Jacobi–Gauss–Lobatto collocation method for the numerical solution of 1 + 1 nonlinear Schrödinger equations

E.H. Doha, A.H. Bhrawy, M.A. Abdelkawy, Robert A. Van Gorder

A Jacobi–Gauss–Lobatto collocation (J-GL-C) method, used in combination with the implicit Runge–Kutta method of fourth order, is proposed as a numerical algorithm for the approximation of solutions to nonlinear Schrödinger equations (NLSE) with initial-boundary data in 1 + 1 dimensions. Our procedure is implemented in two successive steps. In the first one, the J-GL-C is employed for approximating the functional dependence on the spatial variable, using \((N-1)\) nodes of the Jacobi–Gauss–Lobatto interpolation which depends upon two general Jacobi parameters. The resulting equations together with the two-point boundary conditions induce a system of 2\((N-1)\) first-order ordinary differential equations (ODEs) in time. In the second step, the implicit Runge–Kutta method of fourth order is applied to solve this temporal system. The proposed J-GL-C method, used in combination with the implicit Runge–Kutta method of fourth order, is employed to obtain highly accurate numerical approximations to four types of NLSE, including the attractive and repulsive NLSE and a Gross–Pitaevskii equation with space-periodic potential. The numerical results obtained by this algorithm have been compared with various exact solutions in order to demonstrate the accuracy and efficiency of the proposed method. Indeed, for relatively few nodes used, the absolute error in our numerical solutions is sufficiently small.

**1. Introduction**

Several numerical methods have been intensively investigated for the numerical solution of time dependent partial differential equations (PDEs) (see for instance, [1–5]), particularly the NLSE. The discontinuous Galerkin method was proposed in [6] for the numerical solution of single and coupled NLSE. Dehghan and Taleei [7] proposed the Chebyshev pseudospectral differentiation matrices to reduce the solution of NLSE to a system of algebraic equations and very good results were obtained concerning the behavior of the pseudo-spectral solutions. In [8], the combination of multi-domain Chebyshev collocation methods and an explicit Runge–Kutta scheme of order four for solving the coupled version of NLSE was discussed. In [9], a fourth order explicit scheme for solving the coupled version of NLSE was considered, while the Galerkin method for the same problem was considered in [10]. Recently, Xu and Zhang [11] introduced and developed alternating direction implicit methods for numerical treatment of two-dimensional cubic NLSE with studying the stability analysis. Meanwhile, the Pade approximation was considered in [12] for the solution of a higher-order NLSE with fourth order dispersion. More

© 2014 Elsevier Inc. All rights reserved.
recently, Duan and Rong [13] discussed and developed a multi-quadric quasi-interpolation scheme for solving NLSE. Many other methods have been studied in the literature; see, for example, [14–16].

Spectral methods (see [17–21]) are often an efficient and highly accurate scheme when compared with local methods. There are three primary types of spectral methods based on the choice of test functions, namely, the Galerkin, tau and collocation methods. In these methods, one reduces the original PDEs into a system of ODEs (often in the time variable) by assuming a functional dependence on the spatial part of the equation. In such algorithms, one then solves for the time evolution of the solutions. It is within the realm of nonlinear and variable coefficient problems that the advantages of these collocation methods become apparent. Indeed, the projection operators involved in the Galerkin and tau methods lead to complications in deriving the required system of ODEs, from which the final solution is obtained. In both these methods, we require that the projection of the residual on some polynomial space be zero. In contrast, the collocation method obtains the system of ODEs by requiring that the residual vanishes at a certain set of grid points, typically some set of Gauss-type points. As a result, while the Galerkin and tau methods may become untenable for strongly nonlinear problems, the collocation method has been shown in many cases to deal with linear, variable coefficient, and nonlinear problems with equal ease [22–27]. Along these lines, a number of related methods have been proposed for solving related equations [28–34].

The Jacobi polynomials are a class of orthogonal polynomials, which satisfy an orthogonality condition on the interval [−1, 1] with respect to the weight function (1−x)α(1+x)β. Several orthogonal polynomials can obtained as special case of the Jacobi polynomials, such as the Legendre, Chebyshev, and Gegenbauer [35]. In recent years, the use of Jacobi polynomials for solving differential equations through collocation techniques has gained increasing popularity due to the ability to obtain the solution in terms of the general Jacobi parameters α and β (see [36,37]).

The main concern of this paper is to extend the J-GL-C method to the solution of various types of NLSE. In particular, we shall consider the general initial-boundary value problem

\[ i\hbar \psi (y, t) + \partial_y \psi (y, t) + 2y |\psi (y, t)|^2 \psi (y, t) - 2\delta R(y, t)\psi (y, t) = 0, \tag{1.1} \]

where

\[ (y, t) \in D \times [0, T], \quad D = \{ y : A \leq y \leq B \}, \]

with the boundary-initial conditions

\[ \psi (A, t) = \zeta_1 (t), \quad \psi (B, t) = \zeta_2 (t), \tag{1.2} \]

\[ \psi (y, 0) = \xi_1 (y), \quad y \in D. \tag{1.3} \]

Here we have maintained a general potential \( R(y, t) \) and the cubic nonlinearity. This type of equation is also referred to as a Gross–Pitaevskii equation when we have non-zero \( R(y, t) \). The nonlinear PDEs of type (1.1) are collocated only for the space variable at \((N - 1)\) points, for suitable collocation points. We use the \((N - 1)\) nodes of the Jacobi–Gauss–Lobatto interpolation, which depends upon the two general parameters \( \theta \) and \( \vartheta \). These equations, together with two-point boundary conditions, constitute system of \((2N - 2)\) ODEs in the time variable. This system can then be solved by the implicit Runge–Kutta method of fourth order. Finally, the accuracy of the proposed method is demonstrated by four test problems, which correspond to physically meaningful cases. To the best of our knowledge, there are no results on the J-GL-C method for solving NLSE.

This paper is arranged as follows. In Section 2, we derive the numerical method. First, we introduce some properties of Jacobi polynomials and their associated recursion relations. Then the Gauss–Lobatto collocation technique for a general form of the NLSE obtained using the Jacobi polynomials, and the implicit Runge–Kutta method of fourth order is outlined for the resulting system of ODEs. In Section 3, the proposed method is applied to four specific NLSE. It is here where we demonstrate the usefulness of the method, in terms of efficiency and accuracy. Indeed, for relatively few nodes, we obtain rather good results. In fact, for all examples considered, we obtain errors of order \(10^{-6}\) or lower using only nine nodes. Finally, some concluding remarks are given in Section 4.

2. Derivation of the numerical method

In the present section, we derive the numerical method. First, we remind the reader of some useful properties of Jacobi polynomials. Then, we develop the J-GL-C method to numerically solve the NLSE (1.1). We conclude this section with the implicit Runge–Kutta method used to solve the time-dependent ODEs.

2.1. Some properties of the Jacobi polynomials

We collected in this section some basic knowledge of Jacobi polynomials that are most relevant to spectral approximations [35]. It includes Legendre and Chebyshev polynomials as two special cases, so it is worthwhile work with general Jacobi polynomials. A basic property of the Jacobi polynomials is that they are the eigenfunctions to a singular Sturm–Liouville problem:
\begin{equation}
(1-x^2)\phi''(x) + [\theta - \theta + (\theta + \theta + 2)x] \phi'(x) + n(\alpha + \theta + \theta + 1)\phi(x) = 0.
\end{equation}

The following recurrence relation generate the Jacobi polynomials:
\begin{align*}
J_{k+1}^{(\alpha, \beta)}(x) &= (a_k^{(\alpha, \beta)} x - b_k^{(\alpha, \beta)}) J_k^{(\alpha, \beta)}(x) - c_k^{(\alpha, \beta)} J_{k-1}^{(\alpha, \beta)}(x), \quad k \geq 1, \\
J_0^{(\alpha, \beta)}(x) &= 1, \quad J_1^{(\alpha, \beta)}(x) = \frac{1}{2}(\theta + \theta + 2)x + \frac{1}{2}(\theta - \theta),
\end{align*}

where
\begin{align*}
a_k^{(\alpha, \beta)} &= \frac{(2k + \theta + \theta + 1)(2k + \theta + \theta + 2)}{2(k+1)(k+\theta+\theta+1)}, \\
b_k^{(\alpha, \beta)} &= \frac{((\theta^2 - \theta^2)(2k + \theta + \theta + 1)}{2(k+1)(k+\theta+\theta+1)(2k+\theta+\theta+2)} \\
c_k^{(\alpha, \beta)} &= \frac{(k+\theta)(k+\theta+1)(2k+\theta+\theta+2)}{(k+1)(k+\theta+\theta+1)(2k+\theta+\theta+2)}.
\end{align*}

The Jacobi polynomials are satisfying the following identities:
\begin{align}
J_k^{(\alpha, \beta)}(-x) &= (-1)^k J_k^{(\alpha, \beta)}(x), \\
J_k^{(\alpha, \beta)}(-1) &= \frac{(-1)^k \Gamma(k+\theta+1)}{k! \Gamma(\theta+1)}.
\end{align}

Moreover, the $q$ derivative of Jacobi polynomials of degree $k$ ($J_k^{(\alpha, \beta)}(x)$), can be obtained from:
\begin{equation}
D^q J_k^{(\alpha, \beta)}(x) = \frac{\Gamma(j + \theta + \theta + q + 1)}{2^q \Gamma(j + \theta + \theta + 1)} J_{k-1}^{(\alpha+q, \beta+q)}(x).
\end{equation}

Let $w^{(\alpha, \beta)}(x) = (1-x)^{\theta}(1+x)^{\beta}$, then we define the weighted space $L^2_w(\omega, \varphi)$, as usual. The inner product and the norm of $L^2_w(\omega, \varphi)$ with respect to the weight function are defined as:
\begin{equation}
(u, v)_{\omega, \varphi} = \int_1^{-1} u(x) v(x) w^{(\alpha, \beta)}(x) \, dx, \quad \|u\|_{\omega, \varphi} = (u, u)_{\omega, \varphi}^{\frac{1}{2}}.
\end{equation}

The set of Jacobi polynomials forms a complete $L^2_w(\omega, \varphi)$-orthogonal system, and
\begin{equation}
\|J_k^{(\alpha, \beta)}\|_{\omega, \varphi} = h_k = \frac{2^{\theta+\beta+1} \Gamma(k+\theta+1) \Gamma(k+\theta+1)}{(2k+\theta+\theta+1) \Gamma(k+\theta+\theta+1)}.
\end{equation}

### 2.2. Jacobi spectral collocation method

The main objective of this section is to develop the J-GL-C method to numerically solve the NLSE (1.1).

In order to begin the numerical solution, we put the complex functions $\psi$ in real and imaginary parts as:
\begin{equation}
\psi(y, t) = u(y, t) + i v(y, t),
\end{equation}

where $u$ and $v$ are real functions, from Eqs. (1.1) and (2.6), we get
\begin{align}
i \left( \partial_t u(y, t) + \partial_{yy} v(y, t) + 2\gamma (u^2(y, t) + v^2(y, t)) v(y, t) - 2\delta R(y, t) v(y, t) \right) \\
- (\partial_t v(y, t) - \partial_{yy} u(y, t) - 2\gamma (u^2(y, t) + v^2(y, t)) u(y, t) + 2\delta R(y, t) u(y, t) \right) &= 0,
\end{align}

which can be separated as coupled time-dependent nonlinear PDEs,
\begin{align}
\partial_t u(y, t) + \partial_{yy} v(y, t) + 2\gamma (u^2(y, t) + v^2(y, t)) v(y, t) - 2\delta R(y, t) v(y, t) &= 0, \\
\partial_t v(y, t) - \partial_{yy} u(y, t) - 2\gamma (u^2(y, t) + v^2(y, t)) u(y, t) + 2\delta R(y, t) u(y, t) &= 0,
\end{align}

with the boundary-initial conditions
\begin{align}
u(A, t) &= g_1(t), \quad u(B, t) = g_2(t), \\
v(A, t) &= g_3(t), \quad v(B, t) = g_4(t), \\
u(y, 0) &= f_1(y), \quad v(y, 0) = f_2(y), \quad y \in D.
\end{align}
Now, suppose the change of variables
\[
x = \frac{2}{B - A} y + \frac{A + B}{A - B} \quad u(y, t) = w(x, t), \quad v(y, t) = z(x, t),
\]
which will be used to transform problem (1.1)–(1.3) into another one in the classical interval, \([-1, 1]\), for the space variable, to directly implement collocation method based on Jacobi family defined on \([-1, 1]\),
\[
\begin{align*}
\partial_t w(x, t) + \left( \frac{2}{B - A} \right)^2 \partial_{xx} z(x, t) + 2\gamma \left( w^2(x, t) + z^2(x, t) \right) z(x, t) - 2\delta r(x, t) z(x, t) &= 0, \\
\partial_t z(x, t) - \left( \frac{2}{B - A} \right)^2 \partial_{xx} w(x, t) - 2\gamma \left( w^2(x, t) + z^2(x, t) \right) w(x, t) + 2\delta r(x, t) w(x, t) &= 0,
\end{align*}
\]
where
\[(x, t) \in D^* \times [0, T], \quad D^* = \{ x : -1 \leq x \leq 1 \},
\]
with the boundary-initial conditions
\[
\begin{align*}
w(-1, t) &= g_5(t), \quad w(1, t) = g_6(t), \\
z(-1, t) &= g_7(t), \quad z(1, t) = g_8(t), \\
w(x, 0) &= f_3(x), \quad z(x, 0) = f_4(x), \quad x \in D^*.
\end{align*}
\]

The main advantage of the use of the node points of Jacobi–Gauss–Lobatto quadrature is that the distribution of these points in \([-1, 1]\) can be controlled by the two Jacobi parameters, \(\theta \) and \(\delta\). Now, we outline the main step of implementing J-GL-C method for reducing nonlinear coupled system (2.10)–(2.11) to system of ODEs. The expansion of functions \(w(x, t)\) and \(z(x, t)\), using the Jacobi polynomial, \(j^{(\theta, \delta)}\), takes the form
\[
\begin{align*}
w(x, t) &= \sum_{j=0}^{N} a_j(t) j^{(\theta, \delta)}(x), \\
z(x, t) &= \sum_{j=0}^{N} b_j(t) j^{(\theta, \delta)}(x),
\end{align*}
\]
and in virtue of (2.4)–(2.5), we deduce that
\[
\begin{align*}
a_j(t) &= \frac{1}{h_j} \int_{-1}^{1} w(x, t) w^{(\theta, \delta)}(x) j^{(\theta, \delta)}(x) dx, \\
b_j(t) &= \frac{1}{h_j} \int_{-1}^{1} z(x, t) w^{(\theta, \delta)}(x) j^{(\theta, \delta)}(x) dx.
\end{align*}
\]

We can circumvent the need for evaluating the previous integrals, by using Jacobi–Gauss–Lobatto quadrature formula to approximate the integrals. For any \(\phi \in S_{2N+1}(-1, 1)\),
\[
\int_{-1}^{1} w^{(\theta, \delta)}(x) \phi(x) dx = \sum_{j=0}^{N} w^{(\theta, \delta)}(x_{N,j}) \phi(x_{N,j}),
\]
where \(S_{N}(-1, 1)\) is the set of polynomials of degree less than or equal to \(N\), \(x_{N,j}^{(\theta, \delta)}(0 \leq j \leq N)\) and \(w^{(\theta, \delta)}(0 \leq j \leq N)\) are the nodes and the corresponding Christoffel numbers of the Jacobi–Gauss–Lobatto quadrature formula on the interval \((-1, 1)\), respectively.

Due to the orthogonality property of Jacobi polynomials (2.4), the expansion coefficients \(a_j(t), b_j(t)\) in terms of the solution at the Jacobi–Gauss–Lobatto grid points can be approximated by
\[
\begin{align*}
a_j(t) &= \frac{1}{h_j} \sum_{i=0}^{N} j^{(\theta, \delta)}(x_{N,i}^{(\theta, \delta)}) \sigma_{N,i}^{(\theta, \delta)} w(x_{N,i}^{(\theta, \delta)}, t), \\
b_j(t) &= \frac{1}{h_j} \sum_{i=0}^{N} j^{(\theta, \delta)}(x_{N,i}^{(\theta, \delta)}) \sigma_{N,i}^{(\theta, \delta)} z(x_{N,i}^{(\theta, \delta)}, t).
\end{align*}
\]
Accordingly, (2.12)–(2.13) can be given by

\[
\begin{align*}
    w(x, t) &= \sum_{i=0}^{N} \left( \sum_{j=0}^{N} \frac{1}{h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x)^{(0, \theta)}_{N,i} \right) w(x_{N,i}^{(0, \theta)}, t), \\
    z(x, t) &= \sum_{i=0}^{N} \left( \sum_{j=0}^{N} \frac{1}{h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x)^{(0, \theta)}_{N,i} \right) z(x_{N,i}^{(0, \theta)}, t).
\end{align*}
\]

(2.17)

Furthermore, if we differentiate (2.17) once using (2.3), and evaluate it at all J-GL-C points, it is easy to compute the first spatial partial derivative in terms of the values at these collocation points as

\[
\begin{align*}
    w_x(x_{N,i}^{(0, \theta)}, t) &= \sum_{i=0}^{N} A_{ni} w(x_{N,i}^{(0, \theta)}, t), \\
    z_x(x_{N,i}^{(0, \theta)}, t) &= \sum_{i=0}^{N} B_{ni} z(x_{N,i}^{(0, \theta)}, t), \quad n = 0, 1, \ldots, N,
\end{align*}
\]

where

\[
\begin{align*}
    A_{ni} &= \sum_{j=0}^{N} \frac{j + \theta + \vartheta + 1}{2h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) \sigma_j^{(0, \theta)}, \\
    B_{ni} &= \sum_{j=0}^{N} \frac{j + \theta + \vartheta + 1}{2h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) \sigma_j^{(0, \theta)}.
\end{align*}
\]

(2.19)

Similar steps can be applied to the second spatial partial derivative to get

\[
\begin{align*}
    w_{xx}(x_{N,i}^{(0, \theta)}, t) &= \sum_{i=0}^{N} C_{ni} w(x_{N,i}^{(0, \theta)}, t), \\
    z_{xx}(x_{N,i}^{(0, \theta)}, t) &= \sum_{i=0}^{N} D_{ni} z(x_{N,i}^{(0, \theta)}, t), \quad n = 0, 1, \ldots, N,
\end{align*}
\]

where

\[
\begin{align*}
    C_{ni} &= \sum_{j=0}^{N} \frac{(j + \theta + \vartheta + 2)(j + \theta + \vartheta + 1)}{4h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) \sigma_j^{(0, \theta)}, \\
    D_{ni} &= \sum_{j=0}^{N} \frac{(j + \theta + \vartheta + 2)(j + \theta + \vartheta + 1)}{4h_j} j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) j_j^{(0, \theta)} (x_{N,i}^{(0, \theta)}) \sigma_j^{(0, \theta)}.
\end{align*}
\]

(2.21)

For the spectral J-GL-C method, the residual of (1.1) is set to zero at the $N-1$ Jacobi–Gauss–Lobatto points. Also, the boundary conditions (1.2) can be imposed at the two collocation points $-1$ and $1$ in the expansion of functions. Therefore, adopting (2.8)–(2.11), we may write (1.1)–(1.2) in the form

\[
\begin{align*}
    \dot{w}_n(t) &= - \left( \frac{2}{B - A} \right)^2 \sum_{i=0}^{N} D_{ni} z_i(t) - 2\gamma \left( w_n^2(t) + z_n^2(t) \right) z_n(t) + 2\delta r_n(t) z_n(t), \\
    \dot{z}_n(t) &= \left( \frac{2}{B - A} \right)^2 \sum_{i=0}^{N} C_{ni} w_i(t) + 2\gamma \left( w_n^2(t) + z_n^2(t) \right) w_n(t) - 2\delta r_n(t) w_n(t), \quad n = 1, 2, \ldots, N - 1,
\end{align*}
\]

(2.22)

where

\[
\begin{align*}
    w_k(t) &= w(x_{N,k}^{(0, \theta)}, t), \quad z_k(t) = z(x_{N,k}^{(0, \theta)}, t), \quad k = 1, \ldots, N - 1, \quad n = 1, \ldots, N - 1.
\end{align*}
\]

This provides two $(N-1)$ system of first order ODEs in the expansion coefficients $a_j(t)$, namely
\[ w_n(t) = - \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} D_{ni}z_i(t) + d_n(t) \right) - 2\gamma \left( w_n^2(t) + z_n^2(t) \right) z_n(t) + 2\delta r_n(t) z_n(t), \]
\[ \dot{z}_n(t) = \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} C_{ni}w_i(t) + \tilde{d}_n(t) \right) + 2\gamma \left( w_n^2(t) + z_n^2(t) \right) w_n(t) - 2\delta r_n(t) w_n(t), \]
where
\[ d_n(t) = D_{n0}g_7(t) + D_{nN}g_8(t), \]
\[ \tilde{d}_n(t) = C_{n0}g_5(t) + C_{nN}g_6(t). \]

We get the following two system of nonlinear ODEs:
\[ \dot{w}_n(t) = - \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} D_{ni}z_i(t) + d_n(t) \right) - 2\gamma \left( w_n^2(t) + z_n^2(t) \right) z_n(t) + 2\delta r_n(t) z_n(t), \]
\[ \dot{z}_n(t) = \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} C_{ni}w_i(t) + \tilde{d}_n(t) \right) + 2\gamma \left( w_n^2(t) + z_n^2(t) \right) w_n(t) - 2\delta r_n(t) w_n(t), \]
\[ n = 1, 2, \ldots, N - 1, \]
subject to the initial values
\[ w_n(0) = f_3(x_{n,1}^{(\theta, \phi)}), \]
\[ z_n(0) = f_4(x_{n,1}^{(\theta, \phi)}), \quad n = 1, \ldots, N - 1. \]

Finally, (2.24)–(2.25) can be rewritten into two matrix form of \( N - 1 \) first order ODEs with their vectors of initial values:
\[ \mathbf{w}(t) = F(t, \mathbf{w}(t), \mathbf{z}(t)), \]
\[ \mathbf{z}(t) = G(t, \mathbf{w}(t), \mathbf{z}(t)), \]
\[ \mathbf{w}(0) = \mathbf{f}_3, \]
\[ \mathbf{z}(0) = \mathbf{f}_4, \]
where
\[ \mathbf{w}(t) = [\dot{w}_1(t), \dot{w}_2(t), \ldots, \dot{w}_{N-1}(t)]^T, \]
\[ \mathbf{z}(t) = [\dot{z}_1(t), \dot{z}_2(t), \ldots, \dot{z}_{N-1}(t)]^T, \]
\[ \mathbf{f}_3 = [f_3(x_{n,1}^{(\theta, \phi)}), f_3(x_{n,2}^{(\theta, \phi)}), \ldots, f_3(x_{N-1,1}^{(\theta, \phi)})]^T, \]
\[ \mathbf{f}_4 = [f_4(x_{n,1}^{(\theta, \phi)}), f_4(x_{n,2}^{(\theta, \phi)}), \ldots, f_4(x_{N-1,1}^{(\theta, \phi)})]^T, \]
\[ F(t, \mathbf{w}(t), \mathbf{z}(t)) = [F_1(t, \mathbf{w}(t), \mathbf{z}(t)), F_1(t, \mathbf{w}(t), \mathbf{z}(t)), \ldots, F_{N-1}(t, \mathbf{w}(t), \mathbf{z}(t))]^T, \]
and
\[ G(t, \mathbf{w}(t), \mathbf{z}(t)) = [G_1(t, \mathbf{w}(t), \mathbf{z}(t)), G_1(t, \mathbf{w}(t), \mathbf{z}(t)), \ldots, G_{N-1}(t, \mathbf{w}(t), \mathbf{z}(t))]^T. \]

Here,
\[ F_n(t, \mathbf{w}(t), \mathbf{z}(t)) = - \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} D_{ni}z_i(t) + d_n(t) \right) - 2\gamma \left( w_n^2(t) + z_n^2(t) \right) z_n(t) + 2\delta r_n(t) z_n(t), \]
\[ G_n(t, \mathbf{w}(t), \mathbf{z}(t)) = \left( \frac{2}{B - A} \right)^2 \left( \sum_{i=1}^{N-1} C_{ni}w_i(t) + \tilde{d}_n(t) \right) + 2\gamma \left( w_n^2(t) + z_n^2(t) \right) w_n(t) - 2\delta r_n(t) w_n(t). \]

The two \( N - 1 \) dimensional system of ODEs (2.26) can be combined to be written in a \( 2N - 2 \) system of ODEs and then can be solved by using an implicit Runge-Kutta formula of fourth order.
2.3. Implicit Runge–Kutta (IRK) method

The implicit Runge–Kutta (IRK) method is one of the suitable methods for solving a system of ODEs of first order like we have obtained above. Since the IRK methods have excellent stability properties which reduce computational costs. Consider

\[
\begin{align*}
\dot{u}(t) &= H(t, u(t)) \\
u(0) &= f.
\end{align*}
\]

Then, the iterative solution is

\[
\begin{align*}
U_{i,j} &= u_i + \Delta t \sum_{l=1}^{m} a_{lj} H(t_i + c_l \Delta t, U_{i,l}), \quad j = 1, \ldots, m, \\
u_{i+1} &= u_i + \Delta t \sum_{j=1}^{m} b_j H(t_i + c_j \Delta t, U_{i,j}), \quad \forall j
\end{align*}
\]

where \(\Delta t\) is the step size, \(t_i = t_0 + i \Delta t\) and \(u_i\) is approximations to the exact solution \(u(t_i)\). Moreover, \(a_{ij}, b_j\) and \(c_j\) are real coefficients, \(i, j = 1, 2, \ldots, m\), which can be represented by means of the Butcher array,

\[
\begin{array}{c|ccc}
    & a_{11} & \cdots & a_{1s} \\
\hline
    c_1 & \vdots & \ddots & \vdots \\
    \vdots & \ddots & \ddots & \vdots \\
    c_s & a_{s1} & \cdots & a_{ss} \\
    & b_1 & \cdots & b_s
\end{array}
\]

3. Numerical examples

This section considers four numerical examples to show the accuracy and applicability of the proposed method in the present paper. Comparison of the results obtained by various choices of Jacobi parameters \(\theta\) and \(\vartheta\) reveals that the present method is very effective and convenient for various choices of \(\theta\) and \(\vartheta\). We focus on some well-known equations from mathematical physics in order to demonstrate the method. First we consider an attractive NLSE \([38,39]\), and for our next example we consider a repulsive NLSE \([38,39]\). Third, we consider the Gross–Pitaevski equation \([40,41]\), which describes the ground state of a quantum system of identical bosons. We assume that the potential is periodic in space, consistent with a Bose–Einstein condensate. Finally, we consider a NLSE with a linear instability added. In each of these examples, we shall highlight the accuracy and efficiency of the method.

3.1. Numerical solution for an attractive NLSE

We first consider an attractive NLSE \([39]\) in the form:

\[
i \hat{\partial}_{t} \psi + \partial_{xx} \psi + 2|\psi|^{2} \psi = 0, \quad (y, t) \in D \times \mathbb{R}
\]

with the boundary-initial conditions

\[
\begin{align*}
\psi(A, t) &= u(A, t) + iv(A, t) = e^{i(t+b)}, \\
\psi(B, t) &= u(B, t) + iv(B, t) = e^{i(t+B)}, \\
\psi(x, 0) &= u(x, 0) + iv(x, 0) = e^{i(x)}.
\end{align*}
\]

In this example, we consider \(D = [A, B] = [-1, 1]\). Furthermore, we shall approximate the time domain by \([0, T]\) for numerical purposes. The difference between the measured or inferred value of a numerical solution and its actual value (absolute error) is given by (for real part, imaginary part, and complex modulus)

\[
\begin{align*}
E_1(x, t) &= |u(x, t) - \tilde{u}(x, t)|, \\
E_2(x, t) &= |v(x, t) - \tilde{v}(x, t)|, \\
E_3(x, t) &= |\psi(x, t) - \tilde{\psi}(x, t)|,
\end{align*}
\]

where \(\psi(x, t) = u(x, t) + iv(x, t)\) and \(\tilde{\psi}(x, t) = \tilde{u}(x, t) + i\tilde{v}(x, t)\) are the exact solution and the approximate solution at the point \((x, t)\), respectively. It may be shown that the error norms are related. Indeed,

\[
E_3(x, t) = |u(x, t) - \tilde{u}(x, t) + i(v(x, t) - \tilde{v}(x, t))| = \sqrt{E_1(x, t)^2 + E_2(x, t)^2}.
\]
Table 1
Maximum absolute errors with various choices of $N$, $\theta$ and $\vartheta$ for Example 1.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\theta$</th>
<th>$\vartheta$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$\theta$</th>
<th>$\vartheta$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>$3.81 \times 10^{-1}$</td>
<td>$5.73 \times 10^{-1}$</td>
<td>$6.21 \times 10^{-1}$</td>
<td>$-\frac{1}{2}$</td>
<td>$-\frac{1}{2}$</td>
<td>$2.30 \times 10^{-1}$</td>
<td>$4.83 \times 10^{-1}$</td>
<td>$4.83 \times 10^{-2}$</td>
</tr>
<tr>
<td>6</td>
<td>$3.62 \times 10^{-5}$</td>
<td>$15.70 \times 10^{-5}$</td>
<td>$16.04 \times 10^{-5}$</td>
<td>$5.75 \times 10^{-5}$</td>
<td>$3.65 \times 10^{-5}$</td>
<td>$5.75 \times 10^{-5}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$4.62 \times 10^{-8}$</td>
<td>$22.59 \times 10^{-8}$</td>
<td>$22.74 \times 10^{-8}$</td>
<td>$4.93 \times 10^{-8}$</td>
<td>$8.30 \times 10^{-8}$</td>
<td>$9.21 \times 10^{-8}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moreover, the maximum absolute error is given by

$$M_1 = \text{Max}\{E_1(x, t) : \forall (x, t) \in D \times [0, T]\},$$

$$M_2 = \text{Max}\{E_2(x, t) : \forall (x, t) \in D \times [0, T]\},$$

$$M_3 = \text{Max}\{E_3(x, t) : \forall (x, t) \in D \times [0, T]\}.$$  \hspace{1cm} (3.4)

Clearly, $M_3 \leq \sqrt{M_1^2 + M_2^2}$. The maximum absolute errors of $u(x, t)$, $v(x, t)$ and $\psi(x, t)$ related to the problem (3.1)–(3.2) are presented in Table 1 using the J-GL-C method with various choices of $N$, $\theta$ and $\vartheta$. The errors are calculated through a comparison with the exact solution

$$\psi(x, t) = e^{i(x+t)}.$$ \hspace{1cm} (3.5)

We see in this table that the results are very accurate for even small choices of the number of nodes, $N$. In Fig. 1, we see that the absolute errors $E_1(x, t)$ and $E_2(x, t)$ are very small, despite the relatively small number of grid points used. This numerical experiment demonstrates the utility of the method.
Consider the repulsive NLSE [39]

\[ i \psi_t + \partial_{xx} \psi - 2\gamma |\psi|^2 \psi = 0, \]  

(3.6)

with the boundary-initial conditions

\[
\begin{align*}
\psi(-1,t) &= u(-1,t) + iv(-1,t) = e^{i(-1-3t)}, \\
\psi(1,t) &= u(1,t) + iv(1,t) = e^{i(1-3t)}, \\
\psi(x,0) &= u(x,0) + iv(x,0) = e^{ix}.
\end{align*}
\]

(3.7)

Note that the exact solution is given by

\[ \psi(x,t) = e^{i(x-3t)}. \]  

(3.8)

Table 2 lists the maximum absolute errors of \( u(x,t) \), \( v(x,t) \) and \( \psi(x,t) \) of problem (3.6)–(3.7) using the J-GL-C method for with various choices of \( N \), \( \theta \) and \( \vartheta \). The numerical results presented in this table show that the results are very accurate for small value of \( N \). Fig. 2 demonstrates that the absolute errors \( E_1(x,t) \) and \( E_2(x,t) \) are very small for even the small number of grid points taken.

3.3. Nonlinear Gross–Pitaevskii equation with space-periodic potential

Consider the complex nonlinear Gross–Pitaevskii (CNGP) equation with a space-periodic potential [39–41] given by

\[ i \psi_t + \frac{1}{2} \partial_{xx} \psi - |\psi|^2 \psi - \cos(x) \psi = 0, \]  

(3.9)

with the boundary-initial conditions

\[
\begin{align*}
\psi(A,t) &= u(A,t) + iv(A,t) = e^{-3n} \sin(A), \\
\psi(B,t) &= u(B,t) + iv(B,t) = e^{3n} \sin(B), \\
\psi(x,0) &= u(x,0) + iv(x,0) = \sin(x).
\end{align*}
\]

(3.10)

The exact solution of CNGP equation is given by

\[ \psi(x,t) = e^{-\frac{3n}{2}} \sin(x). \]  

(3.11)

To demonstrate the accuracy of the proposed method, in Table 3 we introduce maximum absolute errors of \( u(x,t) \), \( v(x,t) \) and \( \psi(x,t) \) of CNGP equation, using the J-GL-C method with various choices of \( \theta \) and \( \vartheta \) and \( N \). We see that the approximate solutions \( \tilde{u}(x,t) \) (\( \tilde{v}(x,t) \)) and the exact solution \( u(x,t) \) (\( v(x,t) \)) are coincided for different values of \( t \) and \( x \).

3.4. An attractive NLSE with linear instability

We consider an attractive NLSE [39] with an additional linear term

\[ i \psi_t + \partial_{xx} \psi + 2|\psi|^2 \psi - 2\gamma \psi = 0, \]  

(3.12)

with the boundary-initial conditions
The absolute error (a) $E_1(x, t)$ and (b) $E_2(x, t)$ of problem (3.6) where $\theta = \vartheta = -\frac{1}{2}$ and $N = 10$.

Table 3
Maximum absolute errors with various choices of $N$, $\theta$ and $\vartheta$ for Example 3.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$\theta$</th>
<th>$\vartheta$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
<th>$\theta$</th>
<th>$\vartheta$</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$M_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>$4.29 \times 10^{-2}$</td>
<td>$4.00 \times 10^{-2}$</td>
<td>$5.87 \times 10^{-2}$</td>
<td>$-\frac{1}{2}$</td>
<td>$-\frac{1}{2}$</td>
<td>$5.46 \times 10^{-2}$</td>
<td>$4.00 \times 10^{-2}$</td>
<td>$5.87 \times 10^{-2}$</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>$3.84 \times 10^{-6}$</td>
<td>$3.28 \times 10^{-6}$</td>
<td>$3.97 \times 10^{-6}$</td>
<td></td>
<td></td>
<td>$4.90 \times 10^{-6}$</td>
<td>$5.56 \times 10^{-6}$</td>
<td>$6.84 \times 10^{-6}$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>$8.32 \times 10^{-8}$</td>
<td>$7.98 \times 10^{-8}$</td>
<td>$10.49 \times 10^{-8}$</td>
<td></td>
<td></td>
<td>$3.05 \times 10^{-7}$</td>
<td>$1.42 \times 10^{-7}$</td>
<td>$3.05 \times 10^{-7}$</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>$5.25 \times 10^{-2}$</td>
<td>$4.66 \times 10^{-2}$</td>
<td>$7.02 \times 10^{-2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$4.29 \times 10^{-2}$</td>
<td>$4.00 \times 10^{-2}$</td>
<td>$5.87 \times 10^{-2}$</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>$3.67 \times 10^{-6}$</td>
<td>$9.26 \times 10^{-6}$</td>
<td>$9.26 \times 10^{-6}$</td>
<td></td>
<td></td>
<td>$3.97 \times 10^{-6}$</td>
<td>$5.30 \times 10^{-6}$</td>
<td>$5.40 \times 10^{-6}$</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>$2.11 \times 10^{-7}$</td>
<td>$1.69 \times 10^{-7}$</td>
<td>$2.43 \times 10^{-7}$</td>
<td></td>
<td></td>
<td>$1.01 \times 10^{-7}$</td>
<td>$7.13 \times 10^{-8}$</td>
<td>$1.24 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Maximum absolute errors of $u(x, t)$, $v(x, t)$ and $\psi(x, t)$ related to (3.12)–(3.13) are introduced in Table 4 using J-GL-C method for with various choices of $N$, $\theta$ and $\vartheta$, where $\gamma = 1$ compared with the exact solution

$$\psi(A, t) = u(A, t) + iv(A, t) = e^{i(A + (1 - 2\gamma)t)},$$
$$\psi(B, t) = u(B, t) + iv(B, t) = e^{i(B + (1 - 2\gamma)t)},$$
$$\psi(x, 0) = u(x, 0) + iv(x, 0) = e^{ix}.$$ (3.13)

Again and from Table 4, we clearly observe that the global scheme (J-GL-C) is accurate for even small values of $N$.

4. Conclusion

We have proposed an accurate numerical algorithm based on the J-GL-C method for solving complex NLSE. Our procedure is implemented in two successive steps. First, the J-GL-C method is employed for approximating the functional dependence on the spatial variable, using $(N - 1)$ nodes of the Jacobi–Gauss–Lobatto interpolation which depends upon two general Jacobi parameters. The resulting equations together with the two-point boundary conditions give us a system of $2(N - 1)$
first-order ODEs in the time variable. In the second part of the solution method, the implicit Runge–Kutta method of fourth order is applied to solve this system of $2(N-1)$ ODEs in time. Note that it is possible to use other implicit Runge–Kutta formulas, say an implicit Runge–Kutta method of order greater than four, or even a second-order method.

The main advantage of the use of the node points of Jacobi–Gauss–Lobatto quadrature as collocation points is that the distribution of these points in a specified interval can be tuned by the two Jacobi parameters, $\theta$ and $\vartheta$. Through appropriate selection of these parameters, we may recover some of the well-known spectral collocation approximations. Namely, the Legendre and Chebyshev pseudo-spectral approximations can be derived as special cases from our algorithm if we suitably choose the corresponding spacial cases of Jacobi parameters $\theta$ and $\vartheta$.

From the numerical results given in Section 3, it has been concluded that the obtained results are excellent in terms of accuracy for all tested problems. For each of the examples given, we have been able to construct solutions with absolute errors between $10^{-7}$ and $10^{-6}$, and greater accuracy may be obtained if we add more nodes. Note that the initial data considered has always been of modulus unity, so the relative errors should be of the same order of magnitude as the absolute errors. Concerning the efficiency of the method, note that we have obtained approximate numerical solutions of a desirable accuracy by taking only $N = 10$ (i.e., 9 nodes). Considering the size of the domains considered, this is approximately one node per 0.2 units of space. Considering the rather large average spatial gap, the results are rather good. Of course, adding more nodes will increase the accuracy further, and one should be able to employ this method in order to determine solutions of a desired accuracy.

With this paper we have outlined the application of a J-GL-C method for solving complex NLSE. In principle, this algorithm may be extended to related problems, such as to coupled complex nonlinear Schrödinger equations. One might also consider higher order dispersive equations. We should note that, as a numerical method, we are restricted to solving problems over a finite domain. Hence, this method is particularly well suited for boundary value problems with finite spatial domains.

Acknowledgement

R.A.V. supported in part by NSF grant # 1144246.

References


