Tema

Tendências em Matemática Aplicada e Computacional, **18**, N. 2 (2017), 287-304
© 2017 Sociedade Brasileira de Matemática Aplicada e Computacional
www.scielo.br/tema
doi: 10.5540/tema.2017.018.02.0287

Numerical Simulations with the Galerkin Least Squares Finite Element Method for the Burgers' Equation on the Real Line

P.H.A. KONZEN[1]*, F.S. AZEVEDO[2], E. SAUTER[2] and P.R.A. ZINGANO[2]

Received on November 22, 2016 / Accepted on April 27, 2017

ABSTRACT. In this work we present an efficient Galerkin least squares finite element scheme to simulate the Burgers' equation on the whole real line and subjected to initial conditions with compact support. The numerical simulations are performed by considering a sequence of auxiliary spatially dimensionless Dirichlet's problems parameterized by its numerical support $\tilde{K}$. Gaining advantage from the well-known convective-diffusive effects of the Burgers' equation, computations start by choosing $\tilde{K}$ so it contains the support of the initial condition and, as solution diffuses out, $\tilde{K}$ is increased appropriately. By direct comparisons between numerical and analytic solutions and its asymptotic behavior, we conclude that the proposed scheme is accurate even for large times, and it can be applied to numerically investigate properties of this and similar equations on unbounded domains.

Keywords: Burgers' equation on the real line, Galerkin least squares finite element method, asymptotic properties.

1 INTRODUCTION

Consider the viscous Burger's equation defined on the real line:

$$\frac{\partial u}{\partial t} + bu\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}, \quad (x \in \mathbb{R}, t > 0), \tag{1.1}$$

subjected to the initial condition:

$$u(x, 0) = g(x), \quad (x \in \mathbb{R}), \tag{1.2}$$

where $b \neq 0$ is a given parameter, $\nu > 0$ is a given viscosity coefficient and g is a given function with compact support on $\mathbb{R}$.

*Corresponding author: Pedro Henrique de Almeida Konsen – E-mail: pedro.konzen@ufrgs.br

[1] Departamento de Matemática Pura e Aplicada, Instituto de Matemática e Estatística – Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves, 9500 – Prédio 43-111 – Agronomia, 91509-900 Porto Alegre, RS, Brasil.

[2] Programa de Pós-Graduação em Matemática Aplicada, Instituto de Matemática e Estatística – Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves, 9500 – Prédio 43-111 – Agronomia, 91509-900 Porto Alegre, RS, Brasil. E-mails: fazedo@gmail.com; esequia@gmail.com; pzingano@mat.ufrgs.br

Burgers' equation is known to have appeared firstly in 1915 in the work of Harry Bateman [10], but it receives its name after the Dutch physicist J.M. Burgers, who applied this equation in the understanding of turbulent fluids [12]. This homogeneous quasilinear parabolic partial differential equation appears in the modeling of several phenomena such as shock flows, wave propagation in combustion chambers, vehicular traffic movement, acoustic transmission, etc. (see, for instance, [21] and the references therein). Another import characteristic of this equation is its several well known analytic solutions in bounded and unbounded domains. Therefore, this equation is already a classical test case in mathematical analysis and numerical simulations of convective-diffusive partial differential equations.

From the analytic point of view the literature is rich in discussing solutions and properties for the Burgers' equation on bounded and unbounded regions and subjected to a variety of initial and boundary conditions (see, for instance, [1, 9, 12, 20, 24, 28, 47, 52]). Now, from the numerical simulation point of view the majority of the studies found in the literature are concerned about the Burgers' equation defined in a bounded region and subjected to Dirichlet's boundary conditions. Several numerical schemes have been applied to simulate this problem, for instance: finite element methods [2, 3, 5, 14, 15, 16, 19, 22, 31, 35, 39, 40, 43, 45, 49, 55], finite difference methods [14, 22, 25, 38, 42], variational schemes [4, 13, 46], spectral methods [8, 37], Hardy's multiquadric method [29], matched asymptotic expansion methods [44], multisymplectic box methods [51], homotopy analysis methods [32], the quintic B-spline collocation procedure [48], the gradient reproducing kernel particle method [27], quasi-interpolation techniques [54], uniform Haar wavelets [33]. Numerical solutions by spatial discretization techniques of the Burgers' equation in unbounded domains have been obtained by applying the artificial boundary method [26, 53, 50]. A comprehensive review of techniques for the solution of the Burgers' equation is found in [57].

In this work we present an efficient numerical scheme based on the Galerkin least squares finite element method to simulate Burgers' equation on the real line and subjected to initial conditions with compact support. The proposed scheme explore the convective-diffusive nature of the differential equation. If for small times the convective effects are predominant demanding very fine and localized meshes, for large times diffusion takes over and the solution tends to relax demanding less refined but large meshes. We deal with it by computing the finite element discretization of a sequence of dimensionless spatially forms of the Burgers' equation on a fixed mesh and parameterized by its domain. This idea has been proved very computational efficient producing accurate results.

In the works of Han [26] and Sun [50] the problem (1.1)-(1.2) has been faced applying artificial boundary conditions to approximate it as a boundary-valued problem in a bounded domain, which they have solved by means of finite difference approximations. They have also tested their scheme comparing their computed solutions against analytic solutions for diffusion coefficients $\nu = 1, 0.1$ and $t < 25$. Our proposed GLS-FEM scheme uses much simpler boundary conditions (actually also allows other boundary conditions) and allowed us to accurately compute solutions of problem (1.1)-(1.2) with smaller $\nu = 10^{-4}$ and much larger times. This is here supported by direct comparisons between numerical and analytic solutions and their asymptotic behavior.

In the next section we briefly discuss the analytic solution of problem (1.1)-(1.2) and its asymptotic properties. In Section 3 we present the proposed time and space discretization of the spatially dimensionless form of the Burgers' equation. In Section 4 we discuss the details of the implementation scheme. Then in Section 5 we present numerical experiments, which endorse the efficiency and accuracy of the scheme as to its potential to be applied to investigate solution properties in unbounded domains. Finally, in Section 6 we close by summarizing the principal aspects of this work.

2 ANALYTIC SOLUTION

Here we recall the well known closed-form expression for $u(x,t)$ obtained by J. Cole and E. Hopf [17, 30]. Introducing $\beta(x,t)$ and $\beta_0(x)$ by the Hopf-Cole transformation:

$$\beta(x,t) := \exp\left\{-\frac{b}{2\nu}\int_0^x u(y,t)\,dy\right\}, \quad \beta_0(x) := \exp\left\{-\frac{b}{2\nu}\int_0^x g(y)\,dy\right\}, \tag{2.1}$$

one obtains that β solves the following initial value problem for the heat equation:

$$\frac{\partial\beta}{\partial t} = \nu\frac{\partial^2\beta}{\partial x^2}, \quad (x\in\mathbb{R}, t>0) \tag{2.2}$$

$$\beta(x,0) = \beta_0(x), \quad (x\in\mathbb{R}), \tag{2.3}$$

whose unique bounded solution is given by Poisson's formula:

$$\beta(x,t) = \frac{1}{\sqrt{4\pi\nu t}}\int_{-\infty}^{\infty} e^{-\frac{|x-y|^2}{4\nu t}}\beta_0(y)\,dy, \quad (x\in\mathbb{R}, t>0). \tag{2.4}$$

Since $u = -\frac{2\nu}{b}\frac{\beta_x}{\beta}$, it follows that:

$$u(x,t) = \frac{1}{b}\frac{\int_{-\infty}^{\infty}\frac{x-y}{t}e^{-\frac{|x-y|^2}{4\nu t}}\beta_0(y)\,dy}{\int_{-\infty}^{\infty}e^{-\frac{|x-y|^2}{4\nu t}}\beta_0(y)\,dy}, \quad (x\in\mathbb{R}, t>0). \tag{2.5}$$

This also shows that problem (1.1)-(1.2) has a unique solution $u(\cdot,t)\in C^0\left([0,\infty), L^1(\mathbb{R})\right)$, given by (2.5) above, which satisfies: $u\in C^\infty(\mathbb{R}\times(0,\infty))$ and $u(\cdot,t)\in C^0\left((0,\infty), W^{k,p}(\mathbb{R})\right)$ for every $k\geq 1$, $p\geq 1$. Here, $W^{k,p}(\mathbb{R})$ is the Sobolev space of functions in $L^p(\mathbb{R})$ whose k-th order derivatives belong to $L^p(\mathbb{R})$. Moreover, by (2.5) and standard heat kernel estimates one gets that:

$$\|u(\cdot,t)\|_{L^2(\mathbb{R})} = O(t^{-\frac{1}{4}}), \qquad \|u(\cdot,t)\|_{L^\infty(\mathbb{R})} = O(t^{-\frac{1}{2}}), \tag{2.6}$$

$$\|u_x(\cdot,t)\|_{L^2(\mathbb{R})} = O(t^{-\frac{3}{4}}), \qquad \|u_x(\cdot,t)\|_{L^\infty(\mathbb{R})} = O(t^{-1}), \tag{2.7}$$

$$\|u_{xx}(\cdot,t)\|_{L^2(\mathbb{R})} = O(t^{-\frac{5}{4}}), \qquad \|u_{xx}(\cdot,t)\|_{L^\infty(\mathbb{R})} = O(t^{-\frac{3}{2}}), \tag{2.8}$$

and so on.

A more refined analysis in [56] shows that the asymptotic limits:

$$\gamma_p := \lim_{t\to\infty} t^{\frac{1}{2}\left(1-\frac{1}{p}\right)} \|u(\cdot,t)\|_{L^p(\mathbb{R})}, \quad 1 \le p \le \infty, \tag{2.9}$$

are well defined and have the following values. Let m be the solution mass, that is:

$$m = \int_{-\infty}^{\infty} u(x,t)\,dx = \int_{-\infty}^{\infty} u_0(x)\,dx. \tag{2.10}$$

For $1 < p \le \infty$, we have:

$$\gamma_p = \frac{|m|}{\sqrt{4\pi\nu}}(4\nu)^{\frac{1}{2p}} \frac{2\nu}{bm}\left(1 - e^{-\frac{m}{2\nu}}\right) \|\mathcal{F}\|_{L^p(\mathbb{R})} \tag{2.11}$$

with $\mathcal{F} \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R})$ defined by:

$$\mathcal{F}(x) = \frac{e^{-x^2}}{\lambda - h\,\mathrm{erf}(x)}, \tag{2.12}$$

where erf$(\cdot)$ is the error function:

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-\xi^2}\,d\xi \tag{2.13}$$

and λ, h are given by:

$$\lambda = \frac{1 + e^{-\frac{bm}{2\nu}}}{2}, \quad h = \frac{1 - e^{-\frac{bm}{2\nu}}}{2}. \tag{2.14}$$

When $p = 1$, the limit (2.9) is simply:

$$\gamma_1 = \lim_{t\to\infty} \|u(\cdot,t)\|_{L^1(\mathbb{R})} = |m|, \tag{2.15}$$

and we further have: $\|u_x(\cdot,t)\|_{L^1(\mathbb{R})} = O(t^{-\frac{1}{2}})$, $\|u_{xx}(\cdot,t)\|_{L^1(\mathbb{R})} = O(t^{-1})$, and so on.

These results will be used in Section 5 as further evidence for the accuracy of the numerical approximation scheme developed in the next two sections.

3 FINITE ELEMENT SCHEME

We consider the following auxiliary Dirichlet's problem:

$$\frac{\partial \tilde{u}}{\partial t} + \frac{2b}{(l_b - l_a)}\tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} = \frac{4\nu}{(l_b - l_a)^2}\frac{\partial^2 \tilde{u}}{\partial \tilde{x}^2}, \quad (\tilde{x} \in (-1,1), t > 0), \tag{3.1}$$

$$\tilde{u}(\tilde{x}, 0) = \tilde{g}(\tilde{x}), \quad (\tilde{x} \in (-1,1)), \tag{3.2}$$

$$\tilde{u}(-1,t) = \tilde{u}(1,t) = 0, \quad (t > 0), \tag{3.3}$$

where $\tilde{x} := 2x/(l_b - l_a) - (l_a + l_b)/(l_b - l_a)$ is the dimensionless space variable, $[l_a, l_b]$ is the reference domain, and $\tilde{g}(\tilde{x}) := g(\tilde{x}(l_b - l_a)/2 + (l_a + l_b)/2)$. From now on we will work with this space dimensionless problem and, for the sake of simplicity, we denote the domain length $l_{ab} := l_b - l_a$, and we will omit the tilde, i.e., we will denote $\tilde{x}$ simply by x and $\tilde{u}$ by u.

Following the Rothe's method, we start by discretizing equation (3.1) in time. To this end, we consider the following θ-scheme for the time discretization of equation (3.1):

$$\begin{aligned}\frac{u^{m+1} - u^m}{\delta_t} =& -\frac{2b\theta}{l_{ab}} u^{m+1} \frac{\partial u^{m+1}}{\partial x} - \frac{2b(1-\theta)}{l_{ab}} u^m \frac{\partial u^m}{\partial x} \\ & + \frac{4\nu\theta}{l_{ab}^2} \frac{\partial^2 u^{m+1}}{\partial x^2} + \frac{4\nu(1-\theta)}{l_{ab}^2} \frac{\partial^2 u^m}{\partial x^2}\end{aligned} \tag{3.4}$$

where $u^0 = u(x, 0)$, u^m denotes the approximation of $u(x, t_m)$, $m = 1, 2, \ldots$, $t_m = m\delta_t$, δ_t is a given time step size and $0 \leq \theta \leq 1$. For the sake of simplicity, from now on we denote u^{m+1} by u and u^m by u^0.

Now, we consider the following weak formulation of the problem defined by equations (3.4), (3.2) and (3.3): given $u^0 \in H_0^1(-1, 1)$ find $u \in H_0^1(-1, 1)$ such that:

$$\begin{aligned}&(\varphi, u) + \frac{2\theta\delta_t}{l_{ab}} \left(\varphi, bu\frac{\partial u}{\partial x}\right) + \frac{4\theta\delta_t}{l_{ab}^2} \left(\frac{\partial \varphi}{\partial x}, \nu\frac{\partial u}{\partial x}\right) - \\ &\left(\varphi, u^0\right) + \frac{2(1-\theta)\delta_t}{l_{ab}} \left(\varphi, bu^0\frac{\partial u^0}{\partial x}\right) + \frac{4(1-\theta)\delta_t}{l_{ab}^2} \left(\frac{\partial \varphi}{\partial x}, \nu\frac{\partial u^0}{\partial x}\right) = 0\end{aligned} \tag{3.5}$$

for all $\varphi \in H_0^1(-1, 1)$. Here, and in the sequel, $(\cdot, \cdot)$ denotes the $L^2(-1, 1)$ inner product.

Let us consider the following second order finite element triple $(\mathcal{K}, P_2(\mathcal{K}), \Sigma)$, where the cells $\mathcal{K} \subset \mathcal{T}_h$ are line segments forming a regular triangulation $\mathcal{T}_h$ of the segment $[-1, 1]$, the element shape functions $P_2(\mathcal{K}) = \{v : \mathcal{K} \to \mathbb{R}, v(x) = a_0 + a_1 x + a_2 x^2, a_0, a_1, a_2 \in \mathbb{R}\}$ are second order polynomials, and the degrees of freedom Σ are located at the end points of each $\mathcal{K}$ and its middle point (see, for instance, [34, 41]). This allows us to define the finite element space:

$$V_h := \{v \in C^0(-1, 1) : v|_{\mathcal{K}} \in P_2(\mathcal{K}), \forall \mathcal{K} \in \mathcal{T}_h\} \subset H_0^1(-1, 1).$$

Then, following the Galerkin least squares method (see, for instance, [41]), we iteratively approximate the solution of (3.1) subjected to (3.2) and (3.3) by the solution of the following full discrete problem: given $u_h^0 \in V_h$ find $u_h \in V_h$ such that:

$$\begin{aligned}&(\varphi_i, u_h) + \frac{2\theta\delta_t}{l_{ab}} \left(\varphi_i, bu_h\frac{\partial u_h}{\partial x}\right) + \frac{4\theta\delta_t}{l_{ab}^2} \left(\frac{\partial \varphi_i}{\partial x}, \nu\frac{\partial u_h}{\partial x}\right) + \\ &\theta\delta_t s_h(\varphi_i, u_h) - \left(\varphi_i, u_h^0\right) + \frac{2(1-\theta)\delta_t}{l_{ab}} \left(\varphi_i, bu_h^0\frac{\partial u_h^0}{\partial x}\right) + \\ &\frac{4(1-\theta)\delta_t}{l_{ab}^2} \left(\frac{\partial \varphi_i}{\partial x}, \nu\frac{\partial u_h^0}{\partial x}\right) + (1-\theta)\delta_t s_h(\varphi_i, u_h^0) = 0\end{aligned} \tag{3.6}$$

for all φ_i, $i = 1, 2, \ldots, N$, in the basis of the finite element space V_h. The Galerkin least square stabilization term s_h is given by:

$$s_h(\varphi, u) := \sum_{T \in \mathcal{T}_h} \delta_T \left(-\frac{4\nu}{l_{ab}^2} \frac{\partial^2 \varphi}{\partial x^2} + \frac{2b}{l_{ab}} \varphi \frac{\partial \varphi}{\partial x}, -\frac{4\nu}{l_{ab}^2} \frac{\partial^2 u}{\partial x^2} + \frac{2b}{l_{ab}} u \frac{\partial u}{\partial x} \right), \tag{3.7}$$

and the local stabilization parameter δ_T is chosen such that [11]:

$$\delta_T = \delta_0 l_{ab0} h_T \left(\frac{4\nu}{l_{ab0} h_T} + \|2bu_h\|_{L^\infty(-1,1)} \right)^{-1},$$

where δ_0 is a small positive constant, and l_{ab0} is the length of the initial reference domain (see next section for more details about the choice of the reference domain). We observe that using the fixed l_{ab0} leads to a consistent formulation, since $\delta_T \to 0$ as $h_T \to 0$, but not otherwise, since $l_{ab} \to \infty$. Moreover, this choice has leaded us to a stable numerical scheme as we may observe from our numerical experiments (see Section 5).

At each time step, we solve the nonlinear system of equations (3.6) by the Newton's method. The Newton's formulation then reads: given $u_h^0 \in V_h$ we iteratively compute approximations u_h^{n+1} of u_h by iterating:

$$J(u_h^{(n)}) \delta u^{(n)} = -F(u_h^{(n)}), \tag{3.8a}$$

$$u_h^{(n+1)} = u_h^{(n)} + \delta u^{(n)}, \tag{3.8b}$$

where $F(u_h^n)$ denotes the left-hand-side of equation (3.6) substituting there u_h by u_h^n, δu^n is the Newton update, and the Jacobian matrix $J(u) = [\mathrm{j}_{i,j}]_{i,j=0}^{N,N}$ have its elements defined by:

$$\begin{aligned} \mathrm{j}_{i,j} := {} & (\varphi_i, \varphi_j) + \frac{2\theta\delta_t}{l_{ab}} \left(\varphi_i, b\varphi_j \frac{\partial u}{\partial x} \right) + \frac{2\theta\delta_t}{l_{ab}} \left(\varphi_i, bu \frac{\partial \varphi_j}{\partial x} \right) \\ & + \frac{4\theta\delta_t}{l_{ab}^2} \left(\frac{\partial \varphi_i}{\partial x}, \nu \frac{\partial \varphi_j}{\partial x} \right) + \theta\delta_t s_h'(\varphi_i, u; \varphi_j), \end{aligned} \tag{3.9}$$

where N counts for the number of elements in the triangulation, and

$$s_h'(\varphi_i, u; \varphi_j) := \sum_{i=0}^{N-1} \delta_{T_i} \left(-\frac{4\nu}{l_{ab}^2} \frac{\partial^2 \varphi_i}{\partial x^2} + \frac{2b}{l_{ab}} \varphi_i \frac{\partial \varphi_i}{\partial x}, -\frac{4\nu}{l_{ab}^2} \frac{\partial^2 \varphi_j}{\partial x^2} + \frac{2b}{l_{ab}} \varphi_j \frac{\partial u}{\partial x} + \frac{2b}{l_{ab}} u \frac{\partial \varphi_j}{\partial x} \right). \tag{3.10}$$

We now lead the discussion to the implementation of this Galerkin least squares finite element method (GLS-FEM) to simulate the Burgers' equation defined on the whole real line and subjected to an initial condition with compact support.

4 IMPLEMENTATION SCHEME

Because of the convective-diffusive nature of the Burgers' equation, very fine meshes are demanded to accurately compute the solution for small times, but as time increases the solution tends to relax allowing the application of less refined meshes. By assuming an initial condition with compact support, numerical simulations of the auxiliary Dirichlet's problem (3.1)-(3.3) may produce accurate solutions for finite times. To ensure the accuracy we just need to choose appropriate time step and spatial mesh length, and pick the domain $[l_a, l_b]$ sufficiently large. However, the larger the physical time we would like to consider the larger $[l_a, l_b]$ should be.

The convective effects are predominant for small times and it is appropriate to work with a small $[l_a, l_b]$, which reduces the demanding on the number of vertices of the finite element mesh. On the other hand, as time increases, the solution spreads out demanding a larger $[l_a, l_b]$ but a less refined mesh. We deal with this paradigm as follows.

Let $K := \overline{\{x \in \mathbb{R} : g(x) \neq 0\}} \subset \mathbb{R}$ be the compact support of the initial condition. Without loss of generality, we assume that $0 \in K$ and denote $d = \max_{x \in K}\{|x|\}$. Also, let's denote by $\mathcal{K}_\Gamma$ the boundary elements of the finite element space $(\mathcal{K}, P_2(\mathcal{K}), \Sigma)$. With this in mind, the implementation idea is to start simulating the auxiliary Dirichlet's problem (3.1)-(3.3) by choosing an appropriate domain $[l_a, l_b] \supset K$ such that $l_b - l_a > d$ (see Fig. 1). Then, at each time iteration m we check if the numerical support $\tilde{K}_m := \{x \in \Sigma : |u_h^m(x)| > 10^{-15}\}$ is still a subset of $\mathcal{T}_h \setminus \{\mathcal{K}_\Gamma\}$. If it is not the case, then we increase $|l_a|$ and/or $|l_b|$, interpolate the numerical solution u_h^m onto $\mathcal{T}_h$, and we solve the next time step.

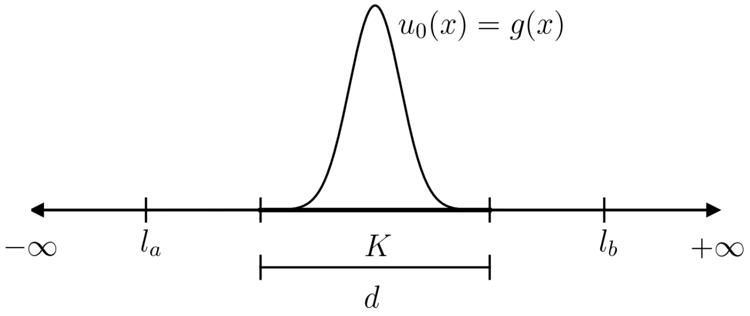

Figure 1: Illustration of the choice of the reference domain $[l_a, l_b]$.

We point out that the above implementation scheme does not demand one to rewrite the finite element triangulation at each increasing of the reference domain $[l_a, l_b]$, since we always simulate using the same fixed triangulation built in the domain $[-1, 1]$.

As time increases and the solution diffuses out, it may be possible to increase the time step δ_t gaining performance of the computations. This is particularly important for us, since we are also interested in investigating the accuracy of the proposed numerical scheme for large times. Therefore, we implemented a time step corrector based on an estimate of the rate of convergence of the Newton iterations. More precisely, at each time step m with step length δ_t^m, we compute the convergence rate estimation $\rho = \left(\|\delta u^n\|_2/\|\delta u^0\|_2\right)^{1/n}$, where n is the number of Newton iterations required for convergence at this time step m, and $\|\cdot\|_2$ denotes the l_2 vector norm (see [36] for more about this typical convergence rate estimation). We assume the convergence of the Newton iterations when $\|F(u_h^n)\|_2 < 10^{-10}$. Then, if $\rho > 0.1$, we increase δ_t by 10 %, and if

$\rho < 0.05$, we decrease it by the same percentage. To ensure a good precision, we allowed the time step corrector to take effect at most at each one-hundred time steps and we set 10^{-1} as the larger time step allowed.

We summarize the implementation procedure as follows:

1. Set a uniform mesh with N vertices built in the domain $[-1, 1]$.
2. Set the finite element triangulation $\mathcal{T}_h$.
3. Set an appropriate $[l_a, l_b] \supset K$.
4. Set the initial solution vector $u_h^0 \leftarrow [g(x_i)]_{i=0}^{2N+1}$, where x_i is the abscissa of the i-th degree of freedom.
5. Set the present solution vector $u_h \leftarrow u_h^0$.
6. Set the time step δ_t.
7. Loop over time steps:
    - (a) Loop over Newton steps:
        - i. Assemble the Newton system.
        - ii. Solve the system.
        - iii. Set $u_h \leftarrow u_h + \delta u_h$.
    - (b) If $\tilde{K} \not\subset \mathcal{T}_h \setminus \{\mathcal{K}_\Gamma\}$, then:
        - i. Set to double l_a and/or l_b.
        - ii. Interpolate u_h considering the new reference domain.
    - (c) Correct time step.
    - (d) Set $u^0 \leftarrow u$.

The numerical simulations were implemented in C++ using the *deal.II* open source finite element library [6, 7]. We applied the UMFPACK sparse direct linear solver implemented there to compute the Newton update δu^m from equation (3.8a). Evaluation of the analytic solution, its asymptotic behavior and post processing procedures were performed with the help of the Python-based ecosystem *Scipy*.

4.1 Computation of the analytic solution

Computations of the analytic solution (Eq. (2.5)) become not trivial for small values of ν, because it suffers the phenomena of loss of significance using finite-precision arithmetic. To ensure a high precision in such computations we used the library for floating-point arithmetic with arbitrary precision *mpmath*. The integrals in (2.5) were computed using the Gauss-Legendre quadrature implemented in *mpmath.quad* Python code.

For small values of ν this quadrature scheme was not able to accurately compute the analytical solution only by itself even setting hundreds of digits. However, we have satisfactorily overcome this issue (at least for ν as small as 10^{-3}) by observing that: (1) for $x = y$ the integrand of the numerator in equation (2.5) is null; (2) the integrands have huge scale differences from the domain $(-\infty, x)$ to (x, ∞); (3) the integrands exponentially tend to zero for $|x - y| \to \infty$. These observations motivated us to rewrite those integrals as follows:

$$\int_{-\infty}^{\infty} \mathcal{I}(x, y)\, dy = \sum_{i=0}^{N_0-1} \int_{a_i}^{a_{i+1}} \mathcal{I}(x, y)\, dy + \sum_{j=0}^{N_1-1} \int_{b_j}^{b_{j+1}} \mathcal{I}(x, y)\, dy,$$

where $\mathcal{I}(x, y)$ denote the integrands, $a_i = -L_0 + ih_0$, $h_0 = (x + L_0)/N_0$, $x > -L_0$, $i = 0, 1, \ldots, N_0$, $b_j = x + jh_1$, $h_1 = (L_1 - x)/N_1$, $x < L_1$, $j = 0, 1, \ldots, N_1$.

This implies that in order to evaluate the analytical solution u at a given point x, we computed $2(N_0 + N_1)$ integrals using the already mentioned quadrature procedure. By choosing appropriately L_0, L_1, N_0 and N_1 (which strongly depend on ν), we could compute the analytical solution with at least 10 precision digits.

5 NUMERICAL EXPERIMENTS

Here we present some numerical experiments with the proposed GLS-FEM scheme applied to the Burgers' equation on the real line. We first discuss the accuracy of the transient numerical solutions, and then explore their asymptotic behavior. Moreover, in Subsection 5.3 we finish our numerical experiments discussing an application to gas dynamic phenomenon.

5.1 Transient solutions

In the following we discuss the performance of the proposed numerical scheme for $|b| = 1$ and for different diffusion parameters ν at several times t. We assume the following initial condition:

$$u_0(x) = \begin{cases} \alpha\, e^{-10x^2} & , -2 \le x \le 2, \\ 0 & , \text{otherwise} \end{cases} ,$$

where we have chosen $\alpha \approx 0,89206$, which give us an initial solution with compact support and mass $m = \|u_0\|_{L^1(\mathbb{R})} = 0.5$. The transient numerical solutions presented here were obtained by setting the initial domain $[l_a, l_b] = [-2, 2]$ and the initial time step $\delta_t = 10^{-4}$.

To investigate the accuracy of the computations we present the evolution of the global relative error of the computed numerical solutions, i.e.:

$$\epsilon(t) := \frac{\|u_h(\cdot, t) - u(\cdot, t)\|_{L^2(\mathbb{R})}}{\|u(\cdot, t)\|_{L^2(\mathbb{R})}},$$

where u_h denotes the GLS-FEM solution and u denotes the analytic solution. The integrals involved on the computation of ϵ were approximated by the composite Simpsons rule with $2N+1$ sub-intervals, where N is the number of cells of the triangulation of u_h.

Firstly, let us shortly discuss about the choice of the parameters of δ_0 (for the GLS-stabilization) and θ (for the time scheme). The later may be heuristically chosen as small as enough to stabilize the numerical computations. After extensive numerical experiments we concluded that $\delta_0 = 0.1$ is an appropriate choice for $1 \leq \nu \leq 10^{-4}$, which is corroborated with the following reported results.

The time scheme parameter θ is classically chosen as: (1) $\theta = 0$ the explicit Euler method; (2) $\theta = 0.5$ the Crank-Nicolson method; (3) $\theta = 1$ the implicit Euler method. As our initial condition is very smooth, it is expected that $\theta = 0.5$ should lead to a good numerical performance, i.e. a good balance between computational costs and precision. As an example, let us assume $b = 1.0$, $\nu = 1$, and compute the GLS-FEM solution at $t_f = 1000$s using $N = 400$. With $\theta = 0$ the proposed numerical scheme was unstable yielding to spurious solutions. With $\theta = 1$ computations needed 16375 time steps and 16421 overall Newton steps to yield a numeric solution with an overall relative error $\epsilon(t_f) = 2.72 \times 10^{-1}$. Now, with $\theta = 0.5$ we need 16375 time steps, 16413 overall Newton steps, and reached a numeric solution with $\epsilon(t_f) = 2.17 \times 10^{-6}$.

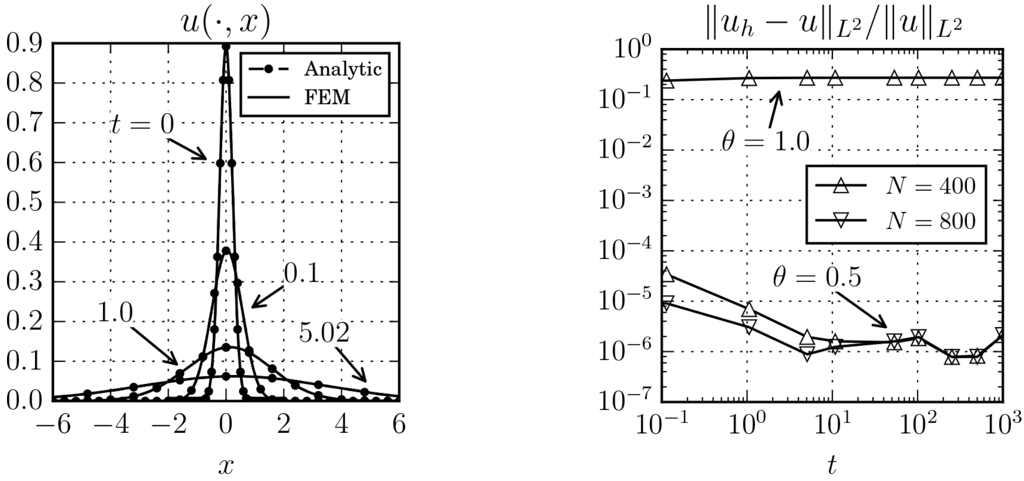

Figure 2: Transient numerical solutions *versus* analytic solutions for $b = 1$ and $\nu = 1$. Left: solution profiles at $t = 0.0$, 0.1, 1.0 and 5.02. Right: relative error on the L^2-norm for numerical solutions with meshes of $N = 400$ and 800.

Figure 2 presents a comparison between GLS-FEM numerical solutions and analytic solutions computed at several times $0 \leq t \leq 1000$. The right graphic in Figure 2 presents the evolution of the global relative error of the computed numerical solutions. The figure shows ϵ yielded with $\theta = 1.0$ and $N = 400$, and also with $\theta = 0.5$ and $N = 400$, 800. It is clear that the choice of $\theta = 0.5$ has produced more accurate results with $\epsilon(t) < 10^{-4}$ for both meshes. From now on, we will assume $\theta = 0.5$.

The left graphic of Figure 2 shows solution profiles at $t = 0.0$, 0.1, 1.0 and 5.02. Dashed lines with points present analytic solution profiles, and solid lines present the computed GLS-FEM solutions with $N = 400$. For the chosen parameters the Burgers' equation is diffusion dominated (Péclet number $Pe < 1$), and we can observe that the solution rapidly diffuses out.

Now, Figure 3 presents the results found when $b = -1$ and $\nu = 0.1$. In this case, the convective effects are stronger, but the solution is still strongly diffusive, as we can observe by the solution profiles given on the left graphic of this figure. The numerical solution has again a good accuracy as one can see at the right graphic of this figure. Moreover, we point out that for small times t the numerical scheme has a truncation error of at least the order h^2. However, for large times the diffusion effects are much stronger and, together with the increasing of the time step, causes the lost of such truncation order.

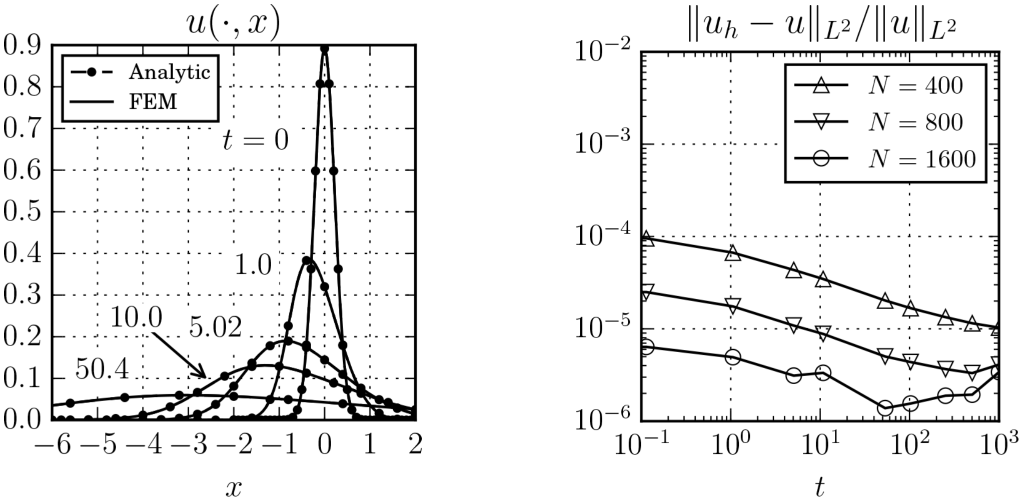

Figure 3: Transient numerical solutions *versus* analytic solutions for $b = -1$ and $\nu = 10^{-1}$. Left: numerical solution with $N = 400$ and analytic solution profiles at $t = 0.0$, 1.0, 5.02, 10.0 and 50.4. Right: relative error on the L^2-norm for numerical solutions with meshes of $N = 400$, 800 and 1600.

As we decrease the diffusion coefficient we need more refined meshes to obtain such accurate results. The Figure 4 shows the comparison between numerical and analytic solution for the case of $b = 1$ and $\nu = 10^{-2}$. Here, the convection effects are much stronger for small times and even a mesh with $N = 1600$ was not enough to ensure a relative error of 10^{-4}. Nevertheless, the results presented in this figure indicate that further refinements will produce such accurate numerical results. We will come back to this point later.

As we may expect by further decreasing the diffusion coefficient even a mesh with $h_T \approx 10^{-3}$ yields numerical results with a relative accuracy of just $\epsilon(t) < 10^{-1}$. This is the case we can observe in Figure 5, which presents results of simulations for the case of $b = -1$ and $\nu = 10^{-3}$. Again, the computed relative errors indicate we may obtain more accurate solutions by working with more refined meshes, but it also indicates the time discretization scheme (more precisely, the time step size) is getting more and more importance as we decrease the diffusion parameter.

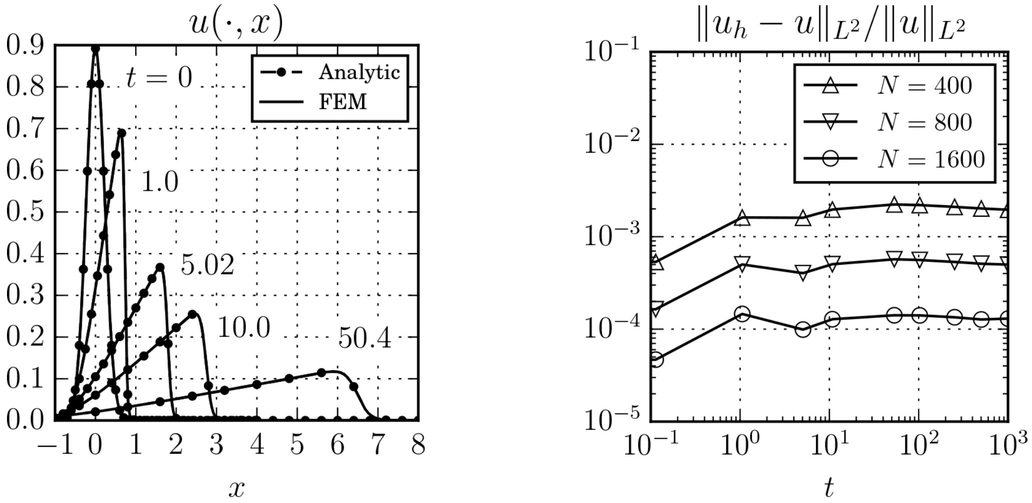

Figure 4: Transient numerical solutions *versus* analytic solutions for $b = 1$ and $\nu = 10^{-2}$. Left: numerical solution with $N = 400$ and analytic solution profiles at $t = 0.0$, 1.0, 5.02, 10.0 and 50.4. Right: relative error on the L^2-norm for numerical solutions with meshes of $N = 400$, 800 and 1600.

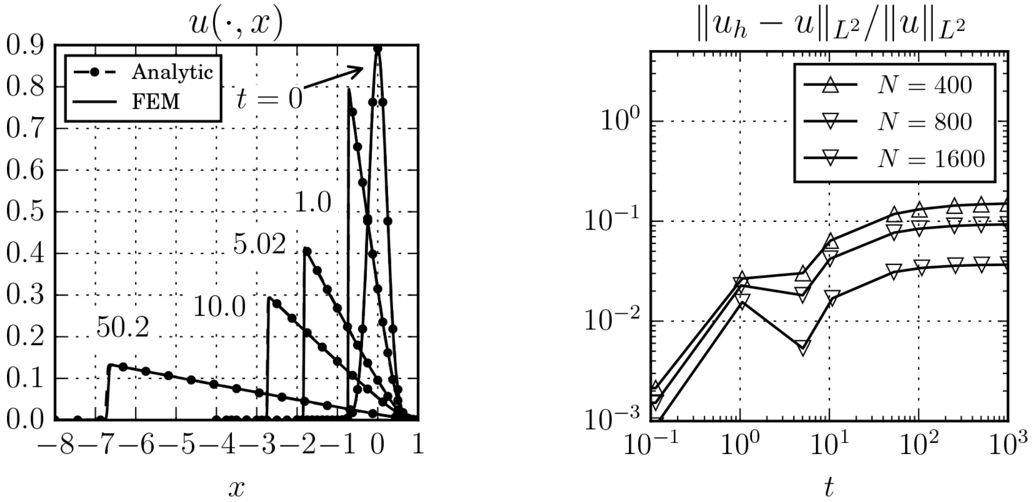

Figure 5: Transient numerical solutions *versus* analytic solutions for $b = -1$ and $\nu = 10^{-3}$. Left: numerical solution with $N = 800$ and analytic solution profiles at $t = 0.0$, 1.0, 5.02, 10.0 and 50.2. Right: relative error on the L^2-norm for numerical solutions with meshes of $N = 400$, 800 and 1600.

Since it is very computationally demanding to compute the analytic solution for smaller values of the diffusion parameter, we are not further able to globally compare the numerical and analytic solution by accurately evaluating the relative error ϵ. However, we may still investigate the accuracy of the numerical scheme by analyzing its asymptotic behavior.

5.2 Asymptotic behavior

As a further evidence of the accuracy of the proposed numerical scheme, we compare the asymptotic behavior of the numerical and analytic solutions by evaluating the γ_p limits defined in equations (2.9), (2.11), and (2.15). This is done by computing what we call the numerical γ_p, which we define as:

$$\tilde{\gamma}_p := t_f^{\frac{1}{2}\left(1-\frac{1}{p}\right)} \|u_h(\cdot, t_f)\|_{L^p(\mathbb{R})}, \tag{5.1}$$

where t_f is such that $\|F(u_h(\cdot, t_f))\|_2 < 10^{-9}$, i.e, the l_2 vector norm of the residual at $t = t_f$ is less or equal to 10^{-9}. We note that this provides an approximation of γ_p as the numerical solution u_h approximates the analytic solution u, and t_f is large enough. Moreover we observe that this is a very fine test of the accuracy of the numerical solution, since it allowed us to check it for very large times.

In Table 1 we present the computed values of $\tilde{\gamma}_p$ *versus* γ_p for $p = 1$ (mass) and for $p = 2$ for solutions with diffusion coefficients from 1 to 10^{-4}. The numerical scheme parameters were chosen as in the last subsection.

In opposition to the analytic solution given by equation (2.5), we observe that γ_p can be easily computed also for very small diffusive parameters. This allows us to globally measure the accuracy of the numerical solution even for high convective regimes. Going back to Table 1, the comparison between $\tilde{\gamma}_1$ and γ_1 corroborates that the numerical scheme is not mass conserving, but the mass loss can be kept low by using a sufficiently refined mesh.

Finally we note that $\tilde{\gamma}_2$ also under-determines γ_2. This is a qualitative indication of the good behavior of the numerical solution for large times, since $t^{1/4}\|u(\cdot, t)\|_{L^2(\mathbb{R})}$ is a monotonic increasing function for large times.

5.3 An application to gas dynamic phenomena

We finish our numerical experiments with an application to gas dynamic phenomenon. The convection and decay of a compression pulse of an isentropic gas can be modeled by the Burgers' equation (1.1) with a Dirac delta function at zero time (see [23], Section 7.2). Defining a Reynolds number by $\mathrm{Re} := m/(2\nu)$, we have the following analytic solution

$$u(x, t) = \sqrt{\frac{2\nu}{t}} \frac{e^{-0.5n^2}}{P + I_C}, \quad (t > 0), \tag{5.2}$$

where $n = x/(2\nu t)^{1/2}$,

$$P = \frac{\sqrt{2\pi}}{e^{\mathrm{Re}-1}}, \quad I_C = \int_n^\infty e^{-0.5y^2} dy.$$

Table 1: Numerical $\tilde{\gamma}_p$ *versus* analytic γ_p for $p = 1$ and $p = 2$.

ν	N	t_f	$\tilde{\gamma}_1$	γ_1	$\tilde{\gamma}_2$	γ_2
1.0	400	2302.52	0.500000	0.500000	0.223280	0.223280
	800	2252.52	0.500000		0.223280	
10^{-1}	400	5942.52	0.499997	0.500000	0.392038	0.392044
	800	5242.52	0.499999		0.392039	
	1600	4502.52	0.500000		0.392039	
10^{-2}	400	30612.5	0.499675	0.500000	0.540157	0.540443
	800	23222.5	0.499919		0.540368	
	1600	17602.5	0.499980		0.540420	
10^{-3}	400	75321.9	0.488639	0.500000	0.562050	0.571942
	800	59621.4	0.493748		0.566502	
	1600	55088.3	0.497676		0.569917	
	3200	41902.3	0.499300		0.571323	
	6400	34198.3	0.499808		0.571761	
10^{-4}	400	$> 10^5$	0.496048	0.500000	0.572933	0.576621
	800	$> 10^5$	0.485923		0.564271	
	1600	$> 10^5$	0.484147		0.562800	
	3200	85778.5	0.485278		0.563789	
	6400	68903.4	0.491352		0.569077	
	12800	65027.8	0.496346		0.573414	
	25600	52017.5	0.498787		0.575524	

The proposed GSL-FEM scheme can be applied to simulate this physical problem, for instance, by simulating (1.1)-(1.2) with

$$g(x) = \frac{m}{\sigma\sqrt{2\pi}} e^{-0.5(x/\sigma)^2}, \quad (x \in \mathbb{R}), \tag{5.3}$$

where we have changed the Dirac function at $x = 0$ by a Gaussian function of mass m.

Figure 6 shows the analytic solution (5.2) and the GLS-FEM numerical solution of problem (1.1)-(1.2) with the above $g(x)$ at $t = t_f$ for several values of Re. In all cases we have set $m = 1.0, \sigma = 0.01$, $N = 1600$, and t_f as defined in the last subsection. Again we observe a very good agreement between numerical and analytic solutions for very large times, even that here the initial solution has been grossly approximated. The effect of such approximation is however more notable for small times, which indicates a time delay in the numerical solutions.

6 FINAL CONSIDERATIONS

In this work we have presented an efficient Galerkin least squares finite element scheme to simulate the Burgers' equation on the whole real line and subjected to initial conditions with

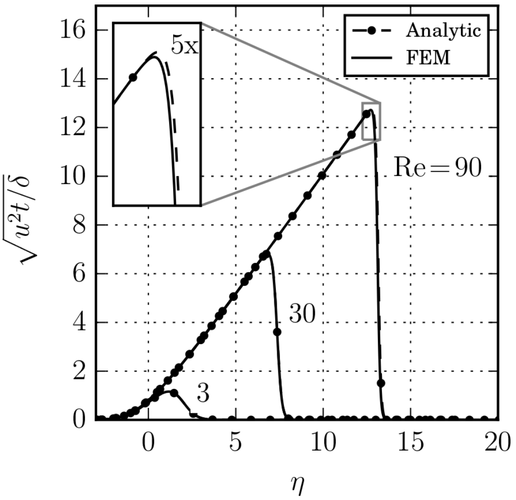

Figure 6: Analytic solution (5.2) and GLS-FEM solution for the convection and decay of a compression pulse. Re = 3: solution computed at $t_f = 5704$s. Re = 30: $t_f = 24838$s. Re = 90: $t_f = 40987$s.

compact support. The scheme consists in computing the finite element discretization of a sequence of dimensionless spatially forms of the Burgers' equation on a fixed triangulation and parameterized by its domain, which is chosen to contain the numerical support of the solution at each time step.

In order to investigate the accuracy of the proposed scheme, we performed direct comparisons between numerical and analytic solutions. For moderated diffusion coefficients the comparisons showed the scheme can be very accurate if one works with sufficiently refined meshes. Moreover, by analyzing asymptotic parameters of the solutions we could argue that the scheme is accurate even for very large times.

As we may expect the scheme demands more and more refined meshes as we decrease the diffusion parameter. An alternative would be to work with automatic local refined meshes. We observe that this could be particularly tricky, because one will need to be careful at each time that the reference domain is enlarged.

Finally we recall the Burgers' equation is a prototype for many scientific related problems, and the proposed numerical scheme can be extended as well to assist the study of such related problems. For instance, it can be used to provide insights into analytic studies of problems on the whole real line.

RESUMO. Neste trabalho, apresentamos um eficiente esquema de elementos finitos com mínimos quadrados de Galerkin para simular a equação de Burgers na reta toda, sujeita a condições iniciais com suporte compacto. As simulações numéricas foram realizadas tomando-se uma sequência de problemas auxiliares de Dirichlet adimensionais no espaço e parametrizada pelo seu suporte numérico $\tilde{K}$. Tomando vantagem dos bem conhecidos efeitos convectivos e difusivos da equação de Burgers, as computações iniciam-se escolhendo $\tilde{K}$ de forma a conter o suporte da condição inicial e, conforme a solução se difunde, $\tilde{K}$ é aumentada apropriadamente. Por comparação direta entre as soluções analítica e numérica e, pelo seus comportamentos assintóticos, concluímos que o esquema proposto é preciso mesmo para tempos grandes. Assim, este pode ser aplicado para numericamente investigar propriedades desta e de similares equações em domínios não limitados.

Palavras-chave: equação de Burgers na reta, método de elementos finitos com mínimos quadrados de Galerkin, propriedades assintóticas.

REFERENCES

[1] M.B. Abd el Malek & S.M.A. El-Mansi. Group theoretic methods applied to Burgers' equation. *J. Comput. Appl. Math.*, **115** (2000), 1–12.

[2] E.N. Aksan. A numerical solution of Burgers' equation by finite element method constructed on the method of discretization in time. *Appl. Math. Comput.*, **170** (2005), 895–904.

[3] E.N. Aksan. Quadratic B-spline finite element method for numerical solution of the Burgers' equation. *Appl. Math. Comput.*, **174** (2006), 884–896.

[4] E.N. Aksan & A. Özdecs. A numerical solution of Burgers' equation. *Appl. Math. Comput.*, **156** (2004), 395–402.

[5] P. Arminjon & C. Beauchamp. Continuous and discontinuous finite element methods for Burgers' equation. *Comput. Methods Appl. Mech. Engrg.*, **25** (1981), 65–84.

[6] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin & D. Wells. The `deal.II` library, version 8.47. *Journal of Numerical Mathematics*, **24** (2016).

[7] W. Bangerth, R. Hartmann & G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, **33(4)** (2007), 24/1–24/27.

[8] C. Basdevant, M. Deville, P. Haldenwang, J.M. Lacroix, J. Quazzani, R. Peyret & P. Orlandi. Spectral and finite difference solutions of the Burgers' equation. *Comput Fluids*, **14** (1986), 23–41.

[9] M. Basto, V. Semiao & F. Calheiros. Dynamics in spectral solutions of Burgers equation. *J. Comput. Appl. Math.*, **205** (2006), 296–304.

[10] M.A.H. Bateman. Some recent researches on the motion of fluids. *Mon. Wea. Rev.*, **43** (1915), 163–170.

[11] M. Braack. Finite elemente. available online, `http://www.numerik.uni-kiel.de/~mabr/lehre/skripte/fem-braack.pdf`, Jan. 2015.

[12] J.M. Burgers. *The nonlinear diffusion equation*. Springer, (1974).

[13] J. Caldwell, R. Saunders & P. Wanless. A note on variation-iterative schemes applied to Burgers' equation. *J. Comput. Phys.*, **58** (1985), 275–281.

[14] J. Caldwell & P. Smith. Solution of Burgers' equation with a large Reynolds number. *Appl. Math. Modelling*, **6** (1982), 381–385.

[15] J. Caldwell, P. Wanless & A.E. Cook. A finite element approach to Burgers' equation. *Appl. Math. Modelling*, **5** (1981), 189–193.

[16] J. Caldwell, P. Wanless & A.E. Cook. Solution of Burgers' equation for large Reynolds number using finite elements with moving nodes. *Appl. Math. Modelling*, **11** (1987), 211–214.

[17] J.D. Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quart. Appl. Math*, **9**(3) (1951), 225–236.

[18] R. Courant & K.O. Friedrichs. Supersonic flow and shock waves. Springer, New York, (1948).

[19] A. Dogan. A Galerkin finite element method to Burgers' equation. *Appl. Math. Comput.*, **157** (2004), 331–346.

[20] L.C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics*. The American Mathematical Society, 2nd edition, (2010).

[21] C.A. Fletcher. *Numerical Solutions of Partial Differential Equations*, chapter Burgers' equation: a model for all reasons, pages 139–225. Nort-Holland, Amsterdam, (1982).

[22] C.A. Fletcher. A comparison of finite element and finite difference solutions of the one- and two-dimensional Burgers' equations. *J. Comput. Phys.*, **51** (1983), 159–188.

[23] C.A. Fletcher. The Galerkin method and Burgers' equation. In: Computational techniques for differential equations. Mathematics Studies 83. Ed.: J. Noye, North-Holland, (1984).

[24] A. Gorguis. A comparision between Cole-Hopf transformation and the decomposition method for solving Burgers' equations. *Appl. Math. Comput.*, **173** (2006), 126–136.

[25] M. Gülsu. A finite difference approach for solution of Burgers' equation. *Appl. Math. Comput.*, **175** (2006), 1245–1255.

[26] H.-d. Han, X.-n. Wu & Z.-l. Xu. Artificial boundary method for Burgers' equation using nonlinear boundary conditions. *J. Comput. Math.*, **24**(3) (2006), 295–304.

[27] A. Hashemian & H. Shodja. A meshless approach for solution of Burgers' equation. *J. Comput. Appl. Math.*, **220** (2008), 226–239.

[28] C.J. Holland. On the limiting behavior of Burger's equation. *J. Math. Anal. Appl.*, **57** (1977), 156–160.

[29] Y.C. Hon & X.Z. Mao. An efficient numerical scheme for Burgers' equation. *J. Comput. Appl. Math.*, **95** (1998), 37–50.

[30] E. Hopf. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Comm. Pure and Appl. Math.*, **3** (1950), 201–230.

[31] A.N. Hrymak, G.J. McRae & A.W. Westerberg. An implementation of a moving finite element method. *J. Comput. Phys.*, **63** (1986), 168–190.

[32] M. Inc. On numerical solution of Burgers' equation by homotopy analysis method. *J. Phys. A*, **372** (2008), 356–360.

[33] R. Jiwari. A hybrid numerical scheme for the numerical solution of the Burgers' equation. *Comput. Phys. Commun.*, **188** (2015), 59–67.

[34] C. Johnson. *Numerical solutions of partial differential equations by the finite element method*. Dover, (2009).

[35] M. Kadalbajoo & A. Awasthi. A numerical method based on Crank-Nicolson scheme for Burgers' equation. *Appl. Math. Comput.*, **182** (2006), 1430–1442.

[36] C.T. Kelley. *Solving nonlinear equations with the Newton's method*. SIAM, (2003).

[37] A.H. Khater, R.S. Temsah & M.M. Hassan. A Chebyshev spectral collocation method for solving Burgers'-type equations. *J. Comput. Appl. Math.*, **222** (2008), 333–350.

[38] S. Kutluay, A.R. Bahadir & A. Özdeş. Numerical solution of one-dimensional Burgers equation: explicit and exact-explicit finite difference methods. *J. Comput. Appl. Math.*, **103** (1999), 251–261.

[39] S. Kutluay, A. Esen & I. Dag. Numerical solutions of the Burgers' equation by the least-squares quadratic B-spline finite element method. *J. Comput. Appl. Math.*, **167** (2004), 21–33.

[40] C.A. Ladeia, N.M. Romero, P.L. Natti & E.R. Cirilo. Formulações semi-discretas para a equação 1d de Burgers. *TEMA (São Carlos)*, **14(3)** (2013), 319–331.

[41] Mats G. Larson & Fredrik Bengzon. *The Finite Element Method: Theory, Implementation, and Applications*. Springer-Verlag Berlin Heidelberg, (2013).

[42] V. Mukundan & A. Awasthi. Efficient numerical techniques for Burgers' equation. *Appl. Math. Comput.*, **262** (2015), 282–297.

[43] T. Öziş, E.N. Aksan & A. Özdeş. A finite element approach for solution of Burgers' equation. *Appl. Math. Comput.*, **139** (2003), 417–428.

[44] T. Öziş & Y. Aslan. The semi-approximate approach for solving Burgers' equation with high Reynolds number. *Appl. Math. Comput.*, **163** (2005), 131–145.

[45] T. Öziş, A. Esen & S. Kutluay. Numerical solution of Burgers' equation by quadratic B-spline finite elements. *Appl. Math. Comput.*, **165** (2005), 237–249.

[46] T. Ozis & A. Ozdes. A direct variational methods applied to Burgers' equation. *J. Comput. Appl. Math.*, **71** (1996), 163–175.

[47] E.Y. Rodin. On some approximate and exact solutions of boundary value problems for Burgers' equation. *J. Math. Anal. Appl.*, **30** (1970), 401–414.

[48] B. Saka & İ. Dağ. A numerical study of the Burgers' equation. *J. Frankl. Inst.*, **345** (2008), 328–348.

[49] L. Shao, X. Feng & Y. He. The local discontinuous Galerkin finite element method for Burgers' equation. *Math. Comput. Model.*, **54** (2011), 2943–2954.

[50] Z.-Z. Sun & X.-N. Wu. A difference scheme for Burgers equation in an unbounded domain. *Appl. Math. Comput.*, **209** (2009), 285–304.

[51] A.H.A.E. Tabatabaei, E. Shakour & M. Dehghan. Some implicit methods for the numerical solution of Burgers' equation. *Appl. Math. Comput.*, **191** (2007), 560–570.

[52] W.L. Wood. An exact solution for Burger's equation. *Commun. Numer. Meth. Engng.*, **22** (2006), 797–798.

[53] X. Wu & J. Zhang. Artificial boundary method for two-dimensional Burgers' equation. *Comput. Math. Appl.*, **56** (2008), 242–256.

[54] M. Xu, R.-H. Wang, J.-H. Zhang & Q. Fang. A novel numerical scheme for solving Burgers' equation. *Appl. Math. Comput.*, **217** (2011), 4473–4482.

[55] X.H. Zhang, J. Ouyang & L. Zhang. Element-free characteristic Galerkin method for Burgers' equation. *Eng. Anal. Boundary Elem.*, **33** (2009), 356–362.

[56] P.R. Zingano. Some asymptotic limits for solutions of Burgers equation. available at: `http://arxiv.org/pdf/math/0512503.pdf`, 1997. Universidade Federal do Rio Grande do Sul.

[57] S. Dhawan, S. Kapoor, S. Kumar & S. Rawat. Contemporary review of techniques for the solution of nonlinear Burgers equation. *J. Comput. Sci.*, **3** (2012), 405–419.