Recent upgrades in a 2D turbulent transport solver based on a hybrid discontinuous Galerkin method for the simulation of fusion plasma in tokamak

Giacomo Piraccini, Marcello Capasso, Manuel Scotto d'Abusco, Giorgio Giorgiani, Frédéric Schwander, Eric Serre, Hugo Bufferand, Guido Ciraolo, Patrick Tamain

MDPI

Article

Recent upgrades in a 2D turbulent transport solver based on a hybrid discontinuous Galerkin method for the simulation of fusion plasma in tokamak

Giacomo Piraccini [1,*], Marcello Capasso [1], Manuel Scotto D'Abusco [1,2], Giorgio Giorgiani [1], Frédéric Schwander [1], Eric Serre [1], Hugo Bufferand [2], Guido Ciraolo [2] and Patrick Tamain [2]

[1] Aix Marseille Univ, CNRS, Centrale Marseille, M2P2, Marseille, France; eric.serre@univ-amu.fr
[2] IRFM, CEA Cadarache, F-13108 St. Paul-lez-Durance, France; hugo.bufferand@cea.fr (H.B.); guido.ciraolo@cea.fr (G.C.); patrick.tamain@cea.fr (P.T.);
* Correspondence: giacomo.piraccini@univ-amu.fr;

Abstract: The simulation of fusion plasmas in realistic magnetic configurations and tokamak geometries still requires the development of advanced numerical algorithms owing to the complexity of the problem. In this context, we propose a Hybrid Discontinuous Galerkin (HDG) method to solve 2D transport fluid equations in realistic magnetic and tokamak wall geometries. This high-order solver can handle magnetic equilibrium free structured and unstructured meshes allowing a much more accurate discretization of the plasma facing components than current solvers based on magnetic field aligned methods associated to finite-differences (volumes) discretization. In addition, the method allows to handle realistic magnetic equilibrium, eventually non steady, a critical point in the modelling of full discharges including ramp up and ramp down phases. In this paper, we introduce the HDG algorithm with a special focus on recent developments related to the treatment of the cross-field diffusive terms, and to an adaptive mesh refinement technique improving the numerical efficiency and robustness of the scheme. The updated solver is verified with a manufactured solution method, and numerical tests are provided to illustrate the new capabilities of the code.

Keywords: hybrid discontinuous Galerkin; fusion plasma modelling; tokamak; adaptive refinement

Citation: Piraccini, G.; Capasso, M.; Scotto D'Abusco, M.; Giorgiani, G.; Schwander, F.; Serre, E.; Bufferand, H.; Ciraolo, G.;Tamain, P.; Recent upgrades in a 2D turbulent transport solver based on a hybrid discontinuous Galerkin method for the simulation of fusion plasma in tokamak. *Journal Not Specified* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Research in magnetic confinement fusion plasmas explores the possibility of producing carbon-free electric power by using fusion in deuterium-tritium plasmas heated to temperatures up to 10^7-10^8K, and confined by magnetic field in machines of toroidal shape known as tokamaks. With ITER and the promise of burning plasmas, the control of heat exhaust in high energy confinement configurations has become a topic of critical importance for the operation [20]. The difficulty to get global experimental measurements in tokamak makes complementary numerical simulations in realistic tokamak conditions a valuable asset to design optimised plasma scenarios, allowing to control the heat outfluxes and to prevent material damages. However, such numerical simulation remains a very challenging issue. This problem is multi-physics and multi-scales due to plasma wall interactions and turbulence. The geometry adds also a complexity in realistic configurations due to the shape of the tokamak wall and of the magnetic equilibrium. In addition, the strong anisotropy of the magnetic field components leads to a preferred orientation denoted as the parallel direction, with reference to the direction along the magnetic field lines. This leads to specific numerical issues as ill-conditioned algebraic operators to invert, and significant spurious numerical diffusion in the direction orthogonal to the anisotropy direction. Routine simulations able to provide information in acceptable timings in a tokamak of the size of ITER are still today restricted to 2D

models based on averaged axisymmetric fluid-drift Braginskii equations [2–4]. The current code of the fusion community are generally based on first and second order finite differences or finite volumes, and so on structured meshes. Their discretization is aligned along the magnetic field lines to take advantage of the transport features and limit spurious numerical diffusion [19]. Thus, the accurate discretization of realistic tokamak wall geometries as well as plasma regions around singularities such the X-point or the tokamak center remains challenging with these codes. In addition, the simulation of transient phases of the plasma discharge when the plasma equilibrium moves is not affordable without a very expensive on the fly re-meshing of the computational domain. To overcome these limitations, we have recently considered an Hybrid Discontinuous Galerkin (HDG) method. Such discretization based on structured or unstructured meshes is magnetic equilibrium free that allows accurate simulations of the whole vacuum chamber whatever the geometrical complexity of the tokamak wall or the magnetic equilibrium shape. It also allows to handle a non-steady magnetic equilibrium [1] – a critical point to model a full discharge including start-up and shut-down phases [17]. The ramp up and shut-down phases last about 30% of the full discharge time (a few seconds). These times are however long compared to characteristic turbulence times which are of the order of few micro-seconds. Indeed, the high-order accuracy of the spatial discretization allows controlling the spurious numerical diffusion despite the strong anisotropy, as recently shown in [10] when increasing the order of interpolation p for a fixed spatial resolution. In this paper, we present an updated algorithm for solving 2D fluid-drift Braginskii equations in realistic tokamak geometries. The algorithm has been modified to handle nonlinear perpendicular diffusion terms with independent coefficients for each flow variable that allows a much more accurate description of the perpendicular transport related to turbulence. In addition, a h-refinement technique has been implemented to improve the numerical performance both in terms of memory and CPU time. This adaptive mesh refinement method can dynamically re-adjust the mesh locally according to error estimators based on the output data. The first results show it improves the global accuracy of the solution without using a global refinement of the mesh in the whole computational domain. The paper begins by introducing the physical model (section 2) and the general features of the numerical algorithm (section 3). The original developments are presented in sections 4 and 5 for the diffusive cross-field terms and the h-refinement technique, respectively. Concluding remarks and perspectives are summarised in section 6.

2. Physical model

The 2D computational domain mimics actual tokamaks with limiter or X-point and correspond to the entire volume of plasma going from the core up to the wall as shown on Figure 1.

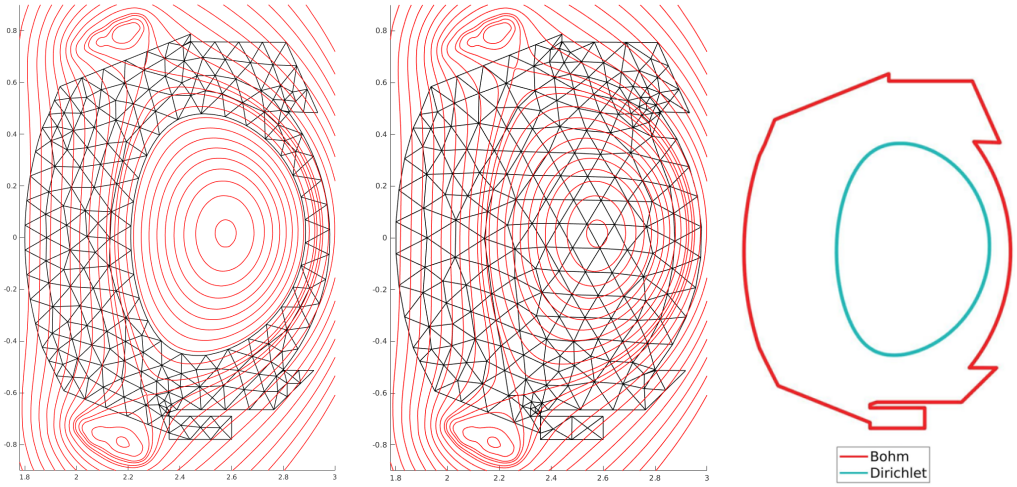

Figure 1. WEST tokamak poloidal cross-section. Example of typical triangular meshes restricted at the plasma edge (left) or in the whole section (center). On the right, sketch of the computational domain with boundary conditions for plasma edge simulations (Sec.2.2). The lines correspond to the magnetic flux surface as assigned in the code.

The magnetic field $\boldsymbol{B}$ is assigned including both closed flux surfaces in the center and open flux surfaces with field lines impacting the wall at the edge. These flux surfaces are separated by a magnetic field line in the poloidal called the separatrix in the poloidal cross-section. The strong difference of intensity between the toroidal and poloidal components $||B_p|| << ||B_t||$ defines a privileged direction denoted as the parallel direction, with reference to the direction along the magnetic field lines. To take advantage of this flow anisotropy, the equations are projected along the magnetic field lines using the differential operator $\boldsymbol{\nabla}_{\parallel} = \boldsymbol{b} \cdot \boldsymbol{\nabla}$ and $\boldsymbol{\nabla}_{\perp} = \boldsymbol{\nabla} - \boldsymbol{b} \cdot \boldsymbol{\nabla}$, where $b = \frac{\boldsymbol{B}}{||\boldsymbol{B}||}$ is the unitary vector in the parallel direction.

2.1. Equations of the model

The mathematical model relies on 2D fluid conservation equations based on Braginskii simplified closures [8]. Under some hypothesis and ordering detailed in Ref. [21], it corresponds to a standard model in the fusion community of advection diffusion equation that governs the transport of the mean plasma quantities as the density n the parallel momentum nu, and the ion and electron total energy $E_i = \frac{3}{2}k_b T_i + \frac{1}{2}m_i u^2$ and $E_e = \frac{3}{2}k_b T_e$, respectively, with m_i is the mass of the ion and T_i and T_e are the ion and electron temperatures, respectively. The conservation equations below correspond to a compressible adiabatic gas in the parallel direction and to an incompressible fluid in the perpendicular direction where turbulence process dominates. The system writes:

$$\partial_t n + \nabla \cdot (nu\mathbf{b}) - \nabla \cdot (D\nabla_{\perp} n) = S_n \tag{1}$$

$$\partial_t (m_i nu) + \nabla \cdot (m_i nu^2 \mathbf{b}) + \nabla_{\parallel}(k_b n(T_e + T_i)) - \nabla \cdot (\mu \nabla_{\perp}(m_i nu)) = S_{\Gamma} \tag{2}$$

$$\begin{aligned} &\partial_t \left(\frac{3}{2} k_b n T_i + \frac{1}{2} m_i n u^2 \right) + \nabla \cdot \left(\left(\frac{5}{2} k_b n T_i + \frac{1}{2} m_i n u^2 \right) u\mathbf{b} \right) - nueE_{\parallel} \\ &- \nabla \cdot \left(\frac{3}{2} k_b (T_i D \nabla_{\perp} n + n\chi_i \nabla_{\perp} T_i) \right) - \nabla \cdot \left(-\frac{1}{2} m_i u^2 D \nabla_{\perp} n + \frac{1}{2} m_i \mu n \nabla_{\perp} u^2 \right) \\ &- \nabla \cdot (k_{\parallel i} T_i^{\frac{5}{2}} \nabla_{\parallel} T_i \mathbf{b}) + \frac{3}{2} \frac{k_b n}{\hat{\tau}_{ie}} (T_e - T_i) = S_{E_i} \end{aligned} \tag{3}$$

$$\partial_t\left(\frac{3}{2}k_b n T_e\right) + \nabla\cdot\left(\frac{5}{2}k_b n T_e u\mathbf{b}\right) + nueE_{\parallel} - \nabla\cdot\left(\frac{3}{2}k_b(T_e D\nabla_{\perp} n + n\chi_e\nabla_{\perp}T_e)\right) - \nabla\cdot(k_{\parallel e}T_e^{\frac{5}{2}}\nabla_{\parallel}T_e\mathbf{b}) - \frac{3}{2}\frac{k_b n}{\hat{\tau}_{ie}}(T_e - T_i) = S_{E_e} \tag{4}$$

where p_i and p_e are the diagonal part of the ion and electron pressure stress tensor and they are equal to $p_i = nk_bT_i$ and $p_e = nk_bT_e$ $[m^{-1}s^{-2}]$, respectively. The constant diffusion coefficients that take into account the collisions transport and turbulent effects in the cross field direction are denoted D, μ, χ_i and χ_e for n, nu, E_i and E_e, respectively. Their values are chosen as a compromise between estimations provided by theory or experimental measurements and numerical stability constraints. They are usually less or equal to 1 m^2s^{-1}. The terms $(k_{||,i}T_i^{5/2})$ and $(k_{||,e}T_e^{5/2})$ correspond to nonlinear parallel diffusions for ion and electron, respectively. The parallel diffusion coefficients depend on the mass of the species, and are equal for the deuterium to $k_{||,i} = 60$ $[Wm^{-1}eV^{-7/2}]$ and $k_{\parallel,e} = 2000$ $[Wm^{-1}eV^{-7/2}]$. The parameter $\hat{\tau}_{ie}$ is the relaxation time for the collisions coupling term between electrons and ions, $W = \frac{3}{2}\frac{k_b n}{\hat{\tau}_{ie}}(T_e - T_i)$. It is defined as:

$$\hat{\tau}_{ie} = \frac{3\sqrt{2}}{e^4}\frac{\varepsilon_0^2}{\Lambda}\pi^{\frac{3}{2}}\frac{m_i}{m_e}\sqrt{m_e}e^{\frac{3}{2}}\frac{T_e^{\frac{3}{2}}}{n}$$

where the Coulomb logarithm $\Lambda = 12$, the ionic mass $m_i = 3.35\cdot10^{-27}$ $[kg]$, the electronic mass $m_e = 9.11\cdot10^{-31}$ $[kg]$, the vacuum permeability $\varepsilon_0 = 8.85\cdot10^{-12}$ $[C\,N^{-1}\,m^{-1}]$ and the electron charge $e = 1.60\cdot10^{-19}$ $[C]$. Finally, S_n,S_Γ, S_{E_i}, S_{E_e} correspond to sources terms.

2.2. Boundary conditions

In the direction parallel to the magnetic field lines, the boundary conditions for the plasma are specific and correspond to the Bohm boundary conditions modelling plasma wall interactions [23]. They assume a parallel velocity of the plasma equal or larger than the sound speed $c_s = \sqrt{\frac{k_b(T_e+T_i)}{m_i}}$ and leave free the density value at the wall that corresponds to ([1]):

$$\begin{aligned} u &\geq c_s \quad if \quad \mathbf{b}\cdot\mathbf{n} > 0 \\ u &\leq -c_s \quad if \quad \mathbf{b}\cdot\mathbf{n} < 0 \end{aligned} \tag{5}$$

where $\boldsymbol{n}$ is the outer normal of the surface. For the electrons and ions energy equations, the Bohm conditions impose the parallel fluxes to the sheath transmission values, leading to:

$$\begin{aligned} (nE_i + p_i)u - \frac{k_{\parallel i}}{m_i}T_i^{5/2}\nabla_{\parallel}T_i &= \gamma_i u p_i + \frac{1}{2}nu^3 \\ (nE_e + p_e)u - \frac{k_{\parallel e}}{m_i}T_e^{5/2}\nabla_{\parallel}T_e &= \gamma_e u p_e \end{aligned} \tag{6}$$

where $\gamma_i = 2.5$ and $\gamma_e = 4.5$. In the perpendicular direction to the magnetic field lines, homogeneous Neumann conditions are considered for all variables.

3. The hybrid discontinuous Galerkin method

A specific hybrid discontinuous Galerkin (HDG) algorithm has been developed for many years [1,10–12], and implemented in the family of codes SOLEDGE3X [4], well-known in the international fusion community to efficiently address turbulent transport in different machines all around Europe. A complete description of the method is provided in Appendix A as well as in former papers [1,10–12]. In HDG, the system of equations 1-4 is written in terms of conservative variables considering the vector:

$$\boldsymbol{U} = \{U_1, U_2, U_3, U_4\}^T = \{n, nu, nE_i, nE_e\}^T$$

where the superscript $\odot^T$ stands for transpose. The discontinuous partition induces a two-steps problem. In a first step, the set of conservative equations written in a weak formulation is solved element by element to express the discrete unknowns $\mathbf{U}(\mathbf{x},\mathbf{t})$ at the element nodes in terms of another approximation of the solution, called the trace solution $\hat{\mathbf{U}}$, which is defined on the borders of the element. In a second step, a global equation is set by imposing in a weak form the continuity of the fluxes across the borders of the elements to obtain $\hat{\mathbf{U}}$ in the whole mesh skeleton. Once $\hat{\mathbf{U}}$ is obtained, it is possible to recover the elementary solution $\mathbf{U}$ on each element using a local post processing. The introduction of this trace solution restricted to the skeleton of the mesh leads to a linear system of smaller size than in a classical discontinuous Galerkin method. The time discretization is fully implicit, and the non-linear terms are linearized using a classic iterative Newton-Raphson method.

4. Implementation of independent nonlinear diffusive cross-field terms

In the model introduced in section 2, the cross-field transport coefficients for $n,\ u,\ T_i,\ T_e$ play a fundamental role in the reliability of the solutions by modelling the perpendicular anomalous transport of particles and energy. Thus, their values directly impact the balance between the parallel and perpendicular transport which governs the plasma flow in the tokamak. With the implicit time integration scheme, the implementation of diffusion coefficients non equal for each flow variable is not straightforward. In this case indeed, the expression of the coefficients in function of conservatives variables introduce additional nonlinear coupling between the equations as described thereafter. When assuming $D=\mu=\chi_i=\chi_e$, the terms of the perpendicular dynamics in equation A2 depend only linearly of the unknown $\mathbf{Q}$ as follows:

$$-\nabla\cdot\begin{Bmatrix} D\nabla_\perp(n)\\ \mu\nabla_\perp(nu)\\ \chi_i\nabla_\perp(nE_i)\\ \chi_e\nabla_\perp(nE_e)\end{Bmatrix} = -D_f\boldsymbol{Q}+D_f\boldsymbol{Qb}\otimes\boldsymbol{b} \tag{7}$$

which represents the gradient of the conservative variable $\mathbf{U}$. When these coefficients are chosen non-equal, a nonlinear dependency occurs in the equations system written in conservative variables that writes as:

$$\begin{aligned}
h_\Gamma &= \mathbf{Q_{t,\perp}}\cdot\mathbf{W}_\Gamma = \mathbf{Q_{t,\perp}}\cdot\begin{bmatrix}(D-\mu)\frac{U_2}{U_1}\\0\\0\\0\end{bmatrix}+\mu\mathbf{Q_{2,\perp}}\\
h_{E_i} &= \chi_i\nabla_\perp(nE_i) = \mathbf{Q_{t,\perp}}\cdot\mathbf{W}_{E_i} = \mathbf{Q_{t,\perp}}\cdot\begin{bmatrix}(D-\chi_i)\frac{U_3}{U_1}(D-\mu)\frac{U_2^2}{U_1^2}\\-(D-\mu)\frac{U_2}{U_1}\\0\\0\end{bmatrix}+\chi_i\mathbf{Q_{3,\perp}}\\
h_{E_e} &= \mathbf{Q_{t,\perp}}\cdot\mathbf{W}_{E_e} = \mathbf{Q_{t,\perp}}\cdot\begin{bmatrix}(D-\chi_e)\frac{U_4}{U_1}\\0\\0\\0\end{bmatrix}+\chi_e\mathbf{Q_{4,\perp}}
\end{aligned} \tag{8}$$

Notice here that for $D=\mu=\chi_i=\chi_e$, equation 8 writes as equation 7. The linearization and integration of these nonlinear additional terms is detailed thereafter.

4.1. Linearization

The nonlinear terms of equation 8 are written as:

$$\mathbf{h}(\mathbf{U},\mathbf{Q}) = \mathbf{Q_t}\mathbf{W}(\mathbf{U}) \tag{9}$$

that linearize according to the formula A15 as:

$$\begin{aligned}
\mathbf{h}(\mathbf{U}^k,\mathbf{Q}^k) &= \mathbf{h}(\mathbf{U}^{k-1},\mathbf{Q}^{k-1}) + \frac{d}{d\epsilon}\mathbf{h}(\mathbf{U}^{k-1}+\epsilon\mathbf{dU},\mathbf{Q}^{k-1}+\epsilon\mathbf{dQ})|_{\epsilon=0}+ \\
&\quad + O(\mathbf{dU}^2,\mathbf{dQ}^2) = \\
&= \mathbf{Q}^{k-1}\mathbf{W}(\mathbf{U}^{k-1}) + \frac{d}{d\epsilon}((\mathbf{Q}^{k-1}+\epsilon\mathbf{dQ})\mathbf{W}(\mathbf{U}^{k-1}+\epsilon\mathbf{dU}))|_{\epsilon=0}+ \\
&\quad + O(\mathbf{dU}^2,\mathbf{dQ}^2) = \\
&= \mathbf{Q}^{k-1}\mathbf{W}(\mathbf{U}^{k-1}) + \mathbf{dQ}\mathbf{W}(\mathbf{U}^{k-1}) + \mathbf{Q}^{k-1}\frac{d\mathbf{W}}{d\mathbf{U}}\Big|_{k-1}\mathbf{dU}+ \\
&\quad + O(\mathbf{dU}^2,\mathbf{dQ}^2) = \\
&= \mathbf{Q}^{k-1}\mathbf{W}(\mathbf{U}^{k-1}) + \mathbf{Q}^{k}\mathbf{W}(\mathbf{U}^{k-1}) - \mathbf{Q}^{k-1}\mathbf{W}(\mathbf{U}^{k-1})+ \\
&\quad + \mathbf{Q}^{k-1}\frac{d\mathbf{W}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k - \mathbf{Q}^{k-1}\frac{d\mathbf{W}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^{k-1} + O(\mathbf{dU}^2,\mathbf{dQ}^2) = \\
&= \mathbf{Q}^{k}\mathbf{W}(\mathbf{U}^{k-1}) + \mathbf{Q}^{k-1}\frac{d\mathbf{W}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k + O(\mathbf{dU}^2,\mathbf{dQ}^2)
\end{aligned} \tag{10}$$

where $\mathbf{dU}$ and $\mathbf{dQ}$ have been replaced by $\mathbf{dU} = \mathbf{U}^k - \mathbf{U}^{k-1}$ and $\mathbf{dQ} = \mathbf{Q}^k - \mathbf{Q}^{k-1}$. Then, $\mathbf{h}_\Gamma(\mathbf{U}^k,\mathbf{Q}^k)$, $\mathbf{h}_{E_i}(\mathbf{U}^k,\mathbf{Q}^k)$ and $\mathbf{h}_{E_e}(\mathbf{U}^k,\mathbf{Q}^k)$ linearize as:

$$\begin{aligned}
\mathbf{h}_\Gamma(\mathbf{U}^k,\mathbf{Q}^k) &= \mathbf{Q}^{k}\mathbf{W}_\Gamma(\mathbf{U}^{k-1}) + \mathbf{Q}^{k-1}\frac{d\mathbf{W}_\Gamma}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k + O(\mathbf{dU}^2,\mathbf{dQ}^2) \\
\mathbf{h}_{E_i}(\mathbf{U}^k,\mathbf{Q}^k) &= \mathbf{Q}^{k}\mathbf{W}_{E_i}(\mathbf{U}^{k-1}) + \mathbf{Q}^{k-1}\frac{d\mathbf{W}_{E_i}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k + O(\mathbf{dU}^2,\mathbf{dQ}^2) \\
\mathbf{h}_{E_e}(\mathbf{U}^k,\mathbf{Q}^k) &= \mathbf{Q}^{k}\mathbf{W}_{E_e}(\mathbf{U}^{k-1}) + \mathbf{Q}^{k-1}\frac{d\mathbf{W}_{E_e}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k + O(\mathbf{dU}^2,\mathbf{dQ}^2)
\end{aligned} \tag{11}$$

where:

$$\begin{aligned}
\frac{d\mathbf{W}_\Gamma}{d\mathbf{U}} &= \begin{bmatrix} -(D-\mu)\frac{U_2}{U_1^2} & (D-\mu)\frac{1}{U_1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\frac{d\mathbf{W}_{E_i}}{d\mathbf{U}} &= \begin{bmatrix} -\left((D-\chi_i)\frac{U_3}{U_1^2} + 2(D-\mu)\frac{U_2^2}{U_1^3}\right) & 2(D-\mu)\frac{U_2}{U_1^2} & (D-\chi_i)\frac{1}{U_1} & 0 \\ (D-\mu)\frac{U_2}{U_1^2} & -(D-\mu)\frac{1}{U_1} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
\frac{d\mathbf{W}_{E_e}}{d\mathbf{U}} &= \begin{bmatrix} -(D-\chi_e)\frac{U_4}{U_1^2} & 0 & 0 & (D-\chi_e)\frac{1}{U_1} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{12}$$

Defining now:

$$\begin{aligned}
\mathbf{h}_\Gamma^{\mathbf{U}} &= \mathbf{Q}^{k-1}\frac{d\mathbf{W}_\Gamma}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k; \qquad \mathbf{h}_\Gamma^{\mathbf{Q}} = \mathbf{Q}^{k}\mathbf{W}_\Gamma(\mathbf{U}^{k-1}) \\
\mathbf{h}_{E_i}^{\mathbf{U}} &= \mathbf{Q}^{k-1}\frac{d\mathbf{W}_{E_i}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k; \qquad \mathbf{h}_{E_i}^{\mathbf{Q}} = \mathbf{Q}^{k}\mathbf{W}_{E_i}(\mathbf{U}^{k-1}) \\
\mathbf{h}_{E_e}^{\mathbf{U}} &= \mathbf{Q}^{k-1}\frac{d\mathbf{W}_{E_e}}{d\mathbf{U}}\Big|_{k-1}\mathbf{U}^k; \qquad \mathbf{h}_{E_e}^{\mathbf{Q}} = \mathbf{Q}^{k}\mathbf{W}_{E_e}(\mathbf{U}^{k-1})
\end{aligned} \tag{13}$$

the split momentum diffusion terms write as:

$$\begin{aligned}\mathbf{h}_\Gamma &= \mathbf{h}_\Gamma^{\mathbf{U}} + \mathbf{h}_\Gamma^{\mathbf{Q}} \\ \mathbf{h}_{E_i} &= \mathbf{h}_{E_i}^{\mathbf{U}} + \mathbf{h}_{E_i}^{\mathbf{Q}} \\ \mathbf{h}_{E_e} &= \mathbf{h}_{E_e}^{\mathbf{U}} + \mathbf{h}_{E_e}^{\mathbf{Q}}\end{aligned} \tag{14}$$

The superscripts $\mathbf{U}$ and $\mathbf{Q}$ stand for the terms whose unknown are $\mathbf{U}^k$ and $\mathbf{Q}^k$, respectively. These terms must be now incorporated into the matrices of the discrete linear system.

4.2. The new discrete linear system

From equation 8, the diffusion terms can be actually written as the sum of two terms as $\mathbf{S}_d + D_f\mathbf{Q}$ with $\mathbf{S}_d = 0$ for $D = \mu = \chi_i = \chi_e$. To incorporate the new term $\mathbf{S}_d$ into the linear system, $\mathbf{S}_d$ is first written in the matrix form as:

$$\mathbf{S}_d = \mathbf{S}^{\mathbf{U}} + \mathbf{S}^{\mathbf{Q}} = \begin{bmatrix} 0 \\ \mathbf{h}_\Gamma^{\mathbf{U}} \\ \mathbf{h}_{E_i}^{\mathbf{U}} \\ \mathbf{h}_{E_e}^{\mathbf{U}} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{h}_\Gamma^{\mathbf{Q}} \\ \mathbf{h}_{E_i}^{\mathbf{Q}} \\ \mathbf{h}_{E_e}^{\mathbf{Q}} \end{bmatrix} \tag{15}$$

Focusing on the second equation of the system A12, the local problem writes as:

$$\begin{aligned}&(\boldsymbol{v}, \partial_t \boldsymbol{U})_{\Omega_i} - \Big(\boldsymbol{\nabla v}, \boldsymbol{F} - D_f\boldsymbol{Q} + D_f\boldsymbol{Q}\boldsymbol{b}\otimes\boldsymbol{b} - \boldsymbol{F}_t\Big)_{\Omega_i} + \\ &+ \big\langle \boldsymbol{v}, \Big(\hat{\boldsymbol{F}} - D_f\hat{\boldsymbol{Q}} + D_f\hat{\boldsymbol{Q}}\boldsymbol{b}\otimes\boldsymbol{b} - \hat{\boldsymbol{F}}_t\Big)\boldsymbol{n}\big\rangle_{\partial\Omega_i} + \Big(\boldsymbol{v}, \boldsymbol{f}_{E_{||}}\Big)_{\Omega_i} + \Big(\boldsymbol{v}, \boldsymbol{f}_{E_{EX}}\Big)_{\Omega_i} - \\ &- (\boldsymbol{v}, \boldsymbol{g})_{\Omega_i} - (\boldsymbol{\nabla v}, -\boldsymbol{S}^{\boldsymbol{U}} + \boldsymbol{S}^{\boldsymbol{U}}\boldsymbol{b}\otimes\boldsymbol{b})_{\Omega_i} - (\boldsymbol{\nabla v}, -\boldsymbol{S}^{\boldsymbol{Q}} + \boldsymbol{S}^{\boldsymbol{Q}}\boldsymbol{b}\otimes\boldsymbol{b})_{\Omega_i} \\ &+ \big\langle \boldsymbol{v}, \big(-\boldsymbol{S}^{\hat{\boldsymbol{U}}} + \boldsymbol{S}^{\hat{\boldsymbol{U}}}\boldsymbol{b}\otimes\boldsymbol{b}\big)\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i} + \big\langle \boldsymbol{v}, \big(-\boldsymbol{S}^{\hat{\boldsymbol{Q}}} + \boldsymbol{S}^{\hat{\boldsymbol{Q}}}\boldsymbol{b}\otimes\boldsymbol{b}\big)\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i} = (\boldsymbol{v}, \boldsymbol{s})_{\Omega_i}\end{aligned} \tag{16}$$

Using the convention introduced in B.3 the terms with the unknown $\mathbf{U}$ are inserted in the matrix of the local problem A_{uu} while the terms with the unknown $\mathbf{Q}$ are inserted in A_{uq}. Then, in the discrete local problem the new matrices write as:

$$\begin{aligned}A_{uu} &\implies A_{uu} - (\boldsymbol{\nabla v}, -\boldsymbol{S}^{\boldsymbol{U}} + \boldsymbol{S}^{\boldsymbol{U}}\boldsymbol{b}\otimes\boldsymbol{b})_{\Omega_i} + \big\langle \boldsymbol{v}, \big(-\boldsymbol{S}^{\hat{\boldsymbol{U}}} + \boldsymbol{S}^{\hat{\boldsymbol{U}}}\boldsymbol{b}\otimes\boldsymbol{b}\big)\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i} \\ A_{uq} &\implies A_{uq} - (\boldsymbol{\nabla v}, -\boldsymbol{S}^{\boldsymbol{Q}} + \boldsymbol{S}^{\boldsymbol{Q}}\boldsymbol{b}\otimes\boldsymbol{b})_{\Omega_i} + \big\langle \boldsymbol{v}, \big(-\boldsymbol{S}^{\hat{\boldsymbol{Q}}} + \boldsymbol{S}^{\hat{\boldsymbol{Q}}}\boldsymbol{b}\otimes\boldsymbol{b}\big)\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_i}\end{aligned} \tag{17}$$

Let's notice that the new matrices are just related to the second equation of the system A12, and the changes are limited to the local element by element problem. It is worthy to observe that in the formulation of the global problem, the perpendicular gradient term is included in the imposition of the normal fluxes at the element boundary in equation A14. Moreover, it is also present in the flux vector that defines the Bohm boundary condition where the normal gradient is imposed equal to 0. Also in this case the contribution of the split diffusion term has to be considered, and the matrices for the assembling of the global problem A_{ll}, A_{lq} are modified in the same way by the additional terms of equation 17.

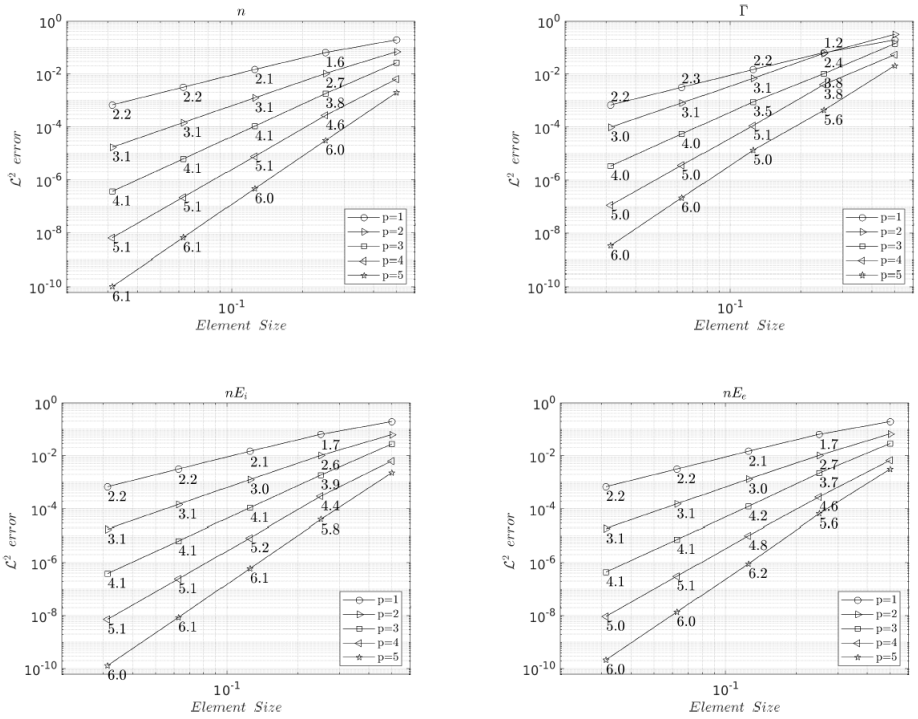

Figure 2. Convergence plots in $\mathcal{L}^2$-norm of all variables for different values of the polynomial interpolation p. $D = 0.1,\ \mu = 0.2,\ \chi_i = 0.3,\ \chi_e = 0.4\ m^2/s$.

4.3. Code verification

The Method of the Manufactured Solution (MMS) [1] is used to verify the code with the new formulation (Eq. 16). The transport coefficients are specially set all different from each other: $D = 0.1,\ \mu = 0.2,\ \chi_i = 0.3,\ \chi_e = 0.4\ m^2/s$. The following analytical solution is used with $\omega_x = \omega_y = 1$:

$$\begin{aligned} n &= 2 + \sin(2\pi\omega_x x)\sin(2\pi\omega_y y); \qquad & E_i &= 20 + \cos(2\pi\omega_x x)\sin(2\pi\omega_y y) \\ u &= \cos(2\pi\omega_x x)\cos(2\pi\omega_y y); & E_e &= 10 - \sin(2\pi\omega_x x)\cos(2\pi\omega_y y) \end{aligned} \tag{18}$$

Results of convergence plotted on Figure 2 show the expected theoretical rate of convergence in $p + 1$, and thus the correct implementation of the non-equal diffusion coefficients in the solver.

4.4. Example of simulation in the WEST tokamak

In order to show the new capability of the code to run with different cross-field coefficients, equations 1-4 are resolved in the WEST geometry (Figure1). We assume $\chi_i = \chi_e$ as it is usual in current computations of the literature, in agreement with experimental measurements carried out at the tokamak cross-section midplane [27]. For simplicity here we choose $\nu = \chi_i = \chi_e = 1$, and only D the particles diffusion is varied in a short range between 1 and 0.6 to avoid the use of too fine meshes. A mesh of 15591 elements with $p = 6$-elements is used. These steady state simulations require a run-time of about 40 minutes each on 32 cpu.

On Figure 3, the 2D contours for all flow variables are compared to ones obtained with a former version of the algorithm where all cross-field coefficients had to be equal to 1. The two solutions globally agree showing that the new version of the solver is able to provide 2D plasma equilibrium in realistic geometry. As expected, the solution at $D = 0.6$ however shows some differences. The contours are sharper contours in particular at the X-point and the density is higher in the core, of about a factor 1.6, since less matter is allowed to diffuse from it. On the parallel Mach number, the tongue of

positive velocity extends towards the top to the same extent while slightly decreasing its width, as to be expected for a lower density diffusion. Moreover, the parallel Mach number is higher at the X-points. Regarding now, the ion and electron temperatures, they are globally lower for $D = 0.6$, meaning that for this value of density diffusion, the plasma in the core has a higher density but lower temperatures.

5. Spatial adaptivity

Plasma solutions of interest for tokamak operation may exhibit large gradients both in the radial and parallel flow directions when targeting realistic conditions for the simulations, corresponding generally to small values of the cross-field diffusion coefficients [4]. This routinely lead to demanding requirements on the local spatial resolution of the mesh. In practice, failure to design a mesh that accommodates these resolution requirements result in aliasing errors in some elements of the mesh, that may lead to divergence of the Newton-Raphson iterations during the convergence toward the steady state solution. With the objective of enabling a robust numerical modelling of plasma transport in the edge, an adaptive *h*-refinement has been implemented. The *h*-refinement method is based here on an oscillation indicator to target flow regions with steep gradients or discontinuities inside the domain of computation. The element size is then optimised by imposing iterative, local mesh refinements in these flow regions while keeping a coarse mesh elsewhere [18].

5.1. Refinement process strategy

Experience in the computation of steady-state solutions of plasma transport in the edge has led to the emergence of a strategy combining Newton-Raphson iterations, with progressive lowering of cross-field diffusion coefficients in equations 1-4 in order to reach the desired value imposed by the simulation of tokamak operation (around $1\,\mathrm{m}^2{\cdot}\mathrm{s}^{-1}$ or lower). The Newton-Raphson iterative process is led to convergence for each value of the diffusion, and the obtained solution is used as initial condition of the Newton-Raphson iterations for the next smaller value of diffusion. The *h*-refinement is adopted for optimizing the mesh design, refining each element on which oscillations are detected. The procedure is stopped when the iterations reach the desired level of accuracy for the targeted diffusion coefficients values.

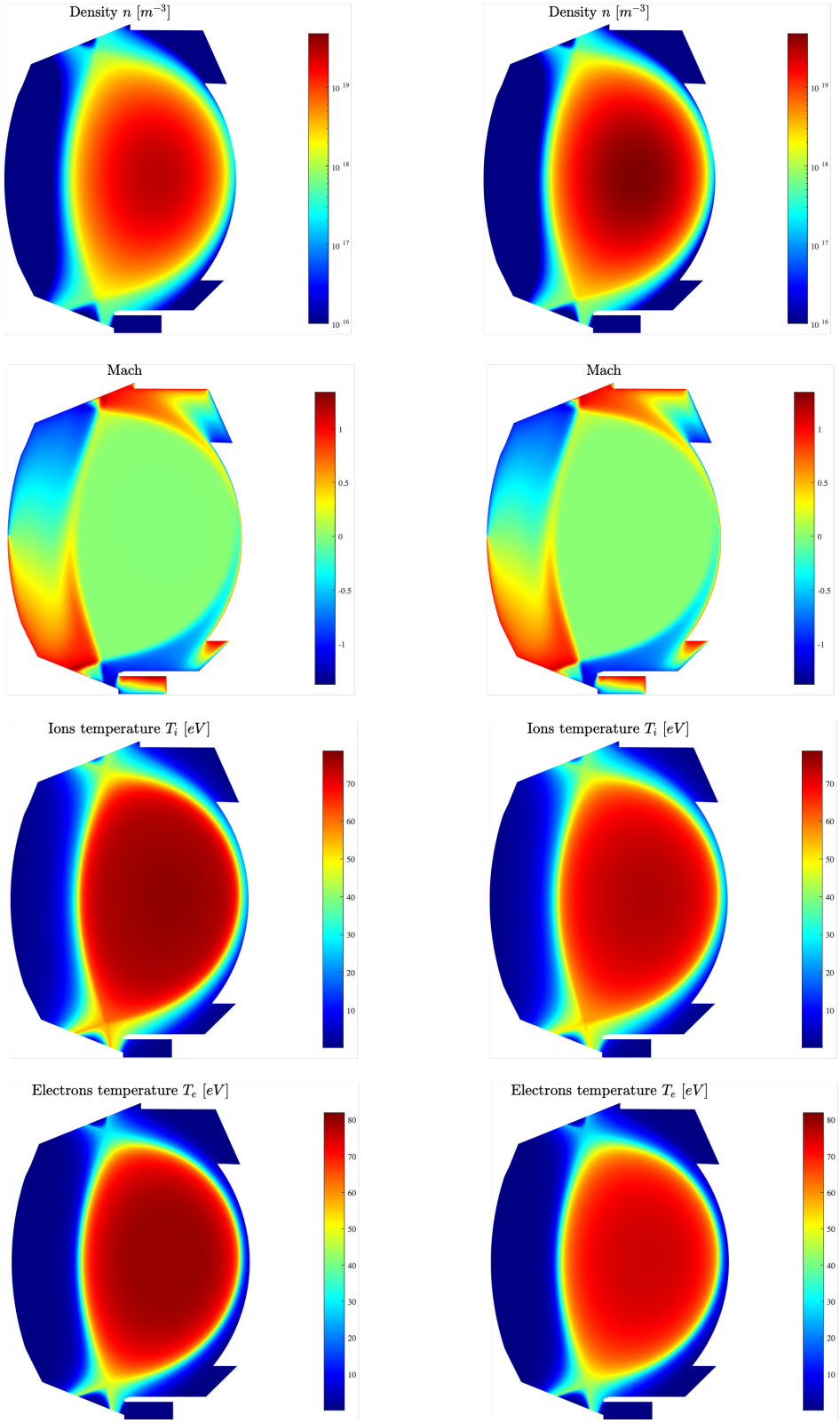

Figure 3. Large scale flows in the WEST tokamak poloidal cross-section. Isolines of density, parallel Mach number, ion and electron temperature at $\nu = \chi_i = \chi_e = 1$ and $D = 1$ on the conlumn on the left and $D = 0.6$ on the column on the right.

171 The whole process can be thus summarised as:

- Initialize the calculation with a rather coarse mesh and large values of cross-field diffusion coefficients
- Convergence to the steady solution using Newton-Raphson iterations
  - if convergence, computations are going on, lowering diffusion
  - if non convergence, the refinement procedure is started
    - Interpolation of the solution on the new mesh locally refined
    - Convergence to the steady solution using Newton-Raphson iterations
- Stop when diffusion coefficients reach the target values

The mesh refinement is performed using the open-source software Mmg [5,6]. It uses a map of elemental size in which the desired element size on each vertex must be precised. This current elemental size can be defined on each vertex of an existing mesh [16] using the elemental areas $\{|\Omega_k|\}$ as follows:

$$h_j = \frac{\sum_{i \in S_j} |\Omega_i| \tilde{h}_i}{\sum_{i \in S_j} |\Omega_i|} \tag{19}$$

in which S_j denotes the set of element indices having node j as a vertex. At the iteration n of the refinement process, a basic and straightforward formula provides a guess of the desired mesh size at the next iteration, on the element j where oscillations are detected, using the expression:

$$h_{target,j}^{(n+1)} = \frac{(h_j^{(n)})}{\alpha} \tag{20}$$

where α ($\alpha > 1$) is a control parameter to tune in order to perform the refinement. After several tests, the optimal value $\alpha = 2$ has been found. In this process, it is obvious that the mesh size is decreased locally and in an isotropic way. The possibility to coarsen the mesh has not been taken into account here because in the present configurations the initial meshes are already very coarse.

The efficiency of such a refinement strategy is mainly based on the choice of a suitable mesh refinement estimator. This estimator must be well-calibrated to avoid unnecessary costly over-refinements or, on the contrary, to keep spurious undetected oscillations in the solution. Here, the estimator can be more considered as an indicator uniquely able to identify spurious oscillations in the solution, related to unresolved steep gradients or discontinuities.

5.2. Oscillation-based error indicator

Adaptive mesh refinement is usually considered to converge to a numerical solution with a desired accuracy whilst using a minimal number of degrees of freedom. Adaptive mesh refinement is especially appealing in DG and HDG discretizations using *hp*-refinement as it warrants exponential convergence with the number of degrees of freedom [24]. The present refinement strategy is not drived by an accuracy criterion, but by a stability criterion to ensure the convergence of Newton-Raphson iterations towards the steady solution of equations 1, 2, 3 and 4. This strategy is based upon the observation that lack of convergence mostly stems from locally insufficient spatial resolution leading to aliasing errors. These errors deteriorate the convergence of the implicit solver and the global accuracy of the solution, and even more may lead to the divergence of the computations. This problem can be overcome by increasing the resolution locally to enhance the precision of the interpolation and to damp spurious oscillations. Usually, the estimators are based on the output data of the simulation [22] to detect oscillations. The technique is inspired from shock-capturing techniques [1], although here the quantity evaluated is an oscillation rather than a discontinuity in the solution. We use a simple sensor S_k, defined on each element with index k defined as a function of the parallel velocity u. For a computation with a polynomial approximation of order p, this sensor

consists in the norm of the local contribution of order p, normalized by the norm of the full solution on the element. It is thus defined as

$$S_k = \frac{(u - \hat{u}, u - \hat{u})_{\Omega_k}}{(u, u)_{\Omega_k}} \tag{21}$$

where u is the solution of order p, and $\hat{u}$ is the projection of the modal expansion on the space of polynomials of order $p - 1$.

5.3. Results

For simplicity, a reduced 2D fluid isothermal model is derived from equations 1-2 to solve the density n and the parallel momentum nu in a realistic WEST geometry (see in Ref. [1]). As in the complete model, Bohm boundary conditions are prescribed in the parallel direction to the magnetic field lines. Although simpler, this reduced model allows to evaluate most of the numerical issues. It takes into account the anisotropy in the flow dynamics between the parallel and perpendicular directions and the balance between the transport in the two flow directions is simply modulated by varying the diffusion D ($D = \mu$). Lowering D makes the parallel transport dominant that can be very demanding for the solver, particularly in the present configuration where the mesh is not aligned along the magnetic field lines [10]. Thus, the mesh has to be successively refined when decreasing D to converge toward a plasma equilibrium as already shown in [1] for uniform meshes. In addition, there is also a geometrical complexity with a magnetic equilibrium with two X-points as well as a tokamak wall with sharp edges and corners as well as small cavities around, Figure 1. This is thus an attractive configuration to test the local h-refinement technique proposed in this work.

Calculations are performed here using different meshes, automatically designed by the adative procedure described above. Only $p = 4$-polynomials are considered.

Typical contour plots of the density and the parallel Mach number are shown on Fig. 4 in the WEST poloidal cross-section for $D = 0,83\,\mathrm{m}^2{\cdot}\mathrm{s}^{-1}$. The large scales flows prediction shows very similar trends with respect to the literature [1]. The density is maximum at the core boundary, where the Dirichlet condition $n = 1$ is applied, and rapidly decreased to low values in the boundary layer, called the scrape-off layer (SOL), beyond the separatrix. The parallel Mach number u/c_s shows positive and negative Mach number regions, and a flow reversal around midplane. As expected from theoretical analysis [25] and from numerical investigations [26], the solution exhibits transitions to supersonic flows in the vicinity of both divertor legs.

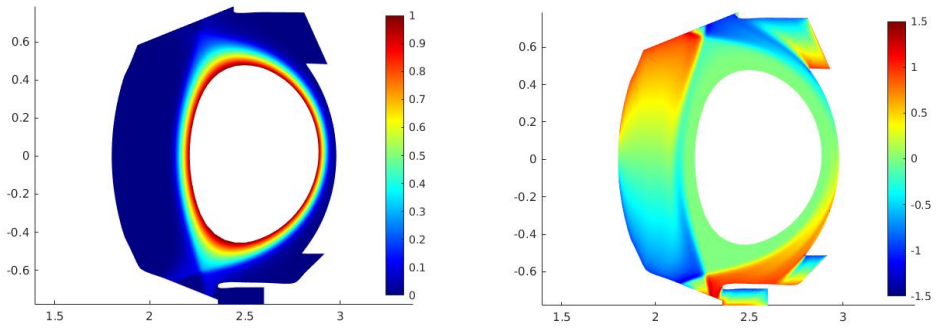

Figure 4. Large scale flows in the WEST tokamak poloidal cross-section. Isolines of density n (left) and parallel Mach number u/C_S (right). Computations are carried out for $D = \mu = 0.83(m^2s^1)$. Solutions are shown at the last iteration of the adaptive process.

To show the adaptive h-refinement process, Figure 5 shows the grid refinement at three successive steps for a diffusion $D = 2,63\,\mathrm{m}^2{\cdot}\mathrm{s}^{-1}$. For each mesh, the oscillations of the solution detected by the estimator are emphasised. Starting with a relatively

coarse mesh, the results show that the refinement process reduces the elements size only in the flow regions where oscillations are detected. Accordingly, the number of elements increases progressively in the poloidal cross-section with $N_e = 1192$, 2221 and 2554 but much less than if a uniform refinement had been considered. At the final step, oscillations are totally damped by the increase resolution around and the solver converges. With this procedure the mesh is automatically designed with a number of degree of freedom which is close to be optimal.

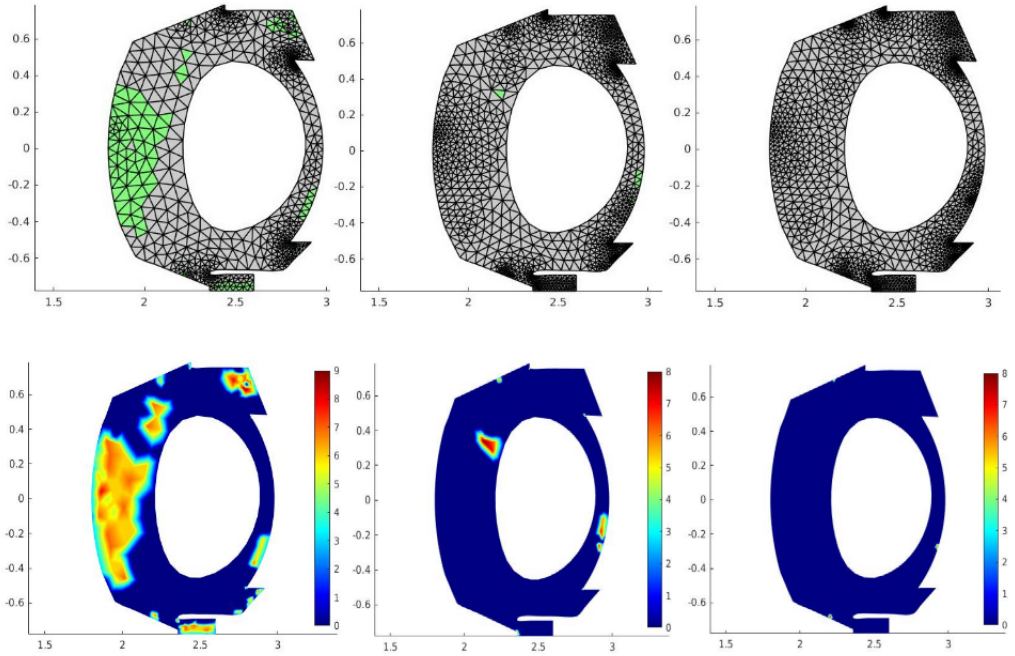

Figure 5. Meshes and solution oscillations at three steps during the adaptive h-refinement process for $D = 2.63\,\mathrm{m}^2\cdot\mathrm{s}^{-1}$. Meshes distribution with coloured elements corresponding to solution oscillations (top line). 2D maps of oscillations amplitude calculated on the nodes (bottom line). The colorbar shows the oscillations amplitude from Eq. 21 and averaged over neighbouring elements at every node.

As mentioned above, lowering the diffusion coefficient toward realistic values challenges the numerical solver by making the parallel transport dominant with oscillations if the resolution is not fine enough. The mesh must be then automatically adapted for each value of the cross-field diffusion to ensure the convergence of the algorithm. Once the solution converged, the diffusion is lowered again, and a new mesh is generated with an optimal design. This is shown on Figure 6 where the diffusion coefficient is progressively lowered of a factor 100, and the mesh automatically refined accordingly.

As soon as the mesh is fine enough, we can clearly expect to save on the time needed to converge. However, it is not straightforward to quantify precisely this saving. We have first compared the simulation times to convergence when lowering the diffusion coefficients (Figure 6) between simulations using the automative adaptive refinement procedure and simulations performed with an unique mesh for each value of the diffusion, corresponding to the most refined mesh designed during the automative procedure. Results are reported on Table 1 below, and show a saving of time up to 28% as D is strictly smaller than $D = 0.83\,m^2 \cdot s^{-1}$. Let's remind that target values for tokamak operation simulations are smaller than $D = 1\,m^2 \cdot s^{-1}$. As expected when the number of elements in the mesh is not high enough there is no saving, and even an additional cost due the time needed by the algorithm to design the mesh which is naturally not taken into account in the second set of simulations.

D $(m^2 \cdot s^{-1})$	N_e	**nDOF**	**h-refinement** (time (s))	**No h-refinement** (time (s))	**Time saving** (%)
26.31	388	5 820	13.96	14.20	+2%
8.32	1 192	17 880	30.84	35.57	+13%
2.63	3 219	49 590	126.51	131.67	+4%
0.83	6 066	114 120	203.75	280.78	+28%
0.26	10 032	150 480	285.61	366.41	+23%

Table 1. Cpu times in second to convergence depending on the diffusion coefficients and the corresponding meshes for simulations with and without h-refinement technique. N_e is the number of elements, nDOF is the number of degrees of freedom for $p = 4$-polynomials. Without h-refinement an unique mesh is used for each value of the diffusion, corresponding to the most refined mesh designed during the automative procedure.

As an additional information, we have compared times to converge at $D = 2.63\,\mathrm{m}^2{\cdot}\mathrm{s}^{-1}$ using the adaptive procedure described above (Figure 5) and an uniform mesh with elements size equal to the size of the smallest element provided by the adaptive procedure. Doing that, the respective meshes are composed by 3219 and 98372 elements, respectively. The corresponding times to converge are respectively equal to 126,51s and 5488s, that corresponds to an increase of a factor 43 when using an uniform mesh. Naturally, this is only informative since uniform meshes are rarely used, but the the time to design accurately a mesh for each value of the diffusion coefficient when lowering it can be long and impossible to estimate because depending on the user's skills. The automatic design of the mesh, which does not required any adjustment by hand, during the iterative process is clearly a great advantage of this procedure.

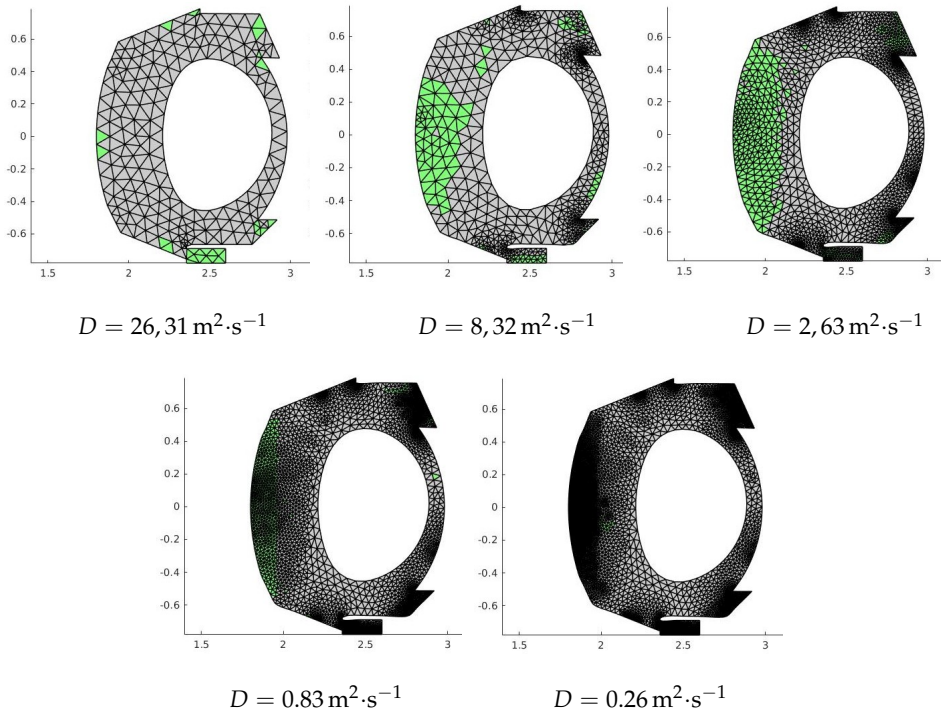

Figure 6. Five meshes and locations of the solution oscillations (colored areas) when lowering the cross-field diffusion coefficient. The corresponding numbers of elements and degree of freedom are given in Table 1.

6. Conclusions

This paper presents a high-order solver based on the Hybrid Discontinuous Galerkin method to perform plasma simulations in tokamak. It solves a 2D fluid transport model

for the density, parallel momentum, and the total energy for a deuterium plasma. This model is relevant with those currently implemented in fluid codes used in the fusion community. The main features of this solver are the use of unstructured meshes together with a high-order spatial approximation which allows to disalign the discretization from the magnetic field, unlike what is required in lower-order numerical schemes in order to control the spurious numerical diffusion due to the strong anisotropy of the flow. Thus, realistic tokamak wall geometries as well as magnetic equilibrium of complex shape and eventually unsteady can be accurately treated.

The code development is still in progress. In this paper, we have generalised the treatment of the cross-field diffusion terms. The possibility to handle diffusion coefficients chosen independently for each variable is a real improvement in the modelling of the cross-field turbulent transport. To progress toward better numerical performance, the first steps of a h-refinement technique have been introduced to optimise the mesh design and save cpu time and memory. Involving an error indicator based on spurious oscillations related to aliasing error, the mesh is refined locally and automatically around steep gradients of the solution that allows to damp efficiently the oscillations. This technique allows to save cpu time, and clearly improves the stability and the robustness of the algorithm.

This work is thus a step forward in the development of a very efficient and accurate numerical solver able to solve 2D transport fluid model in realistic tokamak configurations relevant for the operation.

Author Contributions: Conceptualization, E.S.; methodology, F.S. and P.T. and H.B.; software, G.G., G.P. and M.S.; validation, G.P. and M.C.; formal analysis, F.S.; investigation, G.P. and M.C and M.S.; resources, E.S., P.T. and G.C.; writing—original draft preparation, G.P.; writing—review and editing, E.S. and F.S.; visualization, ; supervision, F.S. and E.S.; project administration, E.S.; funding acquisition, E.S, P.T.. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work has been carried out thanks to the support of the A*MIDEX project (ANR-11-IDEX-0001 02, TOP project) funded by the 'Investissements d'Avenir' French Government program, managed by the French National Research Agency (ANR). This work has been also supported by the French National Research Agency grant SISTEM (ANR-19-CE46-0005-03).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Equations in conservative variables

This appendix details the base of the HDG algorithm used in this work.

Let's consider a computational domain Ω with closed boundary $\partial\Omega$ over a range of time $]0, T_f[$. The domain of computation Ω is divided in N_{el} disjoint elements Ω_i with boundaries $\partial\Omega_i$ such that:

$$\overline{\Omega} = \bigcup_{i=1}^{N_{el}} \overline{\Omega_i}, \qquad \Omega_i \cap \Omega_j = \emptyset \quad for \quad i \neq j, \quad and \quad \mathcal{T} = \bigcup_{i=1}^{N_{el}} \partial\Omega_i$$

Equations 1-4 must be written in conservative variables. Let's introduce $U = \{U_1, U_2, U_3, U_4\}^T = \{n, nu, nE_i, nE_e\}^T$ where the superscript $\odot^T$ stands for transpose. The plasma physical quantities u, p_i, p_e, T_i and T_e write in conservative variables as:

$$\begin{aligned} u &= \frac{U_2}{U_1}, \\ p_i &= \frac{2}{3M_{ref}}(U_3 - \frac{1}{2}\frac{U_2^2}{U_1}) \\ p_e &= \frac{2}{3M_{ref}}U_4, \\ T_i &= \frac{2}{3M_{ref}}(\frac{U_3}{U_1} - \frac{1}{2}\frac{U_2^2}{U_1^2}), \\ T_e &= \frac{2}{3M_{ref}}\frac{U_4}{U_1}. \end{aligned} \tag{A1}$$

where M_{ref} is a dimensionless parameter that appears by making the equations dimensionless, its value is $M_{ref} = \frac{T_0 e}{m_i u_0^2} \approx 12.5$, where e is the electron charge $(1.6e^{-19}\ C)$, m_i is the ion mass $(3.35e^{-27}\ kg)$, T_0 and u_0 are the reference temperature and velocity (50 eV and 1.3839 ms^{-1} respectively).

Equations 1-4 recast as:

$$\begin{cases} \boldsymbol{Q} - \boldsymbol{\nabla U} = 0 & in\ \Omega\times]0, T_f[\\ \partial_t \boldsymbol{U} + \boldsymbol{\nabla}\cdot(\boldsymbol{F} - D_f\boldsymbol{Q} + D_f\boldsymbol{Q}\boldsymbol{b}\otimes\boldsymbol{b} - \boldsymbol{F_t}) + \\ \qquad + \boldsymbol{f}_{E_{||}} + \boldsymbol{f}_{EX} - \boldsymbol{g} = \boldsymbol{s} & in\ \Omega\times]0, T_f[\\ \mathbf{U}(\mathbf{x}, 0) = \mathbf{U_0} & in\ \Omega \end{cases} \tag{A2}$$

where the new unknown $\mathbf{Q}$ is:

$$\mathbf{Q} = \nabla\mathbf{U} = \begin{bmatrix} \nabla\mathbf{U_1}^T \\ \nabla\mathbf{U_2}^T \\ \nabla\mathbf{U_3}^T \\ \nabla\mathbf{U_4}^T \end{bmatrix} = \begin{bmatrix} U_{1,x} & U_{1,y} \\ U_{2,x} & U_{2,y} \\ U_{3,x} & U_{3,y} \\ U_{4,x} & U_{4,y} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \\ Q_{31} & Q_{32} \\ Q_{41} & Q_{42} \end{bmatrix}$$

D_f is the diffusion tensor. It is diagonal only when the perpendicular transport coefficient are assumed equals to each other, i.e. $D = \mu = \chi_i = \chi_e$. The convective flux tensor $\mathbf{F}(\mathbf{U})$ is written as:

$$\boldsymbol{F} = \begin{Bmatrix} nu \\ nu^2 + M_{ref}(p_i + p_e) \\ (nE_i + M_{ref}p_i)u \\ (nE_e + M_{ref}p_e)u \end{Bmatrix} \otimes \mathbf{b^T} = \begin{Bmatrix} U_2 \\ \frac{U_2^2}{U_1} + \frac{2}{3}\left(U_3 + U_4 - \frac{1}{2}\frac{U_2^2}{U_1}\right) \\ \left(U_3 + \frac{2}{3}\left(U_3 - \frac{1}{2}\frac{U_2^2}{U_1}\right)\right)\frac{U_2}{U_1} \\ (U_4 + \frac{2}{3}U_4)\frac{U_2}{U_1} \end{Bmatrix} \otimes \boldsymbol{b^T}$$

The ion and electron temperature gradients have to be written in terms of conservative variables. For the ion, the gradient writes as:

$$\nabla T_i = \frac{2}{3M_{ref}}\nabla\Big(\frac{U_3}{U_1} - \frac{1}{2}\frac{U_2^2}{U_1^2}\Big) = \frac{2}{3M_{ref}}\Big(\nabla U_1(\frac{U_2^2}{U_1^3} - \frac{U_3}{U_1^2}) + \nabla U_2(-\frac{U_2}{U_1^2}) + \nabla U_3(\frac{1}{U_1})\Big),$$

and using the following definition:

$$\boldsymbol{V_i}(U) = \left\{\begin{array}{c} \frac{U_2^2}{U_1^3} - \frac{U_3}{U_1^2} \\ -\frac{U_2}{U_1^2} \\ \frac{1}{U_1} \\ 0 \end{array}\right\}$$

it can be simplified as:

$$\nabla T_i = \frac{2}{3M_{ref}}\mathbf{Q}_t\mathbf{V}_i(\mathbf{U}), \tag{A3}$$

where the transpose of the variable gradient has been introduced $\mathbf{Q}_t = \mathbf{Q}^T$. For the electron, the gradient writes as:

$$\nabla T_e = \frac{2}{3M_{ref}}\nabla\Big(\frac{U_4}{U_1}\Big) = \frac{2}{3M_{ref}}\Big(\nabla U_1(-\frac{U_4}{U_1^2}) + \nabla U_4(\frac{1}{U_1})\Big),$$

and can be simplified using the following definition:

$$\boldsymbol{V_e}(U) = \left\{\begin{array}{c} -\frac{U_4}{U_1^2} \\ 0 \\ 0 \\ \frac{1}{U_1} \end{array}\right\}$$

as:

$$\nabla T_e = \frac{2}{3M_{ref}}\mathbf{Q}_t\mathbf{V}_e(\mathbf{U}). \tag{A4}$$

Hence, using the definition of the parallel gradient, we have

$$\begin{aligned} \nabla_\parallel T_i &= \nabla T_i \cdot \mathbf{b} = \frac{2}{3M_{ref}}\mathbf{Q}_t\mathbf{V}_i(\mathbf{U}), \\ \nabla_\parallel T_e &= \nabla T_e \cdot \mathbf{b} = \frac{2}{3M_{ref}}\mathbf{Q}_t\mathbf{V}_e(\mathbf{U}), \end{aligned} \tag{A5}$$

From the expressions of these parallel gradients we derive the energy flux $\mathbf{F}_t$ related to the parallel diffusion of the temperature as:

$$\mathbf{F}_t = \left\{\begin{array}{c} 0 \\ 0 \\ k_{\parallel,i}T_i^{5/2}\nabla_\parallel T_i \\ k_{\parallel,e}T_e^{5/2}\nabla_\parallel T_e \end{array}\right\} \otimes \mathbf{b^T} = \left\{\begin{array}{c} 0 \\ 0 \\ k_{||i}\Big(\frac{2}{3M_{ref}}\Big)^{7/2}\left(\frac{U_3}{U_1 - \frac{1}{2}\frac{U_2^2}{U_1}^2}\right)^{5/2}\boldsymbol{Q^T V_i}(U)\cdot\boldsymbol{b} \\ k_{||e}\Big(\frac{2}{3M_{ref}}\Big)^{7/2}\Big(\frac{U_4}{U_1}\Big)^{5/2}\boldsymbol{Q^T V_e}(U)\cdot\boldsymbol{b} \end{array}\right\} \otimes \boldsymbol{b^T}. \tag{A6}$$

The vector related to the contribution of the parallel electric field $\mathbf{f}_{E_\parallel}$ is

$$\mathbf{f}_{E_\parallel} = M_{ref} u \nabla_\parallel p_e \begin{Bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{Bmatrix} = \frac{2}{3}\frac{U_2}{U_1}\nabla U_4 \cdot \mathbf{b} \begin{Bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{Bmatrix} = \frac{2}{3}\mathbf{Q}_t \mathbf{W}(\mathbf{U}) \cdot \mathbf{b} \begin{Bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{Bmatrix} \tag{A7}$$

having defined the vector

$$\mathbf{W}(\mathbf{U}) = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \frac{U_2}{U_1} \end{Bmatrix}.$$

The vector of temperature exchange between ions and electrons $\mathbf{f}_{EX}$ is

$$\mathbf{f}_{EX} = \frac{n^2}{\tau_{ie}}\frac{T_e - T_i}{T_{3/2}} \begin{Bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{Bmatrix} = \frac{1}{\tau_{ie}} \left(\frac{2}{3M_{ref}}\right)^{-1/2} \frac{U_1^{5/2}}{U_4^{3/2}} \left(U_3 - U_4 + \frac{1}{2}\frac{U_2^2}{U_1}\right) \begin{Bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{Bmatrix}. \tag{A8}$$

Finally the curvature term $\mathbf{g}$ is

$$\mathbf{g} = \begin{Bmatrix} 0 \\ (p_i + p_e)\nabla \cdot \mathbf{b} \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ \frac{2}{3}\left(U_3 + U_4 - \frac{1}{2}\frac{U_2^2}{U_1}\right)\nabla \cdot \mathbf{b} \\ 0 \\ 0 \end{Bmatrix}. \tag{A9}$$

and

$$\boldsymbol{s} = \{S_n, S_\Gamma, S_{E_i}, S_{E_e}\}^T \tag{A10}$$

is the vector of source terms. When the latter are chosen with an analytical form they constitute the right hand side RHS of the conservative system of equations A2. Otherwise, if they depend by the plasma quantities, they are made explicit function of the conservative variables $\mathbf{U}$ and treated in the same manner of the vectors above.

Appendix B. The HDG solver

The resolution of Eqs. A2 is made through two steps.

Appendix B.1. The local problem

The local problem coincides with the system A2 presented above and solved in each element Ω_i. A Dirichlet condition is imposed in each element boundary $\partial\Omega_i$, which constrains $\boldsymbol{U}$ to be equal to $\hat{\boldsymbol{U}}(\boldsymbol{x}, t)$ for $\boldsymbol{x} \in \mathcal{T}$. The local problem now consists into determine $\mathbf{Q}$ and $\mathbf{U}$ in function of the imposed values $\hat{\mathbf{U}}(\mathbf{x}, t)$ on the mesh skeleton $\mathcal{T}$. Thus, for $i = 1, ..., N_{el}$ the local system of equation to solve in the HDG formulation can be written as follows:

$$\begin{cases} \boldsymbol{Q} - \boldsymbol{\nabla U} = 0 & in\ \Omega_i \times]0, T_f[\\ \partial_t \boldsymbol{U} + \boldsymbol{\nabla} \cdot (\boldsymbol{F} - D_f \boldsymbol{Q} + D_f \boldsymbol{Q} \boldsymbol{b} \otimes \boldsymbol{b} - \boldsymbol{F_t}) + \\ \quad + \boldsymbol{f}_{E_{||}} + \boldsymbol{f}_{EX} - \boldsymbol{g} = \boldsymbol{s} & in\ \Omega_i \times]0, T_f[\\ \mathbf{U}(\mathbf{x}, t) = \hat{\mathbf{U}}(\mathbf{x}, t) & in\ \partial\Omega_i \times]0, T_f[\\ \mathbf{U}(\mathbf{x}, 0) = \mathbf{U_0} & in\ \Omega_i \end{cases} \tag{A11}$$

The continuity of the unknowns is guaranteed due to the fact that the Dirichlet condition imposed on the left and on the right element of a given face is the same, for the given values of $\hat{\boldsymbol{U}}$ on the element boundary. The approximated solution is then obtained after the discretization of the system of equation A11 on a finite two-dimensional space defined in this way:

$$\mathcal{V}_h = \{v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^p(\Omega_i) \; for \; i = 1, ..., N_{el}\}$$
$$\Lambda_h = \{\hat{v} \in \mathcal{L}^2(\mathcal{T}) : \hat{v}|_{\Gamma_i} \in \mathcal{P}^p(\Gamma_i) \; for \; i = 1, ..., N_f\},$$

where Γ_i is one face of the element border and $\mathcal{P}^p$ is the space of the polynomials of degree less or equal to p. Therefore, $\mathcal{V}_h$ defines the space of the set of functions for the discretization of the internal part of the elements while Λ_h determines the one related to the trace unknowns on the elements border. So, the arbitrary precision of the numerical scheme is ruled by the degree of the interpolant polynomials. The Fekete nodal distribution is used as standard nodal basis to avoid ill conditioning issues. In Figure A1 is represented the node distribution in the space $\mathcal{V}_h$ and Λ_h for a triangular element with polynomial degree $p = 5$.

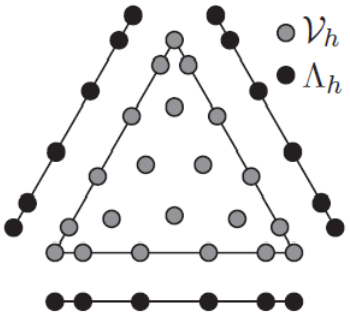

Figure A1. Nodal representation in the space $\mathcal{V}_h$ and Λ_h for a triangle element of $p = 5$. From [10].

In order to derive the weak formulation of the system A11 we use the same procedure explained in [1] obtaining:

$$\begin{cases} (\boldsymbol{\mathcal{G}}, \boldsymbol{Q})_{\Omega_i} + (\boldsymbol{\nabla}\boldsymbol{\mathcal{G}}, \boldsymbol{U})_{\Omega_i} - \langle \boldsymbol{\mathcal{G}}\boldsymbol{n}, \hat{\boldsymbol{U}} \rangle_{\partial\Omega_i} = 0 \\ (\boldsymbol{v}, \partial_t \boldsymbol{U})_{\Omega_i} - \left(\boldsymbol{\nabla}\boldsymbol{v}, \boldsymbol{F} - D_f \boldsymbol{Q} + D_f \boldsymbol{Q}\boldsymbol{b} \otimes \boldsymbol{b} - \boldsymbol{F}_t\right)_{\Omega_i} \\ + \langle \boldsymbol{v}, \left(\hat{\boldsymbol{F}} - D_f \hat{\boldsymbol{Q}} + D_f \hat{\boldsymbol{Q}}\boldsymbol{b} \otimes \boldsymbol{b} - \hat{\boldsymbol{F}}_t\right)\boldsymbol{n} \rangle_{\partial\Omega_i} \\ + \left(\boldsymbol{v}, \boldsymbol{f}_{E_{||}}\right)_{\Omega_i} + \left(\boldsymbol{v}, \boldsymbol{f}_{E_{EX}}\right)_{\Omega_i} - (\boldsymbol{v}, \boldsymbol{g})_{\Omega_i} = (\boldsymbol{v}, \boldsymbol{s})_{\Omega_i} \end{cases} \tag{A12}$$

The local problem results ends up in the search for an approximation $(\mathbf{Q}, \mathbf{U}) \in [\mathcal{V}_h]^{d\times d} \times [\mathcal{V}_h]^d$, with a given $\hat{\mathbf{U}} \in [\Lambda_h]^d$, for all $(\mathcal{G}, \mathbf{U}) \in [\mathcal{V}_h]^{4\times 2} \times [\mathcal{V}_h]^4$ that satisfies the system of equations A12 for $i = 1, ..., N_{el}$. In A12, $(.,.)_{\Omega_i}$ denotes the $\mathcal{L}^2$ scalar product in the element Ω_i, while $\langle .,. \rangle$ stands for the scalar product of the traces in $\partial\Omega_i$. Eventually, the traces of $\boldsymbol{F}$ and $\boldsymbol{Q}$ on the element boundary have been replaced by numerical traces in this way:

$$\begin{aligned} \hat{\boldsymbol{F}}(\hat{U}) &= \boldsymbol{F}(\hat{U}) + \boldsymbol{\tau}(\boldsymbol{U} - \hat{\boldsymbol{U}}) \otimes \boldsymbol{n} \\ \hat{\boldsymbol{Q}} &= \boldsymbol{Q} \\ \hat{\boldsymbol{F}}_t(\hat{U}) &= \boldsymbol{F}_t(\hat{U}) \end{aligned} \tag{A13}$$

where $\boldsymbol{n}$ is the outer normal to the element face and $\boldsymbol{\tau}$ is the local stabilization matrix. It is important to underline that $\boldsymbol{\tau}$ plays a fundamental role on both the stability and the accuracy of the numerical scheme, and in the literature its role has been already investigated for a large number of problems by Cockburn et al. [15]. In this work, we

consider its expression in a diagonal form: $\boldsymbol{\tau} = \tau \boldsymbol{I}$, with $\boldsymbol{I}$ the identity matrix, and depends by the parameters of the simulation (perpendicular and parallel diffusion coefficients, sound speed, size of mesh elements etc.).

Appendix B.2. The global problem

The system A12 allows to compute the solution $\boldsymbol{U}$ and $\boldsymbol{Q}$ in the whole domain of computation in function of the trace of the unknowns on the element border $\hat{\boldsymbol{U}}$. By setting up the global problem it is possible to determine this variable, which allows to solve for $\hat{\boldsymbol{U}}$ in the entire mesh skeleton. Imposing the continuity of the fluxes across the element border we can obtain the equation for $\hat{\boldsymbol{U}}$, which, in weak form, it determines the global problem. Substituting the definition of the fluxes, it can be written as follows:

$$\left\langle \hat{\boldsymbol{v}}, \Big(\boldsymbol{F} - D_f \boldsymbol{Q} + D_f \boldsymbol{Q}\boldsymbol{b} \otimes \boldsymbol{b} - \boldsymbol{F}_t \Big)\boldsymbol{n} + \boldsymbol{\tau}(\boldsymbol{U} - \hat{\boldsymbol{U}}) \right\rangle_{\mathcal{T}\setminus\partial\Omega} + \left\langle \hat{\boldsymbol{v}}, \boldsymbol{B}_{BC} \right\rangle_{\partial\Omega} = 0 \qquad \text{(A14)}$$

where $\boldsymbol{\mathcal{T}}$ represents the skeleton of the triangulation, and $\boldsymbol{B}_{BC}$ is a flux vector that defines the boundary condition on $\partial\Omega$. Thus, the global problem becomes the search of an approximation $\hat{\mathbf{U}} \in [\Lambda_h]^4$ for the system A14, for all $\hat{\mathbf{v}} \in [\Lambda_h]^4$. Here $\boldsymbol{U}$ and $\boldsymbol{Q}$ are the solutions of the local problem A12 in function of $\hat{\boldsymbol{U}}$. Eventually, the system A14 weakly imposes the normal fluxes at the element boundary and it depends only by the unknown $\hat{\boldsymbol{U}}$, reducing the size of the linear system generated by the element discretization.

Appendix B.3. Discrete form of the weak equations

In the previous appendix sections we have introduced all the necessary ingredients to build up the discrete form of the weak problem A12 that is worthwhile and complementary in order to explain the results showed in section 4. Thus, just by assembling everything together it is possible to obtain the final form of linear system to be solved. In the code is used a totally implicit approach, so the time derivative is discretized with a scheme of the form:

$$\partial_t \mathbf{U} \approx \delta \frac{\mathbf{U}}{\Delta t} - \mathbf{f_0}$$

where δ is a constant parameter that depends of the time integration scheme, and $\mathbf{f_0}$ is a vector that takes into account the previous time steps. Now we need to use a linearization technique exploited also for the non-linear terms inside the model. Considering a set of variables $\{w_1, w_2, ...\}$ these non-linear terms have been solved using a Newton-Raphson iterative procedure. In a Newton-Raphson framework, the bilinear forms are linearized using a second-order approximation. The linearization used for a generic term $\mathbf{f}$ is the following:

$$\begin{aligned} \mathbf{f}(w_1^k, w_2^k, ...) = \mathbf{f}(w_1^{k-1}, w_2^{k-1}, ...) + \frac{d}{d\epsilon}\mathbf{f}(w_1^{k-1} + \epsilon dw_1, w_2^{k-1} + \epsilon dw_2, ...)|_{\epsilon=0} \\ + \mathcal{O}(dw_1^2, dw_2^2, ...), \end{aligned} \qquad \text{(A15)}$$

where k is the NR iteration and $dw_i = w_i^k - w_i^{k-1}$. Now, proceeding with our problem, substituting the definition of the numerical traces introduced in equation A13 and

rearranging the terms with reference to the three variables of the local problem $\mathbf{U}$, $\mathbf{Q}$, $\hat{\mathbf{U}}$, the resulting weak problem can be written:

$$\begin{cases} \left(\nabla\mathbf{v}, D_f\mathbf{Q} - D_f\mathbf{b}\otimes\mathbf{Q}\mathbf{b} + \mathbf{F}_{\mathbf{t}}^{\mathbf{Q}}\right)_{\Omega_i} + \left\langle \mathbf{v}, (-D_f\mathbf{Q} + D_f\mathbf{b}\otimes\mathbf{Q}\mathbf{b} - \mathbf{F}_{\mathbf{t}}^{\mathbf{Q}})\cdot n\right\rangle_{\partial\Omega_i} + \\ + \left(\mathbf{v}, \mathbf{f}^{\mathbf{Q}}{}_{E_\parallel}\right)_{\Omega_i} + \left(\mathbf{v}, \frac{\delta}{\Delta t}\mathbf{U}\right)_{\Omega_i} - \left(\nabla\mathbf{v}, \mathcal{A}_{(\mathbf{U})}{}^{k-1}\mathbf{U} - \mathbf{F}_{\mathbf{t}}^{\mathbf{U}}\right)_{\Omega_i} + \left\langle \mathbf{v}, \boldsymbol{\text{ø}}\mathbf{U}\right\rangle_{\partial\Omega_i} + \\ + \left(\mathbf{v}, \mathbf{f}^{\mathbf{Q}}{}_{E_\parallel} + \mathbf{f}^{\mathbf{U}}{}_{E_\parallel}\right)_{\Omega_i} + \left(\mathbf{v}, \mathbf{f}^{\mathbf{U}}{}_{EX}\right)_{\Omega_i} - \left(\mathbf{v}, \frac{\mathbf{dg}}{\mathbf{dU}}\Big|_{k-1}\mathbf{U}\right)_{\Omega_i} + \\ + \left\langle \mathbf{v}, (\mathcal{A}_{\mathbf{U}}^{k-1}\hat{\mathbf{U}} - \mathbf{F}_{\mathbf{t}}^{\hat{\mathbf{U}}})\cdot\mathbf{n}\right\rangle_{\partial\Omega_i} - \left\langle \mathbf{v}, \boldsymbol{\text{ø}}\hat{\mathbf{U}}\right\rangle_{\partial\Omega_i} = \left(\mathbf{v}, \mathbf{f_0}\right)_{\Omega_i} + \left(\mathbf{v}, \mathbf{s}\right)_{\Omega_i} - \\ - \left(\nabla\mathbf{v}, \mathbf{F}_{\mathbf{t}}^{\mathbf{0}}\right)_{\Omega_i} + \left\langle \mathbf{v}, \mathbf{F}_{\mathbf{t}}^{\mathbf{0}}\cdot\mathbf{n}\right\rangle_{\partial\Omega_i} - \left(\mathbf{v}, \mathbf{f}^{\mathbf{0}}{}_{EX}\right)_{\Omega_i} \\ \left(\mathcal{G}, \mathbf{Q}\right)_{\Omega_i} + \left(\nabla\cdot\mathcal{G}, \mathbf{U}\right)_{\Omega_i} - \left\langle \mathcal{G}\mathbf{n}, \hat{\mathbf{U}}\right\rangle_{\partial\Omega_i} = 0. \end{cases} \tag{A16}$$

for each element $i = 1, ..., N_{el}$. In order to develop a high-order finite-element scheme, an high-order polynomial interpolation is considered in each element to represent the unknowns. Defining a set of basis functions, the vector of nodal values for the vector unknown $\mathbf{U}$, $\hat{\mathbf{U}}$ and similarly for the tensor unknown $\mathbf{Q}$ in the element Ω_i can be represented as:

$$\mathbf{U} = \sum_{j=1}^{N_p} N_j \mathbf{I_4} \mathcal{U}^j \qquad \mathbf{Q} = \sum_{j=1}^{N_p} N_j \mathbf{I_8} \mathcal{Q}^j \qquad \hat{\mathbf{U}} = \sum_{j=1}^{N_f p} \hat{N}_j \mathbf{I_4} \hat{\mathcal{U}}^j \tag{A17}$$

where N_p is the number of nodes in each element and N_j, $\hat{N}_j$ is the j-th basis belonging to $\mathcal{V}_h$ and Λ_h respectively, and $\mathcal{U}^j$, $\mathcal{Q}^j$, $\hat{\mathcal{U}}^j$ are the nodal value of the unknowns $\mathbf{U}, \mathbf{Q}, \hat{\mathbf{U}}$ in the j-th node. The test functions are chosen in the same space of the basis functions, so we can define $\mathbf{v}, \mathcal{G}$ and $\hat{\mathbf{v}}$ as follows:

$$\mathbf{v} = \sum_{j=1}^{N_p} N_j \mathbf{I_4} \mathbf{v} \qquad \mathcal{G} = \sum_{j=1}^{N_p} N_j \mathbf{I_8} \mathcal{G} \qquad \hat{\mathbf{v}} = \sum_{j=1}^{N_f p} \hat{N}_j \mathbf{I_4} \hat{\mathbf{v}} \tag{A18}$$

where the vector $\mathbf{v}$ is the correspondent column of the identity matrix for each equation, respectively. The vectors $\mathcal{G}$ and $\hat{\mathbf{v}}$ are constructed in a similar way. Using the nodal decomposition introduced in A17-A18, the system of equations for the local problem A16 can be rewritten:

$$\begin{aligned} A_{uq}\overline{\mathcal{Q}} + A_{uu}\overline{\mathcal{U}} + A_{ul}\overline{\hat{\mathcal{U}}} &= \mathbf{S} \\ A_{qq}\overline{\mathcal{Q}} + A_{qu}\overline{\mathcal{U}} + A_{ql}\overline{\hat{\mathcal{U}}} &= 0 \end{aligned} \tag{A19}$$

where we define the vectors $\overline{\mathcal{U}} = [\mathcal{U}_1, ..., \mathcal{U}_{N_p}]$, $\overline{\mathcal{Q}} = [\mathcal{Q}_1, ..., \mathcal{Q}_{2N_p}]$ and $\overline{\hat{\mathcal{U}}} = [\hat{\mathcal{U}}_1, ..., \hat{\mathcal{U}}_{N_{fp}}]$ and the following bilinear form is introduced:

$$
\begin{aligned}
A_{uq} &= \Big(\nabla \mathbf{v}, D_f \mathbf{Q}\Big)_{\Omega_i} - \Big\langle \mathbf{v}, D_f \mathbf{Q}\Big\rangle_{\partial\Omega_i} - \Big(\mathbf{v}, D_f \mathbf{Q}\mathbf{b} \otimes \mathbf{b}\Big)_{\Omega_i} + \\
&\quad + \Big\langle \nabla \mathbf{v}, D_f \mathbf{Q}\mathbf{b} \otimes \mathbf{b}\Big\rangle_{\partial\Omega_i} + \Big(\nabla \mathbf{v}, \mathbf{F}^{\mathbf{Q}}_{\mathbf{t}}\Big)_{\Omega_i} - \Big\langle \mathbf{v}, \mathbf{F}^{\mathbf{Q}}_{\mathbf{t}} \cdot \mathbf{n}\Big\rangle_{\partial\Omega_i} + \Big(\mathbf{v}, \mathbf{f}^{\mathbf{Q}}{}_{E_\parallel}\Big)_{\Omega_i}, \\
A_{uu} &= \Big(\mathbf{v}, \frac{\delta}{\Delta t}\mathbf{U}\Big)_{\Omega_i} + \Big\langle \mathbf{v}, \boldsymbol{\text{ø}}\mathbf{U}\Big\rangle_{\partial\Omega_i} - \Big(\nabla \mathbf{v}, \mathcal{A}^{k-1}\mathbf{U}\Big)_{\Omega_i} + \Big(\nabla \mathbf{v}, \mathbf{F}^{\mathbf{U}}_{\mathbf{t}}\Big)_{\Omega_i} + \\
&\quad + \Big(\mathbf{v}, \mathbf{f}^{\mathbf{U}}{}_{E_\parallel}\Big)_{\Omega_i} + \Big(\mathbf{v}, \mathbf{f}^{\mathbf{U}}{}_{EX}\Big)_{\Omega_i} - \Big(\mathbf{v}, \frac{\mathbf{dg}}{\mathbf{dU}}\Big|_{k-1}\mathbf{U}\Big)_{\Omega_i}, \\
A_{ul} &= \Big\langle \mathbf{v}, (\mathcal{A}^{k-1}\hat{\mathbf{U}}) \cdot \mathbf{n}\Big\rangle_{\partial\Omega_i} - \Big\langle \mathbf{v}, \mathbf{F}^{\hat{\mathbf{U}}}_{\mathbf{t}} \cdot \mathbf{n}\Big\rangle_{\partial\Omega_i} - \Big\langle \mathbf{v}, \boldsymbol{\text{ø}}\hat{\mathbf{U}}\Big\rangle_{\partial\Omega_i}, \\
\mathbf{S} &= \Big(\mathbf{v}, \mathbf{f_0}\Big)_{\Omega_i} + \Big(\mathbf{v}, \mathbf{s}\Big)_{\Omega_i} - \Big(\nabla \mathbf{v}, \mathbf{F}^{\mathbf{0}}_{\mathbf{t}}\Big)_{\Omega_i} + \Big\langle \mathbf{v}, \mathbf{F}^{\mathbf{0}}_{\mathbf{t}} \cdot \mathbf{b}\Big\rangle_{\partial\Omega_i} - \Big\langle \mathbf{v}, \mathbf{f}^{\mathbf{0}}_{\mathbf{EX}} \cdot \mathbf{b}\Big\rangle_{\partial\Omega_i}, \\
A_{qq} &= \Big(\mathcal{G}, \mathbf{Q}\Big)_{\Omega_i}, \qquad A_{qu} = \Big(\nabla \cdot \mathcal{G}, \mathbf{U}\Big)_{\Omega_i}, \qquad A_{qu} = \Big\langle \mathcal{G}\mathbf{n}, \hat{\mathbf{U}}\Big\rangle_{\partial\Omega_i}.
\end{aligned}
\tag{A20}
$$

The problem in A19 coincide with solving $N_p + 2N_p$ equations, so, clearly, it is not sufficient to compute the $(N_p + 2N_p + N_{ef} \times N_{fp})$ coefficients $\mathcal{U}, \mathcal{Q}, \hat{\mathcal{U}}$ where N_{ef} is the number of faces in each element. Nevertheless it is possible to find a relation between them using the Newton-Raphson procedure for the computation of the residuals. Thus, for each iteration k this procedure allow us to solve the local linear system A19 for the variable $\mathbf{U}$ and $\mathbf{Q}$ in function of the variable $\hat{\mathbf{U}}$ on the faces of the element. Writing in a more compact form for each element $i = 1, ..., N_{el}$ we have:

$$
\begin{aligned}
\mathcal{U}_i^{n,k} &= \mathbf{U}_i^{k,n}\hat{\mathcal{U}}_i^{n,k} + \mathcal{F}_i^{n,k}, \\
\mathcal{Q}_i^{n,k} &= \mathbf{Q}_i^{k,n}\hat{\mathcal{U}}_i^{n,k} + \mathcal{H}_i^{n,k}
\end{aligned}
\tag{A21}
$$

where $\mathcal{U}_i^{n,k}$, $\mathcal{Q}_i^{n,k}$, $\hat{\mathcal{U}}_i^{n,k}$ are respectively the nodal solutions of the unknown $\mathbf{U}$, $\mathbf{Q}$ for the element Ω_i and the nodal solution of the trace $\hat{\mathbf{U}}$ for the faces of the element $\partial\Omega_i$, at the time step n and NR iteration k. The terms $\mathbf{U}_i^{k,n}$ and $\mathbf{Q}_i^{k,n}$ are the elemental matrices at the time step n and NR iteration k, while $\mathcal{F}_i^{n,k}$ and $\mathcal{H}_i^{n,k}$ are the right-hand side vectors for the two systems. At this point the nodal values $\mathcal{U}$, $\mathcal{Q}$ can be replaced by the solution of the local problem A21 and it is possible to write a set of equations involving only the nodal values $\hat{\mathcal{U}}$ in the whole mesh:

$$
\mathcal{K}^{k,n}\hat{\mathcal{U}}^{k,n} = \mathcal{R}^{k,n},
\tag{A22}
$$

where $\mathcal{K}^{k,n}$ is the global matrix and $\mathcal{R}_i^{k,n}$ is the global right hand side at each iteration of the Newton-Raphson method used and at each time step. It is straightforward that the inversion of the problem A22 represents the solution of the HDG problem.

References

1. G. Giorgiani et al., An hybrid discontinuous galerkin method for tokamak edge plasma simulations in global realistic geometry, *J. Comp. Phys.* **2018**.
2. R. Simoni et al., Models and numerics in the multi-fluid 2-d edge plasma code edge2d/u, *Contrib. Plasma Phys.*,34(2- 3):368–373, **1994**.
3. S. Wiesen et al., The new solps-iter code package, *Journal of Nuclear Materials*, 463:480–484, **2015**.
4. H. Bufferand et al., Numerical modelling for divertor design of the WEST device with a focus on plasma–wall interactions, *Nuclear Fusion*, 55(5):053025, **2015**.
5. C.Dobrzynski et al., Anisotropic Delaunay mesh adaptation for unsteady simulations, *Proceedings of the 17th international Meshing Roundtable*, **2008**.

6. C. Dapogny et al., Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, *JCP*, 262, pp. 358–378 **2014**.
7. K.J. Fidkowski, Comparison of hybrid and standard discontinuous Galerkin methods in a mesh-optimisation setting, *International Journal of Computational Fluid Dynamics*, **2019**.
8. S.I. Braginskii, Transport processes in a plasma, *Reviews of Plasma Physics*, **1965**.
9. W.H. Reed, Triangular mesh methods for the neutron transport equation, *Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory*, **1973**.
10. G. Giorgiani et al., A high-order non field-aligned approach for the discretization of strongly anisotropic diffusion operators in magnetic fusion, *Computer Physics Communications*, **2020**.
11. G. Giorgiani et al., A new high-order fluid solver for tokamak edge plasma transport simulations based on a magnetic-field independent discretization, *CPP*, **2018**.
12. G. Giorgiani et al., A magnetic-field independent approach for strongly anisotropic equations arising plasma-edge transport simulations, *Nuclear Materials and Energy*, **2019**.
13. G. Giorgiani et al., Hybridizable Discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations, *Computers & Fluids*, **2014**.
14. J. Peraire et al., A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations, *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, **2010**.
15. N.C. Nguyen et al., An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations, *JCP*, **2009**.
16. P.M. Knupp, Algebraic mesh quality metrics for unstructured initial meshes, *Finite Elements in Analysis and Design*, 39, 217-241 **2003**.
17. M.S. D'Abusco et al., Core-edge 2D fluid modeling of full tokamak discharge with varying magnetic equilibrium: from WEST start-up to ramp-down, *NF*, **2022**.
18. G. Piraccini et al, Spatial adaptivity in SOLEDGE3X-HDG for edge plasma simulations in versatile magnetic and reactor geometries, *CPP*, **2022**.
19. J.A. Soler et al., A new conservative finite-difference scheme for anisotropic elliptic problems in bounded domain, *Journal of Computational Physics*, 405:109093, **2020**.
20. A. Loarte et al., Chapter 4: Power and particle control, *Nuclear Fusion*, 47 S203–S263, **2007**.
21. P.C. Stangeby, The Plasma Boundary of Magnetic Fusion Devices, *CRC Press*, **2000**.
22. J.T. Oden, Toward a universal h-p adaptive finite element strategy, Part 2. A posteriori error estimation , *Computer methods in applied mechanics and engineering*, **1989**.
23. P. Stangeby, The plasma boundary of magnetic fusion devices, Eds. CRC Press, **2000**.
24. I. Babuška et al., Approximation properties of the h-p version of the finite element method, *Computer Methods in Applied Mechanics and Engineering* 133:3, 319-346, **1996**.
25. P. Ghendrih et al., Transition to supersonic flows in the edge plasma , *Plasma Phys. Control. Fusion*, 53(5) **2011**.
26. H. Bufferand et al., Study of transition to supersonic regime in divertor SOL using SOLEDGE2D code, *Plasma Phys. Control. Fusion*, 54(4-6) **2014**.
27. Horton et al., Characterization of the H-mode edge barrier at ASDEX Upgrade., *Nuclear fusion 45*, no. 8: 856. **2005**.