HAL
open science

A stabilized Powell-Sabin finite-element method for the 2D Euler equations in supersonic regime

Giorgio Giorgiani, Hervé Guillard, Boniface Nkonga, Eric Serre

To cite this version:

Giorgio Giorgiani, Hervé Guillard, Boniface Nkonga, Eric Serre. A stabilized Powell-Sabin finite-element method for the 2D Euler equations in supersonic regime. [Research Report] INRIA Sophia Antipolis - Méditerranée. 2017, pp.1-31. hal-01638140

HAL Id: hal-01638140

https://inria.hal.science/hal-01638140

Submitted on 19 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A stabilized Powell-Sabin finite-element method for the 2D Euler equations in supersonic regime

Giorgio Giorgiani[a,*], Hervé Guillard[b], Boniface Nkonga[b], Eric Serre[a]

[a] *Aix-Marseille Univ., CNRS, Centrale Marseille, M2P2 Marseille, France*
[b] *INRIA Sophia Antipolis, 2004 Route des Lucioles, 06902 Valbonne*

Abstract

In this paper is presented a Powell-Sabin finite-elements scheme (PS-FEM) for the solution of the 2D Euler equations in supersonic regime. The spatial discretization is based on PS splines, that are piecewise quadratic polynomials with a global C^1 continuity, defined on conforming triangulations.

Some geometrical issues related the practical construction of the PS elements are discussed, in particular, the generation of the control triangles and the imposition of the boundary conditions. A stabilized formulation is considered, and a novel shock-capturing technique in the context of continuous finite-elements is proposed to reduce oscillations around the discontinuity, and compared with the classic technique proposed by Tedzuyar [1]. The code is verified using manufactured solutions and validated using two challenging numerical examples, which allows to evaluate the performance of the PS discretization in capturing the shocks.

Keywords: Shock capturing, high-order, C^1 approximations, splines, CFD, compressible flow.

*Corresponding author

Email addresses: `giorgio.giorgiani@univ-amu.fr` (Giorgio Giorgiani), `herve.guillard@inria.fr` (Hervé Guillard), `boniface.nkonga@inria.fr` (Boniface Nkonga), `eric.serr@univ-amu.fr` (Eric Serre)

1. Introduction

The use of an artificial viscosity to capture discontinuous solutions started in the 1950s with the pioneering work of von Neumann and Richtmyer [2], but its application in the framework of high-order schemes is relatively new. Most of the works focus on finite volumes (WENO techniques [3]) or discontinuous Galerkin (DG) methods [4, 5, 6, 7]. However in many physical applications, such as magnetohydrodynamics, shells analysis and vibrations, numerical methods with high degrees of regularity are of interest. In particular, finite-element methods based on spline shape functions have raised a particular interest in the recent years. On one hand, C^1-spline methods allow to discretize higher-order derivatives (that is, derivatives of order superior to the second), and on the other hand, they provide accurate geometrical representation of the computational domain. Moreover, it has also been shown recently that the additional global smoothness of the spline interpolant introduces stability to the numerical solution for highly convective equations, see [8] and [9], and also for turbulence computations, see [10]. Hence, finite-element methods based on spline shape functions can be advantageous also in the context of fluid-dynamic problems.

Nowadays, splines are extensively used in the graphical design industry to create smooth surfaces. Their success comes from very attractive features, like compactness, ability to represent curves and surfaces with an arbitrary level of regularity by simply increasing the polynomial degree of the basis, and their locality allowing local modifications by simply moving a control point and keeping the rest of the curve undisturbed. Spline curves are usually represented as a linear combination of basis functions, called *B-splines*. The basis functions of a given order p are defined with a recursive relation starting with piecewise constant basis functions (that is, $p = 0$),

$$B_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases}$$

$$B_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} B_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} B_{i+1,p-1}(\xi),$$

where the points ξ_i are the knots of the spline, $i = 1, 2, \ldots, n+p+1$, and n is the number of basis functions. Hence, the spline curve expressed as a combination of B-splines is

$$S(\xi) = \sum_{i=1}^{n} B_{i,p} \mathbf{P}_i, \tag{1}$$

where $\mathbf{P}_i$ are the control points of the spline.

Starting from the one-dimensional case (1), B-splines can be extended to higher dimensions through a tensor product representation. However, tensor product B-splines are restricted to structured rectangular meshes. The refinement procedure with tensor B-splines relies on the insertion of knots, leading hence to a global modification of the domain discretization. Thus, no local refinement is possible.

Various solutions are proposed in literature to overcome this limitation, as for example, the introduction of *T-splines* in [11]. Our work focus on Powell-Sabin (PS) splines, which are bivariate splines defined on irregular triangulations, [12, 13, 8, 14]. PS splines are more precisely piecewise quadratic polynomial with C^1 continuity, defined on an unstructured triangulation of the domain. Compared to tensor product B-splines, PS splines allow a straightforward adaptive refinement of the mesh, which is a key ingredient in the simulation of anisotropic equations. Moreover, all the unknowns are located on the nodes of the triangulation, meaning that there are no unknowns on the element faces or in the interior of the elements. This is not the case for other C^1 interpolation techniques, such as, for example, the Clough-Tocher elements. The stencil is consequently the same for each node of the mesh leading to a linear system matrix with a regular block shape. This facilitates the implementation aspects of the code, enhances the parallel performance and improves the efficiency of the linear system solution with the use of specific matrix-vector multiplication algorithms.

In this work, a PS finite-element method is presented for the solution of the two dimensional (2D) compressible Euler equation in supersonic regime.

Numerical dissipation in the stream-wise direction only is added using a SUPG stabilization technique [15]. Oscillations around the discontinuities are damped by adding a shock-capturing term. Several techniques are considered for the choice of the stabilization terms and a comparison is carried out on two classical test cases of literature. In particular, the well-known shock-capturing term proposed by Tedzuyar [1] in the context of continuous FEM is considered, and compared to a new formulation inspired by the work of Bassi [5] on DG methods.

The outline of the paper is as follows. Section 2 introduces the PS splines and their representation. Section 3 deals with the geometrical and mathematical tools to define the PS elements, and in particular the generation of the shape functions. In Section 4, the PS-FEM is presented for the solution of the 2D Euler equations, in particular, a detailed discussion is given on the imposition of the boundary conditions. The SUPG and shock-capturing terms are described in 5. The code is verified using a manufactured solution in 6, that allows to check the theoretical third-order convergence in space. Finally in 7, the double Mach reflection problem and the supersonic flow past a forward facing step are used to validate our code comparing with reference solutions of literature, and also to evaluate the performance of the new shock-capturing technique.

2. Powell-Sabin splines definition on a triangulation of a polygonal domain

To obtain an interpolant of a generic function with C^1 continuity, we start from the definition of bivariate polynomials on a triangle. Considering a single triangle Ω_k of vertices $\mathbf{V}_i = (x_i, y_i)$, with $i = 1, 2, 3$, any bivariate polynomial $p(x, y)$ of degree ≤ 2 in the space $\Pi_2 = \{\sum_{i=0}^{2} \sum_{j=0}^{2-i} a_{i,j} x^i y^j, a_{i,j} \in \mathbb{R}\}$ can be written in the Bernstein-Bézier representation as

$$p(x, y) := p(\xi) = \sum_{\substack{i+j+k=2 \\ i,j,k \geq 0}} b_{i,j,k} B^2_{i,j,k}(\xi(x, y)), \tag{2}$$

where $\xi = (\xi_1, \xi_2, \xi_3)$ are the barycentric coordinates of a point $(x, y) \in \mathbb{R}^2$ and $B^2_{i,j,k}(\xi)$ are the Bernstein polynomials of degree 2 in Ω_k, that are

$$B^2_{i,j,k} = \frac{2!}{i!j!k!}\xi_1^i \xi_2^j \xi_3^k.$$

The set of Bernstein polynomials $B^2_{i,j,k}(\xi)$ defines a basis of the polynomials space Π_2, see [16]. Hence, the second order polynomial $p(x, y)$ is uniquely defined by the six coefficients $b_{i,j,k}$, which are called the Bézier ordinates with respect to the triangle Ω_k[1].

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain with boundary $\partial\Omega$ and $\mathcal{T}$ be a conforming triangulation of Ω with vertices $\mathbf{V}_l$ $(l = 1, \ldots, N_v)$ and elements Ω_e $(e = 1, \ldots, N_e)$, we consider the following interpolation problem: find $s(x, y) \in \mathcal{S}^2_1$ such that,

$$s(\mathbf{V}_l) = f_l, \quad \frac{\partial s}{\partial x}(\mathbf{V}_l) = f_{xl}, \quad \frac{\partial s}{\partial y}(\mathbf{V}_l) = f_{yl}, \quad l = 1, \ldots, N_v, \tag{3}$$

(f_l, f_{xl}, f_{yl}), $l = 1, \ldots, N_v$ being any given set of triples, and $\mathcal{S}^2_1$ being the function space of piecewise quadratic polynomials on $\mathcal{T}$.

Problem (3) cannot be solved directly because it requires the imposition of nine parameters to determine the second-order polynomial on each triangle, while only six coefficients are available, see equation (2). A way to solve the interpolation problem (3) is to subdivide each triangle in sub-triangles, and to define the interpolant in the refined triangulation. One of the solutions proposed by Powell and Sabin in [12] is based on the subdivision of each triangle in $\mathcal{T}$ into six smaller triangles (*PS6-split*), see figure 1. Hence, the relations expressed in (3) are only imposed on the vertices of the original triangulation, while on in the other nodes is only imposed the C_1 continuity of the interpolating function. More details can be found in [12].

The so called Powell-Sabin refinement of $\mathcal{T}$ is denoted with $\mathcal{T}^*$. The procedure to define $\mathcal{T}^*$ is described in [8] and reads as follows.

[1]Note that (2) can be expanded as $p(\xi) = b_{2,0,0}\xi_1^2 + b_{0,2,0}\xi_2^2 + b_{0,0,2}\xi_3^2 + 2b_{1,1,0}\xi_1\xi_2 + 2b_{1,0,1}\xi_1\xi_3 + 2b_{0,1,1}\xi_2\xi_3$.

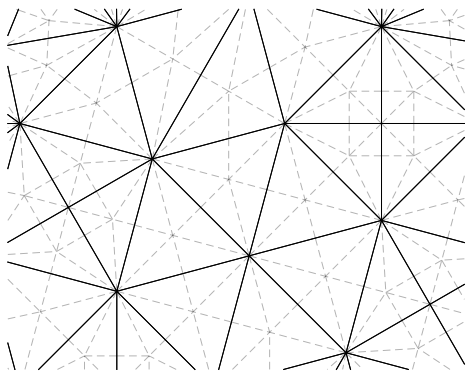


Figure 1: Powell-Sabin refinement of an unstructured mesh.

- Select a split point $\mathbf{C}_k$ inside each triangle Ω_k and connect it to the three vertices of Ω_k.

- For each pair of triangles Ω_p and Ω_q with a common edge, connect the two points $\mathbf{C}_p$ and $\mathbf{C}_q$. If Ω_p is a boundary triangle, connect also $\mathbf{C}_p$ to an arbitrary point on the boundary edge.

The PS refinement defines also a set of points associated to each vertex and called *PS points*. The PS points associated to the vertex $\mathbf{V}_l$ are defined as the midpoints of all the edges of the PS refinement containing $\mathbf{V}_l$, plus the point $\mathbf{V}_l$ itself. These PS points are crucial for the definition of the shape functions as explained in the next section.

Having defined the PS refinement $\mathcal{T}^*$, the function space of piecewise quadratic polynomials with C^1 continuity can be denoted as $S_2^1(\mathcal{T}^*)$. Each function in $S_2^1(\mathcal{T}^*)$ is uniquely defined by its values and derivatives at the vertices of $\mathcal{T}$. Thus the dimension of the function space $S_2^1(\mathcal{T}^*)$ is equal to $3N_v$.

3. Powell-Sabin finite-elements

In order to derive a finite-element scheme based on PS splines, one needs to define the shape functions in the space $S_2^1(\mathcal{T}^*)$. It is thus necessary to express each element $s(x, y) \in S_2^1(\mathcal{T}^*)$ as a linear combination of PS spline

basis functions,

$$s(x,y) = \sum_{l=1}^{N_v} \sum_{r=1}^{3} c_{l,r} B_l^{(r)}(x,y),$$

where the functions $B_l^{(r)}$ are called *Powell-Sabin B-splines*, and $c_{l,r}$ are the coefficients of the representation.

Each B-spline $B_l^{(r)}(x,y)$ can be seen as the solution of the interpolation problem (3) with all $(f_i, f_{xi}, f_{yi}) = 0$, except for $(f_l, f_{xl}, f_{yl}) = (\alpha, \beta, \gamma) \neq 0$. The quantity (α, β, γ) is called a *triple*, and represents the value of the function and its derivatives with respect to the Cartesian axis in a given vertex. Then, it is easy to see that each basis $B_l^{(r)}$ $(r = 1, 2, 3)$ vanishes outside the molecule M_l of vertex $\mathbf{V}_l$ defined as the union of all triangles Ω_e containing $\mathbf{V}_l$. This guarantees the compactness of the support for PS B-splines.

In principle, it is possible to define the PS B-splines by simply choosing, for each vertex $\mathbf{V}_l \in \mathcal{T}$, three linearly independent vectors $(\alpha_{l,r}, \beta_{l,r}, \gamma_{l,r})$ $(r = 1, 2, 3)$, and by solving the interpolation problem (3). This procedure generates three linearly independent basis functions for each vertex. However, in a finite-element context, it is interesting to work with *normalized* B-splines, that are basis functions that form a partition of unity on Ω, i.e.

$$B_l^{(r)}(x,y) \geq 0 \quad \forall x, y \in \Omega,$$

$$\sum_{l=1}^{N_v} \sum_{r=1}^{3} B_l^{(r)}(x,y) = 1 \quad \forall x, y \in \Omega,$$

which leads to the following constrains on the triples

$$\sum_{r=1}^{3} \alpha_{l,r} = 1, \quad \sum_{r=1}^{3} \beta_{l,r} = 0, \quad \sum_{r=1}^{3} \gamma_{l,r} = 0.$$

This property is also crucial to ensure the completeness of the basis.

The procedure proposed in [14] leads to PS B-splines that form a partition of unity, and that have good features from a numerical point of view. This procedure is based on the fact that each set of three independent B-splines associated to a vertex $\mathbf{V}_l$ uniquely define a *control triangle* $T_l = (\mathbf{Q}_{l,1}, \mathbf{Q}_{l,2}, \mathbf{Q}_{l,3})$ for the PS spline, with vertices $\mathbf{Q}_{l,j} = (X_{l,j}, Y_{l,j})$ $(j = 1, 2, 3)$. The control

triangle of a PS spline plays the same role as the control polygon for univariate splines in terms of control of the shape of the spline. In fact, Ref. [14] shows that the control triangle and the coefficients $c_{l,j}$ define three control points $(X_{l,j}, Y_{l,j}, c_{l,j})$ defining a plane that is always tangent to the spline surface at the vertex $\mathbf{V}_l$. From a numerical point of view, optimal shape functions are associated to a control triangle of minimal area [14].

The points $\mathbf{Q}_{l,j}$ are related to the three triples and the vertex $\mathbf{V}_l$ via the relations,

$$\begin{bmatrix} \alpha_{l,1} & \alpha_{l,2} & \alpha_{l,3} \\ \beta_{l,1} & \beta_{l,2} & \beta_{l,3} \\ \gamma_{l,1} & \gamma_{l,2} & \gamma_{l,3} \end{bmatrix} \begin{bmatrix} X_{l,1} & Y_{l,1} & 1 \\ X_{l,2} & Y_{l,2} & 1 \\ X_{l,3} & Y_{l,3} & 1 \end{bmatrix} = \begin{bmatrix} x_l & y_l & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{4}$$

The positivity of the B-splines associated to the vertex $\mathbf{V}_l$ is guaranteed if all the barycentric coordinates with respect the control triangle T_l of all the PS points associated to $\mathbf{V}_l$ are positive. Geometrically, this means that all the PS points associated to $\mathbf{V}_l$ are contained within the control triangle T_l as shown on figure 2.

Hence, the procedure of computing the three shape functions for each vertex can be defined as:

- Find a suitable control triangle for the vertex $\mathbf{V}_l$ that contains all the PS points associated to $\mathbf{V}_l$;
- Compute the triple defining the three shape functions using (4).

The control triangles for each vertex are constructed looking for the minimal area triangle that contains all the PS points. This lead to an optimization problem consisting in circumscribing a convex polygon with a minimal area triangle. An optimal algorithm is proposed by O'Rourke in [17] for solving such a problem, with a computational cost $\mathcal{O}(n)$, with n the number of PS points. This algorithm has been implemented and used for optimal design of the PS basis functions. In figure 2, an example of control triangle is depicted, with the relative PS control points obtained with the O'Rourke algorithm, while in figure 3 the whole set of control triangles for an unstructured mesh is shown.

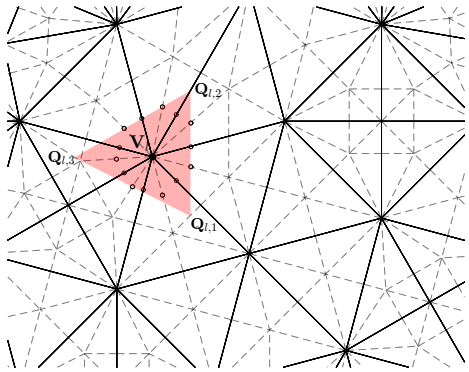


Figure 2: Powell-Sabin control triangle (in red) and relative Powell-Sabin points for the vertex $\mathbf{V}_l$.

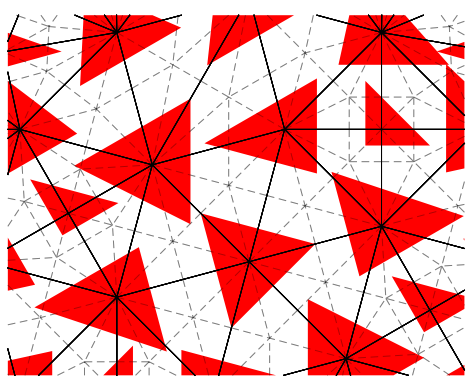


Figure 3: Control triangles (in red) for an unstructured mesh.

Table 1: Comparisons of condition numbers of the stiffness and mass matrices when considering optimal and non-optimal control triangles in a square domain $[0,1] \times [0,1]$ and for three structured meshes. Ne is the number of triangular elements of the mesh, h is the element size.

		CN. stiffness		CN. mass	
Ne	h	optimal	non-optimal	optimal	non-optimal
200	70.71E-03	83	389	171	4466
800	35.36E-03	326	453	163	4466
3200	17.68E-03	1305	1585	154	4469

In order to understand the role of the control triangle, the basis functions obtained with the optimal and non-optimal triangle are compared. This is carried out for the three bases of the vertex $\mathbf{V}_l$ shown in figure 2. The non-optimal triangle is obtained from the optimal one, by moving each vertex along the line joining the vertex with the node to which the triangle belongs, to a distance equal to five times the original one. The optimal and non-optimal basis functions are shown in figure 4. It is clear that, as the area of the control triangle increases, the three bases become more alike. In Tab. 1 is shown the impact on the condition number of the stiffness and mass matrices associated to the linear system generated with these basis functions. In the first case, the minimal area triangles are used to compute the bases. In the second case, non-optimal triangles generated as before are used. Results show that increasing the control triangle area increases the condition number, which in turn may affect the performance of the linear system solver and the accuracy of linear system solution.

4. Solution of the Euler equations with Powell-Sabin elements

The 2D Euler equations of gas dynamics are considered,

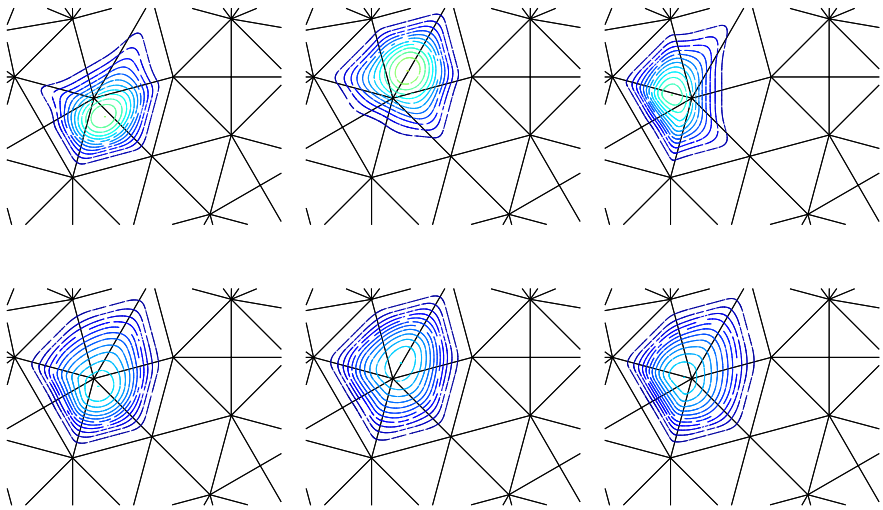


Figure 4: Basis functions for the vertex $\mathbf{V}_l$ generated with an optimal control triangle (upper line) and with a non-optimal control triangle (lower line).

$$
\begin{aligned}
\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{u}) &= 0 \\
\frac{\partial \rho \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{\mathcal{I}}) &= \mathbf{0} \\
\frac{\partial \rho E}{\partial t} + \boldsymbol{\nabla}\cdot((\rho E + p)\boldsymbol{u}) &= 0
\end{aligned}
\tag{5}
$$

where ρ is the density, $\boldsymbol{u}$ is the velocity vector, E is the total energy, $\boldsymbol{\mathcal{I}}$ is the identity matrix and p is the pressure. This latter is defined by the equation of state for a perfect gas $p = (\gamma_g - 1)\rho\epsilon$, where $\gamma_g = 1.4$ is the ratio of specific heats and $\epsilon = E - 1/2\|\boldsymbol{u}\|^2$ is the internal energy.

Problem (5) is recast in a conservative form,

$$
\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla}\cdot\boldsymbol{\mathcal{F}} = \mathbf{0}, \tag{6}
$$

where $\boldsymbol{U} = (\rho, \rho u, \rho v, \rho E)^T$ defines the vector of conservative variables, and $\boldsymbol{\mathcal{F}}$

is the flux tensor with components in Cartesian coordinates defined as

$$\boldsymbol{\mathcal{F}} = \left[\boldsymbol{F}_x \ \boldsymbol{F}_y\right], \quad \boldsymbol{F}_x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (\rho E + p) u \end{Bmatrix}, \quad \boldsymbol{F}_y = \begin{Bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho E + p) v \end{Bmatrix},$$

where (u, v) are the components of the velocity vector in the x and y directions, respectively.

The problem (6) is discretized using PS finite-elements. To simplify the presentation, the same symbol is used for the numerical approximation, belonging to the finite dimensional space, and the exact solution $\boldsymbol{U}$. Hence, the finite-element problem corresponding to (6) consists in finding $\boldsymbol{U} \in [S_2^1(\mathcal{T}^*)]^4$ such that

$$\int_\Omega \boldsymbol{V} \cdot \frac{\partial \boldsymbol{U}}{\partial t} \, d\Omega - \int_\Omega \boldsymbol{\nabla V} : \boldsymbol{\mathcal{F}} \, d\Omega + \int_{\partial\Omega} \boldsymbol{\mathcal{F} V} \cdot \boldsymbol{n} \, d\Gamma = 0, \tag{7}$$

for all the test functions $\boldsymbol{V} \in [S_2^1(\mathcal{T}^*)]^4$.

4.1. Imposition of the boundary conditions

Solid walls and free-stream boundaries are considered as boundary conditions. Walls are modelled by a slip boundary condition, which correspond to imposing the normal component of the flux $\boldsymbol{\mathcal{F}}$ equal to zero at the wall. This kind of condition does not entail any specific problem, and it is weakly imposed by the boundary integral in (7). On the contrary, inlet and outlet boundary conditions are imposed setting different components of the unknown vector at the boundary nodes, depending on the subsonic/supersonic regime at the boundary, see [18]. The problem of imposing boundary values deserves particular attention in the framework of PS finite-elements. A careful choice of the control triangle associated to the boundary node $\mathbf{V}_l$ simplifies this task. Two different cases must be considered: the boundary node belongs to two edges forming an angle different to π, or equal to π. From equation (4), the following relations are

derived

$$\alpha_{l,1}\mathbf{Q}_{l,1} + \alpha_{l,2}\mathbf{Q}_{l,2} + \alpha_{l,3}\mathbf{Q}_{l,3} = \mathbf{V}_l, \tag{8}$$

$$\beta_{l,1}\mathbf{Q}_{l,1} + \beta_{l,2}\mathbf{Q}_{l,2} + \beta_{l,3}\mathbf{Q}_{l,3} = \mathbf{e}_x, \tag{9}$$

$$\gamma_{l,1}\mathbf{Q}_{l,1} + \gamma_{l,2}\mathbf{Q}_{l,2} + \gamma_{l,3}\mathbf{Q}_{l,3} = \mathbf{e}_y, \tag{10}$$

where $\mathbf{e}_x$ and $\mathbf{e}_y$ are respectively the unity vector in the x and y directions. Equation (8) states that $(\alpha_{l,1}, \alpha_{l,2}, \alpha_{l,3})$ are the barycentric coordinates of the vertex $\mathbf{V}_l$ with respect to the control triangle, while (9) and (10) mean that $(\beta_{l,1}, \beta_{l,3}, \beta_{l,3})$ and $(\gamma_{l,1}, \gamma_{l,2}, \gamma_{l,3})$ are respectively the barycentric coordinates of the vectors $\mathbf{e}_x$ and $\mathbf{e}_y$ with respect to the control triangle.

In the case where $\mathbf{V}_l$ belongs to an angle different from π, if the control triangle has two sides aligned with the two boundary edges, as in figure 5 (left), according to equation (8) only one shape function has a value different from zero in $\mathbf{V}_l$. This allows to directly set the boundary value to the unknown associated to the non-zero shape function. Moreover, according to (9) and (10) the other two shape functions have zero derivative in the directions of the two edges concurring in $\mathbf{V}_l$. In particular, using the notation of figure 5 (left), where triangle vertices are now called $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$, it results

$$\alpha_{l,B} = \alpha_{l,C} = 0, (\Rightarrow \alpha_{l,A} = 1),$$
$$\begin{Bmatrix} \beta_{l,B} \\ \gamma_{l,B} \end{Bmatrix} \cdot \boldsymbol{t} = 0,$$
$$\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} \cdot \boldsymbol{r} = 0.$$

Denoting with U the component of $\boldsymbol{U}$ to be imposed, and $\hat{u}$ the $\mathcal{L}^2$ projection of the prescribed values on the space of univariate quadratic splines on $\partial\Omega$, the

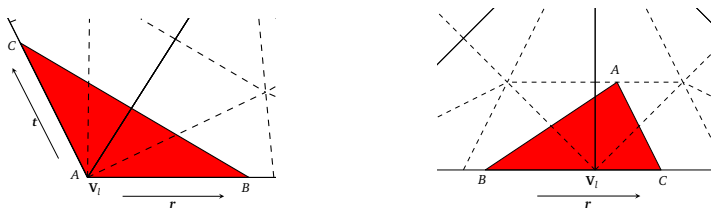


Figure 5: Control triangles for imposing boundary values: angle different from π (left) and equal to π (right).

resulting equations relative to the unknowns of the vertex $\mathbf{V}_l$ write

$$
\begin{aligned}
U_{l,A} &= \hat{u}(\mathbf{V}_l), \\
U_{l,A}\begin{Bmatrix}\beta_{l,A}\\ \gamma_{l,A}\end{Bmatrix}\cdot\boldsymbol{r} + U_{l,B}\begin{Bmatrix}\beta_{l,B}\\ \gamma_{l,B}\end{Bmatrix}\cdot\boldsymbol{r} &= \boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\cdot\boldsymbol{r}, \\
U_{l,A}\begin{Bmatrix}\beta_{l,A}\\ \gamma_{l,A}\end{Bmatrix}\cdot\boldsymbol{t} + U_{l,C}\begin{Bmatrix}\beta_{l,C}\\ \gamma_{l,C}\end{Bmatrix}\cdot\boldsymbol{t} &= \boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\cdot\boldsymbol{t},
\end{aligned}
$$

that is

$$
\begin{aligned}
U_{l,A} &= \hat{u}(\mathbf{V}_l), \\
U_{l,B} &= \frac{\boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\cdot\boldsymbol{r} - \hat{u}(\mathbf{V}_l)\begin{Bmatrix}\beta_{l,A}\\ \gamma_{l,A}\end{Bmatrix}\cdot\boldsymbol{r}}{\begin{Bmatrix}\beta_{l,B}\\ \gamma_{l,B}\end{Bmatrix}\cdot\boldsymbol{r}}, \\
U_{l,C} &= \frac{\boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\cdot\boldsymbol{t} - \hat{u}(\mathbf{V}_l)\begin{Bmatrix}\beta_{l,A}\\ \gamma_{l,A}\end{Bmatrix}\cdot\boldsymbol{t}}{\begin{Bmatrix}\beta_{l,C}\\ \gamma_{l,C}\end{Bmatrix}\cdot\boldsymbol{t}},
\end{aligned}
$$

where $U_{l,A}, U_{l,B}, U_{l,C}$ are the unknowns related to the shape functions with triples $(\alpha_{l,A}, \beta_{l,A}, \gamma_{l,A})$, $(\alpha_{l,B}, \beta_{l,B}, \gamma_{l,B})$ and $(\alpha_{l,C}, \beta_{l,C}, \gamma_{l,C})$, respectively .

In the case where $\mathbf{V}_l$ belongs to an angle equal to π, if the control triangle is aligned with the boundary, as in figure 5 (right), according to equation (8), there is one shape function with zero value on $\mathbf{V}_l$. Hence, the unknown related

to this shape function is not affected by the boundary condition. Moreover, equations (9) and (10) guarantee again that the same shape function has zero derivative in the direction parallel to the boundary. Thus, with reference to figure 5 (right), results

$$\alpha_{l,A} = 0,\ \alpha_{l,B} \neq 0,\ \alpha_{l,C} \neq 0,$$
$$\begin{Bmatrix} \beta_{l,A} \\ \gamma_{l,A} \end{Bmatrix} \cdot \boldsymbol{r} = 0,$$

leading to the following relations for the unknowns in the vertex $\boldsymbol{V}_l$

$$U_{l,B}\alpha_{l,B} + U_{l,C}\alpha_{l,C} = \hat{u}(\mathbf{V}_l),$$
$$U_{l,B}\begin{Bmatrix} \beta_{l,B} \\ \gamma_{l,B} \end{Bmatrix} \cdot \boldsymbol{r} + U_{l,C}\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} \cdot \boldsymbol{r} = \boldsymbol{\nabla}\hat{u}(\mathbf{V}_l) \cdot \boldsymbol{r},$$

that is

$$U_{l,B} = \frac{\left(\hat{u}(\mathbf{V}_l)\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} - \alpha_{l,C}\boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\right) \cdot \boldsymbol{r}}{\left(\alpha_{l,B}\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} - \alpha_{l,C}\begin{Bmatrix} \beta_{l,B} \\ \gamma_{l,B} \end{Bmatrix}\right) \cdot \boldsymbol{r}},$$
$$U_{l,C} = \frac{\left(\hat{u}(\mathbf{V}_l)\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} - \alpha_{l,C}\boldsymbol{\nabla}\hat{u}(\mathbf{V}_l)\right) \cdot \boldsymbol{r}}{\left(\alpha_{l,B}\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix} - \alpha_{l,B}\begin{Bmatrix} \beta_{l,C} \\ \gamma_{l,C} \end{Bmatrix}\right) \cdot \boldsymbol{r}}.$$

5. Stabilization by artificial viscosity

Artificial viscosity is introduced through two extra terms $\mathcal{D}_{\text{SUPG}}$ and $\mathcal{D}_{\text{SC}}$, leading to a stabilized formulation of the Galerkin weak form 7 that reads

$$\int_\Omega \boldsymbol{V} \cdot \frac{\partial \boldsymbol{U}}{\partial t}\, d\Omega - \int_\Omega \boldsymbol{\nabla}\boldsymbol{V} : \boldsymbol{\mathcal{F}}\, d\Omega + \int_{\partial\Omega} \boldsymbol{\mathcal{F}}\boldsymbol{V} \cdot \boldsymbol{n}\, d\Gamma + \mathcal{D}_{\text{SUPG}} + \mathcal{D}_{\text{SC}} = 0.$$

The two stabilization terms correspond to a SUPG term, $\mathcal{D}_{\text{SUPG}}$, and a shock-capturing term, $\mathcal{D}_{\text{SC}}$ that will be described in the following. The definition of the stabilization terms involves the Jacobian of the fluxes $\mathbb{A}$, that is defined as

$$\mathbb{A} = \frac{\partial \boldsymbol{\mathcal{F}}}{\partial \boldsymbol{U}},$$

and it can be expressed as a second-order tensor for each Cartesian direction, that is

$$\frac{\partial \boldsymbol{F}_x}{\partial \boldsymbol{U}} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma_g-1}{2}\|\boldsymbol{u}\|^2 - u^2 & (3-\gamma_g)u & (1-\gamma_g)v & \gamma_g - 1 \\ -uv & v & u & 0 \\ \frac{\gamma_g-1}{2}u\|\boldsymbol{u}\|^2 - uH & -u^2(\gamma_g-1)+H & (1-\gamma_g)uv & \gamma_g u \end{pmatrix},$$

$$\frac{\partial \boldsymbol{F}_y}{\partial \boldsymbol{U}} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma_g-1}{2}\|\boldsymbol{u}\|^2 - v^2 & (1-\gamma_g)u & (3-\gamma_g)v & \gamma_g - 1 \\ \frac{\gamma_g-1}{2}v\|\boldsymbol{u}\|^2 - vH & (1-\gamma_g)uv & -v^2(\gamma_g-1)+H & \gamma_g v \end{pmatrix},$$

having defined the entalpy, $H = E + p/\rho$.

5.1. SUPG stabilization

The SUPG stabilization technique is used to produce a stable upwind discretization without introducing excessive numerical dissipation. The SUPG method was introduced by T. Hughes in [15], and it is one of the most established stabilized formulations in finite-element computations of flows. The SUPG method introduces a certain amount of artificial viscosity in the streamline direction only. In this work, a simplified version of the SUPG term is used,

$$\mathcal{D}_{\text{SUPG}} = \sum_{e=1}^{Ne} \int_{\Omega_e} \mathbb{A}^T \boldsymbol{\nabla V} : \left(\boldsymbol{\tau}_{\text{SUPG}} \mathbb{A} \boldsymbol{\nabla U}\right) \, d\Omega_e, \tag{11}$$

which does not take into account the complete residual of the original equation, but only the divergence term. This SUPG stabilization assumes that the time derivative contribution is negligible compared to the the other contributions, while the consistency of the formulation is ensured by the choice of $\boldsymbol{\tau}_{\text{SUPG}}$. The elimination of the time derivative term entails a substantial reduction in the implementation complexity. This formulation has been found to be enough efficient for stabilizing the PS discretization of the Euler equations.

The stabilization matrix $\boldsymbol{\tau}_{\text{SUPG}}$ is a key ingredient in the SUPG definition, and its definitions up to now mostly relies on heuristic arguments. A study on the influence of this parameter can be found in [1] for example. In this work, a simple definition is adopted,

$$\boldsymbol{\tau}_{\text{SUPG}} = \tau \boldsymbol{\mathcal{I}},$$

where $\tau = C_{\text{SUPG}} dt$, and C_{SUPG} is a constant. As a consequence, τ is proportional with the time step in the whole domain. This choice allows a computationally inexpensive evaluation of the stabilization matrix $\boldsymbol{\tau}_{\text{SUPG}}$, and guarantees the consistency of the formulation, since the added viscosity vanishes when dt is reduced.

5.2. Shock-capturing

A shock-capturing stabilization term is used to prevent oscillations of the solution around the discontinuities arising in supersonic regimes. These spurious oscillations might lead to severe accuracy loss or stability problems. The shock-capturing stabilization term introduces artificial viscosity only in a sharp zone surrounding the discontinuities. Isotropic and anisotropic shock-capturing formulations can be found in the literature. The isotropic formulation introduces a Laplacian based artificial diffusion, while the anisotropic formulation introduces diffusion in a certain direction usually defined by the gradient of a physical quantity, such as the density or the pressure. In this work, both formulations are considered and tested, with the following expressions:

- *Isotropic* shock-capturing term

$$\mathcal{D}_{\text{SC}} = \sum_{e=1}^{Ne} \int_{\Omega_e} \boldsymbol{\nabla V} : \left(\boldsymbol{\tau}_{\text{SC}} \boldsymbol{\nabla U} \right) \, d\Omega_e, \tag{12}$$

- *Anisotropic* shock-capturing term (see [5])

$$\mathcal{D}_{\text{SC}} = \sum_{e=1}^{Ne} \int_{\Omega_e} \boldsymbol{\nabla V} \; \boldsymbol{b} \cdot \left(\boldsymbol{\tau}_{\text{SC}} \boldsymbol{\nabla U} \; \boldsymbol{b} \right) \, d\Omega_e, \tag{13}$$

where the vector $\boldsymbol{b}$, which gives the direction in which the artificial viscosity is added, is defined as

$$\boldsymbol{b} = \frac{\boldsymbol{\nabla} p}{\|\boldsymbol{\nabla} p\| + \epsilon},$$

with the tolerance ϵ that is used to prevent a non-defined expression where the gradient of the pressure is very small.

The stabilization matrix $\boldsymbol{\tau}_{\text{SC}}$ in (12)(13) is chosen as a diagonal matrix,

$$\boldsymbol{\tau}_{\text{SC}} = \tau_{\text{SC}} \boldsymbol{\mathcal{I}},$$

with a coefficient τ_{SC} that depends on the local characteristics of the flow and it is evaluated in each Gauss point of the domain.

A large choice can be found in literature for the expression of τ_{SC}. In the work of Tedzuyar [1], the expression proposed for τ_{SC} is

$$\tau_{\text{SC}} = \tau_{\text{Tz}} = \frac{\tau(\beta_{sc} = 1) + \tau(\beta_{sc} = 2)}{2}, \tag{14}$$

where

$$\tau(\beta_{sc}) = [\mathbf{Y}^{-1} \mathbb{A} \widetilde{\boldsymbol{\nabla} \boldsymbol{U}}]_e [\widetilde{\boldsymbol{\nabla} \boldsymbol{U}}]_e^{-(2-\beta_{sc})} [\widetilde{\boldsymbol{U}}]_e^{(1-\beta_{sc})} [\widetilde{\boldsymbol{\nabla} \rho}]_e^{-\beta_{sc}}$$

and the terms in the previous expression are defined as

$$[\widetilde{\boldsymbol{\nabla} \rho}]_e = \sum_{r=1}^{3} \sum_{q=1}^{3} \left| \frac{\boldsymbol{\nabla} \rho \cdot \boldsymbol{\nabla} B_r^{(q)}}{|\boldsymbol{\nabla} \rho|} \right|, \quad [\widetilde{\boldsymbol{\nabla} \boldsymbol{U}}]_e = \left(\sum_{d=1}^{2} \left\| \mathbf{Y}^{-1} \frac{\partial \boldsymbol{U}}{\partial x_d} \right\|^2 \right)^{\frac{1}{2}},$$

$$[\widetilde{\boldsymbol{U}}]_e = \|\boldsymbol{U}\|, \quad [\mathbb{A} \widetilde{\boldsymbol{\nabla} \boldsymbol{U}}]_e = \|\mathbb{A} \boldsymbol{\nabla} \boldsymbol{U}\|.$$

The diagonal matrix $\mathbf{Y}$ contains the reference values for each component of the vector solution. Expression (14) was proposed for an isotropic expression of the shock-capturing term (12), but may be used also in an anisotropic setting.

Another option for the coefficient τ_{SC} is proposed in [5], for a DG discretization. A version of this coefficient, adapted to a continuous discretization eliminating the jump contributions, is considered here,

$$\tau_{\text{SC}} = \tau_{\text{B}} = C h^2 \frac{|d_p|}{p} f_p, \tag{15}$$

where

$$d_p = \sum_i \frac{\partial p}{\partial \boldsymbol{U}_i} (\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}})_i, \quad f_p = \frac{\|\boldsymbol{\nabla} p\|}{p} h$$

h being the element size, and C an user-defined parameter. In [5], this formulation was proposed associated to an anisotropic version of the shock-capturing technique. Here, it is used for both the isotropic and anisotropic versions.

6. Verification and convergence study

The convergence properties of the PS-FEM scheme are studied for a smooth solution using the method of the manufactured solutions. An implicit algorithm is used to solve the non-linear system given by the stationary Euler equations,

$$\boldsymbol{\nabla} \cdot \boldsymbol{\mathcal{F}} = \boldsymbol{S},$$

where $\boldsymbol{S}$ is a known source term obtained replacing the unknowns with the analytical solution. A square domain $\Omega = [0, 10] \times [0, 10]$ is considered, and discretized with structured triangular meshes of element size $h = 10/2^{2+i}$, for $i = 0, 1, 2, 3$.

The following analytical solution is considered

$$\begin{aligned} \rho &= c_\rho + \cos(x)\cos(y), \\ u &= c_u + \cos(x)\cos(y), \\ v &= c_v + \sin(x)\sin(y), \\ E &= c_E + \sin(x)\cos(y), \end{aligned}$$

where c_ρ, c_u, c_v and c_E are constant parameters. These parameters are properly chosen to the values $c_\rho = 2$, $c_u = c_v = 6 \times 10^2$, $c_E = 4 \times 10^5$ in order to obtain a supersonic solution at the boundaries of the domain. This makes substantially easier the way of imposing Dirichlet boundary conditions. More precisely, with this choice of the parameters two boundaries correspond to *supersonic inflows*,

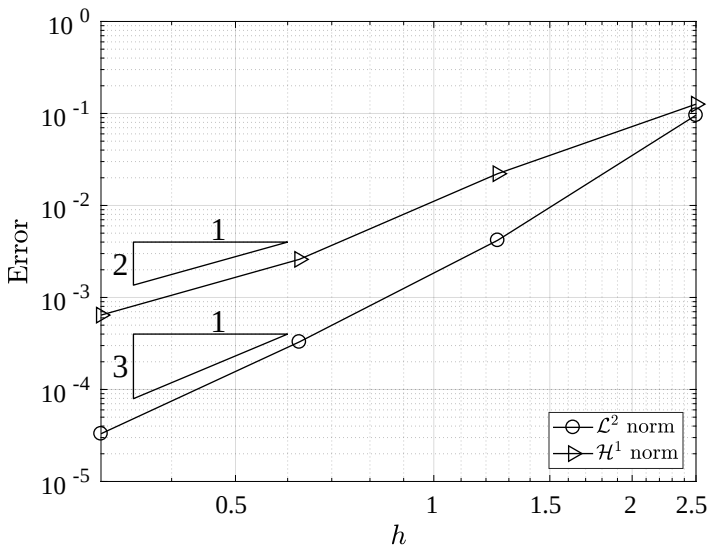


Figure 6: $\mathcal{L}^2$and $\mathcal{H}^1$ error convergence curves of the PS-FEM scheme for the ρ variable. The curves for the other variables are overlapped to this one, and are not displayed.

where all the variables can be set, and two boundaries correspond to *supersonic outflows*, where no unknowns need to be imposed, see in [18] for details. Obviously, no stabilization is considered in this case.

Results of the convergence tests show the correct third-order and second-order convergence rates in the $\mathcal{L}^2$ and $\mathcal{H}^1$ norms, respectively, see figure 6.

7. Validation tests and evaluation of the stabilization performance

The code is validated using two classical test cases introduced in Ref. [19]. The performance of the stabilization is also qualitatively evaluated by estimating the amount of added diffusion due to the shock-capturing term. It is done by examining the smearing of the discontinuities across the mesh, and by assessing the recovery of usual flow features.

An explicit time discretization method is used for both examples with a standard fourth-order Runge-Kutta scheme. As in [20], the CFL is defined as the maximum over the triangles of $(\|\boldsymbol{u}\| + c)P_K dt/A_K$, where c is the local sound speed, and P_K and A_K are respectively the perimeter and the area of the

element K. With this definition, stable computations are found up to CFL $= 1$. A simple positivity correction procedure is used, consisting on setting artificially the internal energy to the threshold value (chosen equal to 10^{-5}) in those Gauss points where a negative value is obtained.

7.1. The Double Mach reflection test

The Double Mach reflection test represents the impinging of a strong shock wave at Mach $= 10$ against an oblique wall, figure 7. This produces a typical reflected shock wave with a curved shape and a bubble of denser gas. The computational domain is a box of dimensions 4×1, and a wall boundary condition (slip boundary) is set in the lower boundary starting from $x = 1/6$. The right boundary, as well as the remaining of the lower boundary (from $x = 0$ to $x = 1/6$) are set to an outflow condition, while the left and the top boundaries are set to an inlet condition. In the upper boundary, a time variable condition is set, simulating the traveling of the shock wave to the right. In the initial condition, the shock wave creates an angle of $60°$ from the x-axis and impinges the wall at $x = 1/6$. Post-shock conditions are set to the left of the wave ($\rho = 8$, $\|\boldsymbol{u}\| = 8.25$ and $p = 116.5$), pre-shock to the right (fluid at rest with $\rho = 1.4$ and $p = 1$). In order to avoid spurious oscillations of the solution in the first steps of the simulation, the fields are initialized with a smooth shock profile in the domain, with the discontinuity smeared over four elements. Results are shown for $t = 0.2$ time units, and in a box 3×1.

Figure 8 shows the resulting density profile on a mesh with $h = 1/480$. A small CFL$= 0.15$ is used in this case, and C_{SUPG} is set to 0.5. In figure 8a, the result is obtained with the isotropic shock-capturing formulation (12) and $\tau_{\text{SC}} = \tau_{\text{Tz}}$. Very similar results (not shown here) are obtained with the same value of τ_{SC} using the anisotropic formulation. The reference values for $\mathbf{Y}$ used in this case are the post-shock conditions. Results show how this shock-capturing technique leads to an overly diffusive solution. The secondary shocks occurring inside the bubble appears to be strongly smeared, and the vortex structure generated by the contact discontinuities, often found in the literature

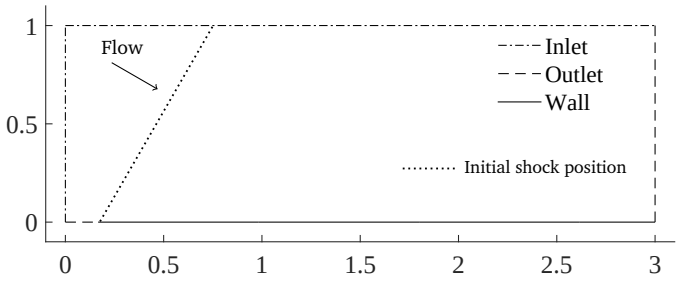


Figure 7: Double Mach reflection case: computational domain and boundary conditions.

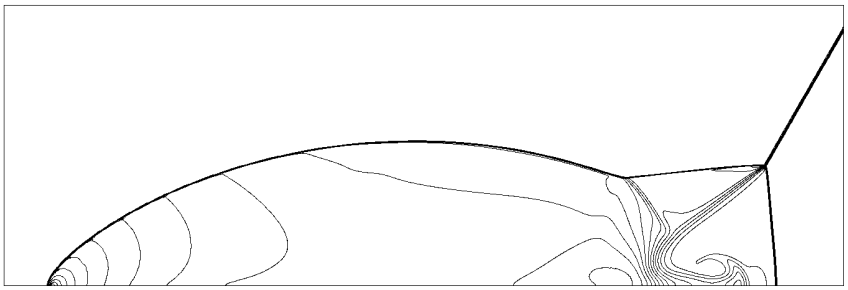
solutions, is completely absent. In figures 8b and 8c the coefficient $\tau_{\text{SC}} = \tau_{\text{B}}$ is used respectively with the isotropic and the anisotropic (13) shock-capturing formulations. The results are significantly improved. The expected features of the solution are now visible. The anisotropic formulation however provides a better accuracy of the solution with small vortices visible in the shear layer at the back of the bubble, even if a stronger numerical noise is clearly visible upstream.

Our results show that the solution accuracy is improved when increasing the artificial diffusion introduced by the SUPG term, as it can be seen in figure 9, with CFL= 1 and $C_{\text{SUPG}} = 0.17$. In fact, the increased SUPG artificial diffusion introduced by a larger dt (τ_{SC} is the double than in the previous cases) is sufficient to smooth out the noise, without affecting the resolution of the shear layers downstream. A zoom of the solution in the back of the domain on figure 10 shows the very well resolved Mach stem roll-up, with very good agreement with literature solutions, see for example in Ref. [3]. Also, the position of the head of the jet matches with the value usually reported in the literature which is estimated between 2.71 and 2.74, see [21].

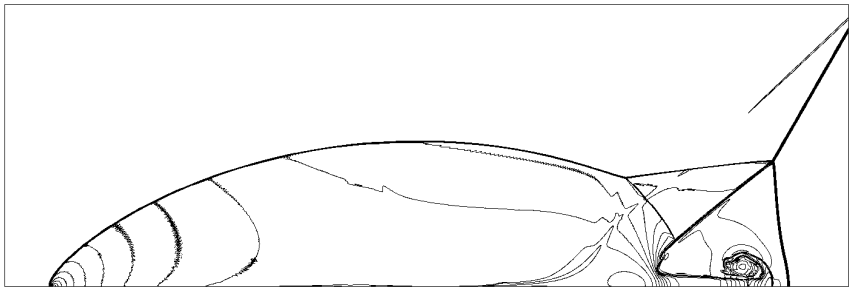
7.2. Numerical simulation of a supersonic flow past a forward facing step

Here, a supersonic flow is entering into an infinite long wind tunnel with a step, figure 11. The interactions of the supersonic flow with the step and the tunnel walls are known to create a typical pattern of shock reflections.

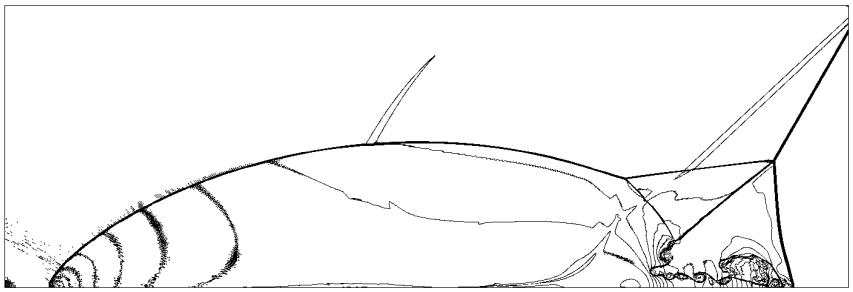
The computational domain has dimensions 3×1 length units, and the flow



(a)



(b)



(c)

Figure 8: Double Mach reflection problem: density ρ, 30 equally spaced contours of density profile at $t = 0.2$ obtained with a mesh of $h = 1/480$, CFL=0.15 and $C_{\mathrm{SUPG}} = 0.5$. Comparison of the results obtained with the isotropic shock-capturing technique and $\tau_{\mathrm{SC}} = \tau_{\mathrm{Tz}}$ (a), isotropic with $\tau_{\mathrm{SC}} = \tau_{\mathrm{B}}$ (b), and the anisotropic with $\tau_{\mathrm{SC}} = \tau_{\mathrm{B}}$ (c).

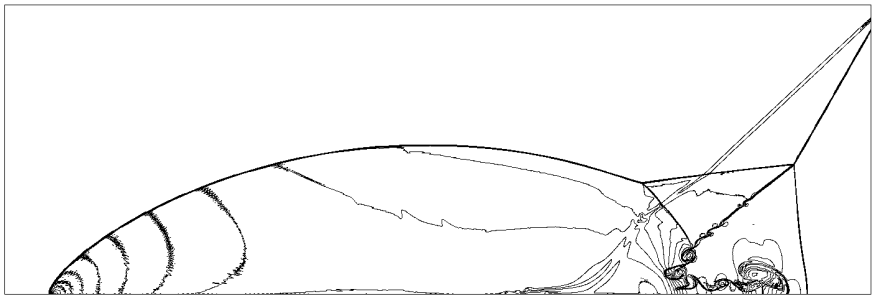


Figure 9: Double Mach reflection problem: density ρ, 30 equally spaced contours of density profile at $t = 0.2$ obtained with a mesh of $h = 1/480$, CFL= 1 and the anisotropic shock-capturing technique, with $\tau_{\rm SC} = \tau_{\rm B}$, $C_{\rm SUPG} = 0.17$.

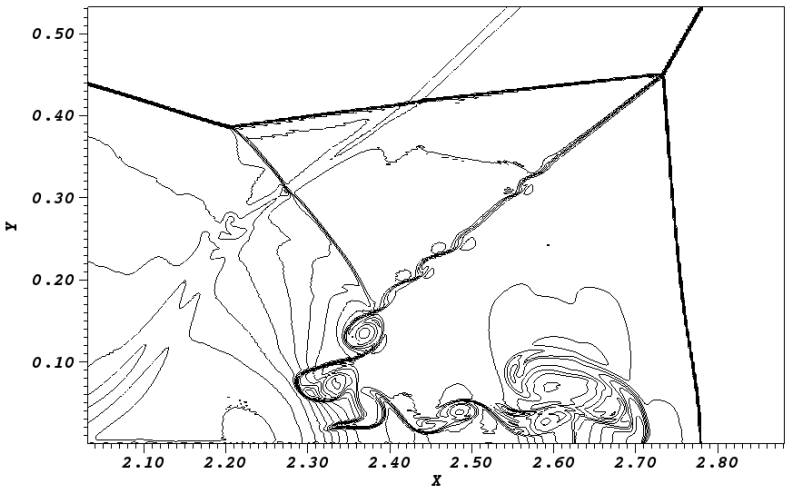


Figure 10: Double Mach reflection problem: zoom in the shock region for $\tau_{\rm SC} = \tau_{\rm B}$, anisotropic shock-capturing technique, $C_{\rm SUPG} = 0.17$ and CFL=1.

facing step is 0.2 length units high and it is located at 0.6 length units from the left boundary. The inflow boundary conditions on the left correspond to an uniform supersonic velocity, Mach 3, and a constant pressure $p = 1$. The tunnel walls correspond to no-slip boundaries, and the boundary on the right hand side is a free exit. The simulation is started setting the variables at the inlet boundary values. No particular treatment is done at the corner of the step. In all computations, CFL=0.7 and $C_{\text{SUPG}} = 0.5$ are used. All the results are shown at $t = 4$ time units.

Figure 12, shows the density iso-contours for a mesh $h = 1/160$ and compares different shock-capturing techniques. Figure 12a is related to the isotropic formulation, with $\tau_{\text{SC}} = \tau_{\text{Tz}}$. Notice that, like in the previous test case, the anisotropic formulation with the same values of τ_{SC} provides similar results (not shown here). The reference values used in the $\mathbf{Y}$ matrix are $\rho_{ref} = 1.4$, $u_{ref} = 3$, $v_{ref} = 0.3$, $p_{ref} = 1$. In particular, since the inlet value of v is zero, v_{ref} has been chosen as the one giving a stable solution with the most sharp shock profile. Result clearly shows an over-diffusive solution compared to reference results of literature using similar discretization (for example in Ref. [20]). In particular, the Mach steam appears slightly too high, and no vortices related to Kelvin-Helmholtz instability appear within the shear layer located close the top of the domain, while it was a typical feature of a well-resolved solution for this test. Plots in figures 12b and 12c are obtained using the isotropic and the anisotropic shock-capturing terms, respectively, but with $\tau_{\text{SC}} = \tau_{\text{B}}$. The contact discontinuity clearly appears to be better resolved, but the interaction with the step produces an unexpected artificial shear layer arising from the corner. Moreover, in figure 12c, it can also be noticed another unexpected boundary layer occurring near the bottom wall, just in front of the step.

These results suggests that a combination of the formulations could provide a better accuracy. This hybrid technique consists in using the isotropic formulation with τ_{Tz} in a small area around the corner defined by the union of $0.18 < y < 0.22$ and $x \geq 0.58$ and with $\tau_{\text{SC}} = \tau_{\text{B}}$ in the bottom of the domain before the step, corresponding to $y < 0.02$. The anisotropic version with $\tau_{\text{SC}} = \tau_{\text{B}}$

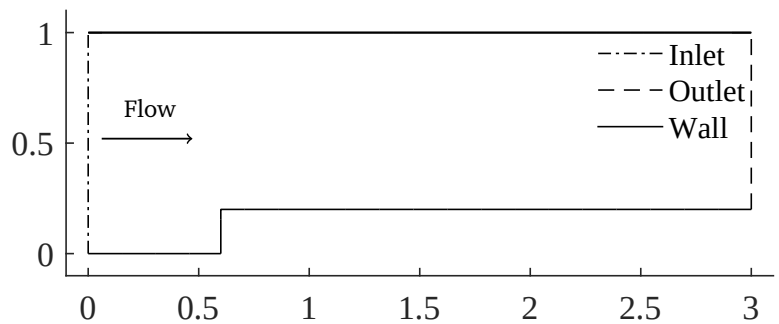


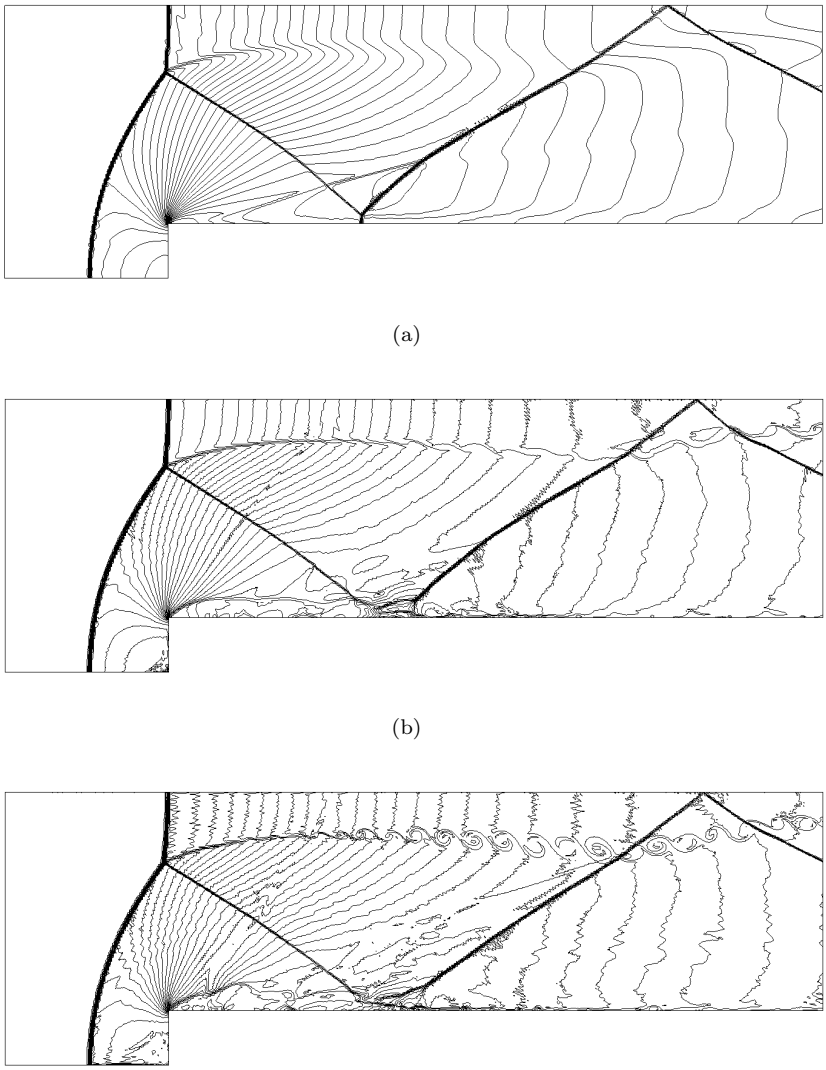
Figure 11: Forward facing step case: computational domain and boundary conditions.

is used in the rest of the domain. The result is shown, for a mesh $h = 1/320$, in figure 13a. The Mach steam has now completely disappeared, and small vortices related the Kelvin-Helmholtz instability are clearly visible. As for the double-Mach reflection test, some noise remains however visible on the contour lines as in the hybrid solution. This is typically related to a better capturing of the discontinuities, see for example Ref. [22]. This solution is compared to the one obtained on the same mesh with an uniform isotropic formulation and $\tau_{\text{SC}} = \tau_{\text{Tz}}$, depicted figure 13b, where we can notice how the results are still impacted by an over-diffusivity, also with this very fine mesh.

Figure 14 shows the shock profiles on a horizontal line at $y = 0.1$, obtained with four different discretizations $h = 1/40, 1/80, 1/260, 1/320$. For each discretization, the corresponding size of the element is also shown. The shock is spread across about four element widths in both cases and in all the discretizations. However, no under-shots are present with the hybrid technique, which also presents a sharper transition at the shock exit.

8. CONCLUSIONS

In this paper a finite-element method based on Powell-Sabin (PS) splines has been presented for the solution of the 2D compressible Euler equations in supersonic regime. A stabilized Galerkin formulation has been considered, with a particular emphasis on the shock-capturing technique. The classic isotropic

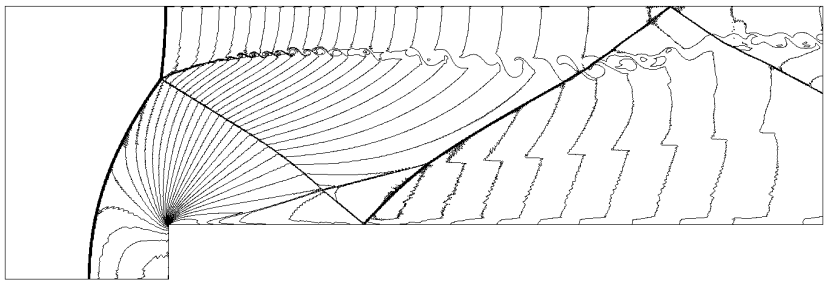


(a)

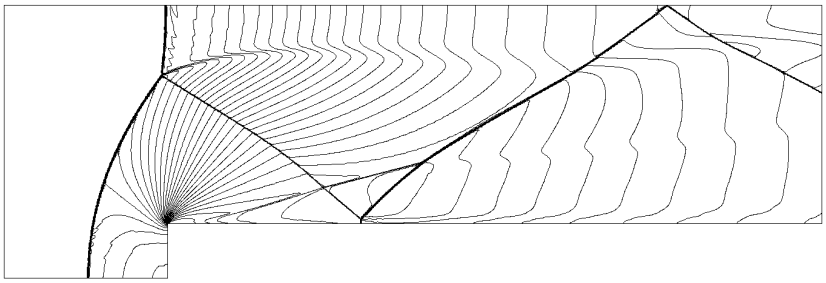
(b)

(c)

Figure 12: Forward facing step case, $h = 1/160$, 30 equally spaced contours of density profile at $t = 4$. Comparison of the results obtained with the isotropic shock-capturing technique and $\tau_{\mathrm{SC}} = \tau_{\mathrm{Tz}}$ (a), isotropic with $\tau_{\mathrm{SC}} = \tau_{\mathrm{B}}$ (b), and the anisotropic with $\tau_{\mathrm{SC}} = \tau_{\mathrm{B}}$ (c).



(a)



(b)

Figure 13: Forward facing step problem, $h = 1/320$, density ρ, 30 equally spaced contour lines at $t = 4$: hybrid shock-capturing technique (a) and isotropic shock-capturing technique with $\tau_{\mathrm{SC}} = \tau_{\mathrm{Tz}}$ (b).

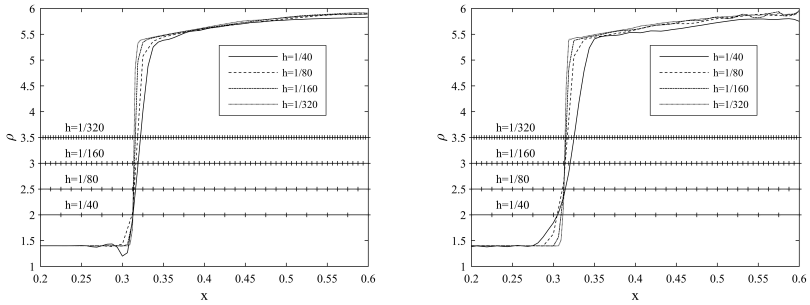


Figure 14: Forward facing step case: shock profile at a section $y = 0.1$ for the four meshes, with the isotropic shock-capturing and $\tau_{\mathrm{SC}} = \tau_{\mathrm{Tz}}$ (left) and the hybrid shock-capturing (right). The discretization is also displayed.

formulation of Tedzuyar [1], usually associated with continuous Galerkin methods, is compared with a novel formulation inspired from the anisotropic approaches typically adopted in the context of discontinuous Galerkin methods. The comparison carried out on two challenging test cases highlights that, while the isotropic technique proposed by Tedzuyar is able to produce a stable solution, it is slightly over-diffusive. The anisotropic technique is able to capture the typical features of the two test cases, such as Kelvin-Helmholtz instability and vortex roll-up, that are known to appear when sufficient resolution is provided to the simulation. An hybrid shock-capturing approach is also proposed and tested, which eliminates the spurious shear layers introduced by the anisotropic version in presence of the corner singularity.

The Powell-Sabin discretization is therefore a competitive alternative to other approaches historically used for shock-capturing problems, with moreover a global C_1 feature that is crucial in certain formulations of physical problems that require this characteristic.

References

[1] T. E. Tezduyar, M. Senga, Stabilization and shock-capturing parameters in SUPG formulation of compressible flows, Comput. Methods Appl. Mech. Eng. 195 (13-16) (2006) 1621–1632.

[2] J. V. Neumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (3) (1950) 232–237.

[3] D. S. Balsara, S. Garain, C.-W. Shu, An efficient class of WENO schemes with adaptive order, J. Comput. Phys. 326 (2016) 780–804.

[4] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods., in: Proc. of the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2006.

[5] F. Bassi, F. Cecchi, N. Franchina, S. Rebay, M. Savini, High-order dis-

continuous Galerkin computation of axisymmetric transonic flows in safety relief valves, Comput. Fluids 49 (1) (2011) 203–213.

[6] A. Huerta, E. Casoni, J. Peraire, A simple shock-capturing technique for high-order discontinuous Galerkin methods, Int. J. Numer. Methods Fluids 69 (10) (2012) 1614–1632.

[7] R. Sevilla, O. Hassan, K. Morgan, An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows, Comput. Methods Appl. Mech. Eng. 253 (2013) 15 – 27.

[8] H. Speleers, C. Manni, F. Pelosi, M. L. Sampoli, Isogeometric analysis with Powell-Sabin splines for advection-diffusion-reaction problems, Comput. Methods Appl. Mech. Eng. 221-222 (0) (2012) 132–148.

[9] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Comput. Methods Appl. Mech. Eng. 194 (39-41) (2005) 4135–4195.

[10] Y. Bazilevs, V. Calo, J. Cottrell, T. Hughes, A. Reali, G. Scovazzi, Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows, Comput. Methods Appl. Mech. Eng. 197 (1-4) (2007) 173–201.

[11] Y. Bazilevs, V. Calo, J. Cottrell, J. Evans, T. Hughes, S. Lipton, M. Scott, T. Sederberg, Isogeometric analysis using T-splines, Comput. Methods Appl. Mech. Eng. 199 (5-8) (2010) 229–263.

[12] M. J. D. Powell, M. A. Sabin, Piecewise quadratic approximations on triangles, ACM Trans. Math. Softw. 3 (4) (1977) 316–325.

[13] N. Jaxon, X. Qian, Isogeometric analysis on triangulations, Comput. Aided Des. 46 (2014) 45–57.

[14] P. Dierckx, On calculating normalized Powell-Sabin B-splines, Comput. Aided Geom. D. 15 (1) (1997) 61–78.

[15] A. N. Brooks, T. J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Methods Appl. Mech. Eng. 32 (1) (1982) 199–259.

[16] M.-J. Lai, L. L. Schumaker, Spline Functions on Triangulations, Cambridge University Press, Cambridge, 2007.

[17] J. O'Rourke, A. Aggarwal, S. Maddila, M. Baldwin, An optimal algorithm for finding minimal enclosing triangles, J. Algorithm. 7 (2) (1986) 258–269.

[18] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley & Sons, Chichester (UK), 2003.

[19] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1) (1984) 115–173.

[20] B. Cockburn, C. W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (6) (1998) 2440–2463.

[21] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (11) (2011) 4248 – 4267.

[22] O. Zanotti, F. Fambri, M. Dumbser, A. Hidalgo, Space-time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting, Comput. Fluids 118 (2015) 204–224.