A semi-Lagrangian Bernstein-Bézier finite element method for two-dimensional coupled Burgers' equations at high Reynolds numbers

Mofdi El-Amrani[a,*], Bassou Khouya[b], Mohammed Seaid[c]

[a]*Dpto. Matemáticas Applicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica, Universidad Rey Juan Carlos, 28933 Móstoles-Madrid, Spain*

[b]*International Water Research Institute, University Mohammed VI Polytechnic, Benguerir, Morocco*

[c]*Department of Engineering, University of Durham, South Road, DH1 3LE, UK*

Abstract

This paper aims to develop a semi-Lagrangian Bernstein-Bézier high-order finite element method for solving the two-dimensional nonlinear coupled Burgers' equations at high Reynolds numbers. The proposed method combines the semi-Lagrangian scheme for the time integration and the high-order Bernstein-Bézier functions for the space discretization in the finite element framework. Unstructured triangular Bernstein-Bézier patches are reconstructed in a simple and inherent manner over finite elements along the characteristic curves defined by the material derivative. A fourth-order Runge-Kutta scheme is used for the approximation of departure points along with a local L^2-projection to compute the solution at the semi-Lagrangian stage. By using these techniques, the nonlinear problem is decoupled and two linear diffusion problems are solved separately for each velocity component. An implicit time-steeping scheme is used and a preconditioned conjugate gradient solver is used for the resulting linear systems of algebraic equations. The proposed method is investigated through several numerical examples including convergence studies. It is found that the proposed method is stable, highly accurate and efficient in solving two-dimensional coupled Burgers' equations at high Reynolds numbers.

Keywords: Burgers' equations; High Reynolds numbers; Bernstein-Bézier finite elements; Semi-Lagrangian method; L^2-projection

1. Introduction

In the last decades, a huge effort has been made by the Computational Fluid Dynamics (CFD) community on the development of either commercial or open-source software due to its great importance in various industrial and environmental [1] purposes including aerodynamics [2, 3, 4], polymers [5], combustion [6] and whether prediction [1].

*Corresponding author

Email addresses: mofdi.elamrani@urjc.es (Mofdi El-Amrani), khouya.bassou@um6p.ma (Bassou Khouya), m.seaid@durham.ac.uk (Mohammed Seaid)

In general, mathematical models describing physical phenomena in these problems consists of the well-established Navier-Stokes equations. Hence, solving the coupled Burgers' system presents a crucial step in whatever CFD software. Computational techniques used to solve these problems can be classified into (i) Eulerian techniques, (ii) Lagrangian procedures and (iii) semi-Lagrangian methods. In the framework of finite elements, the most popular Eulerian methods are the streamline upwind Petrov-Galerkin methods [7, 8], Galerkin/least-squares methods [9, 8] and Taylor-Galerkin methods [10, 11]. Recently, much effort has been made on the development of efficient numerical schemes for solving coupled Burgers' equations [12, 13, 14, 15, 16]. For example, in [12] authors used both differential quadrature and finite difference methods for solving the coupled Burgers' system at high Reynolds numbers. A mesh-free method based on radial basis functions has been used in [13, 16] for solving the nonlinear coupled Burgers' equations. However, it is well known that these Eulerian methods do not perform very satisfactory when the Reynolds number reaches high values unless small time steps and highly refined grids are used in the simulations. In the case of high Reynolds numbers as those considered in this study, these requirements are practically not feasible and may limit the performance of these Eulerian methods. Fully Lagrangian techniques on the other hand, appear theoretically well suited for the numerical solution of advection problems due to the possibility of using large time steps in the simulations. In practice, the computational mesh for the Lagrangian methods moves along the fluid particle trajectories which may yield to mesh distortion after few time steps in the computations. Thus, because of this drawback, the Lagrangian methods are not recommended for the numerical solution of coupled nonlinear Burgers' problems. In the semi-Lagrangian method, known also in the framework of finite elements by Galerkin-characteristics method, to overcome the drawback of fully Lagrangian methods and keeping their advantage, the computational mesh is taken to be fixed while back-tracking the solution along the characteristic curves. The main advantage of the semi-Lagrangian method lies on the fact that the Courant-Friedrichs-Lewy (CFL) condition is highly relaxed compared to its Eulerian counterparts, see for example [17, 18, 19, 20, 21]. In addition, the Lagrangian treatment in the semi-Lagrangian method greatly reduces the time truncation errors in the Eulerian methods, see [22, 18, 23, 24] among others. Thus, the semi-Lagrangian finite element method has the potential to be more suitable than Eulerian and Lagrangian methods for coupled Burgers' system on unstructured meshes. It should be noted that the techniques studied in [17, 18, 19, 20, 21, 23, 24] are based on Lagrange basis functions for which only a second-order accuracy is achieved in their numerical approximations. The aim of the present work is to develop high-order basis functions for the finite element solution of the coupled Burgers' equations.

In general, most of semi-Lagrangian finite element methods are second-order accurate in space and time but the accuracy of this class of numerical methods depends on the order of the interpolation polynomials used to compute the solution in the convection stage and on the time integration procedure for the diffusion stage. For example, to achieve a second-order accuracy in the semi-Lagrangian finite element method, the interpolation polynomials have to be at least second-order accurate and the time integration must be at least semi-implicit for the diffusion terms. In addition, it has been observed that the error in the conventional semi-Lagrangian finite element method for convection-diffusion problems decreases as the time step increases at certain range of parameters, see for instance [20, 19]. High-order accurate numerical methods for convection-dominated problems have the potential to reduce the computational effort required for a given order of solution accuracy. The state of the art in this field is more advanced for the Eulerian methods than for the semi-Lagrangian methods. For example, high-order discretization techniques such as those relying on spectral or *hp*-finite element methods have been shown to achieve fast convergence with low numerical diffusion and dispersion errors for advection-diffusion problems, see for example [25, 26]. It was shown that the *hp*-finite element method delivers exponential convergence for elliptic problems with piecewise analytic data, see the survey [27] and further references are therein. A similar performance was also proven for boundary-layer and singularly perturbed problems in [28, 29]. A study reported in [30] reveals that the choice of high-order shape functions is critical to the stability and efficiency of the finite element discretization. Particularly, high-order finite elements based on Lobatto shape functions have proven to possess better conditioning compared to other types of high-order shape functions widely used in the literature, compare for instance [31]. Assessment of different high-order shape functions including Bernstein, Lobatto and Lagrange Gauss-Lobatto polynomials for interior acoustic problems reported in [32], has shown the advantage of high-order polynomials in reducing the pollution errors and the good performance of Bernstein polynomials when combined with the Krylov subspace solvers. In a closely related study [33], Bernstein shape functions have been demonstrated to yield comparable, and even better performance in terms of accuracy and memory requirements compared to the well-established partition of unity finite methods. The Bernstein polynomials are well known in the field of computer aided geometric design and computer graphics. However, their applications in the finite element community have until now not been widely adopted. Although hierarchical basis functions are often chosen in the design of high-order finite elements for their suitability in *p*-adaptivity. Recently, attention has been paid to the favorable properties of the Bernstein polynomials [34, 35]. Especially, it has been shown that the Bernstein-Bézier finite elements on simplicial domains, hexahedra and pyramids yield optimal complexity for the standard finite element spaces. In a more recent work [36], the Bernstein basis functions combined to an additive Schwarz preconditioner was successfully implemented for challenging applications including boundary layers, non-linear reaction-diffusion problems and wave propagation of solitons.

The main focus of the present study is the development of a class of high-order semi-Lagrangian finite element methods to numerically solve the two-dimensional coupled Burgers' equations at high Reynolds numbers. This goal is achieved by the implementation of Bernstein-Bézier finite elements for the semi-Lagrangian method. It should be stressed that combining the semi-Lagrangian method with the Bernstein-Bézier finite elements, to the best of our knowledge, is reported for the first time. In the context of semi-Lagrangian finite element methods, Bernstein polynomials are used as shape functions associated with elements of the computational mesh to calculate the departure points and update the global solutions. The positivity of these local basis functions and the variation diminishing properties make them a very attractive alternative to the standard Lagrange polynomials. Different from most of the studies on the coupled Burgers' equations which are usually devoted to improving the accuracy of computed solutions in the case of low Reynolds numbers, this study mainly focuses on keeping the stability of the solution at high Reynolds numbers, which is significant in practical applications and also challenging in numerical computations.

In addition, the main contribution in this work is using the Bernstein-Bézier functions to interpolate the solution from previous time step using a local L^2-projection procedure introduced in [19, 24]. The idea behind the L^2-projection approach consists of evaluating the solution at the departure points using a set of quadrature points distributed in the host element. Thus, we combine both main advantages of the semi-Lagrangian method related to the use of large time steps and the ability of Bernstein-Bézier finite elements in preserving high-order accuracy on relatively coarse meshes. A fully implicit time integration scheme is used for the viscous terms and the associated linear systems of algebraic equations are solved using an iterative algorithm. In this study, to increase the efficiency of the proposed method, we also implement a static condensation procedure within the preconditioned conjugate gradient solver. This procedure ensures the elimination of interior Degrees of Freedom (DoFs) at the elemental level leading to reduced linear systems to be solved using the preconditioned conjugate gradient solver proposed in [36]. The performance of the proposed semi-Lagrangian Bernstein-Bézier finite element method is demonstrated for several test examples of coupled Burgers' equations. The obtained results confirm the high performance of this procedure compared to the conventional ordering techniques without the static condensation. In addition, numerical results presented in the current work demonstrate that an interesting feature of the Bernstein-Bézier finite elements is to allow large time steps and coarse meshes in the simulations without deteriorating the high-order accuracy of the computed solutions.

This paper is organized as follows. In section 2 we present the two-dimensional coupled Burgers' equations and the calculation of departure points. Formulation of the semi-Lagrangian Bernstein-Bézier finite element method for solving the two-dimensional coupled Burgers' equations is formulated in section 3. This section includes the Bernstein-Bézier finite elements and an implicit time stepping scheme to solve the diffusion stage. Section 4 is devoted to numerical results for several examples of two-dimensional unsteady nonlinear coupled Burgers' equations. Our new approach is demonstrated to enjoy the expected efficiency as well as the accuracy. Concluding remarks are summarized in section 5.

2. Semi-Lagrangian method for time integration

In the present work, given a two-dimensional bounded domain $\Omega \subset \mathbb{R}^2$ with Lipschitz boundary $\partial\Omega$ and a time interval $[0, T]$, we are interested in solving the unsteady nonlinear coupled Burgers' equations reformulated in the dimensionless form as

$$\begin{aligned} \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= \frac{1}{Re}\Big(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\Big), \\ \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} &= \frac{1}{Re}\Big(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\Big), \end{aligned} \tag{1}$$

where $\boldsymbol{u}(t, \boldsymbol{x}) = (u(t, \boldsymbol{x}), v(t, \boldsymbol{x}))^\top$ is the velocity field at time t and position $\boldsymbol{x} = (x, y)^\top$, with u and v denote the velocity in x-direction and y-direction, respectively. In (1), Re is the Reynolds number controlling the relative importance of convection terms compared to diffusion terms. Note that the system (1) is solved subject to given initial and boundary conditions. We assume that appropriate boundary conditions are given in such a way the problem is well defined and has a unique solution. The main idea behind the semi-Lagrangian method is to reformulate the problem (1) using the total derivative as

$$\begin{aligned} \frac{Du}{Dt} &= \frac{1}{Re}\Big(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\Big), \\ \frac{Dv}{Dt} &= \frac{1}{Re}\Big(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\Big), \end{aligned} \tag{2}$$

where the material derivative $\frac{Dw}{Dt} = \frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y}$ measures the rate of change of a generic function w following the trajectories of the flow particles. The semi-Lagrangian method imposes a regular grid at the new time level and to backtracks the flow trajectories to the previous time level. At the old time level, the quantities that are needed are evaluated by interpolation from their known values on a regular grid. Hence, we divide the time interval $[0, T]$ into N subintervals $[t_{n-1}, t_n]$ with length $\Delta t = t_n - t_{n-1}$ for $n = 1, 2 \ldots, N$. To discretize in time with the Cauchy problem (2), we consider a second-order implicit scheme of Gear type also known in the literature by Backward Differentiation Formula (BDF2) [37]. Thus, integrating the equations (2) along the characteristic curves yields

$$\begin{aligned} \frac{3u(t_{n+1}, \mathcal{X}(t_{n+1}, \boldsymbol{x})) - 4u(t_n, \mathcal{X}(t_n, \boldsymbol{x})) + u(t_{n-1}, \mathcal{X}(t_{n-1}, \boldsymbol{x}))}{2\Delta t} &= \frac{1}{Re}\Big(\frac{\partial^2 u(t, \mathcal{X}(t_{n+1}, \boldsymbol{x}))}{\partial x^2} + \frac{\partial^2 u(t, \mathcal{X}(t_{n+1}, \boldsymbol{x}))}{\partial y^2}\Big), \\ \frac{3v(t_{n+1}, \mathcal{X}(t_{n+1}, \boldsymbol{x})) - 4v(t_n, \mathcal{X}(t_n, \boldsymbol{x})) + v(t_{n-1}, \mathcal{X}(t_{n-1}, \boldsymbol{x}))}{2\Delta t} &= \frac{1}{Re}\Big(\frac{\partial^2 v(t, \mathcal{X}(t_{n+1}, \boldsymbol{x}))}{\partial x^2} + \frac{\partial^2 v(t, \mathcal{X}(t_{n+1}, \boldsymbol{x}))}{\partial y^2}\Big), \end{aligned} \tag{3}$$

where the characteristic curves $\mathcal{X}(\tau, \boldsymbol{x})$ are solutions of the backward differential equations

$$\begin{aligned} \frac{d\mathcal{X}(\tau, \boldsymbol{x})}{d\tau} &= \boldsymbol{u}(\tau, \mathcal{X}(\tau, \boldsymbol{x})), \quad \forall\, \tau \in [t_n, t_{n+1}], \\ \mathcal{X}(t_{n+1}, \boldsymbol{x}) &= \boldsymbol{x}. \end{aligned} \tag{4}$$

Here, $\mathcal{X}(\tau, \boldsymbol{x}) = (X(\tau, \boldsymbol{x}), Y(\tau, \boldsymbol{x}))^\top$ is the departure point at time τ of a particle that will arrive at $\boldsymbol{x}$ at time t_{n+1}. Since the space discretization in this work is carried out using finite elements, it should be noted that the semi-Lagrangian method does not follow the flow particles forward in time as the Lagrangian methods do, instead it traces backwards the position at time t_n of particles that will reach the points of a fixed mesh at time t_{n+1}, see Figure 1 for an illustration. By so doing, the semi-Lagrangian method avoids the grid distortion difficulties that the conventional Lagrangian methods have. Here, the backward treatment in the semi-Lagrangian method does not require the particle to move in one direction since this later dynamic is mainly related to the flow velocity in the problem under study. Although the velocity field covers all directions, the semi-Lagrangian method calculates the characteristic curves for which the departure points $\mathcal{X}(\tau, \boldsymbol{x})$ are reaching the mesh point $\boldsymbol{x}$ at time t_{n+1}. It should also be stressed that accurate approximations of the characteristic curves $\mathcal{X}(\tau, \boldsymbol{x})$ are crucial to the overall accuracy of the semi-Lagrangian method. In [19, 24], an extrapolation procedure based on the mid-point rule is used to approximate the solution of (4), but this method is only second-order accurate and it involves an iterative procedure which may become computationally demanding. It is worth mentioning that the semi-Lagrangian method should be interpreted as a time stepping scheme for which the departure points are required to be accurately approximated by solving the backward differential equations (4). This time integration scheme should not be confused with the time integration method used for the diffusion terms as this later is mainly affected by the spatial discretization used for the diffusion operator. As the semi-Lagrangian method integrates the convection part in time using the total derivative $\frac{D}{Dt}$, the solution of the backward differential problem (4) should be highly accurate to guarantee the high-order accuracy of the overall semi-Lagrangian Bernstein-Bézier finite element method. In the current work, we consider the fourth-order explicit Runge-Kutta method to approximate the solution of the differential equations (4) as

$$\begin{aligned}
\mathbf{K}^{(1)} &= \Delta t\boldsymbol{u}\left(t_{n+1},\boldsymbol{x}\right),\\
\mathbf{K}^{(2)} &= \Delta t\boldsymbol{u}\left(t_{n+\frac{1}{2}},\boldsymbol{x}-\frac{1}{2}\mathbf{K}^{(1)}\right),\\
\mathbf{K}^{(3)} &= \Delta t\boldsymbol{u}\left(t_{n+\frac{1}{2}},\boldsymbol{x}-\frac{1}{2}\mathbf{K}^{(2)}\right),\\
\mathbf{K}^{(4)} &= \Delta t\boldsymbol{u}\left(t_{n},\boldsymbol{x}-\mathbf{K}^{(3)}\right),\\
\mathcal{X}(t_n,\boldsymbol{x}) &= \boldsymbol{x}-\frac{1}{6}\left(\mathbf{K}^{(1)}+2\mathbf{K}^{(2)}+2\mathbf{K}^{(3)}+\mathbf{K}^{(4)}\right).
\end{aligned} \tag{5}$$

In (5), fourth-order extrapolation formulas are used to approximate the velocity fields $\boldsymbol{u}(t_{n+1},\cdot)$ and $\boldsymbol{u}(t_{n+\frac{1}{2}},\cdot)$ for $n \geq 4$. Lower-order extrapolation formulas are used in the case of $n = 1, 2$ and 3. Notice that, in general the departure points $\mathcal{X}(t_n,\boldsymbol{x})$ do not coincide with the spatial position of the point $\boldsymbol{x}$. To find the host element where such point is located we adapt the search-locate algorithm proposed in [38] for unstructured finite elements. For the calculation of the departure points $\mathcal{X}(t_{n-1},\boldsymbol{x})$), we use the same procedure as (5). Note that to solve the backward differential equations (4), the stages in the Runge-Kutta method (5) have been reversed. To demonstrate this step, we consider for a small step Δs, the transformations $\tau = t_{n+1} + \frac{t_n - t_{n+1}}{\Delta s}s$ and $\mathcal{Y}(s,\boldsymbol{x}) = \mathcal{X}(\tau,\boldsymbol{x})$. Hence, solving differential equations (4) is equivalent to solve

$$\begin{aligned}
\frac{d\mathcal{Y}(s,\boldsymbol{x})}{ds} &= \frac{d\mathcal{X}(\tau,\boldsymbol{x})}{ds} = -\frac{\Delta t}{\Delta s}\frac{d\mathcal{X}(\tau,\boldsymbol{x})}{d\tau} = -\frac{\Delta t}{\Delta s}\mathbf{u}(\tau,\mathcal{X}(\tau,\boldsymbol{x})) = \boldsymbol{v}(s,\mathcal{Y}(s,\boldsymbol{x})), \quad \forall\ s\in[0,\Delta s],\\
\mathcal{Y}(0,\boldsymbol{x}) &= \boldsymbol{x},
\end{aligned} \tag{6}$$

where $\boldsymbol{v}\left(s,\mathcal{Y}(s,\boldsymbol{x})\right) = -\frac{\Delta t}{\Delta s}\mathbf{u}(\tau,\mathcal{X}(\tau,\boldsymbol{x}))$. Thus, the approximation of the initial problem (6) by the Runge-Kutta method is carried out as

$$\begin{aligned}
\mathbf{R}^{(1)} &= \Delta s\boldsymbol{v}\left(0,\boldsymbol{x}\right),\\
\mathbf{R}^{(2)} &= \Delta s\boldsymbol{v}\left(\frac{\Delta s}{2},\boldsymbol{x}+\frac{1}{2}\mathbf{R}^{(1)}\right),\\
\mathbf{R}^{(3)} &= \Delta s\boldsymbol{v}\left(\frac{\Delta s}{2},\boldsymbol{x}+\frac{1}{2}\mathbf{R}^{(2)}\right),\\
\mathbf{R}^{(4)} &= \Delta s\boldsymbol{v}\left(\Delta s,\boldsymbol{x}+\mathbf{R}^{(3)}\right),\\
\mathcal{Y}(\Delta s,\boldsymbol{x}) &= \boldsymbol{x}+\frac{1}{6}\left(\mathbf{R}^{(1)}+2\mathbf{R}^{(2)}+2\mathbf{R}^{(3)}+\mathbf{R}^{(4)}\right).
\end{aligned} \tag{7}$$

Next, we substitute $\mathcal{Y}(\Delta s, \boldsymbol{x})$ and $\boldsymbol{v}$ in (7) by its values $\mathcal{Y}(\Delta s, \boldsymbol{x}) = \mathcal{X}(t_n, \boldsymbol{x})$ and $\boldsymbol{v}(s, \mathcal{Y}(s, \boldsymbol{x})) = -\frac{\Delta t}{\Delta s}\mathbf{u}(\tau, \mathcal{X}(\tau, \boldsymbol{x}))$, respectively. This yields,

$$\begin{aligned}
\mathbf{R}^{(1)} &= \Delta s \boldsymbol{v}\left(0, \boldsymbol{x}\right) = -\Delta t \mathbf{u}\left(t_{n+1}, \boldsymbol{x}\right) = -\mathbf{K}^{(1)}, \\
\mathbf{R}^{(2)} &= \Delta s \boldsymbol{v}\left(\frac{\Delta s}{2}, \boldsymbol{x} + \frac{1}{2}\mathbf{R}^{(1)}\right) = -\Delta t \mathbf{u}\left(t_{n+\frac{1}{2}}, \boldsymbol{x} + \frac{1}{2}\mathbf{R}^{(1)}\right) = -\mathbf{K}^{(2)}, \\
\mathbf{R}^{(3)} &= \Delta s \boldsymbol{v}\left(\frac{\Delta s}{2}, \boldsymbol{x} + \frac{1}{2}\mathbf{R}^{(2)}\right) = -\Delta t \mathbf{u}\left(t_{n+\frac{1}{2}}, \boldsymbol{x} + \frac{1}{2}\mathbf{R}^{(2)}\right) = -\mathbf{K}^{(3)}, \\
\mathbf{R}^{(4)} &= \Delta s \boldsymbol{v}\left(\Delta s, \boldsymbol{x} + \mathbf{R}^{(3)}\right) = -\Delta t \mathbf{u}\left(t_n, \boldsymbol{x} + \mathbf{R}^{(3)}\right) = -\mathbf{K}^{(4)}, \\
\mathcal{X}(t_n, \boldsymbol{x}) &= \boldsymbol{x} + \frac{1}{6}\left(\mathbf{R}^{(1)} + 2\mathbf{R}^{(2)} + 2\mathbf{R}^{(3)} + \mathbf{R}^{(4)}\right) = \boldsymbol{x} - \frac{1}{6}\left(\mathbf{K}^{(1)} + 2\mathbf{K}^{(2)} + 2\mathbf{K}^{(3)} + \mathbf{K}^{(4)}\right),
\end{aligned} \tag{8}$$

which reduces to equations (5). Using the notation $w^n(\boldsymbol{x}) = w(t_n, \boldsymbol{x})$, $\widehat{w}^n(\boldsymbol{x}) = w(t_n, \mathcal{X}(t_n, \boldsymbol{x}))$ to denote the values of the generic function w at $(t_n, \boldsymbol{x})$ and $(t_n, \mathcal{X}(t_n, \boldsymbol{x}))$, respectively, the semi-discrete system (3) becomes

$$\begin{aligned}
u^{n+1}(\boldsymbol{x}) - \frac{2\Delta t}{3Re}\left(\frac{\partial^2 u^{n+1}(\boldsymbol{x})}{\partial x^2} + \frac{\partial^2 u^{n+1}(\boldsymbol{x})}{\partial y^2}\right) &= \frac{4}{3}\widehat{u}^n(\boldsymbol{x}) - \frac{1}{3}\widehat{u}^{n-1}(\boldsymbol{x}), \\
v^{n+1}(\boldsymbol{x}) - \frac{2\Delta t}{3Re}\left(\frac{\partial^2 v^{n+1}(\boldsymbol{x})}{\partial x^2} + \frac{\partial^2 v^{n+1}(\boldsymbol{x})}{\partial y^2}\right) &= \frac{4}{3}\widehat{v}^n(\boldsymbol{x}) - \frac{1}{3}\widehat{v}^{n-1}(\boldsymbol{x}).
\end{aligned} \tag{9}$$

It is clear that an advantage of the semi-Lagrangian method lies on the fact that the equations (9) are decoupled and can be solved separately for each solution component. Note that at time $t = 0$ only one initial condition is provided and to obtain the second condition we use the implicit Euler scheme. It should also be noted that, one could also use the fourth-order Runge-Kutta method (5) for solving the diffusion part in the considered problems however, this would not improve the accuracy of the overall method as the computed numerical solutions are expected to be more affected by the accuracy of the spatial discretization than its temporal counterparts. Indeed, at high Reynolds numbers, the diffusion part in the problem under study becomes negligible and it behaves as an hyperbolic system of conservation laws for which high-order spatial discretization is required for the accurate resolution of shocks. In these cases, a second-order time stepping scheme as the one considered in our study is enough to preserve the high-order accuracy of the scheme for shock capturing at high Reynolds numbers. It should also be stressed that using higher order time stepping schemes (such as the fourth-order Runge-Kutta method) for the viscous terms in our study would not change the obtained numerical results but it would increase the computational cost of the proposed method.

3. Finite elements for space discretization

To formulate our finite element semi-Lagrangian method for solving the Burgers' equations (1), we first discretize the spatial domain Ω into a quasi-uniform partition $\Omega_h \subset \Omega$ of non-overlapping triangular elements $\mathcal{K}_e$, $e = 1, \ldots, Ne$. The conforming finite element space for the solution that we use is defined as

$$\mathcal{W}_h^p = \left\{ w_h \in C^0(\bar{\Omega}) \bigcap \mathbb{H}_0^1(\Omega) : \quad w_h\Big|_{\mathcal{K}_e} \circ F_{\mathcal{K}_e} \in \mathbb{P}_p(\widehat{\mathcal{K}}), \quad \forall\, \mathcal{K}_e \in \Omega_h \right\}, \tag{10}$$

where $\mathbb{P}_p(\hat{\mathcal{K}})$ is the set of polynomials of degree $\leq p$ defined on the reference element $\hat{\mathcal{K}}$ and $F_{\mathcal{K}_e} : \hat{\mathcal{K}} \longrightarrow \mathcal{K}_e$ is an one-to-one mapping between physical and reference elements. Hence, we formulate the finite element solution $\boldsymbol{u}_h^{n+1}(\boldsymbol{x}) = \Big(u_h^{n+1}(\boldsymbol{x}), v_h^{n+1}(\boldsymbol{x})\Big)$ at time t_{n+1} as

$$u_h^{n+1}(\boldsymbol{x}) = \sum_{j=1}^{m} U_j^{n+1}\varphi_j(\boldsymbol{x}), \qquad v_h^{n+1}(\boldsymbol{x}) = \sum_{j=1}^{m} V_j^{n+1}\varphi_j(\boldsymbol{x}), \tag{11}$$

where $\{\varphi_j\}_{j=1}^m$ is the set of global nodal basis functions of $\mathcal{W}_h^p$ and m denotes the total number of DoFs associated to the finite element space $\mathcal{W}_h^p$. Next, assuming for simplicity purposes, the problem (9) is supplied with homogeneous Dirichlet boundary conditions on $\partial\Omega$, we replace $u(t_n, \boldsymbol{x})$ and $v(t_n, \boldsymbol{x})$ by their discrete forms u_h^n and v_h^n in (9), then we multiply the system by a test function $\varphi_i(\boldsymbol{x})$, $i = 1, \ldots, m$ and we integrate over the domain Ω_h, the following weak problem is obtained

$$\begin{aligned}
\sum_{j=1}^{m} U_j^{n+1}\Big(\int_{\Omega_h} \varphi_j(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_{\Omega_h} \nabla\varphi_j(\boldsymbol{x})\cdot\nabla\varphi_i(\boldsymbol{x})d\boldsymbol{x}\Big) &= \int_{\Omega_h}\Big(\frac{4}{3}\widehat{u}_h^n(\boldsymbol{x}) - \frac{1}{3}\widehat{u}_h^{n-1}(\boldsymbol{x})\Big)\varphi_i(\boldsymbol{x})d\boldsymbol{x},\\
\sum_{j=1}^{m} V_j^{n+1}\Big(\int_{\Omega_h} \varphi_j(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_{\Omega_h} \nabla\varphi_j(\boldsymbol{x})\cdot\nabla\varphi_i(\boldsymbol{x})d\boldsymbol{x}\Big) &= \int_{\Omega_h}\Big(\frac{4}{3}\widehat{v}_h^n(\boldsymbol{x}) - \frac{1}{3}\widehat{v}_h^{n-1}(\boldsymbol{x})\Big)\varphi_i(\boldsymbol{x})d\boldsymbol{x},
\end{aligned} \tag{12}$$

which can be rewritten in a compact finite element form as

$$\begin{aligned}
\Big(\boldsymbol{M} + \frac{2\Delta t}{3Re}\boldsymbol{S}\Big)\boldsymbol{U}^{n+1} &= \boldsymbol{b}_u,\\
\Big(\boldsymbol{M} + \frac{2\Delta t}{3Re}\boldsymbol{S}\Big)\boldsymbol{V}^{n+1} &= \boldsymbol{b}_v,
\end{aligned} \tag{13}$$

where $\boldsymbol{U}^{n+1}$ and $\boldsymbol{V}^{n+1}$ are m-valued vectors with entries U_j^{n+1} and V_j^{n+1}, respectively. The mass matrix $\boldsymbol{M}$ and stiffness matrix $\boldsymbol{S}$ are $m \times m$-valued matrices the entries of which are

$$M_{ij} = \int_{\Omega_h} \varphi_j(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \quad \text{and} \quad S_{ij} = \int_{\Omega_h} \nabla\varphi_j(\boldsymbol{x})\cdot\nabla\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i, j \leq m.$$

respectively. The right-hand sides $\boldsymbol{b}_u$ and $\boldsymbol{b}_v$ are m-valued vectors with entries

$$\begin{aligned}
b_i^u &= \int_{\Omega_h}\Big(\frac{4}{3}\widehat{u}_h^n(\boldsymbol{x}) - \frac{1}{3}\widehat{u}_h^{n-1}(\boldsymbol{x})\Big)\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i \leq m,\\
b_i^v &= \int_{\Omega_h}\Big(\frac{4}{3}\widehat{v}_h^n(\boldsymbol{x}) - \frac{1}{3}\widehat{v}_h^{n-1}(\boldsymbol{x})\Big)\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i \leq m.
\end{aligned} \tag{14}$$

In the present, a Bernstein-Bézier polynomial basis is adopted for the spatial discretization. It should also be noted that since the Bernstein polynomials are only interpolatory at the mesh grid vertices, a numerical procedure is needed to impose inhomogeneous Dirichlet type boundary conditions. In the present work, this is achieved by using the L^2-projection on the Bernstein-Bézier basis of the local boundary data Lagrange interpolate.

3.1. Bernstein-Bézier finite elements

Let us consider the non-overlapping triangulation of the computational domain Ω and let $\widehat{\mathcal{K}}$ be the master element defined by

$$\widehat{\mathcal{K}} = \Big\{\boldsymbol{\eta} = (\eta, \xi) : \quad 0 \leqslant \eta \leqslant 1, \quad 0 \leqslant \xi \leqslant 1 - \eta\Big\}. \tag{15}$$

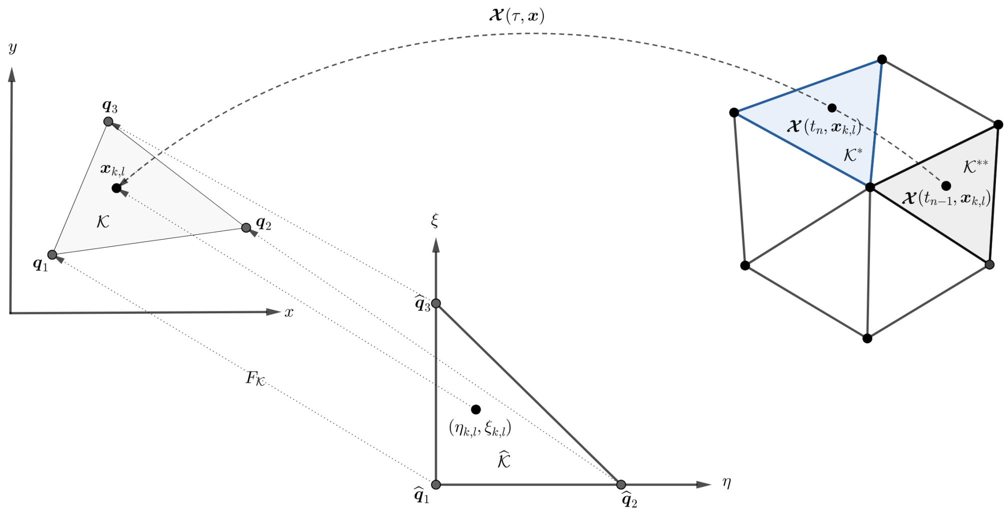
$\boldsymbol{\mathcal{X}}(\tau, \boldsymbol{x})$
y
$\boldsymbol{q}_3$
$\boldsymbol{x}_{k,l}$
$\mathcal{K}$
$\boldsymbol{q}_2$
$\boldsymbol{q}_1$
x
ξ
$\widehat{\boldsymbol{q}}_3$
$F_{\mathcal{K}}$
$(\eta_{k,l}, \xi_{k,l})$
$\widehat{\mathcal{K}}$
$\widehat{\boldsymbol{q}}_1$
$\widehat{\boldsymbol{q}}_2$
η
$\boldsymbol{\mathcal{X}}(t_n, \boldsymbol{x}_{k,l})$
$\mathcal{K}^*$
$\mathcal{K}^{**}$
$\boldsymbol{\mathcal{X}}(t_{n-1}, \boldsymbol{x}_{k,l})$

Figure 1: A schematic diagram showing the main quantities used in the approximation of the departure points. Here, $\mathcal{K}$ is a given mesh element and $(\eta_{k,l}, \xi_{k,l})$ is a Gauss integration point used in the reference element $\widehat{\mathcal{K}}$ and mapped onto $\boldsymbol{x}_{k,l}$ in the element $\mathcal{K}$ whereas, $\mathcal{K}^*$ and $\mathcal{K}^{**}$ are the host elements where the departure point $\mathcal{X}(t_n, \boldsymbol{x}_{k,l})$ and $\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})$ belong, respectively. Here, $F_{\mathcal{K}}$ is the affine one-to-one mapping.

We define the barycentric coordinates ζ_i $(i = 1, 2, 3)$ relative to the master element $\widehat{\mathcal{K}}$ by

$$\zeta_1(\boldsymbol{\eta}) = \eta, \qquad \zeta_2(\boldsymbol{\eta}) = \xi \qquad \text{and} \quad \zeta_3(\boldsymbol{\eta}) = 1 - \eta - \xi. \tag{16}$$

For a multi-index $\boldsymbol{\alpha} \in \mathbb{Z}^3_+$, we define $|\boldsymbol{\alpha}| = \sum_{i=1}^{3} \alpha_i$ and $\boldsymbol{\alpha}! = \prod_{i=1}^{3} \alpha_i!$. Let $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}^3_+$ such that $\boldsymbol{\beta} \geqslant \boldsymbol{\alpha}$ *i.e.*, $\beta_i \geqslant \alpha_i$ for $1 \leqslant i \leqslant 3$, we set $\begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\alpha} \end{pmatrix} = \prod_{i=1}^{3} \begin{pmatrix} \beta_i \\ \alpha_i \end{pmatrix}$ where $\begin{pmatrix} \beta_i \\ \alpha_i \end{pmatrix} = \dfrac{\beta_i!}{\alpha_i!(\beta_i - \alpha_i)!}$. Using these notations, the Bernstein-Bézier shape functions are simply written as

$$B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) = p! \frac{\zeta_1^{\alpha_1}(\boldsymbol{\eta})}{\alpha_1!} \frac{\zeta_2^{\alpha_2}(\boldsymbol{\eta})}{\alpha_2!} \frac{\zeta_3^{\alpha_3}(\boldsymbol{\eta})}{\alpha_3!},$$

where $|\boldsymbol{\alpha}| = p$. Hence, the Bernstein-Bézier basis for the space $\mathbb{P}_p(\widehat{\mathcal{K}})$ of polynomials of total degree at most p consists of the following shape functions:

- Vertex-based shape functions defined by

$$B^p_{(p,0,0)}(\boldsymbol{\eta}) = \zeta_1^p(\boldsymbol{\eta}), \qquad B^p_{(0,p,0)}(\boldsymbol{\eta}) = \zeta_2^p(\boldsymbol{\eta}), \qquad B^p_{(0,0,p)}(\boldsymbol{\eta}) = \zeta_3^p(\boldsymbol{\eta}).$$

- Edge-based shape functions defined by

$$B^p_{(p-k,k,0)}(\boldsymbol{\eta}) = \begin{pmatrix} p \\ k \end{pmatrix} \zeta_1^{p-k}(\boldsymbol{\eta})\zeta_2^k(\boldsymbol{\eta}), \qquad 1 \leqslant k \leqslant p-1,$$

$$B^p_{(0,p-k,k)}(\boldsymbol{\eta}) = \begin{pmatrix} p \\ k \end{pmatrix} \zeta_2^{p-k}(\boldsymbol{\eta})\zeta_3^k(\boldsymbol{\eta}), \qquad 1 \leqslant k \leqslant p-1,$$

$$B^p_{(k,0,p-k)}(\boldsymbol{\eta}) = \begin{pmatrix} p \\ k \end{pmatrix} \zeta_3^{p-k}(\boldsymbol{\eta})\zeta_1^k(\boldsymbol{\eta}), \qquad 1 \leqslant k \leqslant p-1.$$

- Cell-based shape functions defined by

$$B^p_{(i,j,p-i-j)}(\boldsymbol{\eta}) = \begin{pmatrix} p \\ i+j \end{pmatrix}\begin{pmatrix} i+j \\ i \end{pmatrix} \zeta_1^i(\boldsymbol{\eta})\zeta_2^j(\boldsymbol{\eta})\zeta_3^{p-i-j}(\boldsymbol{\eta}), \qquad 1 \leqslant i \leqslant p-2, \qquad 1 \leqslant i+j \leqslant p-1.$$

It should also be stressed that one of the most important proprieties of the Bernstein polynomials lies on the fact that their product yields a scaled Bernstein polynomial as

$$B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})B^q_{\boldsymbol{\beta}}(\boldsymbol{\eta}) = \frac{\begin{pmatrix} \boldsymbol{\alpha}+\boldsymbol{\beta} \\ \boldsymbol{\alpha} \end{pmatrix}}{\begin{pmatrix} p+q \\ p \end{pmatrix}} B^{p+q}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{\eta}), \tag{17}$$

where $|\boldsymbol{\alpha}| = p$ and $|\boldsymbol{\beta}| = q$. Furthermore the integral of a Bernstein polynomial over the reference triangle element has a simple form:

$$\int_{\widehat{\mathcal{K}}} B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})d\boldsymbol{\eta} = \frac{|\widehat{\mathcal{K}}|}{\begin{pmatrix} p+2 \\ 2 \end{pmatrix}}. \tag{18}$$

On the other hand the gradient of Bernstein polynomials can be computed as follows:

$$\nabla B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) = p\sum_{k=1}^{3} B^{p-1}_{\boldsymbol{\alpha}-\boldsymbol{e}_k}(\boldsymbol{\eta})\nabla\zeta_k, \tag{19}$$

where $\boldsymbol{e}_k$ is a multi-index with its kth entry is a unity and its remaining entries are zero, and $B^{p-1}_{\boldsymbol{\alpha}-\boldsymbol{e}_k} = 0$ if $\boldsymbol{\alpha} - \tilde{\boldsymbol{e}}_k$ has a negative component. It should be noted that the Bernstein polynomials are non-negative and form a partition of unity on the element $\widehat{\mathcal{K}}$. Moreover, these polynomials have some attractive features such as variation diminishing and monotonicity preserving properties, see for instance [39, 40] and further references can be found therein.

For a polynomial degree $p \geqslant 1$, the number of DoFs per element is given by

$$N^e = \begin{cases} 3, & \text{if} \quad p = 1, \\ 6, & \text{if} \quad p = 2, \\ 3 + 3(p-1) + \dfrac{(p-2)(p-1)}{2}, & \text{if} \quad p \geqslant 3. \end{cases} \tag{20}$$

Note that, in contrast to the Lagrange finite elements where the degrees of freedom refer to nodal point evaluations, a global orientation of edges is required to enable matching edge modes of a similar shape, thus ensuring C^0 conformity [33, 26, 31]. Finite element approximations with varying polynomial order can also be used by assigning to each vertex, edge and cell in the mesh an arbitrary polynomial degree, see [34, 33] among others.

3.2. Assembly of finite element matrices

In this section, we describe the procedure used to assemble the mass and stiffness matrices along with the right-hand side vectors in the finite element discretization (13). Let us denote by $\mathcal{I}_i$ the set of indices of all elements defining the support of the basis function φ_i. Hence, the span $\{\varphi_i\}_{i=1,\dots,m}$ of $\mathcal{W}_h^p$ is defined element-wise as

$$\varphi_i(\boldsymbol{x}) = \begin{cases} B^p_{\boldsymbol{\alpha}_e}(\boldsymbol{\eta}_e), & \text{if } \boldsymbol{x} \in \mathcal{K}_e, \quad \forall e \in \mathcal{I}_i, \\ 0, & \text{elsewhere,} \end{cases}$$

where $\boldsymbol{\eta}_e = F^{-1}_{\mathcal{K}_e}(\boldsymbol{x})$ and $|\boldsymbol{\alpha_e}| = p$. Using this notation the entries M_{ij} and S_{ij} of the mass and stiffness matrices become

$$\begin{aligned} M_{ij} &= \sum_{e \in \mathcal{I}_i \bigcap \mathcal{I}_j} \int_{\mathcal{K}_e} B^p_{\boldsymbol{\alpha}_e}(\boldsymbol{\eta}_e) B^p_{\boldsymbol{\beta}_e}(\boldsymbol{\eta}_e) d\boldsymbol{x} = \sum_{e \in \mathcal{I}_i \bigcap \mathcal{I}_j} M^{\mathcal{K}_e}_{\boldsymbol{\alpha}_e, \boldsymbol{\beta}_e}, \\ S_{ij} &= \sum_{e \in \mathcal{I}_i \bigcap \mathcal{I}_j} \int_{\mathcal{K}_e} \nabla B^p_{\boldsymbol{\alpha}_e}(\boldsymbol{\eta}_e) \cdot \nabla B^p_{\boldsymbol{\beta}_e}(\boldsymbol{\eta}_e) d\boldsymbol{x} = \sum_{e \in \mathcal{I}_i \bigcap \mathcal{I}_j} S^{\mathcal{K}_e}_{\boldsymbol{\alpha}_e, \boldsymbol{\beta}_e}. \end{aligned} \tag{21}$$

The right-hand side terms in (14) becomes

$$\begin{aligned} b^u_i &= \sum_{e \in \mathcal{I}_i} \int_{\mathcal{K}_e} \left(\frac{4}{3} \widehat{u}^n_h(\boldsymbol{x}) - \frac{1}{3} \widehat{u}^{n-1}_h(\boldsymbol{x}) \right) B^p_{\boldsymbol{\alpha}_e}(\boldsymbol{\eta}_e) \, d\boldsymbol{x} = \sum_{e \in \mathcal{I}_i} b^{u,\mathcal{K}_e}_{\boldsymbol{\alpha}_e}, \\ b^v_i &= \sum_{e \in \mathcal{I}_i} \int_{\mathcal{K}_e} \left(\frac{4}{3} \widehat{v}^n_h(\boldsymbol{x}) - \frac{1}{3} \widehat{v}^{n-1}_h(\boldsymbol{x}) \right) B^p_{\boldsymbol{\alpha}_e}(\boldsymbol{\eta}_e) \, d\boldsymbol{x} = \sum_{e \in \mathcal{I}_i} b^{u,\mathcal{K}_e}_{\boldsymbol{\alpha}_e}. \end{aligned} \tag{22}$$

Next, let us consider a generic element $\mathcal{K} \in \{\mathcal{K}_e, e \in \mathcal{I}_i \bigcap \mathcal{I}_j\}$, a generic $\boldsymbol{\alpha} \in \{\boldsymbol{\alpha}_e, e \in \mathcal{I}_i \bigcap \mathcal{I}_j\}$ and a generic $\boldsymbol{\beta} \in \{\boldsymbol{\beta}_e, e \in \mathcal{I}_i \bigcap \mathcal{I}_j\}$. By setting $\boldsymbol{\eta} = F^{-1}_{\mathcal{K}}(\boldsymbol{x})$, the evaluation of the entries of finite element matrices M_{ij} and S_{ij} is reduced to the computation of the entries of the element matrices $M_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ and $S_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ as

$$\begin{aligned} M_{\boldsymbol{\alpha},\boldsymbol{\beta}} &= \int_{\mathcal{K}} B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) B^p_{\boldsymbol{\beta}}(\boldsymbol{\eta}) d\boldsymbol{x} = \int_{\widehat{\mathcal{K}}} \det(\boldsymbol{J}_{\mathcal{K}}) B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) B^p_{\boldsymbol{\beta}}(\boldsymbol{\eta}) \, \mathrm{d}\boldsymbol{\eta}, \\ S_{\boldsymbol{\alpha},\boldsymbol{\beta}} &= \int_{\mathcal{K}} \nabla B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) \cdot \nabla B^p_{\boldsymbol{\beta}_e}(\boldsymbol{\eta}) d\boldsymbol{x} = \int_{\widehat{\mathcal{K}}} \det(\boldsymbol{J}_{\mathcal{K}}) \, \left(\boldsymbol{J}^{-1}_{\mathcal{K}} \widehat{\nabla} B^p_{\boldsymbol{\beta}} \right) \cdot \left(\boldsymbol{J}^{-1}_{\mathcal{K}} \widehat{\nabla} B^p_{\boldsymbol{\alpha}} \right) \, \mathrm{d}\boldsymbol{\eta}, \end{aligned} \tag{23}$$

where $\boldsymbol{J}_{\mathcal{K}} = \left(\frac{\mathrm{d}F_{\mathcal{K}}}{\mathrm{d}\boldsymbol{\eta}} \right)^\top$ is the Jacobian matrix, the operator ∇ and $\widehat{\nabla}$ are defined as $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^\top$ and $\widehat{\nabla} = (\frac{\partial}{\partial \eta}, \frac{\partial}{\partial \xi})^\top$, respectively. Since the geometry is interpolated using the affine map $F_{\mathcal{K}}$, the Jacobian matrix $\boldsymbol{J}_{\mathcal{K}}$ has constant entries and $\det(J_{\mathcal{K}}) = \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}$ and therefore analytical integration rules as those proposed in [34, 35] can be used for the evaluation of both entries of element stiffness and mass matrices. Thus, using the proprieties (17) and (18), the integral over the element $\mathcal{K}$ in (23) can be evaluated as

$$M_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad = \quad \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\frac{\binom{\boldsymbol{\alpha}+\boldsymbol{\beta}}{\boldsymbol{\beta}}}{\binom{2p}{p}}\int_{\widehat{\mathcal{K}}} B^{2p}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{\eta})\,\mathrm{d}\boldsymbol{\eta} = \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\frac{\binom{\boldsymbol{\alpha}+\boldsymbol{\beta}}{\boldsymbol{\beta}}}{\binom{2p}{p}\binom{p+2}{2}}. \tag{24}$$

Similarly, using the property (19), the entries of the element stiffness matrix can be computed as

$$S_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \sum_{k=1}^{3}\sum_{l=1}^{3}\left(\boldsymbol{J}_{\mathcal{K}}^{-1}\widehat{\nabla}\zeta_k\right)\cdot\left(\boldsymbol{J}_{\mathcal{K}}^{-1}\widehat{\nabla}\zeta_l\right)M_{\boldsymbol{\alpha}-\boldsymbol{e}_k,\boldsymbol{\beta}-\boldsymbol{e}_l}. \tag{25}$$

Hence, using the same notation as above, the evaluation of the right-hand entries in (14) is reduced to the computation of the entries $b^u_{\boldsymbol{\alpha}}$ and $b^v_{\boldsymbol{\alpha}}$

$$\begin{aligned} b^u_{\boldsymbol{\alpha}} &= \int_{\mathcal{K}}\Big(\frac{4}{3}\widehat{u}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{u}^{n-1}_h(\boldsymbol{x})\Big)B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})d\boldsymbol{x} = \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\int_{\widehat{\mathcal{K}}}\Big(\frac{4}{3}\widehat{u}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{u}^{n-1}_h(\boldsymbol{x})\Big)B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})d\boldsymbol{\eta},\\ b^v_{\boldsymbol{\alpha}} &= \int_{\mathcal{K}}\Big(\frac{4}{3}\widehat{v}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{v}^{n-1}_h(\boldsymbol{x})\Big)B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})d\boldsymbol{x} = \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\int_{\widehat{\mathcal{K}}}\Big(\frac{4}{3}\widehat{v}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{v}^{n-1}_h(\boldsymbol{x})\Big)B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta})d\boldsymbol{\eta}. \end{aligned} \tag{26}$$

It is evident that, if the above integrals are evaluated exactly then it is easy to show that the semi-Lagrangian method is unconditionally stable in the L^2-norm, see for instance [18]. In a general framework, this cannot be done and one has to approximate the integrals by numerical integration. In the current study, we consider the procedure proposed in [34] to compute the Bernstein-Bézier moments. Note that it has shown in [34] that the Duffy transformation enables a tensorial reconstruction of the Bernstein-Bézier basis on simplices. Thus, the well-established sum factorization is used to efficiently evaluate and integrate these polynomials based on the Stroud conical quadrature. This transformation maps the unit quadrilateral with coordinates $\boldsymbol{r} = (r, s) \in \widehat{Q} = [0, 1]^2$ to the reference triangle $\widehat{\mathcal{K}}$ and it can be defined by

$$\eta = r, \qquad \xi = (1-r)s. \tag{27}$$

Let us denote by $B^p_i(s) = \binom{p}{i}s^i(1-s)^{p-i}$ the one-dimensional Bernstein polynomial on the unit interval $[0, 1]$. Hence, it is shown in [34] that

$$B^p_{\boldsymbol{\alpha}}(\boldsymbol{\eta}) = B^p_{\alpha_1}(r)B^{p-\alpha_1}_{\alpha_2}(s). \tag{28}$$

Thus, the integrals in (26) become

$$\begin{aligned} b^u_{\boldsymbol{\alpha}} &= \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\int_0^1\left(\int_0^1\left(\frac{4}{3}\widehat{u}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{u}^{n-1}_h(\boldsymbol{x}))\right)(1-r)B^p_{\alpha_1}(r)dr\right)B^{p-\alpha_1}_{\alpha_2}(s)ds,\\ b^v_{\boldsymbol{\alpha}} &= \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\int_0^1\left(\int_0^1\left(\frac{4}{3}\widehat{v}^n_h(\boldsymbol{x})-\frac{1}{3}\widehat{v}^{n-1}_h(\boldsymbol{x})\right)(1-r)B^p_{\alpha_1}(r)dr\right)B^{p-\alpha_1}_{\alpha_2}(s)ds. \end{aligned} \tag{29}$$

Recall the q-point Gauss-Jacobi quadrature defined as

$$\int_0^1 w(z)f(z)\,\mathrm{d}z \approx \sum_{g=0}^{q} w^{(d)}_g f(z^{(d)}_g), \tag{30}$$

where the weight function $w(z) = (1-z)^d$, with $d > -1$, $\{z_g^{(d)}\}$ is the set of abscissa, and $\{w_g^{(d)}\}$ are the associated weights. In our case, we set $d = 1$ for the variable r and $d = 0$ for the variable s. Hence, the quantities needed are $\{z_g^{(0)}\}$, $\{w_g^{(0)}\}$, $\{z_g^{(1)}\}$ and $\{z_g^{(1)}\}$, and the evaluation of the Bernstein-Bézier moments (29) yields

$$\begin{aligned} b_\alpha^u &\approx \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\sum_{k=1}^{q} w_k^{(1)} B_{\alpha_2}^{p-\alpha_1}\left(z_k^{(0)}\right)\left(\sum_{l=1}^{q} w_l^{(1)}\; B_{\alpha_1}^{p}(z_l^{(1)})\Big(\frac{4}{3}\widehat{u}_h^n(\boldsymbol{x}_{k,l}) - \frac{1}{3}\widehat{u}_h^{n-1}(\boldsymbol{x}_{k,l})\Big)\right), \\ b_\alpha^v &\approx \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\sum_{k=1}^{q} w_k^{(1)} B_{\alpha_2}^{p-\alpha_1}\left(z_k^{(0)}\right)\left(\sum_{l=1}^{q} w_l^{(1)}\; B_{\alpha_1}^{p}(z_l^{(1)})\Big(\frac{4}{3}\widehat{v}_h^n(\boldsymbol{x}_{k,l}) - \frac{1}{3}\widehat{v}_h^{n-1}(\boldsymbol{x}_{k,l})\Big)\right), \end{aligned} \tag{31}$$

where $\boldsymbol{x}_{k,l} = F_{\mathcal{K}}(\boldsymbol{\eta}_{k,l})$ and $\boldsymbol{\eta}_{k,l} = (z_k^{(1)}, (1-z_k^{(1)})z_l^{(0)})$. By setting $\widehat{u}_{k,l}^n = \widehat{u}_h^n(\boldsymbol{x}_{k,l}), \widehat{u}_{k,l}^{n-1} = \widehat{u}_h^{n-1}(\boldsymbol{x}_{k,l}), \widehat{v}_{k,l}^n = \widehat{v}_h^n(\boldsymbol{x}_{k,l})$ and $\widehat{v}_{k,l}^{n-1} = \widehat{v}_h^{n-1}(\boldsymbol{x}_{k,l})$, the integrals in (31) are rewritten as

$$\begin{aligned} b_\alpha^u &\approx \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\sum_{k=1}^{q} w_k^{(1)} B_{\alpha_2}^{p-\alpha_1}\left(z_k^{(0)}\right)\left(\sum_{l=1}^{q} w_l^{(1)}\; B_{\alpha_1}^{p}(z_l^{(1)})(\frac{4}{3}\widehat{u}_{k,l}^n - \frac{1}{3}\widehat{u}_{k,l}^{n-1})\right), \\ b_\alpha^v &\approx \frac{|\mathcal{K}|}{|\widehat{\mathcal{K}}|}\sum_{k=1}^{q} w_k^{(1)} B_{\alpha_2}^{p-\alpha_1}\left(z_k^{(0)}\right)\left(\sum_{l=1}^{q} w_l^{(1)}\; B_{\alpha_1}^{p}(z_l^{(1)})(\frac{4}{3}\widehat{v}_{k,l}^n - \frac{1}{3}\widehat{v}_{k,l}^{n-1})\right). \end{aligned} \tag{32}$$

Note that the crucial step in this approach is the evaluation of the coefficients $\widehat{u}_{k,l}^n, \widehat{u}_{k,l}^{n-1}, \widehat{v}_{k,l}^n$ and $\widehat{v}_{k,l}^{n-1}$. Let us denote by $\mathcal{K}^*$ and $\mathcal{K}^{**}$ the elements where the departure points $\mathcal{X}(t_n, \boldsymbol{x}_{k,l})$ and $\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})$ belong, respectively. Hence, the coefficients in (32) are computed as

$$\begin{aligned} \widehat{u}_{k,l}^n &= u_h^n\Big(\mathcal{X}(t_n, \boldsymbol{x}_{k,l})\Big) = \sum_{|\alpha|=p} U_\alpha^n B_\alpha^p(\boldsymbol{\eta}^*), \\ \widehat{u}_{k,l}^{n-1} &= u_h^{n-1}\Big(\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})\Big) = \sum_{|\alpha|=p} U_\alpha^{n-1} B_\alpha^p(\boldsymbol{\eta}^{**}), \\ \widehat{v}_{k,l}^n &= v_h^n\Big(\mathcal{X}(t_n, \boldsymbol{x}_{k,l})\Big) = \sum_{|\alpha|=p} V_\alpha^n B_\alpha^p(\boldsymbol{\eta}^*), \\ \widehat{v}_{k,l}^{n-1} &= v_h^{n-1}\Big(\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})\Big) = \sum_{|\alpha|=p} V_\alpha^{n-1} B_\alpha^p(\boldsymbol{\eta}^{**}), \end{aligned}$$

where $\boldsymbol{\eta}^* = F_{\mathcal{K}^*}^{-1}\Big(\mathcal{X}(t_n, \boldsymbol{x}_{k,l})\Big)$ and $\boldsymbol{\eta}^{**} = F_{\mathcal{K}^{**}}^{-1}\Big(\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})\Big)$. These summations are evaluated using the well-established Casteljau algorithm [41] on the reference element. It is worth noting that although we have considered only Dirichlet-type boundary condition in (1), the proposed method can also be used for general boundary conditions. For completeness, the formulation of our method for the general Robin-type boundary conditions is included in Appendix A. In summary, Algorithm depicts the steps used by the Bernstein-Bézier semi-Lagrangian finite element method to solve the coupled burgers' system.

Note that, since the cell-based shape functions are internal *i.e.*, they vanish on the element boundaries and are therefore not connected to the neighboring elements, the static condensation can be applied at the elemental level to remove the internal DoFs from the global finite element system during the assembly. Once the matrix and the right-hand side of the statically condensed system are formed, the internal DoFs in the solution can be recovered during

Algorithm The Bernstein-Bézier semi-Lagrangian finite element method.

```
Compute the entries $M_{ij}$ and $S_{ij}$ according to (21)-(25) and assemble the finite element matrices $\boldsymbol{M}$ and $\boldsymbol{S}$.
for each time step do
    for each mesh element $\mathcal{K}$ do
        For each Stroud quadrature point $\boldsymbol{x}_{k,l}$ calculate the departure points $\mathcal{X}(t_n, \boldsymbol{x}_{k,l})$ and $\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})$ using (5).
        Identify the mesh elements $\mathcal{K}^*$ and $\mathcal{K}^{**}$ where the departure points $\mathcal{X}(t_n, \boldsymbol{x}_{k,l})$ and $\mathcal{X}(t_{n-1}, \boldsymbol{x}_{k,l})$ belong.
        Evaluate the Bernstein-Bézier moments $b^u_\alpha$ and $b^v_\alpha$ using (32).
        Compute the entries $b^u_i$ and $b^v_i$ according to (22) and assemble the global right-hand side vectors $\boldsymbol{b}^u$ and $\boldsymbol{b}^v$.
    end for
    Compute $\boldsymbol{U}^{n+1}$ and $\boldsymbol{V}^{n+1}$ by solving the linear system (13).
    Update the solutions $u_h^{n+1}$ and $v_h^{n+1}$ element-wise according to (11).
end for
```

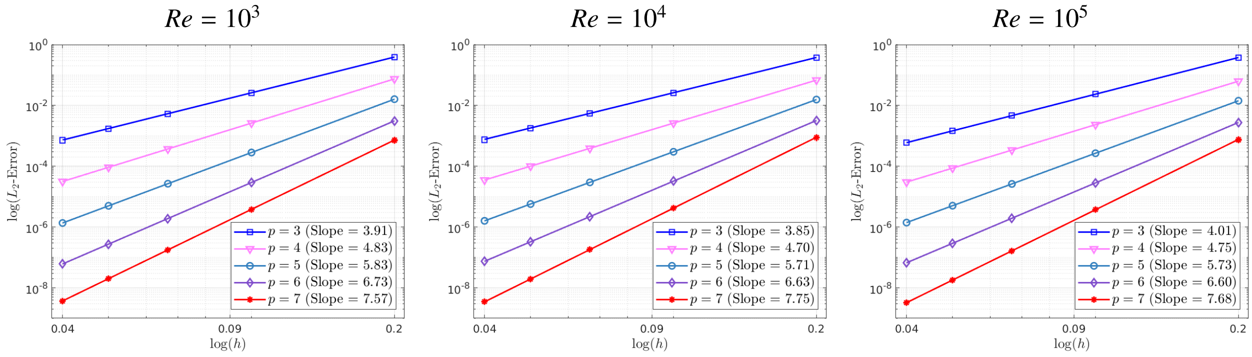

Figure 2: Convergence results using the L^2-error for Example 1 in the squared domain for $Re = 10^3$ (left), $Re = 10^4$ (middle) and $Re = 10^5$ (right) using different meshes and polynomial degrees at time $t = 1$.

the post-processing by solving element-wise local linear problems. This procedure is very efficient in reducing the size and enhancing the condition number of hp-finite element system matrices. Furthermore, the considered method requires solution of uncoupled elliptic problems such that their finite element discretization leads to linear systems of algebraic equations for which, very efficient solvers can be implemented. Therefore, by taking advantage of these properties, we solve the linear systems in (13) by the preconditioned conjugate gradient solver. This yields to an efficient method for solving this class of linear systems of algebraic equations, see for example [36].

4. Numerical results

We present numerical results for two examples of Burgers' equations to demonstrate the accuracy and efficiency of the proposed semi-Lagrangian Bernstein-Bézier finite element method. The analytical solutions for these examples are known, so that we can evaluate the relative L^1-error and relative L^2-error at time t_n as

$$L^1\text{-error} = \frac{\displaystyle\int_\Omega \left|\boldsymbol{u}_h^n - \boldsymbol{u}_{\text{exact}}^n\right| \, d\boldsymbol{x}}{\displaystyle\int_\Omega \left|\boldsymbol{u}_{\text{exact}}^n\right| \, d\boldsymbol{x}}, \qquad L^2\text{-error} = \frac{\sqrt{\displaystyle\int_\Omega \left|\boldsymbol{u}_h^n - \boldsymbol{u}_{\text{exact}}^n\right|^2 \, d\boldsymbol{x}}}{\sqrt{\displaystyle\int_\Omega \left|\boldsymbol{u}_{\text{exact}}^n\right|^2 \, d\boldsymbol{x}}}, \tag{33}$$

where $\boldsymbol{u}_{\text{exact}}^n$ and $\boldsymbol{u}_h^n$ are respectively, the exact and numerical solutions at gridpoint $\boldsymbol{x}_h$ and time t_n. We also define the CFL number associated to the Burgers' system (1) as

$$\text{CFL} = \max_{x,y}\left(\sqrt{|u|^2 + |v|^2}\right)\frac{\Delta t}{h/p}. \tag{34}$$

Note that to avoid the departure points to exit the computational domain we adopt the adaptive techniques discussed in [42]. The idea behind this adaptive procedure consists on adjusting the approximation of departure points in (4) using adaptive time sub-intervals. For the problems considered in the present study, this procedure is needed only for those Gauss-Jacobi quadrature points located at the inflow boundary of the computational domain. In all our computations carried out in this section, the resulting linear systems of algebraic equations are solved using the preconditioned conjugate gradient solver using the mass matrix as a preconditioner as discussed in [36]. The tolerance used for the stopping criteria is set to $10^{-6}h/p$, which is small enough to guarantee that the algorithm truncation errors dominate the total numerical errors. It should be noted that, to attend this tolerance, the maximum number of iterations in the linear solver does not overpass 40 iterations in all our simulations carried out in this section. All the computations were performed on an Intel® Core(TM) i7-7700HQ CPU @ 2.80GHz with 8 GB of RAM.

4.1. Example 1

Our first test case consists on solving the Burgers' system (1) in a squared domain $\Omega = [0, 1] \times [0, 1]$ subject to initial and boundary conditions obtained from the following analytical solution

$$\begin{aligned} u(x, y, t) &= -\frac{4\pi \exp\left(-\dfrac{5\pi^2 t}{Re}\right)\cos(2\pi x)\sin(\pi y)}{Re\left(2 + \pi\exp\left(-\dfrac{5\pi^2 t}{Re}\right)\sin(2\pi x)\sin(\pi y)\right)}, \\ v(x, y, t) &= -\frac{2\pi \exp\left(-\dfrac{5\pi^2 t}{Re}\right)\sin(2\pi x)\sin(\pi y)}{Re\left(2 + \pi\exp\left(-\dfrac{5\pi^2 t}{Re}\right)\sin(2\pi x)\sin(\pi y)\right)}. \end{aligned} \tag{35}$$

This example has also been used in [43, 44] among others to examine numerical methods for solving the two-dimensional Burgers' equations. Notice that the analytical solution (35) is used to quantify the errors and convergence rates in the proposed semi-Lagrangian Bernstein-Bézier finite element method for solving the Burgers' equations for three different Reynolds numbers $Re = 10^3$, $Re = 10^4$ and $Re = 10^5$ using different polynomial degrees p. In Figure 2 we present the convergence results obtained for the L^2-error using different meshes and different polynomial degrees

Table 1: Results for Example 1 in the squared domain for $Re = 10^3$ and $Re = 10^5$ on a mesh with $h = \frac{1}{20}$ at time $t = 1$ using different timesteps and two polynomial degrees $p = 6$ and $p = 7$.

		$Re = 10^3$				$Re = 10^5$			
p	Δt	L^1-error	Rate	L^2-error	Rate	L^1-error	Rate	L^2-error	Rate
6	4E-02	2.6519E-05	—	4.3371E-05	—	2.6483E-06	—	4.3476E-06	—
	2E-02	6.7930E-06	1.965	1.1087E-05	1.968	6.9814E-07	1.923	1.1494E-06	1.919
	1E-02	1.7328E-06	1.971	2.8296E-06	1.970	2.3838E-07	1.550	4.1446E-07	1.472
7	4E-02	2.6518E-05	—	4.3370E-05	—	2.6511E-06	—	4.3371E-06	—
	2E-02	6.7888E-06	1.966	1.1079E-05	1.969	6.7877E-07	1.966	1.1082E-06	1.969
	1E-02	1.7168E-06	1.983	2.7989E-06	1.985	1.7203E-07	1.980	2.8088E-07	1.980

Table 2: Computational costs for simulations obtained for Example 1 in the squared domain for $Re = 10^3$ at $t = 1$ using different meshes, polynomial degrees and CFL numbers. Note that the CPU times are given in seconds.

	$h = \frac{1}{4}$			$h = \frac{1}{8}$			$h = \frac{1}{16}$		
p	CFL = 4	CFL = 8	CFL = 10	CFL = 4	CFL = 8	CFL = 10	CFL = 4	CFL = 8	CFL = 10
4	6.210	2.355	2.531	22.986	11.719	9.286	52.434	26.118	21.321
5	9.934	4.565	3.774	38.464	18.104	14.878	85.325	42.746	33.899
6	14.215	6.198	5.525	55.676	27.722	22.417	127.071	64.095	50.270
7	19.990	9.035	8.246	78.646	39.780	31.919	181.886	90.708	73.396

at time $t = 1$ with CFL = 6. The convergence rates referred to by slopes of the error plots are also included in this figure. It is clear that keeping the polynomial degree p fixed and refining the spatial step h results in a substantial decrease in the computed L^2-errors. From the values of convergence rates in Figure 2, we also observe that the expected order of convergence is achieved for each selected polynomial degree p. It has also been observed that these convergence rates have not been deteriorated by the increase in the values of the Reynolds number, and the order of the proposed semi-Lagrangian Bernstein-Bézier finite element method remains almost the same for the considered Reynolds numbers $Re = 10^3$, $Re = 10^4$ and $Re = 10^5$.

It should be mentioned that to reduce the computational cost, the considered CFL number is chosen large enough which yield explicit Eulerian-based methods noncompetitive. In general, for a stable explicit Eulerian-based method, the CFL number must be less than unity. To examine the accuracy of the proposed BDF2 time stepping, we summarize in Table 1 both L^1-error and L^2-error obtained for $Re = 10^3$ and $Re = 10^5$ at time $t = 1$ using different timesteps and

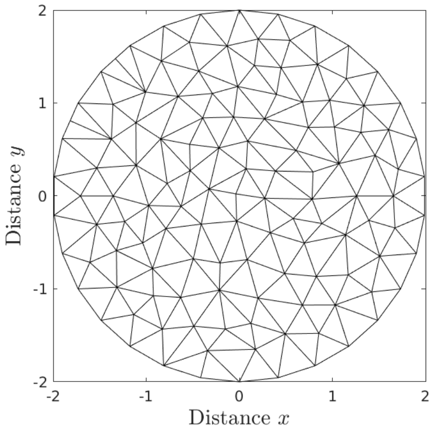

Figure 3: Computational mesh used for Example 1.

two polynomial degrees $p = 6$ and $p = 7$. Note that to relatively minimize errors of the spatial discretization, a fine structured mesh with $h = \frac{1}{20}$ is used in these simulations. For each polynomial degree p, we refine the timestep Δt and compute the corresponding L^1-error and L^2-error. The associated convergence rates in the time stepping are also included in Table 1. It is evident that decreasing the time step Δt results in a decrease of L^1-error and L^2-error for all considered polynomial degrees p and Reynolds numbers. The second-order convergence of the BDF2 time stepping is also achieved in this example, compare the convergence rates in Table 1. A slightly better convergence results are obtained for the simulations with $Re = 10^3$ than those obtained with $Re = 10^5$. It should also be stressed that similar results, not reported here for brevity, have been obtained using other polynomial degrees and Reynolds numbers. To quantify the comparison for this example, we present in Table 2 the computational costs obtained using different polynomial orders, mesh sizes and CFL numbers. Here, for each polynomial order and mesh size we increase the CFL number until the L^1-error starts to increase. Hence, three consistent values of CFL numbers are included in Table 2. It is evident to observe that the CPU time decreases proportionally to the inverse of the CFL number since the number of time steps decrease when the value of CFL increases for a fixed polynomial order and a fixed mesh size. Thus, this feature makes the proposed semi-Lagrangian method able to achieve high accuracy as well as high efficiency for the coupled Burgers' equations at high Reynolds numbers using relatively coarse meshes and large time steps.

Next, we examine the performance of the proposed semi-Lagrangian Bernstein-Bézier finite element method to solve the coupled Burgers' system on unstructured grids. To this end, we solve the above example in a circular domain centered at the origin with radius 2. A coarse mesh with 216 elements as shown in Figure 3 along with CFL = 9 are used in our simulations for this test example. Note that, for brevity in the presentation, we omit the results for the component u and we present the numerical solution of the component v only. In Figure 4 we display the results of the solution v using $p = 2$, 3 and 4 at time $t = 1$ for $Re = 10^4$. Note that for this test example, the solution (35) vanishes when the Reynolds number reaches high values. We also include the analytical solution in this figure for comparison reasons. For a better insight, Figure 5 exhibits radial cross-sections at $y = 0$ of the solution v for the results using

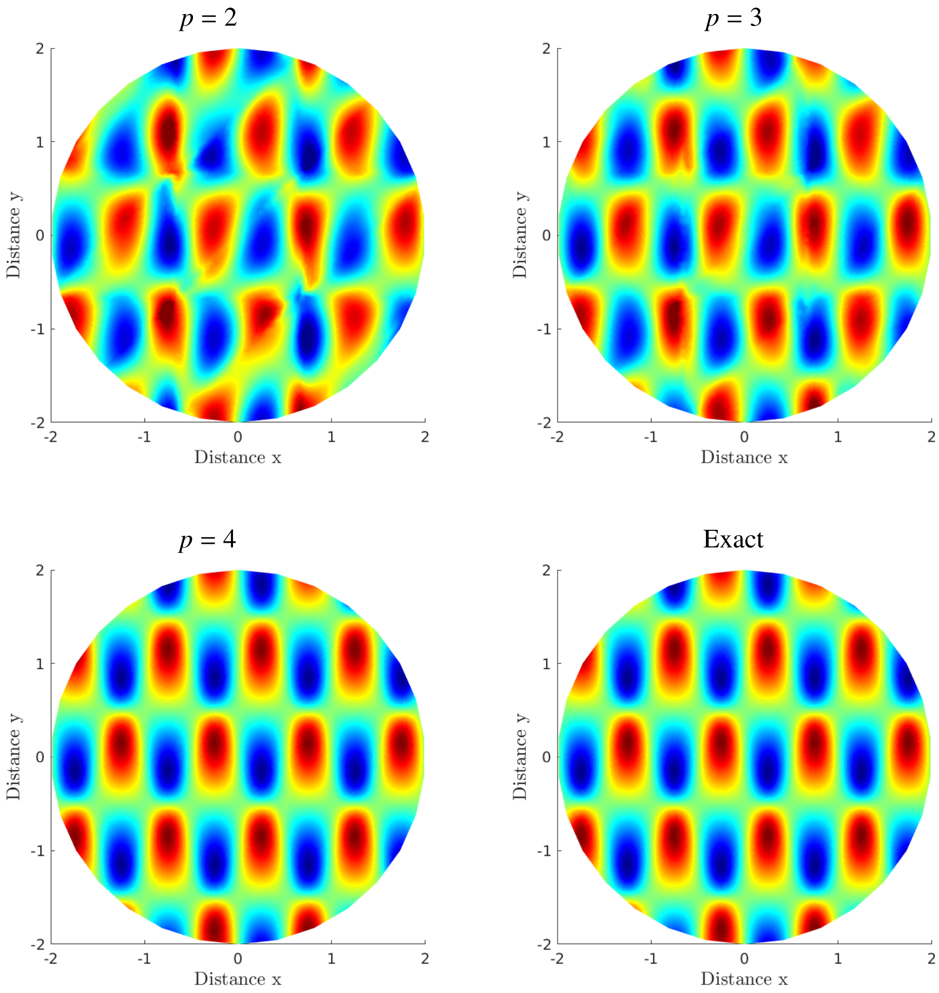

Figure 4: Snapshots of the solution v for Example 1 in the circular domain with $Re = 10^4$ at time $t = 1$ using $p = 2$, 3 and 4.

the considered polynomial degrees. The clear indication from Figure 4 and Figure 5 is that the numerical diffusion is clearly pronounced in the results obtained using quadratic polynomials ($p = 2$) for both Reynolds numbers $Re = 10^3$ (left plot) and $Re = 10^4$. From the same figures, the numerical solutions obtained using the cubic polynomials reduce this numerical dissipation and those results obtained using the quartic polynomials appear to provide the most accurate results compared to the analytical solution for this example. Refining the polynomial degrees further we have observed no visible differences in the results obtained using our method and the analytical solutions. It should be pointed out that the performance of the proposed semi-Lagrangian Bernstein-Bézier finite element method is very attractive since the computed solutions remain stable and highly accurate even when coarse meshes are used without requiring nonlinear solvers or small time steps to be taken in the simulations.

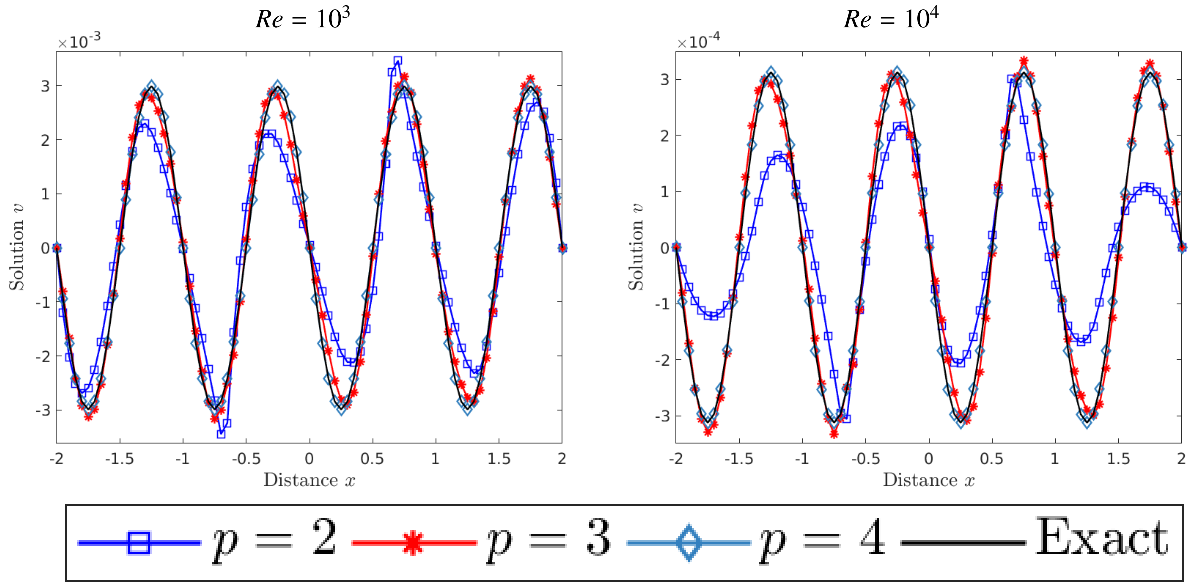

Figure 5: Radial cross-sections at $y = 0$ of the solution v for Example 1 in the circular domain using different polynomial degrees for $Re = 10^3$ (left plot) and $Re = 10^4$ (right plot).

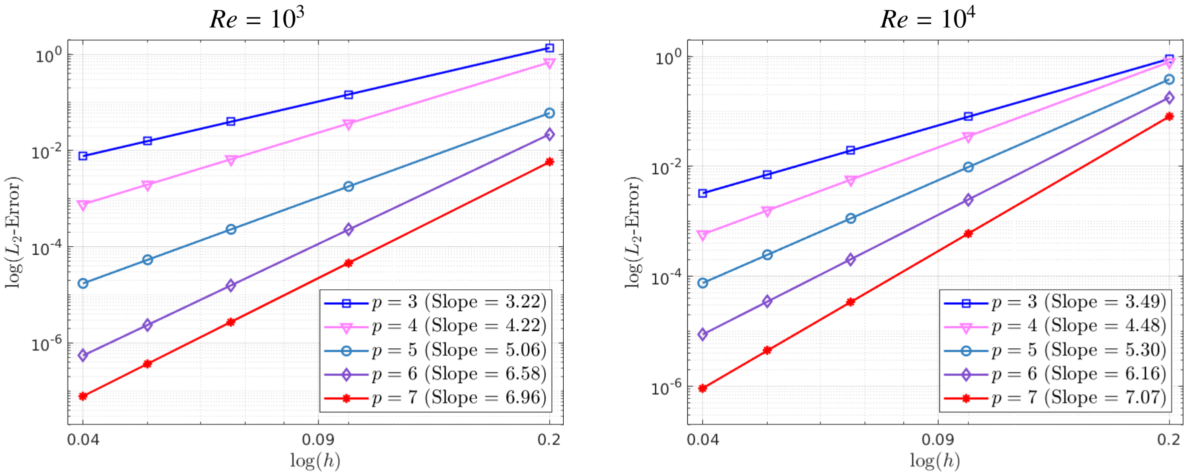

Figure 6: Convergence results using the L^2-error for Example 2 in the squared domain for $Re = 10^3$ (left) and $Re = 10^4$ (right) using different meshes and polynomial degrees at time $t = 1$.

4.2. Example 2

To assess the numerical performance of our semi-Lagrangian Bernstein-Bézier finite element method to accurately solve the nonlinear Burgers' system with steep gradients, we solve the Burgers' system (1) in a squared domain $\Omega = [0, 1] \times [0, 1]$ equipped with initial and boundary conditions obtained from the following analytical solution

$$\begin{aligned} u(x,y,t) &= \frac{3}{4} - \frac{1}{4\left(1+\exp\left(-\frac{(4x-4y+t)\,Re}{32}\right)\right)}, \\ v(x,y,t) &= \frac{3}{4} + \frac{1}{4\left(1+\exp\left(-\frac{(4x-4y+t)\,Re}{32}\right)\right)}. \end{aligned} \tag{36}$$

A similar test example has been considered in [45, 43, 46, 44, 47] among others. Unlike the previous example, the solution of this problem is expected to develop strong shocks at high Reynolds numbers which are challenging to capture using the conventional Eulerian-based finite element methods. First, we perform a convergence study for this problem in the same manner as in the previous test example. In Figure 6 we report the convergence plots in terms of the L^2-error for two Reynolds numbers $Re = 10^3$ and $Re = 10^4$ using different meshes and polynomial degrees at $t = 1$ with CFL = 6. We also include the convergence rates referred to by slopes of the error plots in this figure. Again, by maintaining the polynomial degree p fixed and adopting the h-refinement it yields a decay of all values of the L^2-error. For instance, at the high Reynolds number $Re = 10^4$ for which the convection is strongly dominated in (1), obtaining satisfactory results is evident in this case and it can noted that the p-refinement strategy is precise enough in this situation. For $Re = 10^3$ which can be considered as a critical value for which shocks are developed in the computed solutions, it can be observed from Figure 6 that all optimal orders are clearly attained in the proposed method. Note that the convergence rates in Figure 6 reveal that the expected order of convergence is achieved in our semi-Lagrangian Bernstein-Bézier finite element method for both Reynolds numbers $Re = 10^3$ and $Re = 10^4$. It is also evident from the results reported in Figure 6 that the convergence rates in the proposed semi-Lagrangian Bernstein-Bézier finite element method are not highly affected by the Reynolds numbers taken in the numerical simulations. Thus, fixing the polynomial degree p and adopting an h-refinement results in a decrease of the computed errors for the selected Reynolds numbers.

As in the previous example, convergence rates in time for the semi-Lagrangian Bernstein-Bézier finite element method have also been assessed in this test example. To achieve this comparison, we compute the L^1-error and L^2-error using a fixed fine mesh with $h = \frac{1}{20}$ and we carry out some numerical experiments varying the polynomial degree p and the timestep Δt. The obtained results for $Re = 10^3$ and $Re = 10^4$ at time $t = 1$ using the polynomial degrees $p = 6$ and $p = 7$ are listed in Table 3. It is clear that decreasing the timestep Δt yields a decrease in both L^1-error and L^2-error for all considered polynomial degrees and Reynolds numbers. It is also evident that the semi-Lagrangian Bernstein-Bézier finite element method using the BDF2 time stepping exhibits an accuracy of the expected second-order convergence in time. As in the previous example, a slightly more accurate L^1-error and L^2-error are obtained for the simulations with $Re = 10^3$ than those obtained with $Re = 10^4$.

Table 3: Results for Example 2 in the squared domain for $Re = 10^3$ and $Re = 10^4$ on a mesh with $h = \frac{1}{20}$ at time $t = 1$ using different timesteps and the polynomial degrees $p = 6$ and $p = 7$.

		$Re = 10^3$				$Re = 10^4$			
p	Δt	L^1-error	Rate	L^2-error	Rate	L^1-error	Rate	L^2-error	Rate
6	4E-02	5.4241E-04	—	7.6452E-04	—	2.0808E-03	—	8.4176E-03	—
	2E-02	1.3937E-04	1.960	1.9580E-04	1.965	5.3520E-04	1.959	2.1726E-03	1.954
	1E-02	3.5321E-05	1.980	4.9517E-05	1.983	1.4320E-04	1.902	5.8578E-04	1.891
7	4E-02	5.4241E-04	—	7.6453E-04	—	2.0806E-03	—	8.4175E-03	—
	2E-02	1.3938E-04	1.960	1.9580E-04	1.965	5.2087E-04	1.998	2.1088E-03	1.997
	1E-02	3.5318E-05	1.981	4.9516E-05	1.983	1.3416E-04	1.957	5.5533E-04	1.925

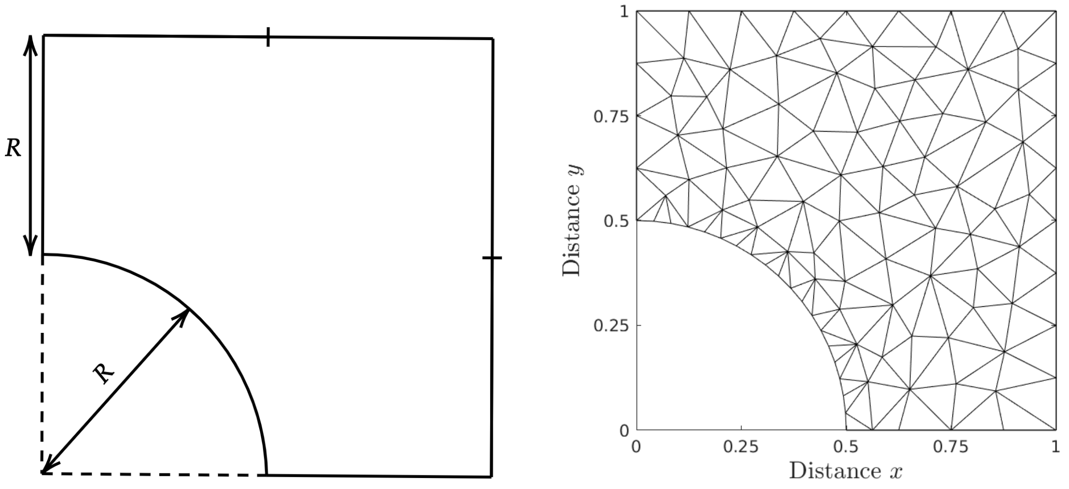

Figure 7: Computational geometry (left plot) and the computational mesh (right plot) used for Example 2.

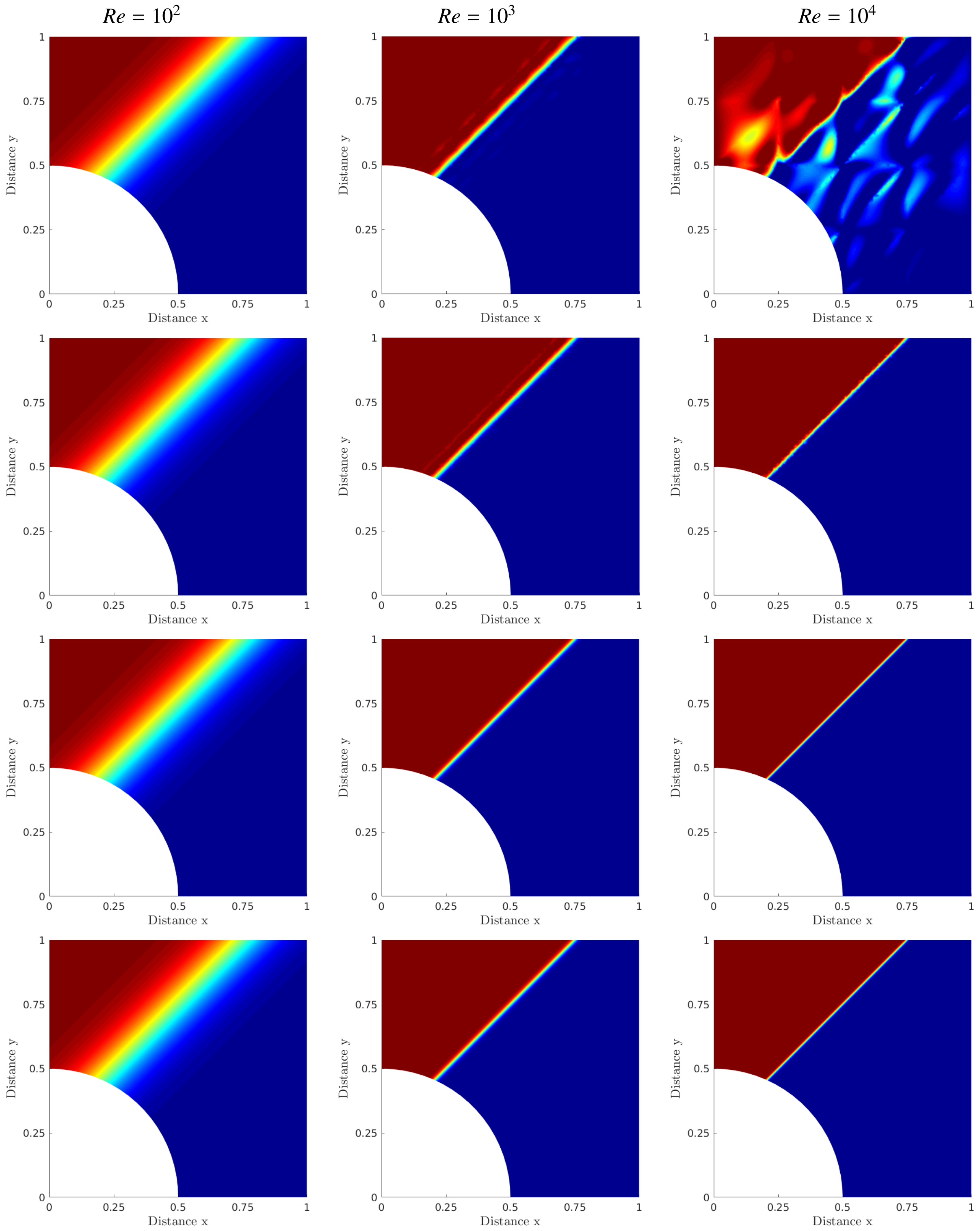

Figure 8: Snapshots of the solution u for Example 2 in the complex domain with $Re = 10^2$ (first column) $Re = 10^3$ (second column) and $Re = 10^4$ (third column) at time $t = 1$ obtained using $p = 3$ (first row), $p = 5$ (second row) and $p = 7$ (third row) and the exact solution (forth row).

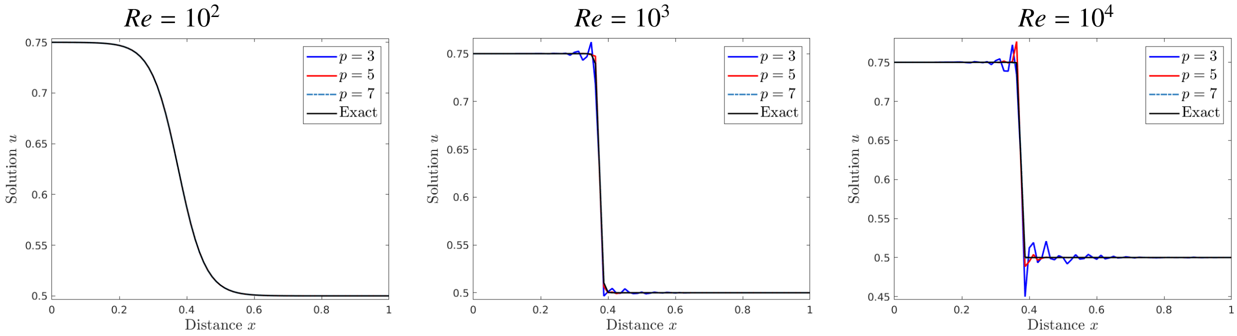

Figure 9: Cross-sections at the diagonal $x = 1 - y$ of the solution u for Example 2 in the complex domain using different polynomial degrees for $Re = 10^2$ (left plot), $Re = 10^3$ (middle plot) and $Re = 10^4$ (right plot).

Now we turn our attention to the performance of the proposed semi-Lagrangian Bernstein-Bézier finite element method for solving the coupled Burgers' equations in a complex domain. To this end, we solve the equations (1) and (36) in the domain shown in Figure 7. The domain consists of a unit square with a circular hole of radius $R = 0.5$ and only quarter of this domain is used in our simulations. Figure 7 also illustrates the unstructured mesh with 168 elements used in the simulations for this test example. Since the solution in (36) satisfies $u + v = 3/2$, we omit the results for the component v and we present the numerical results for the component u only. In Figure 8 we present snapshots of the solution obtained at $t = 1$ using CFL = 6 and different polynomial orders. Here, results are presented for three different Reynolds coefficients namely, $Re = 10^2$, $Re = 10^3$ and $Re = 10^4$. The analytical solutions are also included in Figure 8 for comparison reasons. It is clear that, by increasing the values of the Reynolds number Re, the convective term becomes dominant and steep internal layers are formed near the vicinity of the front lines in the computational domain. At the high Reynolds number $Re = 10^4$, the cubic Bernstein-Bézier elements fail to produce good results and the results obtained using higher polynomial degrees match the analytical solution. Note that for the case with $Re = 10^4$, the numerical solution develops a strong shock which can not be captured by Bernstein-Bézier elements of low degrees. Apparently, noticeable oscillations are detected for results obtained when low values of polynomial orders are used. These results give a clear view of the overall nonlinear convective pattern and effects of the Reynolds numbers. It is worth remarking that the steepness of the diagonal shock with increasing Re is evident from these plots and the shock region is smaller for $Re = 10^4$ than for $Re = 10^2$. These features clearly demonstrate the high accuracy achieved by the proposed semi-Lagrangian Bernstein-Bézier finite element method for solving the coupled Burgers' system at high Reynolds numbers using coarse meshes and large timesteps. To further illustrate the effect of the p-refinement on the numerical solutions for this example, we present in Figure 9 the cross-sections of the numerical solutions at the main diagonal $y = 1 - x$ for $Re = 10^2$, $Re = 10^3$ and $Re = 10^4$ using different polynomial degrees. At $Re = 10^2$, no significant differences are observed in the obtained results using the considered polynomial degrees $p = 3$, 5 and 7. Increasing the Reynolds number to $Re = 10^3$ and 10^4, nonphysical oscillations are clearly visible in those regions where the solution gradients are steep when small values of the polynomial order are used $p = 3$ and $p = 5$. However, for $p = 7$, the solution structures are in good agreement with the exact solution presented in Figure 9. From the same results we can confirm that the Bernstein-Bézier finite elements of order $p = 7$ are suitable to capture the sharp gradients in the solutions and produce highly accurate results for this example on a coarse mesh without requiring nonlinear solvers or small timesteps to be taken in the simulations. It should be stressed that the oscillations appearing in the shock areas at high Reynolds numbers in Figure 8 and Figure 9 should not be confused with the Gibbs and Runge phenomena widely known in numerical methods for solving hyperbolic equations of conservation laws because the Bernstein-Bézier basis functions are monotone by reconstruction. In addition, the local L^2-projection used in the present study for the semi-Lagrangian method has the properties to overcome the Gibbs and Runge phenomena by introducing numerical diffusion at the shock regions. It is expected that the oscillations in Figure 8 and Figure 9 to be eliminated by either refining the mesh or by increasing the degree of Bernstein-Bézier basis functions.

Finally, from the computational results obtained for this class of two-dimensional coupled Burgers' problems, one may conclude the following: (i) the proposed semi-Lagrangian Bernstein-Bézier finite element method highly resolve these test problems on coarse meshes, (ii) the convergence rates of the method are not deteriorated when increasing both the Reynolds numbers and CFL numbers. These features, as well as its favorable high-order accuracy and strong stability properties, make the proposed semi-Lagrangian Bernstein-Bézier finite element method an attractive alternative for solvers based on Eulerian-based finite element discretizations of the nonlinear Burgers' equations.

5. Conclusions

We presented a semi-Lagrangian Bernstein-Bézier finite element method for solving two-dimensional coupled Burgers' system on unstructured meshes at high Reynolds numbers. A key idea to obtain an efficient and stable finite element method is the use of high order Bernstein-Bézier basis functions on coarse computational meshes along with the modified method of characteristics to accurately resolve the convection terms using large time steps. The proposed method preserves the advantages of the semi-Lagrangian integration including the unconditional stability, reduction of time truncation errors, decoupling the system and eliminating the nonlinear terms, while achieving high-order accuracy in the computed solutions at all selected Reynolds numbers. Numerical results for two examples of Burgers' equations with known analytical solutions are able to provide very accurate results in both regular and complex geometries. It has been shown that the proposed semi-Lagrangian Bernstein-Bézier finite element method enjoys the computational advantages and it achieves accurate solutions for high Reynolds numbers. In the second paper in this series, we will extend the work presented here to solving the incompressible Navier-Stokes equations in three space dimensions using unstructured tetrahedral meshes.

Appendix A. Formulation for Robin-type boundary conditions

Let us consider the coupled Burgers' equations (1) subject to the following Robin-type boundary conditions

$$\begin{aligned} u + \frac{1}{Re}\frac{\partial u}{\partial \mathbf{n}} &= g_u, \qquad \text{on} \qquad [0,T]\times\Gamma, \\ v + \frac{1}{Re}\frac{\partial v}{\partial \mathbf{n}} &= g_v, \qquad \text{on} \qquad [0,T]\times\Gamma, \end{aligned} \tag{A.1}$$

where $\mathbf{n}$ is the outward unit normal on the boundary Γ, g_u and g_v are the prescribed boundary functions. Using the implicit BDF2 scheme for the time integration results in

$$\begin{aligned} u^{n+1} - \frac{2\Delta t}{3Re}\left(\frac{\partial^2 u^{n+1}}{\partial x^2} + \frac{\partial^2 u^{n+1}}{\partial y^2}\right) &= \frac{4}{3}\widehat{u}^n - \frac{1}{3}\widehat{u}^{n-1}, \\ v^{n+1} - \frac{2\Delta t}{3Re}\left(\frac{\partial^2 v^{n+1}}{\partial x^2} + \frac{\partial^2 v^{n+1}}{\partial y^2}\right) &= \frac{4}{3}\widehat{v}^n - \frac{1}{3}\widehat{v}^{n-1}, \end{aligned} \tag{A.2}$$

with a similar approximation for the boundary conditions (A.1) as

$$\begin{aligned} u^{n+1} + \frac{1}{Re}\frac{\partial u^{n+1}}{\partial \mathbf{n}} &= g_u^{n+1}, \qquad \text{on} \qquad \Gamma, \\ v^{n+1} + \frac{1}{Re}\frac{\partial v^{n+1}}{\partial \mathbf{n}} &= g_v^{n+1}, \qquad \text{on} \qquad \Gamma. \end{aligned} \tag{A.3}$$

For the weak form, we multiply the system (A.2) by a test function w and we integrate over the domain Ω to obtain the following equations

$$\begin{aligned} \int_\Omega u^{n+1} w d\boldsymbol{x} - \frac{2\Delta t}{3Re}\int_\Omega \left(\frac{\partial^2 u^{n+1}}{\partial x^2} + \frac{\partial^2 u^{n+1}}{\partial y^2}\right) w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{u}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{u}^{n-1} w d\boldsymbol{x}, \\ \int_\Omega v^{n+1} w d\boldsymbol{x} - \frac{2\Delta t}{3Re}\int_\Omega \left(\frac{\partial^2 v^{n+1}}{\partial x^2} + \frac{\partial^2 v^{n+1}}{\partial y^2}\right) w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{v}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{v}^{n-1} w d\boldsymbol{x}. \end{aligned} \tag{A.4}$$

Using the Green theorem, system (A.4) becomes

$$\begin{aligned} \int_\Omega u^{n+1} w d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_\Omega \nabla u^{n+1}\cdot\nabla w d\boldsymbol{x} - \frac{2\Delta t}{3Re}\oint_\Gamma \frac{\partial u^{n+1}}{\partial \mathbf{n}} w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{u}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{u}^{n-1} w d\boldsymbol{x}, \\ \int_\Omega v^{n+1} w d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_\Omega \nabla v^{n+1}\cdot\nabla w d\boldsymbol{x} - \frac{2\Delta t}{3Re}\oint_\Gamma \frac{\partial v^{n+1}}{\partial \mathbf{n}} w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{v}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{v}^{n-1} w d\boldsymbol{x}. \end{aligned} \tag{A.5}$$

Next, we replace the normal derivatives $\frac{1}{Re}\frac{\partial u^{n+1}}{\partial \mathbf{n}}$ and $\frac{1}{Re}\frac{\partial v^{n+1}}{\partial \mathbf{n}}$ in (A.5) by the approximation values in (A.3)

$$\begin{aligned} \int_\Omega u^{n+1} w d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_\Omega \nabla u^{n+1}\cdot\nabla w d\boldsymbol{x} + \frac{2\Delta t}{3}\oint_\Gamma u^{n+1} w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{u}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{u}^{n-1} w d\boldsymbol{x} + \frac{2\Delta t}{3}\oint_\Gamma g_u^{n+1} w d\boldsymbol{x}, \\ \int_\Omega v^{n+1} w d\boldsymbol{x} + \frac{2\Delta t}{3Re}\int_\Omega \nabla v^{n+1}\cdot\nabla w d\boldsymbol{x} - \frac{2\Delta t}{3}\oint_\Gamma v^{n+1} w d\boldsymbol{x} &= \frac{4}{3}\int_\Omega \widehat{v}^n w d\boldsymbol{x} - \frac{1}{3}\int_\Omega \widehat{v}^{n-1} w d\boldsymbol{x} + \frac{2\Delta t}{3}\oint_\Gamma g_v^{n+1} w d\boldsymbol{x}. \end{aligned}$$

Following the same procedure as presented in Section 3, we replace the solution (u, v) by its discrete form and the test function w by the basis function φ_i to obtain

$$\begin{aligned}\left(\boldsymbol{M} + \frac{2\Delta t}{3Re}\boldsymbol{S} + \frac{2\Delta t}{3}\boldsymbol{M}^{\Gamma}\right)\boldsymbol{U}^{n+1} &= \boldsymbol{b}_u + \boldsymbol{b}_u^{\Gamma},\\ \left(\boldsymbol{M} + \frac{2\Delta t}{3Re}\boldsymbol{S} + \frac{2\Delta t}{3}\boldsymbol{M}^{\Gamma}\right)\boldsymbol{V}^{n+1} &= \boldsymbol{b}_v + \boldsymbol{b}_v^{\Gamma},\end{aligned} \tag{A.6}$$

where $\boldsymbol{M}$ is the $m \times m$-valued mass matrix, the entries of which are

$$M_{ij}^{\Gamma} = \oint_{\Gamma_h} \varphi_j(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i, j \leq m,$$

$\boldsymbol{b}_u^{\Gamma}$ and $\boldsymbol{b}_v^{\Gamma}$ are m-valued right-hand side vectors with entries

$$\begin{aligned} b_i^{u,\Gamma} &= \frac{2\Delta t}{3}\oint_{\Gamma_h} g_u^{n+1}(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i \leq m,\\ b_i^{v,\Gamma} &= \frac{2\Delta t}{3}\oint_{\Gamma_h} g_v^{n+1}(\boldsymbol{x})\varphi_i(\boldsymbol{x})d\boldsymbol{x}, \qquad 1 \leq i \leq m.\end{aligned} \tag{A.7}$$

To compute the entries M_{ij}^{Γ} of $\boldsymbol{M}^{\Gamma}$ we follow the same procedure used to compute the entries of $\boldsymbol{M}$. Thus, we consider the edge $E \subset \Gamma_h$ of a generic element $\mathcal{K}$, there exists at least and only two vertex-based shape functions and $(p-1)$ edge-based shape functions $B_{\alpha}^{p}|_E \neq 0$. The shape functions defined along the edge E can be written as a one-dimensional restriction of the two-dimensional Bernstein polynomials and defined by

- Vertex-based shape functions defined by

$$\varphi^{\mathbf{q}_1}(\boldsymbol{x}) = B_{(p,0)}^{p}(\zeta) = \zeta^p, \qquad \varphi^{\mathbf{q}_2}(\boldsymbol{x}) = B_{(0,p)}^{p}(\zeta) = (1-\zeta)^p.$$

- Interior-based shape functions defined by

$$\varphi^{e}(\boldsymbol{x}) = B_{(p-k,k)}^{p}(\zeta) = \begin{pmatrix} p \\ k \end{pmatrix}\zeta^{p-k}(1-\zeta)^k, \qquad 1 \leqslant k \leqslant p-1.$$

Here, ζ is defined such that $\boldsymbol{x} = \zeta\mathbf{q}_1 + (1-\zeta)\mathbf{q}_2$ where $\mathbf{q}_1$ and $\mathbf{q}_2$ are the vertices of the edge E. Therefore,

$$\begin{aligned} M_{\alpha,\beta}^{\Gamma} &= \oint_E B_{\alpha}^{p}(\zeta)B_{\beta}^{p}(\zeta)d\boldsymbol{x},\\ &= \int_0^1 |E|B_{\alpha}^{p}(\zeta)B_{\beta}^{p}(\zeta)\,\mathrm{d}\zeta,\\ &= |E|\frac{\binom{\alpha+\beta}{\beta}}{\binom{2p}{p}}\int_0^1 B_{\alpha+\beta}^{2p}(\zeta)\mathrm{d}\zeta,\\ &= |E|\frac{\binom{\alpha+\beta}{\beta}}{\binom{2p}{p}}\frac{1}{\binom{p+1}{1}},\end{aligned} \tag{A.8}$$

where $|E|$ is the length of the edge E. On the other hand, the approximation of the new quantities in the right-hand side is carried out as follows: for a given edge E where $\varphi_i \neq 0$, there exists an integer α where

$$\varphi_i(\boldsymbol{x}) = B_\alpha^p(\zeta), \qquad \boldsymbol{x} = \zeta \mathbf{q}_1 + (1-\zeta)\mathbf{q}_2.$$

Thus,

$$\begin{aligned} \oint_E g_u^{n+1}(\boldsymbol{x})\varphi_i(\boldsymbol{x}) d\boldsymbol{x} &= |E| \int_0^1 g_u^{n+1}(\boldsymbol{x}) B_\alpha^p(\zeta) d\zeta, \\ &\approx |E| \sum_{k=1}^{q} w_k^{(0)} B_\alpha^p(z_k^{(0)}) g_u^{n+1}(\boldsymbol{x}_k), \end{aligned} \tag{A.9}$$

where $\left(z_k^{(0)}, w_k^{(0)}\right)$ and $\boldsymbol{x}_k = z_k^{(0)}\mathbf{q}_1 + (1-z_k^{(0)})\mathbf{q}_2$ denote the Gauss-Jacobi quadrature points and their associated weights, respectively.

References

[1] P. A. Mirzaei, Cfd modeling of micro and urban climates: Problems to be solved in the new decade, Sustainable Cities and Society 69 (2021) 102839.

[2] J. D. Anderson Jr, Fundamentals of aerodynamics, Tata McGraw-Hill Education, 2010.

[3] J. Vassberg, Expectations for computational fluid dynamics, International Journal of Computational Fluid Dynamics 19 (8) (2005) 549–558.

[4] J. Vos, A. Rizzi, D. Darracq, E. Hirschel, Navier–stokes solvers in european aircraft design, Progress in Aerospace Sciences 38 (8) (2002) 601–697.

[5] G. A. Pope, et al., The application of fractional flow theory to enhanced oil recovery, Society of Petroleum Engineers Journal 20 (03) (1980) 191–205.

[6] A. Liñán, F. A. Williams, Fundamental aspects of combustion (1993).

[7] A. Brooks, T. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer methods in applied mechanics and engineering 32 (1-3) (1982) 199–259.

[8] F. Bertrand, L. Demkowicz, J. Gopalakrishnan, N. Heuer, Recent Advances in Least-Squares and Discontinuous Petrov-Galerkin Finite Element Methods, Computational Methods in Applied Mathematics 19 (3) (2019) 395–397.

[9] T. J. Hughes, L. P. Franca, G. M. Hulbert, A new finite element formulation for computational fluid dynamics: Viii. The Galerkin/least-squares method for advective-diffusive equations, Computer methods in applied mechanics and engineering 73 (2) (1989) 173–189.

[10] J. Donea, A Taylor-Galerkin method for convective transport problems, International Journal for Numerical Methods in Engineering 20 (1) (1984) 101–119.

[11] E. Carew, P. Townsend, M. Webster, A Taylor-Petrov-Galerkin algorithm for viscoelastic flow, Journal of non-newtonian fluid mechanics 50 (2-3) (1993) 253–287.

[12] A. BaShan, A numerical treatment of the coupled viscous burgers' equation in the presence of very large reynolds number, Physica A: Statistical Mechanics and its Applications 545 (2020) 123755.

[13] A. Jafarabadi, E. Shivanian, Numerical simulation of nonlinear coupled burgers' equation through meshless radial point interpolation method, Engineering Analysis with Boundary Elements 95 (2018) 187–199.

[14] J. Kaur, R. K. Gupta, S. Kumar, On explicit exact solutions and conservation laws for time fractional variable - coefficient coupled burgers' equations, Communications in Nonlinear Science and Numerical Simulation 83 (2020) 105108.

[15] C. M. Khalique, S. A. Abdallah, Coupled burgers equations governing polydispersive sedimentation; a lie symmetry approach, Results in Physics 16 (2020) 102967.

[16] Y. Zhang, J. Lin, S. Reutskiy, H. Sun, W. Feng, The improved backward substitution method for the simulation of time-dependent nonlinear coupled burgers' equations, Results in Physics 18 (2020) 103231.

[17] M. El-Amrani, M. Seaid, Numerical simulation of natural and mixed convection flows by Galerkin-characteristics method, Int. J. Num. Meth. Fluids. 53 (2007) 1819–1845.

[18] M. El-Amrani, M. Seaid, Convergence and stability of finite element modified method of characteristics for the incompressible Navier-Stokes equations, J. Numer. Math. 15 (2007) 101–135.

[19] M. El-Amrani, M. Seaid, A finite element semi-Lagrangian method with L^2 interpolation, International Journal for numerical methods in engineering 90 (2012) 1485–1507.

[20] M. El-Amrani, M. Seaid, An essentially non-oscillatory semi-Lagrangian method for tidal flow simulations, International journal for numerical methods in engineering 81 (2010) 805–834.

[21] M. El-Amrani, M. Seaid, Eulerian-lagrangian time-stepping methods for convection-dominated problems, International Journal of Computer Mathematics 85 (3-4) (2008) 421–439.

[22] C. Temperton, A. Staniforth, An efficient two-time-level Galerkin-characteristics semi-implicit integration scheme, Quart. J. Roy. Meteor. Soc. 113 (1987) 1025–1039.

[23] M. Seaid, Semi-lagrangian integration schemes for viscous incompressible flows, Comp. Methods in App. Math. 4 (2002) 392–409.

[24] M. El-Amrani, M. Seaid, An L^2-projection for the galerkin-characteristic solution of incompressible flows, SIAM Journal on Scientific Computing 33 (6) (2011) 3110–3131.

[25] F. X. Giraldo, The lagrange-Galerkin spectral element method on unstructured quadrilateral grids, J. of Comp. Physics 147 (1998) 114–146.

[26] G. E. Karniadakis, S. J. Sherwin, Spectral/hp element methods for computational fluid dynamics, Numerical Mathematics and Scientific Computation., Oxford University Press, 2005.

[27] I. Babuška, M. Suri, The *p* and *hp* versions of the finite element method, basic principles and properties, SIAM Rev. 36 (1994) 578–632.

[28] C. Schwab, M. Suri, The *p* and *hp* versions of the finite element method for problems with boundary layers, Math. Comp. 65 (1996) 1403–1429.

[29] J. Melenk, *hp*-finite element methods for for singular perturbations. Vol. 1796 of Lecture Notes in Mathematics, Springer-Verlag, 2002.

[30] I. Babuška, M. Griebel, J. Pitkaränta, The problem of selecting the shape functions for *p*-type elements, International Journal for Numerical Methods in Engineering 28 (1986) 1891–1908.

[31] P. Šolín, K. Segeth, I. Doležel, Higher-Order Finite Element Methods, Chapman & Hall, 2003.

[32] S. Petersen, D. Dreyer, V. Estorff, Assessment of finite and spectral element shape functions for efficient iterative simulations of interior acoustics, Comput. Methods Appl. Mech. Engrg 195 (2006) 6463–6478.

[33] A. El Kacimi, O. Laghrouche, M. Shadi, J. Trevelyan, Bernstein-Bézier based finite elements for efficient solution of short wave problems., Comput. Methods Appl. Mech. Engrg. 343 (2019) 166–185.

[34] M. Ainsworth, O. Davydov, Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures., SIAM Journal on Scientific Computing 33 (2011) 3087–3109.

[35] R. C. Kirby, K. T. Thinh, Fast simplicial quadrature–based finite element operators using Bernstein polynomials., Numerische Mathematik 121 (2012) 261–279.

[36] M. Ainsworth, S. Jiang, M. Sanchéz, An $O(p^3)$ *hp*-version fem in two dimensions: Preconditioning and post-processing, Comput. Methods Appl. Mech. Engrg 350 (2019) 766–802.

[37] W. Hundsdorfer, Partially implicit bdf2 blends for convection dominated flows, SIAM Journal on Numerical Analysis 38 (6) (2001) 1763–1783.

[38] R. Löhner, J. Ambrosiano, A vectorized particle tracer for unstructured grids, Journal of Computational Physics 91 (1) (1990) 22–31.

[39] T. Goodman, Variation diminishing properties of Bernstein polynomials on triangles, J. of Approx. Theory 50 (1987) 111–126.

[40] M. Floater, J. Pena, Monotonicity preservation on triangles, Mathematics of Computation 69 (2000) 1505–1519.

[41] M. J. Lai, L. L. Schumaker, Spline Functions on Triangulations, Encyclopedia Math. Appl. 110., Cambridge University Press, 2007.

[42] R. Bermejo, L. Saavedra, Lagrange-Galerkin methods for the incompressible Navier-Stokes equations: A review, Communications in Applied and Industrial Mathematics 7 (3) (2016) 26–55.

[43] L. Wenyuan, A fourth-order finite-difference method for solving the system of two-dimensional Burgers' equations, International journal for numerical methods in fluids 64 (5) (2010) 565–590.

[44] E. Ngondiep, An efficient three-level explicit time-split scheme for solving two-dimensional unsteady nonlinear coupled Burgers' equations, International Journal for Numerical Methods in Fluids 92 (4) (2020) 266–284.

[45] Y. Zhang, J. Lin, S. Reutskiy, H. Sun, W. Feng, The improved backward substitution method for the simulation of time-dependent nonlinear coupled Burgers' equations, Results in Physics (2020).

[46] R. Kannan, Z. Wang, A high order spectral volume solution to the Burgers' equation using the Hopf-Cole transformation, International journal for numerical methods in fluids 69 (4) (2012) 781–801.

[47] F. Shi, H. Zheng, Y. Cao, J. Li, R. Zhao, A fast numerical method for solving coupled burgers' equations, Numerical Methods for Partial Differential Equations 33 (6) (2017) 1823–1838.

Durham University
Durham Research Online

Citation on deposit: El-Amrani, M., Khouya, B., & Seaid, M. (2022). A semi-Lagrangian Bernstein–Bézier finite element method for two-dimensional coupled Burgers' equations at high Reynolds numbers. Mathematics and Computers in Simulation, 199, 160-181. https://doi.org/10.1016/j.matcom.2022.03.011

For final citation and metadata, visit Durham Research Online URL: https://durham-repository.worktribe.com/output/1177624

Copyright statement: © 2022. This manuscript version is made available under the CC-BY-NC-ND 4.0
license https://creativecommons.org/licenses/by-nc-nd/4.0/