's interpolation as the method for approximating $f'(\alpha)$, where $N_4(t)$ is Newton's interpolation polynomial of fourth degree, set through five available approximations $x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}$ as follows:

$$\begin{aligned}
N_4'(x_k) &= \left[\frac{d}{dt}N_4(t)\right]_{t=x_k} \\
&= \Big[\frac{d}{dt}\left(f(x_k) + f[x_k, z_{k-1}](t - x_k)\right. \\
&\qquad + f[x_k, z_{k-1}, y_{k-1}](t - x_k) \\
&\qquad \times (t - z_{k-1}) + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](t - x_k) \\
&\qquad \times (t - z_{k-1})(t - y_{k-1}) \\
&\qquad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}] \\
&\qquad \times (t - x_k)(t - z_{k-1})(t - y_{k-1}) \\
&\qquad \left.\times (t - x_{k-1})\right)\Big]_{t=x_k} \\
&= f[x_k, z_{k-1}] + f[x_k, z_{k-1}, y_{k-1}](x_k - z_{k-1}) \\
&\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}](x_k - z_{k-1})(x_k - y_{k-1}) \\
&\quad + f[x_k, z_{k-1}, y_{k-1}, x_{k-1}, w_{k-1}](x_k - z_{k-1}) \\
&\quad \times (x_k - y_{k-1})(x_k - x_{k-1}).
\end{aligned} \tag{17}$$

Here, the with memory development of (3) can be presented as follows:

$$\begin{aligned}
y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\
z_k &= y_k - H(u_k, v_k)\frac{f(y_k)}{f[y_k, w_k]}, \quad u_k = \frac{f(y_k)}{f(x_k)}, \\
&\qquad\qquad v_k = \frac{f(y_k)}{f(w_k)}, \\
x_{k+1} &= z_k - W(s_k) \\
&\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)} \\
&\qquad\qquad s_k = \frac{f(z_k)}{f(x_k)}.
\end{aligned} \tag{18}$$

We attempt to prove that the method with memory (18) has convergence order twelve provided that we use accelerator β_k as in (16). It should be remarked that we have applied the Herzberger's matrix method [7].

Theorem 2. *If an initial approximation x_0 is sufficiently close to the zero α of $f(x)$ and the parameter β_k in the iterative scheme* (18) *is recursively calculated by the forms given in* (16)*, then the order of convergence is twelve.*

Proof. We will use Herzberger's matrix method to determine the order of convergence. Note that the lower bound of order for a single-step s-point method $x_k = G(x_{k-1}, x_{k-2}, \ldots, x_{k-s})$ is the spectral radius of a matrix $M^{(s)} = (m_{ij})$, associated to the method with elements:

$$\begin{aligned}
m_{1,j} &= \text{amount of information required at point } x_{k-j} \\
&\qquad\qquad j = 1, 2, \ldots, s, \\
m_{i,i-1} &= 1 \quad (i = 2, 3, \ldots, s), \\
m_{i,j} &= 0, \quad \text{otherwise.}
\end{aligned} \tag{19}$$

On the other hand, the lower bound of order of an s-step method $G = G_1 \circ G_2 \circ \cdots \circ G_s$ is the spectral radius of the product of matrices $M = M_1 \cdot M_2 \cdots M_s$.

We can express each approximation x_{k+1}, z_k, y_k, and w_k as a function of available information $f(z_k)$, $f(y_k)$, $f(w_k)$, and $f(x_k)$ from the kth iteration and $f(z_{k-1})$, $f(y_{k-1})$, $f(w_{k-1})$, and $f(x_{k-1})$ from the previous iteration, depending on the accelerating technique. Now, we determine the order of convergence for (18) applied for the calculation of β_k.

Method (N4). We use the following matrices to express informational dependence:

$$x_{k+1} = \varphi_1(z_k, y_k, w_k, x_k, z_{k-1}) \longrightarrow M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$z_k = \varphi_2(y_k, w_k, x_k, z_{k-1}, y_{k-1}) \longrightarrow M_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$y_k = \varphi_3(w_k, x_k, z_{k-1}, y_{k-1}, w_{k-1}) \longrightarrow M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$w_k = \varphi_4(x_k, z_{k-1}, y_{k-1}, w_{k-1}, x_{k-1}) \longrightarrow M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{20}$$

Hence, it is easy to derive

$$M^{(N4)} = M_1 M_2 M_3 M_4 = \begin{bmatrix} 8 & 4 & 4 & 4 & 4 \\ 4 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{21}$$

with the eigenvalues $\{12, 0, 0, 0, 0\}$. Consequently, the order of the method with memory (18)-($N4$) is at least twelve. The proof of Theorem 2 is finished. □

Clearly, the proposed with memory scheme possesses a high computational efficiency index $12^{1/4} \approx 1.861$, which makes it interesting for practical problems.

4. Numerical Examples

In this section, we test our proposed methods and compare their results with some other methods of the same order of convergence. The errors $|x_k - \alpha|$ denote approximations to the sought zeros, and $a(-b)$ stands for $a \times 10^{-b}$. Moreover, coc indicates computational order of convergence and is computed by

$$\text{coc} = \frac{\log\left(\left|f(x_k)/f(x_{k-1})\right|\right)}{\log\left(\left|f(x_{k-1})/f(x_{k-2})\right|\right)}. \tag{22}$$

The calculated value coc estimates the theoretical order of convergence well when "pathological behavior" of the iterative method (for instance, slow convergence at the beginning of the implemented iterative method, "oscillating" behavior of approximations, etc.) does not exist.

We have used 1000-fixed floating point arithmetic so as to minimize the effect of round-off errors.

By using weight functions (15), we introduce some concrete methods based on the proposed class. Note that it is assumed that the initial estimate β_0 should be chosen before starting the iterative process and also x_0 is given suitably.

Concrete method 1:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\ z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]}, \\ u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\ x_{k+1} &= z_k - \left(\cos(s_k) + \sin(s_k)\right) \\ &\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\ s_k &= \frac{f(z_k)}{f(x_k)}. \end{aligned} \tag{23}$$

Concrete method 2:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k), \\ z_k &= y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]}, \\ u_k &= \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \\ x_{k+1} &= z_k - \left(\frac{1}{1 - s_k}\right) \\ &\quad \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)}, \\ s_k &= \frac{f(z_k)}{f(x_k)}. \end{aligned} \tag{24}$$

Concrete method 3:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$

$$z_k = y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)},$$
$$x_{k+1} = z_k - (1 + s_k) \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$s_k = \frac{f(z_k)}{f(x_k)}. \tag{25}$$

Concrete method 4:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$
$$z_k = y_k - \left(1 + u_k + 2u_k v_k + u_k^2\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \tag{26}$$
$$x_{k+1} = z_k - \left(e^{s_k}\right) \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$s_k = \frac{f(z_k)}{f(x_k)}.$$

Concrete method 5:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$
$$z_k = y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \tag{27}$$
$$x_{k+1} = z_k - (\cos(s_k) + \sin(s_k)) \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$s_k = \frac{f(z_k)}{f(x_k)}.$$

Concrete method 6:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$
$$z_k = y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \tag{28}$$
$$x_{k+1} = z_k - \left(\frac{1}{1 - s_k}\right) \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$w_k = \frac{f(z_k)}{f(x_k)}.$$

Concrete method 7:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$
$$z_k = y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \tag{29}$$
$$x_{k+1} = z_k - (1 + s_k) \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$w_k = \frac{f(z_k)}{f(x_k)}.$$

Concrete method 8:

$$y_k = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta_k f(x_k),$$
$$z_k = y_k - \left(\frac{1}{1 - u_k - 2u_k v_k}\right) \frac{f(y_k)}{f[y_k, w_k]},$$
$$u_k = \frac{f(y_k)}{f(x_k)}, \quad v_k = \frac{f(y_k)}{f(w_k)}, \tag{30}$$
$$x_{k+1} = z_k - \left(e^{s_k}\right) \times \frac{f(z_k)}{f[z_k, y_k] + f[w_k, y_k, z_k](z_k - y_k)},$$
$$w_k = \frac{f(z_k)}{f(x_k)}.$$

Several iterative methods (IM) of optimal order eight, which also require four function evaluations, for comparisons with our proposed methods have been chosen.

Table 1: Consider $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x\log(x\sin(x)+1)$, $\alpha = 0$, $x_0 = 0.6$, $\beta = -0.01$.

Methods without memory	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	coc
Our method (23)	5.3810(−3)	5.5392(−25)	6.9091(−201)	8.0003
Our method (24)	2.1802(−3)	3.9973(−28)	5.0816(−226)	8.0001
Our method (25)	4.2865(−3)	8.9612(−26)	3.2420(−207)	8.0002
Our method (26)	3.1912(−3)	8.4384(−27)	2.0042(−215)	8.0002
Our method (27)	5.7030(−3)	4.9625(−25)	1.6203(−201)	8.0002
Our method (28)	2.3293(−3)	3.8278(−28)	2.0304(−226)	8.0001
Our method (29)	5.7030(−3)	4.9625(−25)	1.6203(−201)	8.0002
Our method (30)	4.5488(−3)	8.1186(−26)	8.3149(−208)	8.0002
(34), $\beta = -0.01$	9.7732(−3)	4.3543(−24)	6.6317(−195)	8.0002
(35), $\beta = -0.01$	1.7342(−2)	1.1724(−21)	3.5692(−175)	8.0085

Three-step method by Wang et al. [8]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k)}{f[y_k, w_k]} H(t_k), \quad t_k = \frac{f(y_k)}{f(x_k)}, \\ x_{k+1} &= z_k - \frac{f(z_k)}{f[z_k, w_k]} G(t_k, s_k), \quad s_k = \frac{f(z_k)}{f(y_k)}, \end{aligned} \tag{31}$$

where functions H and G are following $H_1(t) = 1 + t$, $G_1(t,s) = 1 + t + s + 2ts - (1+\lambda)t^3$, and $\lambda = 1/(1+\beta f[x,w])$.

Three-step method by Lotfi and Tavakoli [9]:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, \\ z_n &= y_n - H(t_n, u_n)\frac{f(y_n)}{f[y_n, w_n]}, \quad t_n = \frac{f(y)}{f(x)}, \\ &\qquad u_n = \frac{f(w)}{f(x)}, \\ x_{k+1} &= z_n - G(t_n, s_n) W(v_n, s_n) \frac{f(z_n)}{f[z_n, w_n]}, \\ &\qquad s_n = \frac{f(z)}{f(y)}, \quad v_n = \frac{f(z)}{f(x)}, \end{aligned} \tag{32}$$

where

$$\begin{aligned} W(s_n, v_n) &= 1 + s_n^2 + v_n^2, \\ G(t_n, s_n) &= 1 + t_n + s_n + 2t_n s_n + (-1 - \phi_n) t_n^3, \\ &\left(\phi_n = \frac{1}{1 + \beta_n f[x_n, w_n]}\right), \\ H(t_n, u_n) &= 1 + t_n. \end{aligned} \tag{33}$$

Derivative-free Kung-Traub's family [2]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k) f(w_k)}{[f(w_k) - f(y_k)] f[x_k, y_k]}, \\ x_{k+1} &= z_k - \frac{f(y_k) f(w_k)(y_k - x_k + f(x_k)/f[x_k, z_k])}{[f(y_k) - f(z_k)][f(w_k) - f(z_k)]} \\ &\quad + \frac{f(y_k)}{f[y_k, z_k]}. \end{aligned} \tag{34}$$

Three-step methods made by Zheng et al. [10]:

$$\begin{aligned} y_k &= x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad w_k = x_k + \beta f(x_k), \\ z_k &= y_k - \frac{f(y_k)}{f[y_k, w_k] + f[y_k, x_k, w_k](y_k - x_k)}, \\ x_{k+1} &= z_k - \big(f(z_k) \times (f[z_k, y_k] + f[z_k, y_k, x_k](z_k - y_k) \\ &\qquad + f[z_k, y_k, x_k, w_k](z_k - y_k) \\ &\qquad \times (z_k - x_k))^{-1}\big). \end{aligned} \tag{35}$$

In Tables 1, 3, and 5 our without memory proposed method by different weight functions (23)–(30) has been compared with optimal three-point methods (34) and (35), and we observe that all these methods behave very well practically and confirm their theoretical results.

Also Tables 2, 4, and 6 present numerical results for our with memory classes (23)–(30). It is also clear that all these methods behave very well practically and confirm their relevant theories. They all provide wherea bout twelve of convergence order.

Table 2: Consider $f_1(x) = \sin(\pi x)e^{(x^2+x\cos(x)-1)} + x\log(x\sin(x)+1)$, $\alpha = 0$, $x_0 = 0.6$, $\beta_0 = -0.01$.

Methods with memory	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	coc
Our method (23)	5.3810(−3)	2.7788(−35)	1.8668(−426)	12.1156
Our method (24)	2.1802(−3)	3.0336(−38)	5.3469(−462)	12.1571
Our method (25)	4.2865(−3)	5.2041(−36)	3.4739(−435)	12.1273
Our method (26)	3.1912(−3)	5.6445(−37)	9.2065(−447)	12.1411
Our method (27)	5.7030(−3)	4.6138(−35)	8.1591(−424)	12.1138
Our method (28)	2.3293(−3)	5.4580(−38)	6.1228(−459)	12.1556
Our method (29)	4.5488(−3)	8.7828(−36)	1.8467(−432)	12.1256
Our method (30)	3.3939(−3)	9.7470(−37)	6.4438(−444)	12.1395
(31), $\beta = -1$	2.1170(−3)	1.1077(−41)	3.4407(−501)	12.0035
(32), $\beta = -1$	5.7578(−3)	2.7125(−39)	1.4859(−474)	11.9819

Table 3: Consider $f_2(x) = e^{-5x}(x-2)(x^{10}+x+2)$, $\alpha = 2$, $x_0 = 2.2$, $\beta = -1$.

Methods without memory	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	coc
Our method (23)	5.4211(−6)	7.6321(−54)	1.1776(−436)	8.0000
Our method (24)	5.4981(−6)	8.5429(−54)	2.9020(−436)	8.0000
Our method (25)	5.4468(−6)	7.9261(−54)	1.5935(−436)	8.0001
Our method (26)	5.4725(−6)	8.2301(−54)	2.1533(−436)	8.0000
Our method (27)	3.2606(−6)	1.3070(−55)	8.7125(−451)	8.0000
Our method (28)	3.3757(−6)	1.7249(−55)	8.0153(−450)	8.0000
Our method (29)	3.2991(−6)	1.4354(−55)	1.8434(−450)	8.0000
Our method (30)	3.3375(−6)	1.5747(−55)	3.8666(−450)	8.0000
(34), $\beta = -1$	8.0552(−6)	1.4131(−52)	1.2677(−456)	8.0000
(35), $\beta = -0.01$	3.9453(−6)	1.2705(−59)	5.0021(−491)	8.0085

Table 4: Consider $f_2(x) = e^{-5x}(x-2)(x^{10}+x+2)$, $\alpha = 2$, $x_0 = 2.2$, $\beta_0 = -1$.

Methods with memory	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	coc
Our method (23)	5.4211(−6)	2.1857(−80)	4.4911(−970)	11.9590
Our method (24)	5.4981(−6)	2.4529(−80)	1.7931(−969)	11.9587
Our method (25)	5.4468(−6)	2.2719(−80)	7.1443(−970)	11.9589
Our method (26)	5.4725(−6)	2.3611(−80)	9.2065(−969)	11.9588
Our method (27)	5.4981(−6)	2.4529(−80)	1.7931(−969)	11.9587
Our method (28)	3.3757(−6)	7.7624(−82)	1.8084(−987)	11.9732
Our method (29)	3.2991(−6)	6.4445(−82)	1.9393(−988)	11.9737
Our method (30)	3.3375(−6)	7.0781(−82)	5.9748(−988)	11.9735
(31), $\beta = -1$	4.9425(−5)	1.5275(−69)	4.1256(−874)	12.0535
(32), $\beta = -1$	4.0518(−6)	8.2730(−77)	2.4267(−861)	11.0982

Table 5: Consider $f_3(x) = e^{x^3-x} - \cos(x^2-1) + x^3 + 1$, $\alpha = -1$, $x_0 = -1.65$, $\beta = -1$.

Methods without memory	$\lvert x_1 - \alpha\rvert$	$\lvert x_2 - \alpha\rvert$	$\lvert x_3 - \alpha\rvert$	coc
Our method (23)	6.0256(−3)	4.0898(−24)	1.8272(−193)	8.0001
Our method (24)	5.9797(−3)	3.8471(−24)	1.1202(−193)	8.0001
Our method (25)	6.0104(−3)	4.0081(−24)	1.5548(−193)	8.0001
Our method (26)	5.9952(−3)	3.9278(−24)	1.3225(−193)	8.0001
Our method (27)	5.3628(−3)	1.6071(−24)	1.0388(−196)	8.0001
Our method (28)	5.3357(−3)	1.5431(−24)	7.5042(−197)	8.0001
Our method (29)	5.3537(−3)	1.5856(−24)	9.3292(−197)	8.0001
Our method (30)	5.3448(−3)	1.5644(−24)	8.3766(−197)	8.0001
(34), $\beta = -1$	2.9152(−4)	5.3239(−34)	6.5827(−272)	8.0000
(35), $\beta = -0.01$	6.4036(−5)	4.4335(−44)	2.4457(−357)	7.9995

Table 6: Consider $f_3(x) = e^{x^3-x} - \cos(x^2 - 1) + x^3 + 1, \alpha = -1, x_0 = -1.65, \beta_0 = -1$.

Methods with memory	$\lvert x_1 - \alpha \rvert$	$\lvert x_2 - \alpha \rvert$	$\lvert x_3 - \alpha \rvert$	coc
Our method (23)	6.0256(−3)	4.7294(−38)	3.8067(−459)	11.9952
Our method (24)	5.9797(−3)	4.4345(−38)	1.7580(−459)	11.9955
Our method (25)	6.0104(−3)	4.6300(−38)	2.9502(−459)	11.9953
Our method (26)	5.9959(−3)	4.5324(−38)	2.2850(−459)	11.9954
Our method (27)	5.3628(−3)	9.5497(−39)	1.7425(−467)	11.9929
Our method (28)	5.3357(−3)	9.1483(−39)	1.0410(−467)	11.9931
Our method (29)	5.3538(−3)	9.4151(−39)	1.4696(−467)	11.9929
Our method (30)	5.3448(−3)	9.2821(−39)	1.2391(−467)	11.9930
(31), $\beta = -1$	1.3000(−2)	5.5214(−33)	1.2114(−401)	12.1381
(32), $\beta = -1$	1.0937(−2)	3.5221(−34)	1.7174(−415)	12.1081

5. Concluding Remarks

We have constructed a class of methods without and with memory. Our proposed methods do not need any derivative and therefore are applicable to nonsmooth functions too. Another advantage of the proposed methods is that their without memory versions are optimal in the sense of K-T conjecture. In addition, it contains an accelerator parameter which rises convergence order from eight to twelve without any new functional evaluations. In other words, the efficiency index of the with memory class is $12^{1/4} \approx 1.861$.

We finalize this work by suggesting some outlines for future research: first developing the proposed methods for computing multiple roots and second exploring its dynamic or basins of attractions, and finally we wonder why not to use an adaptive arithmetic in each step of the iterative method instead of using a fixed precision, since this higher precision is only necessary in the last step of the iterative process.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The first three authors thank Islamic Azad University, Hamedan Branch, for the financial support of the present research. Also, the authors would like to express their sincere appreciations to Dr. Soleymani and the respected referees for their constructive comments regarding improving this work.

References

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

[2] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the Association for Computing Machinery*, vol. 21, pp. 643–651, 1974.

[3] S. Artidiello, A. Cordero, J. R. Torregrosa, and M. P. Vassileva, "Optimal high-order methods for solving nonlinear equations," *Journal of Applied Mathematics*, vol. 2014, Article ID 591638, 9 pages, 2014.

[4] B. Ignatova, N. Kyurkchiev, and A. Iliev, "Multipoint algorithms arising from optimal in the sense of Kung-Traub iterative procedures for numerical solution of nonlinear equations," *General Mathematics Notes*, vol. 6, pp. 45–79, 2011.

[5] N. Kyurkchiev and A. Iliev, "On some multipoint methods arising from optimal in the sense of Kung-Traub algorithms," *Biomath*, vol. 2, no. 1, pp. 1–7, 2013.

[6] X. Wang and T. Zhang, "A new family of Newton-type iterative methods with and without memory for solving nonlinear equations," *Calcolo*, vol. 51, no. 1, pp. 1–15, 2014.

[7] J. Herzberger, "Über Matrixdarstellungen für Iterationverfahren bei nichtlinearen Gleichungen," *Computing*, vol. 12, no. 3, pp. 215–222, 1974.

[8] X. Wang, J. Džunić, and T. Zhang, "On an efficient family of derivative free three-point methods for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1749–1760, 2012.

[9] T. Lotfi and E. Tavakoli, "On a new efficient Steffensen-like iterative class by applying a suitable self-accelerator parameter," *The Scientific World Journal*, vol. 2014, Article ID 769758, 9 pages, 2014.

[10] Q. Zheng, J. Li, and F. Huang, "An optimal Steffensen-type family for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 217, no. 23, pp. 9592–9597, 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 727490, 6 pages
http://dx.doi.org/10.1155/2014/727490

Research Article

On a New Iterative Scheme without Memory with Optimal Eighth Order

M. Sharifi,[1] S. Karimi Vanani,[1] F. Khaksar Haghani,[1] M. Arab,[1] and S. Shateyi[2]

[1] *Department of Mathematics, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran*
[2] *Department of Mathematics and Applied Mathematics, University of Venda, Thohoyandou 0950, South Africa*

Correspondence should be addressed to S. Shateyi; stanford.shateyi@univen.ac.za

Received 16 July 2014; Revised 27 July 2014; Accepted 4 August 2014; Published 28 August 2014

Academic Editor: Emran Tohidi

The purpose of this paper is to derive and discuss a three-step iterative expression for solving nonlinear equations. In fact, we derive a derivative-free form for one of the existing optimal eighth-order methods and preserve its convergence order. Theoretical results will be upheld by numerical experiments.

1. Introduction

Assume that $f : D \subseteq \mathbb{R} \to \mathbb{R}$ is sufficiently smooth and that $\alpha \in D$ is its simple zero; that is, $f(\alpha) = 0$. This paper concerns with numerical solution of nonlinear scalar equations by iterative expressions. Considering a known optimal eighth-order method with derivative and the conjecture of Cordero and Torregrosa [1], we construct a family of derivative-free methods without memory for solving a nonlinear equation.

To shortly review the literature, we remind readers of the following. Kung and Traub in [2] have provided a class of n-step derivative-involved methods including n evaluations of the function and one of its first derivatives per full iteration to reach the convergence rate 2^n. They also have given a n-step derivative-free family of one parameter (consuming $n+1$ evaluations of the function) to again achieve the optimal convergence rate 2^n.

Remark 1 (Kung-Traub's conjecture [2]). Multipoint iterative methods without memory, requiring $d + 1$ function evaluations per iteration, have the order of convergence at most 2^d. Multipoint methods which satisfy the Kung-Traub conjecture (still unproved) are called optimal methods.

Some well-known methods with eighth order of convergence can be found at [3]. As another example, Liu and Wang [4] suggested some optimal eighth-order methods using four evaluations per full cycle ($\beta_1, \beta_2 \in \mathbb{R}$) in what follows:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{4f(x_n) - f(y_n)}{4f(x_n) - 9f(y_n)}, \\ x_{n+1} &= z_n - \frac{f(z_n)}{f'(x_n)} \left[\frac{8f(y_n)}{4f(x_n) - 11f(y_n)} \right. \\ &\qquad + \left(1 + \frac{f(z_n)}{3f(y_n) - \beta_1 f(z_n)} \right)^3 \\ &\qquad \left. + \frac{4f(z_n)}{f(x_n) + \beta_2 f(z_n)} \right], \end{aligned} \tag{1}$$

where the efficiency index is 1.682. Reference [4] also suggested the following three-step approach ($\alpha_1, \alpha_2 \in \mathbb{R}$) with the same number of evaluations and efficiency index:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= y_n - \frac{f(y_n)}{f'(x_n)} \frac{f(x_n)}{f(x_n) - 2f(y_n)}, \end{aligned}$$

$$x_{n+1} = z_n - \frac{f(z_n)}{f'(x_n)}\left[\left(\frac{f(x_n)-f(y_n)}{f(x_n)-2f(y_n)}\right)^2 + \frac{f(z_n)}{f(y_n)-\alpha_1 f(z_n)} + \frac{4f(z_n)}{f(x_n)+\alpha_2 f(z_n)}\right]. \tag{2}$$

In what follows, in Section 2, the main derivation is provided to design a new derivative-free family with optimal eighth-order convergence for nonlinear equations. Therein, we confirm the conjecture of Cordero-Torregrosa as well. Section 3 illustrates the accuracy of the new obtained three-step family of iterative methods by comparing the results for some nonlinear test functions. Finally, in Section 4, a conclusion will be drawn.

2. A New Derivative-Free Family

There are a number of papers (see, e.g., [1] and the references therein) about the idea of removing derivatives from the iteration function in order to avoid defining new functions and calculate iterates only by using the function that describes the problem and also trying to preserve the convergence order. The interest of these methods is to be applied on nonlinear equations when there are many problems for obtaining and evaluating the derivatives involved or when there is no analytical function to derive.

Hence, our focus in this work is to derive a method without the use of derivatives for nonlinear equations.

Remark 2 (Cordero and Torregrosa's conjecture [1]). Every time that one applies the approximation of the derivative $f'(x_n) \approx f[x_n, w_n]$, with $w_n = x_n + \beta f(x_n)^l$, on an optimal method with the order $2q$, one needs $l \geq q$ for preserving the order of convergence.

We begin by reminding the readers of the three-step iterative method without memory proposed in [5] with optimal eighth order of convergence:

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f'(x_n)}, \\ z_n &= x_n - \left(\frac{f(x_n)}{f'(x_n)}\frac{f(x_n)}{f(x_n)-f(y_n)}\right)\times\left(1+\left(\frac{f(y_n)}{f(x_n)}\right)^2+3\left(\frac{f(y_n)}{f(x_n)}\right)^3\right), \\ x_{n+1} &= z_n - \left(\frac{f(z_n)}{f[z_n,y_n]+f[z_n,x_n,x_n](z_n-y_n)}\right)\times\left(1+2\frac{f(z_n)}{f(x_n)}-18\left(\frac{f(y_n)}{f(x_n)}\right)^4+\left(\frac{f(z_n)}{f(y_n)}\right)^3\right). \end{aligned} \tag{3}$$

The main aim is to follow Remark 2 and to present a derivative-free form of (3) with optimal eighth order of convergence. Therefore, using the approximation $w_n = x_n + \beta f(x_n)^3$, we present the following formulation ($\beta \in \mathbb{R} \setminus \{0\}$):

$$\begin{aligned} y_n &= x_n - \frac{f(x_n)}{f[x_n,w_n]}, \qquad w_n = x_n + \beta f(x_n)^3, \\ z_n &= x_n - \left(\frac{f(x_n)}{f[x_n,w_n]}\frac{f(x_n)}{f(x_n)-f(y_n)}\right)\times\left(1+\left(\frac{f(y_n)}{f(x_n)}\right)^2+3\left(\frac{f(y_n)}{f(x_n)}\right)^3\right), \\ x_{n+1} &= z_n - \left(\frac{f(z_n)}{f[z_n,y_n]+\varphi_{z_n,x_n,x_n}(z_n-y_n)}\right)\times\left(1+2\frac{f(z_n)}{f(x_n)}-18\left(\frac{f(y_n)}{f(x_n)}\right)^4+\left(\frac{f(z_n)}{f(y_n)}\right)^3\right), \end{aligned} \tag{4}$$

wherein

$$\varphi_{z_n,x_n,x_n} = \frac{f[z_n,x_n]-f[x_n,w_n]}{z_n-x_n}. \tag{5}$$

We shall see that the order of convergence for (4) reaches to the optimal case, that is, 8, with only four function evaluations per full iteration, which means that the proposed uniparametric family of derivative-free methods possesses the high efficiency index 1.682 and can be viewed as the derivative-free formulation of (3).

Theorem 3. *Let $\alpha \in D$ be a simple zero of a sufficiently differentiable function $f : D \subset \mathbb{R} \to \mathbb{R}$ for an open interval D, which includes x_0 as an initial approximation of α. Then, the family of derivative-free methods* (4) *is of optimal order eight.*

Proof. To find the asymptotic error constant of (4) where $c_j = f^{(j)}(\alpha)/j!$, $j \geq 1$, we expand any terms of (4) around the simple root α in the nth iterate. Thus, we write

$$f(x_n) = c_1 e_n + c_2 e_n^2 + c_3 e_n^3 + c_4 e_n^4 + c_5 e_n^5 + c_6 e_n^6 + c_7 e_n^7 + c_8 e_n^8 + O\left(e_n^9\right), \tag{6}$$

where $e_n = x_n - \alpha$ and

$$f(w_n) = c_1 b_n + c_2 b_n^2 + c_3 b_n^3 + c_4 b_n^4 + c_5 b_n^5 + c_6 b_n^6 + c_7 b_n^7 + c_8 b_n^8 + O\left(b_n^9\right), \tag{7}$$

wherein $b_n = w_n - \alpha$. Hence, we obtain

$$x_n - \frac{f(x_n)}{f[x_n,w_n]} - \alpha = \frac{c_2 e_n^2}{c_1} + \frac{2\left(-c_2^2 + c_1 c_3\right)e_n^3}{c_1^2} + \cdots + O\left(e_n^9\right). \tag{8}$$

In the same vein, we have

$$f(y_n) = c_2 e_n^2 + \left(-\frac{2c_2^2}{c_1} + 2c_3\right) e_n^3 + \cdots + O\left(e_n^9\right), \quad (9)$$

and for the second substep, we have

$$\begin{aligned} z_n - \alpha &= -\frac{c_2 c_3}{c_1^2} e_n^4 \\ &+ \frac{\left(-\beta c_1^5 c_2^2 + 9c_2^4 + 2c_1 c_2^2 c_3 - 2c_1^2\left(c_3^2 + c_2 c_4\right)\right)}{c_1^4} e_n^5 \\ &+ \cdots + O\left(e_n^9\right). \end{aligned} \quad (10)$$

At this time, Taylor's series expansion of $f(z_n)$ around the root is needed. We find that

$$\begin{aligned} f(z_n) &= -\frac{c_2 c_3 e_n^4}{c_1} \\ &+ \frac{\left(-\beta c_1^5 c_2^2 + 9c_2^4 + 2c_1 c_2^2 c_3 - 2c_1^2\left(c_3^2 + c_2 c_4\right)\right)}{c_1^3} e_n^5 \\ &+ \cdots + O\left(e_n^9\right), \end{aligned} \quad (11)$$

and subsequently

$$\begin{aligned} \varphi_{z_n, x_n, x_n} = c_2 &+ 2c_3 e_n + \left(\beta c_1^3 c_2 + 3c_4\right) e_n^2 \\ &+ 3\beta c_1^2\left(c_2^2 + c_1 c_3\right) e_n^3 + \cdots + O\left(e_n^9\right). \end{aligned} \quad (12)$$

Considering these Taylor's series expansions in the last step of (4) will result in the following final error equation:

$$e_{n+1} = -\frac{c_2^2 c_3\left(c_2\left(\beta c_1^4 - 2c_3\right) + c_1 c_4\right)}{c_1^5} e_n^8 + O\left(e_n^9\right). \quad (13)$$

This shows that the iterative family of derivative-free methods without memory (4) is of optimal order eight. The proof is complete. □

Remark 4. Theorem 3 clearly supports the conjecture of Cordero-Torregrosa for providing low-complexity derivative-free iterative methods without memory out of optimal methods with derivative.

Note that each method of (4) reaches the efficiency index $\sqrt[4]{8} \approx 1.682$, which is greater than $\sqrt[3]{4} \approx 1.587$ of optimal fourth-order techniques and $\sqrt[2]{2} \approx 1.424$ of optimal Newton's method. It has also the same computational efficiency index with (1), (2), and (3).

Remark 5. It must be remarked that, firstly, the paper [6] studied the multipoint iterative schemes using divided differences for self-acceleration of classical methods.

We here state that the free nonzero parameter β in (4) gives us the ability to increase the convergence R-order of (4) more. Such an acceleration in R-order is known as with memorization (see, e.g., [7]) according the classification of Traub [8] for nonlinear solvers. To be more precise, choosing

$$\beta = \frac{2c_2 c_3 - c_1 c_4}{c_1^4 c_2} \quad (14)$$

would yield an acceleration of convergence.

Anyhow, since the simple zero α and subsequently c_j are not known, one should give an approximation for (14) using an approximation polynomial $A(t) \approx f(t)$ in the domain D. Toward this goal, if we consider $A(t)$ to be Newton's interpolatory polynomial of fourth degree passing through the five available nodes x_{n-1}, w_{n-1}, y_{n-1}, z_{n-1}, and x_n at the end of each cycle, then one has the following approximation:

$$\beta_n = \frac{4A^{(3)}(x_n) - A'(x_n) A^{(4)}(x_n)/A''(x_n)}{12A'(x_n)^4}, \quad (15)$$

using a suitable β_0. Consequently, one is able to derive the following accelerated iterative method with memory:

$$\begin{aligned} \beta_n &= \frac{4A^{(3)}(x_n) - A'(x_n) A^{(4)}(x_n)/A''(x_n)}{12A'(x_n)^4}, \\ y_n &= x_n - \frac{f(x_n)}{f[x_n, w_n]}, \qquad w_n = x_n + \beta_n f(x_n)^3, \\ z_n &= x_n - \left(\frac{f(x_n)}{f[x_n, w_n]} \frac{f(x_n)}{f(x_n) - f(y_n)}\right) \\ &\quad \times \left(1 + \left(\frac{f(y_n)}{f(x_n)}\right)^2 + 3\left(\frac{f(y_n)}{f(x_n)}\right)^3\right), \\ x_{n+1} &= z_n - \left(\frac{f(z_n)}{f[z_n, y_n] + \varphi_{z_n, x_n, x_n}(z_n - y_n)}\right) \\ &\quad \times \left(1 + 2\frac{f(z_n)}{f(x_n)} - 18\left(\frac{f(y_n)}{f(x_n)}\right)^4 + \left(\frac{f(z_n)}{f(y_n)}\right)^3\right). \end{aligned} \quad (16)$$

Obviously, if fewer nodes are used for the interpolating polynomial, slower acceleration is achieved. An increase of convergence is achieved in this way without additional functional evaluations, making the proposed root solvers (16) efficient. This acceleration will be seen in Section 3.

Theorem 6. *Let the function $f(x)$ be sufficiently differentiable in a neighborhood of its simple zero α. If an initial approximation x_0 is sufficiently close to α, then the R-order of convergence of* (16) *is at least* $4 + \sqrt{17}$.

Proof. Let $\{x_n\}$ be a sequence of approximations generated by an iterative method with order p. The error relation with the self-accelerating parameter $\beta = \beta_n$ for (16) is in what follows:

$$e_{n+1} = x_{n+1} - \alpha \sim c_{n,8} e_n^8, \quad (17)$$

Table 1: Results of comparisons for Example 7.

Methods	$\lvert f(x_1)\rvert$	$\lvert f(x_2)\rvert$	$\lvert f(x_3)\rvert$	$\lvert f(x_4)\rvert$	COC
LW8	0.85460	4.8410×10^{-7}	2.8515×10^{-57}	4.1317×10^{-459}	8.00000
SM8	0.85215	1.4818×10^{-8}	3.3414×10^{-70}	2.2345×10^{-563}	8.00000
PM	0.64289	2.2594×10^{-9}	9.7768×10^{-77}	1.2017×10^{-615}	8.00000
APM	0.64289	1.2590×10^{-9}	2.0944×10^{-79}	2.9252×10^{-646}	8.12358

Table 2: Results of comparisons for Example 8.

Methods	$\lvert f(x_1)\rvert$	$\lvert f(x_2)\rvert$	$\lvert f(x_3)\rvert$	$\lvert f(x_4)\rvert$	COC
LW8	0.080790	4.3442×10^{-14}	6.3406×10^{-112}	1.3057×10^{-894}	8.00000
SM8	2.0294	2.2403×10^{-9}	1.1696×10^{-74}	6.4533×10^{-597}	8.00000
PM	2.0357	2.2049×10^{-9}	1.0295×10^{-74}	2.3266×10^{-597}	8.00000
APM	2.0357	9.1086×10^{-10}	7.2327×10^{-78}	1.3852×10^{-631}	8.13093

wherein $c_{n,8}$ is the asymptotic error constant. Using a symbolic computation and (13), we attain that

$$\frac{2c_2c_3 - c_1c_4}{c_1^4c_2} \sim e_{n-1}. \tag{18}$$

Substituting the value of $(2c_2c_3 - c_1c_4)/(c_1^4c_2)$ from (18) in (17), one may obtain

$$e_{n+1} \sim e_{n-1}e_n^8. \tag{19}$$

Thus, it is easy to obtain

$$e_n^p \sim A^{-1/p}Ce_n^{8+1/p}, \tag{20}$$

wherein A and C are two constants and subsequently

$$p = 8 + \frac{1}{p}, \tag{21}$$

with two solutions $\{4-\sqrt{17}, 4+\sqrt{17}\}$. Clearly the value for $p = 4+\sqrt{17} \approx 8.12311$ is acceptable and would be the convergence R-order of the method (16) with memory. The proof is complete. □

3. Numerical Testing

The objective of this section is to provide a comparison between the presented schemes and the already known methods in the literature.

For numerical reports here, we have used the optimal eighth-order three-step method (1) as (LW8) with $\beta_1 = \beta_2 = 0$, the optimal eighth-order three-step method (3) as (SM8), our optimal three-step eighth-order method (4) with $\beta = -0.0001$, and the accelerated method with memory (16) denoted by (APM) with $\beta_0 = -0.0001$.

The results are summarized in Tables 1 and 2 after some full iterations. As they show, novel schemes are comparable with all of the methods. All numerical instances were performed by Mathematica 8 using 1000 fixed floating point arithmetic [9].

We have computed the root of each test function for the initial guess x_0 while the iterative schemes were stopped when $\lvert f(x_n)\rvert \leq 10^{-150}$. As can be seen, the obtained results in Tables 1 and 2 are in harmony with the analytical procedure given in Section 2.

The computational order of convergence (COC) has also been computed by

$$\text{COC} = \frac{\ln\left|f(x_n)/f(x_{n-1})\right|}{\ln\left|f(x_{n-1})/f(x_{n-2})\right|}. \tag{22}$$

Example 7. In this test, we compare the behavior of different methods for finding the complex solution of the following nonlinear equation:

$$f(x) = (-1 + 2I) + \frac{1}{x} + x + \sin(x), \tag{23}$$

using the initial approximation $x_0 = 1 - 3I$ where $\alpha = 0.28860\cdots - 1.24220\cdots I$. The results for this test are given in Table 1.

Example 8. We here compare the behavior of different methods for finding the solution of

$$g(x) = (-2 + x)\sin(\tanh(x)), \tag{24}$$

using the initial approximation $x_0 = 1.0$ where $\alpha = 2$. The results for this test are given in Table 2.

It should be mentioned that our method (4) cannot be easily extended for nonlinear systems. The reason is that the weight functions used in (4) do not contain a finite difference operator in the denominators. Such an extension might be pursued for future studies. However, a simple extended version of (4) for the N-dimensional case can be written in what follows:

$$\begin{aligned}
\mathbf{y}^{(n)} &= \mathbf{x}^{(n)} - J_{x,w}^{-1}F\left(\mathbf{x}^{(n)}\right), \quad n = 0, 1, 2, \ldots, \\
\mathbf{z}^{(n)} &= \mathbf{y}^{(n)} - \left[J_{x,y}^{-1}J_{x,w} - I\right]\left[J_{x,w}^{-1}F\left(\mathbf{x}^{(n)}\right)\right], \\
\mathbf{x}^{(n+1)} &= \mathbf{z}^{(n)} - J_{y,z}^{-1}F\left(\mathbf{z}^{(n)}\right),
\end{aligned} \tag{25}$$

Table 3: Results of comparisons for Example 9.

Methods	$\|f(x_1)\|$	$\|f(x_2)\|$	$\|f(x_3)\|$	$\|f(x_4)\|$	COC
PMS	23.6907	2.05639×10^{-8}	6.73602×10^{-48}	2.554×10^{-245}	4.99994

wherein $\mathbf{w}^{(n)} = \mathbf{x}^{(n)} + F(\mathbf{x}^{(n)})$ and it possesses only fifth order of convergence. Note that the extended version of Steffensen's method has been written by

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - J_{x,w}^{-1} F\left(\mathbf{x}^{(n)}\right), \quad n = 0, 1, 2, \ldots, \tag{26}$$

wherein

$$\begin{aligned} J_{x,w} &= J\left(\mathbf{x}^{(n)}, \beta H^{(n)}\right) \\ &= \left(F\left(\mathbf{x}^{(n)} + H^{(n)} e^1\right) - F\left(\mathbf{x}^{(n)}\right), \ldots, \right. \\ &\qquad \left. F\left(\mathbf{x}^{(n)} + H^{(n)} e^N\right) - F\left(\mathbf{x}^{(n)}\right)\right) {H^{(n)}}^{-1}, \end{aligned} \tag{27}$$

with $H^{(n)} = \mathrm{diag}(\beta f_1(\mathbf{x}^{(n)}), \ldots, \beta f_N(\mathbf{x}^{(n)}))$. Now we apply (25) to solve a nonlinear integral equation, and keeping the rate of convergence at eight will remain as an open problem for future works.

Example 9. Consider the mixed Hammerstein integral equation [10]:

$$x(s) = 1 + \frac{1}{5} \int_0^1 G(s,t)\, x(t)^3 dt, \tag{28}$$

where $x \in C[0,1]$, $s, t \in [0,1]$, and the kernel G is given by

$$G(s,t) = \begin{cases} (1-s)\, t, & t \le s, \\ s\,(1-t), & t > s. \end{cases} \tag{29}$$

In order to solve this nonlinear integral equation, we transform the above equation into a finite-dimensional problem by using Gauss-Legendre quadrature formula given as

$$\int_0^1 f(t)\, dt \approx \sum_{j=1}^{t} w_j f\left(t_j\right), \tag{30}$$

where the abscissas t_j and the weights w_j are determined for $t = 10$ by Gauss-Legendre quadrature formula. Denoting the approximation of $x(t_i)$ by x_i $(i = 1, 2, \cdots, t)$, we obtain the system of nonlinear equations

$$f(x_1, \ldots, x_t) = 5x_i - 5 - \sum_{j=1}^{t} a_{ij} x_j^3 = 0, \tag{31}$$

where, for $i = 1, 2, \ldots, t$, we have

$$a_{ij} = \begin{cases} w_j t_j \left(1 - t_i\right), & \text{if } j \le i, \\ w_j t_i \left(1 - t_j\right), & \text{if } i < j, \end{cases} \tag{32}$$

wherein the abscissas t_j and the weights w_j are known.

Using the initial approximation $\mathbf{x}^{(0)} = (0.5, \ldots, 0.5)^T$, we apply the proposed method (25) denoted by PMS with $\beta = 0.001$ which is multiplication-rich to find the final solution vector of the nonlinear integral equation (31). Table 3 shows the residuals in l_2 norm, when $t = 10$ is the size of the nonlinear system of equations.

4. Concluding Remarks

Solving nonlinear equations is a classical problem which has interesting applications in various branches of science and engineering (see, e.g., [11]). In this study, we have described an iterative method without memory to find a simple root α of a nonlinear equation $f(x) = 0$ on an open interval D.

The derived scheme was developed by applying the conjecture of Cordero-Torregrosa and it was proved that it converges to the simple zero of a nonlinear equation with optimal eighth order of convergence. This shows that it has the optimal efficiency index 1.682. We, furthermore, discussed how to increase the R-order of convergence via with memorization. Some examples have also been included to support the theoretical parts.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Authors' Contribution

The authors have made the same contribution. All authors read and approved the final paper.

References

[1] A. Cordero and J. R. Torregrosa, "Low-complexity root-finding iteration functions with no derivatives of any order of convergence," *Journal of Computational and Applied Mathematics*, 2014.

[2] H. T. Kung and J. F. Traub, "Optimal order of one-point and multipoint iteration," *Journal of the Association for Computing Machinery*, vol. 21, pp. 643–651, 1974.

[3] A. Iliev and N. Kyurkchiev, *Nontrivial Methods in Numerical Analysis: Selected Topics in Numerical Analysis*, LAP LAMBERT Academic Publishing, 2010.

[4] L. Liu and X. Wang, "Eighth-order methods with high efficiency index for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 215, no. 9, pp. 3449–3454, 2010.

[5] F. Soleymani, S. Karimi Vanani, H. I. Siyyam, and I. A. Al-Subaihi, "Numerical solution of nonlinear equations by an optimal eighth-order class of iterative methods," *Annali dell'Universita di Ferrara*, vol. 59, no. 1, pp. 159–171, 2013.

[6] V. Hristov, A. Iliev, and N. Kyurkchiev, "A note on the convergence of nonstationary finite-difference analogues," *Computational Mathematics and Mathematical Physics*, vol. 45, pp. 194–201, 2005.

[7] T. Lotfi, A. Cordero, J. R. Torregrosa, M. A. Abadi, and M. M. Zadeh, "On generalization based on Bi et al. iterative methods with eighth-order convergence for solving nonlinear equations," *The Scientific World Journal*, vol. 2014, Article ID 272949, 8 pages, 2014.

[8] J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, New Jersey, NJ, USA, 1964.

[9] R. Hazrat, *Mathematica: A Problem-Centered Approach*, Springer, Berlin, Germany, 2010.

[10] F. Soleymani, M. Sharifi, S. Shateyi, and F. K. Haghani1, "A class of Steffensen-type iterative methods for nonlinear systems," *Journal of Applied Mathematics*, vol. 2014, Article ID 705375, 9 pages, 2014.

[11] N. Kyurkchiev and A. Iliev, "On some multipoint methods arising from optimal in the sense of Kung-Traub algorithms," *International Journal on Mathematical Methods and Models in Biosciences*, vol. 2, no. 1, Article ID 1305155, 2013.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 581987, 13 pages
http://dx.doi.org/10.1155/2014/581987

Research Article

A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations

S. S. Motsa, V. M. Magagula, and P. Sibanda

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to S. S. Motsa; sandilemotsa@gmail.com

Received 15 July 2014; Accepted 12 August 2014; Published 27 August 2014

Academic Editor: Hassan Saberi Nik

This paper presents a new method for solving higher order nonlinear evolution partial differential equations (NPDEs). The method combines quasilinearisation, the Chebyshev spectral collocation method, and bivariate Lagrange interpolation. In this paper, we use the method to solve several nonlinear evolution equations, such as the modified KdV-Burgers equation, highly nonlinear modified KdV equation, Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation, and the Fitzhugh-Nagumo equation. The results are compared with known exact analytical solutions from literature to confirm accuracy, convergence, and effectiveness of the method. There is congruence between the numerical results and the exact solutions to a high order of accuracy. Tables were generated to present the order of accuracy of the method; convergence graphs to verify convergence of the method and error graphs are presented to show the excellent agreement between the results from this study and the known results from literature.

1. Introduction

Nonlinearity exists everywhere and, in general, nature is nonlinear. Nonlinear evolution partial differential equations arise in many fields of science, particularly in physics, engineering, chemistry, finance, and biological systems. They are widely used to describe complex phenomena in various fields of sciences, such as wave propagation phenomena, fluid mechanics, plasma physics, quantum mechanics, nonlinear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, financial industry, and numerous areas of mathematical modeling. The development of both numerical and analytical methods for solving complicated, highly nonlinear evolution partial differential equations continues to be an area of interest to scientists whose research aim is to enrich deep understanding of such alluring nonlinear problems.

Innumerable number of methods for obtaining analytical and approximate solutions to nonlinear evolution equations have been proposed. Some of the analytical methods that have been used to solve evolution nonlinear partial differential equations include Adomian's decomposition method [1–3], homotopy analysis method [4–7], tanh-function method [8–10], Haar wavelet method [11–13], and Exp-function method [14–16]. Several numerical methods have been used to solve nonlinear evolution partial differential equations. These include the explicit-implicit method [17], Chebyshev finite difference methods [18], finite difference methods [19], finite element methods [20], and pseudospectral methods [21, 22].

Some drawbacks of approximate analytical methods include slow convergence, particularly for large time ($t > 1$). They may also be cumbersome to use as some involve manual integration of approximate series solutions and, hence, it is difficult to find closed solutions sometimes. On the other hand, some numerical methods may not work in some cases, for example, when the required solution has to be found near a singularity. Certain numerical methods, for example, finite differences require many grid points to achieve good accuracy and, hence, require a lot of computer memory and computational time. Conventional first-order finite difference methods may result in monotonic and stable solutions, but they are strongly dissipative causing the solution of the strongly convective partial differential equations to become smeared out and often grossly inaccurate. On the other hand,

higher order difference methods are less dissipative but are prone to numerical instabilities.

Spectral methods have been used successfully in many different fields in sciences and engineering because of their ability to give accurate solutions of differential equations. Khater et al. [23] applied the Chebyshev spectral collocation method to solve Burgers type of equations in space and finite differences to approximate the time derivative. The Chebyshev spectral collocation method has been used together with the fourth-order Runge-Kutta method to solve the nonlinear PDEs in this study. The Chebyshev spectral collocation is first applied to the NPDE and this yields a system of ordinary differential equations, which are solved using the fourth-order Runge-Kutta method. Olmos and Shizgal [24], Javidi [25, 26], Dehghan and Fakhar-Izadi [27], Driscoll [28], and Driscoll [28] solved the Fisher, Burgers-Fisher, Burgers-Huxley, Fitzhugh-Nagumo, and KdV equations, respectively, using a combination of the Chebyshev spectral collocation method and fourth-order Runge-Kutta method. Darvishi et al. [29, 30] solved the KdV and the Burgers-Huxley equations using a combination of the Chebyshev spectral collocation method and Darvishi's preconditioning. Jacobs and Harley [31] and Tohidi and Kilicman [32] used spectral collocation directly for solving linear partial differential equations. Accuracy will be compromised if they implement their approach in solving nonlinear partial differential equations since they use Kronecker multiplication.

Chebyshev spectral methods are defined everywhere in the computational domain. Therefore, it is easy to get an accurate value of the function under consideration at any point of the domain, beside the collocation points. This property is often exploited, in particular to get a significant graphic representation of the solution, making the possible oscillations due to a wrong approximation of the derivative apparent. Spectral collocation methods are easy to implement and are adaptable to various problems, including variable coefficient and nonlinear differential equations. The error associated with the Chebyshev approximation is $\mathcal{O}(1/N^r)$ where N refers to the truncation and r is connected to the number of continuous derivatives of the function. The interest in using Chebyshev spectral methods in solving nonlinear PDEs stems from the fact that these methods require less grid points to achieve accurate results. They are computational and efficient compared to traditional methods like finite difference and finite element methods. Chebyshev spectral collocation method has been used in conjunction with additional methods which may have their own drawbacks. Here, we provide an alternative method that is not dependent on another method to approximate the solution.

The main objective of this work is to introduce a new method that uses Chebyshev spectral collocation, bivariate Lagrange interpolation polynomials together with quasilinearisation techniques. The nonlinear evolution equations are first linearized using the quasilinearisation method. The Chebyshev spectral collocation method with Lagrange interpolation polynomials are applied independently in space and time variables of the linearized evolution partial differential equation. This new method is termed bivariate interpolated spectral quasilinearisation method (BI-SQLM). We present the BI-SQLM algorithm in a general setting, where it can be used to solve any rth order nonlinear evolution equations. The applicability, accuracy, and reliability of the proposed BI-SQLM are confirmed by solving the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Cahn-Hillard equation,the fourth-order KdV equation, Fisher's, Burgers-Fisher, Burger-Huxley, and the Fitzhugh-Nagumo equations. The results of the BI-SQLM are compared against known exact solutions that have been reported in the scientific literature. It is observed that the method achieves high accuracy with relatively fewer spatial grid points. It also converges fast to the exact solution and approximates the solution of the problem in a computationally efficient manner with simulations completed in fractions of a second in all cases. Tables are generated to show the order of accuracy of the method and time taken to compute the solutions. It is observed that, as the number of grid points is increased, the error decreases. Error graphs and graphs showing the excellent agreement of the exact and analytical solutions for all the nonlinear evolution equations are also presented.

The paper is organized as follows. In Section 2, we introduce the BI-SQLM algorithm for a general nonlinear evolution PDE. In Section 3, we describe the application of the BI-SQLM to selected test problems. The numerical simulations and results are presented in Section 4. Finally, we conclude in Section 5.

2. Bivariate Interpolated Spectral Quasilinearization Method (BI-SQLM)

In this section, we introduce the *Bivariate Interpolated Spectral Quasilinearization Method* (BI-SQLM) for finding solutions to nonlinear evolution PDEs. Without loss of generality, we consider nonlinear PDEs of the form

$$\frac{\partial u}{\partial \tau} = H\left(u, \frac{\partial u}{\partial \eta}, \frac{\partial^2 u}{\partial \eta^2}, \ldots, \frac{\partial^n u}{\partial \eta^n}\right), \quad \text{with the physical region } \tau \in [0, T], \quad \eta \in [a, b], \tag{1}$$

where n is the order of differentiation, $u(\eta, \tau)$ is the required solution, and H is a nonlinear operator which contains all the spatial derivatives of u. The given physical region, $\tau \in [0, T]$, is converted to the region $t \in [-1, 1]$ using the linear transformation $\tau = T(t + 1)/2$ and $\eta \in [a, b]$ is converted to the region $x \in [-1, 1]$ using the linear transformation

$$\eta = \frac{1}{2}(b - a)\, x + \frac{1}{2}(b + a). \tag{2}$$

Equation (1) can be expressed as

$$\frac{\partial u}{\partial t} = H\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \ldots, \frac{\partial^n u}{\partial x^n}\right), \quad t \in [-1, 1], \; x \in [-1, 1]. \tag{3}$$

The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u(x,t) \approx \sum_{i=0}^{N_x}\sum_{j=0}^{N_t} u\left(x_i,t_j\right) L_i(x)\, L_j(t), \tag{4}$$

which interpolates $u(x,t)$ at selected points in both the x and t directions defined by

$$\{x_i\} = \left\{\cos\left(\frac{\pi i}{N_x}\right)\right\}_{i=0}^{N_x}, \qquad \left\{t_j\right\} = \left\{\cos\left(\frac{\pi j}{N_t}\right)\right\}_{j=0}^{N_t}. \tag{5}$$

The choice of the Chebyshev-Gauss-Lobatto grid points (5) ensures that there is a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. The functions $L_i(x)$ are the characteristic Lagrange cardinal polynomials

$$L_i(x) = \prod_{\substack{i=0\\ i\neq k}}^{N_x} \frac{x-x_k}{x_i-x_k}, \tag{6}$$

where

$$L_i\left(x_k\right) = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k. \end{cases} \tag{7}$$

The function $L_j(t)$ is defined in a similar manner. Before linearizing (3), it is convenient to split H into its linear and nonlinear components and rewrite the governing equation in the form

$$F\left[u,u',\ldots,u^{(n)}\right] + G\left[u,u',\ldots,u^{(n)}\right] - \dot{u} = 0, \tag{8}$$

where the dot and primes denote the time and space derivatives, respectively, F is a linear operator, and G is a nonlinear operator. Assuming that the difference $u_{r+1} - u_r$ and all it's space derivative is small, we first approximate the nonlinear operator G using the linear terms of the Taylor series and, hence,

$$\begin{aligned} G\left[u,u',\ldots,u^{(n)}\right] &\approx G\left[u_r,u_r',\ldots,u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \frac{\partial G}{\partial u^{(k)}}\left(u_{r+1}^{(k)} - u_r^{(k)}\right), \end{aligned} \tag{9}$$

where r and $r+1$ denote previous and current iterations, respectively. We remark that this quasilinearization method (QLM) approach is a generalisation of the Newton-Raphson method and was first proposed by Bellman and Kalaba [33] for solving nonlinear boundary value problems.

Equation (9) can be expressed as

$$\begin{aligned} G\left[u,u',\ldots,u^{(n)}\right] &\approx G\left[u_r,u_r',\ldots,u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] u_{r+1}^{(k)} \\ &\quad - \sum_{k=0}^{n} \phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] u_r^{(k)}, \end{aligned} \tag{10}$$

where

$$\phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] = \frac{\partial G}{\partial u^{(k)}}\left[u_r,u_r',\ldots,u_r^{(n)}\right]. \tag{11}$$

Substituting (10) into (8), we get

$$\begin{aligned} &F\left[u_{r+1},u_{r+1}',\ldots,u_{r+1}^{(n)}\right] + \sum_{k=0}^{n} \phi_{k,r} u_{r+1}^{(k)} - \dot{u}_{r+1} \\ &\quad = R_r\left[u_r,u_r',\ldots,u_r^{(n)}\right], \end{aligned} \tag{12}$$

where

$$R_r\left[u_r,u_r',\ldots,u_r^{(n)}\right] = \sum_{k=0}^{n} \phi_{k,r} u_r^{(k)} - G\left[u_r,u_r',\ldots,u_r^{(n)}\right]. \tag{13}$$

A crucial step in the implementation of the solution procedure is the evaluation of the time derivative at the grid points t_j $(j = 0,1,\ldots,N_t)$ and the spatial derivatives at the grid points x_i $(i = 0,1,\ldots,N_x)$. The values of the time derivatives at the Chebyshev-Gauss-Lobatto points (x_i,t_j) are computed as (for $j = 0,1,2,\ldots,N_t$)

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_{x=x_i,t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u\left(x_p,t_k\right) L_p(x_i) \frac{dL_k\left(t_j\right)}{dt} \\ &= \sum_{k=0}^{N_t} u\left(x_i,t_k\right) d_{jk} = \sum_{k=0}^{N_t} d_{jk}\, u\left(x_i,t_k\right), \end{aligned} \tag{14}$$

where $d_{jk} = dL_k(t_j)/dt$ is the standard first derivative Chebyshev differentiation matrix of size $(N_t+1)\times(N_t+1)$ as defined in [34]. The values of the space derivatives at the Chebyshev-Gauss-Lobatto points (x_i,t_j) $(i = 0,1,2,\ldots,N_x)$ are computed as

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x=x_i,t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u\left(x_p,t_k\right) \frac{dL_p\left(x_i\right)}{dx} L_k\left(t_j\right) \\ &= \sum_{p=0}^{N_x} u\left(x_p,t_j\right) D_{ip} = \sum_{p=0}^{N_x} D_{ip}\, u\left(x_p,t_j\right), \end{aligned} \tag{15}$$

where $D_{ip} = dL_p(x_i)/dx$ is the standard first derivative Chebyshev differentiation matrix of size $(N_x+1)\times(N_x+1)$. Similarly, for an nth order derivative, we have

$$\begin{aligned} \left.\frac{\partial^n u}{\partial x^n}\right|_{x=x_i,t=t_j} &= \sum_{p=0}^{N_x} D_{ip}^n u\left(x_p,t_j\right) = \mathbf{D}^n \mathbf{U}_j, \\ &\qquad i = 0,1,2,\ldots,N_x, \end{aligned} \tag{16}$$

where the vector $\mathbf{U}_j$ is defined as

$$\mathbf{U}_j = \left[u_j(x_0), u_j(x_1), \ldots, u_j\left(x_{N_x}\right)\right]^T \tag{17}$$

and the superscript T denotes matrix transpose. Substituting (16) into (12) we get

$$F\left[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \ldots, \mathbf{U}^{(n)}_{r+1,j}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t} d_{jk} \mathbf{U}_{r+1,k} = R_r\left[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \ldots, \mathbf{U}^{(n)}_{r,j}\right] \tag{18}$$

for $j = 0, 1, 2, \ldots, N_t$, where

$$\mathbf{U}^{(n)}_{r+1,j} = \mathbf{D}^n \mathbf{U}_{r+1,j},$$
$$\mathbf{\Phi}_{k,r} = \begin{bmatrix} \phi_{k,r}\left(x_0, t_j\right) & & & \\ & \phi_{k,r}\left(x_1, t_j\right) & & \\ & & \ddots & \\ & & & \phi_{k,r}\left(x_{N_x}, t_j\right) \end{bmatrix}. \tag{19}$$

The initial condition for (3) corresponds to $\tau_{N_t} = -1$ and, hence, we express (18) as

$$F\left[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \ldots, \mathbf{U}^{(n)}_{r+1,j}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t-1} d_{jk} \mathbf{U}_{r+1,k} = \mathbf{R}_j, \tag{20}$$

where

$$\mathbf{R}_j = R_r\left[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \ldots, \mathbf{U}^{(n)}_{r,j}\right] + d_{jN_t} \mathbf{U}_{N_t}, \quad j = 0, 1, 2, \ldots, N_t - 1. \tag{21}$$

Equation (20) can be expressed as the following $N_t(N_x + 1) \times N_t(N_x + 1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \tag{22}$$

where

$$A_{i,i} = F\left[\mathbf{I}, \mathbf{D}, \ldots, \mathbf{D}^{(n)}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{D}^{(k)} - d_{i,i}\mathbf{I}, \qquad A_{i,j} = -d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \tag{23}$$

and $\mathbf{I}$ is the identity matrix of size $(N_x + 1) \times (N_x + 1)$. Solving (19) gives $u(x_i, t_j)$ and, hence, we use (4) to approximate $u(x, t)$.

3. Numerical Experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (3) with exact solutions. In order to determine the level of accuracy of the BI-SQLM approximate solution, at a particular time level, in comparison with the exact solution, we report maximum error which is defined by

$$E_N = \max_r \left\{\left|u\left(x_r, t\right) - \widetilde{u}\left(x_r, t\right)\right|, : 0 \leq r \leq N\right\}, \tag{24}$$

where $\widetilde{u}(x_r, t)$ is the approximate solution and is the $u(x_r, t)$ exact solution at the time level t.

Example 1. We consider the generalized Burgers-Fisher equation [35]:

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u\left(1 - u^{\delta}\right), \tag{25}$$

with initial condition

$$u(x, 0) = \left\{\frac{1}{2} + \frac{1}{2} \tanh\left(\frac{-\alpha\delta}{2(\delta + 1)} x\right)\right\}^{1/\delta} \tag{26}$$

and exact solution

$$u(x, t) = \left\{\frac{1}{2} + \frac{1}{2} \tanh\left(\frac{-\alpha\delta}{2(\delta + 1)} \times \left[x - \left(\frac{\alpha}{\delta + 1} + \frac{\beta(\delta + 1)}{\alpha}\right) t\right]\right)\right\}^{1/\delta}, \tag{27}$$

where α, β, and δ are parameters. For illustration purposes, these parameters are chosen to be $\alpha = \beta = \delta = 1$ in this paper. The linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' + u, \qquad G(u) = -uu' - u^2. \tag{28}$$

We first linearize the nonlinear operator G. We approximate G using the equation

$$G \approx G\left[u_r, u'_r, u''_r\right] + \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_{r+1} - \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_r. \tag{29}$$

The coefficients are given by

$$\phi_{0,r} = \frac{\partial G}{\partial u}\left[u_r, u'_r, u''_r\right] = -\left(u'_r + 2u_r\right), \quad \phi_{1,r} = \frac{\partial G}{\partial u'}\left[u_r, u'_r, u''_r\right] = -u_r, \quad \phi_{2,r} = \frac{\partial G}{\partial u''}\left[u_r, u'_r, u''_r\right] = 0, \quad R_r = \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_r - G\left[u_r, u'_r, u''_r\right] = -u_r^2 - u_r u'_r. \tag{30}$$

Therefore, the linearized equation can be expressed as

$$u''_{r+1} + \phi_{1,r}u'_{r+1} + \phi_{0,r}u_{r+1} + u_{r+1} - \dot{u} = R_r. \tag{31}$$

Applying the spectral method both in x and t and initial condition, we get

$$\begin{aligned}\mathbf{D}^2\mathbf{U}_{r+1,i} + \mathbf{\Phi}_{1,r}\mathbf{D}\mathbf{U}_{r+1,i} + \mathbf{\Phi}_{0,r}\mathbf{U}_{r+1,i} \\ + \mathbf{U}_{r+1,i} - 2\sum_{j=0}^{N_t-1} d_{ij}\mathbf{U}_{r+1,j} = \mathbf{R}_i.\end{aligned} \tag{32}$$

Equation (32) can be expressed as

$$\begin{aligned}&\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix},\end{aligned} \tag{33}$$

where

$$\begin{aligned}A_{i,i} &= \mathbf{D}^2 + \mathbf{\Phi}_{1,r}^{(i)}\mathbf{D} + \mathbf{\Phi}_{0,r}^{(i)} + \left(1 - 2d_{i,i}\right)\mathbf{I}, \\ A_{i,j} &= -2d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \\ \mathbf{R}_i &= R_r + 2d_{iN_t}\mathbf{U}_{r,N_t}.\end{aligned} \tag{34}$$

The boundary conditions are implemented in the first and last row of the matrices A_{ij} and the column vectors $\mathbf{R}_i$ for $i = 0, 1, \ldots, N_t - 1$ and $j = 0, 1, \ldots, N_t - 1$. The procedure for finding the variable coefficients ϕ_i and matrices for the remaining examples is similar.

Example 2. We consider Fisher's equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u\left(1 - u\right), \tag{35}$$

subject to the initial condition

$$u\left(x, 0\right) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x}\right)^2} \tag{36}$$

and exact solution [36]

$$u\left(x, t\right) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x - 5\alpha t/6}\right)^2}, \tag{37}$$

where α is a constant. The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. For this example, the appropriate linear operator F and nonlinear operator G are chosen as

$$F\left(u\right) = u'' + \alpha u, \qquad G\left(u\right) = -\alpha u^2. \tag{38}$$

Table 1: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	1.986$e-008$	1.119$e-011$	7.398$e-013$	7.171$e-013$
0.2	3.934$e-008$	3.121$e-011$	1.552$e-012$	1.561$e-012$
0.3	5.577$e-008$	4.864$e-011$	1.004$e-012$	1.005$e-012$
0.4	6.997$e-008$	6.802$e-011$	7.895$e-013$	8.124$e-013$
0.5	8.107$e-008$	7.971$e-011$	1.088$e-012$	1.027$e-012$
0.6	8.891$e-008$	8.560$e-011$	8.805$e-013$	7.847$e-013$
0.7	9.344$e-008$	8.953$e-011$	6.418$e-013$	6.463$e-013$
0.8	9.431$e-008$	8.759$e-011$	6.199$e-013$	6.164$e-013$
0.9	9.178$e-008$	8.325$e-011$	3.978$e-013$	3.695$e-013$
1.0	8.787$e-008$	7.421$e-011$	7.988$e-014$	5.596$e-014$
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

Table 2: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	1.142$e-007$	1.369$e-010$	5.891$e-012$	6.143$e-012$
0.2	1.178$e-007$	1.373$e-010$	9.570$e-012$	1.013$e-011$
0.3	1.186$e-007$	1.479$e-010$	1.489$e-011$	1.512$e-011$
0.4	1.069$e-007$	9.450$e-011$	1.703$e-011$	1.702$e-011$
0.5	9.030$e-008$	7.944$e-011$	5.283$e-012$	5.736$e-012$
0.6	6.963$e-008$	6.618$e-011$	1.639$e-011$	1.626$e-011$
0.7	4.638$e-008$	1.579$e-011$	1.362$e-011$	1.364$e-011$
0.8	2.457$e-008$	4.030$e-011$	3.934$e-012$	3.852$e-012$
0.9	2.028$e-008$	6.006$e-011$	4.466$e-012$	4.727$e-012$
1.0	3.147$e-008$	7.708$e-011$	7.757$e-013$	7.261$e-013$
CPU Time (sec)	0.010152	0.015387	0.019163	0.021564

Example 3. Consider the Fitzhugh-Nagumo equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u\left(u - \alpha\right)\left(1 - u\right) \tag{39}$$

with initial condition

$$u\left(x, 0\right) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}}\right)\right]. \tag{40}$$

This equation has the exact solution [37]

$$u\left(x, t\right) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}} + \frac{2\alpha - 1}{4}t\right)\right], \tag{41}$$

where α is a parameter. In this example, the linear operator F and nonlinear operator G are chosen as

$$F\left(u\right) = u'' - \alpha u, \qquad G\left(u\right) = \left(1 + \alpha\right)u^2 - u^3. \tag{42}$$

Table 3: Maximum errors E_N for the Fitzhug-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	5.719$e-007$	1.196$e-009$	2.367$e-012$	9.881$e-014$
0.2	6.193$e-007$	1.299$e-009$	2.761$e-012$	3.952$e-014$
0.3	6.662$e-007$	1.463$e-009$	3.259$e-012$	8.216$e-014$
0.4	6.779$e-007$	1.448$e-009$	3.341$e-012$	8.094$e-014$
0.5	6.920$e-007$	1.526$e-009$	3.587$e-012$	5.063$e-014$
0.6	7.019$e-007$	1.573$e-009$	3.729$e-012$	3.775$e-014$
0.7	6.933$e-007$	1.516$e-009$	3.660$e-012$	8.915$e-014$
0.8	6.828$e-007$	81.535$e-009$	3.635$e-012$	7.594$e-014$
0.9	6.765$e-007$	1.528$e-009$	3.519$e-012$	3.242$e-013$
1.0	6.687$e-007$	1.490$e-009$	3.405$e-012$	1.688$e-013$
CPU time (sec)	0.024281	0.024901	0.026810	0.032389

Table 4: Maximum errors E_N for the Burger-Huxley equation when $\gamma = 0.75$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	2.217$e-006$	8.482$e-009$	2.166$e-011$	7.822$e-014$
0.2	2.596$e-006$	9.369$e-009$	2.536$e-011$	1.184$e-013$
0.3	2.859$e-006$	1.073$e-008$	3.201$e-011$	1.049$e-013$
0.4	3.001$e-006$	1.112$e-008$	3.652$e-011$	9.426$e-014$
0.5	3.137$e-006$	1.213$e-008$	4.262$e-011$	1.510$e-013$
0.6	3.270$e-006$	1.311$e-008$	4.842$e-011$	2.127$e-013$
0.7	3.367$e-006$	1.359$e-008$	5.289$e-011$	1.230$e-013$
0.8	3.467$e-006$	1.438$e-008$	5.803$e-011$	1.549$e-013$
0.9	3.562$e-006$	1.504$e-008$	6.260$e-011$	3.063$e-013$
1.0	3.640$e-006$	1.559$e-008$	6.674$e-011$	2.951$e-013$
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

Table 5: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	1.803$e-007$	3.419$e-010$	4.449$e-013$	1.572$e-013$
0.2	2.614$e-007$	4.347$e-010$	5.049$e-013$	5.992$e-014$
0.3	2.717$e-007$	4.677$e-010$	5.532$e-013$	8.128$e-013$
0.4	2.009$e-007$	3.663$e-010$	4.771$e-013$	6.158$e-013$
0.5	2.580$e-007$	4.410$e-010$	7.518$e-013$	2.555$e-013$
0.6	2.653$e-007$	4.606$e-010$	8.738$e-013$	5.756$e-013$
0.7	2.248$e-007$	4.039$e-010$	6.210$e-013$	2.393$e-013$
0.8	2.572$e-007$	4.476$e-010$	5.432$e-013$	6.812$e-013$
0.9	2.436$e-007$	4.351$e-010$	6.111$e-013$	6.287$e-013$
1.0	8.275$e-008$	3.721$e-010$	7.569$e-013$	1.087$e-007$
CPU time (sec)	0.015646	0.021226	0.030159	0.035675

Table 6: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	7.788$e-005$	3.553$e-007$	7.601$e-010$	2.080$e-010$
0.2	1.153$e-004$	4.000$e-007$	5.684$e-010$	1.189$e-010$
0.3	1.011$e-004$	3.739$e-007$	4.471$e-010$	4.503$e-010$
0.4	3.926$e-005$	1.785$e-007$	6.544$e-010$	4.987$e-010$
0.5	6.727$e-005$	2.342$e-007$	2.638$e-010$	1.528$e-010$
0.6	6.065$e-005$	2.207$e-007$	4.565$e-010$	4.568$e-010$
0.7	2.511$e-005$	1.105$e-007$	4.749$e-010$	3.748$e-010$
0.8	4.074$e-005$	1.427$e-007$	1.062$e-010$	1.604$e-010$
0.9	2.386$e-005$	1.018$e-007$	2.343$e-010$	8.114$e-011$
1.0	1.440$e-006$	7.256$e-008$	1.436$e-009$	1.513$e-011$
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

Example 4. Consider the Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} u_x = \frac{\partial^2 u}{\partial x^2} + \beta u \left(1 - u^{\delta}\right)\left(u^{\delta} - \gamma\right), \tag{43}$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study), and $\gamma \in (0, 1)$. The exact solution subject to the initial condition

$$u(x, 0) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r - \alpha}x\right], \tag{44}$$

is reported in [38, 39] as

$$u(x, t) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r - \alpha}(x - ct)\right], \tag{45}$$

where

$$r = \sqrt{\alpha^2 + 8\beta}, \qquad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4} \tag{46}$$

The general solution (45) was reported in [40, 41]. In this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \beta\gamma u,$$
$$G(u) = -\alpha u u' + \beta(1 + \gamma)u^2 - \beta u^3. \tag{47}$$

Example 5. We consider the modified KdV-Burgers equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 u}{\partial x^2} - 6u^2\frac{\partial u}{\partial x} \tag{48}$$

subject to the initial condition

$$u(x, 0) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6}\right) \tag{49}$$

and exact solution [42]

$$u(x, t) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6} - \frac{t}{27}\right). \tag{50}$$

Table 7: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	1.119e − 011	7.398e − 013	8.266e − 013	3.808e − 014
0.4	3.121e − 011	1.552e − 012	7.378e − 013	3.780e − 014
0.6	4.864e − 011	1.004e − 012	3.402e − 012	7.283e − 014
0.8	6.802e − 011	7.895e − 013	1.118e − 012	3.714e − 014
1.0	7.971e − 011	1.088e − 012	1.473e − 012	1.691e − 013
1.2	8.560e − 011	8.805e − 013	2.611e − 012	3.119e − 013
1.4	8.953e − 011	6.418e − 013	6.671e − 012	1.796e − 013
1.6	8.759e − 011	6.199e − 013	1.118e − 011	1.097e − 013
1.8	8.325e − 011	3.978e − 013	7.515e − 013	6.273e − 014
2.0	7.421e − 011	7.988e − 014	3.682e − 012	2.311e − 013
CPU time (sec)	0.013542	0.022967	0.023792	0.024758

Table 8: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	1.223e − 007	1.400e − 008	1.402e − 008	1.094e − 012
0.4	1.145e − 007	1.919e − 008	1.918e − 008	3.919e − 012
0.6	9.192e − 008	2.082e − 008	2.085e − 008	1.953e − 012
0.8	2.293e − 008	1.793e − 008	1.793e − 008	6.340e − 013
1.0	2.395e − 008	1.337e − 008	1.339e − 008	2.381e − 012
1.2	5.778e − 008	1.954e − 008	1.930e − 008	1.005e − 011
1.4	6.045e − 008	1.620e − 008	1.620e − 008	3.535e − 012
1.6	5.244e − 008	7.218e − 009	7.345e − 009	5.765e − 012
1.8	4.395e − 008	6.828e − 009	6.784e − 009	3.983e − 012
2.0	2.944e − 008	9.406e − 010	8.820e − 010	3.812e − 012
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

The modified KdV-Burgers equation describes various kinds of phenomena such as a mathematical model of turbulence [43] and the approximate theory of flow through a shock wave traveling in viscous fluid [44]. For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''' - u'', \qquad G(u) = -6u'u^2. \tag{51}$$

Example 6. We consider the high nonlinear modified KdV equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} + \left(\frac{\partial u}{\partial x}\right)^2 - u^2 \tag{52}$$

subject to the initial condition

$$u(x,0) = \frac{1}{2} + \frac{e^{-x}}{4} \tag{53}$$

Table 9: Maximum errors E_N for the Fitzhugh-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	6.326e − 007	1.311e − 009	2.886e − 012	1.131e − 012
0.4	6.721e − 007	1.467e − 009	3.310e − 012	1.564e − 012
0.6	7.140e − 007	1.602e − 009	3.617e − 012	1.936e − 012
0.8	6.730e − 007	1.496e − 009	4.707e − 012	1.196e − 012
1.0	6.660e − 007	1.487e − 009	3.675e − 012	1.264e − 012
1.2	6.449e − 007	1.366e − 009	1.897e − 012	1.727e − 012
1.4	5.690e − 007	1.083e − 009	2.972e − 012	1.200e − 012
1.6	4.931e − 007	8.010e − 010	1.519e − 012	8.590e − 013
1.8	3.986e − 007	4.658e − 010	1.068e − 012	6.790e − 013
2.0	2.904e − 007	2.968e − 010	1.592e − 012	1.770e − 013
CPU time (sec)	0.041048	0.049629	0.055008	0.053863

Table 10: Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.5$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	2.866e − 006	1.119e − 008	3.670e − 011	1.150e − 012
0.4	3.401e − 006	1.420e − 008	5.744e − 011	1.638e − 012
0.6	3.814e − 006	1.687e − 008	7.426e − 011	1.958e − 012
0.8	3.915e − 006	1.729e − 008	8.171e − 011	7.002e − 013
1.0	3.938e − 006	1.738e − 008	8.157e − 011	1.267e − 012
1.2	3.808e − 006	1.624e − 008	7.687e − 011	1.710e − 012
1.4	3.456e − 006	1.527e − 008	6.965e − 011	5.109e − 013
1.6	3.230e − 006	1.349e − 008	5.535e − 011	8.203e − 013
1.8	2.925e − 006	1.078e − 008	3.598e − 011	8.294e − 013
2.0	2.497e − 006	7.505e − 009	2.265e − 011	9.726e − 014
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

and exact solution

$$u(x,t) = \frac{1}{t+2} + \frac{e^{-(x+t)}}{(t+2)^2}. \tag{54}$$

For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''', \qquad G(u) = \left(u'\right)^2 - u^2. \tag{55}$$

4. Results and Discussion

In this section we present the numerical solutions obtained using the BI-SQLM algorithm. The number of collocation points in the space x variable used to generate the results is $N_x = 10$ in all cases. Similarly, the number of collocation points in the time t variable used is $N_t = 10$ in all cases. It was found that sufficient accuracy was achieved using these values in all numerical simulations.

Table 11: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	2.137$e-007$	3.820$e-010$	4.846$e-013$	9.998$e-013$
0.4	2.480$e-007$	4.267$e-010$	5.596$e-013$	8.775$e-013$
0.6	2.691$e-007$	4.676$e-010$	6.565$e-013$	2.054$e-012$
0.8	2.214$e-007$	3.979$e-010$	8.776$e-013$	1.168$e-012$
1.0	2.538$e-007$	4.463$e-010$	9.650$e-013$	8.410$e-013$
1.2	2.650$e-007$	4.680$e-010$	7.450$e-013$	5.113$e-013$
1.4	2.383$e-007$	4.296$e-010$	7.500$e-013$	1.110$e-012$
1.6	2.568$e-007$	4.572$e-010$	9.704$e-013$	2.837$e-013$
1.8	2.520$e-007$	4.529$e-010$	7.443$e-013$	5.353$e-013$
2.0	2.370$e-007$	4.438$e-010$	2.719$e-013$	8.849$e-013$
CPU time (sec)	0.062066	0.081646	0.080718	0.10775

Table 12: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	1.986$e-008$	1.119$e-011$	7.398$e-013$	7.171$e-013$
0.4	8.010$e-005$	3.577$e-007$	3.902$e-008$	1.979$e-010$
0.6	7.235$e-005$	2.549$e-007$	2.016$e-008$	4.899$e-010$
0.8	6.284$e-005$	1.663$e-007$	1.155$e-007$	2.679$e-010$
1.0	1.642$e-005$	1.620$e-007$	1.243$e-007$	2.474$e-010$
1.2	2.753$e-005$	1.073$e-007$	1.073$e-007$	1.679$e-010$
1.4	3.738$e-006$	8.971$e-008$	8.598$e-008$	4.788$e-011$
1.6	1.223$e-005$	2.153$e-008$	2.503$e-008$	2.941$e-011$
1.8	5.836$e-006$	2.986$e-008$	9.127$e-009$	5.177$e-011$
2.0	9.310$e-006$	6.548$e-008$	7.277$e-008$	1.453$e-009$
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

In Tables 1, 2, 3, 4, 5, and 6 we give the maximum errors between the exact and BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at $t \in [0.1, 1]$. The results were computed in the space domain $x \in [0, 1]$. To give a sense of the computational efficiency of the method, the computational time to generate the results is also given. Tables 1–6 clearly show the accuracy of the method. The accuracy is seen to improve with an increase in the number of collocation points N_x. It is remarkable to note that accurate results with errors of order up to 10^{-14} are obtained using very few collocation points in both the x and t variables $N_t \leq 10$, $N_x \leq 10$. This is a clear indication that the BI-SQLM is powerful method that is appropriate in solving nonlinear evolution PDEs. We remark, also, that the BI-SQLM is computationally fast as accurate results are generated in a fraction of a second in all the examples considered in this work.

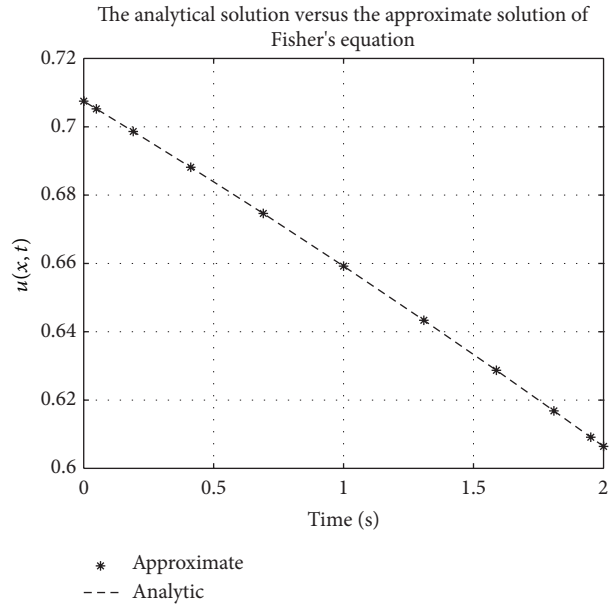


Figure 1: Fishers equation analytical solution graph.

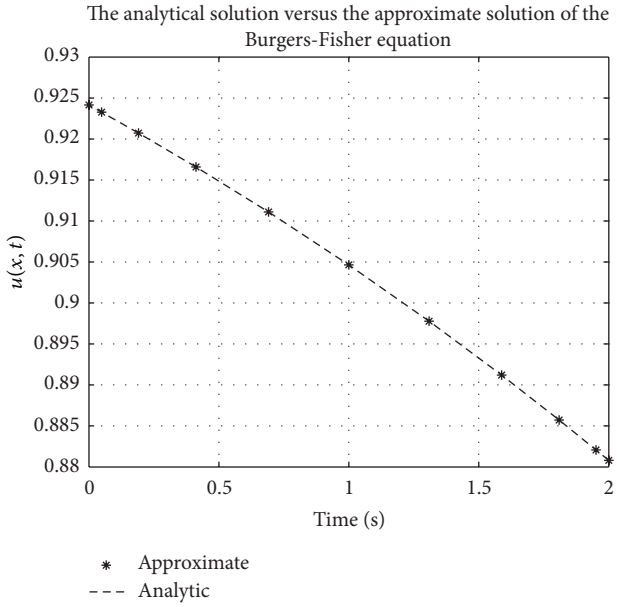


Figure 2: Burger-Fishers equation analytical solution graph.

In Tables 7, 8, 9, 10, 11, and 12 we give the maximum errors of the BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at selected values of $t = 2$ for different collocation points, N_t, in the t-variable. The results in Tables 7–12 were computed on the space domain $x \in [0, 1]$. We note that the accuracy does not detoriate when $t > 1$ for this method as is often the case with numerical schemes such as finite differences.

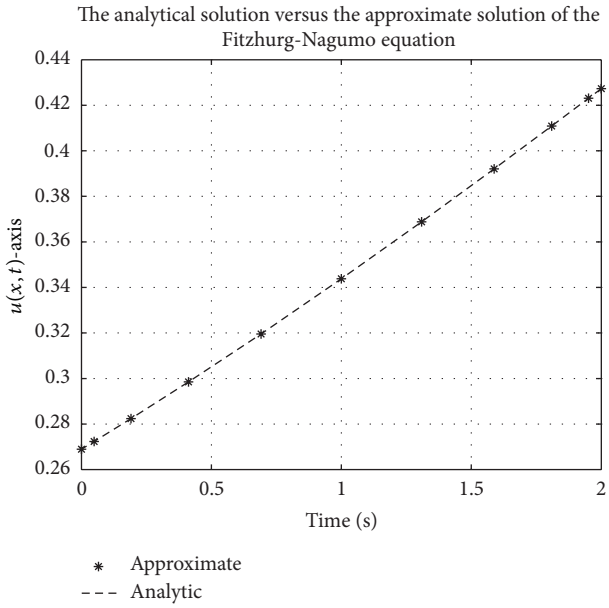


Figure 3: Fitzhugh-Nagumo equation analytical solution graph.

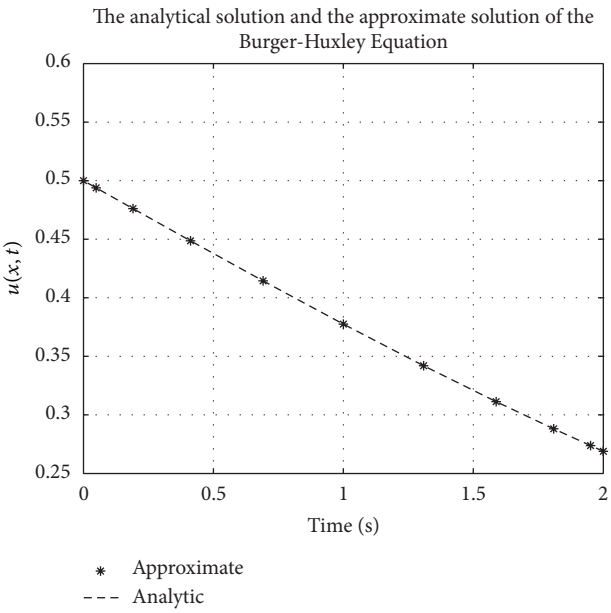


Figure 4: Burgers-Huxley equation analytical solution graph.

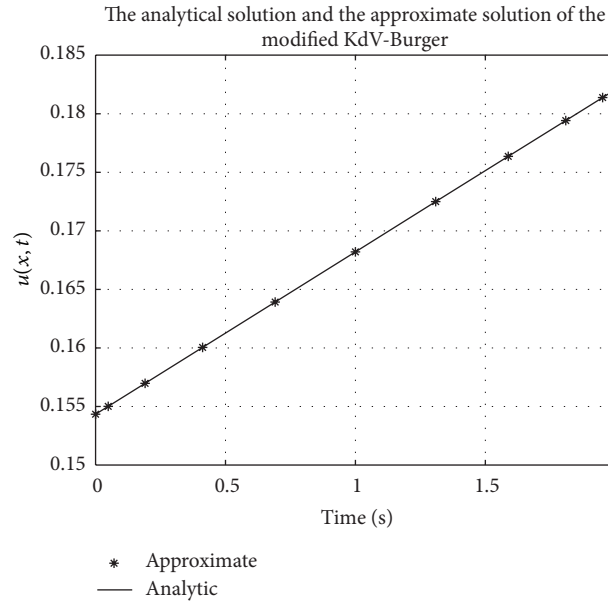


Figure 5: Modified KdV-Burger equation analytical solution graph.

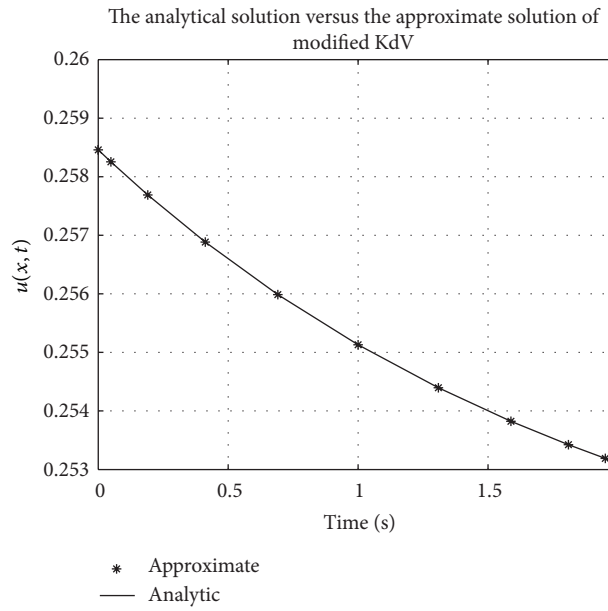


Figure 6: Modified KdV equation analytical solution graph.

Figures 1, 2, 3, 4, 5, and 6 show a comparison of the analytical and approximate solutions of the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$. The approximate solutions are in excellent agreement with the analytical solutions, and this demonstrates the accuracy of the algorithm presented in this study.

In Figures 7, 8, 9, 10, 11, and 12, we present error analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$.

In Figures 13, 14, 15, 16, 17, and 18, convergence analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively. The figures present a variation of the error norm at a fixed value of time ($t = 1$) with iterations of the BI-SQLM scheme. It can be seen that, in almost all the examples considered, the iteration scheme takes about 3 or 4 iterations to converge fully. Beyond the point where full convergence is reached, error norm levels off and does

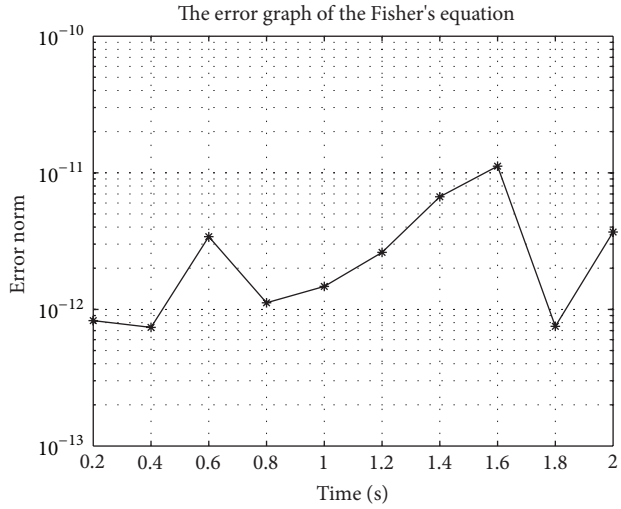


Figure 7: Fishers equation error graph.

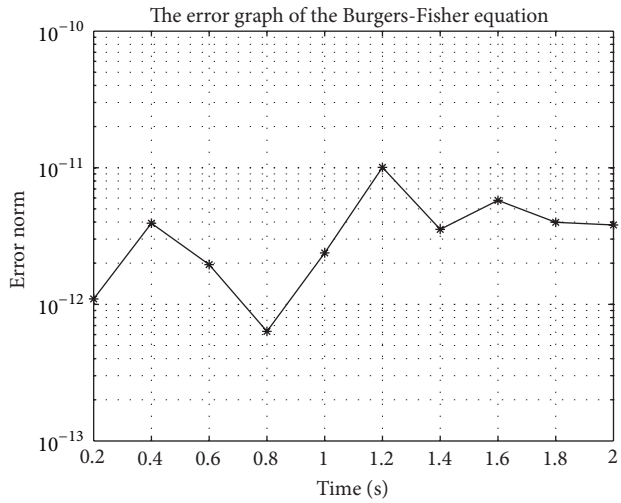


Figure 8: Burger-Fishers equation error graph.

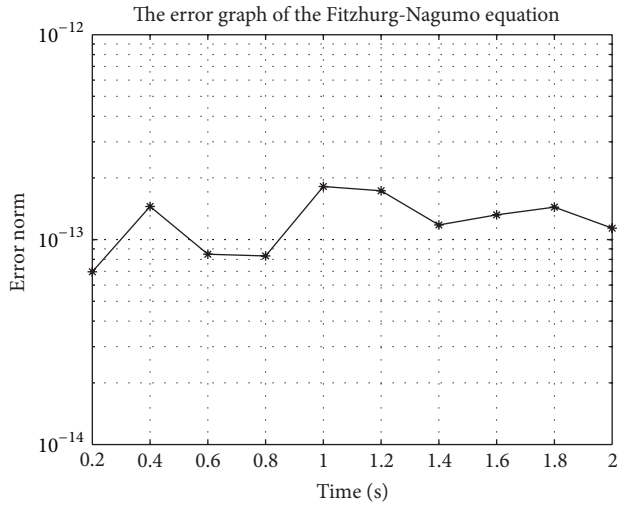


Figure 9: Fitzhugh-Nagumo equation error graph.

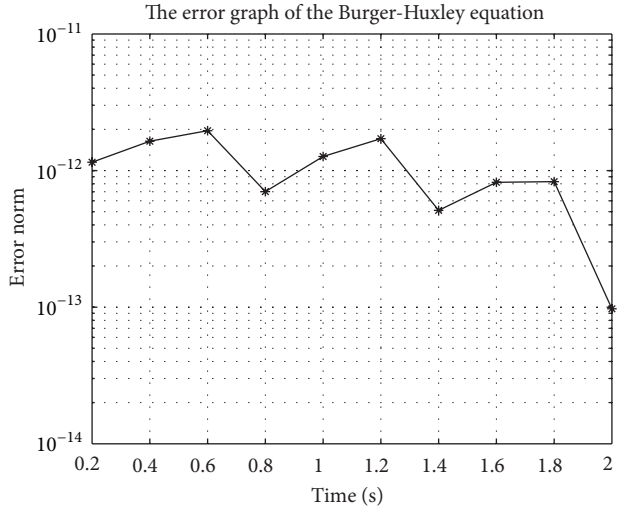


Figure 10: Burgers-Huxley equation error graph.

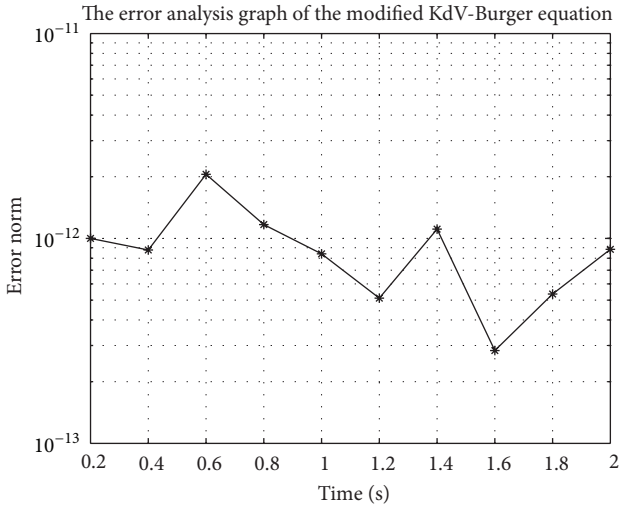


Figure 11: Modified KdV-Burger equation error graph.

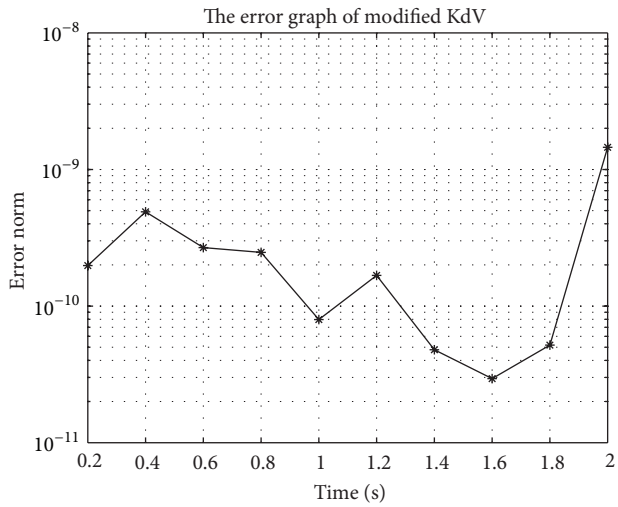


Figure 12: Modified KdV equation error graph.

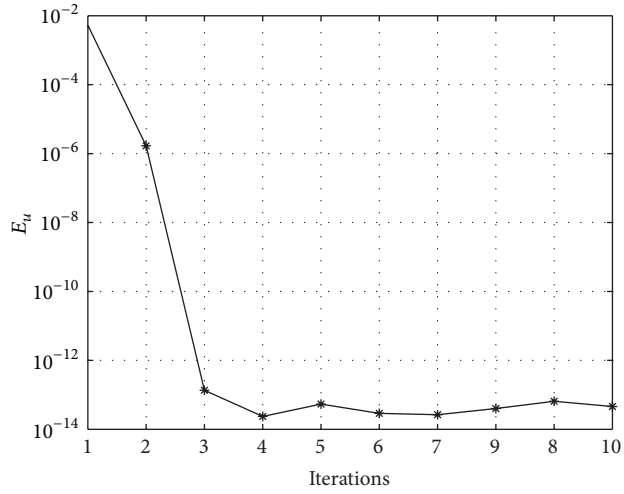


Figure 13: Fishers equation convergence graph.

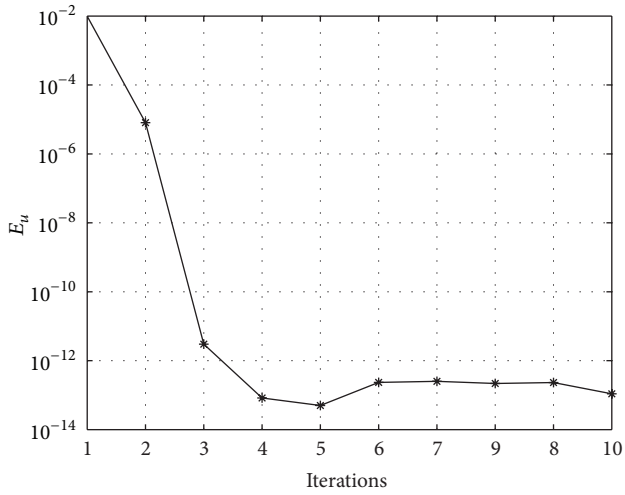


Figure 14: Burger-Fishers equation convergence graph.

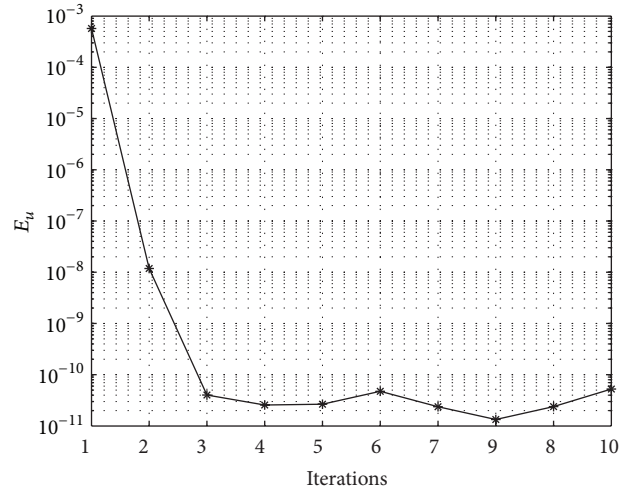


Figure 15: Fitzhugh-Nagumo equation convergence graph.

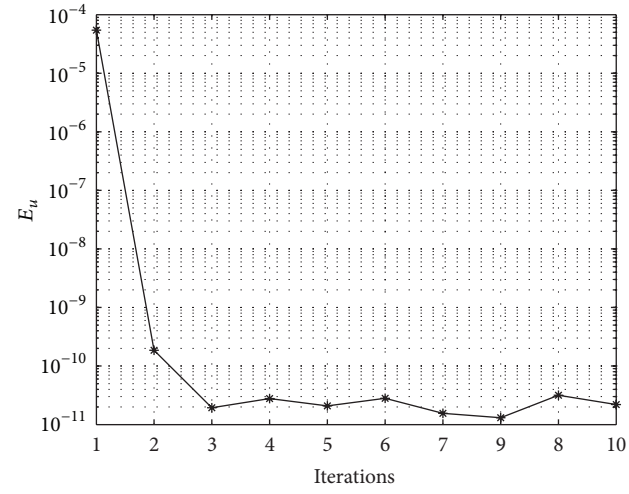


Figure 16: Burgers-Huxley equation convergence graph.

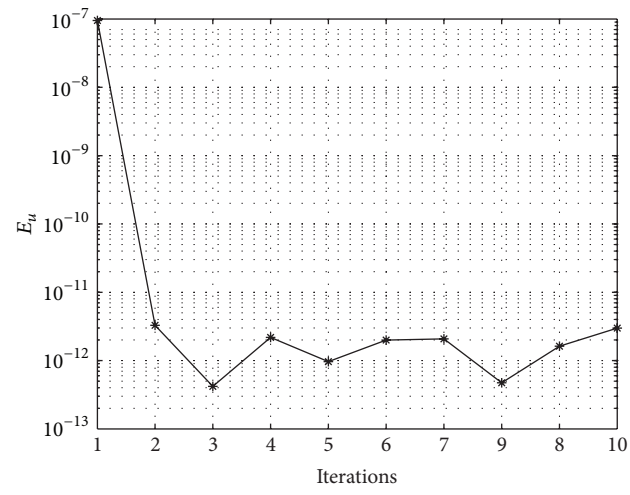


Figure 17: Modified KdV-Burger equation convergence graph.

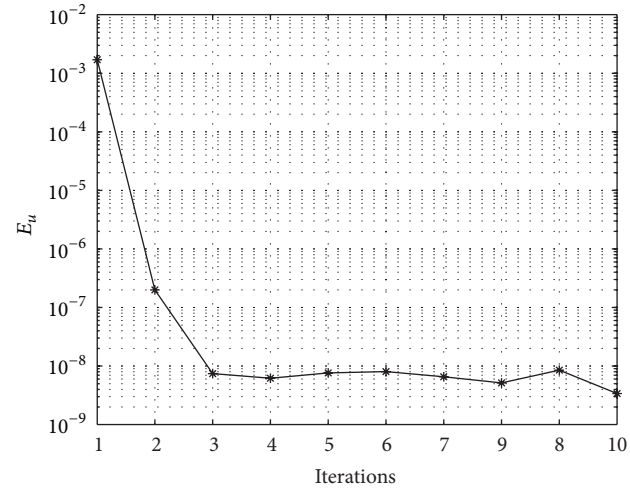


Figure 18: Modified KdV equation convergence graph.

not improve with an increase in the number of iterations. This plateau level gives an estimate of the maximum error that can be achieved when using the proposed method with a certain number of collocation points. It is worth remarking that the accuracy of the method depends on the number of collocation points in both the x and t directions. The results from Figures 13–18 clearly demonstrate that the BI-SQLM is accurate.

5. Conclusion

This paper has presented a new Chebyshev collocation spectral method for solving general nonlinear evolution partial differential equations. The bivariate interpolated spectral quasilinearisation method (BI-SQLM) was developed by combining elements of the quasilinearisation method and Chebyshev spectral collocation with bivariate Lagrange interpolation. The main goal of the current study was to assess the accuracy, robustness, and effectiveness of the method in solving nonlinear partial differential equations.

Numerical simulations were conducted on the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, and Burgers-Huxley equation. It is evident from the study that the BI-SQLM gives accurate results in a computationally efficient manner. Further evidence from this study is that the BI-SQLM gives solutions that are uniformly accurate and valid in large intervals of space and time domains. The apparent success of the method can be attributed to the use of the Chebyshev spectral collocation method with bivariate Lagrange interpolation in space and time for differentiating. This work contributes to the existing body of literature on quasilinearisation tools for solving complex nonlinear partial differential equations. Further work needs to be done to establish whether the BI-SQLM can be equally successful in solving coupled systems of equations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported in part by the National Research Foundation of South Africa (Grant no. 85596).

References

[1] G. Adomian, *Stochastic Systems*, vol. 169 of *Mathematics in Science and Engineering*, Academic Press, Orlando, Fla, USA, 1983.

[2] G. Adomian, "A review of the decomposition method in applied mathematics," *Journal of Mathematical Analysis and Applications*, vol. 135, no. 2, pp. 501–544, 1988.

[3] L. Bougoffa and R. C. Rach, "Solving nonlocal initial-boundary value problems for linear and nonlinear parabolic and hyperbolic partial differential equations by the Adomian decomposition method," *Applied Mathematics and Computation*, vol. 225, pp. 50–61, 2013.

[4] S. J. Liao, *Advances in Homotopy Analysis Method*, World Scientific Publishing, Singapore, 2014.

[5] J. He, "Application of homotopy perturbation method to nonlinear wave equations," *Chaos, Solitons and Fractals*, vol. 26, no. 3, pp. 695–700, 2005.

[6] S. Abbasbandy, "The application of homotopy analysis method to solve a generalized Hirota-Satsuma coupled KdV equation," *Physics Letters A: General, Atomic and Solid State Physics*, vol. 361, no. 6, pp. 478–483, 2007.

[7] L. Song and H. Zhang, "Application of homotopy analysis method to fractional KdV-Burgers-KURamoto equation," *Physics Letters A*, vol. 367, no. 1-2, pp. 88–94, 2007.

[8] E. J. Parkes and B. R. Duffy, "An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations," *Computer Physics Communications*, vol. 98, no. 3, pp. 288–300, 1996.

[9] B. R. Duffy and E. J. Parkes, "Travelling solitary wave solutions to a seventh-order generalized KdV equation," *Physics Letters A*, vol. 214, no. 5-6, pp. 271–272, 1996.

[10] Z. B. Li, "Exact solitary wave solutions of nonlinear evolution equations," in *Mathematics Mechanization and Application*, X. S. Gao and D. M. Wang, Eds., Academic Press, San Diego, Calif, USA, 2000.

[11] U. Lepik, "Numerical solution of evolution equations by the Haar wavelet method," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 695–704, 2007.

[12] I. Celik, "Haar wavelet method for solving generalized Burgers-Huxley equation," *Arab Journal of Mathematical Sciences*, vol. 18, no. 1, pp. 25–37, 2012.

[13] G. Hariharan, K. Kannan, and K. R. Sharma, "Haar wavelet method for solving Fisher's equation," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 284–292, 2009.

[14] J. He and X. Wu, "Exp-function method for nonlinear wave equations," *Chaos, Solitons & Fractals*, vol. 30, no. 3, pp. 700–708, 2006.

[15] C. Chun, "Solitons and periodic solutions for the fifth-order KdV equation with the Exp-function method," *Physics Letters A*, vol. 372, no. 16, pp. 2760–2766, 2008.

[16] X. H. Wu and J. H. He, "EXP-function method and its application to nonlinear equations," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 903–910, 2008.

[17] F. W. Wubs and E. D. de Goede, "An explicit-implicit method for a class of time-dependent partial differential equations," *Applied Numerical Mathematics*, vol. 9, no. 2, pp. 157–181, 1992.

[18] E. M. E. Elbarbary and M. El-Kady, "Chebyshev finite difference approximation for the boundary value problems," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 513–523, 2003.

[19] A. C. Vliegenthart, "On finite-difference methods for the Korteweg-de Vries equation," *Journal of Engineering Mathematics*, vol. 5, pp. 137–155, 1971.

[20] J. Argyris and M. Haase, "An engineer's guide to soliton phenomena: application of the finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 61, no. 1, pp. 71–122, 1987.

[21] G. F. Carey and Y. Shen, "Approximations of the KdV equation by least squares finite elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 93, no. 1, pp. 1–11, 1991.

[22] K. Djidjeli, W. G. Price, P. Temarel, and E. H. Twizell, "A linearized implicit pseudo-spectral method for certain nonlinear water wave equations," *Communications in Numerical*

Methods in Engineering with Biomedical Applications, vol. 14, no. 10, pp. 977–993, 1998.

[23] A. H. Khater, R. S. Temsah, and M. M. Hassan, "A Chebyshev spectral collocation method for solving Burger's-type equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 333–350, 2008.

[24] D. Olmos and B. D. Shizgal, "A pseudospectral method of solution of Fisher's equation," *Journal of Computational and Applied Mathematics*, vol. 193, no. 1, pp. 219–242, 2006.

[25] M. Javidi, "Spectral collocation method for the solution of the generalized Burger-Fisher equation," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 345–352, 2006.

[26] M. Javidi, "A numerical solution of the generalized Burgers-Huxley equation by spectral collocation method," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 338–344, 2006.

[27] M. Dehghan and F. Fakhar-Izadi, "Pseudospectral methods for Nagumo equation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, no. 4, pp. 553–561, 2011.

[28] T. A. Driscoll, "A composite Runge-Kutta method for the spectral solution of semilinear PDEs," *Journal of Computational Physics*, vol. 182, no. 2, pp. 357–367, 2002.

[29] M. T. Darvishi, S. Kheybari, and F. Khani, "A numerical solution of Korteweg-de Vries equation by pseudospectral method using Darvishi's preconditionings," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 98–105, 2006.

[30] M. T. Darvishi, S. Kheybari, and F. Khani, "Spectral collocation method and Darvishi's preconditionings to solve the generalized Burgers-Huxley equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 10, pp. 2091–2103, 2008.

[31] B. A. Jacobs and C. Harley, "Two hybrid methods for solving two-dimensional linear time-fractional partial differential equations," *Abstract and Applied Analysis*, vol. 2014, Article ID 757204, 10 pages, 2014.

[32] E. Tohidi and A. Kilicman, "An efficient spectral approximation for solving several types of parabolic pdes with nonlocal boundary conditions," *Mathematical Problems in Engineering*, vol. 2014, Article ID 369029, 6 pages, 2014.

[33] R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary-Value Problems*, vol. 3 of *Modern Analytic and Computional Methods in Science and Mathematics*, American Elsevier, New York, NY, USA, 1965.

[34] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, Pa, USA, 2000.

[35] A. Golbabai and M. Javidi, "A spectral domain decomposition approach for the generalized Burger's-Fisher equation," *Chaos, Solitons & Fractals*, vol. 39, no. 1, pp. 385–392, 2009.

[36] A. Wazwaz and A. Gorguis, "An analytic study of Fisher's equation by using Adomian decomposition method," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 609–620, 2004.

[37] H. Li and Y. Guo, "New exact solutions to the FitzHugh-Nagumo equation," *Applied Mathematics and Computation*, vol. 180, no. 2, pp. 524–528, 2006.

[38] E. Fan, "Traveling wave solutions for nonlinear equations using symbolic computation," *Computers & Mathematics with Applications*, vol. 43, no. 6-7, pp. 671–680, 2002.

[39] Y. N. Kyrychko, M. V. Bartuccelli, and K. B. Blyuss, "Persistence of travelling wave solutions of a fourth order diffusion system," *Journal of Computational and Applied Mathematics*, vol. 176, no. 2, pp. 433–443, 2005.

[40] I. Hashim, M. S. M. Noorani, and M. R. Said Al-Hadidi, "Solving the generalized Burgers-Huxley equation using the Adomian decomposition method," *Mathematical and Computer Modelling*, vol. 43, no. 11-12, pp. 1404–1411, 2006.

[41] X. Y. Wang, Z. S. Zhu, and Y. K. Lu, "Solitary wave solutions of the generalised Burgers-Huxley equation," *Journal of Physics A: Mathematical and General*, vol. 23, no. 3, pp. 271–274, 1990.

[42] M. A. Helal and M. S. Mehanna, "A comparison between two different methods for solving KdV-Burgers equation," *Chaos, Solitons and Fractals*, vol. 28, no. 2, pp. 320–326, 2006.

[43] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," in *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

[44] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, pp. 225–236, 1951.