Journal of Scientific Computing (2022) 92:58
https://doi.org/10.1007/s10915-022-01888-7

Bernstein-Bézier Galerkin-Characteristics Finite Element Method for Convection-Diffusion Problems

Mofdi El-Amrani[1] · Abdellah El-Kacimi[2] · Bassou Khouya[3] · Mohammed Seaid[4]

Received: 31 July 2021 / Revised: 10 March 2022 / Accepted: 25 April 2022

Abstract
A class of Bernstein-Bézier basis based high-order finite element methods is developed for the Galerkin-characteristics solution of convection-diffusion problems. The Galerkin-characteristics formulation is derived using a semi-Lagrangian discretization of the total derivative in the considered problems. The spatial discretization is performed using the finite element method on unstructured meshes. The Lagrangian interpretation in this approach greatly reduces the time truncation errors in the Eulerian methods. To achieve high-order accuracy in the Galerkin-characteristics solver, the semi-Lagrangian method requires high-order interpolating procedures. In the present work, this step is carried out using the Bernstein-Bézier basis functions to evaluate the solution at the departure points. Triangular Bernstein-Bézier patches are constructed in a simple and inherent manner over finite elements along the characteristics. An efficient preconditioned conjugate gradient solver is used for the linear systems of algebraic equations. Several numerical examples including advection-diffusion equations with known analytical solutions and the viscous Burgers problem are considered to illustrate the accuracy, robustness and performance of the proposed approach. The computed results support our expectations for a stable and highly accurate Bernstein-Bézier Galerkin-characteristics finite element method for convection-diffusion problems.

✉ Mofdi El-Amrani
mofdi.elamrani@urjc.es

Abdellah El-Kacimi
ab1kacimi@gmail.com

Bassou Khouya
khouya.bassou@um6p.ma

Mohammed Seaid
m.seaid@durham.ac.uk

[1] Departamento Matemáticas Applicada, Ciencia e Ingeniería de Materiales y Tecnología Electrónica, Universidad Rey Juan Carlos, 28933 Móstoles-Madrid, Spain

[2] Laboratory of Modelling and Combinatorial, FP Safi, Cadi Ayyad University, Marrakech, Morocco

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

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

Published online: 06 July 2022

Keywords Bernstein-Bézier discretization · Finite element method · Galerkin-characteristics algorithm · Convection-diffusion problems

1 Introduction

Convection-diffusion equations have been widely used to model many applications in engineering involving advection of scalar quantities such as density, temperature or concentration among others. For instance, convection-diffusion problems have been proposed to study water transport in soils [25], heat transfer of nanofluids [36], and the transport in ferrofluids under rotating magnetic fields [10]. On the other hand, various numerical methods have been developed in recent years to solve the convection-diffusion equations. Most of these computational techniques can be classified into three main categories: (i) Eulerian methods, (ii) Lagrangian methods and (iii) semi-Lagrangian methods. In the framework of finite elements, the most popular Eulerian methods are the streamline upwind Petrov-Galerkin methods [9, 11], Galerkin/least-squares methods [9, 26] and Taylor-Galerkin methods [12, 13]. However, it is well known that these Eulerian methods do not perform very satisfactory in the case of convection-dominated problems unless small time steps and highly refined grids are used in the simulations. In the case of pure convection problems as those considered in this study, these requirements are practically not feasible and may limit the performance of these Eulerian methods. The Lagrangian techniques on the other hand, are 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 trajectory 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 complex convection-diffusion problems. In the semi-Lagrangian methods known also in the framework of finite elements by Galerkin-characteristics, the computational mesh is taken to be fixed to overcome the drawback of the Lagrangian methods while keeping the advantage of the Lagrangian tracking algorithm along the characteristic curves. The main advantage of the Galerkin-characteristics method lies on the fact that the Courant-Friedrichs-Lewy (CFL) condition is highly relaxed compared to its Eulerian counterparts, see for example [15–18, 32, 39, 42]. In addition, the Lagrangian treatment in the Galerkin-characteristics algorithms greatly reduces the time truncation errors in the Eulerian methods, see [7, 8, 15, 38, 41] among others. Thus, the Galerkin-characteristics finite element method has the potential to be more suitable than the Eulerian and Lagrangian methods for convection-diffusion problems on unstructured meshes. Numerical assessment of the conventional Galerkin-characteristics method has been carried out in [16] for convection problems and comparisons to well-established Eulerian methods can also been found in this reference.

In general, most of Galerkin-characteristics 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 operator. For example, to achieve a second-order accuracy in the Galerkin-characteristics 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 Galerkin-characteristics finite element method for convection-diffusion problems decreases as the time step increases at certain range of parameters, see for instance

[17, 18, 20]. 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 method have been shown to achieve fast convergence with low numerical diffusion and dispersion errors for advection-diffusion problems [22, 28]. The *p*-finite element method and *hp*-finite element method where introduced by Babuška, Szabó and their coworkers in the mid-1970s. It was shown that the *hp*-finite element method delivers exponential convergence for elliptic problems with piecewise analytic data, see the survey [6] and further references are therein. A similar performance was also proven for boundary-layer and singularly perturbed problems [33, 37]. A study reported in [5] reveals that the choice of high-order shape functions is critical to the stability and efficiency of the finite element procedure. 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 [43]. Assessment of different high-order shape functions in [35], including Bernstein, Lobatto and Lagrange Gauss-Lobatto polynomials for interior acoustic problems, 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 [19], 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 method. 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 [1, 24, 29]. Especially, it has been shown that Bernstein-Bézier finite elements over simplicial domains, hexahedra and pyramids yield optimal complexity for the standard finite element spaces. In a more recent work [3], 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 Galerkin-characteristics finite element methods to numerically solve convection-dominated problems. This goal is achieved by the implementation of Bernstein-Bézier finite elements for the Galerkin-characteristics method. It should be stressed that combining the Galerkin-characteristics method with the Bernstein-Bézier finite elements, to the best of our knowledge, is reported for the first time. In the context of Galerkin-characteristics 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. To increase the efficiency of the proposed method, we also implement an efficient preconditioned iterative solver using the mass matrix as a preconditioner as proposed in [3]. The key idea behind this considered algorithm lies on the static condensation of the cell-based Degrees of Freedoms (DoFs) on each element and it can be viewed as an extension of the Additive Schwarz method. This results in a *p*-independent uniform bound on the growth of the associated condition number. It should also be noted that in case of the pure convection problem, the resulting linear system in the proposed semi-Lagrangian method consists of the mass matrix which make the proposed preconditioned conjugate gradient solver more convenient than its convection-diffusion counterpart. Furthermore, for the convection-dominated problems the growth of the condition number is relaxed by the viscosity coefficient and therefore, the performance of the considered linear solver is not affected. Numerical results presented in this 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. Introduction of the Bernstein-Bézier finite element discretization is presented in Sect. 2. Section 3 is devoted to the formulation of the Bernstein-Bézier Galerkin-characteristics finite element method for pure convection problems. This section includes the calculation of characteristic curves and the implementation of the computational algorithm. The extension of the method for solving convection-diffusion equations is discussed in Sect. 4. In Sect. 5, we examine the numerical performance of the proposed method using several test examples of convection-diffusion problems including the viscous Burger equation. The proposed Bernstein-Bézier Galerkin-characteristics finite element method is demonstrated to enjoy the expected high-order accuracy as well as efficiency. Concluding remarks are summarized in Sect. 6.

2 Bernstein-Bézier Finite Element Approximation

In order to formulate our Galerkin-characteristics method, the computational domain Ω assumed for simplicity to be a two dimensional bounded and polygonal domain, is first partitioned into non-overlapping triangular finite elements. Let $\widehat{T}$ be the reference element defined by

$$\widehat{T} = \Big\{\boldsymbol{\xi} = (\xi_1, \xi_2) : \quad 0 \le \xi_1 \le 1, \quad 0 \le \xi_2 \le 1 - \xi_1\Big\}, \tag{1}$$

where $\widehat{q}_1 = (0,0)$, $\widehat{q}_2 = (1,0)$ and $\widehat{q}_3 = (0,1)$ are vertices of the reference element as shown in Fig. 1. The barycentric coordinates λ_i $(i = 1, 2, 3)$ relative to the reference element $\widehat{T}$ are defined by

$$\lambda_1(\boldsymbol{\xi}) = \xi_1, \qquad \lambda_2(\boldsymbol{\xi}) = \xi_2 \qquad \text{and} \quad \lambda_3(\boldsymbol{\xi}) = 1 - \xi_1 - \xi_2. \tag{2}$$

The Bernstein-Bézier basis for the space $\mathbb{P}_p(\widehat{T})$ of polynomials of total degree at most p consists of the following shape functions:

- Vertex-based shape functions defined by

$$B^p_{(p,0,0)}(\boldsymbol{\xi}) = \lambda_1^p(\boldsymbol{\xi}), \qquad B^p_{(0,p,0)}(\boldsymbol{\xi}) = \lambda_2^p(\boldsymbol{\xi}), \qquad B^p_{(0,0,p)}(\boldsymbol{\xi}) = \lambda_3^p(\boldsymbol{\xi}),$$

- Edge-based shape functions defined by

$$\begin{aligned} B^p_{(p-k,k,0)}(\boldsymbol{\xi}) &= \binom{p}{k} \lambda_1^{p-k}(\boldsymbol{\xi})\lambda_2^k(\boldsymbol{\xi}), \qquad 1 \le k \le p-1, \\ B^p_{(0,p-k,k)}(\boldsymbol{\xi}) &= \binom{p}{k} \lambda_2^{p-k}(\boldsymbol{\xi})\lambda_3^k(\boldsymbol{\xi}), \qquad 1 \le k \le p-1, \\ B^p_{(k,0,p-k)}(\boldsymbol{\xi}) &= \binom{p}{k} \lambda_3^{p-k}(\boldsymbol{\xi})\lambda_1^k(\boldsymbol{\xi}), \qquad 1 \le k \le p-1. \end{aligned}$$

- Cell-based shape functions defined by

$$B^p_{(i,j,p-i-j)}(\boldsymbol{\xi})=\binom{p}{i+j}\binom{i+j}{i}\lambda_1^i(\boldsymbol{\xi})\lambda_2^j(\boldsymbol{\xi})\lambda_3^{p-i-j}(\boldsymbol{\xi}),\ \ 1\leq i\leq p-2,\ \ 1\leq i+j\leq p-1.$$

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 $\boldsymbol{\beta}\geq\boldsymbol{\alpha}$ *i.e.*, $\beta_i\geq\alpha_i$ for $\leq i\leq 3$, we set $\binom{\boldsymbol{\beta}}{\boldsymbol{\alpha}}=\prod_{i=1}^{3}\binom{\beta_i}{\alpha_i}$. Using these notations, the Bernstein-Bézier shape functions can also be formulated in a simple compact form as

$$B^p_{\boldsymbol{\alpha}}=\binom{p}{\boldsymbol{\alpha}}\prod_{i=1}^{3}\lambda_i^{\alpha_i}, \tag{3}$$

where $\binom{p}{\boldsymbol{\alpha}}=\frac{p!}{\boldsymbol{\alpha}!}$ and $|\boldsymbol{\alpha}|=p$. 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

$$B^p_{\boldsymbol{\alpha}}(\boldsymbol{\xi})B^q_{\boldsymbol{\beta}}(\boldsymbol{\xi})=\frac{\binom{\boldsymbol{\alpha}+\boldsymbol{\beta}}{\boldsymbol{\alpha}}}{\binom{p+q}{p}}B^{p+q}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{\xi}), \tag{4}$$

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

$$\int_{\widehat{T}}B^p_{\boldsymbol{\alpha}}(\boldsymbol{\xi})d\boldsymbol{\xi}=\frac{|\widehat{T}|}{\binom{p+2}{2}} \tag{5}$$

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

$$\nabla B^p_{\boldsymbol{\alpha}}(\boldsymbol{\xi})=p\sum_{k=1}^{3}B^{p-1}_{\boldsymbol{\alpha}-\boldsymbol{e}_k}(\boldsymbol{\xi})\nabla\lambda_k, \tag{6}$$

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 $\widehat{T}$. Moreover, these polynomials have some attractive features such as variation diminishing and monotonicity preserving properties, see for instance [21, 23] and further references can be found therein.

Following the same procedure carried out in the classical finite element methods, each mesh element T in the triangulation $\mathcal{T}_h$ is mapped to the reference element $\widehat{T}$ in which all computations are performed. Here, we use the following reference affine map

$$\boldsymbol{x}=\Phi_T(\boldsymbol{\xi})=\lambda_1(\boldsymbol{\xi})\boldsymbol{q}_1+\lambda_2(\boldsymbol{\xi})\boldsymbol{q}_2+\lambda_3(\boldsymbol{\xi})\boldsymbol{q}_3, \tag{7}$$

where $\boldsymbol{q}_1,\boldsymbol{q}_2$ and $\boldsymbol{q}_3$ are the vertices of the physical element T. The conforming finite element space for the solution that we consider is defined as

$$V_h=\left\{w_h\in C^0(\bar{\Omega}):\quad w_h\Big|_T\circ\Phi_T\in\mathbb{P}_p(\widehat{T}),\quad\forall\, T\in\mathcal{T}_h\right\}. \tag{8}$$

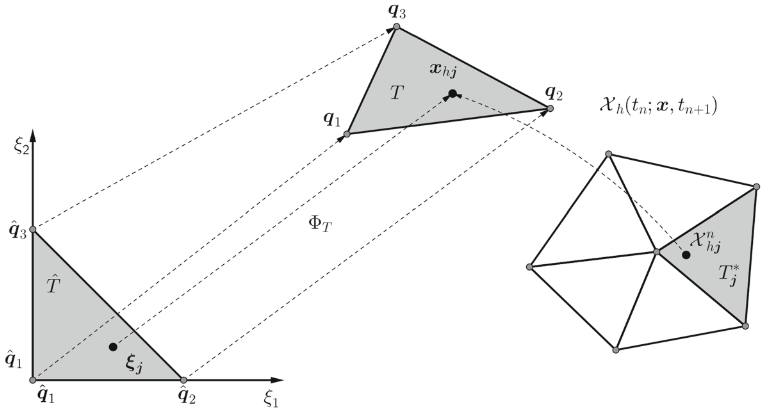

Fig. 1 A schematic diagram showing the main quantities used in the approximation of the departure points. Here, T is a given mesh element, $\boldsymbol{\xi}_j$ is a Stroud quadrature point used in the reference element $\widehat{T}$ and mapped onto $\boldsymbol{x}_{hj}$ in the element T, T^*_j is the host element where the departure point $\boldsymbol{\mathcal{X}}^n_{hj}$ belongs, and Φ_T is the affine one-to-one mapping

For a polynomial degree $p \geq 1$, the number of Degrees of Freedom (DoF) per element is given by

$$N^e = \begin{cases} 3, & \text{if} \quad p = 1, \\ 6, & \text{if} \quad p = 2, \\ 3 + 3(p-1) + \dfrac{(p-2)(p-1)}{2}, & \text{if} \quad p \geq 3. \end{cases} \tag{9}$$

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 [19, 28, 43]. 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 [1, 19] among others.

3 Bernstein-Bézier Finite Element Galerkin-Characteristics Method

To describe the formulation of the proposed Bernstein-Bézier finite element Galerkin-characteristics method, we consider the following two-dimensional Cauchy problem for the convection equation

$$\begin{aligned} \frac{Dc}{Dt} := \frac{\partial c}{\partial t} + \boldsymbol{v}(\boldsymbol{x}, t) \cdot \nabla c = 0, \qquad (\boldsymbol{x}, t) \in \Omega \times (0, T], \\ c(\boldsymbol{x}, 0) = c_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega, \end{aligned} \tag{10}$$

where $\boldsymbol{x} = (x, y)^\top$ is the space variable, $\nabla c = (\frac{\partial c}{\partial x}, \frac{\partial c}{\partial y})^\top$ is the gradient vector, Ω is a spatial bounded domain in $\mathbb{R}^2$ with boundary $\partial\Omega$, and $[0, T]$ is a time interval. Here, $c(\boldsymbol{x}, t)$ denotes the concentration of some species, $\boldsymbol{v}(\boldsymbol{x}, t) = (u(\boldsymbol{x}, t), v(\boldsymbol{x}, t))^\top$ is the velocity field assumed to depend on the solution c as well, and $c_0(\boldsymbol{x})$ is a given initial function. We assume that appropriate boundary conditions are given in such a way the problem is well defined and has a unique solution. Note that $\frac{Dc}{Dt}$ in (10) measures the rate of change of the concentration c following the trajectories of the flow particles. The main idea behind the

Galerkin-characteristics method is to impose a regular grid at the new time level and to backtrack 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.

Next, we divide the time interval $[0, T]$ into N subintervals $[t_n, t_{n+1}]$ with length $\Delta t = t_{n+1} - t_n$ for $n = 0, 1, \dots, N$. We use the notation w^n to denote the value of a generic function w at time t_n. Hence, the characteristic curves associated with the convection problem (10) are the solution of the backward differential equation

$$\begin{aligned} \frac{d\boldsymbol{\mathcal{X}}(\tau; \boldsymbol{x}, t_{n+1})}{d\tau} &= \boldsymbol{v}\big(\boldsymbol{\mathcal{X}}(\tau; \boldsymbol{x}, t_{n+1}), \tau\big), \quad \tau \in [t_n, t_{n+1}], \\ \boldsymbol{\mathcal{X}}(t_{n+1}; t_{n+1}, \boldsymbol{x}) &= \boldsymbol{x}, \end{aligned} \tag{11}$$

where $\boldsymbol{\mathcal{X}}(\tau; \boldsymbol{x}, t_{n+1}) = \big(X(\tau; \boldsymbol{x}, t_{n+1}), Y(\tau; \boldsymbol{x}, t_{n+1})\big)^\top$ is the departure point at time τ of a particle that will arrive at $\boldsymbol{x} = (x, y)^\top$ at time t_{n+1}. Note that the Galerkin-characteristics methods do not follow the flow particles forward in time, as the Lagrangian schemes do, instead they trace backwards the position at time t_n of particles that will reach the points of a fixed mesh at time t_{n+1}, see Fig. 1 for an illustration. By so doing, the Galerkin-characteristics methods avoid the grid distortion difficulties that the conventional Lagrangian methods have. The solution of the differential eq. (11) can be expressed as

$$\boldsymbol{\mathcal{X}}(t_n; \boldsymbol{x}, t_{n+1}) = \boldsymbol{x} - \int_{t_n}^{t_{n+1}} \boldsymbol{v}\left(\boldsymbol{\mathcal{X}}(\tau; \boldsymbol{x}, t_{n+1}), \tau\right)\, d\tau. \tag{12}$$

Hence, integrating the convection eq. (10) along the characteristic curves yields

$$c^{n+1}(\boldsymbol{x}) = c^n\left(\boldsymbol{\mathcal{X}}(t_n; \boldsymbol{x}, t_{n+1})\right). \tag{13}$$

Note that the Bernstein polynomials are not interpolatory by reconstruction *i.e.*, the degrees of freedom associated to the shape functions do not directly lie on the solution evaluated at control points. Hence, the solution at the next time level should be obtained by a weak formulation. Thus, multiplying both sides of equation (13) by a test function $w \in H^1(\Omega)$ and integrating over Ω, it leads to the following weak form

$$\int_\Omega w\; c^{n+1}\, d\boldsymbol{x} = \int_\Omega w\; c^n \circ \boldsymbol{\mathcal{X}}^n\, d\boldsymbol{x}. \tag{14}$$

Here, the finite element solution c_h^{n+1} is sought element-wise at each time step as

$$c_h^{n+1}(\boldsymbol{x}) = \sum_{|\boldsymbol{\alpha}|=p} C_{\boldsymbol{\alpha}}^{n+1} B_{\boldsymbol{\alpha}}^p(\boldsymbol{\xi}), \tag{15}$$

with $\boldsymbol{x} = \Phi_T(\boldsymbol{\xi})$ and $T \in \mathcal{T}_h$. The approximation of the weak form (14) using the conforming finite element space V_h yields the following linear system of algebraic equations

$$\boldsymbol{M}\boldsymbol{C}^{n+1} = \boldsymbol{b}^n, \tag{16}$$

with $\boldsymbol{M}$ is an $n_{\text{ndof}} \times n_{\text{ndof}}$-valued sparse symmetric matrix, $\boldsymbol{b}$ is an n_{ndof}-valued right-hand side column vector and $\boldsymbol{C}$ is n_{ndof}-valued column vector of the unknowns, where n_{ndof} is the total number of degrees of freedom. The entries of the mass element matrix can be written as

$$M_{\boldsymbol{\alpha},\boldsymbol{\beta}}^p = \int_{\widehat{T}} \det(\boldsymbol{J}_T) B_{\boldsymbol{\alpha}}^p(\boldsymbol{\xi}) B_{\boldsymbol{\beta}}^p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}, \tag{17}$$

where $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| = p$ and $\boldsymbol{J}_T = \left(\frac{\mathrm{d}\Phi_T}{\mathrm{d}\boldsymbol{\xi}}\right)^\top$ is the Jacobian matrix. Since the geometry is interpolated using the affine map Φ_T, analytical integration rules as those proposed in [1, 29] can be used for the evaluation of the element mass matrix. Thus, using the proprieties (4) and (5), the integral in (17) can be evaluated as

$$M^p_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \frac{|T|}{|\widehat{T}|} \frac{\binom{\boldsymbol{\alpha}+\boldsymbol{\beta}}{\boldsymbol{\beta}}}{\binom{2p}{p}} \int_{\widehat{T}} B^{2p}_{\boldsymbol{\alpha}+\boldsymbol{\beta}}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi} = \frac{|T|}{|\widehat{T}|} \frac{\binom{\boldsymbol{\alpha}+\boldsymbol{\beta}}{\boldsymbol{\beta}}}{\binom{2p}{p}\binom{2p+2}{2}}. \tag{18}$$

Note that the crucial step in this approach is the evaluation of the right-hand entries in (16) given by

$$b^n_{\boldsymbol{\alpha}} = \int_{\widehat{T}} \det(\boldsymbol{J}_T)\, c^n_h \circ \boldsymbol{\mathcal{X}}^n_h\, B^p_{\boldsymbol{\alpha}}(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}. \tag{19}$$

is evident that, if the above integrals are evaluated exactly then it is easy to show that the Galerkin-characteristics method is unconditionally stable in the L^2-norm, see for instance [15, 40]. In a general framework, this cannot be done and one has to approximate the integrals by numerical integration.

It has shown in [1] 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 $\boldsymbol{t} = (t_1, t_2) \in \widehat{S} = [0, 1]^2$ to the reference triangle $\widehat{T}$ and it can be defined by

$$F(\boldsymbol{t}) = \boldsymbol{\xi} = \Big(t_1, (1 - t_1)t_2\Big)^\top. \tag{20}$$

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

$$B^p_{\boldsymbol{\alpha}}(\boldsymbol{\xi}) = B^p_{\alpha_1}(t_1) B^{p-\alpha_1}_{\alpha_2}(t_2), \tag{21}$$

where $|\boldsymbol{\alpha}| = p$. Therefore, the Bernstein polynomial form (15) becomes

$$c^{n+1}_h(\boldsymbol{x}) = \sum_{\alpha_1=0}^{p} B^p_{\alpha_1}(t_1) \sum_{\alpha_2=0}^{p-\alpha_1} C^{n+1}_{\boldsymbol{\alpha}}\, B^{p-\alpha_1}_{\alpha_2}(t_2), \tag{22}$$

with $\boldsymbol{x} = \Phi_T \circ F(\boldsymbol{t})$ and $T \in \mathcal{T}_h$. Recall the q-point Gauss-Jacobi quadrature defined as

$$\int_0^1 w(t) f(t)\, \mathrm{d}t \approx \sum_{i=0}^{q} w^{(a,b)}_i f(s^{(a,b)}_i), \tag{23}$$

where the weight function $w(t) = (1-t)^a t^b$, with $a, b > -1$, $\{s^{(a,b)}_i\}$ is the set of nodes, and $\{w^{(a,b)}_i\}$ are the weights. Thus, using the Stroud quadrature rule, the relations (20) and (22), the evaluation of the integral (19) yields

$$\begin{aligned} b^n_{\boldsymbol{\alpha}} &= \frac{|T|}{|\widehat{T}|} \int_0^1 \left(\int_0^1 (1-t_1) B^p_{\alpha_1}(t_1) c^n_h \circ \boldsymbol{\mathcal{X}}^n_h\, \mathrm{d}t_1 \right) B^{p-\alpha_1}_{\alpha_2}(t_2)\, \mathrm{d}t_2, \\ &\approx \frac{|T|}{|\widehat{T}|} \sum_{j_2=1}^{q} w^{(0,0)}_{j_2} B^{p-\alpha_1}_{\alpha_2}\left(s^{(0,0)}_{j_2}\right) \left(\sum_{j_1=1}^{q} w^{(1,0)}_{j_1}\, B^p_{\alpha_1}(s^{(1,0)}_{j_1})\, c^n_h \left(\boldsymbol{\mathcal{X}}^n_{hj}\right) \right), \end{aligned} \tag{24}$$

where $\boldsymbol{j} = (j_1, j_2)$. By setting $\boldsymbol{\xi}^*_{\boldsymbol{j}} = \Phi^{-1}_{T^*_{\boldsymbol{j}}}(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}})$, $t^*_{\boldsymbol{j},1} = \xi^*_{\boldsymbol{j},1}$ and $t^*_{\boldsymbol{j},2} = \dfrac{\xi^*_{\boldsymbol{j},2}}{1-\xi^*_{\boldsymbol{j},1}}$, the solution c^n_h can be evaluated at the point $\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}$ as

$$\begin{aligned} c^n_h(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}) &= \sum_{|\boldsymbol{\alpha}|=p} C^n_{\boldsymbol{\alpha}} B^p_{\boldsymbol{\alpha}}\left(\boldsymbol{\xi}^*_{\boldsymbol{j}}\right), \\ &= \sum_{\alpha_1=0}^{p} B^p_{\alpha_1}\left(t^*_{\boldsymbol{j},1}\right) \sum_{\alpha_2=0}^{p-\alpha_1} C^n_{\boldsymbol{\alpha}}\, B^{p-\alpha_1}_{\alpha_2}\left(t^*_{\boldsymbol{j},2}\right). \end{aligned} \tag{25}$$

Note that if $1 - \xi^*_{\boldsymbol{j},1} = 0$ the Duffy transformation degenerates and since in this case $\lambda_1(\xi^*_{\boldsymbol{j},1}) = 1$ and $\lambda_2(\xi^*_{\boldsymbol{j},1}) = \lambda_2(\xi^*_{\boldsymbol{j},1}) = 0$, it follows that $c^n_h(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}) = C^n_{(p,0,0)}$.

The evaluation step (25) requires finding for each arrival point $\boldsymbol{x}_{h\boldsymbol{j}} = \Phi_T(\boldsymbol{\xi}_{\boldsymbol{j}})$, the host element $T^*_{\boldsymbol{j}}$ where the departure point $\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}} = \boldsymbol{\mathcal{X}}_h\left(t_n; \boldsymbol{x}_{h\boldsymbol{j}}, t_{n+1}\right)$ is located, with

$$\xi_{j_1} = s^{(1,0)}_{j_1} \quad \text{and} \quad \xi_{j_2} = (1 - s^{(1,0)}_{j_1}) s^{(0,0)}_{j_2}. \tag{26}$$

At the implementation level, the interpolation of c^n_h at the departure point $\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}$ is performed in the same manner as in [1], by exploiting property (25). Thus, we first evaluate all the univariate Bernstein polynomials $B^p_{\alpha_1}$ and $B^{p-\alpha_1}_{\alpha_2}$, with $\alpha_1 = 0, 1, \ldots, p$ and $\alpha_2 = 0, 1, \ldots, p - \alpha_1$, at the points $t^*_{\boldsymbol{j},1}$ and $t^*_{\boldsymbol{j},2}$, respectively, based on the recursion formula

$$B^{k+1}_{\alpha}(t) = t B^k_{\alpha-1}(t) + (1-t) B^k_{\alpha}(t), \tag{27}$$

where $B^k_{-1} = B^k_{k+1} = 0$ and $B^0_0 = 1$. Note that this procedure requires $\mathcal{O}(p^2)$ floating point operations. Next, for $\alpha_1 = 0, 1, \ldots, p$, we compute the auxiliary coefficients

$$C^*_{\alpha_1} = \sum_{\alpha_2=0}^{p-\alpha_1} C^n_{\boldsymbol{\alpha}}\, B^{p-\alpha_1}_{\alpha_2}(t^*_{\boldsymbol{j},2}), \tag{28}$$

which yields a cost of $\mathcal{O}(p^2)$. Hence, we evaluate $c^n_h(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}})$ as

$$c^n_h(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}) = \sum_{\alpha_1=0}^{p} C^*_{\alpha_1} B^p_{\alpha_1}(t^*_{\boldsymbol{j},1}). \tag{29}$$

Notice that the computational cost of this algorithm per point is of $\mathcal{O}(p^2)$ operations for an arbitrary point. It should also be stressed that De Casteljau's algorithm [31] can be used for evaluating the previous coefficients. However, this results in $\mathcal{O}(p^3)$ operations, unless the given point $\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}$ belongs to an edge of the triangle for which the computational cost will decrease to $\mathcal{O}(p^2)$ operations. Once $c^n_h(\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}})$ is computed for all $j_1, j_2 = 1, \ldots, q$, each entry $b^n_{\boldsymbol{\alpha}}$ given by (24) can be evaluated in $\mathcal{O}(q^2)$ operations. Since the required number q of quadrature points should be chosen such that $q = \mathcal{O}(p)$ to ensure a sufficiently accurate numerical integration, the total cost to set up the element right-hand side is $\mathcal{O}(p^2)$.

In the current study, to compute the departure point $\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}}$ from a given integration point $\boldsymbol{x}_{h\boldsymbol{j}}$, the discrete analogous of (12) is first reformulated as

$$\boldsymbol{\mathcal{X}}^n_{h\boldsymbol{j}} = \boldsymbol{x}_{h\boldsymbol{j}} - \boldsymbol{\delta}_{h\boldsymbol{j}}, \tag{30}$$

where the displacement $\boldsymbol{\delta}_{hj}$ is evaluated using the following iterative procedure based on the Adams-Bashforth method

$$\begin{aligned}\boldsymbol{\delta}_{hj}^{(0)} &= \frac{\Delta t}{2}\Big(\boldsymbol{v}^n\left(\boldsymbol{x}_{hj}\right) - \boldsymbol{v}^{n-1}\left(\boldsymbol{x}_{hj}\right)\Big),\\ \boldsymbol{\delta}_{hj}^{(k+1)} &= \frac{\Delta t}{2}\left(3\boldsymbol{v}^n\left(\boldsymbol{x}_{hj} - \frac{1}{2}\boldsymbol{\delta}_{hj}^{(k)}\right) - \boldsymbol{v}^{n-1}\left(\boldsymbol{x}_{hj} - \frac{1}{2}\boldsymbol{\delta}_{hj}^{(k)}\right)\right), \qquad k = 0, 1, \ldots, \end{aligned} \tag{31}$$

with the velocity fields $\boldsymbol{v}^n\big(\boldsymbol{x}_{hj} - \frac{1}{2}\boldsymbol{\delta}_{hj}^{(k)}\big)$ and $\boldsymbol{v}^{n-1}\big(\boldsymbol{x}_{hj} - \frac{1}{2}\boldsymbol{\delta}_{hj}^{(k)}\big)$ evaluated based on the finite element interpolation on the mesh element where $\boldsymbol{x}_{hj} - \frac{1}{2}\boldsymbol{\delta}_{hj}^{(k)}$ is located. This iterative procedure was first proposed in [41] for the finite difference semi-Lagrangian methods and investigated in [15] for the finite element discretizations. In the present work, the iterations (31) are stopped when the following criteria

$$\frac{\left\|\boldsymbol{\delta}_{hj}^{(k+1)} - \boldsymbol{\delta}_{hj}^{(k)}\right\|}{\left\|\boldsymbol{\delta}_{hj}^{(k)}\right\|} \leq \epsilon, \tag{32}$$

is satisfied for the Euclidean norm $\|\cdot\|$ and a given tolerance ϵ. In general, the departure points do not coincide with the spatial position of a mesh point in the triangular mesh. Therefore, the method used to compute $\boldsymbol{\mathcal{X}}_{hj}^{n}$ should be equipped with a search-locate algorithm to find the host element where such point is located. In our simulations presented in Sect. 5, we have implemented a search-locate algorithm designed in [4] for the semi-Lagrangian methods in unstructured finite element discretizations. In addition, the iterations in (31) were continued until the trajectory changed by less than $\epsilon = 10^{-6}$.

4 Implementation for Convection-Diffusion Problems

In this section we consider convection-diffusion problems reformulated using the total derivative as

$$\begin{aligned}\frac{Dc}{Dt} - \nu\Delta c &= f(\boldsymbol{x}, t), \qquad (\boldsymbol{x}, t) \in \Omega \times (0, T],\\ c(\boldsymbol{x}, 0) &= c_0(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega,\end{aligned} \tag{33}$$

where ν is the diffusion coefficient and $f(\mathbf{x}, t)$ the source term. We assume that eq. (33) is equipped with well defined boundary and initial conditions depending on the problem under study. In the current study, to deal with the diffusion part in the eq. (33) we consider a second-order implicit scheme of Gear type also known in the literature by backward differentiation formula (BDF2). Using the same notations introduced in the previous section, the time semi-discrete form of the eq. (33) reads

$$\frac{3c^{n+1} - 4c^n \circ \boldsymbol{\mathcal{X}}^n + c^{n-1} \circ \boldsymbol{\mathcal{X}}^{n-1}}{2\Delta t} - \nu\Delta c^{n+1} = f^{n+1}, \tag{34}$$

which can be rearranged in a compact form as

$$\frac{3}{2\Delta t}c^{n+1} - \nu\Delta c^{n+1} = F^n, \tag{35}$$

with $F^n = \frac{4}{2\Delta t}c^n \circ \boldsymbol{\mathcal{X}}^n - \frac{1}{2\Delta t}c^{n-1} \circ \boldsymbol{\mathcal{X}}^{n-1} + f^{n+1}$. Notice that to advance the solution c^{n+1} in time in (34), the two solutions c^{n-1} and c^n are required. At time $t = 0$ only one initial condition is provided and to obtain the second condition we use the implicit Euler scheme. Suppose for simplicity purposes, the convection-diffusion problem (33) is supplied with an homogeneous Dirichlet boundary condition on $\partial\Omega$. Then, the discrete weak form of (35) reads as: Find $c_h^{n+1} \in V_h^0 = V_h \cap H_0^1(\Omega)$, such that

$$\frac{3}{2\Delta t}\int_{\Omega} v_h\, c_h^{n+1}\, d\boldsymbol{x} + \nu \int_{\Omega} \nabla v_h \cdot \nabla c_h^{n+1}\, d\boldsymbol{x} = \int_{\Omega} v_h\, F^n\, d\boldsymbol{x}, \quad \forall v_h \in V_h^0, \tag{36}$$

where V_h is the conforming finite element space defined in (8). By virtue of definitions of the finite element operators given above, this reduces to the following linear system of algebraic equations

$$\frac{3}{2\Delta t}\boldsymbol{M}\boldsymbol{C}^{n+1} + \nu \boldsymbol{K}\boldsymbol{C}^{n+1} = \boldsymbol{b}^n, \tag{37}$$

where $\boldsymbol{K}$ is an $n_{\text{ndof}} \times n_{\text{ndof}}$-valued sparse symmetric matrix and $\boldsymbol{b}^n$ is an n_{ndof}-valued right-hand side column vector. It should also noted that since the Bernstein polynomials are only interpolatory at the mesh grid vertices, a numerical procedure is needed to impose nonhomogeneous Dirichlet type boundary condition. 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.

The entries $K_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ of the element stiffness matrix $\boldsymbol{K}$ are given by

$$K_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \int_{\widehat{T}} \det(\boldsymbol{J}_T)\, \left(\boldsymbol{J}_T^{-1}\widehat{\nabla} B_{\boldsymbol{\beta}}^p\right) \cdot \left(\boldsymbol{J}_T^{-1}\widehat{\nabla} B_{\boldsymbol{\alpha}}^p\right)\, \mathrm{d}\boldsymbol{\xi}, \tag{38}$$

and the entries $b_{\boldsymbol{\alpha}}^n$ of the right-hand side column vector $\boldsymbol{b}^n$ are given by

$$b_{\boldsymbol{\alpha}}^n = \int_{\widehat{T}} \det(\boldsymbol{J}_T)\, \left(\frac{4}{2\Delta t}c^n \circ \boldsymbol{\mathcal{X}}^n - \frac{1}{2\Delta t}c^{n-1} \circ \boldsymbol{\mathcal{X}}^{n-1} + f^{n+1}\right)\, B_{\boldsymbol{\alpha}}^p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}. \tag{39}$$

where $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| = p$, $\widehat{\nabla}$ refers to the gradient with respect to the local coordinates $\boldsymbol{\xi}$. Following the same procedure used in the previous section to compute analytically the entries of the elements mass matrix, one can easily verify that the computation of the entries of the element stiffness matrix can be carried out analytically using eq. (6) as

$$K_{\boldsymbol{\alpha},\boldsymbol{\beta}} = p^2 \frac{|T|}{|\widehat{T}|} \sum_{k,l=1}^{3} \left(\boldsymbol{J}_T^{-1}\widehat{\nabla}\lambda_k\right) \cdot \left(\boldsymbol{J}_T^{-1}\widehat{\nabla}\lambda_l\right) M_{\boldsymbol{\alpha}-\boldsymbol{e}_k,\boldsymbol{\beta}-\boldsymbol{e}_l}^{p-1}, \tag{40}$$

where $M_{\boldsymbol{\alpha}-\boldsymbol{e}_k,\boldsymbol{\beta}-\boldsymbol{e}_l}^{p-1}$ is deduced from the closed form (18). In addition, the integral in (39) is computed in the same manner as previously based on the Stroud quadrature and property (22). Here, a rule of $q = p + 2$ quadrature points which is exact if the integrand is a polynomial of degree no higher than $2p + 3$ is adopted.

In summary, the Bernstein-Bézier Galerkin-characteristics finite element method to solve either the convection problem (10) or convection-diffusion problem (33) is carried out in the following steps:

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 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

Algorithm The Bernstein-Bézier Galerkin-characteristics finite element method

```
for each time step do
    for each mesh element $T$ do
        For each Stroud quadrature point $\boldsymbol{x}_{hj}$ calculate the departure points $\boldsymbol{\mathcal{X}}^n_{hj}$ using (30)–(31).
        Identify the mesh element $T^*_{\boldsymbol{j}}$ where the departure point $\boldsymbol{\mathcal{X}}^n_{hj}$ belongs using the search-locate algorithm proposed in [4].
        Evaluate the gridpoint approximations using (25).
        Compute the element right-hand side using (39) and assemble it in the global column vector $\boldsymbol{b}^n$.
    end for
    Compute $\boldsymbol{C}^{n+1}$ by solving the linear system (16) or (37).
    Update the solution $c_h^{n+1}$ element-wise according to (22).
end for
```

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 (16) or (37) by the preconditioned conjugate gradient solver. This yields an efficient method for solving this class of linear systems of algebraic equations for which the preconditioner is implemented at a cost of $\mathcal{O}(p^3)$ operations as discussed in [3]. Here, the additive Schwarz preconditioner is used in the same manner as studied in [3]. The key idea consists in using the relationship between the Bernstein and Jacobi polynomials which makes the passage from the Jacobi to the Bernstein basis functions and vice versa easy without inverting any matrix. These techniques have been proposed in [3] to build a preconditionner for the Bernstein polynomials using the well-established preconditionner introduced in [2] for the Jacobi polynomials but with less computational cost. This gives rise to a preconditioned system for which the condition number is bounded independently of the polynomial order p and the mesh size h.

5 Results and Examples

A number of numerical examples are selected to illustrate the accuracy of the new Galerkin-characteristics method with Bernstein-Bézier finite elements introduced in the above sections. These examples range from a linear passive advection of some initial conditions to a nonlinear viscous Burgers problem. For some of these test examples the analytical solutions are known, so that we can evaluate the relative L^1-error and relative L^2-error at time t_n as

$$L^1\text{-error} = \frac{\displaystyle\int_\Omega \left|c_h^n - c_{\text{exact}}^n\right| \, d\boldsymbol{x}}{\displaystyle\int_\Omega \left|c_{\text{exact}}^n\right| \, d\boldsymbol{x}}, \qquad L^2\text{-error} = \frac{\sqrt{\displaystyle\int_\Omega \left|c_h^n - c_{\text{exact}}^n\right|^2 \, d\boldsymbol{x}}}{\sqrt{\displaystyle\int_\Omega \left|c_{\text{exact}}^n\right|^2 \, d\boldsymbol{x}}}, \tag{41}$$

where c_{exact}^n and c_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 problems (10) and (33) as

$$\text{CFL} = \max_{x,y} \left(\sqrt{|u|^2 + |v|^2}\right) \frac{\Delta t}{h/p}. \tag{42}$$

In all our computations carried out in this section, the resulting linear systems of algebraic equations are solved using the preconditioned conjugate gradient solver and stopping criteria set to $10^{-6}h/p$, which is small enough to guarantee that the algorithm truncation errors dominate the total numerical errors. All the computations were performed on an Intel® Core(TM) i7-7700HQ CPU@2.80GHz with 8 GB of RAM. It should be pointed out that in all our simulations reported in this section, the number iterations in the linear solver do not exceed 30 iterations.

5.1 A Gaussian Pulse Example

In this example we consider the advection-diffusion of a Gaussian pulse in a rotating velocity field widely used in the literature to ascertain the performance of transport schemes, see for example [18, 41]. Thus, the governing equations are of the form (33) with $\boldsymbol{v} = (-\omega y, \omega x)^\top$ and $\omega = 4$. Initial and boundary conditions are taken from the exact solution

$$c_{\text{exact}}(x, y, t) = \frac{\sigma^2}{\sigma^2 + 4\nu t} \exp\left(-\frac{(\bar{x} - x_0)^2 + (\bar{y} - y_0)^2}{\sigma^2 + 4\nu t}\right),$$

where $\bar{x} = x\cos(\omega t) + y\sin(\omega t)$, $\bar{y} = -x\sin(\omega t) + y\cos(\omega t)$, $x_0 = -0.25$, $y_0 = 0$ and $\sigma^2 = 0.002$. The computational domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$ is covered by different uniform finite element meshes and the time period required for one complete rotation is $\frac{\pi}{2}$. From the definition (42), the CFL number associated to this example is $\omega\frac{\sqrt{2}}{2}\frac{\Delta t}{h/p}$ and it is set to different values in our simulations.

The purpose of this test example is to quantify the errors and convergence rates for the proposed Bernstein-Bézier Galerkin-characteristics finite element method. First we consider the case of pure advection for this example corresponding to $\nu = 0$ in (33). In Table 1 we summarize the results obtained for the L^1-error and convergence rate after one revolution using different meshes, polynomial degrees and CFL numbers. In this table we also include the CPU times for each run. In terms of L^1-error, keeping the polynomial degree p fixed and refining the spatial step h results in a substantial decrease in the computed L^1-errors. From the values of convergence rates in Table 1 we 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 CFL, and the order of the Bernstein-Bézier Galerkin-characteristics finite element method remains almost the same for the considered CFL numbers of 2.5, 5 and 10. Notice that to reduce the computational cost, these CFL numbers are chosen as large as possible which yield explicit Eulerian-based methods noncompetitive.

Next we include the physical diffusion in this problem by solving the advection-diffusion of the Gaussian pulse with diffusion coefficient $\nu = 10^{-6}$. As in the previous case, Table 2 summarizes the results obtained for the L^1-error and convergence rate after one revolution using different meshes, polynomial degrees and CFL numbers. It is clear from the results in Table 2 that for the considered diffusion coefficient, the expected convergence rates are preserved in the Bernstein-Bézier Galerkin-characteristics finite element method for the considered CFL numbers of 2.5, 5 and 10. Notice that, to be confined with convection-dominated problems only small values of the diffusion coefficient are accounted for in our simulations. Furthermore, for the considered small values of the diffusion coefficient, the time steps obtained according to the selected CFL numbers are small enough such that the errors associated with the discretization of the advection term are the dominant ones. Therefore, no substantial decrease in the computed errors is achieved when decreasing the CFL number. From the computational results obtained for the advection-diffusion of a Gaussian pulse, one may conclude the following: (i) the Bernstein-Bézier Galerkin-characteristics finite element

Table 1 Results for pure advection of the Gaussian pulse test after one revolution using gradually refined meshes, polynomial degrees and CFL numbers. The CPU times are given in seconds

p	h	CFL = 2.5			CFL = 5			CFL = 10		
		L^1-error	Rate	CPU	L^1-error	Rate	CPU	L^1-Error	Rate	CPU
4	$\frac{1}{4}$	6.697893E-03	—	0.11	6.151320E-03	—	0.05	4.832959E-03	—	0.02
	$\frac{1}{8}$	7.805689E-04	3.101	0.3	5.830570E-04	3.399	0.18	4.610989E-04	3.389	0.1
	$\frac{1}{16}$	1.769447E-05	5.463	0.57	1.715533E-05	5.086	0.39	1.532397E-05	4.911	0.27
6	$\frac{1}{4}$	3.158762E-03	—	2.89	2.901864E-03	—	1.34	2.186185E-03	—	0.56
	$\frac{1}{8}$	6.843947E-05	5.528	8.12	5.457443E-05	5.732	4.24	4.364640E-05	5.646	2.23
	$\frac{1}{16}$	5.456978E-07	6.970	17	5.927810E-07	6.524	8.39	4.721362E-07	6.530	4.36
8	$\frac{1}{4}$	1.086083E-03	—	4.26	1.014439E-03	—	2.51	7.606702E-04	—	1.2
	$\frac{1}{8}$	6.104673E-06	7.475	16.6	5.784217E-06	7.454	8.35	4.574125E-06	7.377	4.5
	$\frac{1}{16}$	1.642417E-08	8.537	40.5	1.760275E-08	8.360	21	1.369221E-08	8.383	10
10	$\frac{1}{4}$	2.977811E-04	—	8.89	2.787392E-04	—	4.43	2.210738E-04	—	2.72
	$\frac{1}{8}$	6.415371E-07	8.858	31	5.892589E-07	8.885	15.6	4.576490E-07	8.916	8.37
	$\frac{1}{16}$	5.228476E-10	10.260	101	4.844669E-10	10.248	51.4	3.782640E-10	10.240	25.3
12	$\frac{1}{4}$	8.361595E-05	—	20	6.775920E-05	—	10	5.898460E-05	—	5.1
	$\frac{1}{8}$	5.947648E-08	10.457	80	5.266340E-08	10.329	41	4.171318E-08	10.465	21
	$\frac{1}{16}$	1.235810E-11	12.232	261	1.124147E-11	12.193	130	8.754072E-12	12.218	67

Table 2 Results for advection-diffusion of the Gaussian pulse test with $\nu = 10^{-6}$ after one revolution using gradually refined meshes, polynomial degrees and CFL numbers. The CPU times are given in seconds

p	h	CFL = 2.5			CFL = 5			CFL = 10		
		L^1-error	Rate	CPU	L^1-error	Rate	CPU	L^1-Error	Rate	CPU
4	$\frac{1}{4}$	9.758258E-03	—	1.1	1.129623E-02	—	0.31	1.578555E-02	—	0.14
	$\frac{1}{8}$	1.073174E-03	3.185	2.8	1.372828E-03	3.041	1.2	1.486636E-03	3.408	0.42
	$\frac{1}{16}$	2.382930E-05	5.493	3.81	2.020783E-05	6.086	1.98	3.343908E-05	5.474	0.81
6	$\frac{1}{4}$	3.583828E-03	—	4.31	3.972312E-03	—	2.15	3.437458E-03	—	1.01
	$\frac{1}{8}$	6.794438E-05	5.721	6.51	8.612120E-05	5.527	3.12	1.082179E-04	4.989	2.59
	$\frac{1}{16}$	9.362504E-07	6.181	22.3	7.642595E-07	6.816	11.2	8.863210E-07	6.932	5.8
8	$\frac{1}{4}$	1.259800E-03	—	33	1.199896E-03	—	3.06	1.276370E-03	—	1.49
	$\frac{1}{8}$	7.655819E-06	7.362	22	7.746590E-06	7.275	10.9	1.013230E-05	6.977	5.3
	$\frac{1}{16}$	2.715505E-08	8.139	68	1.711275E-08	8.822	23.3	2.183772E-08	8.858	11.5
10	$\frac{1}{4}$	3.339670E-04	–	13	3.990726E-04	–	6.81	3.688932E-04	–	3.1
	$\frac{1}{8}$	9.307604E-07	8.487	39	7.485459E-07	9.058	19.3	7.593619E-07	8.924	9.51
	$\frac{1}{16}$	7.037026E-10	10.369	97	5.478460E-10	10.416	49	4.899500E-10	10.598	28
12	$\frac{1}{4}$	7.532651E-05	—	28	9.839621E-05	—	15	8.981801E-05	—	7.2
	$\frac{1}{8}$	8.992725E-08	9.710	63	5.729266E-08	10.746	31	5.472017E-08	10.681	25
	$\frac{1}{16}$	1.573105E-11	12.481	248	1.333939E-11	12.068	125	9.910890E-12	12.431	73

Table 3 Results for the time-dependent Burgers example with $\nu = 10^{-2}$ at time $t = 2$ using gradually refined meshes, polynomial degrees and two CFL numbers

p	h	CFL = 2.5				CFL = 5			
		L^1-error	Rate	L^2-error	Rate	L^1-error	Rate	L^2-error	Rate
4	$\frac{1}{4}$	1.0838E-02	–	2.8508E-02	–	1.1806E-02	–	3.5439E-02	–
	$\frac{1}{8}$	1.0541E-03	3.362	3.2272E-03	3.143	1.0721E-03	3.461	3.2339E-03	3.454
	$\frac{1}{16}$	8.0829E-04	3.705	2.6049E-04	3.631	7.5805E-04	3.822	2.3509E-04	3.728
5	$\frac{1}{4}$	6.3325E-03	–	1.2336E-02	–	6.5782E-03	–	1.9600E-02	–
	$\frac{1}{8}$	3.9441E-04	4.005	1.3553E-03	3.819	4.0017E-04	4.039	1.3649E-03	3.844
	$\frac{1}{16}$	1.6207E-05	4.605	5.7178E-05	4.567	1.1809E-04	4.889	4.0870E-04	4.818
6	$\frac{1}{4}$	6.0445E-03	–	9.5468E-03	–	6.3438E-03	–	1.6631E-02	–
	$\frac{1}{8}$	2.9681E-04	4.348	5.5904E-04	4.094	3.1086E-04	4.351	7.1143E-04	4.164
	$\frac{1}{16}$	5.4354E-06	5.771	1.1542E-05	5.598	7.3908E-05	6.068	1.1556E-05	5.944
7	$\frac{1}{4}$	5.6751E-03	–	7.8822E-03	–	5.3972E-03	–	1.0806E-02	–
	$\frac{1}{8}$	1.2291E-04	5.529	1.8258E-04	5.432	8.6702E-05	5.960	2.1312E-04	5.664
	$\frac{1}{16}$	1.3045E-06	6.558	2.0740E-06	6.460	6.3156E-07	7.101	1.8258E-06	6.867

method highly solve this test problem on coarse meshes, (ii) the convergence rate of the method is not deteriorated when increasing the CFL numbers.

5.2 Time-Dependent Burgers Example

To further quantify the errors for the proposed Bernstein-Bézier Galerkin-characteristics finite element method for time-dependent problems, we solve the following nonlinear Burgers problem

$$\frac{\partial c}{\partial t} + c\frac{\partial c}{\partial x} + c\frac{\partial c}{\partial y} - \nu \Delta c = 0,$$

in the squared domain $\Omega = [0, 2] \times [0, 2]$ subject to initial and boundary conditions obtained from the following exact solution

$$c(x, y, t) = \frac{1}{1 + \exp\left(\dfrac{x + y - t}{2\nu}\right)}.$$

It is clear that the velocity field in this example depends on the time for which time accuracy of the proposed method can be evaluated. A similar example has been considered in [27] using a spectral volume method. In Table 3 we present the results obtained for the L^1-error, L^2-error and convergence rates at time $t = 2$ for $\nu = 10^{-2}$ using different meshes, different polynomial degrees and two CFL numbers CFL = 2.5 and CFL = 5. As in the previous simulations, keeping the polynomial degree p fixed and refining the mesh size h results in a substantial decrease in the computed L^1-error, L^2-error for both selected CFL numbers. The convergence rates in Table 3 confirm the expected order of convergence of the proposed Bernstein-Bézier Galerkin-characteristics finite element method for each selected polynomial degree p. It should be also noted that lower convergence rates have been observed for this

Table 4 Results for the time-dependent Burgers example with $h = \frac{1}{32}$ at time $t = 2$ using different time steps, polynomial degrees and two different viscosity coefficients

p	Δt	$\nu = 10^{-2}$				$\nu = 10^{-3}$			
		L^1-error	Rate	L^2-error	Rate	L^1-error	Rate	L^2-error	Rate
4	4E-02	9.4688E-06	–	1.3164E-03	–	9.6022E-05	–	3.8078E-04	–
	2E-02	2.4329E-06	1.961	3.3705E-04	1.966	2.7837E-05	1.786	1.1533E-04	1.723
	1E-02	6.1947E-07	1.974	8.5564E-05	1.978	7.2198E-06	1.947	3.2167E-05	1.842
5	4E-02	9.4689E-06	–	1.3164E-03	–	9.6392E-05	–	3.8065E-04	–
	2E-02	2.4329E-06	1.961	3.3712E-04	1.965	2.9626E-05	1.702	1.1659E-04	1.707
	1E-02	6.1665E-07	1.980	8.5267E-05	1.983	7.2042E-06	2.040	3.1392E-05	1.893
6	4E-02	9.4694E-06	–	1.3164E-03	–	9.4117E-05	–	3.8059E-04	–
	2E-02	2.4331E-06	1.960	3.3714E-04	1.965	2.6544E-05	1.826	1.0946E-04	1.798
	1E-02	6.1663E-07	1.980	8.5262E-05	1.983	7.1705E-06	1.888	2.9503E-05	1.891
7	4E-02	9.4694E-06	–	1.3164E-03	–	9.4491E-05	–	3.8051E-04	–
	2E-02	2.4332E-06	1.960	3.3715E-04	1.965	2.6688E-05	1.824	1.0885E-04	1.806
	1E-02	6.1658E-07	1.981	8.5259E-05	1.983	7.1413E-06	1.902	2.9019E-05	1.907

test example using CFL = 2.5 than the case using CFL = 5. These features can be attributed to the nonlinear nature of the problem and also to the time dependence of the velocity field involved in the calculation of departure points.

The next idea is to check the convergence rates in time of the proposed Bernstein-Bézier Galerkin-characteristics finite element method for this test example. In doing so, we compute the L^1-error and L^2-error using a fixed mesh with $h = \frac{1}{32}$ and carry out some numerical experiments varying the polynomial degree and the time step Δt. The obtained results at time $t = 2$ for two different diffusion coefficient $\nu = 10^{-2}$ and $\nu = 10^{-3}$ are listed in Table 4. It is clear that decreasing the time step Δt yields a decrease in both L^1-error and L^2-error for all considered polynomial degrees p and diffusion coefficients. A second-order convergence in time is also clearly observed in Table 4 for the considered polynomial degrees p and diffusion coefficients. Note that for this test example, better convergence results are obtained for the simulations with $\nu = 10^{-2}$ than the simulations with $\nu = 10^{-3}$.

5.3 Deformational Flow Example

In this example we solve the deformational flow problem widely used in the literature to examine the performance of Galerkin-characteristics methods, see for example [14, 18, 34]. Here, the problem statement consists of the linear advection eq. (10) in a circular spatial domain centered at $(x_0 = 0, y_0 = 0)$ with radius 4 and equipped with a highly deformational velocity defined by a steady circular vortex with tangential velocity depending on the radius of the vortex as

$$v_t(x, y) = v_0 \text{sech}^2(R) \tanh(R),$$

where $v_0 = 2.58$ and the radial distance $R = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ is the radius of the vortex. Initial and boundary conditions are obtained from the analytical solution

$$c_{\text{exact}}(x, y, t) = -\tanh\left(\frac{y - y_0}{\eta}\cos(\omega t) - \frac{x - x_0}{\eta}\sin(\omega t)\right), \tag{43}$$

with $\omega = \dfrac{v_t}{R}$ is the angular velocity of the circular vortex and η is a parameter controlling the steepness of the solution (43). Here, $\eta = 0.05$ which yields a tight hyperbolic tangent profile in the initial condition that results in a non-smooth solution as the problem is integrated forward in time. Figure 2 illustrates the computational mesh and the initial condition used in the simulations for this test example. The steep gradient at the centerline of the computational domain can be clearly seen in the initial condition. The unstructured mesh shown in Fig. 2 contains 237 elements. This coarse mesh is considered in the simulations to demonstrate the high accuracy of the proposed Bernstein-Bézier Galerkin-characteristics finite element method to solve highly deformational flow problems without need to very fine meshes.

In Fig. 3 we display the results obtained at time $t = 4$ and CFL $= 3$ using different values for the polynomial degree p. We also include the analytical solution in this figure for comparison reasons. For a better insight, Fig. 4 exhibits cross-sections at $x = 0$ and $y = 0$ of the results using the considered polynomial degrees. It is clear that, using $p = 2$ in the Bernstein-Bézier Galerkin-characteristics finite element method, nonphysical oscillations are generated in the obtained results, specially at the center of the spatial domain where the gradient is sharper. Increasing the value of polynomial degree to $p = 4$, the amplitude of these oscillations is reduced but the numerical dissipation is still present in these results. From the same figures we observe a complete absence of these oscillations and numerical diffusion in the results obtained using $p = 8$ and $p = 10$. Observe the good agreement between the

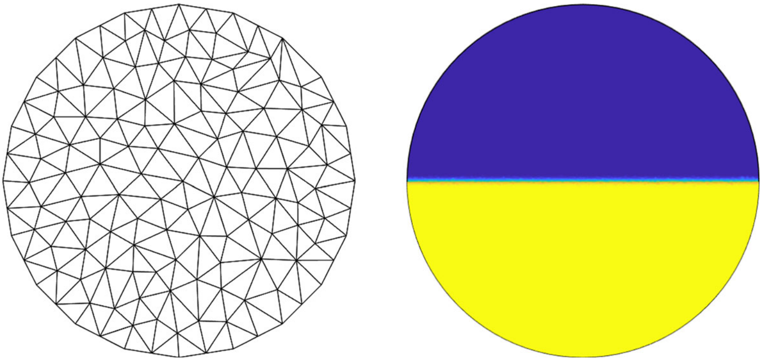

Fig. 2 Computational mesh (left) and initial solution (right) for the deformational flow problem

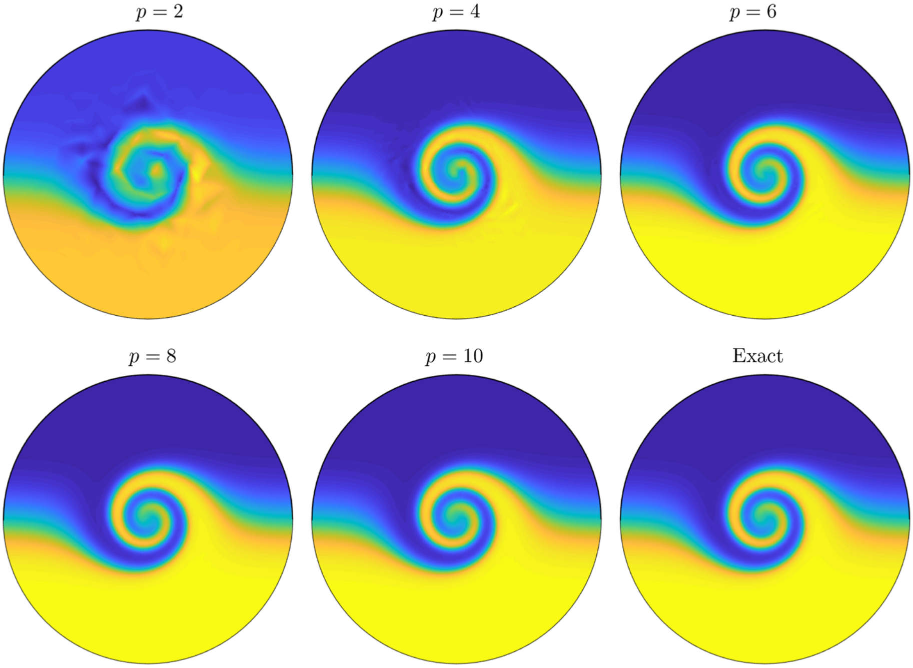

Fig. 3 Results for the deformational flow example at time $t = 4$ using different polynomial degrees $p = 2$, 4, 6, 8, 10 and the exact solution

numerical results and the analytical solutions in Fig. 4 even when using the coarse mesh and the large CFL number. Since the analytical solution for this flow problem is provided in (43), errors can be quantified. In Table 5 we present the L^1-error and L^2-error obtained using three meshes and different polynomial degrees. These unstructured meshes are used such that the number of elements in one mesh is about the double of its coarser counterpart. It is clear that increasing the polynomial order or refining the mesh result in a decrease in both L^1-error and L^2-error. This confirms the high accuracy and the good convergence of the proposed method for solving deformational flow problems. It should be pointed out that the performance of the proposed Bernstein-Bézier Galerkin-characteristics finite element method is very attractive

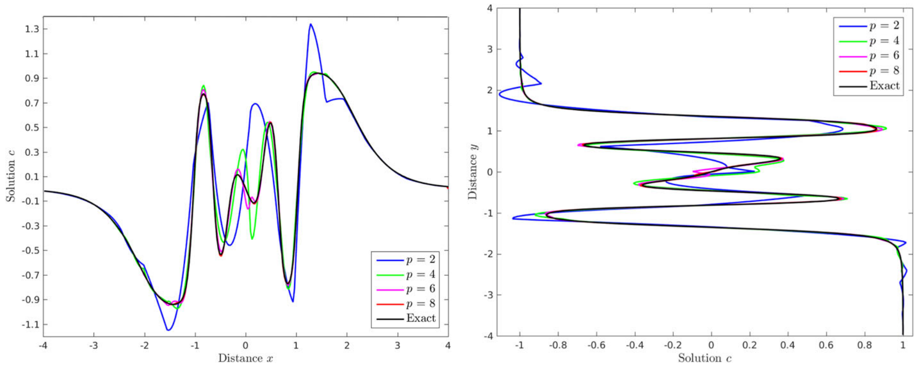

Fig. 4 Cross-sections of the results in Fig. 3 at $y = 0$ (left) and at $x = 0$ (right) using different polynomial degrees

Table 5 Results for the deformational flow problem at time $t = 4$ using different meshes and polynomial degrees

p	# elements	L^1-error	L^2-error
4	116	1.8782E-01	5.9450E-01
	237	1.9592E-02	6.7721E-02
	475	1.2060E-03	7.7143E-03
5	116	1.0075E-01	3.6147E-01
	237	5.3357E-03	1.7799E-02
	475	1.8008E-04	5.8875E-04
6	116	6.7891E-02	2.4142E-01
	237	2.5388E-03	8.5233E-03
	475	3.9449E-05	1.4136E-05
7	116	5.1184E-02	1.5885E-01
	237	1.1232E-03	3.5988E-03
	475	8.1647E-06	2.8567E-05
8	116	3.2184E-02	9.8834E-02
	237	1.2301E-04	1.2543E-03
	475	5.5196E-07	6.0321E-06
9	116	1.1134E-02	7.7831E-02
	237	6.1507E-05	4.9389E-04
	475	1.3799E-07	1.2295E-06
10	116	8.8932E-03	4.6541E-02
	237	2.6327E-05	1.5609E-04
	475	2.7555E-08	1.8767E-07

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.

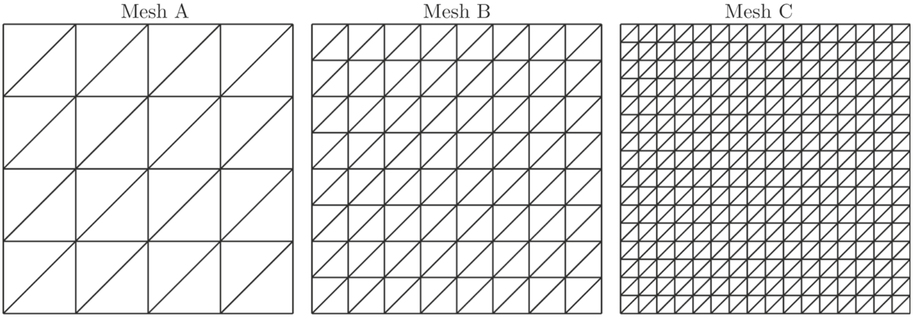

Fig. 5 Computational meshes used for the moving fronts problem

5.4 Moving Fronts Example

We consider the problem of moving fronts modeled by the convection-diffusion eq. (33) equipped with a time-dependent velocity field. Initially, two separate fronts propagate along the main diagonal of the computational domain at different speeds and eventually coalesce into a single front for a longer time. This problem has also been solved in [30] using a family of finite element alternating-direction methods combined with a modified method of characteristics. Here, we solve this problem in a squared domain $\Omega = [0, 1] \times [0, 1]$ with the velocity field defined as

$$\begin{aligned} u(x, y, t) &= \frac{-0.1e^{-A(x,t)} + 0.5e^{-B(x,t)} + e^{-C(x,t)}}{e^{-A(x),t} + e^{-B(x,t)} + e^{-C(x,t)}}, \\ v(x, y, t) &= \frac{-0.1e^{-A(y,t)} + 0.5e^{-B(y,t)} + e^{-C(y,t)}}{e^{-A(y,t)} + e^{-B(y,t)} + e^{-C(y,t)}}, \end{aligned}$$

where

$$\begin{aligned} A(\alpha, t) &= \frac{0.05}{\nu}\left(\alpha - 0.5 + 4.95t\right), \qquad B(\alpha, t) = \frac{0.25}{\nu}\left(\alpha - 0.5 + 0.75t\right), \\ C(t, \alpha) &= \frac{0.50}{\nu}\left(\alpha - 0.375\right), \end{aligned}$$

with $\alpha = x$ for the velocity u and $\alpha = y$ for the velocity v. Initial and boundary conditions are defined by the following exact solution

$$c(x, y, t) = u(x, y, t)v(x, y, t).$$

First we examine the mesh convergence in the proposed Bernstein-Bézier Galerkin-characteristics finite element method. To this end, we consider three structured meshes Mesh A, Mesh B and Mesh C with different element densities as depicted in Fig. 5. Here, the number of triangular elements in Mesh A, Mesh B and Mesh C is 32, 128 and 512, respectively. In Fig. 6 we present 20 equi-distributed contourlines of the solutions obtained at $t = 0.2$ using CFL = 3 and $p = 10$. Here, results are presented for four different diffusion coefficients namely, $\nu = 5 \times 10^{-4}$, $\nu = 10^{-4}$, $\nu = 5 \times 10^{-3}$ and $\nu = 10^{-2}$. The analytical solutions are also included in Fig. 6 for comparison purposes. It is clear that, by decreasing the values of the physical diffusion ν, the convective term becomes dominant and steep internal layers are formed near the vicinity of front lines in the computational domain. The mesh convergence on these results is clearly shown for the Bernstein-Bézier Galerkin-characteristics finite element

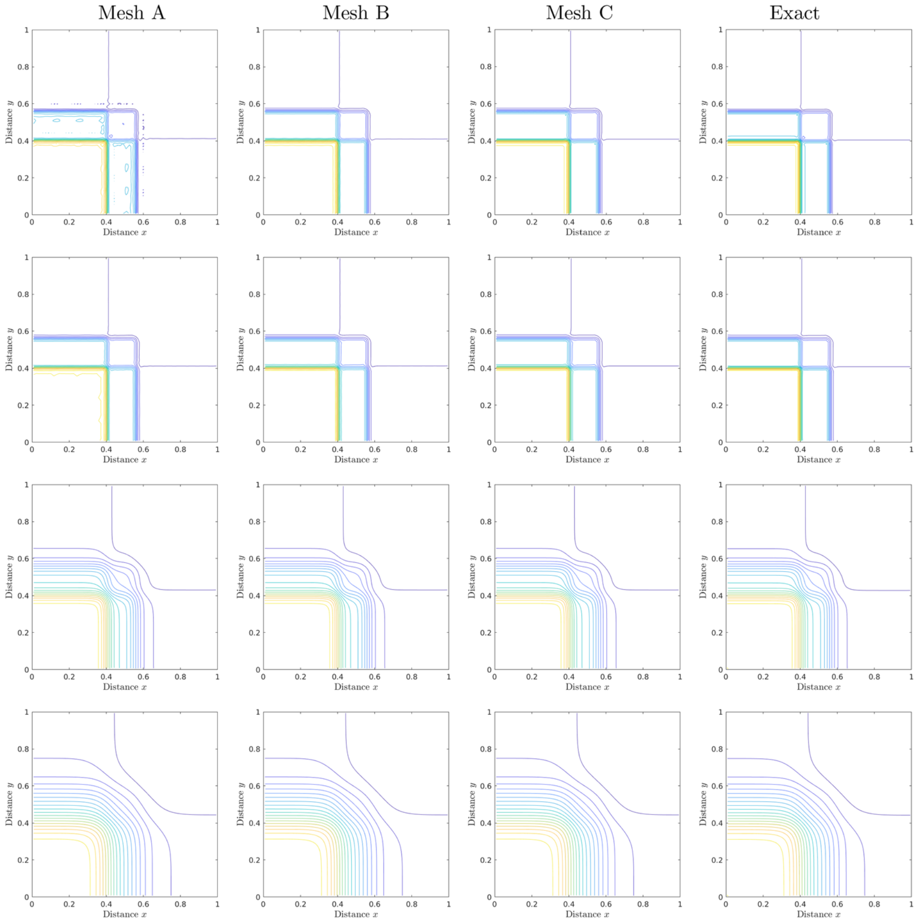

Fig. 6 Results for the moving fronts problem at time $t = 0.2$ using $p = 10$ and $\nu = 5 \times 10^{-4}$ (first row), $\nu = 10^{-3}$ (second row), $\nu = 5 \times 10^{-3}$ (third row) and $\nu = 10^{-2}$ (fourth row) on Mesh A (first column), Mesh B (second column), Mesh C (third column) and exact solution (fourth column)

method using the selected polynomial degree $p = 10$. However, noticeable oscillations are detected for results obtained on the coarse mesh Mesh A. These results give a clear view of the overall transport pattern and effects of the diffusion coefficient ν on the structure of moving fronts in the computational domain. It is worth remarking that the thinning of the internal layers with decreasing ν is evident from these plots and the rate of this thinning is slower for $\nu = 5 \times 10^{-4}$ than for $\nu = 10^{-3}$. These features clearly demonstrate the high accuracy achieved by the proposed Bernstein-Bézier Galerkin-characteristics finite element method for solving moving fronts problems using coarse meshes and large time steps. In addition, compared to the results published for example in [30], it can be seen that our method resolves accurately the solution features and the moving fronts seem to be localized in the correct place in the flow domain without requiring very fine meshes as those reported in [30].

To further qualify the results for these meshes, we plot in Fig. 7 the cross-section results at the main diagonal $y = 1 - x$ using $\nu = 5 \times 10^{-4}$ on the considered three meshes using different polynomial degrees. Those results obtained using $\nu = 10^{-3}$ are depicted in Fig. 8.

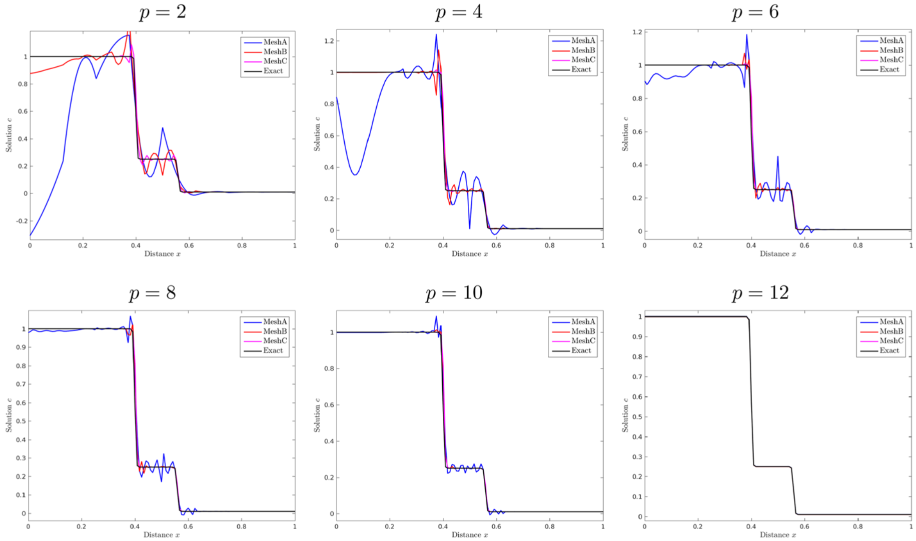

Fig. 7 Cross-sections of the results for the moving fronts problem at the main diagonal $y = 1 - x$ using $\nu = 5 \times 10^{-4}$ and different polynomial degrees on the considered three meshes

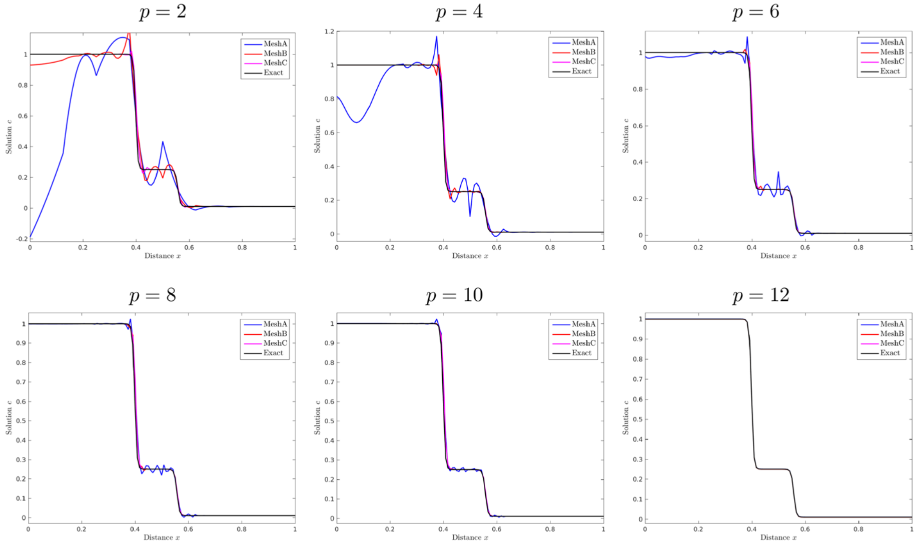

Fig. 8 Same as Fig. 7 but using $\nu = 10^{-3}$

For large values of p, it is clear that the method on the three meshes produces practically identical results. This can be attributed to the high accuracy achieved by Bernstein-Bézier Galerkin-characteristics finite element method using $p = 12$ even on the coarse mesh Mesh A. However, using low polynomial degrees on the meshes Mesh A and Mesh B, nonphysical oscillations are clearly visible in those regions where the solution gradients are steep such as the moving fronts. Apparently, the solution structures are in good agreement with the exact solutions presented in these figures. Needless to say that for convection-dominated

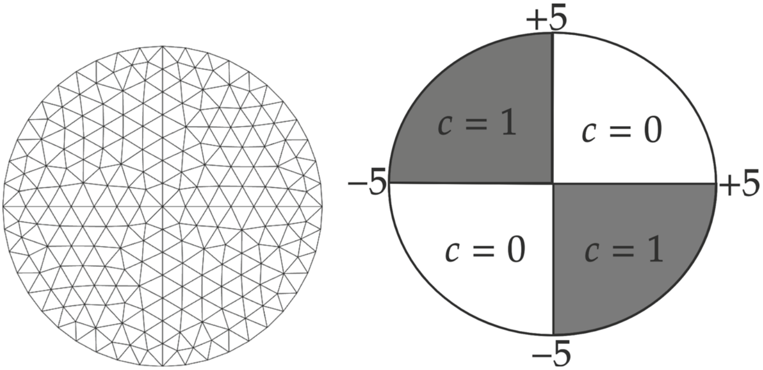

Fig. 9 Computational mesh (left) and initial solution (right) for the viscous Burgers problem. We set $c = 0.5$ on both center lines in the initial solution

simulations, keeping the mesh fixed and changing the polynomial degree, the Bernstein-Bézier Galerkin-characteristics finite element method does not diffuse the fronts or yields spurious oscillations near the steep gradients.

5.5 Viscous Burgers Flow Example

Our final example solves the following viscous Burgers equation which evolves to a highly convective steady-state

$$\frac{\partial c}{\partial t} + \lambda y \left(c - \frac{1}{2} \right) \frac{\partial c}{\partial x} + \lambda x \left(c - \frac{1}{2} \right) \frac{\partial c}{\partial y} - \Delta c = 0,$$

where λ is a constant controlling the magnitude of the nonlinear convective term. The boundary conditions are of Dirichlet type given by the exact steady-state solution

$$c(x, y) = \frac{1}{2} \left(1 - \tanh \left(\frac{\lambda x y}{2} \right) \right).$$

This problem has been solved in a squared domain in [30] using a modified method of characteristics. Here, this problem is solved in a circular domain with radius 5 and centered at the origin. The domain dimensions and the initial conditions used in our computations are illustrated in Fig. 9. We use the Bernstein-Bézier Galerkin-characteristics finite element method to compute the steady-state solutions for three different values of λ namely, $\lambda = 1$, $\lambda = 5$ and $\lambda = 10$. For these steady-state solutions, the time integration process is stopped when the inequality

$$\frac{\| c^{n+1} - c^n \|}{\| c^n \|} \leq \tau, \tag{44}$$

is satisfied. Here $\| \cdot \|$ denotes the L^1-norm and τ is a given tolerance fixed to $10^{-6} \Delta t$ in our computations. It should be pointed out that, the number of iterations to reach this tolerance depends on the values taken by λ such that, for a fixed CFL number, more iterations are required for larger values of λ.

Figure 10 illustrates the obtained results using a mesh of 432 elements and CFL $= 3$. Here, we plot 10 equi-distributed contours of the solutions using two values of the polynomial degree $p = 6$ and $p = 10$. For comparison, we have also included analytical steady-state

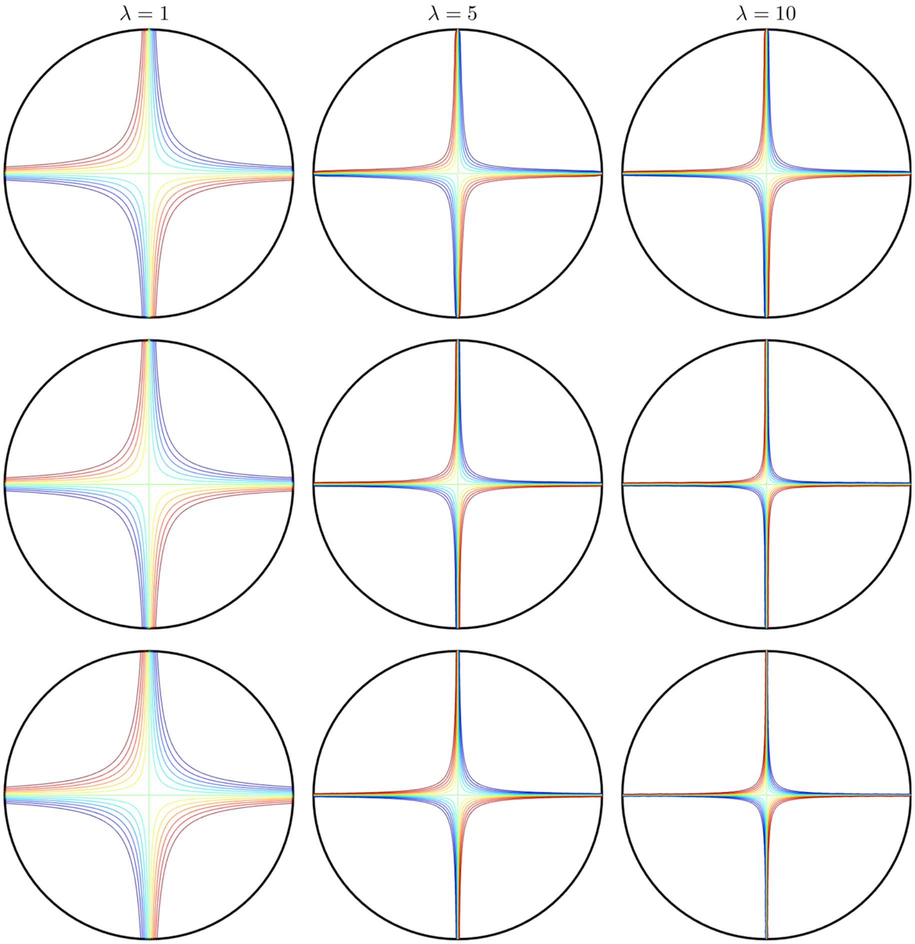

Fig. 10 Results for the viscous Burgers equation with λ = 1 (first column), λ = 5 (second column) and λ = 10 (third column) using $p = 6$ (first row), $p = 10$ (second row) and exact solution (third row)

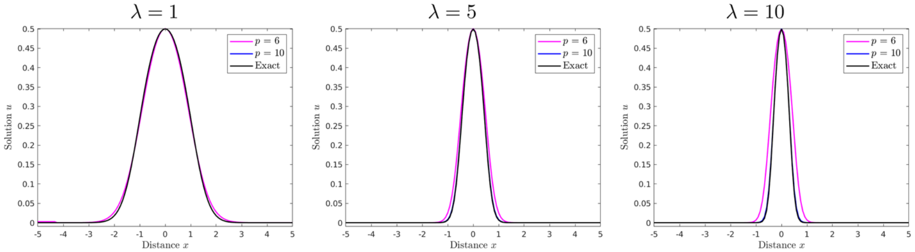

Fig. 11 Cross-sections at $y = x$ of the results for the viscous Burgers problem using λ = 1 (first column), λ = 5 (second column) and λ = 10 (third column)

solutions in this figure. It is clear that, by increasing the values of λ, the convective term becomes dominant and steep internal boundary layers are formed near the vicinity of center lines. For low values of λ, the internal boundary layers are wide and diffuse in the flow domain. As λ increases, the internal boundary layers concentrate and move towards the domain center. It is apparent that the flow structure is in good agreement with the previous work in [30]. These plots give a clear view of the overall flow pattern and the effect of the convection control parameter λ on the structure of steady internal boundary layers in the cavity. It is worth remarking that the thinning of the internal boundary layers with increasing λ is evident from these plots, although the rate of this thinning is slower for $p = 6$ method than for $p = 10$. These features clearly demonstrate the high accuracy achieved by the proposed Bernstein-Bézier Galerkin-characteristics finite element method for solving viscous Burgers problems at steady-state regimes. In addition, compared to the results published for example in [30], it can be seen that our method resolves accurately the flow structures and the boundary layers seem to be localized in the correct place in the flow domain.

For visualizing the comparisons, we display in Fig. 11 a cross-section at the main diagonal $y = x$ using the selected values of λ and p. For $\lambda = 1$, it is clear that the results obtained using $p = 6$ and $p = 10$ produce practically identical results. This can be attributed to the large physical diffusion presented in the problem. However, increasing the value of λ to 5 and 10 the results computed using $p = 10$ are more accurate than those computed using $p = 6$. Apparently, By using $p = 10$, high resolution is achieved in those regions where the flow gradients are steep such as the moving fronts. Comparing the results obtained using the considered values of p, it is clear that using $p = 6$ produces diffusive solutions resulting in smearing the shocks. On the other hand, this numerical diffusion has remarkably been reduced in the results computed using $p = 10$.

6 Conclusions

In this paper, we have presented a Bernstein-Bézier Galerkin-characteristics finite element method for solving convection-diffusion equations on unstructured meshes. The proposed solver inherits the advantages of semi-Lagrangian integration (unconditional stability and reduction of time truncation errors) while preserving high-order accuracy. Moreover, this implementation simplifies the computational complexity of the Bernstein-Bézier Galerkin-characteristics solver for a fixed polynomial approximation during the time integration process. To increase the performance of the proposed method the element-level static condensation is implemented. The favorable performance of the developed algorithm is demonstrated using a series of numerical examples including the viscous Burgers equation. The results obtained for these examples show that the Bernstein-Bézier Galerkin-characteristics finite element method has the advantage of requiring less computational resources for the integration of convection-dominated problems than an Eulerian-based finite element methods, typical of those widely used in Eulerian algorithms for convection-dominated problems. This fact, as well as its favorable high-order accuracy and strong stability properties, make it an attractive alternative for convection solvers based on Galerkin-characteristic techniques. Finally, the method presented in this paper is not appropriate for nonlinear invscid equations with strong shocks. However, using the idea of the limiting techniques studied in [17], and following the arguments of Sect. 3, it is possible to reconstruct a shock-preserving Bernstein-Bézier Galerkin-characteristics finite element method for nonlinear equations. Results on these schemes will be reported in the near future. Future work will also concentrate on extending these methods for the incompressible viscous flows in three space dimensions using unstructured meshes and developing highly efficient preconditioned iterative solvers for the associated linear systems.

Acknowledgements Financial support provided by the project of Consejo Superior de Investigaciones Científicas (CSIC) under the contract M MTM2017-89423-P is gratefully acknowledged.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

References

1. Ainsworth, M., Andriamaro, G., Davydov, O.: Bernstein-Bézier finite elements of arbitrary order and optimal assembly procedures. SIAM J. on Sci. Comput. **33**, 3087–3109 (2011)
2. Ainsworth, M., Jiang, S.: Preconditioning the mass matrix for high order finite element approximation on triangles. SIAM J. on Numerical Anal. **57**(1), 355–377 (2019)
3. Ainsworth, M., Jiang, S., Sanchéz, M.A.: An $\mathcal{O}(p^3)$ hp-version fem in two dimensions: Preconditioning and post-processing. Comput. Methods Appl. Mech. Engrg **350**, 766–802 (2019)
4. Allievi, A., Bermejo, R.: A generalized particle search-locate algorithm for arbitrary grids. J. Comp. Phys. **132**, 157–166 (1992)
5. Babuška, I., Griebel, M., Pitkaränta, J.: The problem of selecting the shape functions for p-type elements. Int. J. for Numerical Methods in Eng. **28**, 1891–1908 (1986)
6. Babuška, I., Suri, M.: The p and hp versions of the finite element method, basic principles and properties. SIAM Rev. **36**, 578–632 (1994)
7. Bermúdez, A., Nogueiras, M.R., Vázquez, C.: Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. Part I: Time discretization. SIAM J. Numer. Anal. **44**, 1829–1853 (2006)
8. Bermúdez, A., Nogueiras, M.R., Vázquez, C.: Numerical analysis of convection-diffusion-reaction problems with higher order characteristics/finite elements. Part II: Fully discretized scheme and quadrature formulas. SIAM J. Numer. Anal. **44**, 1854–1876 (2006)
9. Bertrand, F., Demkowicz, L., Gopalakrishnan, J., Heuer, N.: Recent Advances in Least-Squares and Discontinuous Petrov-Galerkin Finite Element Methods. Comput. Methods in Appl. Math. **19**(3), 395–397 (2019)
10. Boroun, S.H., Larachi, F.: Anomalous anisotropic transport of scalars in dilute ferrofluids under uniform rotating magnetic fields-mixing time measurements and ferrohydrodynamic simulations. Chem.l Eng. J. **380**, 122504 (2020)
11. Brooks, A.N., Hughes, T.J.R.: Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Computer methods in app. mechanics and eng. **32**(1–3), 199–259 (1982)
12. Carew, E.O.A., Townsend, P., Webster, M.F.: A Taylor-Petrov-Galerkin algorithm for viscoelastic flow. J. of non-newtonian fluid mechanics **50**(2–3), 253–287 (1993)
13. Donea, J.: A Taylor-Galerkin method for convective transport problems. Int. J. for Numerical Methods in Eng. **20**(1), 101–119 (1984)
14. Doswell, C.A.: A kinematic analysis of frontogenesis associated with a nondivergent vortex. J. of Atmospheric Sci. **41**(7), 1242–1248 (1984)
15. El-Amrani, M., Seaid, M.: Convergence and stability of finite element modified method of characteristics for the incompressible Navier-Stokes equations. J. Numer. Math. **15**, 101–135 (2007)

16. El-Amrani, M., Seaid, M.: Numerical simulation of natural and mixed convection flows by Galerkin-characteristics method. Int. J. Num. Meth. Fluids. **53**, 1819–1845 (2007)
17. El-Amrani, M., Seaid, M.: An essentially non-oscillatory semi-Lagrangian method for tidal flow simulations. Int. j. for numerical methods in eng. **81**, 805–834 (2010)
18. El-Amrani, M., Seaid, M.: A finite element semi-Lagrangian method with L2 interpolation. Int. J. for numerical methods in eng. **90**, 1485–1507 (2012)
19. El Kacimi, A., Laghrouche, O., Shadi, M., Trevelyan, J.: Bernstein-Bézier based finite elements for efficient solution of short wave problems. Comput. Methods Appl. Mech. Engrg. **343**, 166–185 (2019)
20. Falcone, M., Ferretti, R.: Convergence analysis for a class of high-order semi-Lagrangian advection schemes. SIAM J. on Numerical Anal. **35**(3), 909–940 (1998)
21. Floater, M.S., Pena, J.M.: Monotonicity preservation on triangles. Math. of Comput. **69**, 1505–1519 (2000)
22. Giraldo, F.X.: The lagrange-Galerkin spectral element method on unstructured quadrilateral grids. J. of Comp. Phys. **147**, 114–146 (1998)
23. Goodman, T.N.T.: Variation diminishing properties of Bernstein polynomials on triangles. J. of Approx. Theory **50**, 111–126 (1987)
24. Hackemack, M.W.: Discontinuous Galerkin solutions for elliptic problems on polygonal grids using arbitrary-order Bernstein-Bézier functions. J. of Comput. Phys. **437**, 110293 (2021)
25. Hedayati-Dezfooli, M., Leong, W.H.: An experimental study of coupled heat and moisture transfer in soils at high temperature conditions for a medium coarse soil. Int. J. of Heat and Mass Transfer **137**, 372–389 (2019)
26. Hughes, T., Jr., Franca, L.P., Hulbert, G.M.: A new finite element formulation for computational fluid dynamics: Viii. The Galerkin/least-squares method for advective-diffusive equations. Computer methods in appl. mechanics and eng. **73**(2), 173–189 (1989)
27. Kannan, R., Wang, Z.J.: A high order spectral volume solution to the Burgers' equation using the Hopf-Cole transformation. Int. j. for numerical methods in fluids **69**(4), 781–801 (2012)
28. Karniadakis, G.E., Sherwin, S.J.: Spectral/hp element methods for comput. fluid dynamics. Oxford University Press, Numerical Mathematics and Scientific Computation (2005)
29. Kirby, R.C., Thinh, K.T.: Fast simplicial quadrature-based finite element operators using Bernstein polynomials. Numerische Mathematik **121**, 261–279 (2012)
30. Krisnamachari, S.V., Hayes, L.J., Russel, T.F.: A finite element alternating-direction method combined with a modified method of characteristics for convection-diffusion problems. SIAM J. Numer. Anal. **26**, 1462–1473 (1989)
31. Lai, M.J., Schumaker, L.L.: Spline Functions on Triangulations, Encyclopedia Math. Appl., vol. 110. Cambridge University Press, Cambridge, England (2007)
32. Lee, D., Lowrie, R., Petersen, M., Ringler, T., Hecht, M.: A high order characteristic discontinuous Galerkin scheme for advection on unstructured meshes. J. of Comput. Phys. **324**, 289–302 (2016)
33. Melenk, J.M.: hp-finite element methods for for singular perturbations. Vol. 1796 of Lecture Notes in Mathematics. Springer-Verlag (2002)
34. Nair, R., Côté, J., Staniforth, A.: Monotonic cascade interpolation for Galerkin-characteristics advection. Quart. J. Roy. Met. Soc. **125**, 197–212 (1999)
35. Petersen, S., Dreyer, D., Estorff, V.O.: Assessment of finite and spectral element shape functions for efficient iterative simulations of interior acoustics. Comput. Methods Appl. Mech. Engrg **195**, 6463–6478 (2006)
36. Ryzhkov, I.I., Minakov, A.V.: The effect of nanoparticle diffusion and thermophoresis on convective heat transfer of nanofluid in a circular tube. Int. J. of Heat and Mass Transfer **77**, 956–969 (2014)
37. Schwab, C., Suri, M.: The p and hp versions of the finite element method for problems with boundary layers. Math. Comp. **65**, 1403–1429 (1996)
38. Seaid, M.: Semi-lagrangian integration schemes for viscous incompressible flows. Comp. Methods in App. Math. **4**, 392–409 (2002)
39. Simon, K., Behrens, J.: Semi-lagrangian subgrid reconstruction for advection-dominant multiscale problems with rough data. J. of Sci. Comput. **87**(2), 1–33 (2021)
40. Süli, E.: Convergence and nonlinear stability of the lagrange-Galerkin method for the navier-stokes equations. Numer. Math. **53**, 1025–1039 (1988)
41. Temperton, C., Staniforth, A.: An efficient two-time-level Galerkin-characteristics semi-implicit integration scheme. Quart. J. Roy. Meteor. Soc. **113**, 1025–1039 (1987)

42. Tumolo, G., Bonaventura, L.: A semi-implicit, semi-Lagrangian discontinuous Galerkin framework for adaptive numerical weather prediction. Quarterly J. of the Royal Meteorological Soc. **141**(692), 2582–2601 (2015)
43. Šolín, P., Segeth, K., Doležel, I.: Higher-Order Finite Element Methods. Chapman & Hall, New York (2003)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.