

A POWELL-SABIN FINITE ELEMENT SCHEME FOR PARTIAL DIFFERENTIAL EQUATIONS

GIORGIO GIORGIANI¹, HERVÉ GUILLARD² AND BONIFACE NKONGA²

Abstract. In this paper are analyzed finite element methods based on Powell-Sabin splines, for the solution of partial differential equations in two dimensions. PS splines are piecewise quadratic polynomials defined on a triangulation of the domain, and exhibit a global C^1 continuity. Critical issues when dealing with PS splines, and described in this work, are the construction of the shape functions and the imposition of the boundary conditions. The PS finite element method is used at first to solve an elliptic problem describing plasma equilibrium in a tokamak. Finally, a transient convective problem is also considered, and a stabilized formulation is presented.

INTRODUCTION

Spline curves are piecewise polynomial functions with global smoothness properties. Nowadays, splines are extensively used in the graphical design industry to create smooth surfaces. 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, $i = 1, 2, \dots, n + p + 1$, 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.

The success of splines is basically due to the fact that they have very attractive and unique characteristics: they have a compact representation, they are able to represent curves and surfaces with an arbitrary level of regularity, simply increasing the polynomial degree of the basis, and they are efficient in terms of locality, that is, they can be modified locally by moving a control point, without perturbing the rest of the curve. The *control*

¹ Maison de la Simulation USR 3441, Bâtiment 565 - Digiteo - PC 190, CEA Saclay, 91191 Gif-sur-Yvette cedex, France

² Inria Sophia-Antipolis 2004 Route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex and Univ. Nice Sophia Antipolis, LJAD, UMR 7351, 06100 Nice, France

polygon of the spline is obtained connecting the control points with straight lines, and has a fundamental role in controlling the shape of the spline curve.

The use of splines as shape functions for solving partial differential equations has been subject of many researches in the recent years. For example, in the context of Isogeometric Analysis, see [1, 3], the use of splines both for the geometry representation and the approximation of the numerical solution allows to achieve a precise geometry description of the computational domain, and improves the connection between Computer Aided Design (CAD), used for geometry modeling, and numerical simulations, easing the set up of finite element models.

However, using splines as shape functions brings other advantages. For example, the additional global smoothness of the spline interpolant introduces stability to the numerical solution, in particular for highly convective equations, see [1] and [7], and also for turbulence computations, see [2].

Moreover, the use of C^1 approximations also allows to discretize higher-order derivatives (that is, derivatives of order superior to the second). This feature can be interesting in equations involving the potential of a given quantity, and allows to circumvent the divergence free constraint related to the potential function. A typical example are the MHD equations of a strongly magnetized plasma: in this case, being always $\nabla \cdot \mathbf{B} = 0$, where \mathbf{B} is the magnetic field, one can write $\mathbf{B} = \nabla \times \mathcal{A}$, where \mathcal{A} is the magnetic potential. Hence, a formulation based on the magnetic potential allows to automatically guarantee the divergence free constraint of the magnetic field.

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 solve this limitation, for example, the introduction of *T-splines* in [3].

This work is focused on another approach, based on the definition of bivariate splines on irregular triangulations, that is, Powell-Sabin (PS) splines [5, 7, 8]. PS splines are piecewise quadratic polynomial with C^1 continuity, defined on a unstructured triangulation of the domain. The major advantage with respect to tensor product B-splines is that PS splines allow a straightforward adaptive refinement of the mesh, which is a key ingredient in the simulation of anisotropic equations.

Another advantage of Powell-Sabin elements is related to the fact that the unknowns are concentrated on the nodes of the triangulation. This means 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 the Clough-Tocher elements. The consequence is that the stencil is the same for each node of the mesh, which produces a linear system matrix with a regular block shape that, on one hand, facilitate the implementation aspect and enhance the parallel performance, and on the other hand improves the efficiency of the linear system solution.

Hence, in this paper PS splines are used as shape functions in a Galerkin finite element method, for the solution of partial differential equations.

The outline of the paper is as follows. Section 1 introduces the PS splines and their representation. Section 2 deals with the geometrical and mathematical tools to define the PS elements and in particular the generation of the shape functions. Section 3 is devoted to the solution of elliptic problem by means of PS finite elements. In particular, a detailed discussion is given on the imposition of Dirichlet boundary conditions. Finally in Section 4 the method is also applied to the solution of hyperbolic problems: the isothermal Euler equations are considered, and a stabilized formulation is used for obtaining the numerical solution.

1. POWELL-SABIN SPLINES

In this section is introduced the framework and the notation to define spline functions on a triangulation of a polygonal domain. Starting from the definition of bivariate polynomials on a triangle, the goal is to obtain an interpolant of a generic function with C^1 continuity. 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

$$p(x, y) := p(\tau) = \sum_{\substack{i+j+k=2 \\ i,j,k \geq 0}} b_{i,j,k} B_{i,j,k}^2(\tau(x, y)), \quad (2)$$

where $\tau = (\tau_1, \tau_2, \tau_3)$ are the barycentric coordinates of a point $(x, y) \in \mathbb{R}^2$ and $B_{i,j,k}^2(\tau)$ are the Bernstein polynomials of degree 2 in Ω_k , that is

$$B_{i,j,k}^2 = \frac{2!}{i!j!k!} \tau_1^i \tau_2^j \tau_3^k. \quad (3)$$

The set of Bernstein polynomials $B_{i,j,k}^2(\tau)$ is a basis for the space of polynomials Π_2 , see [4], hence the six coefficients $b_{i,j,k}$ uniquely define the second order polynomial $p(x, y)$ and they 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 $\partial\Omega$. Let \mathcal{T} be a conforming triangulation of Ω with vertices \mathbf{V}_l , $l = 1, \dots, N_v$ and elements Ω_e , $e = 1, \dots, N_e$. Defining \mathcal{S}_1^2 as the linear space of piecewise quadratic polynomials on \mathcal{T} , the following interpolation problem is considered: given any set of triples (f_l, f_{xl}, f_{yl}) , $l = 1, \dots, N_v$, find $s(x, y) \in \mathcal{S}_1^2$ such that,

$$s(\mathbf{V}_l) = f_l, \quad \frac{\partial s}{\partial x}(\mathbf{V}_l) = f_{xl}, \quad \frac{\partial s}{\partial y}(\mathbf{V}_l) = f_{yl}, \quad l = 1, \dots, N_v. \quad (4)$$

It is clear that such a problem has no solution in general: in fact, problem (4) requires the imposition of nine parameters to define the second order polynomial on each triangle, while only six coefficients are available, see equation (2).

Thus, to obtain a solution to the interpolation problem (4), one possibility 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 [6] is based on the subdivision of each triangle in \mathcal{T} into six smaller triangles (*PS6-split*). Hence, the conditions in (4) are imposed only on the vertices of the original triangulation, while in the other added nodes only C_1 continuity of the interpolating function is imposed. More details can be found in [6].

The so called Powell-Sabin refinement of \mathcal{T} is denoted with \mathcal{T}^* and is obtained dividing each triangle in \mathcal{T} in six sub-triangles. The procedure to define \mathcal{T}^* is described in [7] and reads as follows.

- 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, called *PS points*, associated to each vertex. 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. The PS points are fundamental for the definition of the shape functions as explained in the next section.

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

2. POWELL-SABIN FINITE ELEMENTS

2.1. Powell-Sabin B-splines

In this section PS splines are introduced in a finite-element framework to solve partial differential equations, defining shape functions belonging to the space $S_2^1(\mathcal{T}^*)$. Hence, it is necessary to express each element $s(x, y) \in$

¹Note that (2) can be expanded as $p(\tau) = b_{2,0,0}\tau_1^2 + b_{0,2,0}\tau_2^2 + b_{0,0,2}\tau_3^2 + 2b_{1,1,0}\tau_1\tau_2 + 2b_{1,0,1}\tau_1\tau_3 + 2b_{0,1,1}\tau_2\tau_3$.

$S_2^1(\mathcal{T}^*)$ as a linear combination of PS spline basis functions, that is

$$s(x, y) = \sum_{l=1}^{N_v} \sum_{r=1}^3 c_{l,r} B_l^{(r)}(x, y), \quad (5)$$

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 (4) 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 the derivatives with respect the Cartesian axes 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 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 solving the interpolation problem (4). 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 is, basis functions that form a partition of unity on Ω , i.e.

$$\begin{aligned} B_l^{(r)}(x, y) &\geq 0 \quad \forall x, y \in \Omega, \\ \sum_{l=1}^{N_v} \sum_{r=1}^3 B_l^{(r)}(x, y) &= 1 \quad \forall x, y \in \Omega, \end{aligned}$$

which leads to the following constraints 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 [8] allows to obtain PS B-splines that form a partition of unity and with good characteristic from a numerical point of view. It 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 as a role similar to the control polygon for univariate splines in terms of control of the shape of the spline. In fact, in [8] it is shown 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 spatial plane that is always tangent to the spline surface at the vertex \mathbf{V}_l . Optimal shape functions from a numerical point of view are associated to a minimal area control triangle [8].

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

$$\begin{bmatrix} \alpha_{l,1} & \alpha_{l,2} & \alpha_{l,3} \\ \beta_{l,1} & \beta_{l,2} & \beta_{l,3} \\ \gamma_{l,1} & \gamma_{l,2} & \gamma_{l,3} \end{bmatrix} \begin{bmatrix} X_{l,1} & Y_{l,1} & 1 \\ X_{l,2} & Y_{l,2} & 1 \\ X_{l,3} & Y_{l,3} & 1 \end{bmatrix} = \begin{bmatrix} x_l & y_l & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

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. This means geometrically that all the PS points associated to \mathbf{V}_l are contained in the control triangle T_l , see Figure 1.

Hence, the procedure of computing the three shape functions for each vertex is:

- 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 solving the linear system (6).

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

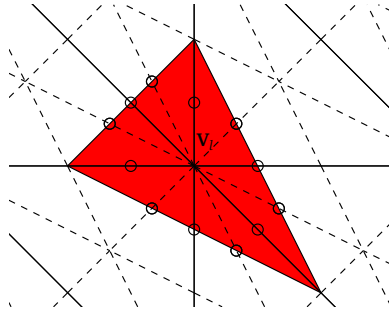


FIGURE 1. Powell-Sabin control triangle and relative Powell-Sabin points for the vertex V_l .

TABLE 1. Condition number comparison for the stiffness matrix and the mass matrix obtained with optimal and non-optimal control triangles in a square domain.

Ne	h	Stiffness matrix		Mass matrix	
		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

minimal area triangle. An optimal algorithm is proposed by O'Rourke in [9] for solving such a problem, with a computational cost $\mathcal{O}(n)$, with n the number of PS points. This is the algorithm used here in the numerical tests. In Figure 1 is depicted an example of control triangle, with the relative PS control points, obtained with the O'Rourke algorithm.

In order to understand the role of the control triangle in the definition of the basis functions, a comparison of the condition number of the linear system matrix as been carried out for three structured computational meshes. In the first case, the minimal area triangle have been used to compute the basis. In the second case, the minimal area triangle has been expanded moving (to a distance equal to five times the original one) each vertex along the line joining the vertex with the node to which the triangle belongs. Table 1 shows the results: it is possible to notice how increasing the control triangle area degradetes the conditon number of mass matrix and the stiffness matrix.

3. GALERKIN APPROXIMATION OF THE GRAD-SHAFRANOV EQUATION FOR MHD EQUILIBRIUM IN TOKAMAKS

In this section the PS finite element method is used to solve the Grad-Shafranov equation. The Grad-Shafranov equation is a 2D elliptic non-linear partial differential equation describing the equilibrium of a tokamak plasma. It is obtained reducing the ideal MHD equations to 2D using the toroidal axisymmetry and reads

$$\begin{cases} \Delta u - \frac{1}{x} \frac{\partial u}{\partial x} = F(u, x, y), & \text{in } \Omega, \\ u = u_D & \text{in } \partial\Omega_D, \\ \nabla u \cdot \mathbf{n} = u_N & \text{in } \partial\Omega_N, \end{cases} \quad (7)$$

where F is a given function, \mathbf{n} is the exterior normal to the boundary, u_D is the imposed value on the Dirichlet boundary and u_N is the imposed flux on the Neumann boundary. Multiplying for a test function v and

integrating by parts, the weak form of (7) is obtained

$$\int_{\Omega} \left(\nabla u \cdot \nabla v + \frac{1}{x} \frac{\partial u}{\partial x} v \right) d\Omega - \int_{\partial\Omega} v \nabla u \cdot \mathbf{n} d\Gamma = - \int_{\Omega} v F d\Omega. \quad (8)$$

The finite element solution of (8) requires looking for an approximation of u , called u_h , in a finite dimensional space. In this case, the finite dimensional space is the space of PS B-splines, that is, $u_h \in S_2^1(\mathcal{T}^*)$, hence

$$u \approx u_h = \sum_{l=1}^{N_v} \sum_{r=1}^3 U_{l,r} B_l^{(r)}(x, y), \quad (9)$$

where $U_{r,l}$ are the coefficients of the finite element approximation. In the Galerkin method the test function v are chosen in the same finite dimensional space, that is $v \in S_2^1(\mathcal{T}^*)$. Then, the discretized weak form leads to the evaluation of the following integrals

$$\begin{aligned} & \int_{\Omega} \left(\nabla B_i^{(q)}(x, y) \cdot \nabla B_j^{(r)}(x, y) + \frac{\partial B_j^{(r)}(x, y)}{\partial x} B_i^{(q)}(x, y) \right) d\Omega, \\ & \int_{\partial\Omega} B_i^{(q)}(x, y) \nabla B_j^{(r)}(x, y) \cdot \mathbf{n} d\Gamma, \\ & \int_{\Omega} B_i^{(q)}(x, y) F d\Omega. \end{aligned}$$

The local support of the PS B-splines guarantees that the final linear system is in sparse form. Moreover, choosing the minimal area triangles leads to shape functions with good numerical characteristic, hence the condition number of the linear system matrix will be low, see also Table 1.

3.1. Imposition of the boundary conditions

The problem of imposing Dirichlet boundary conditions deserves particular attention in the framework of PS finite elements. Only polygonal domain are considered here, being the case of curved boundary subject of future research. A careful choice of the control triangle associated to the boundary node \mathbf{V}_l simplify the task of imposing Dirichlet boundary conditions. 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 (6), the following relations are derived

$$\alpha_{l,1} \mathbf{Q}_{l,1} + \alpha_{l,2} \mathbf{Q}_{l,2} + \alpha_{l,3} \mathbf{Q}_{l,3} = \mathbf{V}_l, \quad (10)$$

$$\beta_{l,1} \mathbf{Q}_{l,1} + \beta_{l,2} \mathbf{Q}_{l,2} + \beta_{l,3} \mathbf{Q}_{l,3} = \mathbf{e}_x, \quad (11)$$

$$\gamma_{l,1} \mathbf{Q}_{l,1} + \gamma_{l,2} \mathbf{Q}_{l,2} + \gamma_{l,3} \mathbf{Q}_{l,3} = \mathbf{e}_y, \quad (12)$$

where \mathbf{e}_x and \mathbf{e}_y are respectively the unity vector in the x and y directions. Equation (10) 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 (11) and (12) mean that $(\beta_{l,1}, \beta_{l,2}, \beta_{l,3})$ 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 \mathbf{V}_l forms an angle different from π , if the control triangle has two sides aligned with the two boundary edges as in Figure 2 (left), from equation (10) derives that only one shape function has value different from zero in \mathbf{V}_l . This allows to directly set the Dirichlet value to the unknown associated to the non-zero shape function. Moreover, from (11) and (12) derives that the other two shape functions have zero derivative in the directions the two edges concurring in \mathbf{V}_l . In particular, using the notation depicted in Figure 2 (left) where

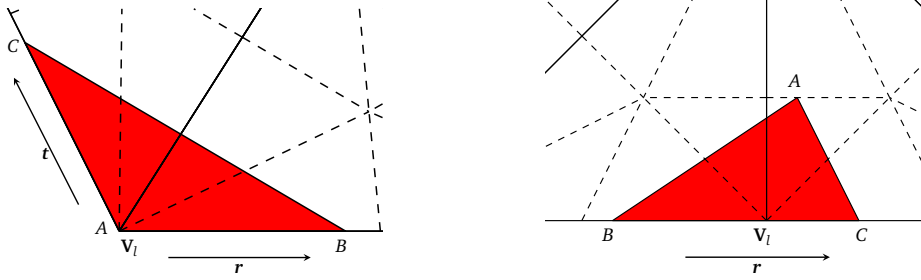


FIGURE 2. Control triangles for imposing Dirichlet boundary conditions: on an angle different from π (left) and equal to π (right).

triangle vertices are now called **A**, **B** and **C**, it results

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

Denoting with \hat{u} the \mathcal{L}^2 projection of the Dirichlet values u_D on the space of univariate quadratic splines on $\partial\Omega_D$, the resulting equations relative to the unknowns of the vertex \mathbf{V}_l are

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

that is

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

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

In the case \mathbf{V}_l belongs to a plane angle, if the control triangle is aligned with the boundary as in Figure 2 (right), there is one shape function with zero value on \mathbf{V}_l , see equation (10). Hence, the unknown related to this shape function is not affected by the boundary condition. Moreover, again equations (11) and (12) guarantee

TABLE 2. Convergence study for the Grad-Shafranov problem in a square domain.

Ne	h	e_1	Order	e_2	Order	e_∞	Order
200	70.71E-03	0.472E-05	-	0.637E-05	-	0.298E-04	-
800	35.36E-03	0.588E-06	3.01E+00	0.785E-06	3.02E+00	0.387E-05	2.95E+00
3200	17.68E-03	0.733E-07	3.00E+00	0.975E-07	3.01E+00	0.492E-06	2.97E+00
12800	8.84E-03	0.916E-08	3.00E+00	0.121E-07	3.00E+00	0.620E-07	2.99E+00

that the same shape function has zero derivative in the direction parallel to the boundary. Thus, with reference to Figure 2 (right), results

$$\alpha_{l,A} = 0, \quad \alpha_{l,B} \neq 0, \quad \alpha_{l,C} \neq 0, \\ \left\{ \begin{array}{l} \beta_{l,A} \\ \gamma_{l,A} \end{array} \right\} \cdot \mathbf{r} = 0,$$

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

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

that is

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

Neumann boundary conditions do not need to be enforced on the nodes as in other Galerkin approaches, but are weakly imposed by the boundary integral in (8).

3.2. Numerical results

The case $F = 1 + x^2$ is considered, with exact solution

$$u = \frac{1}{2} \left(1 - y^2 - \frac{1}{4} (x^2 - 1)^2 \right).$$

Only Dirichlet boundary conditions are considered. The problem is solved in a square domain $\Omega = [0, 1] \times [0, 1]$ and in a drop-shaped domain. An example of meshes and the control triangles are shown in Figure 3 for both cases. The boundary conditions are computed using the exact solution and imposed on the mesh sides (values and derivatives along the boundary edges). Hence, no curved geometry are considered.

In Table 2 are shown the convergence rates of the error $u - u_h$ in different norms, while the element size is estimated as $h \simeq \sqrt{\|\Omega\|/N_e}$, being $\|\Omega\|$ the area of the domain. The expected third order convergence rate is achieved in all the norms.

TABLE 3. Convergence study for the Grad-Shafranov problem in a drop shaped domain

Ne	h	e_1	Order	e_2	Order	e_∞	Order
120	88.37E-03	0.125E-04	-	0.183E-04	-	0.103E-03	-
416	47.57E-03	0.191E-05	3.03E+00	0.271E-05	3.09E+00	0.202E-04	2.63E+00
1656	23.86E-03	0.252E-06	2.94E+00	0.340E-06	3.01E+00	0.269E-05	2.92E+00
6636	11.92E-03	0.299E-07	3.07E+00	0.400E-07	3.09E+00	0.279E-06	3.26E+00

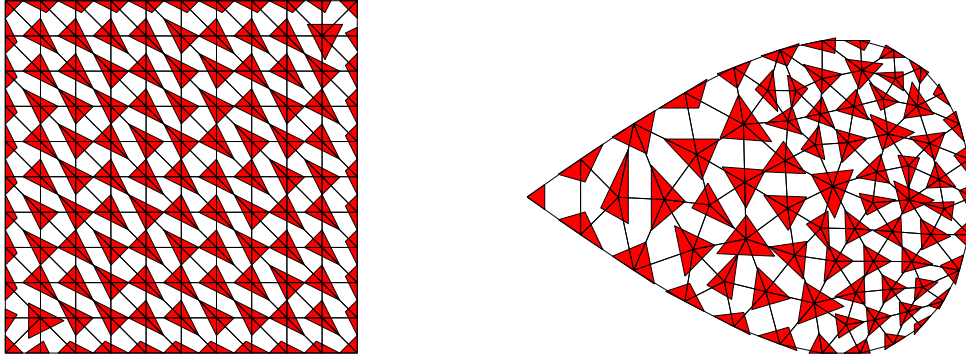


FIGURE 3. Example of control triangles on a square structured mesh (left) and on an unstructured mesh (right).

4. STABILIZED APPROXIMATION OF THE ISOTHERMAL EULER EQUATIONS

In this section the isothermal form of the Euler equation is considered,

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) &= \mathbf{0} \end{aligned} \quad (13)$$

where ρ is the density, \mathbf{u} is the velocity vector, \mathbf{I} is the identity matrix and $p = a^2 \rho$ is the pressure. The sound speed a is considered constant in the isothermal approximation.

Problem (13) can be rewritten in conservative form, that is, in 2D,

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{0}, \quad (14)$$

where \mathbf{U} is the vector of conservative variables and \mathbf{F} is the flux tensor,

$$\mathbf{U} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \end{Bmatrix}, \quad \mathbf{F} = \{ \mathbf{F}_x \ \mathbf{F}_y \}, \quad \mathbf{F}_x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \end{Bmatrix}, \quad \mathbf{F}_y = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \end{Bmatrix},$$

where u and v are the velocity respectively in the x and y directions.

A stabilized formulation based on PS splines is considered to solve problem (14). Stabilized formulations are obtained adding extra terms in the Galerkin weak form, with the goal of reducing the instabilities with the introduction of artificial viscosity. Here, two stabilization terms are considered.

The first term is the streamline upwind Petrov-Galerkin (SUPG) term, used to produce a stable upwind discretization without introducing excessive numerical dissipation. The SUPG method was introduced by T.

Hughes in [11] and it is one of the most established stabilized formulations in finite element flow computations. The SUPG method introduces a certain amount of artificial viscosity in the streamline direction only.

The second stabilization term is a shock capturing term, and it is used to prevent oscillations around the discontinuities arising in the solution in supersonic regimes. These spurious oscillations might lead to severe accuracy loss or stability problems. Differently from the SUPG stabilization term, the shock-capturing stabilization term introduce an isotropic artificial diffusion, but only in a sharp zone surrounding the discontinuities.

To simplify the presentation, in an abuse of notation, the same symbol is used for the numerical approximation, belonging to the finite dimensional space, and the exact solution \mathbf{U} . Hence, the stabilized finite element problem corresponding to (14) reads: find $\mathbf{U} \in [S_2^1(\mathcal{T}^*)]^3$ such that

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} \cdot \mathbf{V} \, d\Omega - \int_{\Omega} \mathbf{F} : \nabla \mathbf{V} \, d\Omega + \int_{\partial\Omega} \mathbf{F} \mathbf{V} \cdot \mathbf{n} \, d\Gamma + \mathcal{D}_{SUPG} + \mathcal{D}_{SC} = 0, \quad (15)$$

for all $\mathbf{V} \in [S_2^1(\mathcal{T}^*)]^3$. High-order time discretization is not of concern here, hence a simplified version of the SUPG stabilization term is used

$$\mathcal{D}_{SUPG} = \sum_{e=1}^{N_e} \int_{\Omega_e} (\bar{\bar{\mathbf{A}}} \nabla \mathbf{U}) : (\tau_{SUPG} \bar{\bar{\mathbf{A}}} \nabla \mathbf{V}) \, d\Omega_e, \quad (16)$$

which does not take into account the complete residual of the original equation, but only the convective term.

The shock capturing stabilization term is

$$\mathcal{D}_{SC} = \sum_{e=1}^{N_e} \int_{\Omega_e} \nabla \mathbf{U} : (\tau_{SC} \nabla \mathbf{V}) \, d\Omega_e. \quad (17)$$

The third order tensor $\bar{\bar{\mathbf{A}}}$ in (16) is the Jacobian of the flux, that is

$$\bar{\bar{\mathbf{A}}} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}},$$

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

$$\mathbf{A}_x = \frac{\partial \mathbf{F}_x}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 1 & 0 \\ a - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix}, \quad \mathbf{A}_y = \frac{\partial \mathbf{F}_y}{\partial \mathbf{U}} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ a - v^2 & 0 & 2v \end{pmatrix}.$$

The terms τ_{SUPG} and τ_{SC} are stabilization matrices. Various options to compute these terms can be found in literature. In this work, for the SUPG stabilization term a diagonal matrix is used, $\tau_{SUPG} = \tau \mathbf{I}$, and a constant value τ is used in the whole mesh, $\tau = dt/2$, being dt the time step.

For the shock-capturing operator, the choices described in [12, 13] are followed: three sensors are used to set up the artificial viscosity to be used to stabilize discontinuities. The first sensor is based on the relative density variation and it is defined, in each element, as

$$[\widetilde{\nabla \rho}]_e = \sum_{r=1}^3 \sum_{q=1}^3 \left| \frac{\nabla \rho \cdot \nabla B_r^{(q)}}{|\nabla \rho|} \right|.$$

The second sensor is based on the relative gradient according to the principle axes and is written

$$[\widetilde{\nabla \mathbf{U}}]_e = \sum_{d=1}^2 \left(\left\| \frac{\partial \mathbf{U}}{\partial x_d} \right\|^2 \right)^{\frac{1}{2}},$$

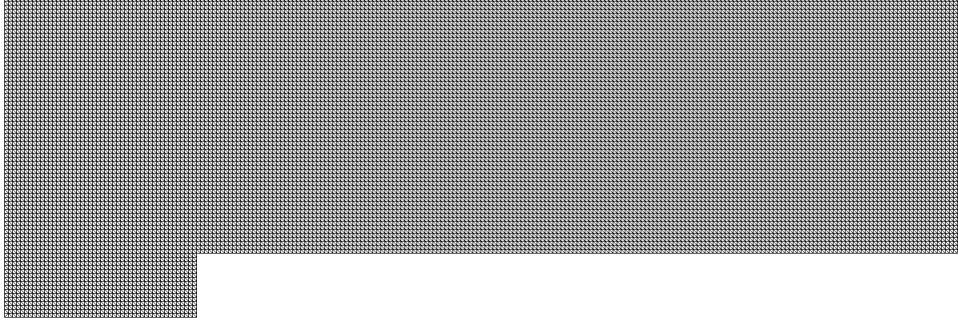


FIGURE 4. Computational mesh for the forward-facing step test case, with $h = 1/80$.

while the third sensor takes into account the local relative convective flux

$$[\widetilde{\mathbf{A}\nabla\mathbf{U}}]_e = \left\| \widetilde{\mathbf{A}\nabla\mathbf{U}} \right\|.$$

Hence, the shock capturing artificial viscosity is defined as a combination of the three sensors with a managing parameter β as follows

$$\tau_{\text{SC}}(\beta) = [\widetilde{\nabla\rho}]_e [\widetilde{\nabla\mathbf{U}}]_e^{-(2-\beta)} [\widetilde{\mathbf{A}\nabla\mathbf{U}}]_e^{-\beta},$$

where $\beta = 2$ is used for smooth shocks and $\beta = 1$ for strong shocks. A compromise between the two definitions is proposed in [13],

$$\tau_{\text{SC}} = \frac{\tau_{\text{SC}}(\beta = 1) + \tau_{\text{SC}}(\beta = 2)}{2},$$

and it is used in this work.

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

This is classical test case for 2D numerical codes for supersonic flows and a detailed description can be found in [14]. It consists in a supersonic flow entering an infinite long wind tunnel with a step. The interaction of the supersonic flow with the step and the tunnel walls creates a typical patten of shock reflections.

The numerical set up of the simulation is as follows: the computational domain has dimensions 3×1 length units, and the step is 0.2 length units high and it is located at 0.6 length units from the left boundary. An isothermal flow with unitary sound speed $a = 1$ is entering from the left boundary with a uniform horizontal velocity at Mach 3 and density $\rho = 1$. The tunnel walls are non-slip boundaries, while the right boundary is a free exit. The simulation is started from flow at rest everywhere, with density $\rho = 1$. No particular treatment is done for the corner of the step, which represent a singular point may introduce numerical errors. The suppression of this spurious errors is subject of the ongoing investigation.

The time discretization is performed with the first order explicit scheme, and a time step $dt = 10^{-5}$ is used. The computational mesh has an element size of $h = 1/80$, and it is shown in Figure 4. Results of the simulation are depicted in Figure 5 for the density variable. Despite the simple choice regarding the SUPG stabilization parameter, a stable solution is obtained, confirming the stabilizing properties of the spline interpolant found in literature. Sharp resolution of the shock can be appreciated from the figures, even thought further investigations are needed.

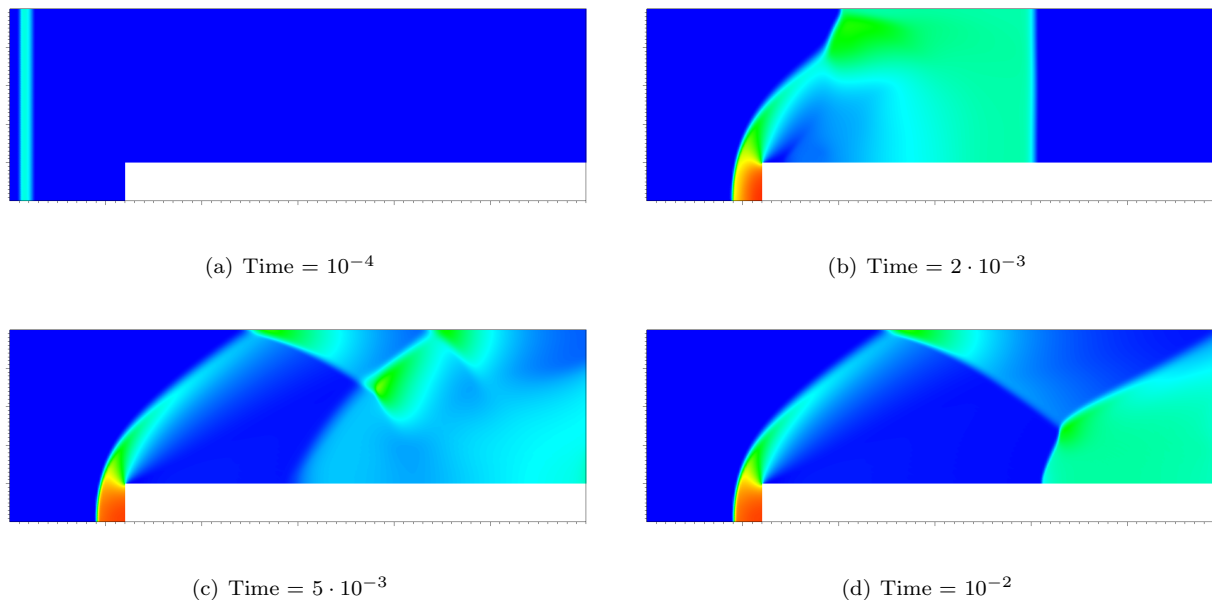


FIGURE 5. Solution of the density for the forward-facing step in different time instants.

5. CONCLUSIONS

In this paper is presented a finite element method based on Powell-Sabin (PS) splines. The mathematical and geometrical tools to define the PS shape functions are presented, in particular, the subdivision of the elements that allows to define the PS splines as piecewise second order polynomials in each sub-triangle, the definition of the PS points and the control triangles.

A Galerkin method is presented for the solution of elliptic problems on polygonal domains. The problem of imposing Dirichlet boundary conditions is examined. The method is used to solve the Grad-Shafranov equation on structured and unstructured meshes. The convergence tests give optimal third order slope in different error norms. The ongoing work include the description of curved geometry and the imposition of the boundary conditions on curved elements.

Finally the PS finite element method is used to solve the unsteady isothermal Euler equations. A stabilized formulation is adopted, with SUPG and shock-capturing terms. A forward-facing step with Mach 3 flow is used to validate the method, using classic stabilization parameters. Satisfactory results are obtained in terms of stabilization of the solution and resolution of the discontinuities. Further investigation on stabilization parameters specifically designed for PS finite element is needed.

REFERENCES

- [1] T. J. R. Hughes, J. A. 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.
- [2] Y. Bazilevs, V.M. Calo, J. A. Cottrell, T. J. R. 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.
- [3] Y. Bazilevs, V.M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton and M. A. Scott and T. W. Sederberg, Isogeometric analysis using T-splines, *Comput. Methods Appl. Mech. Eng.* 199 (5-8) (2010) 229–263.
- [4] M. J. Lai, L. L. Schumaker, *Spline Functions on Triangulations*, Cambridge University Press, 2007.
- [5] N. Jaxon, X. Qian, Piecewise quadratic approximations on triangles, *Computer-Aided Design* 46 (2014) 45–57.
- [6] M. J. D. Powell, M. A. Sabin, Piecewise quadratic approximations on triangles, *ACM Trans. Math. Softw.* 3 (4) (1977) 316–325.

- [7] 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 (2012) 132–148.
- [8] P. Dierckx, On calculating normalized Powell-Sabin B-splines, *Comput. Aided Geom. D.*, 15 (1) (1997) 61–78.
- [9] J. O'Rourke, A. Aggarwal, S. Maddila, M. Baldwin, An optimal algorithm for finding minimal enclosing triangles, *J. Algorithm.*, 7 (2) (1986) 258–269
- [10] H. Speleers, P. Dierckx, S. Vandewalle, Powell-Sabin splines with boundary conditions for polygonal and non-polygonal domains, *J. Comput. Appl. Math.*, 206 (1) (2007) 55–72
- [11] A. N. Brooks and T. J. R. 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 (1990) 199–259
- [12] C. Wervaecke, H. Beaugendre, B. Nkonga, A fully coupled RANS Spalart-Allmaras SUPG formulation for turbulent compressible flows on stretched-unstructured grids, *Comput. Methods Appl. Mech. Eng.*, 233-236 (2012) 109–122
- [13] T. E. Tezduyar, M. Senga, Stabilization and shock-capturing parameters in SUPG formulation of compressible flows, *Comput. Methods Appl. Mech. Eng.*, 195 (2006) 1621–1632
- [14] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.*, 54 (1) (1984) 115–173