A high-order non field-aligned approach for the discretization of strongly anistropic diffusion operators in magnetic fusion

G Giorgiani, H. Bufferand, F. Schwander, E. Serre, P. Tamain

A high-order non field-aligned approach for the discretization of strongly anistropic diffusion operators in magnetic fusion

G. Giorgiani[a,], H. Bufferand[b], F. Schwander[a], E. Serre[a], P. Tamain[b]

[a] *Aix-Marseille Univ., CNRS, Centrale Marseille, M2P2 Marseille, France*
[b] *IRFM, CEA Cadarache, F-13108 St. Paul-lez-Durance, France*

Abstract

In this work we present a hybrid discontinuous Galerkin scheme for the solution of extremely anisotropic diffusion problems arising in magnetized plasmas for fusion applications. Unstructured meshes, non-aligned with respect to the dominant diffusion direction, allow an unequalled flexibility in discretizing geometries of any shape, but may lead to spurious numerical diffusion. Curved triangles or quadrangles are used to discretize the poloidal plane of the machine, while a structured discretization is used in the toroidal direction. The proper design of the numerical fluxes guarantees the correct convergence order at any anisotropy level. Computations performed on well-designed 2D and 3D numerical tests show that non-aligned discretizations are able to provide spurious diffusion free solutions as long as high-order interpolations are used. Introducing an explicit measure of the numerical diffusion, a careful investigation is carried out showing an exponential increase of this latest with respect to the non-alignment of the mesh with the diffusion direction, as well as an exponential decrease with the polynomial degree of interpolation. A brief assessment of the method with respect to two finite-difference schemes using non-aligned discretization, but classically used in fusion modeling, is also presented.

Keywords: Anisotropic diffusion, hybrid discontinuous Galerkin, high-order method, tokamak edge plasma, fusion.

PROGRAM SUMMARY

Program Title: Laplace-HDG (Laplace Hybrid Discontinuous Galerkin)
Licensing provisions: GPLv3
Programming language: Fortran 95
Nature of problem: Anisotropic Laplace problem in 2D with Dirichlet boundary conditions
Solution method: Hybrid discontinuous Galerkin scheme

1. Introduction

Anisotropic diffusion problems arise in a variety of applications in physics and engineering, such as, for example, hydrodynamic transport [1], image processing [2, 3, 4], diffusion in porous media [5], electromagnetic propagation [6], biomedical applications [7, 8], etc.

The typical mathematical model for the convection-diffusion of a scalar quantity u in $\mathbb{R}^3$ writes as:

$$\frac{\partial u}{\partial t} + \boldsymbol{a} \cdot \boldsymbol{\nabla} u - \boldsymbol{\nabla} \cdot (\boldsymbol{\mathcal{K}} \boldsymbol{\nabla} u) = f \tag{1}$$

where $\boldsymbol{\mathcal{K}}$ is an anisotropic diffusion tensor, $\boldsymbol{a}$ is the convection velocity and f is a source term. In this work we focus on magnetized plasmas for fusion applications, where the scale separation induced by the magnetic field produces conduction coefficients that are several order of magnitudes larger in the parallel direction than in the perpendicular one. In this context, the usual strategy to solve problem (1) is based on finite-difference schemes with structured meshes aligned with the magnetic lines. The separation of the dynamics in the two directions is therefore guaranteed by the space discretization, and no spurious cross-field diffusion occurs, see [9]. Flux aligned coordinates are still used in the fusion community, see for example [10]. However, they require simplifications on the magnetic field topology: in fact, often the magnetic lines wind around a toroidal surface never closing on themselves (the magnetic surface are, in general, irrational). Therefore, the possibility of using completely aligned meshes in general tokamak simulations is excluded.

Hence, it is more and more common in the fusion community the use of toroidal coordinates for the spatial discretization of (1) in axisymmetric geometries mimicking the topology of fusion devices such as tokamaks (neglecting the use of non-symmetric components such as toroidal limiters or antennas). This entails the need of two kinds of discretizations: a poloidal one and a toroidal one, see Fig. 1. While the poloidal discretization carries the burden of describing the geometry of the reactor chamber and the plasma facing components, the toroidal one can be simply thought of a periodic extrusion in the toroidal direction of the poloidal discretization. Within this approach, field aligned coordinates *only in the poloidal plane* are still widely used for both 2D and 3D computations. However, a precise geometry description is cumbersome with this technique: in [11], for example, the tokamak chamber is described only until the last flux surface before the wall. In [12], a penalization technique is used to model the real chamber geometry. Unfortunately, penalization techniques are usually only first order accurate with respect to the geometry definition and also the imposition of the boundary conditions. Moreover, even if the use of field aligned coordinates in the poloidal plane allows to threat complicated magnetic topologies

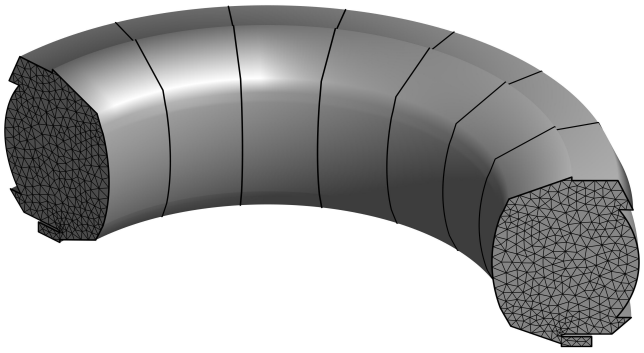


Figure 1: Example of the 3D discretization of a toroidal geometry with triangles in the poloidal plane.

and X-point configurations, using domain decomposition techniques to avoid the problem of the singularity at the X-point, it still introduces low resolution areas due to the flux expansion at the singularity. Finally, it also prevents the simulation of evolving magnetic equilibria, which would require expensive on the fly re-meshing at each time step.

In the objective of relaxing the need of aligning the grid, several schemes based on non-aligned meshes have been proposed in the literature, mostly in the context of finite-differences (see details in [13]). Schemes based on *non-aligned discretizations* [14, 15] improve the computation of the fluxes by a proper choice of the stencil with respect to a naive discretization, or rely on high-order finite-difference schemes to reduce the numerical diffusion, such as in [16]. On the other hand, schemes based on *aligned discretizations* [17, 18, 19, 20] employ interpolations on the magnetic field lines to perform finite-difference discretizations along the anisotropy directions.

In the perpendicular plane, such *aligned discretizations* seem so far to be limited to structured meshes for simplicity and efficieny of implementation. This tends thus to complicate the description of realistic geometries and the imposition of the boundary condition. Let's notice however that such *aligned discretizations* in the parallel direction can be associated to unstructured mesh in the perpendicular plane, see for example in Ref. [21].

Switching to general non-structured non-aligned meshes (of triangles or quads) allows to exploit a huge flexibility in terms of geometry description and local refinement. An intermediate approach is proposed in [22], where a technique aimed at aligning a generic triangular mesh with respect to a given flow is developed in the framework of finite-volumes. The technique is shown to reduce the numerical diffusion errors for 2D computations. Triangular grid alignment in the framework

of magnetic fusion is an active field, see for example [23, 24], and it can be seen as another opportunity to reduce the numerical errors and the computational cost, for simulations where the magnetic equilibrium is fixed and therefore the mesh may be designed to mimick the elongated structures appearing in the field direction. For this reason, a discussion on the definition of alignment for triangular meshes is proposed in Sec. 5. However, in this work the focus is put in reducing the diffusion for non-aligned grids by means of increasing the polynomial order of interpolation. Some effort in this sense has already been done. In [25], a spectral element method is used to solve a heat diffusion problem in plasmas, showing that high-order elements yield a given accuracy with less total degrees of freedom than lower-order elements. In [26], a second and fourth order finite volume schemes are tested for solving a temporal evolution diffusion problem, revealing that the fourth order scheme is more efficient than the second order one for reducing the numerical perpendicular diffusion. High-order finite-volumes schemes however suffer from large stencils, preventing the use of unstructured meshes. In fact, all the tests in [26] are performed using structured meshes. A well-established solution to that is to localize the high-order interpolation *inside* each element: the discontinuous Galerkin (DG) scheme provides this approach. In [27], three DG schemes are used to discretize the parallel diffusion operator, showing exponential decrease of the numerical diffusion as the interpolation order is increased.

Introduced by Cockburn in [28], the hybrid discontinuous Galerkin scheme (HDG) retains the interesting characteristic of classic DG schemes, such as stability and local conservation, while reducing the size of the assembled matrix produced by the spatial discretization thanks to the hybridization of the primal unknown, which allows to express the nodal values of the solution in terms of the nodal values of the trace unknown, defined on the element borders. In [29], a HDG scheme is proposed for the simulation of anisotropic and non-homogeneous media arising in the study of flow through porous media. The results are encouraging for mild anisotropy values considered, and are extended here to extreme anisotropies arising in fusion applications. In [30], a first application of the HDG scheme to an isothermal plasma-transport model in tokamaks is presented. The scheme is extended in [31] to a non-isothermal model containing parallel diffusion operators. Encouraging results in [31] stimulate the deeper investigation proposed in this manuscript on the discretization of parallel diffusion operators with non-aligned high-order schemes.

The following Sec. 2 introduces the anisotropic convection-diffusion problem to be solved. Following our recent work [30], a HDG discretization suited to handle highly anisotropic diffusion operators is proposed in Sec. 3. A discussion on the design of the stabilization parameter is presented in Sec. 4. In Sec. 5, the definition

of alignment for triangular meshes is considered, with particular emphasis on the importance of high-order computations on the control of the numerical diffusion in case of non-aligned meshes. In the following three sections is investigated the impact of the mesh alignment and the element type (quadrangles or triangles) on the solution accuracy, using 2D and 3D well-designed numerical tests. Sec.6 concerns the discretization of an annulus domain mimicking the poloidal section of a tokamak. In Sec.7, the impact of the disalignment in the toroidal discretization is investigated and a brief assessment of the method is proposed with respect to the classic and the Günter's finite-difference schemes, both being based on non-aligned discretization but have been used in magnetic fusion modeling. Finally, a 3D test in a realistic tokamak geometry is studied in Sec.8. Concluding remarks and perspectives are given in Sec.9.

2. The anisotropic convection-diffusion problem

A general time evolving convection-diffusion problem is given by Eq. 1, defined in an open bounded domain $\Omega \in \mathbb{R}^3$ of boundary $\partial\Omega$, with both Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries:

$$\begin{aligned} \frac{\partial u}{\partial t} + \boldsymbol{a}\cdot\boldsymbol{\nabla}u - \boldsymbol{\nabla}\cdot(\boldsymbol{\mathcal{K}}\boldsymbol{\nabla}u) &= f \qquad \text{in } \Omega\times]0,T_f[, \\ u &= u_D \qquad \text{on } \partial\Omega_D\times]0,T_f[, \\ \boldsymbol{\mathcal{K}}\boldsymbol{\nabla}u &= g_N \qquad \text{on } \partial\Omega_N\times]0,T_f[, \\ u(\boldsymbol{x},0) &= u_0 \qquad \text{in } \Omega, \end{aligned} \tag{2}$$

where u_0 is the initial field, u_D is the prescribed value on the Dirichlet boundary and g_N is the prescribed flux on the Neumann boundary, and T_f is the final time. The diffusion tensor $\boldsymbol{\mathcal{K}}$ can be written as

$$\boldsymbol{\mathcal{K}} = k_\parallel \boldsymbol{b}\otimes\boldsymbol{b} + k_\perp\Big(\overline{\overline{\boldsymbol{I}}} - \boldsymbol{b}\otimes\boldsymbol{b}\Big), \tag{3}$$

where the vector $\boldsymbol{b} \in \mathbb{R}^3$ represents the anisotropy direction, $k_\parallel$ and $k_\perp$ are respectively the parallel and perpendicular diffusion coefficients, and $\overline{\overline{\boldsymbol{I}}}$ is the identity matrix.

In the context of magnetized plasmas in tokamak, the vector $\boldsymbol{b} = \boldsymbol{B}/\|\boldsymbol{B}\|$ defines the magnetic field direction, with $\boldsymbol{B} = (B_p, B_t)$ the magnetic field and B_p and B_t its poloidal and the toroidal components, respectively. In tokamaks typically $|B_p| << |B_t|$. Moreover, the anisotropy level is such that $k_\parallel$ is several orders of magnitude larger than $k_\perp$.

3. The Hybrid Discontinuous Galerkin (HDG) solver

The HDG scheme presented here extends our previous work in Ref. [30] to deal with highly anisotropic diffusion operator.

3.1. The discontinuous Galerkin setting

The domain Ω is partitioned in $\mathtt{n_{el}}$ disjoint elements Ω_i with boundaries $\partial\Omega_i$, such that

$$\overline{\Omega} = \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j,$$

where the symbol $\bar{\cdot}$ indicates the topological closure of the set, and the union of all $\mathtt{n_{fc}}$ faces (sides for 2D) is denoted as

$$\Sigma = \bigcup_{i=1}^{\mathtt{n_{el}}} \partial\Omega_i.$$

In the discontinuous setting, problem (2) can be re-written as a system of first order partial differential equations with some additional jump conditions, that is, for $i = 1, \ldots, \mathtt{n_{el}}$

$$\left.\begin{aligned} \frac{\partial u}{\partial t} - \boldsymbol{a}\cdot\boldsymbol{q} + \boldsymbol{\nabla}\cdot(\boldsymbol{\mathcal{K}}\boldsymbol{q}) &= f \\ \boldsymbol{q} + \boldsymbol{\nabla} u &= 0 \end{aligned}\right\} \quad \text{in } \Omega_i \times]0, T_f[, \tag{4a}$$

$$u = u_D \quad \text{on } \partial\Omega_i \cap \partial\Omega_D \times]0, T_f[, \tag{4b}$$

$$-\boldsymbol{\mathcal{K}}\boldsymbol{q} = g_N \quad \text{on } \partial\Omega_i \cap \partial\Omega_N \times]0, T_f[, \tag{4c}$$

$$u(\boldsymbol{x}, 0) = u_0, \quad \text{in } \Omega_i, \tag{4d}$$

and

$$[\![u\boldsymbol{n}]\!] = \boldsymbol{0} \quad \text{on } \Sigma \backslash \partial\Omega, \tag{4e}$$

$$[\![\boldsymbol{q}\cdot\boldsymbol{n}]\!] = 0 \quad \text{on } \Sigma \backslash \partial\Omega, \tag{4f}$$

where $\boldsymbol{n}$ is the outer normal to the element face. The *jump* $[\![\cdot]\!]$ operator is defined at each internal face of Σ, i.e. on $\Sigma \backslash \partial\Omega$, using values from the elements to the left and right of the face (say, Ω_i and Ω_j), namely

$$[\![\circledcirc]\!] = \circledcirc_i + \circledcirc_j,$$

and always involving the normal vector $\boldsymbol{n}$, see Ref. [32] for details. Within this setting, equation (4e) imposes the continuity of the unknowns across interior faces, while equation (4f) imposes the continuity of the normal component of the flux.

The added value of HDG, with respect to DG, is the possibility of setting up a linear system involving the nodal values of the so-called *trace unknown*, which is defined on the mesh skeleton, that is the union of all faces Σ. Here, the trace unknown is denoted as $\hat{u}(\boldsymbol{x})$. The introduction of this new variable $\hat{u}(\boldsymbol{x})$ allows to define two types of problems: a local problem for each element and a global one for all faces.

3.2. The local problem

The local element-by-element problem corresponds to the anisotropic convection-diffusion equation in each element, with imposed Dirichlet boundary conditions on the element boundary. These imposed boundary values correspond to the unknown $\hat{u}(\boldsymbol{x})$ for $\boldsymbol{x} \in \Sigma$.

The local element-by-element problem can be solved to determine $\boldsymbol{q}$ and u in terms of the imposed $\hat{u}(\boldsymbol{x})$ on the mesh skeleton Σ. Thus, for $i = 1, \ldots, \mathtt{n_{el}}$ the local HDG problem is

$$\left.\begin{aligned} \frac{\partial u}{\partial t} - \boldsymbol{a}\cdot\boldsymbol{q} + \boldsymbol{\nabla}\cdot(\boldsymbol{\mathcal{K}}\boldsymbol{q}) &= f \\ \boldsymbol{q} + \boldsymbol{\nabla} u &= 0 \end{aligned}\right\} \quad \text{in } \Omega_i, \tag{5a}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) \quad \text{in } \Omega_i, \tag{5b}$$

$$u = \hat{u} \quad \text{on } \partial\Omega_i. \tag{5c}$$

Since the unknown $\hat{u}$ is single valued on Σ, the same Dirichlet boundary condition is imposed to the left and right element of a given face. Consequently, the continuity of the unknowns (eq. (4e)) is ensured by eq. (5c).

The approximated solution is obtained introducing a finite-element discretization: two types of finite dimensional spaces must be defined, one for functions in the elements interior and another for trace functions, namely

$$\begin{aligned}
\mathcal{V}_h^t &:= \left\{ v_h : v_h(\cdot, t) \in \mathcal{V}_h \text{ for any } t \in [0, T_f] \right\}, \text{ with} \\
\mathcal{V}_h &:= \left\{ v_h \in \mathcal{L}^2(\Omega) : v_h|_{\Omega_i} \in \mathcal{P}^p(\Omega_i) \text{ for } i = 1, \ldots, \mathtt{n_{el}} \right\}, \\
\Lambda_h^t(u_D) &:= \left\{ \hat{v}_h : \hat{v}_h(\cdot, t) \in \Lambda_h(u_D) \text{ for any } t \in [0, T_f] \right\}, \text{ with} \\
\Lambda_h(u_D) &:= \left\{ \hat{v}_h \in \mathcal{L}^2(\Sigma) : \hat{v}_h|_{\Gamma_i} \in \mathcal{P}^p(\Gamma_i) \text{ for } i = 1, \ldots, \mathtt{n_{fc}}, \hat{v}_h = \mathbb{P}_\partial u_D \text{ on } \partial\Omega_D \right\},
\end{aligned}$$

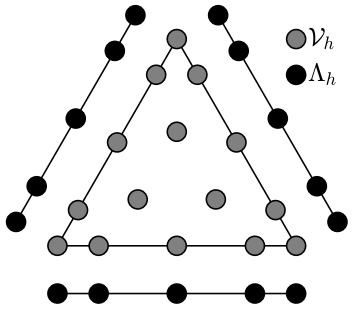


Figure 2: Nodes representing the spaces $\mathcal{V}_h$ and Λ_h for a polynomial degree of order $p = 4$.

where Γ_i is one face of the element and $\mathcal{P}^p$ denotes the space of polynomials of degree less or equal to p (see Remark 1 for details on the polynomial interpolation), and $\mathbb{P}_\partial$ is the $\mathcal{L}^2$ projection on $\partial\Omega_D$.

In order to derive a weak problem for system (5), the first equation is multiplied by a vector test function $\boldsymbol{W} \in [\mathcal{V}_h]^d$ and the second by a scalar test function $\omega \in \mathcal{V}_h$, and the resulting equations are integrated in each element. After integrating by parts, the resulting weak problem for each element corresponding to (5) becomes: given $\hat{u}_h \in \Lambda_h^t(u_D)$, find an approximation $(\boldsymbol{q}_h, u_h) \in [\mathcal{V}_h]^d \times \mathcal{V}_h$ such that

$$\begin{aligned} &\Big(\omega, \frac{\partial u_h}{\partial t}\Big)_{\Omega_i} - \Big(\omega, \boldsymbol{a}\cdot\boldsymbol{q}_h\Big)_{\Omega_i} - \Big(\boldsymbol{\nabla}\omega, \boldsymbol{\mathcal{K}}\boldsymbol{q}_h\Big)_{\Omega_i} \\ &\qquad + \Big\langle \omega, \boldsymbol{\mathcal{K}}\boldsymbol{q}_h\cdot\boldsymbol{n}\Big\rangle_{\partial\Omega_i} + \Big\langle \omega, \tau(u_h - \hat{u}_h)\Big\rangle_{\partial\Omega_i} = \Big(\omega, f\Big)_{\Omega_i}, \\ &\Big(\boldsymbol{W}, \boldsymbol{q}_h\Big)_{\Omega_i} - \Big(\boldsymbol{\nabla}\cdot\boldsymbol{W}, u_h\Big)_{\Omega_i} + \Big\langle \boldsymbol{W}\cdot\boldsymbol{n}, \hat{u}_h\Big\rangle_{\partial\Omega_i} = 0. \end{aligned} \tag{6}$$

for all $(\boldsymbol{W}, \omega) \in [\mathcal{V}_h]^d \times \mathcal{V}_h$, for $i = 1, \ldots, \mathtt{n_{el}}$, where $\Big(\cdot,\cdot\Big)_{\Omega_i}$ denotes the $\mathcal{L}^2$ scalar product in the element Ω_i, $\Big\langle\cdot,\cdot\Big\rangle_B$ denotes the $\mathcal{L}^2$ scalar product of the traces over any $B \subset \Gamma$.

The traces of $\boldsymbol{q}_h$ on the element boundaries have been replaced by the numerical traces in the following way

$$\boldsymbol{\mathcal{K}}\hat{\boldsymbol{q}}_h\cdot\boldsymbol{n} = \boldsymbol{\mathcal{K}}\boldsymbol{q}_h\cdot\boldsymbol{n} + \tau(u_h - \hat{u}_h),$$

where τ is a stability parameter, see section 4 for a more detailed discussion.

Remark 1 (Polynomial interpolation). *Standard nodal basis are considered here, and Fekete node distributions are used to minimize ill-conditioning, see [33]. As an*

illustrative example, Fig. 2 shows the nodes corresponding to the spaces $\mathcal{V}_h$ and Λ_h for a triangle with polynomial degree $p = 4$. However, other approximations can be considered.

3.3. The global problem

The local problems (5), or (6), allow to compute the solution $\boldsymbol{q}_h$ and u_h in the whole domain in terms of the trace of the unknowns on the mesh skeleton, $\hat{u}_h$. Thus, this variable can now be understood as the actual unknown of the problem. This new unknown is determined using the global equation (4f). In fact, as already discussed, eq. (4e) is already fulfilled by the unicity of the trace unknown in each face (5c), and (4f) is the remaining global condition which must be imposed. This equation (in weak form) determine the HDG *global problem*. Namely, find an approximation $\hat{u} \in \Lambda_h^t(u_D)$ such that

$$\sum_{i=1}^{\mathtt{n_{el}}} \Big\langle \mu, \boldsymbol{\mathcal{K}} \boldsymbol{q}_h \cdot \boldsymbol{n} + \tau(u_h - \hat{u}_h) \Big\rangle_{\partial\Omega_i} = \Big\langle \mu, g_N \Big\rangle_{\partial\Omega_N}, \tag{7}$$

for all $\mu \in \Lambda_h(0)$. Here, $u_h \in \mathcal{V}_h$ and $\boldsymbol{q}_h \in [\mathcal{V}_h]^d$ are now solutions of the local problems (6) as a function of $\hat{u}_h$, hence the only unknown in this equation is $\hat{u}_h$.

The discretization of eq. (7) produce a global sparse linear system $\mathcal{M}$ involving the nodal values of the unknown $\hat{u}_h$ in the whole mesh skeleton.

4. Stabilization parameter

The choice of the stabilization parameter is crucial, and its influence has been discussed and analyzed for a large number of problems by Cockburn and co-workers see, for instance, Refs. [34, 28, 35, 36, 37, 38]. For extremely anisotropic problems in particular, an accurate choice of the stabilization parameter allows to obtain accurate solutions and to recover theoretical convergence rates. Based on the exhaustive analysis performed in [35], the stabilization parameter is splitted into a convective and a diffusive part as

$$\tau = \tau_c + \tau_d.$$

The convective part is similar to the one used in [35], that is, $\tau_c = |\boldsymbol{a} \cdot \boldsymbol{n}|$. For the diffusive part τ_d, the following three definitions have been considered:

- $\tau_d^1 = k_{\parallel}$,
- $\tau_d^2 = p/h \; k_{\parallel}$,

- $\tau_d^3 = p/h\ \boldsymbol{n}\cdot(\boldsymbol{\mathcal{K}}\boldsymbol{n})$,

where h is the element size, that is, the lenght of the edge where the parameter is defined. The definitions $\tau_d^{1,2}$ have been proposed in the literature [35]. While τ_d^1 provides an unique value in the whole domain while, τ_d^2 is weighted with the nodal distance in each element. In this work we propose a new definition of the stabilization parameter, τ_d^3, in order to more efficiently take into account the anisotropy of the problem. It is based on the modulation of the stabilization with respect to the actual strength of the parallel diffusion in the normal direction of each face.

These three stabilization parameters for the diffusion part are evaluated with respect to their impact on the accuracy and convergence properties of the method on a purely diffusive 2D test ($\boldsymbol{a}$=$\boldsymbol{0}$). The method of the manufactured solution (MMS) is used to evaluate the $\mathcal{L}^2$-error between the finite-element solution u and the analytical solution u_{an}, for which the source term in Eq.5a has been modified accordingly.

The computational domain is an annulus with exterior radius $r = 2$ and interior radius $r = 1$, and the magnetic field direction is circular with expression

$$b_x = \frac{y}{x^2+y^2}, \quad b_y = -\frac{x}{x^2+y^2}.$$

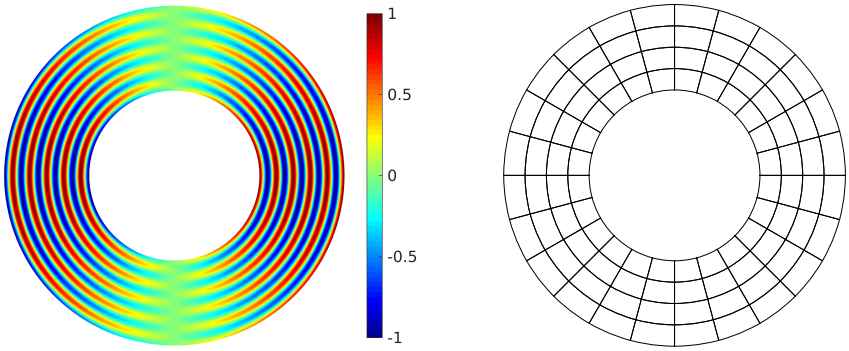
In order to mimic a typical physical situation in a magnetized plasma application, where elongated structures along the magnetic lines are found, the analytical solution u_{an} is chosen aligned with the magnetic field such that:

$$u_{\text{an}} = \cos(2\pi\omega r)\cos(\theta),$$

where r is the distance from the point $(0, 0)$, θ is the angle formed by the position vector and the x-axis, and ω is the frequency of oscillation in the perpendicular direction, see Fig. 3a.

The discretizations considered are based on aligned curved quadrilaterals with different element sizes h and polynomial degrees p, see Fig. 3b. The following combinations of h and p are used to evaluate the mean slope of convergence of the $\mathcal{L}^2$-error: $h = 1/2^4 \cdots 1/2^7$ for $p = 2$, $h = 1/2^3 \cdots 1/2^6$ for $p = 4$, $h = 1/2^2 \cdots 1/2^5$ for $p = 8$.

Results are summarized on Fig. 4. For each value of p, the 1D plots show the value of the $\mathcal{L}^2$-error obtained for the finest grid as well as the mean slope when reducing h. All the three stabilization parameters provide stable computations for $k_\parallel$ varying between 1 and 10^9 ($k_\perp$ is kept constant equal to 1). However, with τ_d^2 and τ_d^3, the value of the $\mathcal{L}^2$-error remains constant whatever the value of $k_\parallel$, except for



(a) (b)

Figure 3: Stabilization parameter study: 2D map of the analytical solution (a) and computational mesh (b).

$p = 8$ where it finishes by increasing at large $k_\parallel$ values ($k_\parallel > 10^6$). This behavior has to be related to the constant increase with $k_\parallel$ of the condition number of the matrix $\mathcal{M}$ (Fig. 5) when solving the linear system corresponding to Eq. 7 with a 2D direct solver. As long as the anisotropy remains moderate ($k_\parallel \leq 10^6$), the $\mathcal{L}^2$-error remains dominated by the polynomial interpolation error, and as expected the accuracy for $p = 8$ is better as for $p = 4$. However, at strong anisotropy ($k_\parallel > 10^6$), the error in the resolution of the linear system dominates over the interpolation error, and thus leads to an increase of the $\mathcal{L}^2$-error up to the value obtained for $p = 4$, as shown also in the convergence analysis in Sec. 6.1. It is important to notice that this is due to the strong anisotropy and not to the polynomial interpolation. For any value of $k_\parallel$, even if the results behaviour is very similar when comparing the $\mathcal{L}^2$-errors, a slightly better accuracy is obtained with the parameter τ_d^3 than with the parameter τ_d^2. It is thus chosen for all the following tests of this work.

5. Alignment study for triangular meshes

Triangular mesh alignment is an active field, in particular in hydrodynamic modeling and magnetized plasma simulations. Using the modified equation analysis, in [22] the relationship between mesh alignment and diffusive errors is investigated in the context of finite-volumes ($p = 0$) schemes and triangular discretizations. The numerical diffusion is estimated in relation with the non-alignment angle, and a best and worst case scenarios of alignment are defined, where, respectively, the numerical diffusion is minimized (*aligned mesh*) or maximized (*non-aligned mesh*). Here, these two extreme cases, see Fig. 6, are tested with different polynomial interpolations, in

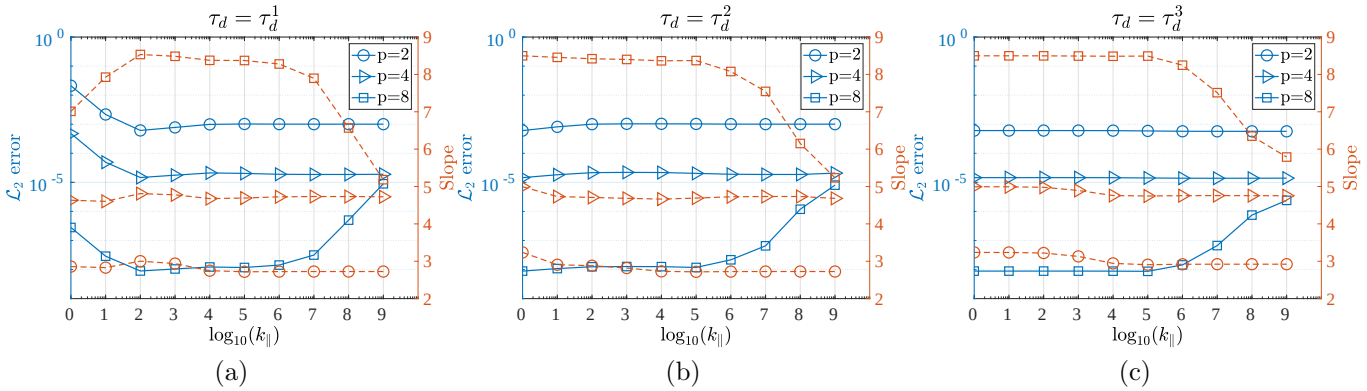


Figure 4: Stabilization parameter study: 1D plots showing the evolutions of the $\mathcal{L}^2$-error (full lines) and of its mean slope (dashed lines) as a function of the anisotropy for $p = 2$, $p = 4$ and $p = 8$ and for the three stabilization parameters considered. The values of the $\mathcal{L}^2$-error correspond to the values obtained on the finest grid while the mean slope is the averaged value of the local slopes measured for each combination of h and p.

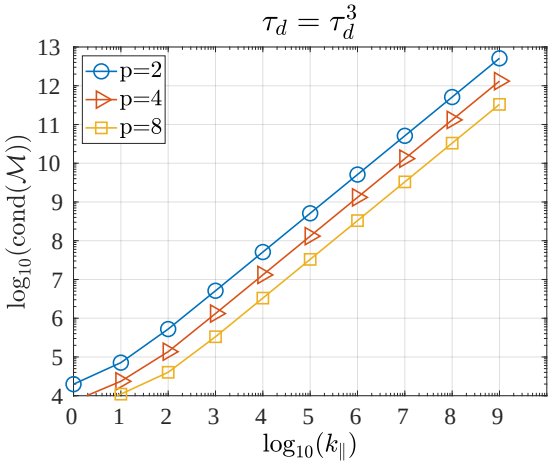
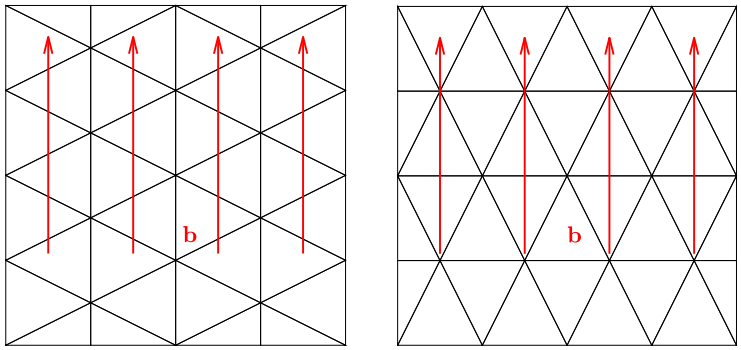


Figure 5: Condition number of the global matrix $\mathcal{M}$ as a function of $k_\|$ for different polynomial degrees and $\tau_d = \tau_d^3$. Similar results are obtained for τ_d^1 and τ_d^2.



(a) (b)

Figure 6: Examples of triangular aligned (a) and non-aligned (b) discretizations in a square domain together with the magnetic field **b**.

order to verify, on the one hand, the validity of the definition of alignment in a finite-element framework (hence $p > 0$), and on the other hand, verify the performance of high-order interpolation on the worst case of non-alignment.

The test-case setup tries to mimic a 1D solution on a 2D computational domain. This allows us, firstly, to have an analytical solution to use as comparison. Moreover, it decouples the effect of the numerical diffusion from the solution of the problem, since the analytical solution does not depend on the parallel diffusion, as explained next.

The computational domain is $\Omega = [0, 1] \times [0, 1]$, with a Dirichlet boundary condition at the vertical boundary $x = 1$, where the solution $u|_{x=1} = 1$ is imposed. The rest of the boundaries are of type Neumann homogeneous. The magnetic field is vertical with components $b_x = 0, b_y = 1$, the perpendicular diffusion is set as $k_\perp = 1$, and a source is imposed with the form

$$s(x) = \sin(\pi x).$$

The 1D analytical solution in the x-direction is found computing the solution to the one dimensional ODE

$$k_\perp \frac{d^2u}{dx^2} + s(x) = 0.$$

The 2D setup of the problem involves imposing a parallel diffusion $k_\parallel$ with different intensities. Since the value of the parallel diffusion does not affect the analytical solution, the effect on the numerical solution is entirely due to the numerical diffusion introduced by the scheme on a given discretization.

Aligned and non-aligned triangular meshes defined in the square are considered, Fig. 6. Various combinations of h and p are chosen, providing roughly the same number of degree of freedom ($\mathrm{N_{dof}}$) for the different cases, see Table 1. Let's note that due to the hybridization technique, the size of the linear system decreases for higher polynomials. In Table 1 it is also shown the condition number of the linear system matrix $\mathcal{M}$ for all the computations and $k_{\|} = 10^4$: the results show that increasing the order of the polynomial does not lead to higher condition numbers.

Results for linear elements are shown in Fig. 7 for $k_{\|} = 1, 10^4$, and 10^9 respectively. The numerical solution on aligned and non-aligned meshes is compared with the analytical solution. The element size is also represented in the figures. The results show that, while for the isotropic ($k_{\|} = 1$) computation the aligned and the non-aligned discretizations provide the same solution, which perfectly matches the analytical one, increasing the anisotropy of the problem degrades the results on the non-aligned mesh, leaving unchanged the one on the aligned mesh. This confirms the role of the numerical diffusion already described in [22] for finite-volume (hence $p = 0$ elements), and extends the result of this paper to linear interpolations. It is interesting to note that h-refining the mesh does not improve the results, as shown in Table 2, where it is reported the $\mathcal{L}^2$-error between the numerical solution and the analytical solution for decreasing element size and $k_{\|} = 10^9$. On the contrary, the increase in the condition number refining the mesh introduces numerical errors, also in the aligned case, that degrades the convergence.

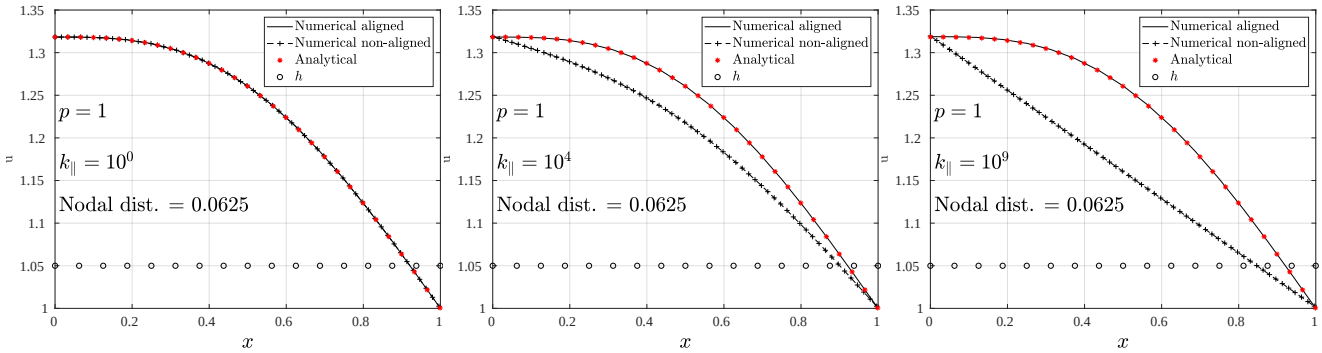
In Fig. 8 are reported the results for the case $k_{\|} = 10^9$ for $p > 1$, where it is clear that increasing the polynomial interpolation is sufficient in this case to eliminate the numerical diffusion introduced by the non-alignment. In fact, even if for $p = 2$ a small deviation is still visible in the non-aligned computation, profiles nearly match for $p = 4$ and $p = 8$. With the non-aligned meshes, the $\mathcal{L}^2$-error between the analytical solution and the numerical ones decrease when increasing p, with values equal to 2.769×10^{-3}, 7.4952×10^{-5}, 5.5058×10^{-5}, for $p = 2, 4$ and 8, respectively. This is also confirmed by the $\mathcal{L}^2$ -error between numerical and analytical solution reported in Table 1.

6. Numerical experiments in the poloidal plane

The ability of the method to handle problem (2) is shown here by considering a 2D annular domain mimicking the poloidal section of a tokamak. The same geometry and magnetic field considered in Sec. 4 are used in the following two tests. The focus is made on the resolution of the anisotropic diffusive operator by assuming a steady and purely diffusive problem ($\boldsymbol{a} = \mathbf{0}$) with $k_{\perp} = 1$ and $k_{\|}$ varying between 1 and 10^9.

Table 1: Alignment study: condition numbers of the global matrix $\mathcal{M}$ and $\mathcal{L}^2$ errors for different polynomials and element sizes and $k_{\parallel} = 10^4$, and for aligned and non-aligned configurations. The discretizations are chosen in order to have roughly the same number of nodes in the whole mesh.

p	h	$\mathrm{N}_{\mathrm{nodes}}$	size($\mathcal{M}$)	cond($\mathcal{M}$) al.	cond($\mathcal{M}$) non-al.	$\mathcal{L}^2$ err al.	$\mathcal{L}^2$ err non-al.
1	$1/2^4$	297	1616	8.6e7	4.0e7	1.2e-4	3.0e-2
2	$1/2^3$	305	636	4.0e7	2.0e7	1.5e-5	3.7e-4
4	$1/2^2$	321	290	1.5e7	8.0e6	2.3e-7	3.6e-7
8	$1/2^1$	353	153	8.3e6	4.3e6	2.3e-9	6.4e-10



(a) (b) (c)

Figure 7: Alignment study for linear elements: 1D plots of the numerical solutions for the aligned and non-aligned configurations, together with the analytical solution. The isotropic case ($k_{\parallel} = 1$) is shown in (a), while (b) and (c) refer to anisotropic cases with respectively $k_{\parallel} = 10^4$ and $k_{\parallel} = 10^9$. The element size is also shown.

Table 2: Alignment study: condition numbers and $\mathcal{L}^2$ errors for linear elements in aligned and non-aligned configurations and $k_{\parallel} = 10^9$, with different element sizes.

h	$\mathrm{N}_{\mathrm{dof}}$	size($\mathcal{M}$)	cond($\mathcal{M}$) al.	cond($\mathcal{M}$) non-al.	$\mathcal{L}^2$ err al.	$\mathcal{L}^2$ err non-al.
$1/2^5$	1105	6368	1.5e13	3.3e13	7.6e-4	7.1e-2
$1/2^6$	4257	25024	1.3e14	6.0e13	3.3e-3	7.2e-2
$1/2^7$	16705	99200	5.2e14	2.4e14	1.7e-2	7.7e-2
$1/2^8$	66177	395008	2.2e15	9.2e14	6.2e-2	9.9e-2

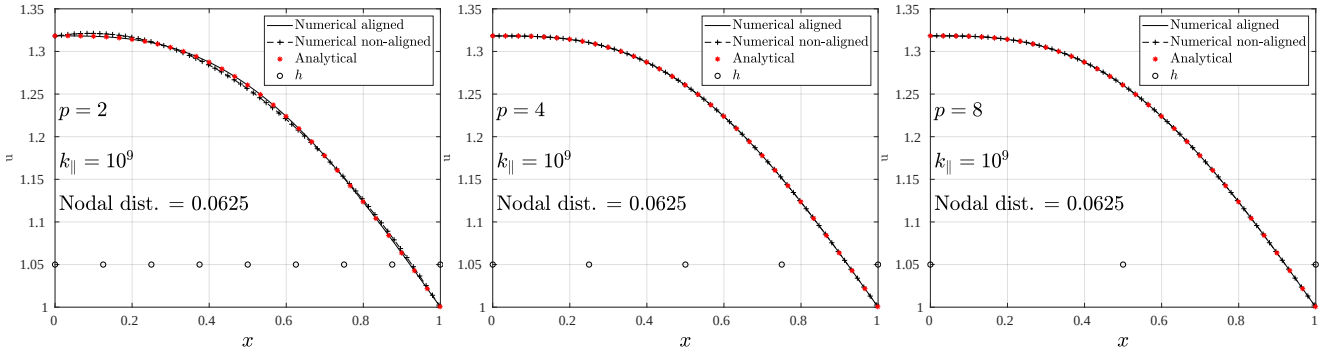


Figure 8: Alignment study: 1D plots of the numerical solutions for the aligned and non-aligned configurations together with the analytical solution, for $p = 2$ in (a), $p = 4$ in (b) and $p = 8$ in (c). The element size is also shown.

6.1. Convergence analysis

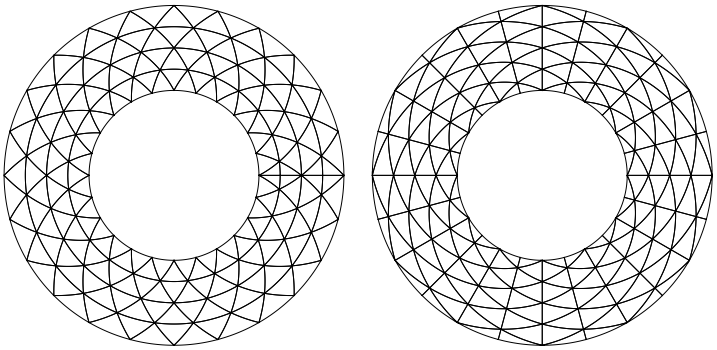
In this test, the method of manufactured solutions is used to compute a set of convergence curves. The main goal of this test is to evaluate the ability of the method to achieve the theoretical convergence rates in presence of strong anisotropic diffusion. In particular, the performance of aligned and non-aligned triangular discretizations is evaluated and compared with a reference computation on quadrilateral aligned meshes 3b.

The discretizations considered here are shown on Fig. 9 for aligned (a) and non-aligned triangular meshes (b). These meshes are generated mapping respectively aligned and non-aligned rectangular meshes similar to those of Fig. 6 onto an annulus. For the quadrilateral aligned discretizations, the meshes are similar to the one shown in Fig. 3b.

Results are shown on Fig. 10 and Fig. 11 for an isotropic ($k_\parallel = k_\perp = 1$) and an anisotropic case ($k_\perp = 1$ and $k_\parallel = 10^9$), respectively.

For the isotropic case (Fig. 10), the $\mathcal{L}^2$-norm convergences are nearly independent of the nature of the mesh, even if for the aligned meshes quadrangles provide slightly better results than triangles. The theoretical convergence rate $(p+1)$ is achieved for all the polynomial approximations.

For the highly anisotropic case however, linear elements ($p = 1$) still converge at the expected rate for the aligned meshes (Fig. 11a, b), but are penalized by the misalignement of the mesh (Fig. 11c). On the contrary, when elements are of higher-order ($p = 2, 4$ and 8) the impact of the misalignement of the mesh on the convergence results seems to be limited. Local values of the slope can be even better



(a) (b)

Figure 9: Examples of computational triangular meshes used for the convergence study: (a) aligned mesh, (b) non-aligned mesh.

when using non-aligned triangles than aligned ones, without any rigorous explanation having been found. For the non-aligned mesh the theoretical rate of convergence is recovered (Fig. 11c), except for $p = 8$ where the error saturates at about 10^{-5} due to the error in resolving the linear system, preventing the convergence slope to reach its theoretical value when continuing to reduce h.

6.2. Diffusion of a Gaussian source

In this test, a Gaussian source is diffused towards an identical sink. The source is defined as

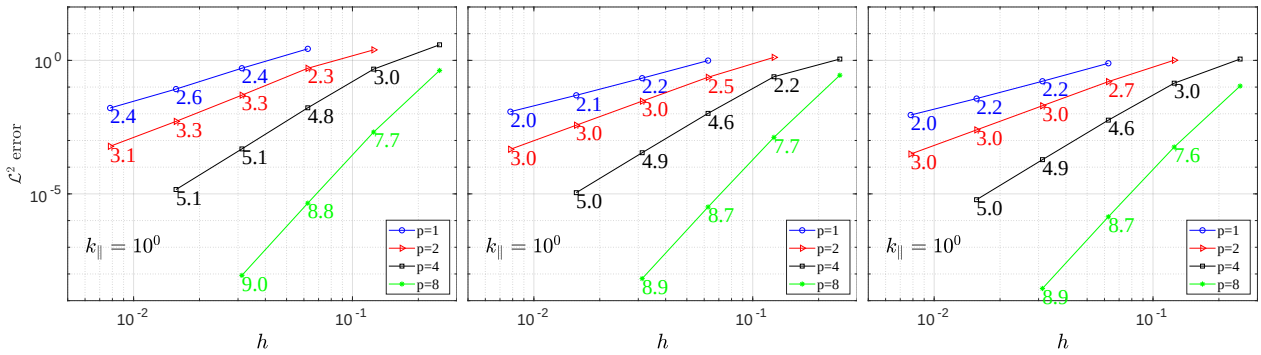
$$f_{sc} = k_{\parallel} \exp(-r_{sc}^2/r_0^2),$$

where $r_{sc} = \sqrt{(x - x_{sc})^2 + (y - y_{sc})^2}$ is the distance from the point $\boldsymbol{x}_{sc} = \{1.5, 0\}$ and $r_0 = 0.05$ defines the area radius where the source is applied. A sink with equal intensity is imposed at location $\boldsymbol{x}_{sk} = \{-1.5, 0\}$,

$$f_{sk} = -k_{\parallel} \exp(-r_{sk}^2/r_0^2),$$

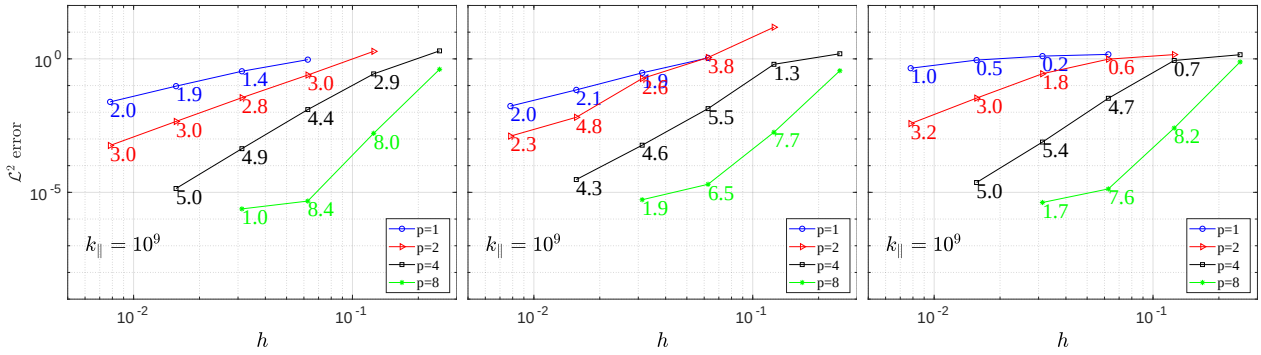
with $r_{sk} = \sqrt{(x - x_{sk})^2 + (y - y_{sk})^2}$. The source is depicted in Fig. 12a, while in Fig. 12b is shown an high-order finite element solution of this problem for a parallel and perpendicular diffusion $k_{\parallel} = 10^9$ and $k_{\perp} = 1$.

The results of the test are analyzed taking profiles of the solution along the four radial lines depicted in Fig. 12b. Since $k_{\parallel} \gg k_{\perp}$, the solution is expected to be diffused only in the parallel direction. This means that the normalized profiles (with respect to the maximum on each radial line) should overlap. Moreover, the normalized profiles should almost overlap to the source profile. The difference between the



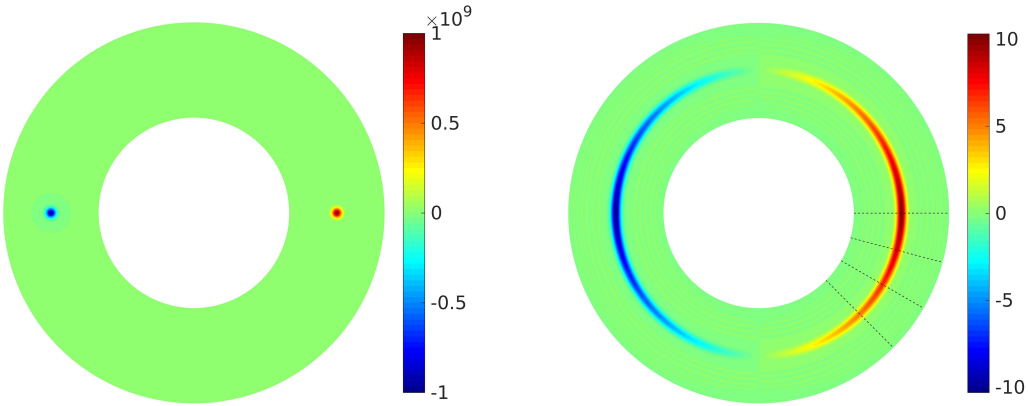
(a) (b) (c)

Figure 10: $\mathcal{L}^2$-norm convergence for $k_\parallel = k_\perp = 1$: (a) quadrilateral aligned mesh, (b) triangular aligned mesh, (c) triangular non-aligned mesh.



(a) (b) (c)

Figure 11: $\mathcal{L}^2$-norm convergence for $k_\parallel = 10^9$ and $k_\perp = 1$: (a) quadrilateral aligned mesh, (b) triangular aligned mesh, (c) triangular non-aligned mesh.



(a) (b)

Figure 12: Diffusion of a Gaussian source: 2D maps of the source (a) and of the solution (b). The dashed lines in the solution plot indicate the lines where the profiles are extracted.

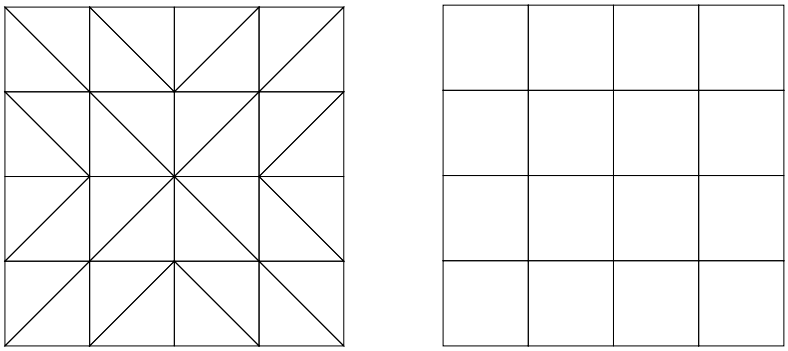
normalized profiles and the source profile can be regarded as the numerical diffusion present in the computation.

The computational meshes considered are taken as in Fig. 9 for the aligned configurations (quadrangles and triangles). For the non-aligned configuration, a square domain is considered with triangular and quadrangular structured meshes, see Fig. 13. This is done to simplify the definition of non-aligned meshes when using quadrilateral elements. All the meshes with the same geometry (circle or square) have the same number of nodes.

Results on aligned meshes are shown in Fig. 14 and Fig. 15. The profiles show no lateral spreading of the solution, confirming that the aligned discretization provide non-diffusive solution at any p. Triangular and quadrangular mesh perform similarly in the aligned configuration. For the non-aligned cases, results are shown in Fig. 16 for triangles and in Fig. 17 for quadrangles. An important spreading of the solution is now visible for low order elements ($p = 1$ and $p = 2$) but for high-order elements a non-diffusive solution is recovered. Quadrangular meshes are also less diffusive with respect to triangular ones.

7. A numerical test in the toroidal direction

We propose in this section a study of the numerical diffusion introduced by the non-alignment of the mesh in the toroidal direction. A simplified 2D rectangular geometry is considered, with dimensions $[0, l] \times [0, 1]$, where the the x and y directions represent respectively the toroidal and radial tokamak dimensions. Periodic bound-



(a) (b)

Figure 13: Diffusion of a Gaussian source: triangular (a) and quadrangular (b) meshes used for the non-aligned computations.

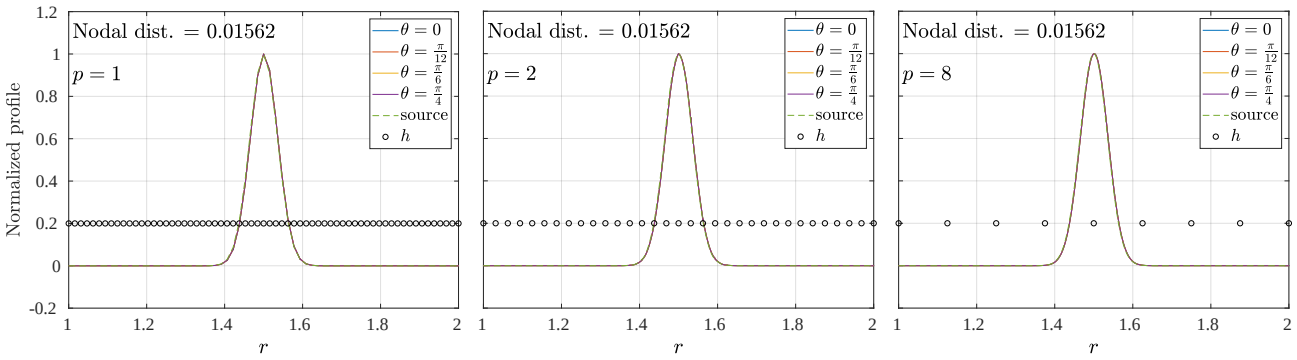


Figure 14: Diffusion of a Gaussian source: normalized profiles for aligned quadrangular meshes as in Fig.3b. In (a) $p = 1$ and $h = 1/2^6$, in (b) $p = 2$ and $h = 1/2^5$, in (c) $p = 8$ and $h = 1/2^3$. All the meshes have the same number of points and the same mean nodal distance.

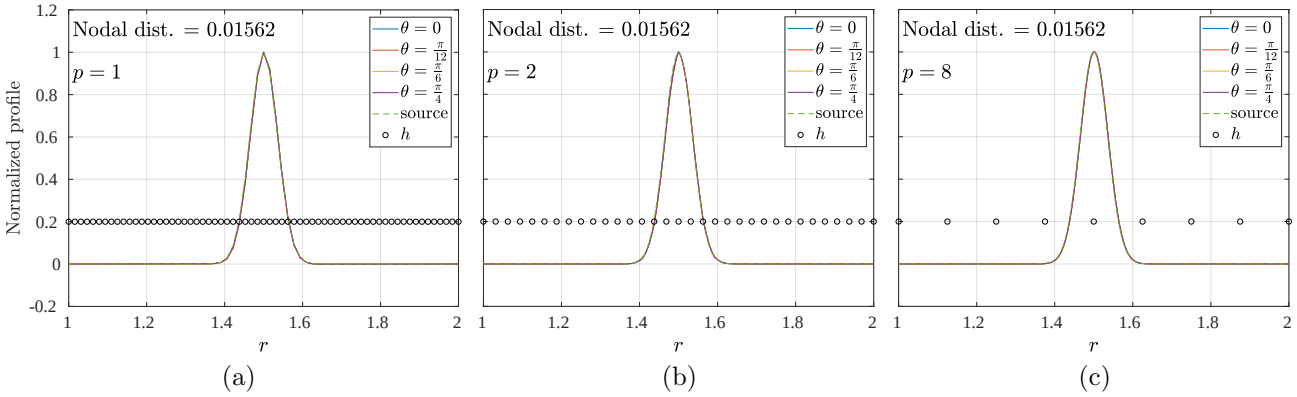


Figure 15: Diffusion of a Gaussian source: normalized profiles for aligned triangular meshes as in Fig. 9a. In (a) $p = 1$ and $h = 1/2^6$, in (b) $p = 2$ and $h = 1/2^5$, in (c) $p = 8$ and $h = 1/2^3$. All the meshes have the same number of points and the same mean nodal distance.

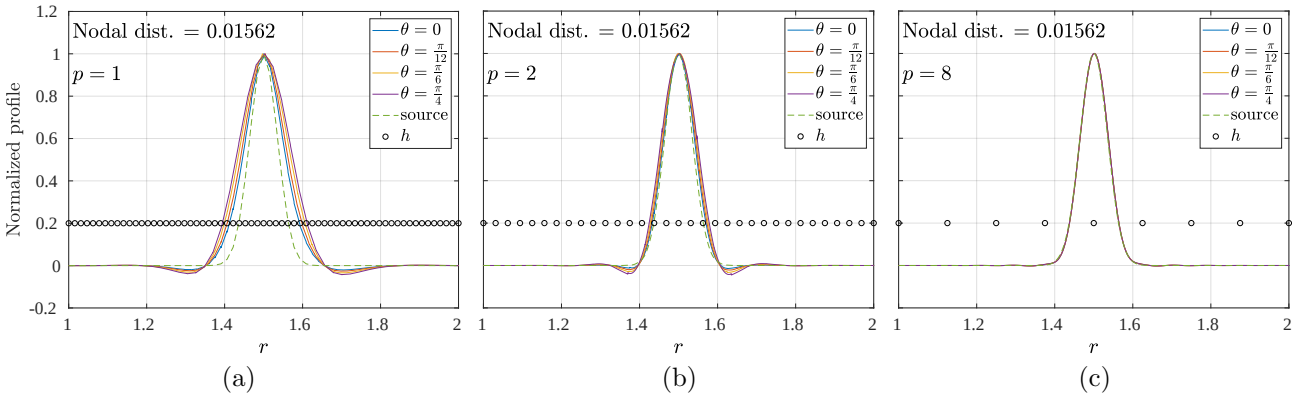


Figure 16: Diffusion of a Gaussian source: normalized profiles computed on a square domain with structured triangles, hence a non-aligned triangular discretization. In (a) $p = 1$ and $h = 1/2^6$, in (b) $p = 2$ and $h = 1/2^5$, in (c) $p = 8$ and $h = 1/2^3$. All the meshes have the same number of points and same mean nodal distance.

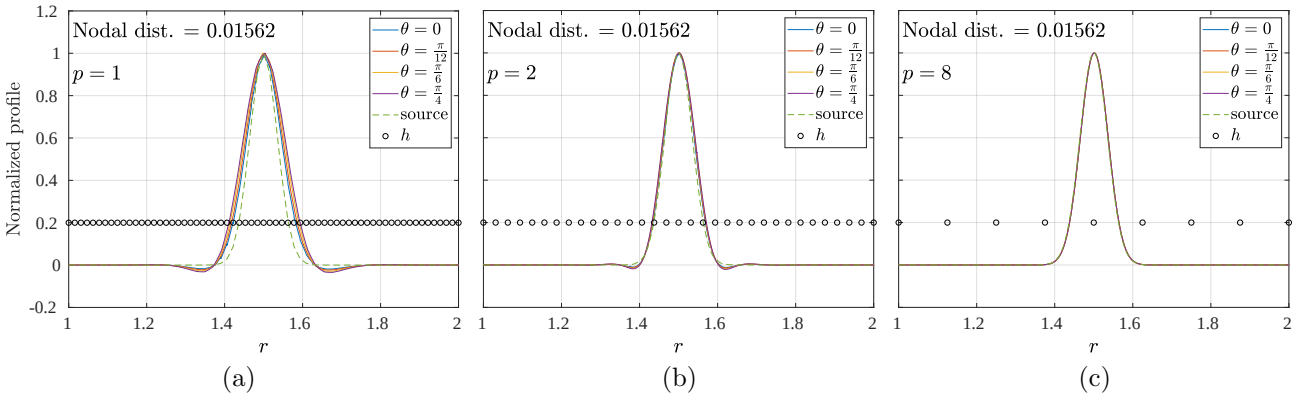


Figure 17: Diffusion of a Gaussian source: normalized profiles computed on a square domain with structured quadrangles, hence a non-aligned quadrangular discretization. In (a) $p = 1$ and $h = 1/2^6$, in (b) $p = 2$ and $h = 1/2^5$, in (c) $p = 8$ and $h = 1/2^3$. All the meshes have the same number of points and same mean nodal distance.

ary conditions are set at $x = 0$ and $x = l$. They are imposed by forcing the equality of the trace solution on the corresponding periodic faces. For $y = 0$ and $y = 1$, Dirichlet boundary conditions with $u = 0$ are set. A very fine discretization is used in the radial direction (10 elements of $p = 8$), which guarantees that no numerical diffusion is introduced. In the toroidal one the polynomials $p = 1, 2, 4, 8$ are tested. The number of elements in the toroidal direction is chosen in order to have a number of nodes equal to $16, 32, 64$ for each p.

The time-evolving system (2) is solved. A periodic solution in the toroidal direction is chosen as initial state with the shape of a sinusoidal blob, see fig. 18. This initial solution is aligned with the magnetic field, which is set with components

$$B_x = 1, \quad B_y = \frac{2d\pi}{l} \cos(2\pi \frac{x}{l}), \tag{8}$$

where $d = 0.1$ determines the amplitude of the sinusoidal oscillation. The angle formed by the magnetic field and the x-axis is referred as *pitch angle*, and it has a strong influence on the numerical diffusion introduced by the toroidal discretization. In this test, the maximum pitch angle α is related to the toroidal dimension l as

$$\alpha = \arctan(2\pi d/l).$$

The numerical diffusion introduced by the toroidal discretization is evaluated as follows. The initial solution u_0 is left to evolve under the diffusive process driven by the perpendicular dynamic, with no convection ($\boldsymbol{a}$=$\mathbf{0}$). In order to minimize the

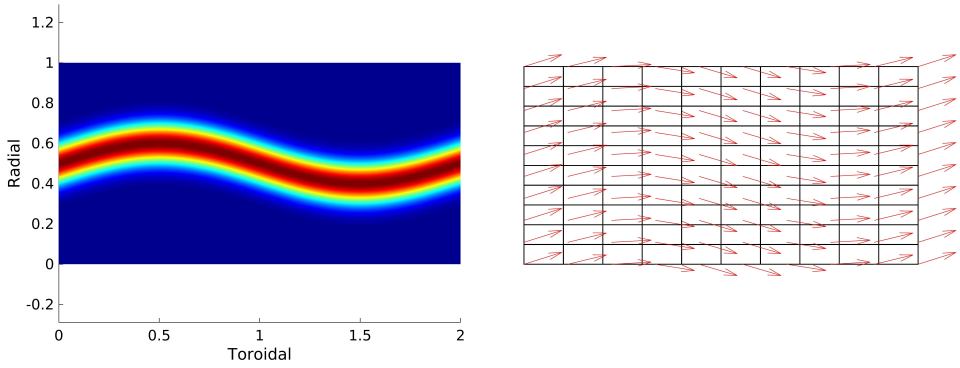


Figure 18: Toroidal study: 2D map of the aligned initial solution u_0 (left) and of the magnetic vector field (right).

time integration errors, while not penalizing the computational cost, a third-order backward differentiation formula is used to discretize the time derivative. The time step is chosen small enough to be sure that no time integration errors are introduced in the results. The time evolution is stopped when the maximum value of the solution attains the value

$$\max u = 0.5 \max u_0.$$

This allows to evaluate the reference time T_r needed to diffuse the initial solution to this extent, which is a measure of the perpendicular diffusivity. A set of reference times are obtained using computations with isotropic diffusion ($k_\parallel = k_\perp$), providing a law relating T_r^{is} and $k_\perp$ of the type

$$T_r^{\text{is}} = T_r^0 / k_\perp,$$

being T_r^0 the reference time for $k_\perp = 1$.

After this reference curve is obtained, a set of computations is performed for anisotropic diffusion, with $k_\perp = 1$ and $k_\parallel = 1, \cdots, 10^9$, and for different values of the pitch angle. Since the solution is aligned on the magnetic field, the difference between the computed T_r^{an} in the anisotropic case and T_r^0 is due entirely to the numerical diffusion. An effective $k_\perp^*$ is then computed as

$$k_\perp^* = T_r^0 / T_r^{\text{an}}.$$

Results are shown in Fig. 19 for $p = 1$, Fig. 20 for $p = 2$, Fig. 21 for $p = 4$ and Fig. 22 for $p = 8$.

As expected, the effective perpendicular diffusion $k_\perp^*$ increases as the parallel physical diffusion is increased, and decreases with the number of points in the toroidal

direction. For low-order discretizations, the effective perpendicular diffusion can reach values several order of magnitudes larger than the physical perpendicular diffusion, also for a large number of points, see for example Fig 19c. Also, increasing the pitch angle increases exponentially the value of $k_\perp^*$.

On the other hand, increasing the polynomial interpolation reduces drastically the effective diffusion. This is also underlined by Fig. 23, where the effective perpendicular diffusion is depicted as a function of the polynomial interpolation in the toroidal direction, for a pitch angle of $4°$, $k_\parallel = 10^6$ and $k_\parallel = 10^9$, 16 and 32 points in the toroidal direction.

As a comparison, the same test has been repeated using two finite-difference schemes based on non-aligned discretizations: a classic scheme with a naive definition of the derivatives and the Günter's scheme specifically designed for magnetized plasma simulations [14]. Thanks to its discretization of the parallel operator, this later has shown in former studies to reduce the spurious perpendicular diffusion observed in standard finite-difference simulations when the anisotropy becomes large (see a recent investigation in [13]). The poloidal y-direction has been discretized using a fine mesh (256 points) to guarantee an error free solution in this direction. The toroidal x-direction has been discretized with 64 nodes. Results on Fig. 24 confirm as expected that the Günter's scheme reduces the numerical diffusion with respect to the standard discretization but its values remain high, in fact close to the ones measured with the HDG scheme when linear elements $p = 1$ are used. This is actually not satisfying to guarantee the accuracy of the solution in such configuration.

A strength of finite-difference schemes remains however their ease of implementation and their reduced numerical cost with respect to many other numerical discretizations for a same number of degree of freedom. This certainly explains why they remain popular for many simulations. To push forward comparison with the present scheme, we have evaluated the computational cost of both methods for a same accuracy and with a strong anisotropy, $k_\parallel = 10^5$ and $k_\perp = 0$. Also for this test, the poloidal y-direction has been discretized using refined meshes composed of 256 points for the Günter's scheme and 10 elements of $p = 8$ for the HDG schemes, respectively, in order to guarantee an error free solution in this direction. HDG computations have been performed with a fixed number of toroidal points equal to $p \times nt = 24$ and for p varying from 1 to 8, meaning that the number of elements in the toroidal direction decreases when the interpolation degree p increases. For the Günter's scheme the number of points have been tuned for each simulation to nearly match the spurious perpendicular diffusion measured from the HDG simulation. As an example, the HDG computation with 3 toroidal elements of degree $p = 8$ leads to a effective perpendicular diffusion of $k_\perp^* = 0.125 \times 10^{-4}$ that requires 1411 points in

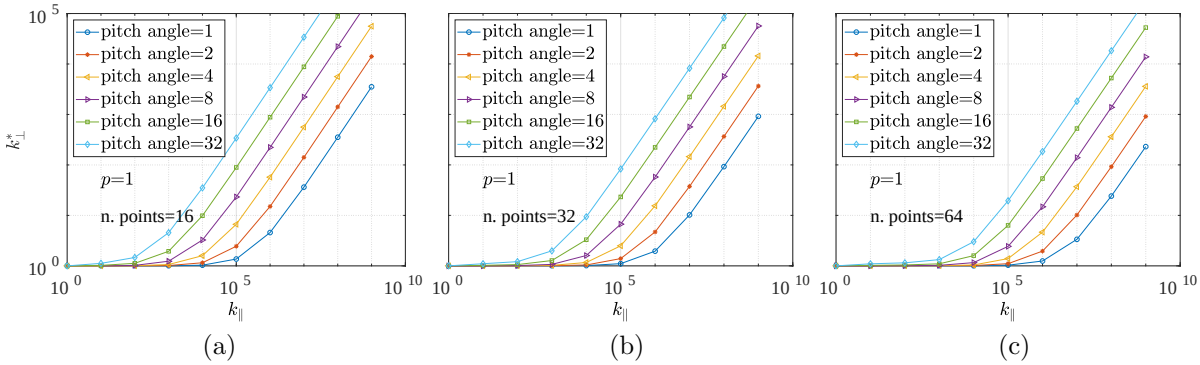


Figure 19: Toroidal test for linear elements $p = 1$: 1D plots of the effective perpendicular diffusion $k_\perp^*$ as a function of the parallel diffusion $k_\parallel$ and for different values of pitch angle and number of nodes in the toroidal direction $nt = 16$ (a), 32 (b) and 64 (c). $k_\perp = 1$.

the toroidal direction for an equivalent FD computation. In this case, the Günter's scheme computation is thus 840 slower than the HDG computation. All the results are plotted on Fig. 25 and show that the HDG scheme becomes faster than the finite-difference scheme as soons as the degree $p \geq 2$.

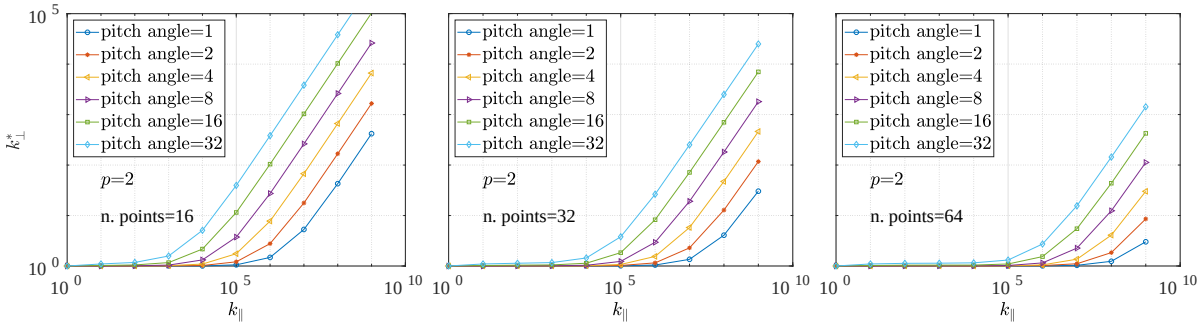


Figure 20: Toroidal test for $p = 2$: effective perpendicular diffusion $k_\perp^*$ as a function of the parallel diffusion $k_\parallel$ and for different values of pitch angle and number of nodes in the toroidal direction $nt = 16$ (a), 32 (b) and 64 (c). $k_\perp = 1$.

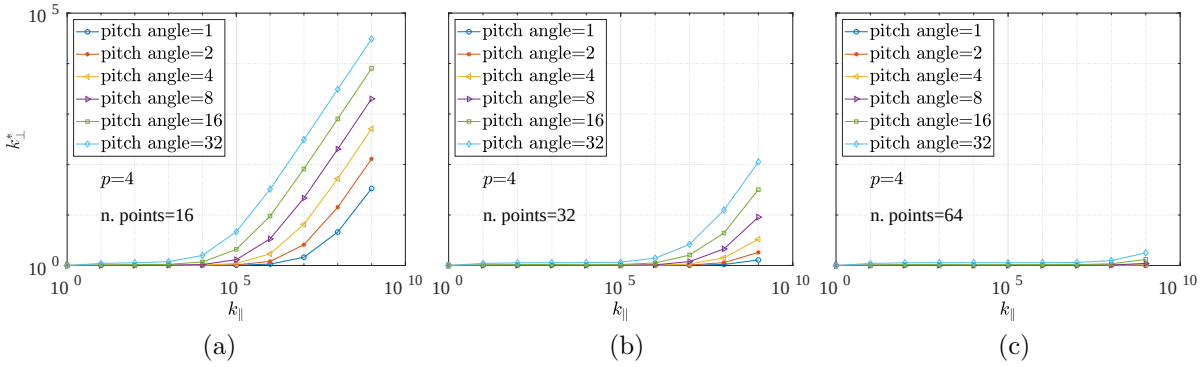


Figure 21: Toroidal test for $p = 4$: effective perpendicular diffusion $k_\perp^*$ as a function of the parallel diffusion $k_\parallel$ for different values of pitch angle and number of nodes in the toroidal direction $nt = 16$ (a), 32 (b) and 64 (c).$k_\perp = 1$.

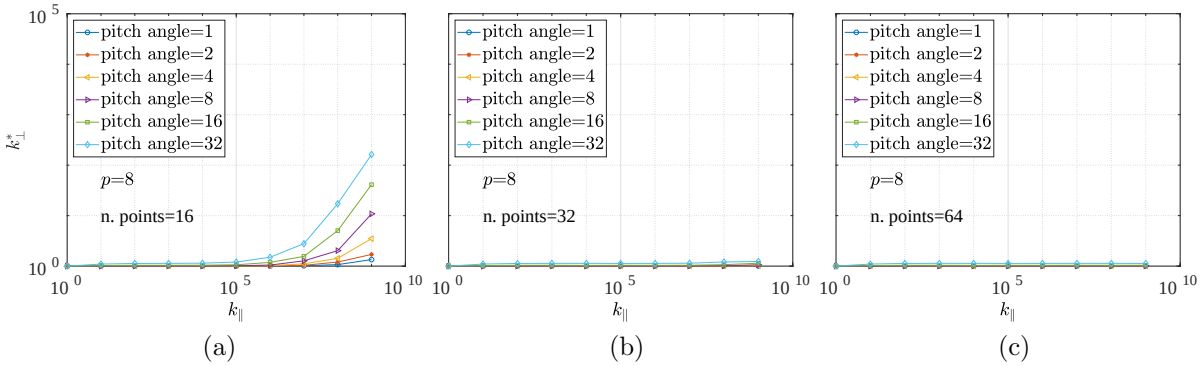


Figure 22: Toroidal test for $p = 8$: effective perpendicular diffusion $k_\perp^*$ as a function of the parallel diffusion $k_\parallel$ for different values of pitch angle and number of nodes in the toroidal direction $nt = 16$ (a), 32 (b) and 64 (c). $k_\perp = 1$.

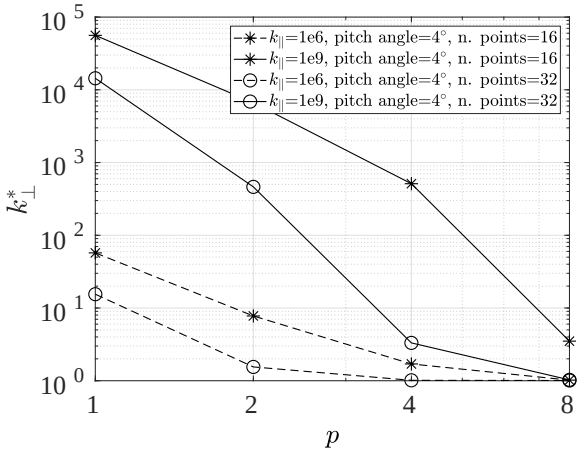


Figure 23: Toroidal test: 1D plots of the effective perpendicular diffusion $k_\perp^*$ as a function of the polynomial degree p in the toroidal discretization, for a pitch angle of 4°, $k_\parallel = 10^6, 10^9$ and $nt = 16, 32$ points in the toroidal direction. $k_\perp = 1$.

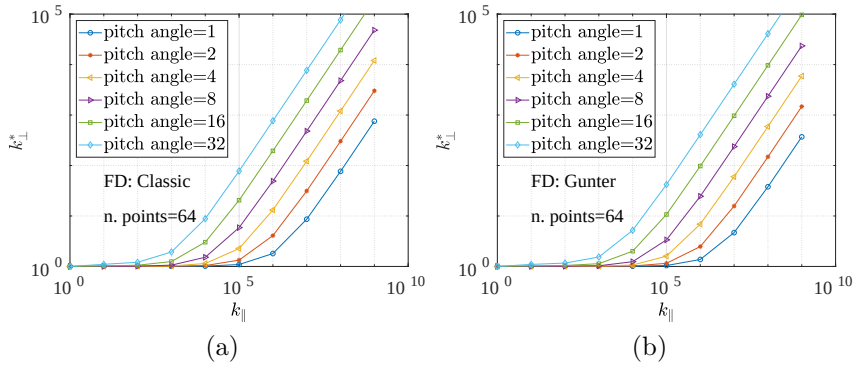


Figure 24: Toroidal tests for the standard finite difference scheme (a) and the Günter's scheme (b): 1D plots of the effective perpendicular diffusion $k_\perp^*$ as a function of the parallel diffusion $k_\parallel$ and for different values of pitch angle and $nt = 64$ nodes in the toroidal direction. $k_\perp = 1$.

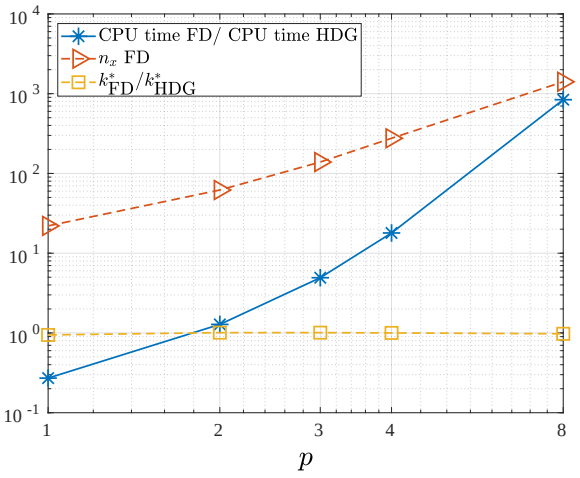


Figure 25: Toroidal test. Measure of the computational cost. Evolution of the ratii of the CPU times between the HDG and the Günter's scheme. HDG computations are carried out with $p * nt = 24$. The number of points in the toroidal direction for the Günter's scheme (n_x FD) is tuned to lead to approximately the same value of the spurious numerical diffusion than the HDG ($k^\star_{FD}/k^\star_{HDG} \simeq 1$).

8. A 3D numerical test in realistic tokamak geometry

The ability of the method to handle highly anisotropic advection-diffusion problem in a realistic geometry is shown by studying the convective transport of an aligned plasma filament in the geometry of the WEST tokamak, see [39]. In order to reduce the computing requirements, only the exterior part of the tokamak is considered, taking out a circular area corresponding grossly to the plasma core, with center $\{x_0 = 0, y_0 = 0\}$ and radius $r_c = 0.4$: in Fig. 26 is shown the triangular discretization used, which consists in 1420 elements of order $p = 4$. With the goal of simplifying the definition of the initial aligned solution and the magnetic field, a slab geometry is defined, neglecting the curvature terms. Therefore, the 3D discretization is obtained extruding the poloidal mesh in the toroidal direction z, defining a Cartesian geometry.

A simplified version of the real magnetic field used in the WEST machine is considered in this test, with an helicoidal shape such that:

$$\begin{aligned} B_x &= \frac{2\pi(y - y_0)}{z_{max}}, \\ B_y &= -\frac{2\pi(x - x_0)}{z_{max}}, \\ B_z &= 1. \end{aligned}$$

The convective transport of an aligned plasma filament is simulated. A circular convective velocity $\boldsymbol{a}$ is considered with components

$$\begin{aligned} a_x &= \omega(y - y_0), \\ a_y &= -\omega(x - x_0), \\ a_z &= 0, \end{aligned}$$

and the angular velocity is $\omega = 2\pi 10^4$.

The initial aligned solution is defined as a Gaussian blob described by the function

$$u_0 = \exp(-\frac{(x - x_s)^2 + (y - y_s)^2}{r_g^2}),$$

with radius $r_g = 0.04$. Hence, the maximum value of the initial solution is $u_0^{max} = 1$. The center of the blob $\boldsymbol{r}_s = \{x_s, y_s\}^T$ follows the magnetic field lines in its helicoidal shape,

$$\boldsymbol{r}_s = \mathcal{R}\boldsymbol{r}_0,$$

where $\boldsymbol{r}_0$ is the position of the center of the blob in the first poloidal plane, $\boldsymbol{r}_0 = \{2.885, 0\}$ and $\mathcal{R}$ is a rotation matrix

$$\mathcal{R} = \begin{bmatrix} \cos(\alpha) \sin(\alpha) \\ -\sin(\alpha) \cos(\alpha) \end{bmatrix}, \quad \alpha = 2\pi \frac{z}{z_{max}},$$

where z is the toroidal coordinate and z_{max} is chosen as 107, which provides a constant pitch-angle of 2.89°. The iso-surface at value 0.5 of the initial solution is depicted in Fig. 27.

A reference time T_r is defined, corresponding to the time needed for the maximum value of the solution to reach the value $u^{max} = 0.9$. Similarly to the previous example, a set of computations with isotropic diffusion is used to define the relation between the diffusion $k_\perp (= k_\parallel)$ and the reference time T_r. Then, two sets of simulations have been performed for anisotropic diffusion with $k_\perp = 0$ and $k_\parallel = 10^6, 10^9$.

For the toroidal discretization, the following combinations of number of elements and polynomial degrees are selected: 128 elements $p = 1$, 64 elements $p = 2$, 32 elements $p = 4$ and 16 elements $p = 8$. All these combinations consist in 128 nodes in the toroidal direction. The linear system size is 5326080 for the $p = 1$, 3312960 for the $p = 2$, 2306400 for the $p = 4$ and 1803120 for the $p = 8$.

All the computations are performed on a single Sky Lake node (32 CPUs, 192 Gb of RAM), using only the OpenMP parallelization algorithm. The linear system of the fully implicit discretization is solved using the PSBLAS library, see [40]. A test of robustness was performed with various iterative solvers available in the library, and finally a stabilized bi-conjugate gradient method with block Jacobi with ILU(0) factorization preconditioner was chosen for all the computations. This later proved to guarantee convergence at any anisotropy level.

Results are shown in Tab. 3, where the reference time T_r and the corresponding effective perpendicular diffusion $k_\perp^*$ are shown for the various cases. As expected, the numerical diffusion decreases exponentially as the polynomial degree is increased, which is also confirmed by the plot in Fig. 28. A visual representation of the turning $p = 8$ solution is depicted in Fig. 29, where the time intervals t =2.5e-5 between the solution is chosen.

9. Conclusions

A high-order HDG scheme has been presented for the discretization of strongly anisotropic elliptic problems in the context of magnetized plasmas for fusion applications. Based on a non flux-aligned unstructured finite-element mesh, the method is potentially able to discretize plasma facing components of any complex shape and

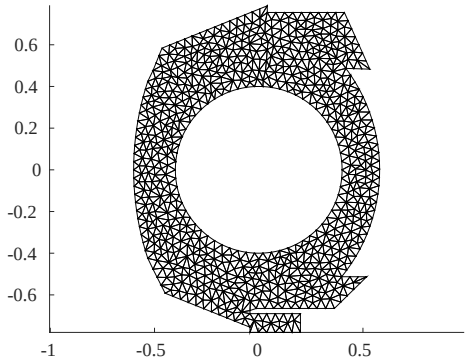


Figure 26: 3D test case: 2D map of the triangular mesh in the poloidal section of the tokamak WEST.

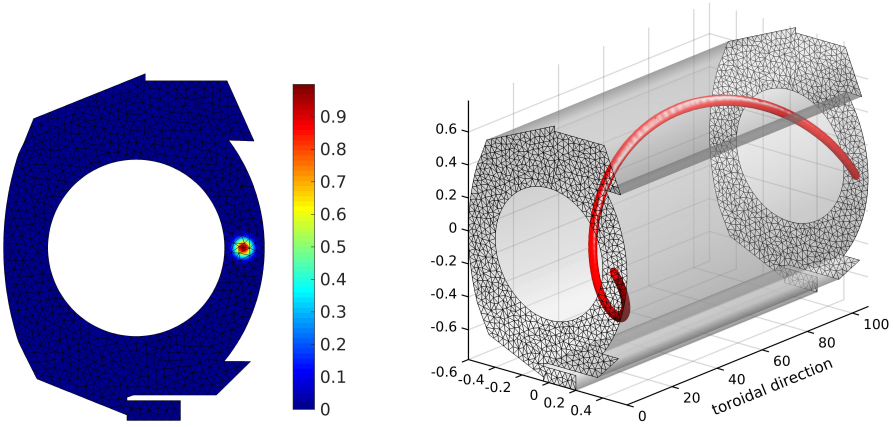


Figure 27: 3D test case and initial solution: 2D plot on the solution in the poloidal plane at $z = 0$ (left) and 3D view of the isosurface 0.5 (right).

Table 3: 3D test case: reference time and numerical effective diffusion for different discretization in the toroidal direction.

		$k_\parallel = 10^6$		$k_\parallel = 10^9$	
nt	p	T_r	$k_\perp*$	T_r	$k_\perp*$
128	1	7.6e-7	59	7.6e-10	55680
64	2	3.6e-6	12.6	3.6e-9	11927
32	4	1.9e-5	2.4	2.0e-8	2145
16	8	1.6e-4	0.3	1.9e-7	231

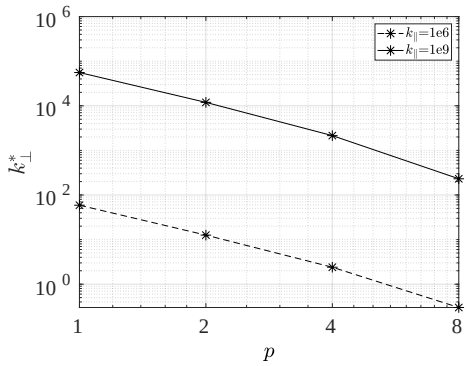


Figure 28: 3D test case: 1D plots of the effective perpendicular diffusion $k_\perp^*$ as a function of the polynomial degree in the toroidal discretization, for $nt = 128$.

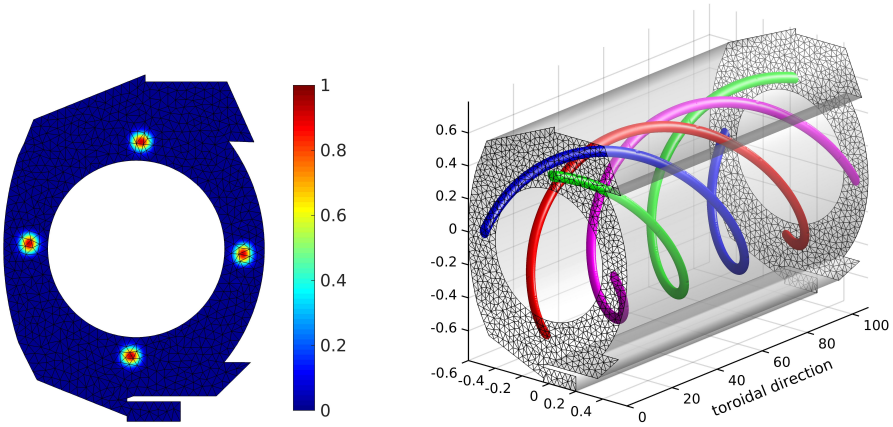


Figure 29: 3D test case and solutions at four different time intervals equispaced of $t = 2.5 \times 10^{-5}$ for the $p = 8$ computation. 2D map on the poloidal plane at $z = 0$ (left) and isosurface at value $0.5u_{\max}$ (right): solution at time $t = 2.5 \times 10^{-5}$ (red), $t = 5 \times 10^{-5}$ (blue), $t = 7.5 \times 10^{-5}$ (green), $t = 10^{-4}$ (magenta).

to handle the plasma in versatile magnetic equilibria, possibly extended up to the center as already shown in Ref. [30]. The numerical diffusion, introduced by the non-aligned discretization, is controlled here by the use of high-order interpolations. A new stabilization parameter is also proposed in this context that allows to provide correct convergence slopes at any anisotropy level.

Numerical results on well-designed 2D and 3D test cases show that diffusion-free numerical solutions can be obtained as long as the interpolation order is high enough. Introducing an explicit measure of the spurious diffusion, an exponential decrease with the polynomial degree used in the computation is indeed shown.

The results presented in this paper show that non-aligned approach can be thus considered for transport simulations as long as high-order interpolations are used. In this context, the HDG method becomes a very attractive candidate for the development of a new family of numerical codes for plasma simulations in fusion.

Acknowledgements The project leading to this publication has received funding from Excellence Initiative of Aix-Marseille University-A*MIDEX, a French "Investissements d'Avenir programme". This work was granted access to the HPC resources of Aix-Marseille Universit financed by the project Equip@Meso (ANR-10-EQPX-29-01). This work was supported by the EUROfusion - Theory and Advanced Simulation Coordination (E-TASC) and has received funding from the Euratom research and training programme 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

[1] M. McNelis, D. Bazow, U. Heinz, Nuclear Physics A 982 (2019) 915–918.

[2] C. Lopez-Molina, M. Galar, H. Bustince, B. D. Baets, Pattern Recognition 47 (2014) 270–281.

[3] F. Hammou, K. Hammouche, J.-G. Postaire, Neurocomputing 306 (2018) 80–93.

[4] H. Ma, Y. Nie, Journal of Visual Communication and Image Representation 40 (2016) 406–417.

[5] A. Salama, A. Negara, M. E. Amin, S. Sun, Journal of Contaminant Hydrology 181 (2015) 114–130.

[6] M. H. Esfahani, H. R. Ghehsareh, S. K. Etesami, Engineering Analysis with Boundary Elements 82 (2017) 91–97.

[7] H. V. Dang, K. T. Ng, Journal of Neuroscience Methods 198 (2011) 359–363.

[8] W. J. Denny, M. T. Walsh, Journal of Biomechanics 47 (2014) 168–177.

[9] W. D'haeseleer, Flux coordinates and magnetic field structure: a guide to a fundamental tool of plasma structure, Springer series in computational physics, Springer-Verlag, 1991.

[10] B. Dudson, M. Umansky, X. Xu, P. Snyder, H. Wilson, Comput. Phys. Commun. 180 (2009) 1467–1480.

[11] P. Tamain, H. Bufferand, G. Ciraolo, C. Colin, D. Galassi, P. Ghendrih, F. Schwander, E. Serre, J. Comp. Phys. 321 (2016) 606–623.

[12] H. Bufferand, B. Bensiali, J. Bucalossi, G. Ciraolo, P. Genesio, P. Ghendrih, Y. Marandet, A. Paredes, F. Schwander, E. Serre, P. Tamain, J. Nucl. Mat. 84 (2013) 445–448.

[13] J. Soler, F. Schwander, G. Giorgiani, J. Liandrat, P. Tamain, E. Serre, Journal of Computational Physics 405 (2019) 109093. 10.1016/j.jcp.2019.109093.

[14] S. Günter, Q. Yu, J. Krger, K. Lackner, J. Comp. Phys. 209 (2005) 354–370.

[15] S. Günter, K. Lackner, C. Tichmann, J. Comp. Phys. 226 (2007) 2306–2316.

[16] P. Paruta, P. Ricci, F. Riva, C. Wersal, C. Beadle, B. Frei, Physics of Plasmas 25 (2018) 112301.

[17] M. Ottaviani, Phys. Lett. A 375 (2011) 1677–1685.

[18] B. van Es, B. Koren, H. J. de Blank, J. Comp. Phys. 272 (2014) 526–549.

[19] A. Stegmeir, D. Coster, O. Maj, K. Hallatschek, K. Lackner, Comput. Phys. Commun. 198 (2016) 139–153.

[20] A. Stegmeir, D. Coster, A. Ross, O. Maj, K. Lackner, E. Poli, Plasma Physics and Controlled Fusion 60 (2017).

[21] D. B., H. J.H., Journal of Computational Physics 408 (2020) 109273.

[22] R. Holleman, O. Fringer, M. Stacey, Int. J. Numer. Meth. Fl. 72 (2013) 1117–1145.

[23] H. Guillard, J. Lakhlili, A. Loseille, A. Loyer, B. Nkonga, A. Ratnani, A. Elarif, Tokamesh : A software for mesh generation in Tokamaks, Research Report RR-9230, CASTOR, 2018. URL: `https://hal.inria.fr/hal-01948060`.

[24] H. Guillard, J. Lakhlili, A. Loyer, A. Ratnani, E. Sonnendrücker, Mesh generation for fusion applications, Technical Report, 2018. URL: `https://hal.inria.fr/hal-01950388`, poster.

[25] E. Meier, V. Lukin, U. Shumlak, Comput. Phys. Commun. 181 (2010) 837–841.

[26] N. Crouseilles, M. Kuhn, G. Latu, J. Sci. Comput. 65 (2015) 1091–1128.

[27] M. Held, M. Wiesenberger, A. Stegmeir, Comput. Phys. Commun. 199 (2016) 29–39.

[28] B. Cockburn, B. Dong, J. Guzmán, Math. Comp. 77(264) (2008) 1887–1916.

[29] A. Samii, C. Michoski, C. Dawson, Comput. Meth. Appl. Mech. Eng. 304 (2016) 118–139.

[30] G. Giorgiani, H. Bufferand, G. Ciraolo, P. Ghendrih, F. Schwander, E. Serre, P. Tamain, J. Comp. Phys. 374 (2018) 515–532.

[31] G. Giorgiani, H. Bufferand, G. Ciraolo, E. Serre, P. Tamain, Nuclear Materials and Energy 19 (2019) 340–345.

[32] A. Montlaur, S. Fernández-Méndez, A. Huerta, Int. J. Numer. Meth. Fl. 57(9) (2008) 1071–1092.

[33] M. A. Taylor, B. A. Wingate, R. E. Vincent, SIAM J. Numer. Anal. 38 (2000) 1707–1720.

[34] B. Cockburn, J. Gopalakrishnan, R. Lazarov, SIAM J. Numer. Anal. 47(2) (2009) 1319–1365.

[35] N. C. Nguyen, J. Peraire, B. Cockburn, J. Comput. Phys. 228(9) (2009) 3232–3254.

[36] N. C. Nguyen, J. Peraire, B. Cockburn, Comput. Methods Appl. Mech. Eng. 199(2) (2010) 582–597.

[37] N. C. Nguyen, J. Peraire, B. Cockburn, J. Comput. Phys. 228(23) (2009) 8841–8855.

[38] N. C. Nguyen, J. Peraire, B. Cockburn, J. Comput. Phys. 230(4) (2011) 1147–1170.

[39] J. Bucalossi, *et al.*, Fusion Eng.& Design 89(7-8) (2014) 907–912.

[40] S. Filippone, ACM Trans. Math. Softw. 26 (2000) 527–550.