A Parameterized Multi-step Newton Method for Solving Systems of Nonlinear Equations

Fayyaz Ahmad
Departament de Física i Enginyeria Nuclear
Universitat Politècnica de Catalunya
Comte d'Urgell 187
08036 Barcelona, Spain
Emran Tohidi
Department of Mathematics
Kosar University of Bojnord
P. O. Box 9415615458, Bojnord, Iran
Juan A. Carrasco
Departament d'Enginyeria Electrònica
Universitat Politècnica de Catalunya
Diagonal 647, plta. 9
08028 Barcelona, Spain

Except for formatting details, this version matches exactly the version published with the same title and authors in *Algorithms*, vol. 71, no. 3, 2016, pp. 631–563

Abstract

We construct a novel multi-step iterative method for solving systems of nonlinear equations by introducing a parameter θ to generalize the multi-step Newton method while keeping its order of convergence and computational cost. By an appropriate selection of θ, the new method can both have faster convergence and have larger radius of convergence. The new iterative method only requires one Jacobian inversion per iteration, and, therefore, can be efficiently implemented using Krylov subspace methods. The new method can be used to solve nonlinear systems of partial differential equations, such as complex generalized Zakharov systems of partial differential equations, by transforming them into systems of nonlinear equations by discretizing approaches in both spacial and temporal dimensions such as, for instance, the Chebyshev pseudo-spectral discretizing method. Quite extensive tests show that the new method can have significantly faster convergence and significantly larger radii of convergence than the multi-step Newton method.

Keywords: Multi-step iterative methods; Multi-step Newton method; systems of nonlinear equations; partial differential equations; discretization methods for partial differential equations.

1 Introduction

Numerical methods for solving nonlinear systems of equations are an important research topic. Nonlinear systems of equations usually arise when discretizing ordinary differential equations (ODEs) and partial differential equations (PDEs). The classical Newton-Raphson method [1] is a basic iterative method for solving nonlinear systems of equations. A large number of papers have considered that method and variants. For instance, Cruz et al. [2] have proposed some gradient-free inexact forms of Newton-Raphson. Moreover, An and Bai [3] have discussed a globally convergent iterative scheme using the GMRES method. It should be noted that they assumed that the Jacobian matrix associated with the considered nonlinear system of equations had a sparse form. In all those methods, LU decomposition or an efficient iterative linear system solver such as the NSCGNR algorithm [4] can be used to avoid the calculation of the inverse of the Jacobian matrix.

Since in multi-step methods the inverse of the Jacobian matrix is computed several times, robust iterative schemes such as Krylov subspace methods [5, 6, 7] should be considered. For instance, the authors of [8] introduced a class of multi-step iterative methods for solving nonlinear systems of equations which avoid the computation of high order Fréchet derivatives. In summary, multi-step iterative methods are computationally attractive. It should be noted that those iterative methods provide an effective way of constructing highly accurate solutions with low computational cost. As a typical iterative method, one can mention the multi-step variant of Newton method [1]. That variant will be called here NR. The NR method for solving a nonlinear system $\mathbf{F}(\mathbf{x}) = 0$ can be described as

$$\text{Base method} \rightarrow \begin{cases} \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0) \\ \mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1 \end{cases}$$

$$\text{Multi-step part} \rightarrow \begin{cases} \text{For } s = 1, m-1 \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_{s+1} = \mathbf{F}(\mathbf{y}_s) \\ \qquad \mathbf{y}_{s+1} = \mathbf{y}_s - \boldsymbol{\phi}_{s+1} \\ \ \text{End} \end{cases}$$

where $\mathbf{F}'(\cdot)$ is the Fréchet derivative [9, 10] or Jacobian of $\mathbf{F}(\cdot)$, $\mathbf{y}_0$ is the initial approximation vector $\mathbf{x}$ for the solution of $\mathbf{F}(\mathbf{x}) = 0$, and $\mathbf{y}_m$ is the approximation vector $\mathbf{x}$ for the solution of $\mathbf{F}(\mathbf{x})$ after an iteration of NR. The NR method uses m (≥ 1) steps to obtain a $m + 1$ convergence order, makes m function evaluations and one Jacobian evaluation, and requires only one LU decomposition and m solutions of lower and upper triangular systems. In this paper, we will construct a new multi-step method which enhances the radius of convergence and the speed of convergence of NR. We will develop the new method by introducing a parameter in NR. A similar idea for scalar algebraic equations has been suggested in [11]. Although we use LU decompositions for solving linear systems in both the base method and the multi-step part, iterative methods such as restarted GMRES could also be used.

The rest of the paper is organized as follows. The new method is presented in Section 2. Section 3 presents the convergence analysis of the new method. In Section 4, we describe how the Chebyshev pseudo-spectral method can be used to discretize a nonlinear system of complex PDEs in spatial and temporal dimensions and reduce it to a system of nonlinear equations, thus building real tests to analyze the new method. Section 5 illustrates the accuracy and efficiency of the new method using two examples generated that way. Section 6 presents the conclusions.

2 New multi-step iterative method

Our new iterative method came out by an attempt to increase the convergence radius in NR without changing its convergence rate and its computational cost. The resulting method (ATC) can be described as

$$
\text{Base method} \rightarrow \begin{cases} \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0) \\ \mathbf{y}_1 = \mathbf{y}_0 - \left(1+\theta-\theta^2\right)\,\boldsymbol{\phi}_1 \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_2 = \mathbf{F}\left(\mathbf{y}_0 - \dfrac{1}{\theta}\boldsymbol{\phi}_1\right) \\ \mathbf{y}_2 = \mathbf{y}_1 - \theta^2\,\boldsymbol{\phi}_2 \end{cases}
$$

$$
\text{Multi-step part} \rightarrow \begin{cases} \text{For } s = 1,\, m-2 \\ \qquad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_{s+2} = \mathbf{F}(\mathbf{y}_{s+1}) \\ \qquad \mathbf{y}_{s+2} = \mathbf{y}_{s+1} - \boldsymbol{\phi}_{s+2} \\ \text{ End} \end{cases}
$$

where $\theta \neq 0$, $\mathbf{y}_0$ is the initial aproximation vector $\mathbf{x}$ for the solution of $\mathbf{F}(\mathbf{x}) = 0$ and $\mathbf{y}_m$ is the approximation vector $\mathbf{x}$ for the solution of $\mathbf{F}(\mathbf{x}) = 0$ after an iteration of the method. The ATC method needs m (≥ 2) steps to obtain a $m+1$ convergence order, makes m function evaluations and one Jacobian evaluation, and requires one LU decomposition, 3 vector-vector multiplications and m solutions of lower and upper triangular systems. The more computationally expensive operations are the LU factorization of the Jacobian and the solutions of the upper and lower triangular systems. Picking up $\theta = 1$ reduces the new method ATC to NR, so the new method can be seen as a generalization of NR keeping the same convergence order. It is clear that by an appropriate selection for the θ parameter the new method can be made to have faster convergence than NR and to have larger convergence radius than NR. While we don't currently have a strategy for picking up a good value for θ, it is possible that such strategies can be developed in the future for particular instances or classes of functions $\mathbf{F}(\cdot)$ such as functions $\mathbf{F}(\cdot)$ arising when solving the Poisson partial differential equation. We will verify that faster convergence is achieved in ATC with respect to NR. That faster convergence must be attributed to the fact that the leading term of the error has smaller value in norm in ATC.

3 Convergence analysis

In this section we first prove that the order of convergence of ATC is four when $m = 3$. Later, we will prove via induction that the order of convergence of ATC is $m+1$. In the constructed proof, we require that the function $\mathbf{F}(\cdot)$ should have at least three Fréchet derivatives. The function $\mathbf{F} : \Gamma \subseteq \mathbb{R}^n \to \mathbb{R}^r$ is Fréchet differentiable [10] at $\mathbf{x} \in \text{interior}(\Gamma)$ if there is an $\mathbf{A} \in \mathbb{L}(\mathbb{R}^n, \mathbb{R}^r)$ such that

$$\lim_{\mathbf{h}\to\mathbf{0}} \frac{||\mathbf{F}(\mathbf{x}+\mathbf{h}) - \mathbf{F}(\mathbf{x}) - \mathbf{A}\mathbf{h}||}{||\mathbf{h}||} = 0 .$$

The linear operator $\mathbf{A}$ is denoted by $\mathbf{F}'(\mathbf{x})$ and is called the Fréchet derivative of $\mathbf{F}(\cdot)$ at $\mathbf{x}$. The higher-order Fréchet derivative of $\mathbf{F}(\mathbf{x})$ with respect to $\mathbf{x}$ can be calculated recursively

$$\begin{aligned}
&\mathbf{F}'(\mathbf{x}) = \text{Jacobian}\left(\mathbf{F}(\mathbf{x})\right) ,\\
&\mathbf{F}^s(\mathbf{x})\mathbf{v}^{s-1} = \text{Jacobian}\left(\mathbf{F}^{s-1}(\mathbf{x})\mathbf{v}^{s-1}\right) , \quad s \geq 2 ,
\end{aligned}$$

where $\mathbf{v}$ is vector.

Theorem 3.1. *Let $\mathbf{F} : \Gamma \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a function with up to third order Fréchet derivative on an open convex neighborhood Γ of $\mathbf{x}^* \in \mathbb{R}^n$ with $\mathbf{F}(\mathbf{x}^*) = 0$ and $\det(\mathbf{F}'(\mathbf{x}^*)) \neq 0$, where $\mathbf{F}'(\mathbf{x})$ denotes the Fréchet derivative of $\mathbf{F}(\mathbf{x})$. Let $\mathbf{C}_1 = \mathbf{F}'(\mathbf{x}^*)$ and $\mathbf{C}_s = \dfrac{1}{s!}\mathbf{F}'(\mathbf{x}^*)^{-1}\mathbf{F}^{(s)}(\mathbf{x}^*)$, for $s \geq 2$, where $\mathbf{F}^{(s)}(\mathbf{x})$ denotes s-order Fréchet derivative of $\mathbf{F}(\mathbf{x})$. Then, for $m = 3$, with an initial guess in the neighborhood of $\mathbf{x}^*$, the sequence $\{\mathbf{x}_k\}$ generated by ATC converges to $\mathbf{x}^*$ with local order of convergence at least four and error*

$$\mathbf{e}_{k+1} = \mathbf{L}{\mathbf{e}_k}^4 + O\left({\mathbf{e}_k}^5\right) ,$$

where $\mathbf{e}_k = \mathbf{x_k} - \mathbf{x}^*$, ${\mathbf{e}_k}^p = \overbrace{(\mathbf{e}_k, \mathbf{e}_k, \ldots, \mathbf{e}_k)}^{p \text{ times}}$, and $\mathbf{L} = -\left(2\left(1 - 1/\theta\right) \mathbf{C}_2\mathbf{C}_3 - 4\,\mathbf{C}_2^3\right)$ is a 4-linear function, i.e. $\mathbf{L} \in \mathbb{L}(\mathbb{R}^n, \mathbb{R}^n, \mathbb{R}^n, \mathbb{R}^n)$ with $\mathbf{L}{\mathbf{e}_k}^4 \in \mathbb{R}^n$.

Proof. Let $\mathbf{F} : \Gamma \subseteq \mathbb{R}^n \to \mathbb{R}^n$ be a function with up to third order Fréchet derivative in Γ. The qth Fréchet derivative of $\mathbf{F}$ at $v \in \mathbb{R}^n$, $q \geq 1$, is a q-linear function $\mathbf{F}^{(q)}(v) : \overbrace{\mathbb{R}^n\mathbb{R}^n \cdots \mathbb{R}^n}^{q \text{ times}}$

with $\mathbf{F}^{(q)}(v)(u_1, u_2, \cdots, u_q) \in \mathbb{R}^n$. Taylor's series expansion of $\mathbf{F}(\mathbf{x}_k)$ around $\mathbf{x}^*$ is

$$\begin{aligned}
\mathbf{F}(\mathbf{x}_k) =& \mathbf{F}(\mathbf{x}^* + \mathbf{x}_k - \mathbf{x}^*) = \mathbf{F}(\mathbf{x}^* + \mathbf{e}_k) \\
=& \mathbf{F}(\mathbf{x}^*) + \mathbf{F}'(\mathbf{x}^*)\,\mathbf{e}_k + \frac{1}{2!}\mathbf{F}''(\mathbf{x}^*)\,\mathbf{e}_k^2 + \frac{1}{3!}\mathbf{F}^{(3)}(\mathbf{x}^*)\,\mathbf{e}_k^3 \\
& + \frac{1}{4!}\mathbf{F}^{(4)}(\mathbf{x}^*)\,\mathbf{e}_k^4 + \cdots \\
=& \mathbf{F}'(\mathbf{x}^*)\Big(\mathbf{e}_k + \frac{1}{2!}\mathbf{F}'(\mathbf{x}^*)^{-1}\mathbf{F}''(\mathbf{x}^*)\,\mathbf{e}_k^2 + \frac{1}{3!}\mathbf{F}'(\mathbf{x}^*)^{-1}\mathbf{F}^{(3)}(\mathbf{x}^*)\,\mathbf{e}_k^3 \\
& + \frac{1}{4!}\mathbf{F}'(\mathbf{x}^*)^{-1}\mathbf{F}^{(4)}(\mathbf{x}^*)\,\mathbf{e}_k^4 + \cdots\Big) \\
=& \mathbf{C}_1\Big(\mathbf{e}_k + \mathbf{C}_2\,\mathbf{e}_k^2 + \mathbf{C}_3\,\mathbf{e}_k^3 + \mathbf{C}_4\,\mathbf{e}_k^4 + O\left({\mathbf{e}_k}^5\right)\Big). \qquad (1)
\end{aligned}$$

Computing the Fréchet derivative of $\mathbf{F}$ with respect to $\mathbf{e}_k$, we get

$$\mathbf{F}'(\mathbf{x}_k) = \mathbf{C}_1\Big(\mathbf{I} + 2\mathbf{C}_2\mathbf{e}_k + 3\mathbf{C}_3{\mathbf{e}_k}^2 + 4\mathbf{C}_4{\mathbf{e}_k}^3 + O\left({\mathbf{e}_k}^4\right)\Big),$$

where $\mathbf{I}$ is the identity matrix. Computing its inverse using a symbolic mathematical package Maple, we obtain

$$\begin{aligned}
\mathbf{F}'(\mathbf{x}_k)^{-1} =& \Big(\mathbf{I} - 2\mathbf{C}_2\mathbf{e}_k + \left(4\mathbf{C}_2^2 - 3\mathbf{C}_3\right)\mathbf{e}_k^2 + \big(6\mathbf{C}_3\mathbf{C}_2 + 6\mathbf{C}_2\mathbf{C}_3 \\
& - 8\mathbf{C}_2^3 - 4\mathbf{C}_4\big)\mathbf{e}_k^3 + \big(8\mathbf{C}_4\mathbf{C}_2 + 9\mathbf{C}_3^2 + 8\mathbf{C}_2\mathbf{C}_4 \\
& -5\mathbf{C}_5 - 12\mathbf{C}_3\mathbf{C}_2^2 - 12\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2 - 12\mathbf{C}_2^2\mathbf{C}_3 + 16\mathbf{C}_2^4\big) \\
& \mathbf{e}_k^4 + O\left({\mathbf{e}_k}^5\right)\Big)\mathbf{C}_1^{-1}. \qquad (2)
\end{aligned}$$

To clarify the notation in the rest of the proof, we note that $\mathbf{x}_k$ is the vector $\mathbf{y}_0$ used in the description of ATC and that $\mathbf{x}_{k+1}$ is the vector $\mathbf{y}_3$ in the description of ATC. The vectors $\boldsymbol{\phi}_1$, $\boldsymbol{\phi}_2$, $\boldsymbol{\phi}_3$, $\mathbf{y}_1$, and $\mathbf{y}_2$ will denote the vectors with same names in the description of ATC which allow to go from $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ when ATC is applied.

Using $\boldsymbol{\phi}_1 = \mathbf{F}'(\mathbf{x}_k)^{-1}\mathbf{F}(\mathbf{x}_k)$, we get

$$\begin{aligned}
\boldsymbol{\phi}_1 =& \Big(\mathbf{I} - 2\mathbf{C}_2\mathbf{e}_k + \left(4\mathbf{C}_2^2 - 3\mathbf{C}_3\right)\mathbf{e}_k^2 + \left(6\mathbf{C}_3\mathbf{C}_2 + 6\mathbf{C}_2\mathbf{C}_3 - 8\mathbf{C}_2^3 - 4\mathbf{C}_4\right)\mathbf{e}_k^3 \\
& + (8\mathbf{C}_4\mathbf{C}_2 + 9\mathbf{C}_3^2 + 8\mathbf{C}_2\mathbf{C}_4 - 5\mathbf{C}_5 - 12\mathbf{C}_3\mathbf{C}_2^2 - 12\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2 - 12\mathbf{C}_2^2\mathbf{C}_3 \\
& + 16\mathbf{C}_2^4)\mathbf{e}_k^4 + O\left({\mathbf{e}_k}^5\right)\Big)\Big(\mathbf{e}_k + \mathbf{C}_2\,\mathbf{e}_k^2 + \mathbf{C}_3\,\mathbf{e}_k^3 + O\left({\mathbf{e}_k}^4\right)\Big) \\
=& \mathbf{e}_k - \mathbf{C}_2\,\mathbf{e}_k^2 + \left(2\mathbf{C}_2^2 - 2\mathbf{C}_3\right)\mathbf{e}_k^3 + \left(4\mathbf{C}_2\mathbf{C}_3 + 3\mathbf{C}_3\mathbf{C}_2 - 4\mathbf{C}_2^3 - 3\mathbf{C}_4\right)\mathbf{e}_k^4 \\
& + (4\mathbf{C}_4\mathbf{C}_2 + 6\mathbf{C}_3^2 + 6\mathbf{C}_2\mathbf{C}_4 + 8\mathbf{C}_2^4 - 6\mathbf{C}_3\mathbf{C}_2^2 - 6\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2 - 8\mathbf{C}_2^2\mathbf{C}_3 \\
& - 4\mathbf{C}_5)\,\mathbf{e}_k^5 + O\left({\mathbf{e}_k}^6\right). \qquad (3)
\end{aligned}$$

Using $\mathbf{y}_1 = \mathbf{x}_k - \left(1+\theta-\theta^2\right) \boldsymbol{\phi}_1$ and plugging (3) we get

$$\begin{aligned}
\mathbf{y}_1 - \mathbf{x}^* =& \mathbf{x}_k - \mathbf{x}^* - \left(1+\theta+\theta^2\right) \boldsymbol{\phi}_1 \\
=& \mathbf{e}_k - \left(1+\theta+\theta^2\right) (\mathbf{e}_k - \mathbf{C}_2 \, \mathbf{e}_k^2 + \left(2\mathbf{C}_2^2 - 2\mathbf{C}_3\right) \, \mathbf{e}_k^3 + (4\mathbf{C}_2\mathbf{C}_3 + \\
& 3\mathbf{C}_3\mathbf{C}_2 - 4\mathbf{C}_2^3 - 3\mathbf{C}_4)\, \mathbf{e}_k^4 + (4\mathbf{C}_4\mathbf{C}_2 + 6\mathbf{C}_3^2 + 6\mathbf{C}_2\mathbf{C}_4 + 8\mathbf{C}_2^4 \\
& - 6\mathbf{C}_3\mathbf{C}_2^2 - 6\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2 - 8\mathbf{C}_2^2\mathbf{C}_3 - 4\mathbf{C}_5)\, \mathbf{e}_k^5 + O\left({\mathbf{e}_k}^6\right)) \\
& = \left(\theta^2-\theta\right) \, \mathbf{e}_k - \left(\theta^2-\theta-1\right) \mathbf{C}_2 \, \mathbf{e}_k^2 + \left(\left(\theta^2-\theta-1\right) \left(\mathbf{C}_2^2 - \mathbf{C}_3\right)\right) \\
& \mathbf{e}_k^3 + \left(\left(\theta^2-\theta-1\right)\left(-4\mathbf{C}_2^3 + 3\mathbf{C}_3\mathbf{C}_2 + 4\mathbf{C}_2\mathbf{C}_3 - 3\mathbf{C}_4\right)\right) \mathbf{e}_k^4 \\
& + O\left({\mathbf{e}_k}^5\right) \, .
\end{aligned} \tag{4}$$

Using $\boldsymbol{\phi}_2 = \mathbf{F}'(\mathbf{x}_k)^{-1}\,\mathbf{F}(\mathbf{x}_k - \boldsymbol{\phi}_1/\theta)$ and substituting (2) and (1) we get

$$\begin{aligned}
\boldsymbol{\phi}_2 =& (\mathbf{I} - 2\mathbf{C}_2\mathbf{e}_k + (4\mathbf{C}_2^2 - 3\mathbf{C}_3)\mathbf{e}_k^2 + (6\mathbf{C}_3\mathbf{C}_2 + 6\mathbf{C}_2\mathbf{C}_3 - 8\mathbf{C}_2^3 - 4\mathbf{C}_4)\mathbf{e}_k^3 \\
& + (8\mathbf{C}_4\mathbf{C}_2 + 9\mathbf{C}_3^2 + 8\mathbf{C}_2\mathbf{C}_4 - 5\mathbf{C}_5 - 12\mathbf{C}_3\mathbf{C}_2^2 - 12\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2 \\
& - 12\mathbf{C}_2^2\mathbf{C}_3 + 16\mathbf{C}_2^4)\mathbf{e}_k^4 + O\left({\mathbf{e}_k}^5\right)) \\
& \left(\mathbf{e}_k - \frac{\boldsymbol{\phi}_1}{\theta} + \mathbf{C}_2\left(\mathbf{e}_k - \frac{\boldsymbol{\phi}_1}{\theta}\right)^2 + \mathbf{C}_3\left(\mathbf{e}_k - \frac{\boldsymbol{\phi}_1}{\theta}\right)^3 + \mathbf{C}_4\left(\mathbf{e}_k - \frac{\boldsymbol{\phi}_1}{\theta}\right)^4\right. \\
& \left. + O\left(\left(\mathbf{e}_k - \frac{\boldsymbol{\phi}_1}{\theta}\right)^5\right)\right) \\
& = \left(1-\frac{1}{\theta}\right) \, \mathbf{e}_k + \left(-1+\frac{1}{\theta}+\frac{1}{\theta^2}\right) \mathbf{C}_2 \, \mathbf{e}_k^2 + \left(2\left(1-\frac{1}{\theta}-\frac{2}{\theta^2}\right) \mathbf{C}_2^2\right. \\
& \left. + \left(-2+\frac{2}{\theta}+\frac{3}{\theta^2}-\frac{1}{\theta^3}\right) \mathbf{C}_3\right) \mathbf{e}_k^3 \\
& + \left(3\left(1-\frac{1}{\theta}-\frac{3}{\theta^2}-\frac{1}{\theta^3}\right) \mathbf{C}_3\mathbf{C}_2 + \left(-4+\frac{4}{\theta}+\frac{13}{\theta^2}\right) \mathbf{C}_2^3\right. \\
& + 2\left(2-\frac{2}{\theta}-\frac{5}{\theta^2}+\frac{1}{\theta^3}\right) \mathbf{C}_2\mathbf{C}_3 + \Big(-3+\frac{3}{\theta}+\frac{6}{\theta^2}-\frac{4}{\theta^3} \\
& \left. + \frac{1}{\theta^4}\Big)\mathbf{C}_4\right) \mathbf{e}_k^4 + O\left({\mathbf{e}_k}^5\right) \, .
\end{aligned} \tag{5}$$

Using $\mathbf{y}_2 = \mathbf{y}_1 - \theta^2\boldsymbol{\phi}_2$ and substituting (4) and (5),

$$\begin{aligned}
\mathbf{y}_2 - \mathbf{x}^* =& \mathbf{y}_1 - \mathbf{x}^* - \theta^2\boldsymbol{\phi}_2 \\
& = \left(\theta^2-\theta\right) \, \mathbf{e}_k - \left(\theta^2-\theta-1\right) \mathbf{C}_2 \, \mathbf{e}_k^2 + \Big(\left(\theta^2-\theta-1\right)
\end{aligned}$$

$$
\begin{aligned}
&\left(\mathbf{C}_2^2-\mathbf{C}_3\right)\Big)\,\mathbf{e}_k^3+\Big(\left(\theta^2-\theta-1\right)(-4\mathbf{C}_2^3+3\mathbf{C}_3\mathbf{C}_2\\
&+4\mathbf{C}_2\mathbf{C}_3-3\mathbf{C}_4)\Big)\,\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\\
&-\theta^2\Bigg(\left(1-\frac{1}{\theta}\right)\,\mathbf{e}_k+\left(-1+\frac{1}{\theta}+\frac{1}{\theta^2}\right)\mathbf{C}_2\,\mathbf{e}_k^2\\
&+\left(2\left(1-\frac{1}{\theta}-\frac{2}{\theta^2}\right)\mathbf{C}_2^2+\left(-2+\frac{2}{\theta}+\frac{3}{\theta^2}-\frac{1}{\theta^3}\right)\mathbf{C}_3\right)\mathbf{e}_k^3\\
&+\Bigg(3\left(1-\frac{1}{\theta}-\frac{3}{\theta^2}-\frac{1}{\theta^3}\right)\mathbf{C}_3\mathbf{C}_2+\left(-4+\frac{4}{\theta}+\frac{13}{\theta^2}\right)\mathbf{C}_2^3\\
&+2\left(2-\frac{2}{\theta}-\frac{5}{\theta^2}+\frac{1}{\theta^3}\right)\mathbf{C}_2\mathbf{C}_3\\
&+\left(-3+\frac{3}{\theta}+\frac{6}{\theta^2}-\frac{4}{\theta^3}+\frac{1}{\theta^4}\right)\mathbf{C}_4\Bigg)\,\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\Bigg)\\
&=\left(2\mathbf{C}_2^2+\left(1-\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3\\
&\quad+\Bigg(3\left(2-\frac{1}{\theta}\right)\mathbf{C}_3\mathbf{C}_2-9\mathbf{C}_2^3+2\left(3-\frac{1}{\theta}\right)\mathbf{C}_2\mathbf{C}_3\\
&+\left(-3+\frac{4}{\theta}-\frac{1}{\theta^2}\right)\mathbf{C}_4\Bigg)\,\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\,.
\end{aligned}
\tag{6}
$$

Substituting $\mathbf{x}_k$ by $\mathbf{y}_2$ in (1) we get, replacing $\mathbf{x}_k$ by the previous expression for $\mathbf{y}_2-\mathbf{x}^*$, and with $\mathbf{z}=\mathbf{x}^*+(\theta^2-\theta)\mathbf{e}_k-(\theta^2-\theta-1)\mathbf{C}_2\,\mathbf{e}_k^2+((\theta^2-\theta-1)(\mathbf{C}_2^2-\mathbf{C}_3))\mathbf{e}_k^3+((\theta^2-\theta-1)(-4\mathbf{C}_2^3+3\mathbf{C}_2\mathbf{C}_2+4\mathbf{C}_2\mathbf{C}_3-3\mathbf{C}_4))\mathbf{e}_k^4+O(\mathbf{e}_k^5)$,

$$
\begin{aligned}
\mathbf{F}(\mathbf{y}_2)=&\mathbf{F}\Bigg(\mathbf{x}^*+\left(\theta^2-\theta\right)\mathbf{e}_k-\left(\theta^2-\theta-1\right)\mathbf{C}_2\,\mathbf{e}_k^2+\Big(\left(\theta^2-\theta-1\right)\\
&\left(\mathbf{C}_2^2-\mathbf{C}_3\right)\Big)\,\mathbf{e}_k^3+\Big(\left(\theta^2-\theta-1\right)(-4\mathbf{C}_2^3+3\mathbf{C}_3\mathbf{C}_2+4\mathbf{C}_2\mathbf{C}_3\\
&-3\mathbf{C}_4)\Big)\,\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\Bigg)\\
=&\mathbf{C}_1\left(\mathbf{z}+\mathbf{C}_2\,\mathbf{z}^2+\mathbf{C}_3\,\mathbf{z}^3+\mathbf{C}_4\,\mathbf{z}^4+O\left(\mathbf{z}^5\right)\right)
\end{aligned}
$$

$$
\begin{aligned}
=&\mathbf{C}_1\left(\left(\left(2\mathbf{C}_2^2+\left(-1+\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3+\left(\left(-3+\frac{4}{\theta}-\frac{1}{\theta^2}\right)\mathbf{C}_4\right.\right.\right.\\
&\left.+2\left(3-\frac{1}{\theta}\right)\mathbf{C}_2\mathbf{C}_3+3\left(2-\frac{1}{\theta}\right)\mathbf{C}_3\mathbf{C}_2-9\mathbf{C}_2^3\right)\mathbf{e}_k^4\\
&\left.+O\left({\mathbf{e}_k}^5\right)\right).
\end{aligned} \tag{7}
$$

Using $\boldsymbol{\phi}_3=\mathbf{F}'(\mathbf{x}_k)^{-1}\mathbf{F}(\mathbf{y}_2)$ and substituting (2) and (7),

$$
\begin{aligned}
\boldsymbol{\phi}_3=&\mathbf{F}'(\mathbf{x}_k)^{-1}\mathbf{F}(\mathbf{y}_2)=\left(\mathbf{I}-2\mathbf{C}_2\mathbf{e}_k+\left(4\mathbf{C}_2^2-3\mathbf{C}_3\right)\mathbf{e}_k^2+(6\mathbf{C}_3\mathbf{C}_2\right.\\
&+6\mathbf{C}_2\mathbf{C}_3-8\mathbf{C}_2^3-4\mathbf{C}_4)\mathbf{e}_k^3+(8\mathbf{C}_4\mathbf{C}_2+9\mathbf{C}_3^2+8\mathbf{C}_2\mathbf{C}_4-5\mathbf{C}_5\\
&\left.-12\mathbf{C}_3\mathbf{C}_2^2-12\mathbf{C}_2\mathbf{C}_3\mathbf{C}_2-12\mathbf{C}_2^2\mathbf{C}_3+16\mathbf{C}_2^4)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\right)\\
&\left(\left(2\mathbf{C}_2^2+\left(-1+\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3+\left(\left(-3+\frac{4}{\theta}-\frac{1}{\theta^2}\right)\mathbf{C}_4\right.\right.\\
&\left.\left.+2\left(3-\frac{1}{\theta}\right)\mathbf{C}_2\mathbf{C}_3+3\left(2-\frac{1}{\theta}\right)\mathbf{C}_3\mathbf{C}_2-9\mathbf{C}_2^3\right)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\right)\\
=&\left(2\mathbf{C}_2^2+\left(-1+\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3+\left(\left(2-\frac{1}{\theta}\right)(3\mathbf{C}_3\mathbf{C}_2\right.\\
&\left.+4\mathbf{C}_2\mathbf{C}_3)-13\mathbf{C}_2^3+\left(-3-\frac{1}{\theta}+\frac{4}{\theta^2}\right)\mathbf{C}_4\right)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right).
\end{aligned} \tag{8}
$$

Using $\mathbf{y}_3=\mathbf{y}_2-\boldsymbol{\phi}_3$ and substituting (6) and (8),

$$
\begin{aligned}
\mathbf{y}_3-\mathbf{x}^*=&\mathbf{y}_2-\mathbf{x}^*-\boldsymbol{\phi}_3=\left(\left(2\mathbf{C}_2^2+\left(1-\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3\right.\\
&+\left(3\left(2-\frac{1}{\theta}\right)\mathbf{C}_3\mathbf{C}_2-9\mathbf{C}_2^3+2\left(3-\frac{1}{\theta}\right)\mathbf{C}_2\mathbf{C}_3\right.\\
&\left.\left.+\left(-3+\frac{4}{\theta}-\frac{1}{\theta^2}\right)\mathbf{C}_4\right)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\right)\\
&\left(\left(2\mathbf{C}_2^2+\left(-1+\frac{1}{\theta}\right)\mathbf{C}_3\right)\mathbf{e}_k^3\right.
\end{aligned}
$$

$$+\left(\left(2-\frac{1}{\theta}\right)(3\mathbf{C}_3\mathbf{C}_2+4\mathbf{C}_2\mathbf{C}_3)-13\mathbf{C}_2^3\right.$$

$$\left.+\left(-3-\frac{1}{\theta}+\frac{4}{\theta^2}\right)\mathbf{C}_4\right)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right)\Bigg)$$

$$=2\mathbf{C}_2\left(\left(\frac{1}{\theta}-1\right)\mathbf{C}_3+2\mathbf{C}_2^2\right)\mathbf{e}_k^4+O\left({\mathbf{e}_k}^5\right).$$

□

Theorem 3.2. *Let* $\mathbf{F}:\Gamma\subseteq\mathbb{R}^n\to\mathbb{R}^n$ *has at least third order Fréchet derivative on an open convex neighborhood* Γ *of* $\mathbf{x}^*\in\mathbb{R}^n$ *with* $\mathbf{F}(\mathbf{x}^*)=0$ *and* $\det(\mathbf{F}'(\mathbf{x}^*))\neq 0$*. Then, the multi-step ATC iterative method has, for* $m\geq 2$*, local convergence order at least* $m+1$*.*

Proof. The proof can be obtained via mathematical induction as done in [9]. □

The error equation for the m-step iterative method ATC is calculated by using the Maple symbolic toolbox that can be written as

$$\mathbf{y}_m-\mathbf{x}^*=(2\mathbf{C}_2)^{m-2}\left(\left(\frac{1}{\theta}-1\right)\mathbf{C}_3+2\mathbf{C}_2^2\right)\mathbf{e}^{m+1}+O\left(\mathbf{e}^{m+2}\right),\quad m\geq 2. \tag{9}$$

The highest Fréchet derivative in the error equation (9) is third order. So, the m-step iterative method ATC has $m+1$ convergence order and it requires that the nonlinear function $\mathbf{F}(\cdot)$ should have at least three Fréchet derivatives. Note that wide classes of important ODEs and PDEs, such as those arising in the Bratu problem, the Frank-Kamenetzkii problem [12], the Lene-Emden equation [13], the Burgers equation [14], the Klein-Gordon equation [15], the two-dimensional sinh-Poisson equation [16], and the three-dimensional nonlinear Poisson equation [17], heat equation, wave equation, Euler's beam equation etc., give rise to $\mathbf{F}(\mathbf{x})$ functions with high order Fréchet derivatives. Then, the multi-step iterative method ATC is applicable to wide classes of important problems. The real parameter $\theta(\neq 0)$ in ATC can be replaced by a vector of non-zero real numbers when $\mathbf{C}_j$ for $j\geq 2$ are diagonal matrices and usually it is the case in the systems of nonlinear equations associated with ODEs and PDEs. The diagonal matrices can be treated as vectors and we define binary and unary operations for them as

$$
\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} v_1\, u_1 \\ v_2\, u_2 \\ \vdots \\ v_n\, u_n \end{bmatrix}, \quad \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}^{-1} = \begin{bmatrix} 1/v_1 \\ 1/v_2 \\ \vdots \\ 1/v_n \end{bmatrix},
$$
$$
\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \pm \text{constant} = \begin{bmatrix} v_1 \pm \text{constant} \\ v_2 \pm \text{constant} \\ \vdots \\ v_n \pm \text{constant} \end{bmatrix}, \tag{10}
$$

and the error equation (9) is verified as

$$
\mathbf{y}_m - \mathbf{x}^* = \left(\left(\frac{1}{\theta} - 1 \right) (2\mathbf{C}_2)^{m-2}\, \mathbf{C}_3 + 2\,(2\mathbf{C}_2)^m \right) \mathbf{e}^{m+1} + O\left(\mathbf{e}^{m+2}\right), \quad m \geq 2.
$$

In that error equation, $(2\mathbf{C}_2)^{m-2}\, \mathbf{C}_3\mathbf{e}^{m+1}$ and $(2\mathbf{C}_2)^m\, \mathbf{e}^{m+1}$ are vectors and $\left(\frac{1}{\theta} - 1\right)\left((2\mathbf{C}_2)^{m-2}\, \mathbf{C}_3\mathbf{e}^{m+1}\right)$ is calculated using (10).

4 A real test problem

In this section, to illustrate the application of the multi-step ATC iterative method, we will consider the nonlinear complex generalized Zakharov system (GZS) of one dimensional PDEs with the Chebyshev pseudo-spectral method for discretize it in spatial and temporal dimensions to reduce it to a nonlinear system of algebraic equations.

4.1 The nonlinear complex generalized Zakharov system

The nonlinear complex Zakharov system has importance in plasma physics [18]. The system includes two coupled nonlinear PDEs which can be written as

$$
i\ \partial_t\, \psi(x,t) + \delta_1 \partial_{tt}\, \psi(x,t) - \delta_2\, \psi(x,t)\, w(x,t) + \delta_3\, |\psi(x,t)|^2 \psi(x,t) = 0 \tag{11}
$$
$$
\partial_{tt}\, w(x,t) - c_s^2 \partial_{xx}\, w(x,t) - \delta_4\, \partial_{xx}\, |\psi(x,t)|^2 = 0 \tag{12}
$$
$$
(x,\ t) \in (a_x,\ b_x) \times (a_t,\ b_t), \tag{13}
$$

subject to the initial and boundary conditions

$$
\begin{array}{ll} \psi(a_x,\ t) = \psi_1(t)\,, & \psi(b_x,\ t) = \psi_2(t) \\ \psi(x,\ 0) = \psi_0(x)\,, & w(a_x,\ t) = w_1(t) \\ w(b_x,\ t) = w_2(x)\,, & w(x,\ 0) = w_3(x) \end{array}, \quad (x,\ t) \in [a_x,\ b_x] \times [a_t,\ b_t]\,. \tag{14}
$$

Several numerical methods have been proposed recently for approximating the solution of (11)–(14) such as the homotopy method [19], the finite difference method [20, 21], and the variational iteration method [22]. Also, Bao et al. [23] suggested some high-accurate numerical methods for solving numerically (11)–(14). Bao and Sun [24] applied a new technique based on time-splitting discretization for approximating the solution of a variant of (11)–(14).

One can split (11) using the real and imaginary parts of $\psi(x,t)$, $u(x,t)$ and $v(x,t)$, as

$$\begin{aligned}
&\partial_t\, u(x,t) + \delta_1 \partial_{xx}\, v(x,t) - \delta_2 v(x,t) w(x,\ t) + \delta_3 \left(u^2(x,t) + v^2(x,t)\right) v(x,t) = 0\,,\\
&-\partial_t\, v(x,t) + \delta_1 \partial_{xx}\, u(x,t) - \delta_2 u(x,t) w(x,t) + \delta_3 \left(u^2(x,t) + v^2(x,t)\right) u(x,t) = 0\,,\\
&\partial_{tt}\, w(x,t) - c_s^2 \partial_{xx}\, w(x,t) - 2\delta_4 \big(u(x,t)\partial_{xx}\, u(x,t) + (\partial_x\, u(x,t))^2\\
&+ v(x,t)\partial_{xx}\, v(x,t) + (\partial_x\, v(x,t))^2\,\big) = 0\,,
\end{aligned} \tag{15}$$

with the initial and boundary conditions

$$\left.\begin{array}{ll}
u(a_x,t) = \alpha_1(t)\,, & u(b_x,t) = \alpha_2(t)\\
v(a_x,t) = \alpha_3(t)\,, & v(b_x,t) = \alpha_4(t)\\
w(a_x,t) = \alpha_5(t)\,, & w(b_x,t) = \alpha_6(t)\\
u(x,a_t) = \beta_1(x)\,, & v(x,a_t) = \beta_2(x)\\
w(x,a_t) = \beta_3(x)\,, & w_t(x,a_t) = \beta_4(x)
\end{array}\right. ,\quad (x,\ t) \in [a_x,\ b_x] \times [a_t,\ b_t]\,. \tag{16}$$

The matrix form of the nonlinear system (15) is

$$\begin{bmatrix} \partial_t & \delta_1\partial_{xx} & 0\\ \delta_1\partial_{xx} & -\partial_t & 0\\ 0 & 0 & \partial_{tt} - c_s^2\partial_{xx} \end{bmatrix} \begin{bmatrix} u\\ v\\ w \end{bmatrix} + \begin{bmatrix} q_1(u,v,w)\\ q_2(u,v,w)\\ q_3(u,v,w) \end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0 \end{bmatrix}, \tag{17}$$

where

$$\begin{aligned}
q_1 &= -\delta_2 v(x,t) w(x,\ t) + \delta_3 \left(u^2(x,t) + v^2(x,t)\right) v(x,t)\,,\\
q_2 &= -\delta_2 u(x,t) w(x,t) + \delta_3 \left(u^2(x,t) + v^2(x,t)\right) u(x,t)\,,\\
q_3 &= -2\delta_4 \left(u(x,t)\partial_{xx}\, u(x,t) + (\partial_x\, u(x,t))^2 + v(x,t)\partial_{xx}\, v(x,t) + (\partial_x\, v(x,t))^2\right),
\end{aligned}$$

the constants δ_1, δ_2, δ_3, δ_4 and c_s are given, and the functions $\alpha_i(t)$, $1 \le i \le 6$ and $\beta_j(x)$, $1 \le j \le 4$ are known.

In the next section, we will use the Chebyshev pseudo-spectral method for discretizing (17) subject to the initial and boundary conditions (16) to reduce (17) to a system of nonlinear algebraic equations.

4.2 The Chebyshev pseudo-spectral method

Spectral methods are the best methods for approximating the solutions of problems in applied mathematics and engineering when the solutions are smooth and their domains are simple. Many researchers have used those methods for the numerical solution of nonlinear PDEs [25], fractional ODEs [26], high-order boundary value problems [27], systems of Volterra integral equations [28], optimal control problems governed by Volterra integral equations [29], Quasi Bang-Bang optimal control problems [30], and ODEs of degenerate types [31]. In relation to many other methods, spectral methods give highly accurate results.

To discretize (17) subject to the initial and boundary conditions (16) using the Chebyshev pseudo-spectral method, we define the following transformations

$$\begin{aligned} y &= \frac{2}{b_x - a_x}\, x - \frac{a_x + b_x}{b_x - a_x}\,, \\ \tau &= \frac{2}{b_t - a_t}\, t - \frac{a_t + b_t}{b_t - a_t}\,, \end{aligned}$$

where $(y,\ \tau) \in [-1,\ 1] \times [-1,\ 1]$. The partial derivatives with respect to the variables associated with the new domain are related to the partial derivatives with respect to the variables associated with the previous domain as

$$\begin{aligned} \partial_x &= \left(\frac{2}{b_x - a_x}\right)\partial_y\,, \quad \partial_{xx} = \left(\frac{2}{b_x - a_x}\right)^2 \partial_{yy}\,, \\ \partial_t &= \left(\frac{2}{b_t - a_t}\right)\partial_\tau\,, \quad \partial_{tt} = \left(\frac{2}{b_t - a_t}\right)^2 \partial_{\tau\tau}\,. \end{aligned}$$

Let n_x and n_t be the number of grid points in, respectively, the spatial and temporal domains associated with the variables y and τ. The partitions of $[-1,\ 1]$ in the space and time directions are performed using Chebyshev-Gauss-Lobatto (CGL) points. The number of grid points is $n = n_x n_t$. Let

$$\begin{aligned} \mathbf{U} &= [u_{1,1}, u_{1,2}, \cdots, u_{1,n_t}, u_{2,1}, u_{2,2}, \cdots, u_{2,n_t}, \cdots, u_{n_x,1}, u_{n_x,2}, \cdots, u_{n_x,n_t}]^T\,, \\ \mathbf{V} &= [v_{1,1}, v_{1,2}, \cdots, v_{1,n_t}, v_{2,1}, v_{2,2}, \cdots, v_{2,n_t}, \cdots, v_{n_x,1}, u_{n_x,2}, \cdots, v_{n_x,n_t}]^T\,, \\ \mathbf{W} &= [w_{1,1}, w_{1,2}, \cdots, w_{1,n_t}, w_{2,1}, w_{2,2}, \cdots, w_{2,n_t}, \cdots, w_{n_x,1}, w_{n_x,2}, \cdots, w_{n_x,n_t}]^T \end{aligned}$$

be vectors collecting the values of the functions $u(y,\tau)$, $v(y,\tau)$ and $w(y,\tau)$ at the grid points. The discrete approximations for the partial derivatives are

$$\begin{aligned} \partial_y &\approx \mathbf{D}_y \otimes \mathbf{I}_\tau\,, \quad \partial_{yy} \approx \mathbf{D}_y^2 \otimes \mathbf{I}_\tau\,, \\ \partial_\tau &\approx \mathbf{I}_y \otimes \mathbf{D}_\tau\,, \quad \partial_{\tau\tau} \approx \mathbf{I}_y \otimes \mathbf{D}_\tau^2\,, \end{aligned} \tag{18}$$

where $\mathbf{D}_y$, $\mathbf{D}_\tau$, $\mathbf{I}_y$, $\mathbf{I}_\tau$ are, respectively, the Chebyshev differentiation and identity matrices for variables y and τ, the dimensions for subscripts y and τ are, respectively, n_x and n_t, and $\otimes$ denotes the Kronecker product. Finally, we define the partial derivative operators as

$$\begin{aligned} \mathbf{A}_x &= \frac{2}{b_x - a_x}\mathbf{D}_y \otimes \mathbf{I}_\tau\,, \quad \mathbf{A}_{xx} = \frac{2}{b_x - a_x}\mathbf{D}_y^2 \otimes \mathbf{I}_\tau\,, \\ \mathbf{A}_t &= \frac{2}{b_t - a_t}\mathbf{I}_y \otimes \mathbf{D}_\tau\,, \quad \mathbf{A}_{tt} = \frac{2}{b_t - a_t}\mathbf{I}_y \otimes \mathbf{D}_\tau^2\,. \end{aligned}$$

The discrete form of (17) is, using $\mathbf{x}_{m\times n}$ to denote an $m \times n$ matrix $\mathbf{x}$,

$$\begin{bmatrix} \mathbf{A}_t & \delta_1 \mathbf{A}_{xx} & \mathbf{O} \\ \delta_1 \mathbf{A}_{xx} & -\mathbf{A}_t & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{A}_{tt} - c_s^2 \mathbf{A}_{xx} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{W} \end{bmatrix} + \begin{bmatrix} \mathbf{Q}_1(\mathbf{U},\mathbf{V},\mathbf{W}) \\ \mathbf{Q}_2(\mathbf{U},\mathbf{V},\mathbf{W}) \\ \mathbf{Q}_3(\mathbf{U},\mathbf{V},\mathbf{W}) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \end{bmatrix}, \tag{19}$$

where

$$\mathbf{Q}_1 = -\delta_2 \mathbf{V} \odot \mathbf{W} + \delta_3 \left(\mathbf{U} \odot \mathbf{U} + \mathbf{V} \odot \mathbf{V} \right) \odot \mathbf{V} \,,$$
$$\mathbf{Q}_2 = -\delta_2 \mathbf{U} \odot \mathbf{W} + \delta_3 \left(\mathbf{U} \odot \mathbf{U} + \mathbf{V} \odot \mathbf{V} \right) \odot \mathbf{U} \,,$$
$$\mathbf{Q}_3 = -2\delta_4 \Big(\mathbf{U} \odot (\mathbf{A}_{xx}\mathbf{U}) + (\mathbf{A}_x\mathbf{U}) \odot (\mathbf{A}_x\mathbf{U}) + \mathbf{V} \odot (\mathbf{A}_{xx}\mathbf{V}) + (\mathbf{A}_x\mathbf{V}) \odot (\mathbf{A}_x\mathbf{V}) \Big),$$

with $\odot$ denoting the point-wise multiplication between vectors.

The compact form of (19) is

$$\mathbf{F}(\mathbf{S}) \equiv \mathbf{A}\mathbf{S} + \mathbf{Q} - \mathbf{B} = \mathbf{0} \,, \tag{20}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_t & \delta_1 \mathbf{A}_{xx} & \mathbf{O} \\ \delta_1 \mathbf{A}_{xx} & -\mathbf{A}_t & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{A}_{tt} - c_s^2 \mathbf{A}_{xx} \end{bmatrix}_{3n\times 3n} , \qquad \mathbf{S} = \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \\ \mathbf{W} \end{bmatrix}_{3n\times 1} ,$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1(\mathbf{U},\mathbf{V},\mathbf{W}) \\ \mathbf{Q}_2(\mathbf{U},\mathbf{V},\mathbf{W}) \\ \mathbf{Q}_3(\mathbf{U},\mathbf{V},\mathbf{W}) \end{bmatrix}_{3n\times 1} , \qquad \mathbf{B} = \begin{bmatrix} \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} \end{bmatrix}_{3n\times 1} .$$

Our aim is to solve the nonlinear system of algebraic equations (20) by the proposed new multi-step ATC method presented in Section 2. We have to adapt the structure of $\mathbf{F}(\mathbf{S}) = \mathbf{0}$ to the initial and boundary conditions. Using Matlab-like notation, the initial and boundary conditions can be written as

Initial conditions

$$\begin{array}{ll} \text{for } \ i = 1 : n_x & \\ \quad indx_1 = (i-1)n_t + 1 \,, & indx_2 = (i-1)n_t + 2 \,, \\ \quad A(indx_1, :) = 0 \,, & A(n + indx_1, :) = 0 \,, \\ \quad A(indx_1, 1:n) = D_1(i,:) \,, & A(n + indx_1, n+1:2n) = D_1(i,:) \,, \\ \quad A(2n + indx_1, :) = 0 \,, & A(2n + indx_2, :) = 0 \,, \\ \quad A(2n + indx_1, 2n+1:3n) = D_1(i,:) \,, & A(2n + indx_2, 2n+1:3n) = D_2(i,:) \,, \\ \quad B(indx_1) = \beta_1(i) \,, & B(n + indx_1) = \beta_2(i) \,, \\ \quad B(2n + indx_1) = \beta_3(i) \,, & B(2n + indx_2) = \beta_4(i) \,, \\ \text{end} & \end{array} \tag{21}$$

where $\mathbf{D}_1 = \mathbf{I}_x \otimes \mathbf{I}_t(1,:)$ and $\mathbf{D}_2 = \mathbf{I}_x \otimes \left(\dfrac{2}{b_t - a_t} \right) \mathbf{D}_\tau$, and

Boundary conditions

$$
\begin{array}{ll}
A(1:n_t,:) = 0\,, & B = 0\,,\\
A(1:n_t,1:n_t) = I_t\,, & B(1:n_t) = \alpha_1(1:n_t)\,,\\
A(n-n_t+1:n,:) = 0\,, & B(n-n_t+1:n) = \alpha_2(1:n_t)\,,\\
A(n-n_t+1:n,n-n_t+1:n) = I_t\,, & B(n+1:n+n_t) = \alpha_3(1:n_t)\,,\\
A(n+1:n+n_t,:) = 0\,, & B(2n-n_t+1:2n) = \alpha_4(1:n_t)\,,\\
A(n+1:n+n_t,n+1:n+n_t) = I_t\,, & B(2n+1:2n+n_t) = \alpha_5(1:n_t)\,,\\
A(2n-n_t+1:2n,:) = 0\,, & B(3n-n_t+1:3n) = \alpha_6(1:n_t)\,,\\
A(2n-n_t+1:2n,2n-n_t+1:2n) = I_t\,, & A(2n+1:2n+n_t,:) = 0\,,\\
A(2n+1:2n+n_t,2n+1:2n+n_t) = I_t\,, & A(3n-n_t+1:3n,:) = 0\,,\\
A(3n-n_t+1:3n,3n-n_t+1:3n) = I_t\,, &
\end{array}
\tag{22}
$$

Finally the rows of $\mathbf{Q}$ and Jacobian of $\mathbf{Q}$ get zeros where $\mathbf{B}$ gets values from initial and boundary conditions. After these modifications, nonlinear system of algebraic equations will be updated and can be solved by any iterative methods such as ATC or NR.

5 Numerical analysis

In this section we show the accuracy and performance of the multi-step iterative method ATC when used to solve a system of nonlinear equations obtained by using the Chebyshev pseudo-spectral method to discretize the nonlinear complex generalized Zakharov system (GZS) of partial differential equations. In [18], two test problems concerning GZS have been solved with good accuracy by using the Jacobi pseudo-spectral collocation method. As a comparison we will solve the same two test problems with higher accuracies than those reported in [18]. The errors will be computed using the $||\cdot||_\infty$ norm over the entire grid as

$$
\begin{aligned}
E_u &= \max_{(x,t)\in\Lambda} |u(x,t) - u_{\mathrm{num}}(x,t)|\,,\\
E_v &= \max_{(x,t)\in\Lambda} |v(x,t) - v_{\mathrm{num}}(x,t)|\,,\\
E_w &= \max_{(x,t)\in\Lambda} |w(x,t) - w_{\mathrm{num}}(x,t)|\,,
\end{aligned}
\tag{23}
$$

where Λ is the grid of values for (x,t) used in the discretization, and $u_{\mathrm{num}}(x,t)$, $v_{\mathrm{num}}(x,t)$ and $w_{\mathrm{num}}(x,t)$ are the computed numerical values of the functions $u(x,t)$, $v(x,t)$ and $w(x,t)$. In all computations, the initial guesses for $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ will be taken equal to the zero vector $\mathbf{O}_{n\times 1}$.

5.1 Complex Zakharov equation

The first test problem is the GZS [18]:

$$
\begin{aligned}
& i\,\partial_t\,\psi - \partial_{xx}\,\psi - \psi\,w = 0\\
& \partial_{tt}\,w - \partial_{xx}\,w + \partial_{xx}\,|\psi|^2 = 0\,,
\end{aligned}
\tag{24}
$$

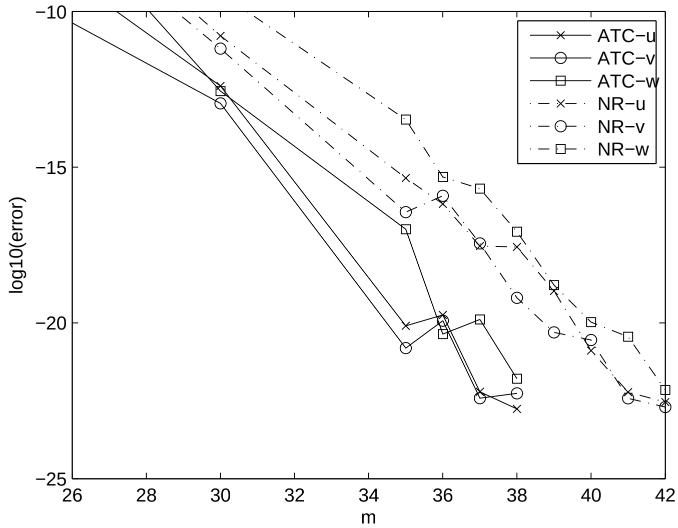

Figure 1: Errors under ATC and NR for the first test problem.

with domain $\Lambda = [-1,\ 1] \times [0, 3.3]$. That GZS has the analytical solution

$$\begin{aligned} &\psi(x,\ t) = u(x,\ t) + i\, v(x,\ t) = \sqrt{3}\, e^{i(x+t)} \tanh\left(\frac{1}{\sqrt{2}}(x+2t)\right) \\ &w(x,\ t) = 1 - \tanh^2\left(\frac{1}{\sqrt{2}}(x+2t)\right). \end{aligned} \tag{25}$$

We choose the parameter $\theta = 1.0 - 0.001\,\mathrm{rand}(3n, 1)$, where $\mathrm{rand}(3n, 1)$ is a uniform random vector of dimension $3n$ in the interval $[0, 1]$ for each component. The role of the parameter θ is important because θ may affect the actual speed of convergence and the convergence radius. We solved the complex Zakharov equation in the domain $[-1, 1] \times [0, 3.3]$ with $n_x = 23$ grid points for the space dimension and $n_t = 48$ grid points for the time dimension. Table 1 compares the errors obtained by ATC and the NR multi-step method as a function of the number of steps m. The table also gives the execution time of ATC for $m = 38$ and NR for $m = 42$, numbers of steps under which the errors in both methods are sufficiently small and similar. We can note that for the same number of steps the errors under ATC are significantly smaller than under NR, particularly when the number of steps m becomes large. With similar error targets, the ATC method is about 7 % faster than the NR method. Figure **??** shows the errors in u, v and w in logarithmic scale against the number of steps m for methods ATC and NT. Figures **??**, **??** and **??** plot, respectively, $u(x, t)$, $v(x, t)$ and $w(x, t)$ at the grid points.

5.2 Complex generalized Zakharov equation

The second test problem we consider is the GZS in complex form, which is [18]:

$$\begin{aligned} &i\partial_t \psi + \partial_{xx}\psi + 2\psi w - 2|\psi|^2 = 0 \\ &\partial_{tt} w - \partial_{xx} w + \partial_{xx}|\psi|^2 = 0, \end{aligned} \tag{26}$$

Table 1: Performance comparison of ATC and NR for the first test problem.

	ATC method			NR method		
	Execution time for $m = 38 = 49.285$ s			Execution time for $m = 42 = 53.096$ s		
m	E_u	E_v	E_w	E_u	E_v	E_w
1	0.47454	0.49695	0.85826	0.47463	0.49733	0.85754
2	1.1371	1.2102	3.72	1.1361	1.21	2.8038
3	4.9923	3.019	3.7464	5.0242	3.0273	3.2173
4	3.2748	5.5037	5.1003	3.2919	5.5702	5.1487
5	3.3246	2.1363	18.102	3.815	2.1929	18.948
10	1.0611	1.8962	3.0289	2.0823	4.5914	8.0358
15	0.047661	0.12338	0.53352	0.31678	0.5274	5.7334
20	0.081501	0.020402	0.0076958	1.0876	0.76647	0.33485
25	0.00029604	5.9708e-05	0.0027184	0.0040615	0.0019482	0.028504
30	4.143e-06	2.3705e-06	3.5209e-06	2.0652e-05	1.3753e-05	8.1167e-05
35	1.8722e-09	9.1849e-10	4.1729e-08	2.1499e-07	7.2315e-08	1.4055e-06
36	2.6555e-09	2.2107e-09	1.4384e-09	9.4002e-08	1.2213e-07	2.2354e-07
37	2.2531e-10	1.8415e-10	2.307e-09	2.4263e-08	2.6427e-08	1.5319e-07
38	1.3032e-10	2.1521e-10	3.4423e-10	2.3561e-08	4.6183e-09	3.8451e-08
39				5.74e-09	1.5185e-09	6.9447e-09
40				8.4909e-10	1.1898e-09	2.1135e-09
41				2.2375e-10	1.8279e-10	1.3227e-09
42				1.6263e-10	1.3847e-10	2.3933e-10

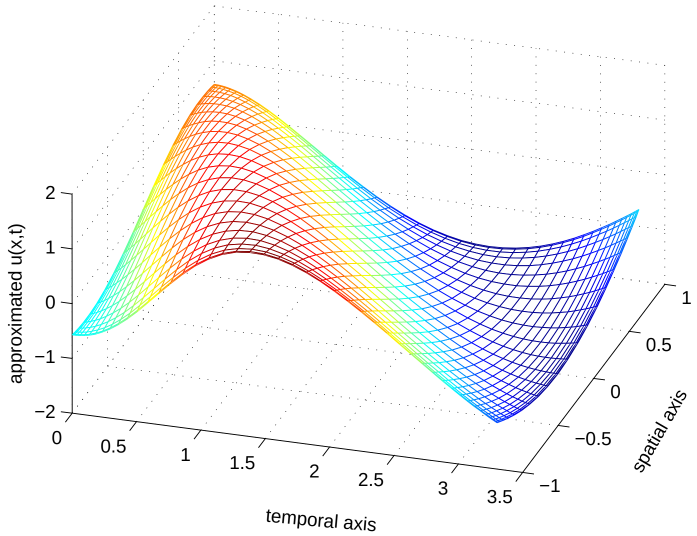

Figure 2: Computed $u(x, t)$ for the first test problem.

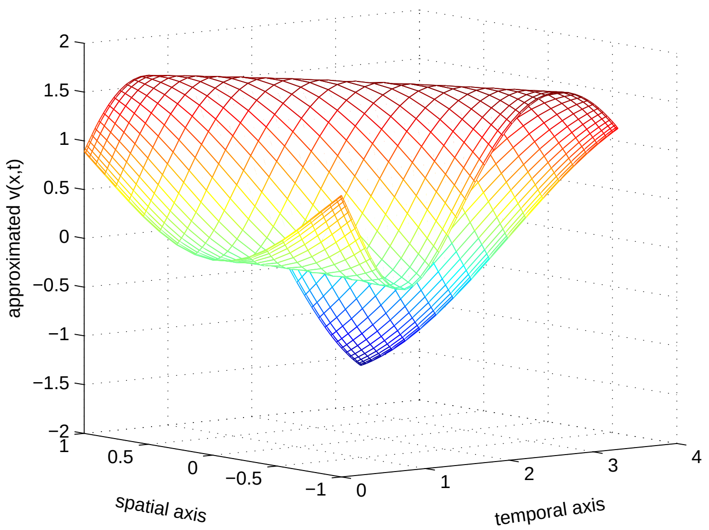

Figure 3: Computed $v(x, t)$ for the first test problem.

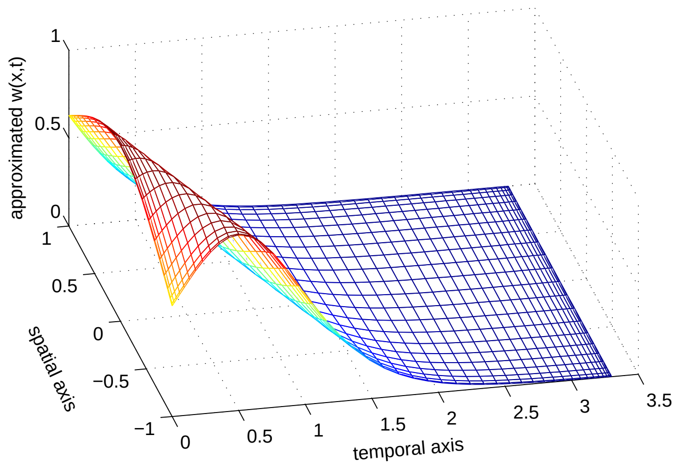

Figure 4: Computed $w(x,t)$ for the first test problem.

with the domains $\Lambda = [-1,\ 1] \times [0, 1.2]$ and $\Lambda = [-1,\ 1] \times [0, 1.3]$. That problem has the analytical solution

$$\begin{aligned} &\psi(x,\ t) = u(x,\ t) + iv(x,\ t) = \frac{\sqrt{3}}{2} e^{-i(x+3t)} \tanh(x+2t) \\ &w(x,\ t) = -\frac{1}{4} \tanh^2(x+2t)\,. \end{aligned} \tag{27}$$

We will consider several numbers of grid points in the space dimension, n_x, and in the time dimension, n_t. We will also consider several values for θ. Table 2 compares the performance of ATC and NR for $\Lambda = [-1, 1] \times [0, 1.2]$, $n_x = 32$, $n_t = 28$, and $\theta = 1.3$, as a function of the number of steps. We can note that, for the same number of steps ATC yields smaller errors than NR. With similar errors, the execution time of ATC for $m = 21$ is smaller than the execution time of NR for $m = 14$. When we integrate the complex generalized Zakharov equation for $\Lambda = [-1, 1] \times [0, 1.3]$, the NR method shows divergence. This is illustrated in Table 3 which compares the behavior of ATC and NR for $\Lambda = [-1, 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$. Therefore, an appropriate selection for the parameter θ increases the convergence radius of ATC in comparison with NR. Figure **??** plots the errors in $u(x,t)$, $v(x,t)$ and $w(x,t)$ as a function of the number of steps m under ATC and NR for $\Lambda = [-1, 1] \times [0, 1.2]$, $n_x = 32$, $n_t = 28$, and $\theta = 1.3$. We can note that, for the same number of steps the errors under ATC are smaller than under NR. Figure **??** gives the absolute errors in u, v, w at the different grid points for $m = 24$ under ATC for $\Lambda = [-1, 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$. Figures **??**, **??** and **??** plot, respectively, $u(x,t)$, $v(x,t)$, $w(x,t)$ and the corresponding absolute errors under ATC for $m = 24$, $\Lambda = [-1, 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$.

The results obtained in [18] are presented in Table 4. We used more number of grid points in spatial dimension than that of [18] and got better numerical accuracy in numerical results.

Table 2: Performance comparison of ATC and NR for the second test problem with domain $\Lambda = [-1, 1] \times [0, 1.2]$, $n_x = 32$, $n_t = 28$ and $\theta = 1.3$.

	ATC method			NR method		
	Execution time for $m = 21 = 11.891$			Execution time for $m = 24$ =12.953		
m	E_u	E_v	E_w	E_u	E_v	E_w
1	0.45709	0.8603	0.21256	0.48046	1.1721	0.23063
2	1.1231	0.77984	0.64226	1.3562	1.2781	0.64226
3	0.96367	1.6933	0.40012	2.0621	2.5231	0.58512
4	1.5751	0.40543	0.32973	3.1627	1.6789	0.46649
5	0.20387	0.2434	0.095349	0.56035	0.64707	0.14493
6	0.062707	0.066969	0.039415	0.03373	0.14675	0.0401
7	0.045025	0.025539	0.011057	0.060376	0.020351	0.016544
8	0.0033644	0.0095145	0.0027489	0.0050918	0.016299	0.0047009
9	0.0036137	0.0015154	0.00086978	0.005709	0.0059042	0.0024091
10	0.00027201	0.00079506	0.00012986	0.0051656	0.0029367	0.0021463
11	0.00020245	7.6515e-05	5.1606e-05	0.0018987	0.0026428	0.00085315
12	3.5057e-05	4.5131e-05	1.0796e-05	0.0013403	0.00052468	0.0004666
13	7.0703e-06	8.5718e-06	2.8361e-06	0.00045672	0.00042808	0.00013906
14	2.6527e-06	1.962e-06	6.4814e-07	0.00011958	9.8066e-05	4.5534e-05
15	3.2703e-07	5.7061e-07	8.6105e-08	4.9319e-05	3.0739e-05	1.0606e-05
16	1.3652e-07	1.1004e-07	4.3605e-08	4.7169e-06	8.9355e-06	2.271e-06
17	1.4762e-08	2.3321e-08	4.677e-09	2.7754e-06	1.2644e-06	5.2229e-07
18	4.0657e-09	4.2216e-09	1.513e-09	1.3475e-07	5.0502e-07	7.0055e-08
19	5.6959e-10	6.0451e-10	3.4595e-10	1.0105e-07	4.7431e-08	2.7238e-08
20	2.8374e-10	2.833e-10	9.2313e-11	1.3424e-08	1.8865e-08	4.1215e-09
21	8.5017e-11	7.0587e-11	7.6073e-11	2e-09	2.5181e-09	9.5584e-10
22				6.8728e-10	3.6522e-10	2.2399e-10
23				6.0788e-11	1.1403e-10	6.8574e-11
24				3.1735e-11	1.5148e-11	7.1093e-11

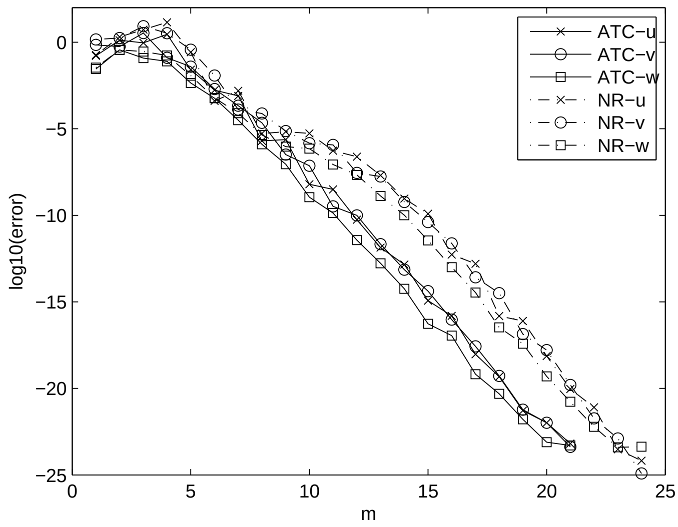

Figure 5: Errors in u, v, w for ATC and NR as a function on the number of steps for $\Lambda = [-1,\ 1] \times [0, 1.2]$, $n_t = 28$, $n_x = 32$, and $\theta = 1.3$

Table 3: Performance comparison of ATC and NR for second test problem with domain $\Lambda = [-1,\ 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$ and $\theta = 2$.

	ATC method				NR method		
	Execution time for $m = 24 = 16.259$			Execution time for $m = 24$ =8.588			
m	E_u	E_v	E_w	m	E_u	E_v	E_w
1	2.3489	1.8274	1 0.49851	1	0.86782	1.1797	0.23024
2	1.1081	0.54759	0.96708	2	1.5424	1.9725	0.96708
7	0.10085	0.067689	0.029542	3	5.615	3.8987	0.95367
13	6.2449e-05	7.1607e-05	1.8598e-05	4	15.611	20.172	1.5002
17	2.4756e-07	3.291e-07	7.4927e-08	5	348.08	205.49	3.8896
19	1.9145e-08	2.1149e-08	5.7041e-09	6	2.355e+05	5.6693e+05	9506.1
22	5.151e-10	6.2691e-10	1.1519e-10	7	7.8999e+14	2.5429e+14	1.1504e+12
23	2.9278e-10	1.2219e-10	9.2533e-11	8	3.528e+42	5.2674e+42	1.3026e+31
24	5.3785e-11	6.9398e-11	7.4369e-11	9	1.8566e+127	1.1924e+127	2.1703e+86

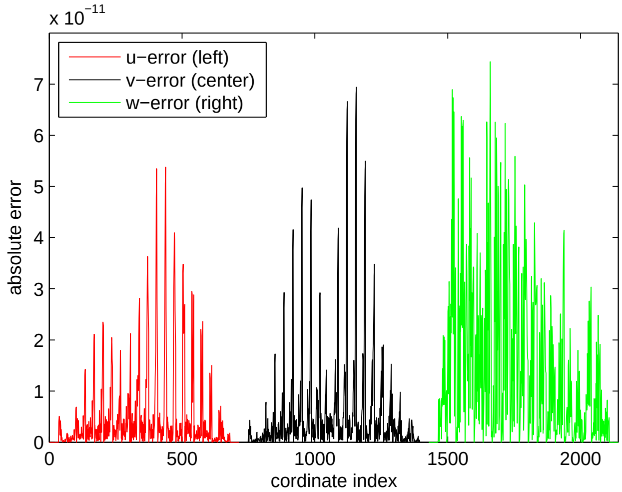

Figure 6: Absolute errors in u, v, w at the grid points for the second test problem under ATC for $m = 24$, $\Lambda = [-1,\ 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$.

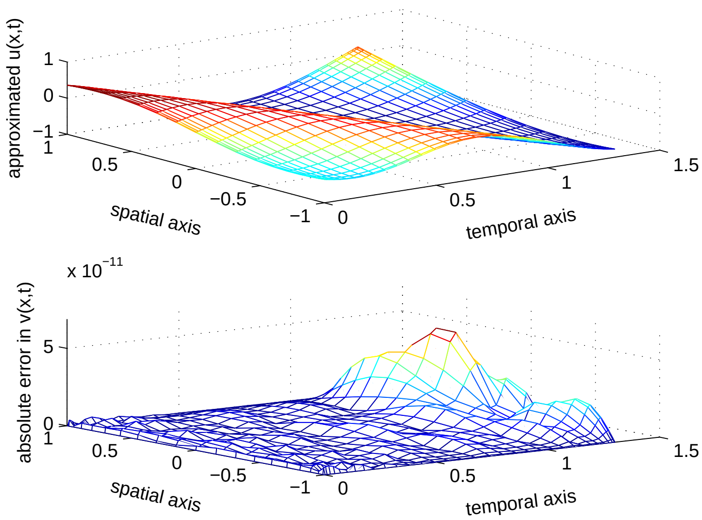

Figure 7: $u(x, t)$ and corresponding absolute errors under ATC for $m = 14$, $\Lambda = [-1,\ 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$

Table 4: Performance of method presented in [18] for the first test problem, $\Lambda = [-1,\ 1] \times [0, 1]$, $n_x = 4, 8, 12, 16$.

n_x	E_u	E_v	E_w
4	4.43e-2	7.12e-2	4.53e-2
8	2.13e-4	1.624e-4	1.20e-4
12	8.34e-7	6.02e-7	2.54e-7
16	3.83e-7	3.4e-7	1.51e-8

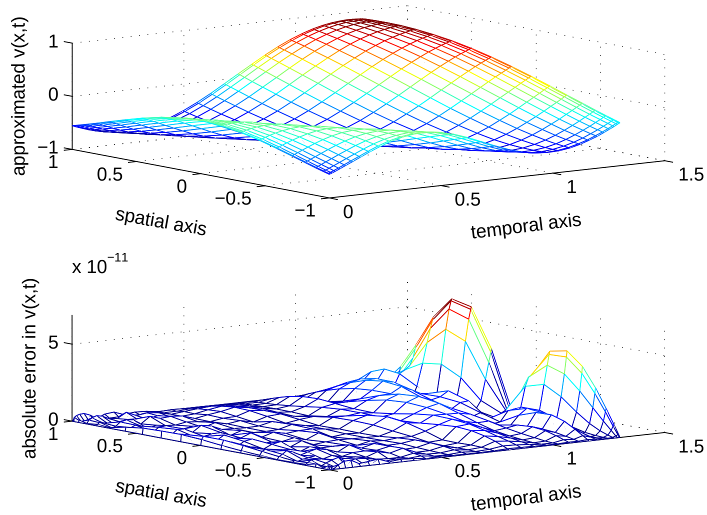

Figure 8: $v(x,t)$ and corresponding absolute errors under ATC for $m = 14$, $\Lambda = [-1,\ 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$

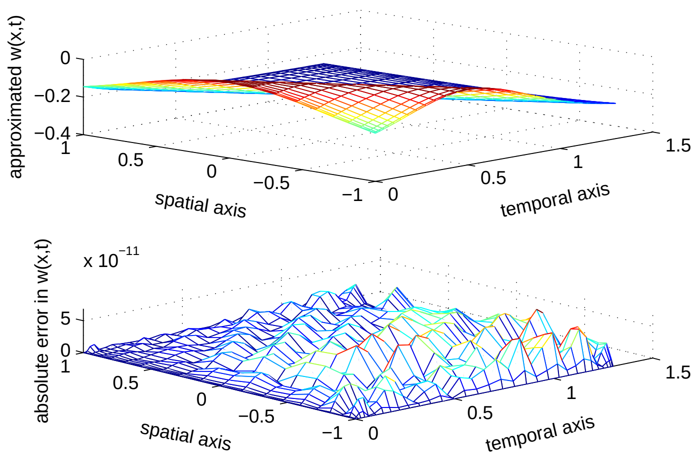

Figure 9: $w(x,t)$ and corresponding absolute errors under ATC for $m = 14$, $\Lambda = [-1,\ 1] \times [0, 1.3]$, $n_x = 21$, $n_t = 34$, and $\theta = 2$

6 Conclusions

Multi-step iterative methods for solving nonlinear systems tend to be computationally economical. The ATC method makes only one Jacobian evaluation. Once the LU-factors of the Jacobian are evaluated, they are used in the multi-step part to make the method computationally efficient. Our numerical results clearly show that ATC has better speed of convergence than NR and, with an appropriate selection of , wider radius of convergence. Applied to the complex generalized Zakharov equation with using the Chebyshev pseudo-spectral method for discretization, the ATC gives more accurate numerical solutions than they have been obtained in [18].

Acknowledgment

The first author is funded by the Spanish MICINN grant AYA2010-15685.

References

[1] J. F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, 1964.

[2] W. L. Cruz, J. M. Martinez, M. Raydan, Spectral residual method without gradient information for solving large-scale nonlinear systems of equations, Math. Comput, 75 (2006) 14291448.

[3] H. B. An, Z. Z. Bai, A globally convergent Newton-GMRES method for large sparse systems of nonlinear equations. Appl. Numer. Math, 57 (2007) 235252.

[4] M. Kh. Zak, F. Toutounian, Nested splitting conjugate gradient method for matrix equation $AXB = C$ and preconditioning, Comput. Math. Appl, 66 (2013) 269-278.

[5] A. Cordero, J.L. Hueso, E. Martinez, J.R. Torregrosa, A modified Newton-Jarratts composition, Numer. Algorithms, 55 (2010) 87-99.

[6] J. R. Sharma, H. Arora, Efficient Jarratt-like methods for solving systems of nonlinear equations, Calcolo, 51 (2014) 193-210.

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

[8] F. Soleymani, T. Lotfi, P. Bakhtiari, A multi-step class of iterative methods for nonlinear systems, Optim. Letters, 8 (2014) 1001-1015.

[9] M. Z. Ullah, S. Serra-Capizzano, F. Ahmad, An efficient multi-step iterative method for computing the numerical solution of systems of nonlinear equations associated with ODEs, Appl. Math. Comput, 250 (2015) 249-259.

[10] J. M. Ortega, W. C. Rheinbodt, Iterative solution of nonlinear equations in several variables, United Kingdom: Academic press limited 24-28 Oval road, London NW1 7DX, 1970.

[11] D. Budzko, A. Cordero, J. R. Torregrosa, Modifications of Newtons method to extend the convergence domain, Bolet. Sociedad. Espanola. Mat. Aplicada, DOI:10.1007/s40324-014-0020-y.

[12] Eman S. Alaidarous, Malik zaka ullah, Fayyaz Ahmad, and A.S. Al-Fhaid, An Efficient Higher-Order Quasilinearization Method for Solving Nonlinear BVPs,Journal of Applied Mathematics Volume 2013 (2013), Article ID 259371, 11 pages.

[13] S. S. Motsa and S. Shateyi, New Analytic Solution to the Lane-Emden Equation of Index 2, Mathematical Problems in Engineering Volume 2012 (2012), Article ID 614796, 19 pages.

[14] C. A. Ladeia, N. M. L. Romeiro, Numerical Solutions of the 1D Convection-Diffusion-Reaction and the Burgers Equation using Implicit Multi-stage and Finite Element Methods, Integral Methods in Science and Engineering 2013, pp 205-216.

[15] Jang, An integral equation formalism for solving the nonlinear Klein-Gordon equation, Applied Mathematics and Computation, 01/2014, 243, 322-338.

[16] D. Gurarie, K. W. Chow, Vortex arrays for sinh-Poisson equation of two-dimensional fluids: Equilibria and stability, physics of fluids, volume 16, number 9, September 2004.

[17] Brett M. Averick, James M. Ortega, fast solution of nonlinear Poisson-type equations, Argonne national laboratory, 9700 South Cass Avenue, August 1991.

[18] A. H. Bhrawy, An efficient Jacobi pseudospectral approximation for nonlinear complex generalized Zakharov system, Appl. Math. Comput, 247 (2014) 30-46.

[19] S. Abbasbandy, E. Babolian, M. Ashtiani, Numerical solution of the generalized Zakharov equation by homotopy analysis method, Commun. Nonlinear Sci. Numer. Simul, 14 (2009) 4114-4121.

[20] Q. Chang, H. Jiang, A conservative difference scheme for the Zakharov equations, J. Comput. Phys. 113 (1994) 309.

[21] Q. Chang, B. Guo, H. Jiang, Finite difference method for generalized Zakharov equations, Math. Comput. 64 (1995) 537.

[22] M. Javidi, A. Golbabai, Exact and numerical solitary wave solutions of generalized Zakharov equation by the variational iteration method, Chaos Solitons Fract. 36 (2008) 309-313.

[23] W. Bao, F. Sun, G.W. Wei, Numerical methods for the generalized Zakharov system, J. Comput. Phys. 190 (2003) 201-228.

[24] W. Bao, F. Sun, Efficient and stable numerical methods for the generalized and vector Zakharov system, SIAM J. Sci. Comput, 26 (2005) 1057-1088.

[25] M. Dehghan, F. F. Izadi, The spectral collocation method with three different bases for solving a nonlinear partial differential equation arising in modeling of nonlinear waves, Math. Comput. Model, 5 (2011) 1865-1877.

[26] E. H. Doha, A. H. Bhrawy, S. S. Ezz-Eldien, Efficient Chebyshev spectral methods for solving multi-term fractional orders differential equations, Appl. Math. Model, 35 (2011) 5662-5672.

[27] E. H. Doha, A. H. Bhrawy, R. M. Hafez, On shifted Jacobi spectral method for high-order multi-point boundary value problems, Commun. Nonlinear Sci. Numer. Simul, 17 (2012) 3802-3810.

[28] O. R. N. Samadi, E. Tohidi, The Spectral Method for Solving Systems of Volterra Integral Equations, J. Appl. Math. Comput, 40 (2012) 477-497.

[29] E. Tohidi, O. R. N. Samadi, Optimal Control of Nonlinear Volterra Integral Equations via Legendre Polynomials, IMA Journal of Mathematical Control and Information 30 (2013), 67-83.

[30] E. Tohidi, S. Lotfi Noghabi, An efficient Legendre Pseudospectral Method for Solving Nonlinear Quasi Bang-Bang Optimal Control Problems, J. Appl. Math. Stat. Inform, 8 (2012), 73-85.

[31] Z. Wang, B.-Y. Guo, Jacobi rational approximation and spectral method for differential equations of degenerate type, Math. Comput. 77 (2008) 883-907.