Towards a hybrid kinetic-fluid description for neutral particles for edge plasma codes

M. Valentinuzzi[1], G. Giorgiani[1], Y. Marandet[2], H. Bufferand[1], G. Ciraolo[1], P. Tamain[1], E. Serre[3], Ph. Ghendrih[1]

[1] *CEA, IRFM, F-13108 Saint-Paul-Lez-Durance, France*

[2] *Aix Marseille Univ, CNRS, Laboratoire PIIM, Marseille, France*

[3] *Aix Marseille Univ, CNRS, Ecole Centrale, Laboratoire M2P2, Marseille, France*

1 Introduction

Power exhaust is one of the major challenges of future devices such as ITER and DEMO. Because of the lack of identified scaling parameters, predictions for divertor plasma conditions in these devices have to rely on edge transport codes, which consist of a fluid plasma code (like Soledge2D [1]) coupled to a kinetic Monte Carlo code (such as Eirene [2]) for the neutral particles, incorporating the complex atomic, molecular and surface processes characteristic of edge plasmas.

However, in the divertor region close to the target plates, owing to the high density ($\sim 10^{20} \div 10^{21} m^3$) and low temperatures (below $\sim 5eV$), the kinetic description can be too detailed and, due to the high number of charge-exchange collisions before ionization, the codes tend to be very inefficient. Furthermore, the use of a Monte Carlo procedure may introduce statistical noise in the plasma solution [3].

A hybrid model then becomes appealing for the neutral gas, treating the latter as a collisional fluid in low-temperature areas and kinetically elsewhere.

In order to achieve this, a new fluid neutrals code has been developed for Soledge2D. The physical model for the code is described in section 2, and the numerical implementation through the novel Hybridizable Discontinuous Galerkin method is introduced in section 3; finally, results in simple slab and real geometry are shown in, respectively, sections 4 and 5.

2 Physical model

The code solves the system of fluid equations shown in equations 1 and 2, in which a closure approximation has been introduced in the momentum equation by taking as distribution function for the neutrals a simple Maxwellian drifting with fluid velocity $\vec{u}_n = \frac{1}{n_n} \iiint f_n \vec{v} d\vec{v}$ and with temperature $T_n = T_i$.

$$\frac{\partial n_n}{\partial t} + \vec{\nabla} \cdot (n_n \vec{u}_n) = -n_e n_n \langle \sigma v \rangle_{iz} + n_e n_i \langle \sigma v \rangle_{rc} \quad (1)$$

$$\frac{\partial n_n \vec{u}_n}{\partial t} + \vec{\nabla} \cdot \left(n_n \vec{u}_n \otimes \vec{u}_n + \frac{1}{m_n} n_n T_n \mathbb{I} \right) = -n_e n_n \vec{u}_n \langle \sigma v \rangle_{iz} + n_e n_i \vec{u}_i \langle \sigma v \rangle_{rc} - (n_i n_n \vec{u}_n - n_n n_i \vec{u}_i) \langle \sigma v \rangle_{cx} \quad (2)$$

$n_{species}$ is the density of the species considered (ions, electrons or neutrals), $u_{species}$ is its fluid velocity and $\langle \sigma v \rangle_{reaction}$ is the effective reaction rate for the considered reaction.

A set of boundary conditions consistent with the kinetic description is imposed on the boundary surfaces [4]: all the neutrals that go in the core will be ionized, so a zero neutral flux in the normal direction (from the core to the SOL) is imposed at the core boundary; at the solid walls, a fraction R_i of the ion flux is recycled as neutrals and a fraction R_n of the neutral flux is reflected, and the total flux $\Gamma_n^+ = -R_i \Gamma_i + (1 - R_n) \Gamma_n^-$ is then imposed on the wall.

3 Numerical implementation

The code has been written using a quite novel numerical method called Hybridizable Discontinuous Galerkin (HDG). First introduced in [5], the HDG method has been already used to solve diffusion, continuum mechanics and fluid-dynamics problems [6].

The basic idea of the method is to re-write the discretized problem in function of a new unknown, defined only on the boundaries of the mesh elements. By doing this, the number of degrees of freedom of the global problem can be reduced, lowering the computational cost of the simulations. This reduction is more marked the higher the number of internal points of the mesh elements, thus making it appealing to use higher order discretizations. Finally, the original unknowns can be recovered locally (i.e. element-by-element) as functions of the boundary unknowns, making it also easy to parallelize the code.

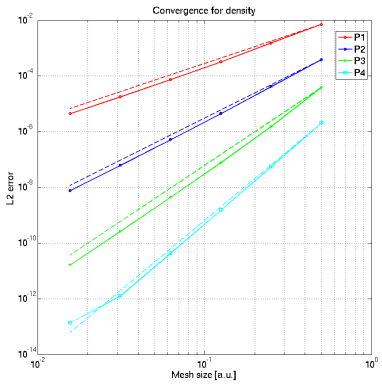

Figure 1: *L2 norm of the error on the density as function of the grid size. Dashed lines show the theoretical order of convergence, while solid lines show the error computed by the code. The different colors correspond to the order of the interpolating polynomials used.*

The code implementation has been tested with the Method of Manufactured Solutions (MMS). The method consists in imposing analytic functions $\tilde{U}$ (f.e. trigonometric functions) as solution of the governing equations, and obtaining an analytic source term $\tilde{S} = \frac{\partial \tilde{U}}{\partial t} + \vec{\nabla} \cdot \vec{F}$; this source term is then forced in the code, and the analytic function is imposed as initial condition and as boundary condition. The code run in this setup will introduce an error on the solution that, if the implementation is correct, should only depend on the grid resolution, and should go down with the latter with a slope (that is the asymptotic order of convergence) characteristic of the numerical method. For the HDG method, the asymptotic order of convergence is equal to the order of the interpolating polynomials plus 1.

It can be seen from fig. 1 that the code recovers the theoretical order of convergence of the HDG method, thus giving us confidence about the correct implementation of the code.

4 Results: slab geometry

As a first test, the code was run in slab geometry with a plasma background created analytically.

In fig. 2 are shown the plasma density and the electron temperature. The x direction mimics the parallel direction, with the $x = 0$ and $x = 1$ boundaries taken as either divertor targets or limiter walls; the y direction, on the other hand, is assumed to be the radial one, with the boundary at $y = 0$ being the first wall and the one at $y = 1$ taken as an ideal interface with the core. It can be seen that the background resembles to be in a sheath limited regime, with (electron and ion) temperature constant along the field lines and an upstream ($x = 0.5$) density $n_{up} = 2n_{target}$.

In fig. 3 the mass balance in the domain is shown during the simulation: the green line shows the total recycling flux coming from the walls, in cyan the total source/sink due to interactions with the plasma, and in red the total flux going towards the core. As it can be seen, a steady state is found once the three contributions cancel out. At the same time, the particle content decreases from the initial value to a constant one of about $3.5 \cdot 10^{17}$ particles consistently with the decrease, in absolute value, of the ionization source.

The spatial distribution of the density and the velocity in the domain, after steady state was reached, can be seen in fig. 4. As one would expect the density peaks at the targets due to the recycling source, while the ionization sink in the rest of the domain prevailed against the recycled influx. As both the ion flux on the wall boundaries and the neutrals temperature (that was assumed to be equal to the ions') were not homogeneous in the vertical direction, the gradient of pressure in the vertical direction creates a downward flux; a similar pressure gradient but in the opposite direction, caused by the presence of the core interface, creates an upward flux in the near the middle of the domain.

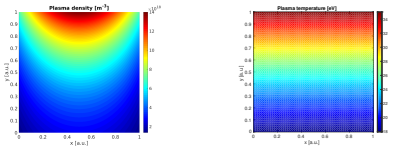

Figure 2: *Plasma density* $[m^{-3}]$ *(left) and electron temperature* $[eV]$ *(right) taken as plasma background.*

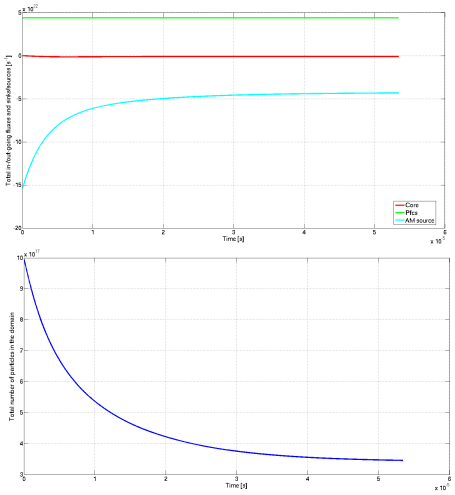

Figure 3: *Mass balance (top) and particles content (bottom)*

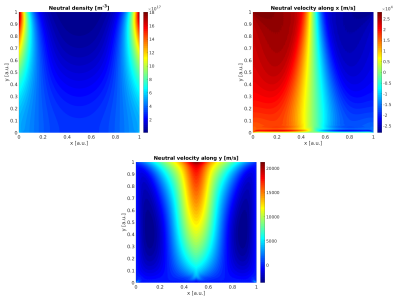

Figure 4: *Neutral density* $[m^{-3}]$ *(top left), horizontal (top right) and vertical (bottom) components of the neutral fluid velocity* $[m/s]$

5 Results: TCV-like geometry

To show the capabilities of the code to work also in realistic geometry, a simulation with a TCV-like one was performed. The plasma was considered once again only as a background, but it was taken from the results of a simulation [7] done with SolEdge2D-Eirene. Plasma density and electron temperature are shown in fig. 5.

The resulting neutral density distribution is shown in fig. 6, with also the neutral density computed with Eirene.

Differences between the two figures are mainly due the different physical descriptions of the transport (fully fluid vs. fully kinetic), keeping in mind that the not so high plasma density did not really justify the use of a fluid model.

The biggest difference is probably the underestimation of the density at targets; this can be explained upon noting that in the Eirene simulation a puff was present near the outer strike point, and both molecules and sputtered atoms (i.e. carbon) were also taken into account, while they were all absent in the fluid code simulation.

Perhaps a more interesting comparison between the two codes could be about the sources they would then fed to the plasma solver. In fig. 7 is thus shown the total particle source due to the interactions between neutral and charged particles. A good agreement can be found between the HDG simulation and Eirene, with again an underestimation of the source near the targets consistent with the lower density.

6 Conclusions

A new fluid code for neutral particles has been developed for SolEdge2D. This code solves a system of Euler-like equations for hydrogen atoms, taking into account a reduced number of reactions between neutral and charged particles. The code implementation has been verified with the method of the manufactured solutions, and initial results in realistic geometries already show a good level of agreement with Eirene.

References

[1] H. Bufferand et al, Nuclear Fusion **55**, 053025 (2015)

[2] D. Reiter, M. Baelmans and P. Borner, Fusion Science and Technology **47**, 172-186 (2005)

[3] Y. Marandet et al, IAEA 2016 proceedings

[4] N. Horsten et al, Nuclear Materials and Energy (In press, 2016)

[5] B. Cockburn et al, SIAM J. Numer. Anal. **47**, 1319-1365 (2009)

[6] G. Giorgiani et al, Computers & Fluids 98, 196-208 (2014)

[7] A. Gallo et al, submitted to Plasma Physics and Controlled Fusion

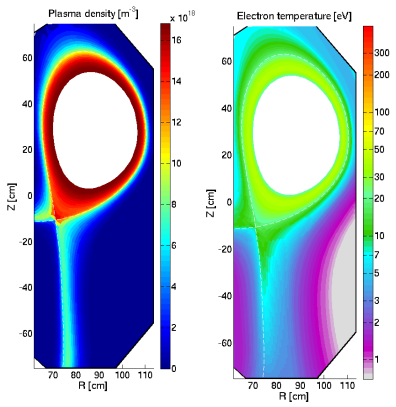

Figure 5: *Plasma density* $[m^{-3}]$ *(left) and electron temperature* $[eV]$ *(right).*

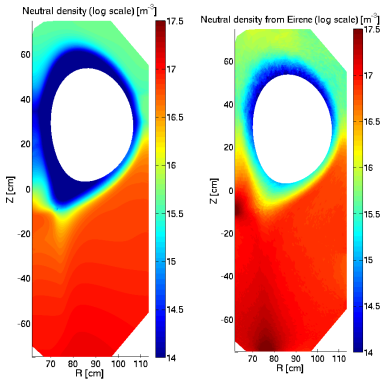

Figure 6: *Neutral density* $[m^{-3}]$ *in logarithmic scale computed by the fluid code (left) and Eirene (right).*

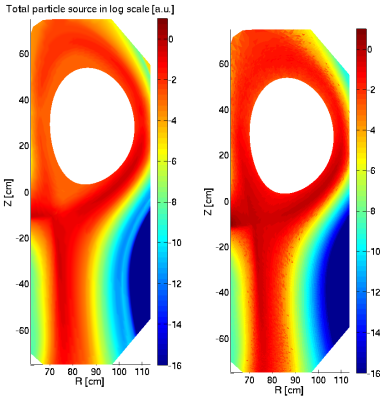

Figure 7: *Total particles sources computed by the fluid code (left) and Eirene (right), in logarithmic scale.*