Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

NUMERICAL SIMULATIONS OF THE 1-D MODIFIED BURGERS EQUATION

Diogo T. Robaina

Escola Superior de
Propaganda e Marketing
R. do Rosário, 90 - Centro
Rio de Janeiro, RJ, 20041-002 BRAZIL

Sanderson L. Gonzaga de Oliveira

Departamento de Ciência da Computação
Universidade Federal de Lavras
Câmpus Universitário, C.P. 3037
Lavras, MG, 37200-000 BRAZIL

Mauricio Kischinhevsky

Instituto de Computação
Universidade Federal Fluminense
Av. Gal. Milton Tavares de Souza, s/n, São Domingos
Niterói, RJ, 24210-346 BRAZIL

Carla Osthoff

Laboratório Nacional de
Computação Científica
Av. Getulio Vargas, 333, Quitandinha
Petrópolis, RJ, 25651-075 BRAZIL

Alexandre C. Sena

Departamento de Informática e Ciência da Computação
Universidade do Estado do Rio de Janeiro
Rua São Francisco Xavier, 524, Maracanã
Rio de Janeiro, RJ, 20550-013 BRAZIL

ABSTRACT

This paper shows the results yielded by several numerical methods when applied to the 1-D modified Burgers' equation. In particular, the paper evaluates a new hybrid method against ten high-order methods when applied to the same equation. The novel numerical method for convection-dominated fluid or heat flows is based on the Hopmoc method and backward differentiation formulas. The results highlight that the new hybrid method yields promising accuracy results with regards to several existing high-order methods.

1 INTRODUCTION

Efficient numerical solution of evolutionary differential equations is essential in several areas in engineering and science. The one-dimensional modified Burgers' equation in the form

$$u_t + u^2 u_x - d u_{xx} = 0 \tag{1}$$

where $u(x,t)$ is the dependent variable, d is the viscosity parameter, and time t and space x are the independent parameters has strong nonlinear aspects of the governing equation in various practical transport problems. Specifically, this equation has a broad range of applications in several areas as a mathematical model for numerous phenomena such as nonlinear waves in a medium with low-frequency pumping or absorption, ion reflection at quasi-perpendicular shocks, turbulence transport, wave processes in a thermoelastic medium, transport and dispersion of pollutants in rivers, and sediment transport (Ray and Gupta 2018).

This paper provides numerical solutions to the 1-D modified Burgers' equation using a new algorithm that integrates backward differentiation formulas with the Hopmoc method. We refer to the new scheme as the BDFHM method. The results of the novel algorithm are compared with ten existing high-order methods when applied to the 1-D modified Burgers' equation.

Section 2 overviews recent publications in the field. Section 3 introduces the BDFHM method. Section 4 shows a computational analysis of the new method. Specifically, this section compares the results obtained by the new method with the results yielded by the original Hopmoc method presented in a previous publication (Oliveira et al. 2009). The same section describes the numerical results when applying the BDFHM method to the 1-D modified Burgers' equation and compares the results of the new method with ten existing high-order methods. Finally, Section 5 provides our conclusions and discusses future directions in this investigation.

2 RELATED WORK

Various mathematical methods have been used to solve the modified Burgers' equation. We provide here only a brief review of the main contributions in the field.

Researchers used the collocation method along with splines to solve the nonlinear equation. Ramadan and El-Danaf (2005) used the collocation method along with quintic B-splines to solve the modified Burgers' equation. The method yielded better results than did the collocation method along with sextic B-splines proposed by Ramadan et al. (2005). Saka and Dag (2008) applied time and space splitting techniques in conjunction with quintic B-spline collocation algorithms to solve the modified Burgers' equation.

Duan et al. (2008) applied a Lattice Boltzmann model to solve the modified Burgers' equation. Irk (2009) used a Crank-Nicolson central differencing approach for time integration and sextic B-spline functions for space integration to solve the equation. Temsah (2009) proposed a numerical scheme based on the El-Gendi method to solve the equation.

Bratsos (2010) used a finite-difference scheme based on fourth-order rational approximations to the matrix-exponential term in a two-time level recurrence relation for the numerical approximation to the modified Burgers' equation. Bratsos and Petrakis (2011) employed an explicit finite-difference scheme based on second-order rational approximations to the matrix-exponential term in a two-time level recurrence relation for the numerical approximation to the equation.

Roshan and Bhamra (2011) employed the Petrov-Galerkin method using a linear hat function as the trial function and a cubic B-spline function as the test function to provide numerical solutions to the modified Burgers' equation. Rong-Pei et al. (2013) solved the modified Burgers' equation by the local discontinuous Galerkin method. Zeytinoglu et al. (2018) used high-order finite-difference schemes to solve the modified Burgers' equation.

3 THE BDFHM METHOD

This section describes a solution of the 1-D advection-diffusion equation

$$u_t + v \cdot u_x = d \cdot u_{xx}, \text{ for } 0 \leq x \leq 1, \tag{2}$$

using a new method where d is the diffusion coefficient and v is a constant positive velocity. The advection-diffusion equation is of central importance in many physical systems, especially those involving fluid flow. Specifically, the numerical solution of the advection-diffusion transport arises from significant problems in physics and chemistry. Relevant examples of its use comprise the transport of contaminants in the air, groundwater, rivers, and lagoons, aerodynamics, astrophysics, biomedical applications, oil reservoir flow, in the modeling of semiconductors, geophysical flows, such as meteorology and oceanography (Cabral et al. 2017). A fluid transports the contaminant or chemical species and dissolves them in reactive or environment fluid flow problems. In computational hydraulics and fluid dynamics problems, the advection-diffusion equation can be used to represent quantities such as energy, heat, mass, vorticity, etc. (Ding and Zhang 2009).

Section 3.1 briefly describes the Hopmoc method. Section 3.2 introduces the BDFHM method.

3.1 Hopmoc Method

The Hopmoc method (see Oliveira et al. (2009) and references therein) is a fast and accurate method for the solution of convection-dominated fluid or heat flows. This method processes in an explicit approach such that the nodal update formulas employed are independent and can be used simultaneously at all mesh nodes.

The Hopmoc method employs finite-difference techniques in a similar way to the Hopscotch method (Gourlay 1970), which is applied to solve parabolic and elliptic partial differential equations. The Hopmoc method divides the set of unknowns into two subsets. The algorithm alternately approximates the unknowns dividing each time step into two semi-steps. For example, consider the use of a quadrangular mesh for the solution of a 2-D problem. In this scenario, an internal mesh node belongs to one of the subsets, and its four adjacent mesh nodes belong to the second subset. At each time semi-step, every unknown belonging to a subset is alternately updated using symmetrical explicit and implicit approaches. More specifically, the first time semi-step updates a subset of unknowns using an explicit strategy. The second implicit time semi-step uses the solution calculated in the previous time semi-step. Thus, no linear system is solved.

The Hopmoc approach evaluates semi-steps along characteristic lines employing concepts of the modified method of characteristics (Douglas and Russell 1955). Specifically, the Hopmoc method uses approximate solutions from previous time steps along the directional derivative following the characteristic line in a similar way to the modified method of characteristics. Therefore, the Hopmoc employs a semi-Lagrangian scheme, i.e., it uses a Eulerian structure, but the discrete equations come from a Lagrangian frame of reference. More precisely, the Hopmoc method employs a spacial discretization along the characteristic line from each mesh node. This method presents the first-order accuracy in both space and time variables.

3.2 BDFHM Method

Backward differentiation formulas (BDFs) are implicit methods for the numerical integration of differential equations. In short, BDFs are linear multistep methods that use information from previous time steps to increase the accuracy of an approximation to the derivative of a given function and time. This section integrates the Hopmoc method and BDFs.

In equation (2), u_t refers to the time derivative and not u evaluated at the discrete time step t. Nevertheless, we abuse the notation and now use t to denote a discrete time step so that $0 \leq t \leq T$, for T time steps. Figure 1 shows a 1-D discretization for the BDFHM method.

The unit vector $\tau = (v \cdot \delta t, \delta t) = \left(x - \bar{x}, t_{n+\frac{1}{2}} - t_n\right)$ represents the characteristic line associated with the transport $u_t + vu_x$ and $\bar{x}$ ($\bar{\bar{x}}$) is the "foot" of the characteristic line in the second (first) time semi-step (see Figure 1). The derivative in the direction of τ is given by

$$\begin{aligned}
u_\tau &= \nabla u \times \frac{\tau}{\|\tau\|} = u_\tau = (u_x, u_t) \times \frac{(v\delta t, \delta t)}{\sqrt{(v\delta t)^2 + (\delta t)^2}} \\
&= (u_x, u_t) \times \left(\frac{v\delta t}{\delta t \sqrt{(v)^2+1}}, \frac{\delta t}{\delta t \sqrt{(v)^2+1}} \right) \\
&= (u_x, u_t) \times \left(\frac{v}{\sqrt{(v)^2+1}}, \frac{1}{\sqrt{(v)^2+1}} \right) = \left(\frac{vu_x}{\sqrt{(v)^2+1}}, \frac{u_t}{\sqrt{(v)^2+1}} \right) \\
&= \frac{1}{\sqrt{(v)^2+1}} \times (vu_x + u_t) \Rightarrow u_\tau \times \sqrt{(v)^2+1} = vu_x + u_t = du_{xx}.
\end{aligned}$$

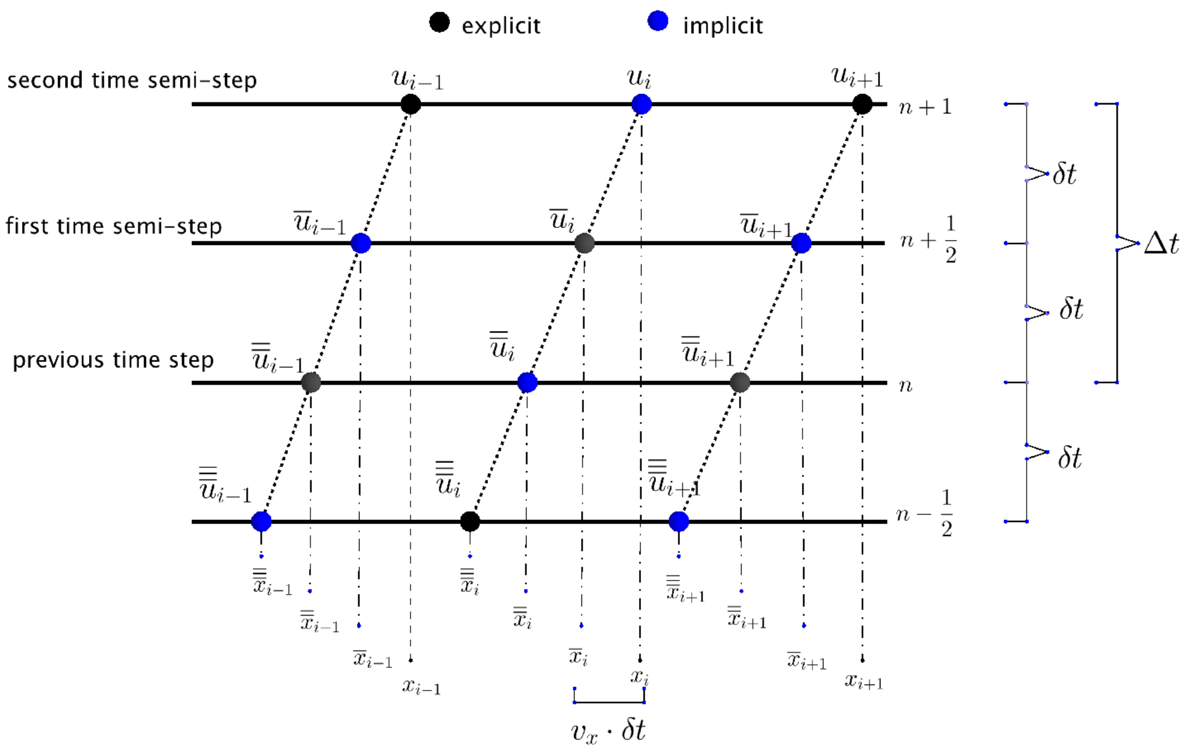

Figure 1: Variable values $\overline{\overline{u}}_i^{n-\frac{1}{2}}$ and $\overline{\overline{u}}_i^{n}$ $\left(\overline{\overline{u}}_i^{n} \text{ and } \overline{u}_i^{n+\frac{1}{2}}\right)$ are used to calculate $\overline{u}_i^{n+\frac{1}{2}}$ $\left(u_i^{n+1}\right)$ in the first (second) time semi-step in the BDFHM method when applied to a 1-D problem.

From the diffusion equation along the characteristic line, the modified method of characteristics approximates the directional derivative in an explicit form through the equation

$$\left(\frac{\overline{u}_i^{n+\frac{1}{2}} - \overline{\overline{u}}_i^{n}}{\|\tau\|}\right) \times \sqrt{(v)^2+1} = d\left(\frac{\overline{\overline{u}}_{i-1}^{n} - 2\overline{\overline{u}}_i^{n} + \overline{\overline{u}}_{i+1}^{n}}{(\Delta x)^2}\right). \tag{3}$$

For clarity, we use a notation with both over-lines (to indicate for example the foot of the characteristic line that is calculated by an interpolation method) and superscript (to indicate time semi-step), i.e., $\overline{\overline{u}}^{n}$ $\left(\overline{u}^{n+\frac{1}{2}}\right)$ represents u evaluated in the foot of the characteristic line calculated in the previous time (semi-) step, $\overline{\overline{u}}^{n-\frac{1}{2}}$ represents u evaluated in the foot of the characteristic line calculated in the first time semi-step from the previous time step, and u^{n+1} represents u evaluated in the second time semi-step of the BDFHM method (see Figure 1).

Substituting $\|\tau\| = \delta t \cdot \sqrt{(v)^2+1}$ in equation (3) yields $\overline{u}_i^{n+\frac{1}{2}} = \overline{\overline{u}}_i^{n} + \delta t \cdot d \cdot \left(\frac{\overline{\overline{u}}_{i-1}^{n} - 2\overline{\overline{u}}_i^{n} + \overline{\overline{u}}_{i+1}^{n}}{(\Delta x)^2}\right)$. Similarly, one can obtain a discretization of the Laplace operator on an implicit form by means of the equation $\overline{u}_i^{n+\frac{1}{2}} = \overline{\overline{u}}_i^{n} + \delta t \cdot d\left(\frac{\overline{\overline{u}}_{i-1}^{n} - 2\overline{\overline{u}}_i^{n} + \overline{\overline{u}}_{i+1}^{n}}{(\Delta x)^2}\right)$. Likewise, the difference operator for the BDFHM can be defined as

$$L_h \overline{\overline{u}}_i^{n} = d \cdot \left[\frac{\overline{\overline{u}}_{i-1}^{n} - 2\overline{\overline{u}}_i^{n} + \overline{\overline{u}}_{i+1}^{n}}{(\Delta x)^2}\right]. \tag{4}$$

When using operator (4), two time semi-steps of the BDFHM method can be represented as $\frac{3}{2}\overline{u}_i^{n+\frac{1}{2}} = 2\overline{\overline{u}}_i^n - \frac{1}{2}\overline{\overline{u}}_i^{n-\frac{1}{2}} + \delta t\left(\theta_i^n L_h \overline{\overline{u}}_i^n + \theta_i^{n+\frac{1}{2}} L_h \overline{u}_i^{n+\frac{1}{2}}\right)$ and $\frac{3}{2}u_i^{n+1} = 2\overline{u}_i^{n+\frac{1}{2}} - \frac{1}{2}\overline{\overline{u}}_i^n + \delta t\left(\theta_i^n L_h \overline{u}_i^{n+\frac{1}{2}} + \theta_i^{n+\frac{1}{2}} L_h u_i^{n+1}\right)$ where $\delta t = \frac{\Delta t}{2}$ and $\theta_i^n = 1$ (0) if $n+i$ is odd (even). A complete time step of the BDFHM method can be described as follows.

1. Initialize $\overline{\overline{\overline{x}}}$ and $\overline{\overline{x}}$ (e.g., using the Hopmoc method) at times steps $t_{-\frac{1}{2}}$ and t_0, respectively.
2. Obtain $\overline{u}$ for all N mesh nodes $(1 \leq i \leq N)$ $\overline{x}_i$ (e.g., using an interpolation method). In particular, the BDFHM method uses two interpolations in the first time semi-step, i.e., it uses an interpolation method to calculate $\overline{\overline{u}}^{n-\frac{1}{2}}$ and $\overline{\overline{u}}^n$, which are used to obtain $\overline{u}^{n+\frac{1}{2}}$. In the second time semi-step, no interpolation method is used since $\overline{\overline{u}}^n$ and $\overline{u}^{n+\frac{1}{2}}$ (that are used to obtain u^n) were calculated in the first time semi-step.
3. Calculate (alternately) $\overline{u}_i^{n+\frac{1}{2}}$ using the explicit (implicit) operator for mesh nodes $n+1+i$ that belong to the odd (even) subset.
4. Calculate (alternately) u_i^{n+1} using the implicit (explicit) operator for mesh nodes $n+2+i$ that belong to the odd (even) subset.

In steps 3 and 4, the explicit approach uses the values from adjacent mesh nodes updated in the previous time step. The implicit approach uses values from adjacent mesh nodes that were updated in the current time step using the explicit scheme. Therefore, no linear system is solved when applying the BDFHM method, as previously mentioned. The use of backward differentiation formulas did not improve the accuracy of the standard Hopmoc method, i.e., the BDFHM method has first-order accuracy in both space and time variables. Figure 2 shows a flowchart that describes a complete time step of the algorithm.

4 RESULTS AND ANALYSIS

Before conducting experiments with the modified Burgers' equation, Section 4.1 analyzes the error of the BDFHM method when applied to the 1-D advection-diffusion equation. Specifically, Section 4.1 compares the results achieved by the BDFHM method with the standard Hopmoc method. Section 4.2 compares the results delivered by the BDFHM method with ten existing high-order approaches for the solution of the modified Burgers' equation.

4.1 Results from the BDFHM and Hopmoc Methods

This section simulates a Gaussian pulse evolution implemented in the C programming language. Consider the 1-D advection-diffusion equation (2) with velocity $v = 1.0$ and diffusion coefficient $d = \frac{2}{Re}$, so that

$$u_t + u_x = \frac{2}{Re}u_{xx}, \text{ for } 0 \leq x \leq 1 \text{ and } 0 \leq t \leq T. \tag{5}$$

In a continuous domain, the exact solution of (5) is given by $u(x,t) = \frac{e^{-\frac{(x-x_0)^2}{2\phi(t)}}}{\sqrt{\phi(t)}}$ where x_0 is the initial location of the Gaussian pulse center, ϕ is a time-dependent Gaussian pulse amplitude, $\phi(t) = \phi_0\left[1 + \frac{4t}{Re\phi_0}\right]$, $Re = \frac{\rho \cdot v \cdot L}{\mu}$ is the Reynolds number, ρ is the density, v is the characteristic velocity, L is a characteristic linear dimension, and μ is the dynamic viscosity. Simulations were carried out for a Gaussian pulse with amplitude 0.04 and $x_0 = 0.2$, $u(x,t)$ is the exact value of the problem with initial condition $u(x,0)$, and boundary conditions $u(0,t)$ and $u(1,t)$, for $t \geq 0$.

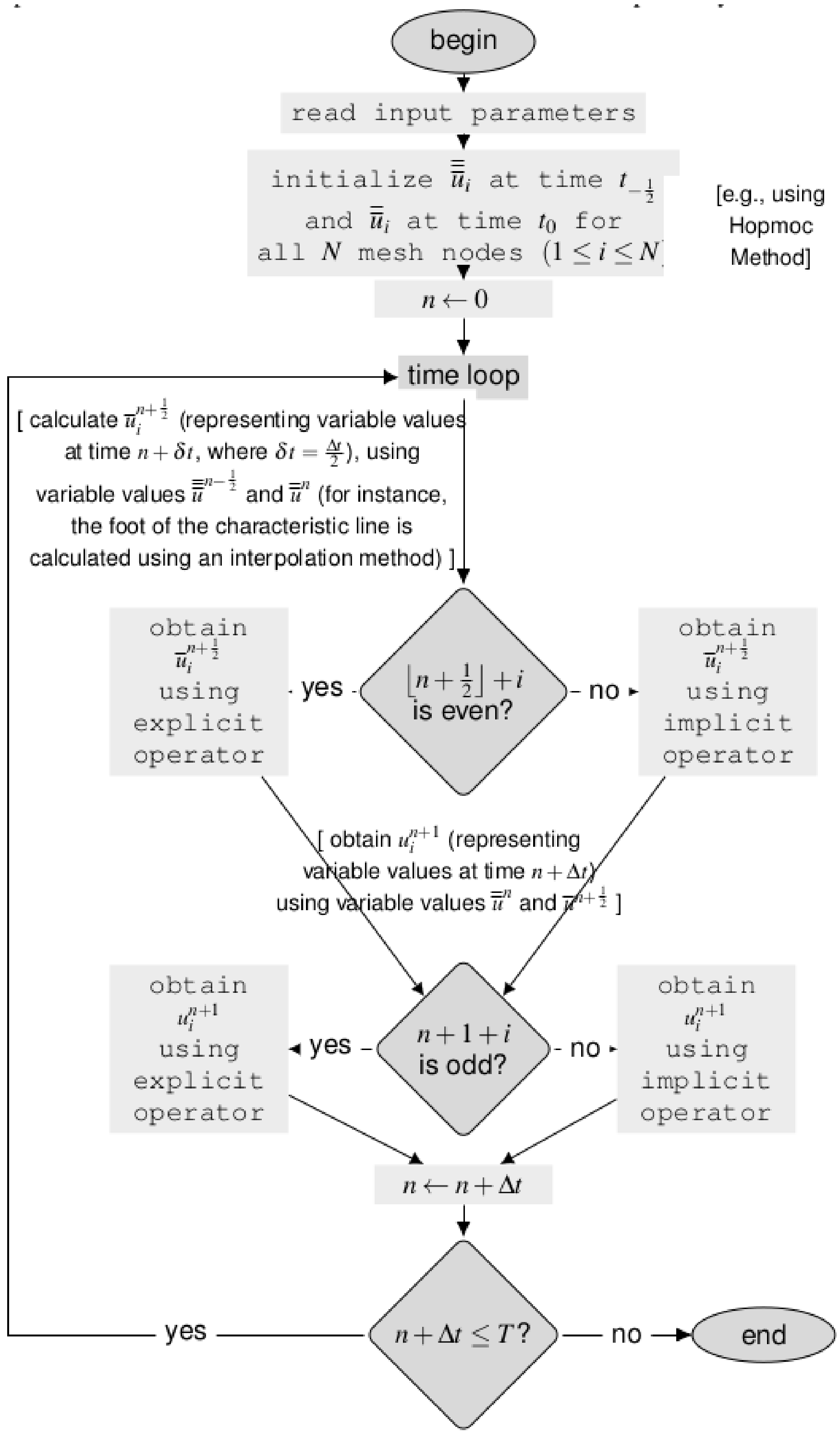

Figure 2: A complete time step of the BDFHM method. This flowchart shows comments between box brackets.

Table 1 shows the infinity error norm $\|\varepsilon\|_\infty$ when applying the BDFHM method to a Gaussian pulse setting $T = 0.5$, $Re = 500$, and using linear interpolation. Both the table and Figure 3 show that $\|\varepsilon\|_\infty$ decreases when using smaller $\delta t / \Delta x$.

Table 1: The infinity error norm $\|\varepsilon\|_\infty$ when applying the BDFHM method to solve a Gaussian pulse setting $T = 0.5$, $\phi = 0.04$, $v = 1$, $Re = 500$, and using a linear interpolation method.

δt	2.0e-3	5.0e-4	12.5e-5	312.5e-7	7812.5e-9
Δx	1.0e-2	5.0e-3	2.5e-3	12.5e-4	62.5e-5
$\delta t/\Delta x$	0.2	0.1	0.05	0.025	0.0125
$\|\varepsilon\|_\infty$	0.2536	0.1306	0.0664	0.0335	0.0168

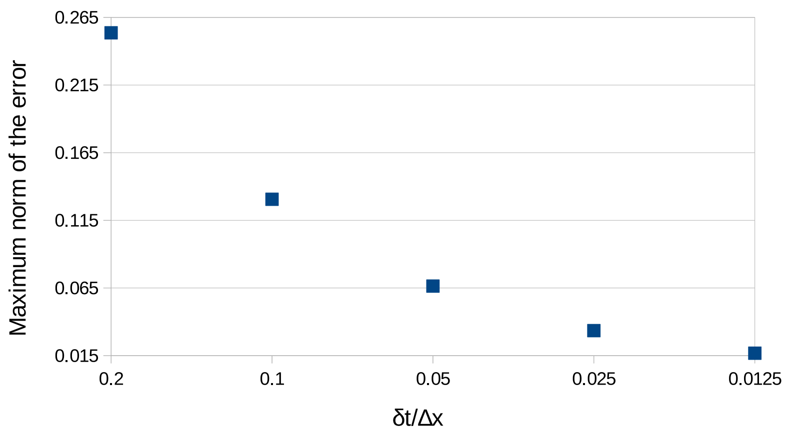

Figure 3: The infinity error norm $\|\varepsilon\|_\infty$ when employing the BDFHM method to solve a Gaussian pulse setting $T = 0.5$, $\phi = 0.04$, $v = 1$, $Re = 500$, and using a linear interpolation method.

Table 2 shows $\|\varepsilon\|_\infty$ when applying the BDFHM method to a Gaussian pulse using $d = 0.002$ in three Reynolds number regimes and $T = 0.5$ as the final time. As expected, the table shows that the strategy to approximate the values in the foot of the characteristic line impacts the infinity norm $\|\varepsilon\|_\infty$ in the BDFHM method. Specifically, the use of a cubic interpolation obtained the smallest $\|\varepsilon\|_\infty$ among the interpolation methods evaluated. Thus, Table 2 shows that $\|\varepsilon\|_\infty$ associated with the BDFHM method is smaller when using a higher order interpolation method and a smaller δt.

Table 2 and Figure 4 (in line chart for clarity) show that the BDFHM method using any linear, quadratic, or cubic interpolation is more accurate than the original Hopmoc method even when using a cubic interpolation method. In particular, the error of the original Hopmoc method using a linear interpolation method decreases when setting δt from 312.5e-6 to 62.5e-5. The potential reason behind this numerical diffusion is due precisely to the use of a linear interpolation method to calculate values from previous time steps in the foot of the characteristic line. On the other hand, the error of the original Hopmoc algorithm along with a cubic interpolation method slightly increases when using a larger δt (see Figure 4).

Figure 5 shows the resulting $\|\varepsilon\|_\infty$ when applying the BDFHM method to a Gaussian pulse setting $T = 0.5$ in three Reynolds number regimes (500, 1,000, and 5,000) and using three types of interpolation methods. Moreover, in accordance with the findings presented in the current literature (Long and Yuan 2009), Figure 5 suggests that $\|\varepsilon\|_\infty$ of the BDFHM method tends to $\mathscr{O}\left((\Delta t)^2\right)$.

As expected, the use of a cubic interpolation method reduced the maximum norm of the error in the numerical simulations comparing with the use of a linear interpolation method. In particular, if $v \to 0$, both BDFHM and Hopmoc methods have the same accuracy, recalling that one obtains the heat conduction equation with $v = 0$ in equation (2).

Table 2: Absolute error $\|\varepsilon\|_\infty$ when applying the original Hopmoc and BDFHM methods to a Gaussian pulse setting $T = 0.5$, $\phi = 0.04$, $\nu = 1$, the Reynolds (Re) number set to 500, 1,000, and 5,000, and using linear (l.), quadratic (q.), and cubic (c.) interpolations (int.).

δt	2.0e-4	2.5e-4	312.5e-6	4.0e-4	5.0e-4	62.5e-5	8.0e-4	int.	Re
$\delta t/\Delta x$	0.2	0.25	0.3125	0.4	0.5	0.625	0.8		
Hopmoc	1.3093	1.3919	1.4892	1.7947	1.8994	1.9801	2.0106	l.	500
BDFHM	0.0271	0.0272	0.0274	0.0277	0.0282	0.0289	0.0301	l.	
BDFHM	0.0015	0.0024	0.0028	0.0037	0.0044	0.0065	0.0078	c.	
Hopmoc	1.2615	1.2695	1.2704	1.2492	1.1920	1.0887	1.1090	l.	1,000
BDFHM	0.0290	0.0291	0.0292	0.0294	0.0297	0.0302	0.0323		
BDFHM	0.0118	0.0119	0.0120	0.0121	0.0123	0.0125	0.0191	q.	
Hopmoc	0.9791	0.9906	1.0036	1.0200	1.0341	1.0553	1.0645	c.	
BDFHM	0.0018	0.0022	0.0032	0.0053	0.0078	0.0103	0.0120		
BDFHM	0.6418	0.6493	0.6591	0.6732	0.6892	0.7072	0.7232	l.	5,000
BDFHM	0.0306	0.0307	0.0308	0.0310	0.0313	0.0318	0.0326	q.	
BDFHM	0.0027	0.0032	0.0037	0.0065	0.0098	0.0135	0.0156	c.	

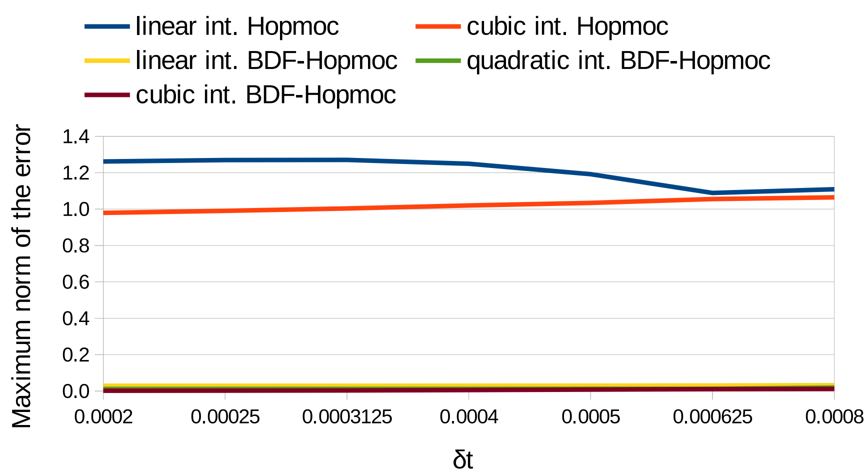

Figure 4: The infinity error norm $\|\varepsilon\|_\infty$ when applying the original Hopmoc and BDFHM methods to a Gaussian pulse with $T = 0.5$ and Reynolds number $Re = 1,000$.

4.2 Modified Burgers' Equation

This section shows the results yielded by 11 methods when applied to the modified Burgers' equation. To solve this equation, one needs to calculate u^2 at each time step. Thus, we used the BDFHM algorithm along with a predictor-corrector method. Algorithm 1 shows a full-time step to solve the problem.

Equation (1) has the analytic solution $u(x,t) = \frac{x}{t}\left[1+\frac{\sqrt{t}}{0.5}e^{\left(\frac{x^2}{4\cdot d\cdot t}\right)}\right]^{-1}$, $0 \le x \le 1$, $t \ge 1$ where the initial condition $u(x,1)$ is obtaind from the equation. Table 3 shows the results yielded by the BDFHM method along with a cubic interpolation method. The table also reproduces the results achieved by ten existing high-order methods:

- Ramadan and El-Danaf (2005) proposed a fifth-order spline-based method;

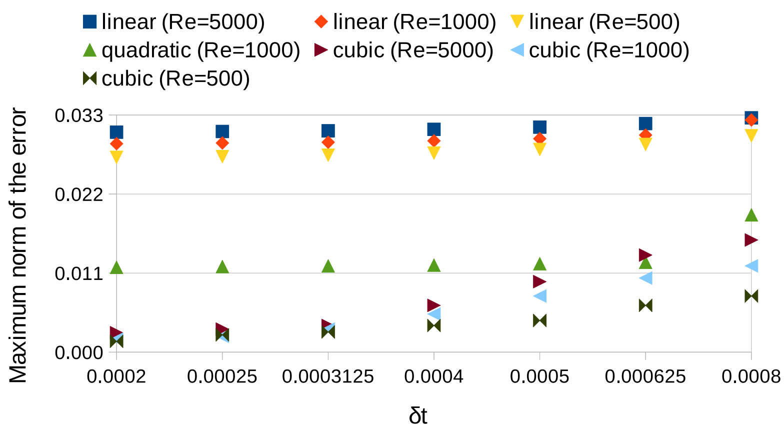

Figure 5: The infinity error norm $\|\varepsilon\|_\infty$ when applying the BDFHM method to a Gaussian pulse setting $T = 0.5$, the Reynolds (Re) number set as 500, 1,000, and 5,000, and using three types of interpolation methods.

Algorithm 1 The 1-D BDFHM method applied to the modified Burgers' equation.

Read entry parameters
Initialize $\overline{\overline{u}}_i$ at time step t_0 and $\overline{u}_i$ at time step $t_{\frac{1}{2}}$ for all stencil points N, i.e., $(1 \leq i \leq N)$
$n \leftarrow 1$
while $n \cdot \Delta t \leq T$ **do**
  Estimate the value $\left(\overline{u}_i^{n+\frac{1}{2}}\right)^2$ using a predictor-corrector method.
  if $\lfloor n + \frac{1}{2} \rfloor + i$ is even **then**
    Obtain $\overline{u}_i^{n+\frac{1}{2}}$ using an explict operator
  else
    Obtain $\overline{u}_i^{n+\frac{1}{2}}$ using an implict operator
  end if
      ▷ obtain u_i^{n+1} (representing the time variable $n \cdot \Delta t$) using the variables $\overline{\overline{u}}^n$ and $\overline{u}^{n+\frac{1}{2}}$
  Estimate the value $\left(\overline{u}_i^{n+1}\right)^2$ using a predictor-corrector method.
  if $n+1+i$ is odd **then**
    Obtain u_i^{n+1} using an explict operator
  else
    Obtain u_i^{n+1} using an implict operator
  end if
  $n \leftarrow n+1$
end while

- Saka and Dag (2008) introduced two fifth-order B-spline-based methods (QBCA1 and QBCA2 for short);
- Irk (2009) proposed a modified Crank-Nicolson method;
- Temsah (2009) introduced a method that has the third-order accuracy in time;
- Bratsos (2010) introduced a finite-difference approach based on forth-order rational approximations;
- Bratsos and Petrakis (2011) proposed an explicit finite-difference scheme based on second-order rational approximants to the matrix-exponential term;

- Roshan and Bhamra (2011) introduced a Petrov-Galerkin method using a linear hat function as the trial function and a cubic B-spline function as the test function;
- Rong-Pei et al. (2013) used the local discontinuous Galerkin (LDG) method;
- Zeytinoglu et al. (2018) (ZSA) proposed a method based on a finite-difference hybrid approximation that has the sixty-order accuracy in space variables.

Table 3: Error norms L_2 and L_∞ yielded by several methods applied to the modified Burgers' equation using $\delta x = 0.001$ and $\delta t = 10^{-5}$. The symbol "—" indicates that the authors did not present the results. The best results are in boldface.

T	2		4		6		10	
Approach	$L_\infty \cdot 10^{-3}$	$L_2 \cdot 10^{-3}$	$L_\infty \cdot 10^{-3}$	$L_2 \cdot 10^{-3}$	$L_\infty \cdot 10^{-3}$	$L_2 \cdot 10^{-3}$	$L_\infty \cdot 10^{-3}$	$L_2 \cdot 10^{-3}$
$d = 0.01$								
Temsah	**0.7580**	—	**0.5640**	—	**0.4590**	—	0.3000	—
Bratsos–Petrakis	0.8164	0.3832	0.6030	**0.3144**	0.4622	**0.2710**	0.2991	**0.1909**
Roshan–Bhamra	0.8177	**0.3755**	0.6081	0.3168	0.4675	0.2749	0.2307	0.1939
ZSA	0.8158	0.3794	0.6050	0.3171	0.4906	0.3224	**0.1234**	0.5338
Ramadan–El-Danaf	1.2170	0.5231	0.9314	0.5162	0.7225	0.4902	1.2812	0.6401
QBCA1	0.8168	0.3793	0.6054	0.3172	0.5258	0.3260	1.2813	0.5470
QBCA2	0.8221	0.3795	0.6117	0.3172	0.5258	0.3243	1.2813	0.5435
Irk	0.8150	0.4132	—	—	—	—	1.2813	0.5509
Bratsos	0.8167	0.3792	0.6056	0.3155	0.4650	0.2731	0.3018	0.1934
LDG	0.8160	0.3794	—	—	0.4650	0.2730	0.3020	0.1930
BDFHM	0.8169	*0.3794*	*0.6054*	*0.3154*	*0.4648*	*0.2730*	0.3017	*0.1932*
$d = 0.005$								
Bratsos–Petrakis	0.5804	0.2285	0.4285	**0.1878**	0.3287	**0.1634**	0.2276	**0.1344**
ZSA	**0.5791**	0.2265	**0.4294**	0.1882	0.3296	0.1646	0.2289	0.1398
QBCA1	0.5800	0.2265	**0.4294**	0.1882	0.3299	0.1646	0.2289	0.1396
QBCA2	0.5866	0.2270	0.4360	0.1883	**0.3265**	0.1643	0.2351	0.1379
Irk	0.5842	0.2340	—	—	—	—	**0.2263**	0.1387
Roshan–Bhamra	0.5811	**0.2233**	0.4321	0.1893	0.3326	0.1664	0.2307	0.1366
BDFHM	*0.5802*	*0.2265*	***0.4294***	*0.1882*	*0.3299*	*0.1646*	*0.2287*	*0.1352*
Ramadan–El-Danaf	0.7226	0.2579	0.5544	0.2528	0.4308	0.2257	0.3001	0.1873
Bratsos	0.5803	0.2265	0.4295	0.1882	0.3299	0.1646	0.2287	0.1352
$d = 0.001$								
Temsah	0.2730	—	**0.1570**	—	**0.1390**	—	**0.0936**	—
ZSA	**0.2583**	0.0682	0.1925	**0.0565**	0.1479	0.0494	0.1026	0.0407
QBCA1	0.2609	0.0681	0.1929	**0.0565**	0.1481	0.0494	0.1026	0.0407
QBCA2	0.2728	0.0695	0.2046	0.0570	0.1566	**0.0492**	0.1084	0.0400
Bratsos	0.2611	0.0682	0.1929	**0.0565**	0.1481	0.0494	0.1026	0.0407
BDFHM	*0.2611*	*0.0682*	0.1929	***0.0565***	*0.1481*	*0.0494*	*0.1026*	*0.0407*
Irk	0.2597	0.0722	—	—	—	—	0.0987	**0.0387**
Bratsos–Petrakis	0.2628	**0.0618**	0.1933	0.0566	0.1481	0.0493	0.1025	0.0405
Ramadan–El-Danaf	0.2797	0.0670	0.2186	0.0667	0.1718	0.0605	0.1213	0.0501
Roshan–Bhamra	0.2619	0.0661	0.1958	0.0574	0.1509	0.0506	0.1047	0.0416

Table 3 shows that the method proposed by Temsah (2009) yielded the best results in six out of eight cases analyzed. The same table shows that the approaches introduced by Bratsos and Petrakis (2011) and Zeytinoglu et al. (2018) obtained seven and five best results, respectively.

Even when compared to a large number of high-order methods, the BDFHM method obtained errors similar to the best results found in 21 out of 24 cases. These cases are in italic font in Table 3.

5 CONCLUSIONS

This paper shows numerical solutions to the 1-D modified Burgers' equation using a new algorithm that integrates backward differentiation formulas with the Hopmoc method. We referred to the new hybrid approach as the BDFHM method. The BDFHM method reached competitive results when compared with ten existing high-order methods.

We intend to intensively analyze the diffusion and dispersion errors of the BDFHM method in a future investigation. We also plan to evaluate how the new approach performs with discontinuities. We used the error norms L_2 and L_∞ to evaluate the BDFHM method. We intend to analyze the performance of the algorithm when using other norms. We also plan to analyze the computational costs of the method in a future study.

The BDFHM method is sensitive to the interpolation technique employed. Thus, we intend to study the BDFHM method in conjunction with total variation diminishing techniques (Harten 1983) and flux-limiting procedures to improve its accuracy results in future works. We also plan to investigate the behavior of the new numerical method when applied to the 2-D modified Burgers' equation.

REFERENCES

Bratsos, A. G. 2010. "A Fourth-order Numerical Dcheme for Dolving the Modified Burgers Equation". *Computers and Mathematics with Applications* 60(5):1393–1400.

Bratsos, A. G., and L. A. Petrakis. 2011. "An Explicit Numerical Dcheme for the Modified Burgers Equation". *International Journal for Numerical Methods in Biomedical Engneering* 27(2):232–237.

Cabral, F., C. Osthoff, G. Costa, D. N. Brandão, M. Kischinhevsky, and S. L. Gonzaga de Oliveira. 2017. "Tuning up TVD Hopmoc Method on Intel MIC Xeon Phi Architectures with Intel Parallel Studio Tools". In *Proceedings of the International Symposium on Computer Architecture and High Performance Computing Workshops (SBACPADW), October 17–20*, 19–24. Campinas, São Paulo.

Ding, H. F., and Y. X. Zhang. 2009. "A New Difference Scheme with High Accuracy and Absolute Stability for Solving Convection Diffusion Equations". *Journal of Computational and Applied Mathematics* 230(2):600–606.

Douglas, J. J., and T. F. Russell. 1955. "Numerical Methods for Convection-Dominated Diffusion Problems Based on Combining the Method of Characteristics with Finite Element or Finite Difference Procedures". *SIAM Journal on Numerical Analysis* 19(5):871–885.

Duan, Y., R. Liu, and Y. Jiang. 2008. "Lattice Boltzmann Model for the Modified Burgers Equation". *Applied Mathematics and Computation* 202(2):489–497.

Gourlay, P. 1970. "Hopscotch: a Fast Second Order Partial Differential Equation Solver". *IMA Journal of Applied Mathematics* 6(4):375–390.

Harten, A. 1983. "High Resolution Schemes for Hyperbolic Conservation Laws". *Journal of Computational Physics* 49(3):357–393.

Irk, D. 2009. "Sextic B–spline Collocation Method for the Modified Burgers' Equation". *Kybernetes* 38(9):1599–1620.

Long, X., and Y. Yuan. 2009. "Multistep Characteristic Method for Incompressible Flow in Porous Media". *Applied Mathematics and Computation* 214(1):259–270.

Oliveira, S., M. Kischinhevsky, and S. L. Gonzaga de Oliveira. 2009. "Convergence Analysis of the Hopmoc Method". *International Journal of Computer Mathematics* 86(8):1375–1393.

Ramadan, M. A., and T. S. El-Danaf. 2005. "Numerical Treatment for the Modified Burgers Equation". *Mathematics and Computers in Simulation* 70(2):90–98.

Ramadan, M. A., T. S. El-Danaf, and F. E. I. Abd-Alaal. 2005. "A Numerical Solution of the Burgers Equation using Septic B-splines". *Chaos, Solitons and Fractals* 26(3):795–804.

Ray, S. S., and A. K. Gupta. 2018. *Wavelet Methods for Solving Partial Differential Equations and Fractional*. Boca Raton: CRC Press.

Rong-Pei, Z., Y. Xi-Jun, and Z. Guo-Zhong. 2013. "Modified Burgers Equation by the Local Discontinuous Galerkin Method". *Chinese Physics B* 22(3):030210.

Roshan, T., and K. S. Bhamra. 2011. "Numerical Solutions of the Modified Burgers Equation by Petrov-Galerkin Method". *Applied Mathematics and Computation* 218(7):3673–3679.

Saka, B., and I. Dag. 2008. "A Numerical Study of the Burgers Equation". *Journal of the Franklin Institute* 345(4):328–348.

Temsah, R. S. 2009. "Numerical Solutions for Convection–Diffusion Equation Using El-Gendi Method". *Simulation* 14(3):760–769.

Zeytinoglu, A., M. Sari, and B. Allahverdiev. 2018. "Numerical Simulations of Shock Wave Propagatingby a Hybrid Approximation Based on High-Order Finite Difference Schemes". *Acta Physica Polonica A* 133(1):140–152.

AUTHOR BIOGRAPHIES

DIOGO T. ROBAINA is an Assistant Professor of Mathematics at the Escola Superior de Propaganda e Marketing, Brazil. He holds a DSc degree in Computer Science from the Universidade Federal Fluminense, Brazil. His areas of interest include convergence analysis of numerical methods. His email address is diogo.robaina@espm.br.

SANDERSON L. GONZAGA DE OLIVEIRA is an Assistant Professor of Computer Science at the Universidade Federal de Lavras, Brazil. He holds a DSc degree in Computer Science from the Universidade Federal Fluminense, Brazil. His research interests include heuristics for bandwidth and profile reductions, numerical methods, and parallel computing. His email address is sanderson@ufla.br.

MAURICIO KISCHINHEVSKY holds a DSc degree in Computer Science from Pontifical Catholic University of Rio de Janeiro, Brazil. He is an Associate Professor of Computer Science at Fluminense Federal University, Brazil. His areas of interest include numerical solution of differential equations, adaptive mesh refinement and semilagrangian methods for evolutionary equations, parallel computation, traffic science, computational science and engineering. His email address is kisch@ic.uff.br.

CARLA OSTHOFF is a researcher at National Laboratory for Scientific Computing (LNCC-Brazil). Her main research areas are: high performance computing, cluster computing, hybrid parallel computing, high performance scientific computing applications and parallel programming models. Prof. Osthoff received her BSc in Electronics Engineering from PUC/Rio de Janeiro, a MSc and a PhD in Computer Science from Universidade Federal do Rio de Janeiro, UFRJ, Brazil. She is currently the coordinator from LNCC High Performance Computing Center. Her email address is osthoff@lncc.br.

ALEXANDRE C. SENA is an Assistant Professor of Computer Science at the Universidade do Estado do Rio de Janeiro, Brazil. He holds a DSc degree in Computer Science from the Universidade Federal Fluminense, Brazil. His research interests include parallel computing and high performance computing. His email address is asena@ime.uerj.br.