Chebyshev and Legendre via Galerkin Method

for Solving KdV Burgers’ Equation

H. E. Fathy1, M. El-Kady2,* and Salah M. El-Sayed3

1 Department of Mathematics, Faculty of Science, Benha University, Benha, Egypt
2 Department of Mathematics, Faculty of Science, Helwan University, Cairo, Egypt
3 Department of Scientific Computing, Faculty of Computers and Informatics, Benha University, Benha 13518, Egypt

Copyright © 2013 H. E. Fathy et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract. In this paper, three numerical solutions for the Korteweg de-Vries Burgers’ (KdVB) equation are presented. Two of these methods are based on cardinal Chebyshev basis function with Galerkin method. Gauss-quadrature formula and El-gendi method are used to convert the problem into system of ordinary differential equations. In the third proposed method, the cardinal Legendre basis function with Galerkin method is used. In this case, the approximations are based on El-gendi method. The numerical results obtained by these ways have been compared with other solutions by Darvishi’s preconditioning to the same problem to show the efficiency of the proposed methods.

Keywords: KdV-Burgers’ equation; Galerkin method; El-gendi method; Legendre and Chebyshev cardinals, rounding error analysis.

1. Introduction

The Korteweg de-Vries Burgers’ equation (KdVB), derived by Su and Gardner [25], is a simple universal model equation which appears as the first approximation in the description of the dispersive – dissipative nonlinear waves and it used in the
study of wave propagation through a liquid-filled elastic tube [16] and for a
description of shallow water waves on a viscous fluid [17], see [26] for more details.
Also, The KdV–Burgers' equation is a one-dimension generalization of the model
description of the density and velocity fields that takes into account pressure forces
as well as the viscosity and the dispersion. It may be a more flexible tool for
physicists than the Burgers' equation [13]. Consider the KdV–Burgers’ equation is
given by:

\[ u_t + c u_x - \mu u_{xx} + \nu u_{xxx} = 0, \quad a < x < b, \quad 0 < t < T, \quad (1) \]

with the following initial and boundary conditions:

\[ u(x,0) = q(x), \quad a < x < b, \]
\[ u(a,t) = f_1(t), \quad u(b,t) = f_2(t), \quad t > 0. \quad (2) \]

Many numerical investigations were carried out by many authors to solve this
problem. For \( \nu = 0 \), we referred to [17], and [20] for the solution of equation (1).
The authors in [5] and [19] have been used \( \mu = 0 \) to find the numerical solutions of
equation (1). In Ref. [23], the author solved the KDV Burgers’ equation by
variational iteration method and compared the obtained results with the Adomian
decomposition method. The author in [15] presented the Homotopy perturbation
method for obtaining the numerical solutions of the KdV Burgers’ equation. Finally,
the authors in [3] solved the problem by means of spectral collocation method and
they used Darvishi’s preconditioning to reduce roundoff error.

Recently, the variety Galerkin algorithms have been applied in [6], [7], [8] and
Finally, the author in [18] proved that if the initial data \( u_0 \in L^2 \), then there exist a
unique solution \( u \in C([0,\infty);L^2) \cup C((0,\infty);H^1) \) of the initial boundary value
problem (1) and obtained the large time asymptotic of solution uniformly with
respect to \( x \in (0,1) \) as \( t \to \infty \).

In this work, the KdV Burgers’ equation is solved by variety Galerkin methods.
These methods have the advantages of both Galerkin and collocation methods,
where the Galerkin method is of considerable complexity, analytically as well as
computationally. In this case, each basis polynomial chosen must satisfy the
boundary conditions individually which causes the Galerkin formulation to become
complicated, particularly when the boundary conditions are time-dependent [1].
Furthermore, the presence of nonlinear term complicates the computation of the
stiffness matrix [14]. However, the Galerkin method is based on a variational
formulation which preserves essential properties of the continuous problem such as
coercively, continuity and symmetry of the bilinear form and it usually leads to
optimal error estimates [22]. On the other hand, the main advantage of the
collocation method is its simplicity and flexibility in implementation [1]. In
addition, the collocation method deals with nonlinear terms more easily than
Galerkin methods. In fact, problems with variable coefficients and general boundary
conditions are treated the same way and result in the same type of linear systems as
problems with constant coefficients and simple boundary conditions. However, the
collocation matrices, even for the simplest differential equation, are full and ill
conditioned [22].
The aim of this paper is to develop a stabilized nodal integration for the Galerkin method with Legendre, Chebyshev and Gauss quadrature formulae to achieve higher efficiency with desired accuracy and convergent properties. These methods are intermediate between collocation and Galerkin methods. They start from a weak form of the equations, but replace integrals by quadrature forms. For some cases, particularly for systems of conservation laws, they are much easier to implement than collocation methods, especially in multiple space dimensions. Generally, Galerkin method use Legendre approximations to avoid dealing with a variable weight function in the inner products [2].

The remainder of this paper is organized as follows: In section 2, we present Galerkin method with Chebyshev cardinal function as a basis function and we give the solution at the Chebyshev Gauss-Lobatto points. In section 3, the Chebyshev Galerkin method with El-gendi quadrature is introduced. In section 4, El-Gendi Legendre Galerkin method is described and we will use the Legendre cardinal function and the approximate solution will be presented at the Legendre Gauss-Lobatto points. In section 5, we will give the condition number for the participated matrices which compose the resulted system. In section 6, we discuss the rounding error in these approximations. Finally, in section 7 we give numerical experiments and present comparisons with Darvishi’s preconditioning method to illustrate the efficiency of our methods.

2. Gauss Chebyshev Galerkin (GCG) method

In this section we will explain the Gauss Chebyshev Galerkin method and illustrate how it used to solve equations (1-2) in case \( a = -1 \) and \( b = 1 \). In this method, the trial and the test spaces are identical, so that we define for \( m \geq 0 \) the space \( H^m(-1,1) \) to be a vector space of functions \( v \in L^2(-1,1) \) such that all distributional derivatives of \( v \) of order up to \( m \) can be represented by functions in \( L^2(-1,1) \). Since the functions of \( H^1(-1,1) \) are continuous up to the boundary by Sobolev imbedding theorem, it is meaningful to introduce the following solution subspace of \( H^1(-1,1) \):

\[
H^1_0(-1,1) = \{ u \in H^1(-1,1) : u(-1,t) = u(1,t) = 0 \}.
\]

(3)

The weak form of equations (1) and (2) are given by:

Find \( u \in H^1_0(-1,1) \) such that:

\[
\int_{-1}^{1} \frac{\partial u}{\partial t} v \, dx - \frac{\alpha}{2} \int_{-1}^{1} u^2 \frac{\partial^2 v}{\partial x^2} \, dx + \nu \int_{-1}^{1} \frac{\partial^2 u}{\partial x^2} \frac{\partial v}{\partial x} \, dx = 0 \quad \forall v \in H^1_0(-1,1), \, t > 0,
\]

(4)

The boundary conditions, of course, are imposed naturally in the weak formula (4) and the integrals above are meaningful.

The motivation for our study is the KdVB equation presented in [18]. We started by considering the approximation:
\[ u^N(x,t) = \sum_{j=0}^{N} U_j(t) \phi_j(x), \quad (5) \]

where \( u^N(x,t) \) denotes the approximate value of \( u(x,t), \ \phi_j(x) \) is the set of appropriate polynomials of length \( N \), and \( U_j(t) \) is a set of coefficients.

We discretize the space using a fixed mesh spacing, \( \Delta t \), denoting the approximation of the solution at the discrete grid-points by \( u^N(x_j,t) \), where the equations (1)-(2) are enforced at the collocation points defined by:

\[ x_j = \cos(j \cdot \frac{\pi}{N}), \quad j = 0, 1, ..., N. \quad (6) \]

The space of approximate solutions is:

\[ P_0^N = H^1_{0,\omega}(-1,1) \cap P^N, \quad (7) \]

where \( H^m_{\omega}(-1,1), \ m \geq 0 \) is the space of all functions \( h \in L^2_{\omega}(-1,1) \) such that the distributional derivatives of \( h \) of order up to \( m \) can be represented by functions in \( L^2_{\omega}(-1,1) \), and \( P^N \) is the space of orthogonal polynomials of degree of less than or equal \( N \). The weighted Galerkin method takes the form:

\[
\left( \frac{\partial u^N}{\partial t}, v \right) - \frac{\alpha}{2} \left( \frac{\partial v}{\partial x}, \frac{\partial u^N}{\partial x} \right) + \nu \left( \frac{\partial^3 u^N}{\partial x^3}, \frac{\partial v}{\partial x} \right) = 0 \quad \forall v \in P_0^N(-1,1), \ t > 0, \quad (8)\]

where the inner product \( (f, g)_{\omega} \) is defined as:

\[ (f, g)_{\omega} = \int_{-1}^{1} f(x) g(x) \omega(x) \, dx. \quad (9) \]

The corresponding Chebyshev Gauss quadrature formula is:

\[ \int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}} \, dx = \sum_{j=0}^{N} \omega_j f(x_j), \quad (10) \]

where \( \omega_j \)'s are given by:

\[ \omega_0 = \frac{\pi}{2N}, \ \omega_j = \frac{\pi}{N}, \ \omega_N = \frac{\pi}{2N}. \]

In the Gauss Chebyshev Galerkin method, the trial function space coincides with the test function space \( P_0^N \) which is a finite dimensional subspace of \( H^1_{\omega,0}(-1,1) \) and

\[ P_0^N = \text{span}\{\phi_0(x), \phi_1(x), ..., \phi_{N-1}\}, \quad (11) \]

where \( \phi_j(x) \) are given by:

\[ \phi_j(x) = \frac{2 \theta_j}{N} \sum_{k=0}^{N} \theta_k T_k(x_j) \eta_k(x), \quad j = 0, 1, ..., N, \]

for all \( \theta_k = 1 \), except \( \theta_0 = \theta_N = 1/2 \) and
\[
\varphi_j(x_k) = \begin{cases} 0 & \text{if } j \neq k, \\ 1 & \text{if } j = k. 
\end{cases}
\]

The grid points \(x_k\) being used are the so-called Chebyshev Gauss–Lobatto points, which are the extremal points of the Chebyshev polynomial \(T_N(x)\). Let \(v = \varphi_j(x)\), so the discrete weak form (4) is now takes the form:

\[
\sum_{k=0}^{N} \left( \frac{\partial u^N}{\partial t} \varphi_j \right)_{x=x_k} \omega_k - \frac{\alpha}{2} \sum_{k=0}^{N} \left( \frac{\partial (u^N)^2 \varphi_j}{\partial x} \right)_{x=x_k} \omega_k \\
+ \nu \sum_{k=0}^{N} \left( \frac{\partial u^N \varphi_j}{\partial x} \right)_{x=x_k} \omega_k - \mu \sum_{k=0}^{N} \left( \frac{\partial^2 u^N \varphi_j}{\partial x^2} - \varphi_j \right)_{x=x_k} \omega_k = 0, \quad j = 1, \ldots, N-1. \tag{12}
\]

The higher order derivatives of the cardinal functions \(\varphi_j(x)\) at the points \(x_l\) have the entries of the differentiation matrix: [9]

\[
\frac{d\varphi_j}{dx}_{x=x_l} = \frac{4\theta_j}{N} \sum_{k=0}^{N} \sum_{(n+k)\text{odd}}^{k-1} \frac{\theta_{k}c_n}{c_n} T_k(x_j)T_n(x_l), \quad j = 0,1,\ldots,N, \tag{13}
\]

the second derivatives are:

\[
\frac{d^2\varphi_j}{dx^2}_{x=x_l} = \frac{4\theta_j}{N} \sum_{k=2}^{N} \sum_{(n+k)\text{even}}^{k-2} \frac{\theta_{k}c_n(k^2-n^2)}{c_n} T_k(x_j)T_n(x_l), \quad j = 0,1,\ldots,N. \tag{14}
\]

Then equation (12) can be written as a system of ordinary differential equations (ODEs) in time as follows:

\[
M \frac{dU}{dt} + \frac{\alpha}{2} G (U \otimes U) + \nu B U - \mu Q U = 0, \quad t > 0,
\]

\[
M \frac{dU}{dt} = \left( \frac{\alpha}{2} G (U \otimes U) + \nu B U - \mu Q U \right), \quad t > 0,
\]

\[
U_0 = U_N = 0. \tag{15}
\]

where the symbol \(\otimes\) refer to the element wise multiplication of vectors. The entries of the stiffness matrix \(B\) are given by:

\[
(B)_{jl} = \sum_{k=0}^{N} \left( \frac{d\varphi_j}{dx} \frac{d\varphi_l}{dx} \right)_{x=x_k} \omega_k.
\]

The matrix \(G\) has the entries:

\[
(G)_{jl} = -\sum_{k=0}^{N} \left( \varphi_j \frac{d\varphi_l}{dx} \right)_{x=x_k} \omega_k,
\]

and the entries of the matrix \(Q\) is given by

\[
(Q)_{jl} = \sum_{k=0}^{N} \left( \frac{d^2\varphi_j}{dx^2} \frac{d\varphi_l}{dx} \right)_{x=x_k} \omega_k.
\]

The diagonal mass matrix \(M\) is given by

\[
M = \text{diag}(\omega_0, \omega_1, \ldots, \omega_N).
\]
The resulted system of ODEs has been solved by using Runge-Kutta solver. Further, because of stability considerations of explicit methods, there is a limit on the allowable time steps [3].

3. El-gendi Chebyshev Galerkin (ECG) method

In this section, El-gendi formula has been used as follows: Let

$$\int_{-1}^{1} g(x)dx \cong \sum_{k=0}^{N} b_{Nk} g(x_{k}),$$

(16)

where $b_{Nk}$ are given by:[10]

$$b_{Nk} = \frac{4}{N} \sum_{j=0}^{N/2-1} \frac{\theta_{j}}{4 j^{2} - 1} \cos \frac{2k\pi j}{N}, \quad k = 1, 2, \ldots, N - 1,$$

$$b_{N0} = b_{NN} = \frac{1}{N^{2} - 1}. \quad (17)$$

For the numerical solution, we consider the approximation (5) and Galerkin scheme (4) takes the form:

$$\sum_{k=0}^{N} \left( \frac{\partial u}{\partial t} \phi_{j} \right)_{x_{k}} b_{Nk} - \frac{\alpha}{2} \sum_{k=0}^{N} \left( u^{2} N \frac{\partial \phi_{j}}{\partial x} \right)_{x_{k}} b_{Nk}$$

$$+ \nu \sum_{k=0}^{N} \left( \frac{\partial u}{\partial x} \frac{\partial \phi_{j}}{\partial x} \right)_{x_{k}} b_{Nk} + \mu \sum_{k=0}^{N} \left( \frac{\partial^{2} u}{\partial x^{2}} \frac{\partial \phi_{j}}{\partial x} \right)_{x_{k}} b_{Nk} = 0, \quad j = 1, \ldots, N - 1. \quad (18)$$

Hence, the equation (18) can be re-written as the system of ordinary differential equations (ODEs) in time as follows:

$$\vec{M} \frac{d\vec{U}}{dt} + \frac{\alpha}{2} \vec{G} (\vec{U} \otimes \vec{U}) + \nu \vec{B} \vec{U} + \mu \vec{Q} \vec{U} = 0, \quad t > 0,$$

$$\vec{M} \frac{d\vec{U}}{dt} = -\left( \frac{\alpha}{2} \vec{G} (\vec{U} \otimes \vec{U}) + \nu \vec{B} \vec{U} + \mu \vec{Q} \vec{U} \right), \quad t > 0,$$

$$\vec{U}_{0} = \vec{U}_{N} = 0. \quad (19)$$

In this case the entries of stiffness matrix $\vec{B}$ are given as following:

$$\left( \vec{B} \right)_{ij} = \sum_{k=0}^{N} \left( \frac{d \phi_{i}}{dx} \frac{d \phi_{j}}{dx} \right)_{x_{k}} b_{Nk}.$$  

The matrix $\vec{G}$ has the entries:

$$\left( \vec{G} \right)_{ij} = -\sum_{k=0}^{N} \left( \phi_{i} \frac{d \phi_{j}}{dx} \right)_{x_{k}} b_{Nk},$$

and the entries of the matrix $\vec{Q}$ is given by

$$\left( \vec{Q} \right)_{ij} = \sum_{k=0}^{N} \left( \frac{d^{2} \phi_{i}}{dx^{2}} \frac{d \phi_{j}}{dx} \right)_{x_{k}} b_{Nk}.$$
The diagonal mass matrix $\tilde{M}$ is given by:

$$
\tilde{M} = \text{diag}(b_{N0}, b_{N1}, \ldots, b_{NN}).
$$

The forth order Runge-Kutta method has been used also to solve the system (19).

### 4. El-gendi Legendre Galerkin (ELG) method

Analogous to the previous section we consider the Legendre cardinal function based on Legendre Gauss-Lobatto (LGL) nodes. El-gendi approximation will be used with a linear combination of the Legendre cardinal function as following: [11]

$$
\psi_j(y) = \frac{1}{N(N-1)} \sum_{k=0}^{N} (2k+1) \frac{L_k(y_j)L_k(y)}{[L_{N-1}(y_j)]^2}, \quad j = 0,1,\ldots,N,
$$

where $y_j$ are the Legendre Gauss-Lobatto points and $\psi_j(y)$ satisfies the condition:

$$
\psi_j(y_k) = \begin{cases} 
1, & \text{if } j = k, \\
0, & \text{if } j \neq k.
\end{cases}
$$

The discrete weak formulation (4) is then takes the form:

$$
\sum_{k=0}^{N} \left( \frac{\partial u^N_j}{\partial t} \psi_j \right)_{y=y_k} b_{Nk} - \frac{\alpha}{2} \sum_{k=0}^{N} \left( (u^N_j)^2 \frac{\partial \psi_j}{\partial y} \right)_{y=y_k} b_{Nk}
+ \nu \sum_{k=0}^{N} \left( \frac{\partial u^N_j}{\partial y} \frac{\partial \psi_j}{\partial y} \right)_{y=y_k} b_{Nk} = 0, \quad j = 1,\ldots,N-1, \quad (20)
$$

where $\left( \frac{d\psi_j}{dy} \right)_{y=y_k}$ is the first order differentiation matrix depends on Legendre polynomial at the LGL nodes and have the entries given by: [12]

$$
\left( \frac{d\psi_j}{dy} \right)_{y=y_k} = \frac{1}{N(N-1)} \sum_{j=1}^{N} \sum_{k=0}^{N} \sum_{s=0}^{[N-1]/2} \sum_{l=0}^{[N-1]/2} z^{(j)}_s \bar{g}^{(j)}_{kl} \frac{1}{(c^N_s)^2} y^{j+s+2N+2} y^{j-2k-1}, \quad (21)
$$

where

$$
z^{(j)}_s = \frac{(-1)^j(2j+1)(2j-2s)!}{2^j(j-s)!(j-2s)!s!}, \quad \bar{g}^{(j)}_{kl} = \frac{(-1)^j(2j-2k)!(j-2k)!}{2^j(j-k)!(j-2k)!k!},
$$

and

$$
c^N_s = \sum_{s=0}^{[N-1]/2} \frac{(-1)^s(2N-2-2s)!}{2^{N-1}(N-1-s)!(N-1-2s)!s!}.
$$

Also, $\left( \frac{d^2\psi_j}{dy^2} \right)_{y=y_k}$ has the following formula:

$$
\left( \frac{d^2\psi_j}{dy^2} \right)_{y=y_k} = \frac{1}{N(N-1)} \sum_{j=2}^{N} \sum_{k=0}^{N} \sum_{s=0}^{[N-1]/2} \sum_{l=0}^{[N-1]/2} z^{(j)}_s \bar{g}^{(j)}_{kl} \frac{1}{(c^N_s)^2} y^{j+s+2N+2} y^{j-2k-2}, \quad (22)
$$
where

\[ g_{k,2}^{(j)} = \frac{(-1)^j(2j-2k)!(j-2k)!(j-2k-1)!}{2^j(j-k)!(j-2k)!k!}. \]

Then equation (20) can be written as the following system:

\[
\begin{align*}
\tilde{M}_L \frac{dU}{dt} + \frac{\alpha}{2} \tilde{G}_L (U \otimes U) + \nu \tilde{B}_L U - \mu \tilde{Q}_L U &= 0, \quad t > 0, \\
\tilde{M}_L \frac{dU}{dt} &= \left( \frac{\alpha}{2} G_L (U \otimes U) + \nu \tilde{B}_L U - \mu \tilde{Q}_L U \right), \quad t > 0, \\
U_0 &= U_N = 0.
\end{align*}
\]

In the Legendre case, the stiffness matrix \( \tilde{B}_L \) is also a symmetric positive-semidefinite matrix whose entries are:

\[
(\tilde{B}_L)_{\beta} = \sum_{k=0}^{N} \left( \frac{dy_{\beta}}{dy} \right)_{y=y_k} b_{Nk}.
\]

The matrix \( \tilde{G}_L \) has entries given by

\[
(\tilde{G}_L)_{\beta} = -\sum_{k=0}^{N} \left( \frac{dy_{\beta}}{dy} \right)_{y=y_k} b_{Nk},
\]

and the entries of the matrix \( \tilde{Q}_L \) is given by

\[
(\tilde{Q}_L)_{\beta} = \sum_{k=0}^{N} \left( \frac{d^2y_{\beta}}{dy^2} \right)_{y=y_k} b_{Nk}.
\]

As above, we used the forth order Runge-Kutta methods to solve the resulted system (23).

The structure (23) has an invertible stiffness matrix \( \det(\tilde{B}) \neq 0 \), therefore, no eigenvalues of an invertible matrix are equal to zero. In fact, throughout this work, the matrices \( B \) and \( \tilde{B} \) (in the three cases GCG, ECG and ELG) are symmetric semipositive definite matrices and all eigenvalues are positive numbers. At the end of the next section, we present a study of stability of these matrices which have affecting on the structures (15), (19) and (23), respectively.

5. Error analysis

In this section, the error bounds of the differentiation matrices given in (13), (14), (21) and (22) are presented, respectively. Also, for the square matrices \( B \) and \( \tilde{B} \), we present here the measurements of the sensitivity of equations (5), (19) and (23) by using the condition number estimations.

5.1 Errors in differentiation matrices

The authors in [9] investigate the effect of roundoff errors on the entries of (13) and (14). To this end, consider the difference between the exact value and the computed value is given by: \( \rho_k^* - \rho_k = \delta_k \), where \( \delta_k \) denotes a small error, with \( |\delta_k| \)
approximately equal to machine precision $\varepsilon$ and $\delta$ is the exact value and $\rho_k^*$ is the computed value. The order of machine precision is given by

$$|\rho_k^* - \rho_k| = (\delta_k + \delta_n) - O\left(\frac{1}{N^2}\delta_k\right) - O\left(\frac{1}{N^2}\delta_n\right).$$

In [9], the authors show that the elements of first order differentiation matrix $(d_{\text{cheb}})_{ij}^{(1)}$ have the following error bound:

$$\left(d_{\text{cheb}})_{ij}^{(1)*} - (d_{\text{cheb}})_{ij}^{(1)} \leq 4\theta_j \left(\frac{1}{N^2}\delta\right)\left(\frac{1}{3}N^2 + \frac{1}{6}\right),$$

while the maximum major element is $(d_{\text{cheb}})_{01}^{(1)}$ and given by

$$\left(d_{\text{cheb}})_{01}^{(1)*} - (d_{\text{cheb}})_{01}^{(1)} \leq \left(\frac{1}{3} - N + \frac{2}{3}N^2\right)\delta. \quad (24)$$

Moreover, the roundoff error of the second order differentiation matrix is given by

$$\left(d_{\text{cheb}})_{ij}^{(2)*} - (d_{\text{cheb}})_{ij}^{(2)} \leq \frac{4\theta_j}{3} \left(\frac{1}{N^2}\delta\right)\left(\frac{1}{5}N^4 - \frac{1}{5}\right),$$

with maximum major element:

$$\left(d_{\text{cheb}})_{01}^{(2)*} - (d_{\text{cheb}})_{01}^{(2)} \leq \left(\frac{N^4}{5} - \frac{N^3}{2} + \frac{N}{2} - \frac{1}{5}\right)\frac{2\delta}{3}.$$ 

On the other hand, the authors in [12] presented the error bound of the first and second order differentiation in equations (21) and (22), respectively, as follows:

$$\left(d_{\text{leg}})_{ij}^{(1)*} - (d_{\text{leg}})_{ij}^{(1)} \leq \left(\frac{1}{N^2}\delta\right)\left(\frac{4N^2 + 9N + 5}{3(N-1)}\right),$$

with maximum major element:

$$\left(d_{\text{leg}})_{01}^{(1)*} - (d_{\text{leg}})_{01}^{(1)} \leq (-1)^N \frac{\delta}{4N(N-1)}\left(\frac{4N^2 + 6N + 1}{4N(N-1)}\right) - \frac{\delta}{4N(N-1)}. \quad (25)$$

The error bound of the second order differentiation matrix is:

$$\left(d_{\text{leg}})_{ij}^{(2)*} - (d_{\text{leg}})_{ij}^{(2)} \leq \left(\frac{1}{N^2}\delta\right)\left(\frac{3N^3 + 4N^2 - 3N - 4}{3(N-1)}\right),$$

with maximum major element:

$$\left(d_{\text{leg}})_{01}^{(2)*} - (d_{\text{leg}})_{01}^{(2)} \leq \delta\left(\frac{3N^3 + 4N^2 - 3N - 4}{6(N-1)}\right).$$

These facts lead us to illustrate that the Legendre polynomial provides an error bound with less computational values than the Chebyshev polynomial. For example, for $N=8$, we found that $(d_{\text{leg}})_{01}^{(1)}$ gives 96.1% minimal error bound than $(d_{\text{cheb}})_{01}^{(1)}$. Whereas, $(d_{\text{leg}})_{01}^{(2)}$ has been reduced with 88.9% of error bound than $(d_{\text{cheb}})_{01}^{(2)}$. On the other hand, the formula (17) has the elements $b_{nk}$ satisfying the property:
This symmetric property leads us to reduce the number of operations to 50% off and hence reducing the rounding error. So that for \( N \) even we get:

\[
\sum_{k=0}^{N} b_{Nk} f_k = \sum_{k=0}^{[N/2]} \lambda_k b_{Nk} f_k + f_{N-k},
\]

where

\[
\lambda_k = \begin{cases} 
1 & \text{if } k = 0, 1, \ldots, (N/2) - 1, \\
1/2 & \text{if } k = N/2.
\end{cases}
\]

and \( f_k = f(x_k) \). The formula (17) can be written in the form:

\[
b_{Nk} = \frac{4}{N} \sum_{j=0}^{N/2} \theta_j a_j T_{2j}(x_k), \quad k = 1, 2, \ldots, N-1,
\]

\[
a_j = \frac{1}{4j^2 - 1}. \quad (27)
\]

In case of \( N \) odd, we have

\[
\sum_{k=0}^{N} b_{Nk} f_k = \sum_{k=0}^{[N/2]} b_{Nk} f_k + f_{N-k},
\]

where \([N/2]\) is the integer number of the division \( N/2 \). Equation (27) is an alternating series which converges as \( N \to \infty \) since \( |T_{2j}(x_k)| \leq 1 \) for all \( j \) and \( k \).

Moreover, \( a_j \)'s are all positive for \( j \geq 1 \), \( a_j \geq a_{j+1} \) and \( \lim_{j \to \infty} a_j = 0 \).

### 5.2 Bounds for condition number of stiffness matrix

Condition number is defined as the product of the norm of \( B \) and the norm of \( B^{-1} \).

\[
\kappa = \|B^{-1}\| \|B\|,
\]

where, the norm of a matrix is a scalar that gives some measure of the magnitude of the elements of the matrix. The condition number of the stiffness matrix is an upper bound to the amplification of errors in structural properties (4). However, one can notice that, even though typical stiffness matrices have condition numbers larger than one million; we do not expect that errors in the structure (4) would be amplified so much. Table (1) shows the condition number for the participated stiffness matrices for each above three cases.

Table (1): Observed condition numbers of the stiffness matrices

<table>
<thead>
<tr>
<th>N</th>
<th>( \kappa ) (GCG)</th>
<th>( \kappa ) (ECG)</th>
<th>( \kappa ) (ELG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1.20E+02</td>
<td>5.64E+01</td>
<td>2.68E+01</td>
</tr>
<tr>
<td>16</td>
<td>1.80E+03</td>
<td>2.14E+02</td>
<td>1.60E+02</td>
</tr>
<tr>
<td>32</td>
<td>2.83E+04</td>
<td>1.65E+03</td>
<td>1.11E+03</td>
</tr>
<tr>
<td>64</td>
<td>4.51E+05</td>
<td>1.30E+04</td>
<td>8.37E+03</td>
</tr>
<tr>
<td>128</td>
<td>7.20E+06</td>
<td>1.04E+05</td>
<td>6.52E+04</td>
</tr>
</tbody>
</table>
The previous Table shows that our expectation is fulfilled. It also presents an example of a case associated with error bound analysis where the worst-case scenario predicted by the condition number is much closer to the actual error amplification. In the above three cases, if the stiffness matrices corresponding to the structure (4) do not change, the numerical solutions will not change and the dispersion space remains unchanged.

6. Numerical Experiments

In this section we will give two examples and we will use MATLAB 7.0 software to obtain the numerical results.

Example (1): [3]
Consider the KDV–Burgers’ equation:

\[ u_t + c u_x - \nu u_{xx} + \mu u_{xxx} = 0, \quad a < x < b, \quad 0 < t < T, \]

where \( \alpha, \nu, \mu \) are arbitrary constants and \( a \) and \( b \) are real numbers.

The initial condition is:

\[ u(x,0) = \frac{6\nu^2}{25\mu} \left[ 1 + \tanh \left( \frac{\nu}{10\mu} x \right) - \frac{1}{2} \sec h^2 \left( \frac{\nu}{10\mu} x \right) \right]. \]

If \( a = 0 \) and \( b = 100 \), then the boundary conditions are:

\[ u(0,t) = \frac{6\nu^2}{25\mu} \left[ 1 + \tanh \left( \frac{3\nu^2}{125\mu^2} t \right) - \frac{1}{2} \sec h^2 \left( \frac{3\nu^2}{125\mu^2} t \right) \right], \]

\[ u(100,t) = \frac{6\nu^2}{25\mu} \left[ 1 + \tanh \left( \frac{\nu}{10\mu} \left( 100 + \frac{3\nu^2}{25\mu} t \right) \right) - \frac{1}{2} \sec h^2 \left( \frac{\nu}{10\mu} \left( 100 + \frac{3\nu^2}{25\mu} t \right) \right) \right]. \]

The exact solution is:

\[ u(x,t) = \frac{6\nu^2}{25\mu} \left[ 1 + \tanh \left( \frac{\nu}{10\mu} \left( x + \frac{3\nu^2}{25\mu} t \right) \right) - \frac{1}{2} \sec h^2 \left( \frac{\nu}{10\mu} \left( x + \frac{3\nu^2}{25\mu} t \right) \right) \right]. \]

The following error notations are defined:

\[ \text{abser} = \left| u_{ex}(x_i) - u_{\text{app}}(x_i) \right|, \quad i = 1, \ldots, N-1, \]

and,

\[ \text{maxer} = \max_i \left| u_{ex}(x_i) - u_{\text{app}}(x_i) \right|, \quad i = 1, \ldots, N-1. \]

where \( u_{ex}(x_i) \) and \( u_{\text{app}}(x_i) \) are the exact and approximate solutions, respectively.

In Table (2), the problem solved for \( \Delta t = 0.05, \nu = 0.001, \mu = 0.001, \alpha = 1 \) and \( N = 16 \). Because of Chebyshev-Gauss-Lobatto points are in interval \([-1,1]\), therefore, the interval \([0,100]\) is mapped to \([-1, 1]\) by a linear mapping defined by:

\[ X_i = \left( \frac{b-a}{2} \right) \eta_i + \left( \frac{b+a}{2} \right), \quad i = 0, \ldots, N, \]

where \( \eta_i \) are the Gauss-Lobatto nodes.
Table (2): $N = 16, \Delta t = 0.05, \nu = 0.001, \mu = 0.001$ and $\alpha = 1$.

<table>
<thead>
<tr>
<th>$T$</th>
<th>X</th>
<th>abser. Darvishi</th>
<th>abser. GCG</th>
<th>abser. ECG</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>8.42</td>
<td>2.94 E-06</td>
<td>1.02 E-06</td>
<td>1.02 E-06</td>
</tr>
<tr>
<td></td>
<td>40.24</td>
<td>1.11 E-08</td>
<td>2.99 E-09</td>
<td>2.92 E-09</td>
</tr>
<tr>
<td></td>
<td>77.77</td>
<td>5.04 E-10</td>
<td>3.02 E-07</td>
<td>4.04 E-13</td>
</tr>
<tr>
<td></td>
<td>99.03</td>
<td>2.03 E-09</td>
<td>1.19 E-08</td>
<td>1.84 E-12</td>
</tr>
<tr>
<td>400</td>
<td>8.42</td>
<td>5.85 E-06</td>
<td>2.04 E-06</td>
<td>2.04 E-06</td>
</tr>
<tr>
<td></td>
<td>40.24</td>
<td>2.05 E-08</td>
<td>5.93 E-09</td>
<td>5.82 E-09</td>
</tr>
<tr>
<td></td>
<td>77.77</td>
<td>4.59 E-11</td>
<td>6.82 E-10</td>
<td>1.79 E-12</td>
</tr>
<tr>
<td></td>
<td>99.03</td>
<td>1.08 E-09</td>
<td>1.79 E-08</td>
<td>2.99 E-12</td>
</tr>
<tr>
<td>600</td>
<td>8.42</td>
<td>8.74 E-06</td>
<td>3.05 E-06</td>
<td>3.05 E-06</td>
</tr>
<tr>
<td></td>
<td>40.24</td>
<td>2.86 E-08</td>
<td>8.79 E-09</td>
<td>8.70 E-09</td>
</tr>
<tr>
<td></td>
<td>77.77</td>
<td>1.46 E-09</td>
<td>6.82 E-10</td>
<td>3.90 E-12</td>
</tr>
<tr>
<td></td>
<td>99.03</td>
<td>6.13 E-09</td>
<td>1.79 E-09</td>
<td>3.29 E-12</td>
</tr>
<tr>
<td>800</td>
<td>8.42</td>
<td>1.16 E-05</td>
<td>4.06 E-06</td>
<td>4.06 E-06</td>
</tr>
<tr>
<td></td>
<td>40.24</td>
<td>3.56 E-08</td>
<td>1.16 E-08</td>
<td>1.15 E-08</td>
</tr>
<tr>
<td></td>
<td>77.77</td>
<td>3.56 E-09</td>
<td>6.81 E-10</td>
<td>5.99 E-12</td>
</tr>
<tr>
<td></td>
<td>99.03</td>
<td>1.12 E-08</td>
<td>1.78 E-08</td>
<td>3.59 E-12</td>
</tr>
</tbody>
</table>

The error magnification given by the proposed method (GCG) is a much sharper estimate of error amplification than (ELG); this is due to the condition number $\kappa$ (ELG). It seems from Table (2) that Gauss and El-gendi Chebyshev Galerkin methods are more accurate than the Darvishi’s preconditioning. Moreover, ECG is more accurate than the GCG especially at the points $X = 77.77$ and $X = 99.03$.

Table (3): $N = 64$ and $\Delta t = 0.05, \nu = 0.001, \mu = 0.001, \alpha = 1$.

<table>
<thead>
<tr>
<th>$T$</th>
<th>maxer. Darvishi</th>
<th>maxer. GCG</th>
<th>maxer. ECG</th>
<th>maxer. ELG</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>1.07 E-04</td>
<td>1.36 E-06</td>
<td>1.36 E-06</td>
<td>1.36 E-06</td>
</tr>
<tr>
<td>400</td>
<td>1.57 E-04</td>
<td>2.72 E-06</td>
<td>2.72 E-06</td>
<td>2.72 E-06</td>
</tr>
<tr>
<td>600</td>
<td>2.88 E-04</td>
<td>4.08 E-06</td>
<td>4.08 E-06</td>
<td>4.09 E-06</td>
</tr>
<tr>
<td>800</td>
<td>4.00 E-04</td>
<td>5.45 E-06</td>
<td>5.44 E-06</td>
<td>5.45 E-06</td>
</tr>
</tbody>
</table>

Table (3) shows the performance of Darvishi’s, GCG, ECG and ELG applied to example (1). For fixed time step, $\Delta t = 0.05$, we found that when the number of nodes increases with the same parameter values, the Galerkin methods have the maximum absolute error less than Darvishi’s preconditioning. Moreover, if $\Delta t < 0.05$, we found that the Darvishi’s preconditioning can arrive to the same accuracy as Galerkin methods. Hence, the Galerkin methods are converged faster and more accurate than Darvishi’s preconditioning. Finally, the results show what we expected from the dispersion. That is, the performance of Galerkin methods is generally identical compared to the Darvishi’s benchmark case.
Chebyshev and Legendre via Galerkin method

Fig.1 Darvishi’s solution in N=64, T=200. Fig.2 GCG solution in N=64, T=200.

Fig.3 ECG solution in N=64, T=200.

On the other hand, the solutions of $u^N(x,t)$ using CGL and LGL discretization are compared in Figures (1)-(3). Severe oscillation in $u^N(x,t)$ is observed in the Darvishi’s method, and oscillation in $(dU/dt)$ near the boundary is also apparent in Fig. (1). The stabilized conforming Gauss and El-gendi Chebyshev Galerkin methods display a superior performance over the direct Darvishi’s preconditioned method.

Example (2): Consider the KDV–Burgers’ equation: [24]

$$u_t + c_1 u u_x - c_2 u_{xx} + \mu u_{xxx} = 0, \quad a < x < b, \quad 0 < t < T,$$

If $a = -10$ and $b = 10$, the exact solution is

$$u(x,t) = A[9 - 6 \tanh(B(x - Ct)) - 3 \tanh^2(B(x - Ct))],$$

where, $A = \frac{\nu}{25\alpha} \mu$, $B = \frac{\nu}{10\mu}$, and $C = \frac{6\nu^2}{25\mu}$.

Table (4): Observation Maxer. for $N=10$, $\nu=0.01$ and $\Delta t = 0.01$.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>T</th>
<th>Darvishi</th>
<th>GCG</th>
<th>ECG</th>
<th>ELG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 1$, $\mu = 0.01$</td>
<td>0.3</td>
<td>5.53 e-06</td>
<td>2.38 e-07</td>
<td>2.11 e-07</td>
<td>2.11 e-07</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.65 e-05</td>
<td>1.58 e-06</td>
<td>6.62 e-07</td>
<td>6.64 e-07</td>
</tr>
<tr>
<td>$\alpha = 0.1$, $\mu = 0.01$</td>
<td>0.3</td>
<td>5.51 e-05</td>
<td>2.38 e-06</td>
<td>2.11 e-06</td>
<td>2.11 e-06</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>1.69 e-04</td>
<td>1.58 e-05</td>
<td>6.62 e-06</td>
<td>6.64 e-06</td>
</tr>
<tr>
<td>$\alpha = 0.01$, $\mu = 0.1$</td>
<td>0.3</td>
<td>5.08 e-05</td>
<td>4.54 e-07</td>
<td>4.14 e-07</td>
<td>1.83 e-07</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>2.01 e-04</td>
<td>3.21 e-06</td>
<td>3.00 e-06</td>
<td>1.36 e-06</td>
</tr>
<tr>
<td>$\alpha = 0.1$, $\mu = 1$</td>
<td>0.3</td>
<td>4.87 e-07</td>
<td>3.86 e-09</td>
<td>4.06 e-09</td>
<td>1.94 e-09</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>2.62 e-06</td>
<td>2.94 e-08</td>
<td>3.12 e-08</td>
<td>1.37 e-08</td>
</tr>
</tbody>
</table>
Table (4) illustrates that approximate solution of the KdV–Burgers’ equation. The Table exhibits different behaviors depending on the parameter values. It is clear when $T = 0.9$, $\alpha = 0.1$, $\mu = 0.01$ the GCG method has maximum absolute error greater than El-gendi Galerkin methods. Also, when $\alpha = 1.0, \nu = 0.01$ we see that ECG and ELG have the same maximum absolute error. On the other hand, when $\alpha = 0.01, 1.0$, and $\mu = 0.1, 1.0$, we see that the maximum absolute error of ELG method is smaller than ECG and GCG methods.

![Fig. 4 ECG solution in N=10, T=0.9, \(\alpha = 0.1, \nu = 0.01, \mu = 1\).](image1)

![Fig. 5 ELG solution in N=10, T=0.9, \(\alpha = 0.1, \nu = 0.01, \mu = 1\).](image2)

![Fig. 6 Darvishi’s solution in N=10, T=0.9, \(\alpha = 0.1, \nu = 0.01, \mu = 1\).](image3)

Note, in particular, the difference between the two oscillatory solutions presented in Fig. 6. We would like to stress that these oscillations appear in the preconditioner solutions, but do not appear with the Galerkin methods used in Fig 4 and Fig. 5. The results also show the scrappy in the convergence behavior which indicates the existence of small eigenvalues related to slow converging components. Analysis of the spectrum of the stiffness matrix shows that the smallest eigenvalues correspond to the preconditioned solution. At this end, we refer to the discretization points in the initial data which has a strong stabilizing effect on the results. It seems to control the oscillations such that they do not change sign and they remain bounded in time.
Fig. 7 The absolute error of three methods in case $\Delta t = 0.01$, $T=0.3, \alpha = 0.1, \mu = 1, \nu = 0.01$ and $N=[2,4,8,10,12,16,32]$.

7. Conclusion

In this paper, three efficient methods depend on Galerkin method to solve the KdV–Burgers’ equation are presented. As can be seen from above Tables, the Gauss-Chebyshev, El-gendi Chebyshev and Legendre Galerkin methods have smaller errors than spectral collocation method with preconditioning. The Galerkin approximations are just as easy to implement as collocation methods. However, we have illustrated how the Chebyshev and Legendre methods approximate with spectral accuracy for KdV Burgers’ equation and how less order of expansion slows down convergence. The numerical experiments show that the performance of Galerkin methods with Legendre approximation is faster than Chebyshev approximation.

References


Received: February 16, 2013