Thèse présentée pour obtenir le grade universitaire de docteur

Discipline: Énergie, Rayonnement et Plasma

Matteo VALENTINUZZI

Modélisation numérique des flux de puissances sur les composants face au plasma de Tokamak à l’aide de techniques de couplage avancées entre codes fluides et cinétiques

Numerical modelling of power flux densities on tokamak plasma facing components by using advanced coupling techniques for kinetic and fluid codes

Soutenue le 13/12/2018 devant le jury composé de:

Prof. Fabio SUBBA Politecnico di Torino, Italie Rapporteur
Prof. Detlev REITER University of Düsseldorf et Forschungszentrum Jülich, Allemagne Rapporteur
Philippe Ghendrih CEA Cadarache, IRFM Examinateur
Francesca RAPETTI Université de Nice Sophia-Antipolis Examinateur
Laboratoire de Mathématiques Dieudonné
Eric SERRE CNRS/AMU, Laboratoire M2P2 Examinateur
Yannick MARANDET CNRS/AMU, Laboratoire PIIM Directeur de thèse
Prof. Domiziano MOSTACCI Università di Bologna, Italie Invité
Guido CIRAOLI CEA Cadarache, IRFM CEA encadrant

Numéro national de thèse/suffixe local: 2018AIXM0532/064ED352
Abstract

Power exhaust is one of the challenging issues that next generation of magnetized fusion devices (ITER, DEMO) will face. Power flux densities on plasma facing components have to be kept below engineering limits (typically $10\text{MW/m}^2$). Calculating these fluxes requires solving for both charged and neutral particles, which play a major role close to the wall.

The plasma is usually described within the fluid approach in 2D (assuming symmetry in the toroidal direction), while neutral particles usually require a kinetic treatment in most of the simulation domain owing to their large mean free paths.

Given the complexity of the geometry, and the fact that many neutral species have to be retained (atoms, molecules), this kinetic problem is usually addressed with a Monte Carlo approach. However, the kinetic treatment becomes CPU intensive, especially in next generation large devices when some regions of the plasma become almost hydrodynamic for neutrals, requiring the computation of very long trajectories with many collisions.

Improving the coupling between the fluid plasma solver and the kinetic Monte Carlo code for the neutrals has thus been identified as a critical bottleneck for performing realistic simulations for next step devices, while limiting the often prohibiting computation times required.

The first part of the thesis investigates the role of the statistical noise introduced by the Monte Carlo procedure on the Soledge2D-Eirene code. Both simplified cases and realistic simulations show that the plasma-neutral system is surprisingly robust to noise provided its correlation time is short compared to the plasma evolution time scales. It is thus possible to run the Soledge2D-Eirene code at a much lower cost simply by reducing the number of neutral histories (by a factor $100 \div 1000$), without losing significantly in accuracy.

While these results are promising, the problem is not completely solved since neutral particles trajectories in highly collisional regions can become exceedingly long.

For this reason, a hybrid kinetic-fluid model for the neutral particles, based on a two-phase approach, has been developed and implemented in order to combine the precision of the kinetic description with the speed of the fluid one.

A new fluid code for a fluid population of atoms has thus been developed and coupled to both the plasma solver Soledge2D and the Monte Carlo code Eirene, now applied to the remaining kinetic population of neutrals. The fluid code has furthermore been implemented with a Hybridizable Discontinuous Galerkin method, with advantages on
geometric flexibility, high order discretization and simpler parallelization.

Simulations in realistic ITER geometry show that the hybrid code is able to reproduce (both qualitatively and quantitatively) the results of the kinetic simulation for a surprisingly large range of “threshold collisionality”, even in mostly kinetic conditions. The hybrid code is shown to modestly out-perform the kinetic Monte Carlo code in a relatively low collisionality case explored in this work. Stronger gains are expected in cases where some regions have higher collisionality.
Contents

1 Power exhaust in Tokamaks .......... 2
  1.1 The power exhaust challenge in Tokamaks .................. 3
  1.2 Two-point model ............................................. 9
  1.3 Two dimensional modeling of the plasma-neutral system ........ 14
    1.3.1 Plasma species ........................................ 14
    1.3.2 Neutral species ...................................... 21
  1.4 Edge transport codes - Soledge2D-Eirene .................. 27
    1.4.1 Soledge2D ............................................ 27
    1.4.2 Eirene ................................................. 31
  1.5 Importance of neutral particles .......................... 35

2 Statistical noise ................. 48
  2.1 Statistical noise due to Eirene ...................... 49
  2.2 Analogy with turbulent transport ...................... 54
  2.3 Synthetic noise ....................................... 59
    2.3.1 General setup ...................................... 59
    2.3.2 Synthetic noise .................................. 61
    2.3.3 Results ............................................ 62
    2.3.4 Implications for edge transport codes .............. 66
  2.4 Effects of statistical noise in realistic Soledge2D-Eirene simulations .................. 67

3 Fluid code .......................... 76
  3.1 Introduction ........................................... 77
  3.2 Physical model ....................................... 78
  3.3 Numerical scheme .................................... 80
    3.3.1 Discretized system of equations ................. 83
    3.3.2 Verification .................................... 89
  3.4 Boundary conditions ................................. 92


Contents

3.5 Sources and coupling to Soledge2D ........................................... 96
3.6 Test cases .................................................................................. 99
   3.6.1 Slab geometry ................................................................. 99
   3.6.2 TCV geometry ............................................................... 101
   3.6.3 WEST geometry ............................................................ 103

4 Hybrid fluid-kinetic description .................................................... 110
   4.1 Introduction ........................................................................... 111
   4.2 Proposed hybrid model ....................................................... 113
   4.3 Evaporation and condensation processes ............................... 116
   4.4 Coupling Eirene and the fluid code ...................................... 121
   4.5 Discussion on the temperature closure ................................. 128
   4.6 ITER simulations: hybrid vs fully-kinetic .............................. 134
      4.6.1 Simulations setup ........................................................ 134
      4.6.2 Hybrid results ............................................................. 139
      4.6.3 Sources by plasma-neutrals interactions ....................... 141
      4.6.4 Particles and energy fluxes on the wall ......................... 145
      4.6.5 Computational performance ....................................... 146

Conclusions & perspectives ............................................................ 152
   Conclusions .............................................................................. 152
   Future work ............................................................................ 155

Bibliography .................................................................................. 158
Chapter 1

Power exhaust in Tokamaks

Contents

1.1 The power exhaust challenge in Tokamaks ..................... 3
1.2 Two-point model .................................................. 9
1.3 Two dimensional modeling of the plasma-neutral system .... 14
   1.3.1 Plasma species ................................................. 14
   1.3.2 Neutral species ............................................... 21
1.4 Edge transport codes - Soledge2D-Eirene ...................... 27
   1.4.1 Soledge2D .................................................... 27
   1.4.2 Eirene ....................................................... 31
1.5 Importance of neutral particles ............................... 35
1.1 The power exhaust challenge in Tokamaks

Power exhaust is one of the major challenges that future devices such as ITER and DEMO will have to overcome in the quest for nuclear fusion based energy.

If we consider a simple power balance,

$$P_{\text{heat}} + P_{\text{fus}} = P_{\text{loss}} + P_{\text{eff}}$$  \hspace{1cm} (1.1)

where $P_{\text{heat}}$ is the power injected by external heating sources (neutral beam injection, radio-frequency heating, ohmic heating), $P_{\text{fus}}$ is the power source generated by the fusion reactions, $P_{\text{loss}}$ contains all the power loss terms (operation of all secondary systems,...) and $P_{\text{eff}}$ is the power that can be extracted from the device.

This is not different from more "ordinary" energy production technologies, where power is generated by a specific physical mechanism like exothermic chemical reactions or nuclear fission reactions, and is then transferred as heat to one or more loops of cooling fluid. An example of a schematic representation of the system processes envisioned for a fusion power plant is shown in figure 1.1, and as such resembles any kind of "ordinary" power plant.

![Schematic diagram of a DEMO power plant](image)

Figure 1.1: Schematic diagram of a DEMO power plant, reprinted from reference [24]. The burning plasma is represented by the faint pink color confined in the D-shaped vessel; the heating sources are represented by the cyan lines, while three cooling systems are shown in this design respectively by the green, pink and blue loops.
One of the main differences is the mechanism employed to generate the heat: in order to profit from the power released during fusion nuclear reactions a hot plasma mixture that can reach temperatures higher than 10keV (more than $10^8 K$, $\sim 10$ times the temperature at the core of the Sun) is required.

Focusing on the $P_{\text{eff}}$ term, D-T reactions release $17.6MeV$ per reaction out of which $\sim 14.1MeV$ ($\sim 80\%$ of the total) are freed as kinetic energy of neutrons which is then deposited as heat in the blanket and, for a future fusion power plant, will be used to generate electric energy. The remaining $\sim 20\%$ of the power generated by the fusion reactions is given to kinetic energy of $\alpha$ particles, and through collisions, to the plasma.

Furthermore, a substantial fraction ($\sim 50\%$) of the latter is radiated in the hot region through bremsstrahlung, synchrotron and line radiation.

If we focus on ITER, its $Q = 10$ scenario ($Q = \frac{P_{\text{fus}}}{P_{\text{heat}}}$) envisions a fusion power production of $500MW$ over an injected power of $50MW$ by the heating systems. Using equation 1.1, we can estimate the total power flowing to the solid walls by the charged plasma particles of $\sim 100MW$ (note that ITER will not produce electric energy; the power extracted by the coolant system will just be dissipated in the environment).

To better understand the challenge posed by the task of dissipating these $\sim 100MW$, we need to introduce some useful concepts.

Due to the high temperatures reached in the center of the device, as a first approximation we can consider that all the gas contained in the Tokamak is ionized, so that we can consider only charged particles. To confine them a mix of poloidal and (much stronger) toroidal magnetic fields are used, as shown in figure 1.2, giving rise to helicoidally shaped magnetic field lines.

The toroidal component of the magnetic field, generated by a number of toroidal field coils situated at various toroidal positions of the device, exhibits a $\frac{1}{R}$ dependence, where $R$ is the distance from the axis (called also major radius). For this reason we will refer to "high field side" (HFS) for regions located close to the axis of the device, and "low field side" (LFS) for regions with higher major radii.

The plasma is contained in a vacuum vessel, and consequently some of the magnetic field lines are bound to intercept the walls of the device; we can thus distinguish between "open" and "closed" field lines depending if they cross or not the solid wall, as shown in green and blue respectively in figure 1.3. The region where the field lines are open is called "scrape-off layer" or SOL for short.

As the presence of a magnetic field forces charged particles to a helicoidal motion around the magnetic field lines (with a radius inversely proportional to the strength of the field)
1.1. The power exhaust challenge in Tokamaks

with only small perpendicular drift velocities due to inhomogeneities of the magnetic field, if on one hand closed field lines restrain the plasma in the hot region ("core") effectively increasing the particle and heat confinement, in the scrape-off layer the plasma can only drift a small distance in the direction perpendicular to the field lines in the same time it takes them to travel to the solid wall in the parallel direction.

To give some numbers, as it will be explained later the ions are able to reach sonic speeds (of the order of $\sim 10^4 \text{m/s}$ in SOL plasmas) along the parallel direction, while the perpendicular speed can be as low as 4 orders of magnitude smaller; as such, for usual lengths of the magnetic field lines (tens of meters) the distance traveled in the perpendicular direction can be of the order of some millimeters.

In the absence of external measures, this would result in demanding heat fluxes on the intersection points between the field lines and the solid walls. Note that not only attention has to be paid to the heat fluxes, but also to the particles fluxes on the solid surfaces as they are responsible for the local production of neutral particles (as it will be explained later) and impurities by sputtering, resulting at the same time in possible pollution of the plasma (for example by diluting the main ion used for the fusion reactions and consequently possibly lowering the fusion reaction rate), in local increase of the radiated power and possibly in net erosion of the wall.

Furthermore, if no measures are introduced uncertainties (in the position of the magnetic fields, in their tolerances, etc...) would also make it difficult to predict where the intersections would actually be during a shot.

What we can do is to thus force the intersecting points to areas of the wall dedicated
to the exhaust of particles and heat, by:

- protruding part of the wall, called *limiter*, towards the magnetic axis until it intercepts a magnetic field line;
- diverging the field lines to a specific zone.

We will focus on the second solution, called *divertor* configuration, shown in figure 1.3.

![Figure 1.3: Poloidal cross section of the magnetic field flux surfaces envisioned for ITER. In particular, in black is shown the position of the solid wall, in red the separatrix, in blue the closed field lines and in green the open field lines characteristics of the scrape-off layer.](image)

The idea is to add an additional coil (not shown in the figure) with a current flowing in the same direction as the plasma current; this way, a magnetic field is generated to cancel the poloidal component of the one created by the plasma itself near the walls, while keeping the core region mostly untouched.

The result is a decoupling of the core region, solely dedicated to the optimization of the fusion performances, and the divertor region, focused instead in the power and particle exhaust. The distinction between the two regions is marked by the X-point, the point where the poloidal magnetic field is ideally zero as shown in figure 1.2. At the same
1.1. The power exhaust challenge in Tokamaks

time, the field lines in the divertor region are bent to intercept the targets in dedicated areas, where the material and the cooling are designed to handle the fluxes.

Furthermore, the divertor configuration localizes the impurity source further from the core, preventing contamination and fuel dilution that are detrimental to the fusion reactions. Finally, the divertor also allows for efficient pumping of the neutrals, by achieving a higher compression in front of the pump openings, even more if aided by effectively closing the divertor entrance through a baffle.

We will call the magnetic field line that passes through the X-point and that works as a boundary between the confined plasma and the scrape-off layer the separatrix, and the two intersecting points "strike points"; we will often refer to "inner" and "outer" strike points (ISP and OSP, respectively) to distinguish between the high- or low-field side strike points. For completeness, the volume contained in the separatrix below the X-point is called private flux region (PFR), as it is not fed directly from the core plasma but rather by perpendicular transport along the separatrix legs; conversely, the volume outside the separatrix and below the X-point is named common flux region (CFR).

It is then in the scrape-off layer that the \( \sim 100 MW \) estimated before for the power to be dissipated flow to the material walls. To better understand the power exhaust problem, let us make some crude approximations to at least estimate the order of magnitude of power flux density that a future tokamak like ITER and DEMO will have to dissipate over their inner surfaces.

Let us start off by assuming a 2:1 asymmetry of heat flux between low- and high-field side (as seen experimentally for example in \cite{49} and references therein), thus resulting in \( \sim 33 MW \) flowing to the OSP and \( \sim 66 MW \) to the ISP.

As already said, plasma particles are able to move only a small distance \( \lambda \) in the radial direction perpendicular to the magnetic field lines in the same time it takes them to impact on the solid walls along the parallel direction. This means that the effective "plasma-wetted" area can be very small: thanks to the toroidal symmetry of the tokamaks, this area can be expressed as \( A = 2\pi R\lambda f \) where \( R \) is the major radius of the strike point, \( \lambda \) is the radial distance described above and \( f \) is the magnetic flux expansion, which takes into account the fact that the magnetic field lines are more densely packed at the upstream location than at the strike points. Let us take \( R = 4 m \) and \( R = 6 m \) for the ISP and OSP respectively and \( f \approx 10 \), realistic values similar to the ones envisioned for ITER. For the width \( \lambda \), if we simply extrapolate from a scaling law given by current devices, we find that in the most pessimistic scenario \( \lambda \approx 1mm \) for magnetic fields of the order of the ones envisioned for ITER\cite{23}*, giving a

\*Note that the scaling law from which this value has been extrapolated has a relatively wide dispersion and, more importantly, that the value of \( \lambda \) here refers to attached conditions, in which ITER will not
plasma-wetted area of $A_{in} \simeq 0.25m^2$ and $A_{out} \simeq 0.38m^2$.

With the information of the plasma-wetted area, we can then estimate the power flux densities impacting on the two strike points, namely $\sim 130MW/m^2$ and $\sim 170MW/m^2$ in the high- and low-field side.

Fortunately there is a way to increase the plasma-wetted area and consequently decrease the heat fluxes, that is to transfer a fraction of the heat to some agent that does not move along magnetic field line, but more isotropically. Because the isotropic transport of the plasma particles is broken by the magnetic fields, the idea is then to rely on particles that do not react to the latter, namely particles that do not have an electric charge - neutral particles, of which we will consider two main species.

The first one is photons, radiated by processes as ionization, recombination, bremsstrahlung, and other scattering events.

The other one is neutral particles as atoms and molecules, that in part contribute to radiate, but can also provide volumetric losses of power and momentum to the plasma through charge-exchange and elastic collisions. This second species of neutral particles, and among them hydrogenic atoms in particular, will be the focus of this manuscript. As such when in the following we will refer to neutral particles or neutrals, we will intend atoms and molecules.

Thanks to the dissipation of some of the power through these neutral particles, simulations predict a possible decrease of the heat flux density below the tolerated limits of about $10 \div 20MW/m^2$ in steady-state, and about twice as much during slow transients. These numbers are estimates to ensure that the material used for the most thermally loaded part of the wall, the divertor (which is usually tungsten for various current devices like WEST, JET, ASDEX Upgrade, ... and it is also going to be installed in ITER), will not re-crystallize ($1600K$) and that it will not melt (pure tungsten melts at $3695K$). Furthermore, this also takes into account for the need to keep the temperature of the coolant below its boiling point to optimize the heat exchange.

Note, finally, that a complementary problem to the power dissipation over the plasma-wetted surface exists, given by the temperature of the plasma impacting on the wall. In fact, the energy of the ions impacting on the solid surfaces is proportional to the temperature of both the ions themselves $T_i$ and, due to the acceleration caused by the presence of an electrical potential drop $\phi$ in a small sheath in front of the walls, of the electrons $T_e$

$$E_i = 2T_i + e\phi(T_e) \sim 2T_i + 3T_e$$  \hspace{1cm} (1.2)

operate. Furthermore, recent results from gyrokinetic simulations suggest that this width could actually be much larger than previously thought, see reference [11]
1.2. Two-point model

As we saw from the previous section, the transport of particles and heat in the scrape-off layer is dominated by the dynamics along the direction parallel to the magnetic field lines. A simple attempt to model the transport in the SOL is thus a zero-dimensional analysis between two points along the same magnetic field line: one point should represent

\[ \Gamma_{\text{sp}} = Y \Gamma_i \]

having assumed hydrogenic ions.

The problem is that the sputtering yield \( Y \) of ions impacting on a surface (and consequently the erosion rate \( \Gamma_{\text{sp}} = Y \Gamma_i \) of the walls) depends on the energy of the ions, as shown in figure 1.4.

On the other hand, as evident the sputtering yield shows a clear energy threshold depending on the surface binding energy and on the mass ratio between the projectile and surface elements\(^\dagger\). For this reason, while we want to decrease the power flux densities on the wall, it is as much important to limit the plasma temperature in the vicinity of the solid surfaces in order to decrease, for a given particle flux, the deposited energy per incident particle.

1.2 Two-point model

\(^\dagger\)Note that this discussion only applies to physical sputtering, as chemical sputtering does not have an energy threshold. Nonetheless, deuterium ions do not produce chemical sputtering on tungsten surfaces.
Chapter 1. Power exhaust in Tokamaks

the conditions of the plasma near the solid wall ("target" location, identified in the following with a subscript $t$), while the second one should represent the plasma on the same magnetic field line but far from the wall ("upstream" location, $up$).

We know that ions and electrons impact the walls with different dynamics due to high mass ratio between the two: electrons, in fact, tend to have higher mobility and the solid surface usually becomes negatively charge; this in turn repels further electrons (except for the high energy tail of their distribution functions) and it accelerates the ions. For this reason, it can be demonstrated [51] that the parallel velocity of the ions at the entrance of the called Debye sheath reaches the sound speed

$$u_i \geq c_s = \sqrt{\frac{T_e + T_i}{m_i}} \quad (1.3)$$

and the parallel ion Mach number $M = \frac{u_i}{c_s}$ thus reaches the unitary value.

Furthermore, due to the low collisionality of the sheath (even if this might not be true in ITER [60]) the heat flux is mainly convected and can thus be expressed as

$$q_e = \gamma_e T_e \Gamma_e$$  \hspace{1cm} (1.4)$$

$$q_i = \gamma_i T_i \Gamma_i$$  \hspace{1cm} (1.5)$$

where $\gamma_e = 2 + \Delta \tilde{\phi} \simeq 5.5$ and $\gamma_i \sim 2.5 \div 3.5$ are the sheath transmission factors for electrons and ions respectively.

The ion flux at the target (or more precisely at the entrance of the sheath in front of it) can be expressed as $\Gamma_{i,t} = n_i u_{i,t} = n_i c_i$.

If we neglect all perpendicular (to the magnetic field lines) transport the total pressure must be conserved at the two locations, so that

$$p_{up} + mn_{up}u_{up}^2 = p_t + mn_{i}u_{i}^2 \quad (1.6)$$

where $p = n_e T_e + n_i T_i$ is the static plasma pressure.

Due to the fact that the plasma is accelerated towards both divertor targets, there must be a point along the magnetic field line where the parallel ion velocity is zero; we will assume that this point (called stagnation point) coincides with the upstream location so that $u_{up} = 0$. If we furthermore assume a plasma made purely of singly charged ions and electrons, in the quasineutrality limit we can write that $n_e = n_i = n$; finally let us also assume that the temperature of ions and electrons are the same $T_e = T_i = T$, so
1.2. Two-point model

that using equation 1.3 the pressure conservation 1.6 simplifies to

\[ 2n_{up}T_{up} = 2n_tT_t + mn_t \frac{2T}{m} \]  

to (1.7)

finally giving \( n_{up}T_{up} = 2n_tT_t \).

Similarly, we can write a conservation equation for the energy between the two positions. Having assumed that the plasma velocity at the upstream location is zero, power at this position can only be transported by conduction \( q_{up} = -\kappa \nabla T \), where the thermal conductivity of the plasma is usually modelled through the Spitzer-Harm expression [55]

\[ \kappa = \kappa_0 T_5^{5/2} \]  

to (1.8)

with \( \kappa_0 \simeq 2000 \text{W/eV}^{7/2} \) for electrons \( (\simeq 60 \text{W/eV}^{7/2} m \) for hydrogen ions). Integrating the conduction expression \( q = -\kappa_0 T_5^{5/2} \frac{\partial T}{\partial s} \) (where we delimited the gradient of temperature only along the magnetic field line)

\[ \int_{up}^t qds = -\kappa_0 \int_{up}^t T_5^{5/2} dT \]  

to (1.9)

so that, assuming that the heat flux is constant between upstream and target locations (no sources/sinks or energy fluxes perpendicular to the magnetic field line), we can write

\[ q = \frac{2}{7} \kappa_0 \frac{T_{up}^{7/2} - T_t^{7/2}}{L} \]  

with \( L \) the distance between upstream and target locations along the field line (named connection length).

At the target, instead, we already saw that the power is mainly convected along the sheath \( q_t = \gamma e T_{e,t} \Gamma_{e,t} + \gamma i T_{i,t} \Gamma_{i,t} = \gamma n_t T_t c_s \).

We can thus write

\[ T_{up}^{7/2} = T_t^{7/2} + \frac{7 q_t L}{2 \kappa_0} \]  

to (1.10)

or, calling \( T_q = \left( \frac{2}{7} \frac{q_t L}{\kappa_0} \right)^{2/7} \)

\[ \left( \frac{T_{up}}{T_q} \right)^{7/2} - \left( \frac{T_t}{T_q} \right)^{7/2} = 1 \]  

to (1.11)

It is evident that there are two opposite limiting cases, namely for \( T_t \gg T_q \) and \( T_t \ll T_q \):

- in the first case, the ratio \( \frac{T_t}{T_q} \) is negligible and we obtain \( T_{up} = T_t \). Since there are no temperature gradients, to conserve the total pressure the density has to decrease from the upstream location to the target \( n_{up} = 2n_t \). Furthermore the temperature value is limited by the power convected in the sheath \( q_t = \gamma T_t \Gamma_t = \gamma n_t T_t \sqrt{\frac{2T_t}{m}} \), that
is to say \( T_t = \left( \frac{q \sqrt{m}}{\sqrt{2 \pi n_t}} \right)^{2/3} \), thus giving the name \textit{sheath limited regime} to this case. An important feature of this regime is the fact that the target density scales "only" linearly with the upstream density.

- in the second case, instead, we obtain a temperature gradient between the upstream location \( T_{up} \approx T_q \) and the target. This means that most of the heat flux is carried by conduction, thus giving the name \textit{conduction limited regime}.

Let us focus on the conduction limited regime for the moment. By manipulating equation (1.11), we can express the target temperature and target density as functions of the upstream quantities

\[
T_t \propto \frac{q^{10/7}}{L^{4/7} n_{up}^2} \quad (1.12)
\]

\[
n_t \propto q^{-8/7} L^{6/7} n_{up}^3 \quad (1.13)
\]

In order to mitigate the temperature at the target it is evident that this second regime is more advantageous than the sheath limited one, as the density at the target now scales as the third power of the upstream density and similarly the temperature at the target decreases quadratically while increasing the upstream density. Furthermore, there is also a change of the dependence of the particle flux on the target \( \Gamma_t \) between the two regimes, as it increases only linearly in the sheath limited regime while it increases quadratically \( \Gamma_t \propto n_{up}^2 \) with the upstream density in the conduction limited one.

While in the conduction limited regime we are able to decrease the temperature of the plasma in front of the targets, the heat load on the solid surface can still exceed the material limits due to the fact that the ions and electrons recombining on the surface still release the ionization potential energy \( \chi \) \((\approx 13.6eV\) per recombining pair in the case of Deuterium, so that if we assume that \( T_e = T_i \) equation 1.2 gives us \( E_i \sim 5T \) and we see that the heat flux on the targets is dominated by the recombination when \( T \sim \frac{13.6}{5} \sim 2.7eV \), resulting in a heat flux \( q = \chi \Gamma \) (for example, assuming a particle flux of \( \sim 10^{24} m^{-2} s^{-1} \), of the same of order of the flux expected on the ITER divertor targets, this amounts to \( \sim 2.2 MW/m^2 \) due just to the recombination of the charged particles at the wall) : to further reduce the latter it is thus necessary to reduce the particle flux.

Fortunately a new regime becomes accessible when the upstream density is increased further, and the particle flux at the target starts to decrease again: the \textit{detached regime}. In figure 1.5 is in fact shown the ion saturation current (that is proportional to the parallel ion flux) measured by Langmuir probes in JET discharges with increasing upstream
1.2. Two-point model

densities: in both the inner and outer divertor it is evident that, beyond a certain value of the upstream density the ion saturation current and the ion particle flux experience a rollover and decrease upon further increasing the density.

![Figure 1.5: Maximum ion saturation current measured in a series of JET L-mode discharges with NBI heating $P_{NBI} = 4MW$ as a function of the line averaged density. The solid line shows the evolution of the $j_{sat}$ during the density ramp phase of the shots, while the dots represent the value in steady state. Reprinted from reference [41].](image)

A way to quantitatively describe the detachment is the degree of detachment (DOD), defined experimentally as the ratio between the extrapolated "attached" ion flux at the target $\Gamma^{extr} = C\langle n_e \rangle^2$ following the two point model and the measured value [41]; $\langle n_e \rangle$ is the main plasma line averaged density and $C$ is a normalized constant obtained from the low density phase of the discharge.

$$DOD = \frac{\Gamma^{extr}}{\Gamma_{meas}}$$  \hfill (1.14)

Detachment can be identified when this index starts to be significantly larger than unity ($DOD >> 1$), as the measured ion flux rolls over with respect to the expected flux from the two-points model.

The two-point model might be expanded to include corrective terms, namely by the addition of "corrective factors" for the momentum $f_{mom}$ and power losses $f_{power}$ due to
interactions with neutral particles and impurity radiation allowing us to provide a simple explanation of the trends observed in experiments. Introducing neutrals and impurities provides a channel to remove momentum and energy from the magnetic field line, but geometrical effects consequently become important as they affect the compression of the neutrals and the radiation patterns. These effects, furthermore, are specific to each geometry and each machine, showing how modeling of the plasma in two or three dimensions is necessary.

1.3 Two dimensional modeling of the plasma-neutral system

The first question we should be asking when looking for a valid description for the transport of both plasma and neutrals is: how do we model them? Are we interested in a description on the length scale of the singular particles, or is a model on larger scale (for example, on the scale of the whole device) valid? This is a question asked also in other scientific domains, like the transport of neutral gases or liquids, and the answer is usually measured by the Knudsen number

\[ Kn = \frac{\lambda}{L} \]  

(1.15)
defined as the ratio between the mean free path \( \lambda \) of the particles and a macroscopic characteristic length \( L \) for the problem (for example, for gas flows in a channel it is often taken as the diameter of the flow channel):

- whenever particles rarely interact with one another (or with other particles) they tend to mostly free stream for long distances between collisions, and the mean free path will be long with respect to \( L \) (\( Kn > 1 \));

- when instead particles collide frequently, the distance between following collisions shrinks and the mean free path will be shorter (\( Kn << 1 \)).

In the first case, a kinetic description of the single particles should then be retained while in the second case we can use averaged equations for macroscopic variables of engineering interest to describe the entire system.

1.3.1 Plasma species

Let us start with the charged particles of the plasma; in the following we will only treat hydrogenic plasmas (\( H, D \) or \( T \)). The self-collisional mean free path for electrons and
1.3. Two dimensional modeling of the plasma-neutral system

Figure 1.6: Example of the dynamic of a particle in two different scenarios: on the left, the particle undergoes a low number of collisions and the mean free path is thus long with respect to the macroscopic length; on the right, the number of collisions increases and the mean free path decreases consequently.

Ions can be approximated as \[ \lambda \]

\[
\lambda_{e-e}[m] \simeq \lambda_{i-i}[m] \simeq \frac{10^{16} T[eV]^2}{n_e[m^{-3}]} \tag{1.16}
\]

To give some perspectives, scrape-off layer plasmas usually have low temperature (few eV to tens of eV) and high density (\(\sim 10^{19} - 10^{20} m^{-3}\)), resulting in mean free paths ranging from few millimeters to few meters. Furthermore the presence of neutrals in front of the targets further might further decrease the total mean free path. This scale length has to be compared to a macroscopic length, and for simplicity we will consider the connection length (that is the length along the parallel direction of an open magnetic field line from one target to the other). Values for the mean free path estimated with equation 1.16 and for the Knudsen number are shown in table 1.1 in function of realistic values of plasma density, temperature and connection length.
As we can see, the Knudsen number for the plasma particles generally results much smaller than unity, and a fluid description can often be applied. It has to be noted that, on the other hand, this situation might be marginal in some cases; furthermore, in conduction limited and detached regimes large and locally concentrated gradients of density and temperature arise and smaller macroscopic lengths $L$ (possibly of the order of the centimeter) should be used for the computation of the Knudsen number.

In these cases, the Knudsen number will increase and a purely fluid approximation fails. While a kinetic description should then be used to model the scrape-off layer in these low-collisionality scenarios, the application of a fully kinetical code for the plasma is currently too complex (for example, due to the need to take into account the self-generated electric field term and long range interactions in the collision operator) and consequently fluid codes are still used for the plasma and, in turn, some kinetic effects have to be introduced (for example by the implementation of heat flux limiters) to have a better agreement with the underlying kinetic description.

The kinetic description of the plasma would require knowing the distribution function $f(\vec{r}, \vec{v}, t)$ for each species (electrons, ions); this quantity is related to the probability of finding a particle at time $t$ in an infinitesimal volume $d\vec{r}$ around the position $\vec{r}$, with a velocity within $d\vec{v}$ around $\vec{v}$, simply as $dP = f(\vec{r}, \vec{v}, t)d\vec{r}d\vec{v}$.

Obviously the number of particles must be conserved so the distribution function, in absence of collisions, must satisfy the continuity equation

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + \frac{d\vec{r}}{dt} \cdot \frac{\partial f}{\partial \vec{r}} + \frac{d\vec{v}}{dt} \cdot \frac{\partial f}{\partial \vec{v}} = 0 \quad (1.17)$$

Noting that $\frac{d\vec{r}}{dt} = \frac{\vec{F}_{ext}}{m}$ where $\vec{F}_{ext}$ is the resultant of any external force acting on the

<table>
<thead>
<tr>
<th>$n_e$ [$m^{-3}$]</th>
<th>$T$ [eV]</th>
<th>$\lambda$ [m]</th>
<th>$Kn$ for $L = 1m$</th>
<th>$Kn$ for $L = 10m$</th>
<th>$Kn$ for $L = 100m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{19}$</td>
<td>1</td>
<td>0.001</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$10^{19}$</td>
<td>10</td>
<td>0.1</td>
<td>$10^{-1}$</td>
<td>$10^{-2}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>$10^{19}$</td>
<td>100</td>
<td>10</td>
<td>$10^{-2}$</td>
<td>1</td>
<td>$10^{-1}$</td>
</tr>
<tr>
<td>$10^{20}$</td>
<td>1</td>
<td>0.0001</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>$10^{20}$</td>
<td>10</td>
<td>0.01</td>
<td>$10^{-3}$</td>
<td>$10^{-4}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>$10^{20}$</td>
<td>100</td>
<td>1</td>
<td>$10^{-1}$</td>
<td>$10^{-1}$</td>
<td>$10^{-2}$</td>
</tr>
</tbody>
</table>

Table 1.1: Knudsen number for the plasma particles for different values of plasma density, temperature and connection length.
particle, the equation can be rewritten as

$$ \frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f + \frac{\vec{F}_{\text{ext}}}{m} \cdot \nabla_{\vec{v}} f = \left( \frac{\partial f}{\partial t} \right)_{C} $$  \hspace{1cm} (1.18)

where we added a collision operator on the right side to represent variations in the position in the phase space of the particles following collisions.

Because we will assume a low Knudsen number for scrape-off layer plasmas, let us define some macroscopic quantities starting from the particle density (units \([m^{-3}]\))

$$ n(\vec{r}, t) = \int f(\vec{r}, \vec{v}, t) d\vec{v} $$  \hspace{1cm} (1.19)

and the particle flux (units \([m^{-2}s^{-1}]\))

$$ \vec{\Gamma}(\vec{r}, t) = \int f(\vec{r}, \vec{v}, t) \vec{v} d\vec{v} $$  \hspace{1cm} (1.20)

We can thus define the fluid velocity as

$$ \vec{u}(\vec{r}, t) = \frac{\vec{\Gamma}}{n} = \frac{1}{n} \int f(\vec{r}, \vec{v}, t) \vec{v} d\vec{v} $$  \hspace{1cm} (1.21)

Let us also define the second and third moments of the distribution function, namely the pressure tensor (units \([Pa]\))

$$ \Pi(\vec{r}, t) = m \int f(\vec{r}, \vec{v}, t) (\vec{v} - \vec{u}(\vec{r}, t)) \otimes (\vec{v} - \vec{u}(\vec{r}, t)) d\vec{v} $$  \hspace{1cm} (1.22)

and the heat flux (units \([W/m^2]\))

$$ \vec{q}(\vec{r}, t) = \frac{1}{2} m \int f(\vec{r}, \vec{v}, t) (\vec{v} - \vec{u}(\vec{r}, t))^2 (\vec{v} - \vec{u}(\vec{r}, t)) d\vec{v} $$  \hspace{1cm} (1.23)

To obtain equations for these macroscopic quantities, we have to take successive moments of the kinetic equation 1.18. Its m-th moment can be written as

$$ \int \frac{\partial f}{\partial t} \vec{v}^m d\vec{v} + \int \vec{v} \cdot \nabla f \vec{v}^m d\vec{v} + \int \frac{\vec{F}_{\text{ext}}}{m} \cdot \nabla_{\vec{v}} f \vec{v}^m d\vec{v} = 0 $$  \hspace{1cm} (1.24)

$$ \Rightarrow \frac{\partial}{\partial t} \int f \vec{v}^m d\vec{v} + \int \nabla_{\vec{r}} (f \vec{v}) \vec{v}^m d\vec{v} - \int f \nabla_{\vec{r}} \cdot \vec{v}^m d\vec{v} + \int \nabla_{\vec{v}} \left( \frac{\vec{F}_{\text{ext}}}{m} \right) \vec{v}^m d\vec{v} - \int f \nabla_{\vec{v}} \cdot \frac{\vec{F}_{\text{ext}}}{m} \vec{v}^m d\vec{v} = 0 $$

For charged particles in strongly magnetized plasmas the external force can be calculated as the Lorentz force \( \vec{F}_{\text{ext}} = q \left( \vec{E} + \vec{v} \times \vec{B} \right) \) with \( \vec{E} \) the electric field, \( q \) the charge of
the particle and $\vec{B}$ the magnetic field, so that

$$\nabla_v \cdot \vec{F}_{ext} = q \left( \nabla_v \cdot \vec{E} + \nabla_v \cdot \vec{v} \times \vec{B} \right) = q \left( \vec{B} \cdot \nabla_v \times \vec{v} - \vec{v} \cdot \nabla_v \times \vec{B} \right) = 0$$

(1.25)

$$\Rightarrow \frac{\partial}{\partial t} \int f \vec{v} \cdot \vec{v} d\vec{v} + \nabla_r \cdot \int f \vec{v}^{m+1} d\vec{v} + \int \nabla_v \cdot \left( f \frac{\vec{F}_{ext}}{m} \right) \vec{v}^{m} d\vec{v} = 0$$

$$\Rightarrow \frac{\partial}{\partial t} \int f \vec{v}^{m} d\vec{v} + \nabla_r \cdot \int f \vec{v}^{m+1} d\vec{v} + \int \nabla_v \cdot \left( f \frac{\vec{F}_{ext}}{m} \vec{v}^{m} \right) d\vec{v} - \int f \frac{\vec{F}_{ext}}{m} \cdot \nabla_v \vec{v}^{m} d\vec{v} = 0$$

Using the property $\nabla_v \vec{v}^m = m I \vec{v}^{m-1}$ and assuming that $f \vec{v}^m$ is negligible on the boundary of the velocity space (the number of particles with velocity that tends to infinity should tend to zero), the equation for the $m$-th moment becomes

$$\frac{\partial}{\partial t} \int f \vec{v}^m d\vec{v} + \nabla_r \cdot \int f \vec{v}^{m+1} d\vec{v} + \int \nabla_v \cdot \left( f \frac{\vec{F}_{ext}}{m} \vec{v}^m \right) d\vec{v} - \int f \frac{\vec{F}_{ext}}{m} \cdot \nabla_v \vec{v}^m d\vec{v} = 0$$

(1.26)

As evident from equation 1.26, the evolution of the $m$-th moment depends also on the $(m - 1)$-th and, more importantly, on the $(m + 1)$-th moment. Without going into the details of the derivation, the system of equations relative to the first three moments $m = 0, 1, 2$ can be written as

$$\frac{\partial n}{\partial t} + \nabla \cdot (n \vec{u}) = 0$$

(1.27)

$$\frac{\partial}{\partial t} (n \vec{u}) + \nabla \cdot \left( n \vec{u} \otimes \vec{u} + \frac{\Pi}{m} \right) - n \frac{\vec{F}_{ext}}{m} = \vec{0}$$

(1.28)

$$\frac{\partial E_t}{\partial t} + \nabla \cdot E_t \vec{u} + \nabla \cdot (n \vec{u} \Pi) + \nabla \cdot \vec{q} - n \vec{u} \cdot \vec{F}_{ext} = 0$$

(1.29)

where we defined the total energy

$$E_t = \frac{1}{2} m Tr \left( \int f \vec{v} \otimes \vec{v} d\vec{v} \right) = \frac{1}{2} m Tr \left( n \vec{u} \otimes \vec{u} + \frac{\Pi}{m} \right) = \frac{1}{2} m n u^2 + \frac{3}{2} p$$

(1.30)

To obtain a solution for the system of equations it is therefore necessary to introduce some assumption to approximate the yet unknown moments of the distribution function: in this specific case, we need to express the heat flux $\vec{q}$ and stress tensor $\Pi$ as functions
of the previous moments $n$ and $\bar{u}$.

As we have already shown, scrape-off layer plasmas are usually assumed to be highly collisional and, as kinetic theory shows, any ensemble of particles in presence of collisions relaxes from any arbitrary initial velocity distribution to a local Maxwellian distribution function

$$f_M(\vec{r}, \vec{v}, t) = n(\vec{r}, t) \left( \frac{2\pi T(\vec{r}, t)}{m} \right)^{-3/2} \exp \left( -\frac{m(\vec{v} - \bar{u}(\vec{r}, t))^2}{2T(\vec{r}, t)} \right)$$  \hspace{1cm} (1.31)

In these conditions the missing moments can be estimated using a closure like, for example, a Chapman-Enskog procedure [13]. The closure approximation can be expressed expanding the distribution function of the charged particles around a Maxwellian vdf in terms of the Knudsen number

$$f(\vec{r}, \vec{v}, t) = f_M + \sum_{i=1}^{\infty} Kn_i f_i(\vec{r}, \vec{v}, t)$$  \hspace{1cm} (1.32)

where $f_i(\vec{r}, \vec{v}, t)$ for $i = 1, ..., \infty$ are corrections to the Maxwellian. Substituting the expansion 1.32 in the system of equations allows us to obtain equations for the corrections $f_i$, and consequently to close the system of equations [6]

$$\vec{q}_e = -3.16 \frac{n_e T_e \tau_e}{m_e} \nabla \parallel T_e - 4.66 \frac{n_e T_e}{m_e \omega_e^2 \tau_e} \nabla \perp T_e - \frac{5 n_e T_e}{2 eB} \vec{b} \times \nabla T_e + 0.71 n_e T_e \bar{u} \parallel + \frac{3 n_e T_e}{2 \omega_e \tau_e} \vec{b} \times \bar{u} =$$

$$= -\chi_e \parallel \nabla \parallel T_e - \chi_e \perp \nabla \perp T_e - \frac{5 n_e T_e}{2 eB} \vec{b} \times \nabla T_e + 0.71 n_e T_e \bar{u} \parallel + \frac{3 n_e T_e}{2 \omega_e \tau_e} \vec{b} \times \bar{u}$$  \hspace{1cm} (1.33)

$$\vec{q}_i = -3.9 \frac{n_i T_i \tau_i}{m_i} \nabla \parallel T_i - \frac{2 n_i T_i}{m_i \omega_i^2 \tau_i} \nabla \perp T_i + \frac{5 n_i T_i}{2 ZeB} \vec{b} \times \nabla T_i =$$

$$= -\chi_i \parallel \nabla \parallel T_i - \chi_i \perp \nabla \perp T_i + \frac{5 n_i T_i}{2 ZeB} \vec{b} \times \nabla T_i$$  \hspace{1cm} (1.34)

where the subscript $e$ or $i$ indicates respectively electrons and ions, the subscript $\parallel$ and $\perp$ are with respect to the direction of the magnetic field $\vec{b} = \frac{\vec{B}}{B}$, $n$ is the density of the species, $T$ its temperature, $\tau = \frac{3\sqrt{mT^{3/2}}}{4\sqrt{2\pi} \Lambda}$ the collision time, $m$ the mass and $\omega = \frac{eB}{mc}$ the cyclotronic frequency, having assumed that the ion is singly ionized. The stress tensor $\Pi$ can also be expressed as function of the other defined quantities, but the expression is a bit more complicated and it can be found in reference [6].

The final system of equations obtained with the above closure can be written as
Chapter 1. Power exhaust in Tokamaks

\[ \frac{\partial n_\alpha}{\partial t} + \nabla \cdot (n_\alpha \vec{u}_\alpha) = 0 \] (1.35)

\[ \frac{\partial n_\alpha \vec{u}_\alpha}{\partial t} + \nabla \cdot \left( n_\alpha \vec{u}_\alpha \otimes \vec{u}_\alpha + \frac{\Pi_\alpha}{m_\alpha} \right) = \frac{e n_\alpha}{m_\alpha} (\vec{E} + \vec{u}_\alpha \times \vec{B}) + \vec{R} \] (1.36)

\[ \frac{\partial}{\partial t} \left( \frac{1}{2} m_\alpha n_\alpha u_\alpha^2 + \frac{3}{2} p_\alpha \right) + \nabla \cdot \left( \frac{1}{2} m_\alpha n_\alpha u_\alpha^2 + \frac{3}{2} p_\alpha \vec{u}_\alpha \right) + \nabla \cdot (\vec{u}_\alpha \cdot \Pi_\alpha) = \nabla \cdot \left( \chi_{\parallel,\alpha} \nabla \| T_\alpha + \chi_{\perp,\alpha} \nabla \perp T_\alpha \right) + Q_{\alpha,\beta} + \vec{R} \cdot \vec{u}_\alpha + e n_\alpha \vec{u}_\alpha \cdot \vec{E} \] (1.37)

takes the name of Braginskii’s equations and it is valid for particles in a plasma of electrons and singly charged ions.

Note that in deriving these equations we also assume quasi-neutrality of the plasma \( n_e = \sum_{j=1}^{N_{ions}} n_j \), so that for a simple D plasma \( n_e = n_i \) and only a single continuity equation is needed.

In the following, we will also assume ambipolarity \( \vec{u}_e = \vec{u}_i \) so that a single parallel momentum equation is needed.

Furthermore the small mass ratio \( m_e/m_i \) makes it possible to neglect the kinetic energy of the electrons in the electron energy equation and momentum exchanges in ion-electron and neutral-electron collisions.

As we can see from the equations, a force term \( \vec{R} \) is introduced in the momentum equation 1.28 and in the energy equation 1.29 due to collisions between ions and electrons, and similarly a heat exchange term \( Q \) is added in the energy equation; of course, to preserve total momentum and energy balance, in a pure D plasma \( \vec{R}_{e-i} = \vec{R}_{i-e} = \vec{R} \) and \( Q_{e-i} + Q_{i-e} + \vec{R} \cdot (\vec{u}_e - \vec{u}_i) = 0 \).

As in some cases the collisionality of the plasma might not fully justify a fluid closure, some kinetic effects have to be retained: for this reason, flux limiters are applied to the Spitzer-Harm expression of the parallel heat flux to account for such effects in low collisionality regimes

\[ q = \frac{q_{SH}}{1 + \alpha \frac{q_{SH}}{q_{max}}} \] (1.38)

where \( q_{SH} = -\kappa \nabla \| T = -\kappa_0 T^{5/2} \nabla \| T \) is the Spitzer-Harm heat flux \([55]\), \( q_{max} = n v_{th} T \) is the limiting value for the heat flux set to the free streaming value with \( v_{th} \) the thermal velocity of the particles, and \( \alpha \) is an ad hoc parameter to choose the maximum value of the heat flux; this parameter \( \alpha \) is usually taken in the range \((0.1, 3.)\). Recent developments \([9, 15]\) try to reformulate the expression of the heat flux taking into account non-local
heat transfer adding a delocalization kernel $w(x, x')$

$$q(x) = - \int w(x, x') \kappa(x') \nabla_{\parallel} T(x') dx' + q_{BC}$$  \hspace{1cm} (1.39)$$

where $x$ is the coordinate along the parallel direction and $q_{BC}$ are terms added to take into account the impact of the sheath boundary conditions in the heat flux and to thus avoid discontinuities of the temperature profile near the walls.

Note finally that in a strongly magnetized plasma, assuming that $\rho_L \ll \frac{B}{|\nabla B|}$, there is a scale separation between the fast gyration of the charged particles and the slower scales related to cross field transport/macroscopic flows we would like to study.

Ordering the equations according to this small parameter shows that the cross-field transport can be described in terms of various drifts resulting from e.g; the average effect of forces acting on the underlying gyrating trajectories. This formulation is the one implemented in the numerical tools we will use in this work.

1.3.2 Neutral species

Let us now consider the neutral gas. As before, we need to ask ourselves whether a fluid description is valid or not. Differently from the plasma species, neutral particles behave kinetically ($Kn > 1$) in most of the volume of the scrape-off layer, and the full kinetic equation 1.18 should be solved to properly model the neutral gas; in this case, on the other hand, we can neglect the external force due to the fact that atoms and molecules do not have an electric charge (and thus do not feel the Lorentz force) and the gravitational pull is small with respect to the dominating collisions and pressure gradient.

Furthermore, in contrast to the kinetic equation for the plasma particles, a much simpler linear Boltzmann equation has to be solved.

The main mechanism (or rather set of mechanisms) that generates the neutral gas is the so-called recycling. The solid walls of the devices act like sinks for the plasma, and as such the ions accelerated by the sheath and the energetic tail of the electrons impact on the surface. A fraction of the impinging flux of ions "instantly" recombines with an electron and is backscattered as atoms while the remaining part is implanted in the wall surface and thermalizes with the bulk wall material. A part of the thermalized particles may get trapped by lattice defects (grain boundaries, vacancies, etc...), but the remaining fraction can diffuse through interstitial gaps and/or porous channels of the material to be finally adsorbed back at surface sites; then the adsorbed particles may recombine to form a molecule ($H_2, D_2, ...$) and the creation of the chemical bond can provide enough energy to desorb the molecule from the surface into the vacuum vessel [22]. A simplified
picture of the possible recycling mechanisms just explained is shown in figure 1.7.

While the description of this process has been provided in a simplified fashion (voluntarily forgetting other processes like sputtering, co-deposition, etc...), it is of the utmost importance for plasma devices as it ensures that, when all the available lattice defects are saturated (if they are not, the plasma is effectively pumped by the wall), most of the charged particles lost to the walls are, promptly or slowly, re-fed to the plasma though the eventual ionization of the atoms and molecules.

If we only take into account the backscattering part of the process, we can define reflection coefficients for the particles $R_n$ and for the energy $R_E$ so that [21]

$$N_{\text{back-scattered}} = R_n N_{\text{impacting}}$$
$$\overline{E}_{\text{back-scattered}} = \frac{R_E}{R_n} \overline{E}_{\text{impacting}}$$  \hspace{1cm} (1.40)

where N indicates the particles flux and $\overline{E}_{\text{back-scattered}}$ is the mean energy of the reflected particles. The reflection coefficients, the direction and the mean energy of the backscattered particles depend on the particle itself and on the material of the wall (in fact, they depend on the mass ratio between the elements), on the angle $\alpha$ against the normal to the wall of the impacting particles and on their energy before the impact. As an example, the particle and energy reflection coefficients for Deuterium impacting on a Tungsten substrate are shown in figure 1.8.

Furthermore, molecules are also usually puffed, as shown in figure 1.9, in different
1.3. Two dimensional modeling of the plasma-neutral system

Figure 1.8: Particle $R_n$ (on the left) and energy $R_E$ (on the right) reflection coefficients for Deuterium on Tungsten as a function of the energy of the impacting particles, obtained from the TRIM database [42]. The different colors show the dependency of the coefficients on the impact angle $\alpha$ (in degrees) against the normal ($\alpha = 0$)

areas of the Tokamaks to reduce the temperature near the divertor and/or increase the density of the plasma.

Atoms, molecules, ions and electrons can chemically interact with one another following a large number of reactions and collisions. The colder the plasma, the more reactions we should take into account as chemistry becomes more and more important; as shown in reference [39], for scrape-off layer plasmas the set of processes of table1.2 is needed for correctly modeling of neutral interactions in detached regimes.

<table>
<thead>
<tr>
<th>Reaction Type</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ionization</td>
<td>$D + e^- \rightarrow D^+ + 2e^-$</td>
</tr>
<tr>
<td>Radiative recombination</td>
<td>$D^- + e^- \rightarrow D + h\nu$</td>
</tr>
<tr>
<td>Three-body recombination</td>
<td>$D^- + 2e^- \rightarrow D + e^-$</td>
</tr>
<tr>
<td>Charge-exchange</td>
<td>$D + D^+ \rightarrow D^+ + D$</td>
</tr>
<tr>
<td>Non-dissociative ionization</td>
<td>$D_2 + e^- \rightarrow D_2^+ + 2e^-$</td>
</tr>
<tr>
<td>Dissociation</td>
<td>$D + e^- \rightarrow 2D + e^-$</td>
</tr>
<tr>
<td>Dissociative ionization</td>
<td>$D_2 + e^- \rightarrow D + D^+ + 2e^-$</td>
</tr>
<tr>
<td>Elastic collision</td>
<td>$D_2 + D^+ \rightarrow D_2 + D^+$</td>
</tr>
<tr>
<td>Ion conversion</td>
<td>$D_2 + D^+ \rightarrow D_2^+ + D$</td>
</tr>
<tr>
<td>Dissociative recombination</td>
<td>$D_2^+ + e^- \rightarrow 2D$</td>
</tr>
<tr>
<td>Dissociative excitation</td>
<td>$D_2^+ + e^- \rightarrow D + D^+ + e^-$</td>
</tr>
<tr>
<td>Dissociative ionization</td>
<td>$D_2^+ + e^- \rightarrow 2D^+ + 2e^-$</td>
</tr>
</tbody>
</table>

Table 1.2: Atomic and molecular reactions used in reference [39].
As we will see, in the next chapters we will mainly focus on the modeling of atoms, so that the reduced set of reactions given only by ionization, recombination and charge-exchange will be used. Since collisions between atoms and plasma particles do not involve long range interactions, the effect of these reactions on the kinetic equation can be simplified as the Boltzmann binary collision operator

$$\left( \frac{\partial f_i}{\partial t} \right)_C = \int (f^*_a f^*_b - f_a f_b) g \sigma (\tilde{v}^*_a \tilde{v}^*_b - \tilde{v}_a \tilde{v}_b)$$

where subscripts $a$ and $b$ indicate the species (in particular, $b$ for the background species), $f^*$ and $\tilde{v}^*$ indicate respectively the distribution function and the velocity of the particles after the collision, $g = |\tilde{v}_a - \tilde{v}_b|$ ($= |\tilde{v}^*_a - \tilde{v}^*_b|$ assuming elastic collisions) is the amplitude of the relative velocity of the particles, and $\sigma$ is the cross section of the collision.

If we focus on the reactions of ionization, recombination and charge exchange, we can then write:

- ionization acts as a sink of atoms \((\frac{\partial f_a}{\partial t})_{iz} = - \int f_a f_e g \sigma_{iz} d\tilde{v}_a d\tilde{v}_e d\tilde{v}_c\)
- for ions, instead, it acts as a source term \((\frac{\partial f_i}{\partial t})_{iz} = f_a \int f_e g \sigma_{iz} d\tilde{v}_e\)
- for electrons, a second electron is created (assumed with negligible kinetic energy as it was previously bound in the atom) \((\frac{\partial f_e}{\partial t})_{iz} = f_e \int f_a g \sigma_{iz} d\tilde{v}_a\)
1.3. Two dimensional modeling of the plasma-neutral system

- recombination is a sink for both ions \( \left( \frac{\partial f_i}{\partial t} \right)_{rc} = -f_i \int f_e g \sigma_{rc} d\vec{v}_e \) and electrons \( \left( \frac{\partial f_e}{\partial t} \right)_{rc} = -f_e \int f_i g \sigma_{rc} d\vec{v}_i \), and creates an atom \( \left( \frac{\partial f_a}{\partial t} \right)_{rc} = f_a \int f_e g \sigma_{rc} d\vec{v}_e \);

- finally charge exchange transfers momentum between ions and atoms, so that \( \left( \frac{\partial f_e}{\partial t} \right)_{cx} = \left( \frac{\partial f_i}{\partial t} \right)_{cx} = f_i \int f_a g \sigma_{cx} d\vec{v}_a - f_a \int f_i g \sigma_{cx} d\vec{v}_i \)

If we then assume a Maxwellian distribution function for the background species we can introduce effective reaction rates

\[
\langle \sigma v \rangle = \frac{1}{n_b} \int f_b \sigma \vec{v}_b d\vec{v}_b
\] (1.42)

which give an information about the average number of reactions that happen per unit of time and unit of projectile density; furthermore the contribution of all excited states of the atom or ion is considered, thus resulting in effective rates: for example, ionization takes into account both single-step (by a single electron with energy greater than the threshold of the ground state atom) and multi-step ionization (considering an atom in an already excited energy state).

![Figure 1.10: Reaction rates for ionization, recombination and charge exchange of Deuterium as functions of projectile temperature, for \( n_e = 1 \cdot 10^{19} m^{-3} \).](image)

Effective reaction rates for ionization, recombination and charge exchange are shown in figure 1.10 in function of the temperature of the "projectile" (ions for the charge exchange, electrons for the others). As evident, ionization predominates over the recombination in the whole temperature range, except at really low temperatures \( T_e < 1 eV \) because of the energy threshold behavior of ionization.
Thus the Boltzmann equation for the atoms becomes

$$\frac{\partial f_a}{\partial t} + \vec{v} \cdot \nabla_r f_a = f_i n_e \langle \sigma v \rangle_{rc} - f_a n_e \langle \sigma v \rangle_{iz} - f_a n_i \langle \sigma v \rangle_{cx} + f_i \int f_a \sigma_{cx} g d\vec{v}_a \quad (1.43)$$

Note that the integral over the velocity of the atoms has to be retained for the charge exchange contribution as the velocity distribution function of the atoms is not known. Nonetheless, if we assume that the dependence of $g\sigma_{cx}$ on the velocity of the atoms is negligible, we can still use the definition of reaction rate in equation 1.42 and approximate this term to

$$\int f_a \sigma_{cx} g d\vec{v}_a \simeq n_a \langle \sigma v \rangle_{cx}$$

For this reason, in the following the Boltzmann equation for the atoms will be written as

$$\frac{\partial f_a}{\partial t} + \vec{v} \cdot \nabla_r f_a = f_i n_e \langle \sigma v \rangle_{rc} - f_a n_e \langle \sigma v \rangle_{iz} + (f_i n_a - f_a n_i) \langle \sigma v \rangle_{cx} \quad (1.44)$$

Similarly, we can derive source terms for the distribution function of electrons and ions

$$S_{f,e} = f_e n_a \langle \sigma v \rangle_{iz} - f_e n_a \langle \sigma v \rangle_{rc}$$
$$S_{f,i} = f_i n_a \langle \sigma v \rangle_{iz} - f_i n_a \langle \sigma v \rangle_{rc} + (f_i n_a - f_a n_i) \langle \sigma v \rangle_{cx}$$

To obtain sources for the fluid equations (1.35-1.37), we calculate then the first three moments of these distribution functions, resulting in

$$S_{n,e} = S_{n,i} = n_a n_e \langle \sigma v \rangle_{iz} - n_i n_e \langle \sigma v \rangle_{rc}$$
$$\vec{S}_{nli} = n_a \vec{u}_a n_e \langle \sigma v \rangle_{iz} - n_i \vec{u}_i n_e \langle \sigma v \rangle_{rc} - (n_i \vec{u}_i n_a - n_a \vec{u}_a n_i) \langle \sigma v \rangle_{cx}$$
$$S_{E,e} = n_a n_e \langle \sigma v \Delta E \rangle_{iz} - n_i n_e \langle \sigma v \Delta E \rangle_{rc}$$
$$S_{E,i} = n_a E_a n_e \langle \sigma v \rangle_{iz} - n_i E_i n_e \langle \sigma v \rangle_{rc} - (n_i E_i n_a - n_a E_a n_i) \langle \sigma v \rangle_{cx} \quad (1.45)$$

where $E_a$ and $E_i$ are the total energy ($E = \frac{1}{2} mn u^2 + \frac{3}{2} p$) of atoms and ions respectively. As already explained, we neglect the momentum of the electrons due to the mass ratio between electrons and ions/atoms, so we can neglect the electron momentum source. In the electron energy source we introduced the electron energy loss weighted reaction rates $\langle \sigma v \Delta E \rangle$ as we have to now take into account the energy loss per event (or the energy cost) $\Delta E$ associated with the process.
1.4 Edge transport codes - Soledge2D-Eirene

Numerical codes have been developed by the community to simulate the physical and chemical aspects of what happens inside a tokamak, ranging from simulations of core dynamics on fusion yields to surface modification under ion fluxes, from optimization of magnetic field coils to design of coolant systems, etc. In this manuscript, we will focus on the codes that have been developed for the study of the plasma transport in the scrape-off layer.

1.4.1 Soledge2D

Usually, these kind of codes (for example SOLPS-ITER [65], EDGE2D [31], SONIC [54], UEDGE [52], ...) apply a fluid description for the charged plasma particles in a two dimensional $(r, \theta)$ grid by assuming toroidal symmetry; furthermore, as the plasma motion is preferentially along magnetic field lines, the particles velocity $\vec{u} = u_{\parallel} \vec{b} + \vec{u}_{\perp}$ is usually divided in a main parallel component $u_{\parallel}$ and in a smaller perpendicular component $u_{\perp}$, where $\vec{b} = \frac{\vec{B}}{|\vec{B}|}$ is the unit vector with the direction of the magnetic field. In the following, we will drop the $\parallel$ subscript for the parallel component of the velocity and directly use $u$ for the parallel velocity.

These transport codes usually fall in the RANS (Reynolds averaged Navier-Stokes) type, in which the field variables $x$ are decomposed in a slowly varying mean field component $\langle x \rangle$ and in a fast fluctuating component $\tilde{x}$ due to turbulence. The underlying hypothesis is that the we can thus separate two different time and length scales for the two components, so that the derivatives of the mean field components with respect to the turbulence time and length variables is negligible. This let us write then equations for the mean fields component of the fields only, in which the effect of the fluctuations can be modeled as diffusive processes [7].

We will focus in the following on the Soledge2D code developed at the IRFM in collaboration with the M2-P2 laboratory and the French Federation for the Fusion by Magnetic Confinement FR-FCM. The equations for the code can be obtained applying the drift ordering to the system 1.35-1.37 and can be expressed in their simplest model (no perpendicular drifts considered, no fluid turbulence equations, single species D plasma) as [10]:
\[
\frac{\partial n}{\partial t} + \nabla \cdot \left( n \nu \vec{b} + n \vec{v}_\perp \right) = S_n
\]  
(1.46)

\[
\frac{\partial nu}{\partial t} + \nabla \cdot \left( nu \left( \vec{u} b + \vec{v}_\perp \right) \right) =
\]

\[
= -\nabla \left( \frac{nT_i}{m_i} \right) + \frac{enE_i}{m_i} + \frac{R_{ei}}{m_i} + \nabla \cdot \left( \nu n \vec{v}_\perp \cdot \vec{u} \right) + S_{nu}
\]
(1.47)

\[
\frac{\partial \left( \frac{3}{2} nT_e \right)}{\partial t} + \nabla \cdot \left( \frac{5}{2} nT_e \left( \vec{u} b + \vec{v}_\perp \right) \right) =
\]

\[
= \nabla \cdot \left( \kappa_e \nabla \parallel T_e \vec{b} + \chi_e n \nabla \perp T_e \right) - enuE\parallel - R_{ei} u - Q_{ei} + S_{Ee}
\]
(1.48)

\[
\frac{\partial \left( \frac{3}{2} nT_i + \frac{1}{2} m_i n u^2 \right)}{\partial t} + \nabla \cdot \left( \left( \frac{5}{2} nT_i + \frac{1}{2} m_i n u^2 \right) \left( \vec{u} b + \vec{v}_\perp \right) \right) =
\]

\[
= \nabla \cdot \left( \kappa_i \nabla \parallel T_i \vec{b} + \chi_i n \nabla \perp T_i + \nu n \nabla \perp \left( \frac{1}{2} m_i u^2 \right) \right) + enuE\parallel + R_{ei} u + Q_{ei} + S_{Ei}
\]
(1.49)

where \( \vec{v}_\perp = -\frac{1}{n} D \nabla \perp n + \vec{v}_\text{pinch} \) and the source terms \( S_n, S_{nu}, S_{Ee} \) and \( S_{Ei} \) are due to collisions and reactions with the neutrals (the energy sources can also include external heating).

As shown, the diffusive approximation is used for the turbulent perpendicular fluxes of particles, momentum and energy with diffusion coefficients \( D, \nu \) and \( \chi_{e,i} \) respectively; furthermore, a term \( \nu_{\text{pinch}} \) is added to model a possible convective contribution to the turbulent flux.

Neo-classical diffusion is usually negligible in scrape-off layer plasmas and the perpendicular transport is dominated by turbulence; for this reason, the respective diffusion coefficients have to be provided as input to the code from the user, usually taken to match experiments. Another possibility would also be to derive effective diffusion coefficients from "first principles" turbulent codes like TOKAM3X [58]. These diffusion coefficients can furthermore be set to be constant in both the radial and poloidal direction or they can be imposed as radial profiles, as shown in figure 1.11; furthermore a poloidal dependence can be added to mimic the presence of a ballooned transport near the outer (low field side) midplane, as seen in experiments [32].

Due to the fact that the plasma motion is preferentially along magnetic field lines, usually all edge transport codes (Soledge2D included) tend to be solved on numerical meshes with elements’ edges aligned along the magnetic field lines, even if some efforts have been made to write codes able to deal with non-field aligned elements [30]. While this does not create any difficulty in regions far from solid surfaces or near the strike points
on the targets, keeping the edges aligned both with the magnetic field lines and with the actual solid surface near the first wall proves challenging. For this reason, current edge transport codes tend to limit the grid to only simulate a relatively small portion of the scrape-off layer in the radial direction; Soledge2D, on the other hand is able to extend the numerical mesh to the wall thanks to the implementation of the Immersed Boundaries technique [37, 46, 8]: the grid is extended into the solid wall so that the grid elements do not have to be aligned with the wall itself, and the fluid equations are rewritten so that the boundary conditions are recovered at the wall (by means of a mask function). This allows Soledge2D to perform simulations with the full geometry up to the wall, as shown in figure 1.12.

A number of conditions are available for the domain boundaries:
Figure 1.12: Numerical grid for Soledge2D in JET geometry. Red edges are for the mesh elements inside the plasma domain, while black ones are outside (penalized zones). The wall profile is highlighted in blue, and the boundary between the edge and the core (which is not simulated) is plotted in green.
• at the walls, the Bohm boundary condition forces (at least) sonic parallel ion velocity $M \geq 1$ at the sheath entrance, and the heat flux is computed using sheath coefficients specified as input parameters;

• at the core-edge interface, a no slip condition is set for the velocity, while either Dirichlet or Neumann conditions can be selected for the continuity equation (so either a density value or a particle flux) and for the energy equation for each species (so temperature value or power crossing the surface).

From the numerical point of view, the code has been written with the finite volumes method, applied on quadrilateral elements; the $(R,Z)$ mesh is mapped to a rectangular $(r,\theta)$ grid and divided in sub-domains in order to speed the code up through openMP parallelization. A second order mixed implicit-explicit scheme is used for the time discretization [7]

$$X^{n+1} = X^n + \left[ \left( \frac{3}{2} \varepsilon^n - \frac{1}{2} \varepsilon^{n-1} \right) + \frac{1}{2} (D^{n+1} + D^n) + P^{n+1} \right] \Delta t \quad (1.50)$$

where the evolution of a variable $X$ at the next time step $n+1$ is computed with a first term that gives the Adams-Bashforth scheme for explicit terms $\varepsilon$ (parallel advection and perpendicular transport, which are slowly varying), a second term for the Crank-Nicolson scheme for the heat transport terms $D$ and a third term for the first order implicit Euler scheme for the penalisation terms $P$. Because of this mixed implicit-explicit time discretization, the maximum time step of the simulations is set by the code itself in function of a CFL condition; usual time steps values are around $10^{-9} \div 10^{-8}s$.

### 1.4.2 Eirene

As we already explained solving the plasma equations is not enough for a detailed model of the scrape-off layer, as plasma-neutrals interactions produce source terms in the equations. In order to evaluate these source terms, the Eirene code [50] is often used in the European community, even if other similar codes are present [18, 33, 64].

Eirene is a kinetic code for neutral particles, developed by D. Reiter and colleagues, that solves the Boltzmann equation for an arbitrary number of species of atoms and molecules.

As shown before, a kinetic description is often required for the neutral gas in most of the volume, and the fact that the isotropy of the transport of neutrals is not broken by

---

1Some molecular ions (like $D^+_2$) are also taken into account in Eirene, but they are not transported from the position where they are born.
the presence of the magnetic field, the full six dimensional behavior has to be retained. This makes it computationally costly to solve the Boltzmann equation for the neutral species by means of numerical method like finite volumes, finite elements, etc, and in turn, linked to the high number of atomic, molecular and surface processes that neutral particles undergo, makes a Monte Carlo procedure appealing. In fact, a Monte Carlo code is not required to explicitly compute the underlying six dimensional distribution function, and can then estimate the moments of the latter on a "simple" three dimensional grid.

The underlying idea of a Monte Carlo procedure is to build a stochastic process whose average is the solution of the governing equations we want to solve. Specifically, we want to solve the Boltzmann equation 1.44 (and similar equations for other neutral species) under the assumption that the history that each particle undergoes can be considered as a Markov chain, that is the probability of an event at each instant of time only depends on the current state and not on what happened previously: the particles do not keep memory of their histories.

In this work we will only on analog sampling in Eirene, in which the event undergone by the particles during their histories mimic the underlying physics, in contrast to non-analog techniques in which variance-reducing events (Russian roulette, splitting, etc...) are artificially introduced to reduce the computational time for a given precision.

Neutral particles are thus launched from the wall (to simulate recycling) and from the domain volume (in case of volumetric recombination, or molecular reactions like dissociation), with their position, momentum and energy randomly sampled from an initial distribution that depends on the plasma background.

The particles are then let freely stream for a distance randomly sampled from an exponential free path distribution, where another sampling is carried out to decide which collision the particles experience, depending on a collision kernel $C(\vec{r}, \vec{v}_0 \rightarrow \vec{v}_1)$. After the collision, a new value for momentum and energy is sampled from the particle distribution after the collision. This process is repeated until the particle is killed, either by a reaction that changes the particle species like an atom ionizing for example, or by an absorption by a surface; the chain of events that a single particle undergoes from its initial launch to its "death" is a single history or chain. In figure 1.13, for example, are shown some of the particle histories followed by Eirene during a WEST simulation. During its history, each particle will contribute to the estimation of a set of quantities of interest or "estimators".
Figure 1.13: Representation of some of the histories computed by Eirene during a Soledge2D simulation in WEST geometry. Red lines show the trajectories of D atoms and in blue $D_2$ molecules. The different markers represent the collisions experienced by the particles: squares indicate electron impact reactions, while three-pointed markers indicate charge exchange collisions.

An example is the track-length estimator, which tallies information $X_g$ about a detector function $g$ along the length $s$ of the trajectory between successive collisions

$$X_g = \sum_{l=1}^{N_t} s_l g_l$$

where the summation is done over the total number of free flights performed by the particle during its history.

As an example, if we are interested in the neutral density in a given grid element $e$, the detector function can be shown to be proportional to the residence time in the element so that the track length estimator for a single history $\omega_h$ is then $n_{n,e}^h = X_{n,e}^h(\omega_h) = \frac{1}{V_e} \sum_{t=1}^{N_t} \frac{w_t s_t}{v_t}$, where $w$ is the statistical weight of the particle, $v$ its velocity, $V_e$ the volume of the element and we have restricted the summation to only the trajectories crossing the element $e$.

The idea of the Monte Carlo code is that by following a high enough number $N$ of histories $\omega_{i=1,\ldots,N}$, the statistical average of the estimators $X_g$ provides an approximation $\tilde{R}$ of some quantities of interest $R$, that in our case are the sources for the plasma
Chapter 1. Power exhaust in Tokamaks

equations.

\[ R \simeq \hat{R} = \frac{1}{N} \sum_{i=1}^{N} X_{g}(\omega_{i}) \]  

(1.52)

At the same time, the successive moment of the probability distribution function, the statistical variance \( \sigma^{2} \), gives us information on how good the approximation of the tally is

\[ \sigma^{2} \simeq \hat{\sigma}^{2} = \frac{1}{N-1} \sum_{i=1}^{N} (X_{i} - \bar{X})^{2} \]  

(1.53)

where the subscript 1 indicates the variance per single history, and \( \bar{X} \) is the average of the estimator over all histories. An estimation for the variance for the whole Monte Carlo code for a high enough number of histories is then obtained using the law of large numbers, so that over N histories it will converge to a constant value given by

\[ \sigma^{2} = \frac{1}{N} \sigma_{1}^{2} \]  

(1.54)

It is obvious that the standard deviation of the method \( \sigma \) scales only with the square root of the number of histories, i.e. to reduce the standard deviation by 10 a hundred-fold increase of the number of histories is needed. At the same time, though, it has to be remembered that the computational cost of the code scales linearly with the number of particles, so that the overall efficiency of the Monte Carlo code is mostly independent of the number of histories.

A compromise between the computational time required by the simulations and the standard deviation allowed on the solution has then to be found: as an example, during Soledge2D-Eirene simulations \( 10^{5} \div 10^{6} \) particles are often used.

On the other hand, the computational time of the Monte Carlo code does not depend only on the number of histories, but also on their length.

The shortest history possible for a particle, in fact, is one in which the particle is killed by an electron impact reaction or absorbed as soon as it is born. On the other hand, as we can see from figure 1.14, atoms for example experience on average a high number of charge exchange collisions before getting ionized in cold plasmas (characteristic for examples of detached regimes). For a 1eV plasma, a simple estimate given by the ratio of the reaction rates of charge exchange and ionization tells us that an atom, on average, may undergo \( 10,000 \div 1,000,000 \) collisions with ions before getting ionized, and thus Eirene will have to follow the particle during the whole time, effectively slowing down the coupled plasma-neutral code simulation. This is true assuming that the atom is not
able to escape the highly-collisional region, but if the mean free path of the neutral is much smaller than the size of the latter this might prove to be the case.

Clearly this proves a major problem for the tokamaks of next generation: given the larger dimensions with respect to existing devices, we can expect the volume of the highly collisional regions in detached regimes to also increase so that, if we assume the neutrals to have similar mean free paths, the residence time of atoms and molecules in these highly collisional regions, and consequently the computational time of the Monte Carlo code, to increase as well.

![Graph](image)

Figure 1.14: Average number of charge exchange collisions undergone by an atom before ionization as a function of electron temperature, calculated as the ratio between the reaction rates of the two collisions. Different color lines are relative to the different values of plasma density reported in the legend.

### 1.5 Importance of neutral particles

Having introduced the physics of the plasma and neutral particles in the scrape-off layer of Tokamaks and having introduced the usual tools used for their modelling, let us now apply the Soledge2D-code to a realistic JET case in the three divertor regimes introduced in the previous sections.

As we saw in the two-point model, we can estimate the value of any of the four variables $n_{up}$, $T_{up}$, $n_t$ and $T_t$ knowing one of them and the value of the heat flux $q$ and of the connection length $L$.

The usual choice is to express the temperatures (upstream and target) and the target density in function of the upstream density, as together with the input power it is the usual control parameter available for Tokamak operation; as such we will perform an upstream density scan.
The upstream density was set at the separatrix at values of \([1, 1.5, 2, 2.5, 3] \cdot 10^{19} m^{-3}\) by varying the puff rate of \(D_2\) molecules, while keeping the remaining input parameters the same. In the specific:

- pure deuterium plasma was assumed (no impurities);
- the power crossing the core-edge interface was 6 MW, evenly divided to electrons and ions;
- the diffusion coefficients were constant in both the radial and poloidal direction, set to \(D = 0.5 m^2/s\) and \(\chi_e = \chi_i = 1.5 m^2/s\);
- the material for the divertor plasma facing components was pure tungsten with a recycling coefficient of \(R_W = 1\), while the rest of the first wall is beryllium with a recycling coefficient of \(R_{Be} = 0.99\);
- two pumps were set in the corners of the divertor, between the vertical tile 3 and the horizontal tile 4, and between the horizontal tile 6 and the vertical tile 7, both with an albedo of 0.99;
- the puff was positioned in the private flux region, on tile 4, puffing \(D_2\) molecules at a temperature of 0.03 eV.

Furthermore, to illustrate the role of the geometry (completely absent in the two point model) the same density ramp was performed with a second magnetic configuration, as shown in figure 1.15: in the following we will refer to the more conventional equilibrium with strike points on tiles 3 and 5 as vertical-horizontal (VH), while we will call corner-corner (CC) the second one with strike points closer to the pump entrances on tiles 4 and 6.

The resulting profiles of density and temperature at the outer midplane are shown in figure 1.16.

As we can see, while in the edge the profiles for the two cases are almost identical, some small differences arise in the scrape-off layer for the upstream quantities. In particular, the VH configuration presents a slightly flatter gradient than the CC configuration, and a small "shoulder" seems to start developing in the far scrape-off layer.

The biggest changes, on the other hand, happen in the divertor because of the different geometry seen by the plasma and by the neutrals. In figures 1.17 and 1.18 is shown the heat flux profiles along the wall for the VH and CC configurations, respectively, with a zoom on the strike points.
1.5. Importance of neutral particles

Figure 1.15: Magnetic equilibria used for the density ramps in Soledge2D-Eirene, on the left: the VH configuration in red was taken from shot #82215, while the CC configuration in blue was reconstructed from shot #89088. On the right, enumeration of the divertor tiles used in the ITER-like wall divertor in JET [59].

Figure 1.16: Radial profile at the outer midplane for plasma density for the vertical-horizontal configuration on the left and for the corner-corner configuration on the right.

Figure 1.17: Heat flux profile along the wall for the VH configuration. The colors represent the simulations with different upstream value of the plasma density.
Figure 1.18: Heat flux profile along the wall for the CC configuration. The colors represent the simulations with different upstream value of the plasma density.

The first thing we can notice is the fact that in the CC configuration the heat load is much more asymmetric between inner and outer strike point: when both strike points are still fully attached, so for low upstream density, the peak heat load on the OSP is $4 \div 5$ times the heat received by the ISP, while for the VH configuration the OSP receives only $\simeq 70\%$ more power than the ISP. Nonetheless, in both configurations both strike points eventually seem to detach above a certain value of the upstream density. This is highlighted in figures 1.19 and 1.20 which show the ion flux at the target and the degree of detachment at the separatrix. Here the degree of detachment was calculated similarly to the experimental definition of equation 1.14: the extrapolated value of the ion flux at the target was taken as $\Gamma^{extr} = C n_{up,sep}^2$, where the coefficient $C$ was obtained normalizing the extrapolated value to the "measured" value on the strike point for the lowest density simulation ($1 \cdot 10^{19}m^{-3}$).

For the inner divertor, a clear roll-over of the ion flux is visible for both configurations, but it seems to happen at a lower upstream density for the corner-corner configuration (between 1.5 and $2 \cdot 10^{19}m^{-3}$) than for the vertical-horizontal configuration (between 2 and $2.5 \cdot 10^{19}m^{-3}$). A similar behavior is seen also on the outer divertor: a rollover of the ion flux is evident for the corner-corner configuration, but at a higher density than the ISP, while the ion flux for the VH keeps on increasing with the same slope.

A value of the DOD significantly larger than unity is taken as index of full detachment, and the value of $DOD = 2$ is often used for the onset of the regime [41]. For the CC configuration, both divertor targets achieve a DOD higher than the "threshold" value above an upstream density of $2 \cdot 10^{19}m^{-3}$; for the VH configuration only the inner target for an upstream density of $3 \cdot 10^{19}m^{-3}$ has a $DOD$ significantly larger than 2, while the
1.5. Importance of neutral particles

Figure 1.19: Target ion flux taken at the inner (left) and outer strike point (right) in function of the upstream density, for the VH and CC configurations in red and blue respectively.

Figure 1.20: Degree of detachment taken at the inner (left) and outer strike point (right) in function of the upstream density, for the VH and CC configurations in red and blue respectively.
outer divertor only barely gets to the "threshold" value, indicative of a weak level of detachment.

It is interesting to note that the DOD at both divertor targets is almost perfectly the same between the two different magnetic configurations, at least up to the value of upstream density for which the CC configuration starts to detach. This suggests that the approximations given by the simple two-point model might be valid for these low density regimes, and that geometric effects (not taken into account in the two-point model) might not play a large role in these situations; nonetheless, when we further increase the upstream density and we reach the onset of detachment, the two configurations show qualitatively different behaviors indicative of the now important impact of the geometry, and of its effect on the transport of charged and neutral particles.

Volumetric recombinant is also indicative of the detached regime, and as we saw previously this process becomes significant with respect to its opposite reaction, ionization, when the electron temperature drops to $\approx 1\,\text{eV}$. As we can see from figure 1.21, in both configurations the plasma in contact with the wall for the lowest upstream density is still hot, at almost the same upstream temperature, indicating a sheath limited regime.

As we increase the upstream density, gradients along the separatrix start to develop and the temperature at the target drops: we can see that in the CC configuration, shown in figure 1.23, almost the whole inner leg is already at a temperature below $3\,\text{eV}$ and the plasma detaches from the ISP, in accord with the rollover in the target ion flux; important
1.5. Importance of neutral particles

gradients develop also for the VH configuration on the HFS (shown in figure 1.22), but the drop in temperature is not yet enough for the onset of the detached regime, staying in a conduction limited one; on the OSP, instead, both configurations keep relatively high temperatures. Finally, with the highest upstream density in the scan performed here, the temperature in the whole divertor drops close to 1eV, indicating that both configurations are in a deep detached regime, even if the DOD for the OSP only barely goes beyond the value of 2.

In accord with the electron temperature, figure 1.24 shows the density of the neutral particles (atoms and molecules together).

As we can see, the neutrals flood all the available cold regions of the divertor, and as soon as they reach a region with a temperature higher than $\sim 3eV$ the electrons quickly ionize the neutral particles.

We can see nonetheless that, in the CC configuration, for a given upstream density, a higher level of compression of the neutrals can be reached at the entrances of the pump ducts with respect to the VH configuration.

Figure 1.25 shows in fact the pumped flux from the two pump ducts: for the LFS pump, the CC configuration is able to achieve a flux $2 \div 3$ times higher than the VH configuration in the whole range of upstream densities explored here; for the HFS, instead, the CC configurations allows a $10 \div 20$ times more efficient pumping at low densities, and the ratio quickly drops to 2 for the $3 \cdot 10^{19}m^{-3}$ case.

Interestingly, the pumped rate at both duct openings for the CC configuration shows
Figure 1.23: Electron temperature profile along the separatrix for the CC configuration.

Figure 1.24: Density of atoms and molecules (in logarithmic scale) for the VH (on the bottom) and for the CC (on the top) configurations.
1.5. Importance of neutral particles

Figure 1.25: Pumped fluxes for the HFS (on the left) and LFS (on the right) pump ducts, for the VH and CC configurations in red and blue respectively. The red triangle represents the simulation in VH configuration with higher pump albedo.

A clear roll-over, consistent with the behavior of the ion flux on the wall; in the VH configurations, instead, the pumped flux keeps on increasing even after the ion flux has already reached its maximum value. This is due to the fact that, in the VH configuration, the duct openings are far from the strike points and, more importantly, the geometry of the vertical targets hides the pump openings and makes it difficult for recycled neutrals to directly flow in the pumps.

It should be noted that the pumps play a major role in the plasma-neutral system: changing the albedo of the pumping surfaces, for example, completely changes the particle balance in the computational domain. This is shown in figure 1.26 for example for the density of the atoms obtained by the simulation with an upstream density of $2 \cdot 10^{19} \text{m}^{-3}$ in which we increased the albedo from $R = 0.95$ to $0.99$.

By increasing the albedo (that is the fraction of incident particles reflected back by the surface) we reduce the flux of particles that are pumped out of the system by a factor 5 ($\Gamma_{\text{pumped}} = (1 - R)\Gamma_{\text{incident}}$), consistently to what was shown by the triangle markers in figure 1.25. To counterbalance the now larger fraction of neutrals reflected back by the pumping surfaces, the density of atoms between the outer strike point and the LFS corner increases by almost an order of magnitude.

While a change of a factor 5 in the pumped flux is relatively large and its effects are therefore magnified, this emphasizes nonetheless the need to accurately model the transport of the neutral particles in the divertor, as even small changes might significantly
change the particle balance in the domain.

Let us conclude this section with some consideration on the computational performance of the code in these simulations. Figure 1.27 shows the average time spent in Eirene to compute the particle histories.

At low densities, both configurations require $3 \div 4s$ on average in Eirene, with a scatter given by different statistical noise realizations, but this number increases by a factor $\sim 6$ at the same value of upstream densities for which we find the rollover of the ion flux. This indicates that, as soon as the plasma detaches from the divertor targets, a highly collisional region develops, responsible for the longer particle histories in the Monte Carlo code.

This is also evident from figure 1.28, which shows a two dimensional version of figure 1.14.

The figure, in fact, shows the average number of charge exchange collisions undergone by an atom before ionization, estimated as the ratio between the local reaction rates of charge exchange and ionization.

As we can see at low density the atoms have, on average, a much greater probability of undergoing an electron-impact reaction resulting in their ionization.

Nonetheless, as we increase the upstream density a larger and larger region in front of the inner target (at first) and of the outer target (at even higher upstream density) in which atoms collide on average $10^4 \div 10^5$ more times with ions than with electrons. When this region gets large enough, the histories in Eirene get longer and longer and the fraction ...

\footnote{Note that the small regions near the corners are not actually highly collisional, but rather they show a plasma density below the limit of validity of the fits used to compute the reaction rates, and in particular the reaction rate for ionization drops to zero and the ratio consequently diverges to infinity.}
Figure 1.27: Average time spent in Eirene to compute the particle histories for the VH and CC configurations, in red and blue respectively. The error bars show the time spent for the shortest and longest MPI processes.

Figure 1.28: Local average number of charge exchange collisions undergone by atoms before ionization for the VH (on the bottom) and the CC (on the top) configurations. Note that this does not directly translate to the length of the Monte Carlo histories, as this local estimate does not take into account that the particles can escape from the highly collisional regions.
of time spent by the full plasma-neutrals code into Eirene increases (from $\sim 59\%$ for the lowest upstream density to $\sim 84\%$ for the highest density).

It is clear then that, if we want to simulate highly collisional regimes, a reduction of the computational time required for the neutral particles is an important step towards simulation for larger devices like ITER and DEMO. As already explained, in future tokamaks we can expect to have even larger highly collisional volumes and the length of the histories that we need to follow in Eirene increases.

A straightforward solution to this problem would then be to simply reduce the number of followed histories (for reference, the Soledge2D-Eirene simulations just analyzed were performed with 20000 histories per core, for a total of 64000 histories), and this strategy will be investigated in the next chapter.
Chapter 2

Statistical noise

Contents

2.1 Statistical noise due to Eirene ............................................. 49
2.2 Analogy with turbulent transport .......................................... 54
2.3 Synthetic noise ................................................................. 59
   2.3.1 General setup ......................................................... 59
   2.3.2 Synthetic noise ....................................................... 61
   2.3.3 Results ................................................................. 62
   2.3.4 Implications for edge transport codes ......................... 66
2.4 Effects of statistical noise in realistic Soledge2D-Eirene simula-
   tions ................................................................. 67
2.1 Statistical noise due to Eirene

As we explained in the previous chapter in section 1.4, the codes used to simulate the transport of neutral gas usually rely on a Monte Carlo procedure to solve the kinetic equations for atoms and molecules.

In these codes, as we saw, the quantities of interest (i.e. the plasma sources) are approximated as statistical averages of certain estimators over a large enough number \( N \) of histories. The statistical variance per single history related to this average, that is the variance of the solution for a given quantity if only one history was simulated, is then

\[
\sigma^2_i \simeq \tilde{\sigma}^2_i = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2
\]

If we then simulate a large enough number of histories we can estimate a mean value for the variance of the set of histories, through the law of large numbers, simply as

\[
\sigma^2 = \frac{1}{N} \sigma^2_i \tag{2.2}
\]

Obviously then, to obtain a variance identically null (or rather, of the order of machine precision) an infinite number of Monte Carlo realizations would be needed. Conversely, whenever we compute a finite number of particle histories an error due to statistical noise is introduced in the system, and this error (measured by the variance of the Monte Carlo code) increases linearly when decreasing the number \( N \) of histories.

This can be visually seen in figure 2.1, which shows the neutral density computed by Soledge2D-Eirene in JET simulations similar to ones performed in section 1.5 with a different number of histories followed in the Monte Carlo solver.

As the number of histories computed decreases, the neutral density (and all other outputs) becomes more and more patchy due to the fact that the limited number of particles is not enough to score in the whole computational domain. In particular, less and less trajectories originate from the first wall far from the targets due to the fact that small impacting ion fluxes result in lower probability of locally launching a recycled neutral*. Furthermore, single trajectories start to become more and more visible.

As such, the coupled edge plasma-neutral simulations are usually performed with a relatively high number of histories as a compromise between the computational time.

*Note that this is a consequence of the implementation of the recycling stratum in Soledge2D-Eirene, and could be resolved to an extent by dividing the recycling of particles on the first wall from the recycling at the targets into an additional stratum.
Chapter 2. Statistical noise

Figure 2.1: Neutral density \( m^{-3} \) (in logarithmic scale) tallied by Eirene with 20, 200 and 20000 histories per core (32 cores were used for these simulations), on the left, center and right respectively.

required by Eirene and the expected variance on the solution; as an example, a Soledge2D-Eirene simulation is usually performed with \( 10^5 - 10^6 \) neutral histories in the Monte Carlo solver.

At the same time, it is important to remember that the computational cost of the Monte Carlo simulations scales linearly with the number \( N \) of histories. For the larger devices of next generation (ITER, DEMO) this contributes, together with the appearance of longer and longer histories in highly collisional regions and for example the fact that in larger devices the time scales of the system also increase, to coupled plasma-neutrals simulations that take weeks, if not months, to converge.

It is then obvious that the simplest way to speed-up Monte Carlo simulations would be to just reduce the number \( N \), but that would decrease the precision of the results due to the introduction of larger and larger statistical noise whose effects on the solution have not been documented yet.

Furthermore, assessing whether a simulation is converged or not in presence of statistical noise is not trivial. Figure 2.2 shows the residuals on the plasma density, estimated as the maximum norm \( \| n^t - n^{t-1} \|_\infty \) of the solution at each iteration, for the JET simulations used for figure 2.1; as we can see the residuals do not converge to machine precision, but rather saturate to relatively high level, as also shown in reference [43]. In fact the simulations in presence of statistical noise do not converge to a strict stationary solu-
2.1. Statistical noise due to Eirene

The fluctuations around the converged value are simply due to the fact that a new realization of the statistical noise has been sampled when Eirene has been called, which produces a noisy source term for Soledge2D to which the plasma has to react; intuitively, we can also already imagine that, as we increase the time between Monte Carlo calls, the plasma will have more and more time to react to the noisy source term and can thus converge to a solution dictated by that particular realization before the next Eirene call. Furthermore as we decrease the number of histories simulated in Eirene, and consequently as we increase the statistical noise level, the residuals saturate to a higher value.

Nonetheless, we can extract some information about the statistical noise of the Monte Carlo procedure. Figure 2.3 shows in fact the probability density function of the neutral density due to the statistical noise introduced by Eirene, reconstructed during the JET simulations above with 20 and 20000 particle histories per core followed in the Monte Carlo code, in the vicinity of the ISP target.

Due to the high number of histories, we can see that the pdf at low statistical noise levels is almost perfectly symmetric around its mean value, reducing to a Gaussian pdf as one would expect because of the central limit theorem. On the other hand, computing
Chapter 2. Statistical noise

Figure 2.3: Probability density function \( W(n_n) \) of the neutral density \( n_n [a.u.] \) extracted from Eirene during the simulations with 20 (on the left) and 20000 histories per core (on the right). Red lines represent a Gamma distribution with noise level \( R = 33\% \) and \( R = 0.99\% \) respectively.

A lower number of histories in Eirene increases the statistical noise and skews the pdf more and more.

This should not surprise: at very low number of histories, we expect the probability of a particle to tally in a given spatial point to be quite low and, due to the law of rare events, the number of particles passing through a mesh cell approximately obeys a Poisson distribution; in turn this means that the probability of a mesh cell to receive contribution from a Monte Carlo history closely behaves like an exponential distribution. This is exactly the same case for the load of a call center: the number of calls arriving at the center in an interval of time obeys a Poisson distribution, and inversely the time between calls behaves like an exponential pdf.

It is interesting to note that the sum of \( n \) (with \( n \) integer) independent random variables with the same exponential distribution \( \text{Exp}(x; \lambda) \) results an Erlang distribution, which is a special case of a Gamma distribution \( \text{Gamma}(x; n, \lambda) \) with \( n \) integer; furthermore, a Gamma distribution with \( n \to \infty \) tends to a normal distribution.

The Gamma distribution can be defined as

\[
\text{Gamma}(x; \alpha, \beta) = \frac{x^{\beta - 1}}{\Gamma(\beta)} e^{-x/\alpha} \quad \text{for } x > 0
\]

where \( \Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx \) is the Gamma function, \( \alpha \) is the shape parameter and \( \beta \) is the rate parameter. We can compute the mean of the Gamma distribution as its first moment

\[
\mu = \int_0^\infty x\text{Gamma}(x; \alpha, \beta)dx = \alpha\beta
\]
2.1. Statistical noise due to Eirene

Figure 2.4: Plots of the Gamma distribution for constant rate parameter $\beta$ while changing the shape parameter $\alpha$, on the left, and for constant shape parameter $\alpha$ while changing the rate parameter $\beta$, on the right. The distributions have been normalized to the maximum value to just show the impact of the parameters on the shape and position of the curves.

Similarly we can calculate the variance as its second moment

$$
\sigma^2 = \int_0^{\infty} x^2 \Gamma(x; \alpha, \beta) dx - \mu^2 = \alpha^2 \beta
$$

and the skewness as its third moment

$$
\gamma = \frac{\int_0^{\infty} x^3 \Gamma(x; \alpha, \beta) dx - 3\mu \sigma^2 - \mu^3}{\sigma^3} = \frac{2}{\sqrt{\beta}}
$$

Finally, the noise level can be defined as the ratio between the standard deviation of the neutral density and its mean $R = \frac{\sigma}{\mu} = \frac{\alpha \sqrt{\beta}}{\alpha \beta} = \beta^{-1/2}$, and thus $\beta = \frac{1}{R^2}$.

To see the effect of the two parameters $\alpha$ and $\beta$, the Gamma distribution is shown in figure 2.4. As it can be seen, as the parameter $\beta$ increases the skewness of the distribution decreases and the latter gets more and more symmetric around the mean and it gets closer and closer to a Gaussian distribution; on the other hand, for $\beta = 1$, the distribution degenerates to an exponential distribution. Furthermore, we can also note that a Gamma distribution with semi-integer shape and rate parameters $\Gamma(x; \frac{\alpha}{2}, \frac{1}{2})$ is identical to a more common $\chi^2(x; a)$ chi-squared distribution.

As we can see from figure 2.3, the behavior of the Gamma distribution closely resembles the statistical noise introduced in real Soledge2D-Eirene simulations by the Monte Carlo code. In fact, the Gamma distribution represented in the figure is not a fit of the
histogram, but it has been parametrized with the moments of the time series. This, as it will be explained later, will make it possible to synthetically reproduce the statistical noise due to Eirene with a Gamma distribution.

Finally it has to be noted that, for low numbers of histories in Eirene, we saw that the pdf of the neutral density does not resemble a Gaussian distribution. This has important consequences on the estimation of the accuracy of the Monte Carlo code, as for example intervals of confidence are defined for Gaussian distributions only.

2.2 Analogy with turbulent transport

As we already introduced, the plasma-neutral system of equations in presence of statistical noise converges to a SSS, for which each run of the Monte Carlo code with a different random seed can be viewed as a single realization of a stochastic process forced by multiplicative noise.

What is of interest then is the statistical properties of the stochastic process, that is in terms of the moments $\mu$, $\sigma^2$, $\gamma$, ... of the plasma fields (density, fluid velocity, temperature, ...) over the pdf of the underlying statistical noise.

Instead of the symbols $\mu$ and $\sigma$ in the following we will use the notation $\langle x \rangle$ and $\langle \delta x \rangle$ for each stochastic variable $x$.

As continuation of the analogy between statistical noise and turbulence, we can then write the noisy neutral fields resulting from an Eirene call as the sum of a mean part $\langle \cdot \rangle$ and of a fluctuating correction $\delta \cdot$ around the mean

\[
\begin{align*}
n_n &= \langle n_n \rangle + \delta n_n \quad (2.7) \\
n_m &= \langle n_m \rangle + \delta n_m \quad (2.8) \\
\bar{u}_n &= \langle \bar{u}_n \rangle + \delta u_n \quad (2.9) \\
\bar{u}_m &= \langle \bar{u}_m \rangle + \delta u_m \quad (2.10) \\
T_n &= \langle T_n \rangle + \delta T_n \quad (2.11) \\
T_m &= \langle T_m \rangle + \delta T_m \quad (2.12)
\end{align*}
\]

where the mean is with respect to various realizations of the statistical noise and the fluctuating corrections are the forcing term produced by the statistical noise.

Because of the coupling of Eirene to Soledge2D through the sources due to interactions of plasma particles with neutrals, the statistical noise then propagates to the plasma fields
The SSS is reached when the statistical properties of the various plasma quantities are time independent, and in particular the average values of the latter are constant in time \( \frac{\partial \langle \cdot \rangle}{\partial t} = 0 \). The equations that rule the SSS can thus be obtained substituting the full noisy fields given by equations 2.7-2.16 in the system of equations 1.46-1.49, and averaging all the terms over the noise realizations.

Some important properties of the average, which will be used extensively in the following, are:

- the average of the fluctuating part of the fields is null \( \langle \delta \cdot \rangle = 0 \) by definition;
- the average part of a field is not affected by the average operator \( \langle \langle \cdot \rangle \rangle = \langle \cdot \rangle \);
- it follows from the previous two properties that the average of the product of the average part of a field and of the fluctuating part of another field is null \( \langle \langle \cdot \rangle \delta \times \rangle = \langle \cdot \rangle \langle \delta \times \rangle = 0 \).

Of course, differential operators and constants (e.g. \( \vec{v}_{\text{pinch}} \), \( D \), \( \nu \), \( \chi_e \) and \( \chi_i \)) are not affected by the average operator; the system of equations for the mean fields can then be rewritten as

\[
\vec{\nabla} \cdot \left( \langle n_e \rangle \langle u_i \rangle \vec{b} + \langle n_e \rangle \vec{v}_{\text{pinch}} - D \vec{v}_{\perp} \langle n_e \rangle \right) + \vec{\nabla} \cdot \left( \langle \delta(n_e u_i) \rangle \vec{b}^b \right) = \langle S_n \rangle \tag{2.17}
\]

\[
\vec{\nabla} \cdot \left( \langle n_e \rangle \langle u_i \rangle^2 \vec{b}^b + \langle n_e \rangle \langle u_i \rangle \vec{v}_{\text{pinch}} - \langle u_i \rangle D \vec{v}_{\perp} \langle n_e \rangle \right) + \\
+ \vec{\nabla} \cdot \left( \langle \delta(n_e u_i^2) \rangle \vec{b}^b + \langle \delta(n_e u_i) \rangle \vec{v}_{\text{pinch}} - \langle \delta(u_i D \vec{v}_{\perp} n_e) \rangle \right) = -\nabla_{||} \frac{\langle n_e \rangle \langle T_i \rangle}{m} + \\
+ \frac{e\langle n_e \rangle E_{||}}{m} + \vec{\nabla}_{\perp} \cdot \left( \nu \langle n_e \rangle \vec{V}_{\perp} \langle u_i \rangle \right) + \nabla_{||} \frac{\langle \delta(n_e T_i) \rangle}{m} + \vec{\nabla} \cdot \left( \nu \langle \delta(n_e \vec{V}_{\perp} u_i) \rangle \right) + \frac{\langle R_{el} \rangle}{m} + \langle S_{\text{mom}} \rangle \tag{2.18}
\]
relative to the statistical uctuating part of the elds, which we can generally represent quantities instead of the mean values.

In particular, the SSS also allo ws us to calculate the standard deviation of the solution, the plasma and neutrals quan tities at the last time step as the solution of the problem.

This means that, in presence of statistical noise, the solution of the system of equations is energy due to only the mean part of the elds.

Furthermore, we see that new terms (highlighted in red) appear in the equations relative to the statistical fluctuating part of the fields, which we can generally represent as $\langle \delta X \rangle$. These terms are generally not zero due to the fact that the terms $X$ are given by products of two or more fluctuating fields, and their mean is thus not null. As an
example, in the continuity equation 2.17, a term $\langle \delta(n_e u_i) \rangle = \langle n_e u_i \rangle - \langle n_e \rangle \langle u_i \rangle = \langle \delta n_e \delta u_i \rangle$ is introduced, which we can recognize as a particle flux resulting from the correlation of the fluctuations of plasma density and ion parallel velocity, again in analogy to the radial turbulent particle fluxes arising in the scrape-off layer [27]; the only difference, here, is that the fluctuations are driven by statistical noise in our case, and by plasma instabilities in the second case.

Beside these, we also see the averages of the source terms due to the neutrals, which are exactly the terms that force the statistical noise in these equations. If for simplicity we only consider atomic reactions, the averages of the source terms can be written as

$$
\langle S_n \rangle = \langle n_e \rangle \langle n_n \rangle iz \langle \langle n_e \rangle, \langle T_e \rangle \rangle + \langle \delta(n_e n_n iz) \rangle - \langle n_e \rangle^2 rc \langle \langle n_e \rangle, \langle T_e \rangle \rangle - \langle \delta(n_e^2 rc) \rangle
$$

$$
\langle S_{mom} \rangle = \langle n_e \rangle \langle n_n \rangle \langle u_n \rangle iz \langle \langle n_e \rangle, \langle T_e \rangle \rangle + \langle \delta(n_e n_n iz u_n) \rangle - \langle n_e \rangle^2 \langle u_i \rangle rc \langle \langle n_e \rangle, \langle T_e \rangle \rangle - \langle \delta(n_e^2 u_i rc) \rangle + \langle n_e \rangle \langle n_n \rangle \langle (u_n - \langle u_i \rangle) cx \langle \langle n_e \rangle, \langle T_i \rangle \rangle + \langle \delta(n_e n_n (u_n - u_i) cx) \rangle
$$

$$
\langle S_{E,i} \rangle = \langle n_e \rangle \langle n_n \rangle iz E \langle \langle n_e \rangle, \langle T_e \rangle \rangle + \langle \delta(n_e n_n iz E) \rangle - \langle n_e \rangle^2 rc E \langle \langle n_e \rangle, \langle T_e \rangle \rangle - \langle \delta(n_e^2 E_i rc) \rangle
$$

$$
\langle S_{E,i} \rangle = \langle n_e \rangle \langle n_n \rangle \langle E_n \rangle iz \langle \langle n_e \rangle, \langle T_e \rangle \rangle + \langle \delta(n_e n_n E_n iz) \rangle - \langle n_e \rangle^2 \langle E_i \rangle rc \langle \langle n_e \rangle, \langle T_e \rangle \rangle - \langle \delta(n_e^2 E_i rc) \rangle + \langle n_e \rangle \langle n_n \rangle \langle (E_n - \langle E_i \rangle) cx \langle \langle n_e \rangle, \langle T_i \rangle \rangle + \langle \delta(n_e n_n (E_n - E_i) cx) \rangle
$$

where we temporarily switched the symbols used for the effective reaction rates $\langle \sigma v \rangle_{iz}$, $\langle \sigma v \rangle_{rc}$, $\langle \sigma v \Delta E \rangle_{iz}$ and $\langle \sigma v \Delta E \rangle_{rc}$ to $iz$, $rc$, $cx$, $izE$ and $rcE$ respectively to avoid using too many angled brackets\(^\dagger\).

Note, nonetheless, that these reaction rates are strongly non-linear functions of the electron density and electron/ion temperature. As such, the full noisy behavior of these fields should, in principle, be retained when calculating the rates, and not only the mean part. As an example, $\langle iz \langle n_e, T_e \rangle \rangle \neq iz \langle \langle n_e \rangle, \langle T_e \rangle \rangle$.

We could thus see the equations solved by a code like Soledge2D-Eirene as introduced in equations 1.46-1.49 as the limit of the new set of equations 2.17-2.20 for the statistical noise tending to zero, or as the number of histories computed in the Monte Carlo code tends to infinity. In fact, due to the fact that the fluctuating terms highlighted in red in the equations above are given by the product of the fluctuating part of the fields, as the number of histories increases the variance of the solution decreases, and the terms

\(^\dagger\)Note that the brackets used for the reaction rates $\langle \sigma v \rangle$ do not have the same meaning of the statistical mean defined in this chapter, but rather indicate an averaging over a Maxwellian distribution function for the projectile particles, as explained in the previous chapter.
become smaller and smaller.

Nonetheless, because of the introduction of these terms the solution of the original equations 1.46-1.49 might be different from the equations of the new noisy equations and the code might converge, in the presence of statistical noise, to a solution different from the one given by the simple noise-free equations.

As explained, the usual procedure of taking the results obtained at the last time step of a simulation once it has reached convergence as the solution of the equations might be misleading, and an averaging procedure of the solution $X$ over a large enough number $N_r$ of noise realizations may be needed

$$\langle X \rangle \simeq \frac{1}{N_r} \sum_{i=1}^{N_r} X_i$$  \hspace{1cm} (2.25)

In particular, a single plasma-neutral simulation with the same random seed would only amount to a single realization of the stochastic process, and performing such an average would mean performing a large number of full simulations with different seeds in the random numbers generator\(^1\).

Fortunately, the ergodic theorem [25] comes to help: for a stationary stochastic variable $x$ (that is, a stochastic variable for which the mean $\langle x \rangle$ is invariant under a translation in time), its time average converges to its mean value for an averaging window that tends to infinity

$$\lim_{T \to \infty} \frac{1}{T} \int_0^T x(t) dt = \langle x \rangle$$  \hspace{1cm} (2.26)

as long as the time correlation function of the stochastic process tends to zero for infinite time translations $\lim_{|t-t'| \to \infty} C_x(t-t') = 0$. The convergence of the time average to the probability average is to be considered in the mean square sense, that is

$$\langle \left( \frac{1}{T} \int_0^T x(t) dt - \langle x \rangle \right)^2 \rangle \sim \frac{\tau_C}{T}$$  \hspace{1cm} (2.27)

for a time $T$ much larger than the correlation time $\tau_C$ of the stochastic process. This means that we can estimate the average of the solution with a single simulation just by letting the simulation run for a large enough number of iteration (enough to be much longer than the correlation time of the statistical noise) after having reached the SSS, and then averaging over the time window thus created.

Furthermore, the SSS gives us also the possibility to estimate all the other moments

\(^1\)Note that, in Soledge2D-Eirene, the usual practice is to use a single seed in the random number generator throughout the simulation in order to completely remove correlations in successive calls of the Monte Carlo code.
of interest, and in particular the spurious terms introduced by the statistical noise, thus assessing their relative weight in the equations as a function of the noise level.

2.3 Synthetic noise

2.3.1 General setup

In order to assess the importance of the noise-induced terms (in red in equations 2.17-2.20), we will use the Soledge2D code coupled to reduced model for the neutral species since we are not, for the moment, interested in a physically precise solution of the full plasma-neutral system. As such, we will use a simple diffusive model

\[ \frac{\partial n_n}{\partial t} - D_n \nabla^2 n_n = -n_e n_n \langle \sigma v \rangle_{iz} + n_e^2 \langle \sigma v \rangle_{rc} \tag{2.28} \]

This approach allows one to obtain the solution of the equations in the case without statistical noise (called noise-free in the following), but also to introduce synthetic noise in the neutral equation in order to mimic the statistical noise due to Eirene.

For simplicity, this analysis will be performed in slab geometry so that we can remove all geometric effects from the discussion. The geometrical and magnetic parameters used are shown in table 2.1.

<table>
<thead>
<tr>
<th>$R [m]$</th>
<th>$r [m]$</th>
<th>$q$</th>
<th>$B_T [T]$</th>
<th>$B_\theta [T]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.5</td>
<td>4</td>
<td>2</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 2.1: Geometrical properties of the slab simulation in terms of major radius $R$, minor radius $r$ and safety factor $q$. The values of the toroidal $B_T$ and poloidal $B_\theta$ magnetic fields are also shown.

The geometry of the slab is shown in figure 2.5: the horizontal direction mimics the radial extent of the domain, while the vertical one represents the poloidal direction. The $\theta = 0$ and $\theta = 1$ boundaries represent the divertor targets, with scrape-off layer connecting the two extending from the separatrix at $r = 0.5$ to the first wall at $r = 1$. Between $r = 0$ and $r = 0.5$, instead, the domain is divided into two zones: between $\theta = 0.25$ and $\theta = 0.75$ we have the closed field lines region with the boundary at $r = 0$ representing the core-edge interface; the private flux region connected to the two divertor common flux regions extends elsewhere, with "cuts" at the imaginary boundaries $\theta = 0.25$ and $\theta = 0.75$. The cuts are introduced in order to "teleport" particles from one side of the
Figure 2.5: Slab geometry used for the noise simulations. In red is shown the confined plasma region (corresponding to zone 5 on the schematic representation of a tokamak on the right, reprinted from [7]), in green the scrape-off layer (zone 2), in blue the divertor common flux regions (zones 1 and 3) and in cyan the two zones that make up the private flux region (zones 4 and 6). Solid black lines represent the position of solid surfaces, while the black dotted lines represent the position of the "cuts". Arrows have been added to help visualize the particles flows.

private flux region to the other and to impose a periodic boundary condition for the confined plasma in between.

The simulations were run with a density at the core-edge interface set at $10^{19} m^{-3}$ and temperature $T_e = T_i = 100 eV$. The recycling coefficient for the target plates was scanned between 0.9 and 0.99, but we will concentrate in the following on the highest value $R = 0.99$, due to the fact that in a high recycling regime we can expect the neutral density and the sources due to plasma-neutrals interactions to be important; in such a regime we try in fact to maximize the effect of neutral physics and we can thus expect the noise to play a major role by introducing significant fluctuations on the source terms (which we can expect to be large compared to the other terms in the plasma equations). Finally, the transport coefficients used in these simulations were set to $D = 0.6 m^2/s$, $\nu = 0.2 m^2/s$ and $\chi_e = \chi_i = 1.333 m^2/s$ for the charged particles, while $D_n = 10^3 m^2/s$ for the neutrals.

The noise free solution is shown in figure 2.6 in terms of plasma density, ion parallel Mach number, electron temperature and ion temperature.
2.3. Synthetic noise

For the noisy simulations, in order to mimic the statistical noise introduced running Eirene, we will randomly sample the neutral density from a Gamma probability density function, which closely resembles the pdf from the statistical noise by Eirene as already shown in figure 2.3. In particular,

- because of the central limit theorem, when the noise level $R$ is low (when we use a large number of histories in Eirene) the Gamma distribution tends to a Gaussian distribution;

- on the other hand for $\beta = 1$, that is when $R = 100\%$, the distribution degenerates to an exponential distribution, similarly to what happens when we use a small number of histories in Eirene and particles are not be able to score in the whole domain;

- finally, the Gamma distribution is defined for positive values of the stochastic variable, in accord with the fact that the neutral density has the physical meaning of number of particles present in a volume and consequently cannot be negative.

In order to avoid introducing biases in the solution, we can impose that the mean value of the Gamma distribution respects the solution of equation 2.28 $\mu = \alpha \beta = \langle n_n \rangle$; because in the following we will use the noise level as the main study parameter, we can thus write the shape parameter in function of the mean and of the noise level $\alpha = \frac{\mu}{\beta} = \langle n_n \rangle R^2$. 

Figure 2.6: Plasma density $[m^{-3}]$ (top left), ion parallel Mach number (top right), electron temperature $[eV]$ (bottom left) and ion temperature $[eV]$ (bottom right) obtained for the noise free simulation.

2.3.2 Synthetic noise

For the noisy simulations, in order to mimic the statistical noise introduced running Eirene, we will randomly sample the neutral density from a Gamma probability density function, which closely resembles the pdf from the statistical noise by Eirene as already shown in figure 2.3. In particular,

- because of the central limit theorem, when the noise level $R$ is low (when we use a large number of histories in Eirene) the Gamma distribution tends to a Gaussian distribution;

- on the other hand for $\beta = 1$, that is when $R = 100\%$, the distribution degenerates to an exponential distribution, similarly to what happens when we use a small number of histories in Eirene and particles are not be able to score in the whole domain;

- finally, the Gamma distribution is defined for positive values of the stochastic variable, in accord with the fact that the neutral density has the physical meaning of number of particles present in a volume and consequently cannot be negative.

In order to avoid introducing biases in the solution, we can impose that the mean value of the Gamma distribution respects the solution of equation 2.28 $\mu = \alpha \beta = \langle n_n \rangle$; because in the following we will use the noise level as the main study parameter, we can thus write the shape parameter in function of the mean and of the noise level $\alpha = \frac{\mu}{\beta} = \langle n_n \rangle R^2$. 

Figure 2.6: Plasma density $[m^{-3}]$ (top left), ion parallel Mach number (top right), electron temperature $[eV]$ (bottom left) and ion temperature $[eV]$ (bottom right) obtained for the noise free simulation.

2.3.2 Synthetic noise

For the noisy simulations, in order to mimic the statistical noise introduced running Eirene, we will randomly sample the neutral density from a Gamma probability density function, which closely resembles the pdf from the statistical noise by Eirene as already shown in figure 2.3. In particular,

- because of the central limit theorem, when the noise level $R$ is low (when we use a large number of histories in Eirene) the Gamma distribution tends to a Gaussian distribution;

- on the other hand for $\beta = 1$, that is when $R = 100\%$, the distribution degenerates to an exponential distribution, similarly to what happens when we use a small number of histories in Eirene and particles are not be able to score in the whole domain;

- finally, the Gamma distribution is defined for positive values of the stochastic variable, in accord with the fact that the neutral density has the physical meaning of number of particles present in a volume and consequently cannot be negative.

In order to avoid introducing biases in the solution, we can impose that the mean value of the Gamma distribution respects the solution of equation 2.28 $\mu = \alpha \beta = \langle n_n \rangle$; because in the following we will use the noise level as the main study parameter, we can thus write the shape parameter in function of the mean and of the noise level $\alpha = \frac{\mu}{\beta} = \langle n_n \rangle R^2$. 

Figure 2.6: Plasma density $[m^{-3}]$ (top left), ion parallel Mach number (top right), electron temperature $[eV]$ (bottom left) and ion temperature $[eV]$ (bottom right) obtained for the noise free simulation.
We will perform different "noisy" simulations with various noise levels \( R \) to simulate running Eirene with a different number of histories (higher \( R \) compares to lower number of histories).

Time correlations in the noise can be introduced freezing the noise for a number of plasma iterations, i.e. not re-sampling the neutral at every plasma iteration, to reproduce the behavior of Soledge2D-Eirene with its short cycling procedure; spatial correlations, instead, are not introduced, so that the noise level \( R \) is kept constant in the whole computational domain.

Once the noisy simulations have reached statistical steady state, they are kept running for a time window \( T >> \tau_C \) much longer than the correlation time of the neutral density fluctuations (measured here as the number of iterations over which the fluctuations are frozen), and an average and respective standard deviation of all the spurious terms highlighted in red in equations 2.17-2.20 and of the solution are tallied during this time window.

As such, we now have:

- the solution in terms of \( n_e, T_e, T_i, \ldots \) for the case without noise (the respective fields in the following will have a subscript \( NF \) for "noise free");
- the solution at the last time step for the noisy simulations (indicated by the subscript \( lts \)), which is usually taken as the solution of the equations;
- the estimates of the average solution \( \langle n_e \rangle, \langle T_e \rangle, \langle T_i \rangle, \ldots \), with its standard deviation \( \delta n_e, \delta T_e, \delta T_i, \ldots \) describing the dispersion of the solution between time steps;
- the estimates of the average \( \langle X(n_e, T_e, T_i, u_i) \rangle \) and the standard deviation \( \delta X(n_e, T_e, T_i, u_i) \) of the spurious noisy terms.

### 2.3.3 Results

The system of equations proves to be incredibly robust to the noise [43], with still small deviations of the noisy simulations with respect to the noise-free one even at extremely high noise levels when small time correlations are introduced.

As in a realistic case where we decrease the number of histories followed in Eirene, the neutral density taken at the last time of simulations with increasing synthetic noise level becomes more and more patchy, as evident from figure 2.7 for two simulations with noise levels of 90% and 400%.

On the other hand, the other main actor that plays a role had been identified as the correlation time of the neutral density fluctuations. In this analysis, we will then focus on
2.3. Synthetic noise

Figure 2.7: Neutral density [a.u] on the lower target at the last time step for a noise level of $R = 90\%$ (on the left) and $R = 400\%$ (on the right), reprinted from reference [44].

the most extreme case in terms of noise level ($R = 400\%$) while varying the correlation time in the range $[1, 100, 1000]\Delta t$ (with $\Delta t$ the time step of the plasma solver Soledge2D, usually $10^{-9} \div 10^{-8}$s); as explained before, this amounts to re-sample the neutral density from the chosen Gamma distribution every $[1, 100, 1000]$ plasma iterations. Also note that, as the correlation time of the noise fluctuations changes, we have to change the length of the averaging window at the end of the simulations accordingly.

For a given noise level, the correlation time does not play any role on the amplitude of the fluctuations of the neutral density during a singular realization of the synthetic noise, but we can see that it has an impact on the average neutral density tallied over the averaging window.

Just by averaging the neutral density at the end of the simulations, in fact, we are able to collect information and converge to a solution similar to the one we obtain without synthetic noise, even at the high noise level used here ($400\%$). Increasing the time correlation, on the other hand, produces larger and larger fluctuating structures as the (average) solution deviates more and more from the noise-free one.

A similar behavior can be found for the plasma solution itself: in figures 2.8 and 2.9 are represented for example the average values of plasma density and the electron temperature along the separatrix $r = 0.5$ (for better clarity, only half of the separatrix
Figure 2.8: Average plasma density $\langle n_e \rangle [m^{-3}]$ along the separatrix for the noise-free case in red with circle markers, and for the noisy simulations with $\tau_C = \Delta t$, $100\Delta t$ and $1000\Delta t$ in black, blue and green respectively.

Figure 2.9: Average electron temperature $\langle T_e \rangle [eV]$ along the separatrix for the noise-free case in red with circle markers, and for the noisy simulations with $\tau_C = \Delta t$, $100\Delta t$ and $1000\Delta t$ in black, blue and green respectively.

The parallel extent is shown due to the symmetry of the problem) estimated during the averaging window. As we can see, the solution obtained from the simulation with $\tau_C = \Delta t$ is indistinguishable from the noise-free solution, with a maximum difference of 0.1%; on the other hand, the other two noisy simulations seem to have introduced quite large errors on the solution, with a smaller plasma density at the targets and, consequently, higher temperature.

Furthermore, focusing for example on the case with the largest time correlation $\tau_C = 1000\Delta t$, figure 2.10 shows that while the average solution is markedly different from the noise-free one, the last time step solution presents a noisy and rough behavior around the average value.

It seems, in fact, that introducing too large time correlations in the statistical noise gives the plasma enough time to react to the fluctuations of the neutral density, producing large excursions in the solution. Conversely, small time correlations seem to filter out the statistical noise: the fluctuations of the neutral density then become too fast for the dynamics of the system, and the latter is then only able to see the average behavior.

This can be seen in figure 2.11 which shows the relative fluctuations level of the plasma density in function of correlation time of the synthetic noise.

Increasing the number of iterations over which the synthetic noise is frozen increases in fact the relative weight of the fluctuating part of the plasma density with respect to its mean value, as the plasma has more and more time to react to the noise realization.

Furthermore, this is reflected in the value of the spurious terms introduced by the noise, as shown in figure 2.12 for the term $\langle \delta(n_e u_i) \rangle$. Due to the plasma fluctuations increasing with the correlation time, the noise induced particle flux gets larger with
2.3. Synthetic noise

Figure 2.10: Plasma density \([m^{-3}]\) profile along the separatrix for \(R = 400\%\) and \(\tau_C = 1000\Delta t\). The solid red lines represent the solution for the case without noise, the blue dotted lines show the average density with error bars representing \(\pm 2\sigma\) and the black dashed line shows the solution at the last iteration.

Figure 2.11: Relative fluctuations level of the plasma density along the separatrix for \(R = 400\%\) and \(\tau_C = \Delta t, 100\Delta t\) and \(1000\Delta t\) in red, blue and green respectively.

Figure 2.12: Relative fluctuations level of the spurious flux \(\langle \delta(n_e u_i) \rangle\) along the separatrix for \(R = 400\%\) and \(\tau_C = \Delta t, 100\Delta t\) and \(1000\Delta t\) in red, blue and green.
respect to the particle flux transported by the mean fields. Note that the relative increase of the fluctuating part for $\theta > 50$ and the discontinuity of the ratio $\frac{\langle \delta(n_{eu}) \rangle}{\langle n_{eu} \rangle}$ at position $\theta = 100$ is only due to the presence of the stagnation point, and the mean parallel velocity being identically zero.

2.3.4 Implications for edge transport codes

It is evident that the analysis performed here does not answer all the questions. In particular, the large number of spurious terms introduced in the equations 2.17-2.20 (which were not explicitly written in order to make the equations clearer) make an in-depth assessment of their relative weights with respect to the noise-free simulation cumbersome. In a more pragmatist way, we nonetheless showed that the main actors at play, which should be monitored during the edge transport simulations, are the noise level and the correlation time of the fluctuations of the neutral quantities. Indeed the spurious noise-induced terms increase in importance when increasing together the two quantities, even if not enough to make us confident of being the cause of the deviations introduced in the solution.

In fact, another possible culprit might actually be the interaction of the statistical noise with the numerical scheme of the plasma solver: at low correlation time, we showed that the plasma is able to filter the fluctuations of the neutral quantities even at large noise levels; on the other hand, when we increase the correlation time the plasma has the time to adapt to the noise realization and could, in fact, produce large gradients. The impact of these large gradients of the plasma quantities on the numerical scheme, by the introduction of possibly important numerical diffusion localized in space and time, is yet to be addressed and will be part of the future work. Nonetheless, we can derive some practical conclusions from this exercise.

- The best approximation of the solution that we can obtain is its time average estimated by paying the cost of running the code for a time much longer than the characteristic time correlations of the statistical fluctuations after having reached the statistical steady state. The usual practice of taking the results of the simulations at the last time step as solution of the problem should be avoided as large biases might be introduced in it when using high noise levels.

- While having this averaging window at the end of the simulations might give the impression to increase the computational cost, we have shown that edge transport codes can be run with large noise levels (i.e. running Eirene with a small number of histories) as long as they are run with relatively small time correlations (i.e. only
2.4. Effects of statistical noise in realistic Soledge2D-Eirene simulations

performing a relatively small number of plasma iterations between Eirene calls) and as long as we average the solution. When the codes are run with high noise levels with large correlations in time, instead, the solution starts to deviate from the "real" solution that would be obtained by running the code in a noise-free fashion.

As such, we could envision a new strategy to run plasma-neutrals simulations: it is possible to speed-up the Monte Carlo solver by using a smaller number of histories, for example during the transient phase of the simulation, and finally running the code in the converged state (in the SSS) for a little longer, while accumulating information on the solution during the averaging window.

Furthermore, we could also think about monitoring the spurious noise-induced terms throughout the simulation and automatically adapt on the fly the number of histories and the number of short cycles to be performed depending on their relative weight.

It should be noted, on the other hand, that this is true for the simplified geometry showed here and how these results extrapolate to realistic cases is still to be assessed.

2.4 Effects of statistical noise in realistic Soledge2D-Eirene simulations

To conclude this chapter, let us try to test whether the conclusions derived from the previous section can be used for simulations in realistic geometry and with the actual statistical noise produced by Eirene.

In this section, we will then use the full Soledge2D-Eirene code in realistic JET geometry. The setup of the simulations is identical to the ones presented in section 1.5, with a focus on the highest upstream density showed therein \((3 \cdot 10^{19} m^{-3})\) as it allowed us to safely obtain a detached plasma for the VH configuration for simplicity. This is, again, to try to maximize the effects of the neutral physics and the impact of the statistical noise on the plasma.

The only parameter varied here is the number of histories followed in Eirene, namely \([20, 200, 20000]\) per core, resulting in \([640, 6400, 640000]\) total histories computed at each call of the Monte Carlo code; constant time correlation was introduced for the three cases by short cycling the code every 100 iterations.

As in the previous JET simulations, the puff rate was automatically adjusted by the code in order to obtain the set upstream density at the separatrix. Furthermore, the same transport coefficients were used, thus resulting in almost identical upstream profiles for density and temperature for the three cases, as shown in figure 2.13.
The results of the Soledge2D-Eirene simulations, in terms of average plasma density $\langle n_e \rangle$ and average electron temperature $\langle T_e \rangle$ are shown in figures 2.14 and 2.15 for the three cases. Averages have been performed during the simulations over a window of $10^6$ Soledge2D iterations, considering a short cycling of Eirene every 100 iterations.

As evident, the three simulations all qualitatively converge to the same solution: a cold detached divertor as we already discussed in section 1.5.

In order to quantitatively assess the agreement of the three cases, figures 2.16 and 2.17 show the parallel profiles of the same quantities along the separatrix. Errorbars have been added to represent the standard deviations of the plasma density and temperature, tallied during the averaging window of the simulations.

As we can see from the figures, by averaging the solution during the simulation once the SSS has been reached, the case for 200 particles only shows small deviations from the one with 20000 particles per core, which we will take as the reference. On the other hand, as we reduce further the number of computed histories in Eirene larger deviations seem to appear, and at the same time the error bars get larger as statistical noise impacts.

---

The cuts in the 2D plots are just due to the fact that the arrays containing the average quantities are generated by Soledge2D on its quadrilateral mesh, which is divided in sub-domains for parallelization.
2.4. Effects of statistical noise in realistic Soledge2D-Eirene simulations

Figure 2.15: Average electron temperature [eV] for the simulations with 20, 200 and 20000 particles per core in Eirene.

Figure 2.16: Average plasma density along the separatrix for the simulations with 20, 200 and 20000 particles per core in Eirene. The subplots on the top right and bottom right represent a zoom on the outer and inner target respectively.
more and more the plasma solution. This is even more evident from figure 2.18 which shows the relative error between the average plasma density obtained by the simulation with 20 and 200 particles with respect to the 20000 case.

For the simulation with 200 particles per core, the difference on the average plasma density are of the order of $\sim 1 \div 5\%$ along both divertor legs, increasing to $\sim 10\%$ in the vicinity of the x-point, and dropping below $\sim 0.1\%$ elsewhere. For the simulation with 20 particles per core, instead, the difference gets for the most part one order of magnitude higher, as one would expect from the ratio of computed histories, with deviations from the 20000 particles case reaching $\sim 100\%$ in the vicinity of the x-point.

Something we should note is that the effect of the noise on the plasma is qualitatively different from the simple slab case presented in the previous section: when we increased the noise level (as shown in reference [43]) or the correlation time in the slab case, the code converged to a solution in which the recycling at the wall was slightly lower resulting in a lower plasma density in front of the targets with respect to the noise-free solution; in this realistic geometry, instead, the plasma density at the target for the case with 20 histories per core converges to a slightly higher value with respect to the 20000 histories per core case, as evident from figure 2.16. This could be explained by the fact that the analysis done here has been performed along the separatrix, while the full two dimensional picture should be analyzed, but more importantly puts into light the fact that the assessment of the effects of the statistical noise require a case-by-case analysis and general conclusions.
2.4. Effects of statistical noise in realistic Soledge2D-Eirene simulations

Figure 2.18: Relative error between the average plasma density along the separatrix for the simulations with 20 and 200 particles per core with respect to the simulation with 20000 histories per core.

on the converged solution (for example, "higher statistical noise results in lower density at the targets") should be avoided.

Furthermore, due to the complexities introduced when performing full Soledge2D-Eirene simulations, additional effects of the statistical noise can be found: as we discussed, the puff rate for the simulations was automatically set by the code in order to obtain the set upstream density at the separatrix of $3 \cdot 10^{19} m^{-3}$; it should be noted, on the other hand, that the three simulations converge to a puff rate slightly different: 3.76, 3.63 and $3.60 \cdot 10^{22} D/s$ for the simulations with 20, 200 and 20000 histories in Eirene respectively. This slight variation might be due to the boundary condition used at the core-edge interface, where an ion flux proportional on the influx of neutrals is set. In fact, in the simulation with 20 particles no neutral is able to reach the core boundary, and the puff has to be slightly stronger to counterbalance the negligible core ion flux; on the other hand, the simulation with 200 particles has a neutral influx at the core four times higher than the case with 20000 particles ($2.5 \cdot 10^{20} D/s$ and $5.8 \cdot 10^{19} D/s$ respectively), which does not explain the slightly higher puff for the former case.

As such, we can assume that the difference between the 200 and 20000 particles is mainly due to statistical noise, as 100 times more histories were available for the puff stratum in the latter simulation.

In accord with the puff rate, also the pumps show a similar behavior with a $\sim 1\%$ higher
pumped rate for the simulation with 200 histories per core with respect to the case with 20000, but $\sim 3\%$ lower than the case with 20 histories per core. On the other hand, the two cases with 20 and 20000 histories per core show a 16 : 84 asymmetry between the HFS and LFS pumps (with the LFS pump receiving most of the flux), while the simulation with 20 particles per core pushes the asymmetry slightly further to 14 : 86.

Nonetheless, even in realistic geometry we can recover the conclusions derived from the slab case: the advantage of the averaging of the solution is in fact evident when looking at the neutrals behavior.

The neutral density taken at the last time step of the simulations has already been presented at the start of this chapter in figure 2.1, showing how for the simulation with 20 particles per core almost no particle was able to score in most of domain, especially in low recycling areas, and single trajectories were visible. The mean neutral density, instead, is shown in figure 2.19.

As evident, when averaging the solution over a large enough number of Eirene calls, even in the case with only 20 particles per core we are able to tally information about the neutral density in most of the domain, with some patches present only in the closed field lines region inside of the separatrix: in this region we might expect then quite large standard deviations of the neutral density, but the mean value of the density is small and consequently its effect on the plasma is negligible. On the other hand, we see that in the divertor region the agreement between the three simulations almost perfect.
As before, the SSS lets us also estimate the spurious terms introduced by the statistical noise in the equations. In figure 2.20 is thus shown the weight of the parallel particle flux driven by fluctuations $\langle \delta(n_e u_i) \rangle$ with respect to the total particle flux $\langle n_e u_i \rangle$.

As we can see we obtain a qualitative behavior similar to what we found in the slab case, with the relative importance of the spurious flux decreasing for lower statistical noise (that is for a larger number of histories followed in Eirene). As before, the peaks are only due to the mean ion parallel velocity nearing zero in the vicinity of the stagnation point at $l_\parallel \simeq 40m$.

Furthermore, in accord with figure 2.18, we see that the large deviations in the mean density fall exactly in the vicinity of the x-point, where the fluctuating contribution to the particle flux is of the same order of magnitude of the total flux. For the rest of the parallel length, we see instead that the fluctuating part of the particle flux only accounts for $1 \div 10\%$ in the most challenging case (20 particles per core), and the contribution drops to $0.01 \div 0.1\%$ for the case with 20000 particles.

To conclude, even in realistic Soledge2D-Eirene simulations we can argue that using a surprisingly low number of histories in the Monte Carlo code (as low as a total of 640 here) while refreshing the statistical noise often enough (we short cycled Eirene for 100 Soledge2D iterations in these cases) we obtain a relatively small error on the mean plasma...
This is in accord with the conclusions drawn in reference [1], which shows that SOLPS simulations in ITER geometry with very few histories (as low as 700) and taking the average solution produce remarkably low biases with respect to simulations with much higher statistics (500000 histories).

This gives us confidence that the new strategy for running edge transport codes can be used for actual simulations: we can accelerate the convergence of the plasma-neutrals system simply by calling the Monte Carlo code with a small number of short cycles between Eirene calls while following a low number of histories in the transient phase of the simulations; once we finally reach the SSS, we can then accumulate information on the quantities of interest by tallying a time average of the solution over a time longer than the correlation time of the statistical noise. As an example, in table 2.2 is shown the average time spent in Eirene to follow the histories for the three simulations explored here, and the relative time spent in Eirene with respect to the total time to perform the whole 100 Soledge2D iterations and the Eirene call.

<table>
<thead>
<tr>
<th>MC histories [hist/core]</th>
<th>Time in Eirene</th>
<th>Eirene/Soledge2D-Eirene</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min [s]</td>
<td>average [s]</td>
</tr>
<tr>
<td>20</td>
<td>0.023</td>
<td>0.035</td>
</tr>
<tr>
<td>200</td>
<td>0.152</td>
<td>0.199</td>
</tr>
<tr>
<td>20000</td>
<td>15.0</td>
<td>18.1</td>
</tr>
</tbody>
</table>

Table 2.2: Time spent by Eirene to follow its histories (for the shortest history, for the longest, and average time) and percentage of time spent in Eirene with respect to the time spent in the whole Soledge2D-Eirene code for the three simulations with 20, 200 and 20000 histories per core.

As we can see, by reducing the number of histories from 20000 to 200 and 20 per core we obtain a speed up of ~ 90 and ~ 500 times respectively.

Note on the other hand that the fraction of time spent in Eirene seems to saturate for the case with 20 histories per core. This is in part due to the presence of particularly long histories (~ 4 times longer than average), but also to the presence of an overhead (which was hidden in simulations run with the "usual strategy" by the large number of histories and their length) in the implementation of Eirene due to, for example, the need

---

*Two additional simulations were also attempted, with even fewer histories (2 per core, for a total of 64 histories) and with longer short cycles (1000 Soledge2D iterations between Eirene calls for the 20 histories per core case), but they crashed shortly after the start, and analysis of these cases is still ongoing.*
to interpolate the plasma quantities from the quadrilateral Soledge2D grid to the triangular grid used by the Monte Carlo code and the computation of the rate coefficients for the neutral-plasma interactions. This overhead could certainly be reduced by optimizing the two codes and the interface for their coupling, possibly further accelerating the convergence (to the SSS) of the edge transport codes whenever we follow a small number of histories in the Monte Carlo procedure.
Chapter 3

Fluid code

Contents

3.1 Introduction ................................................. 77
3.2 Physical model ............................................. 78
3.3 Numerical scheme ................................. 80
   3.3.1 Discretized system of equations .......... 83
   3.3.2 Verification .................................. 89
3.4 Boundary conditions .............................. 92
3.5 Sources and coupling to Soledge2D ............ 96
3.6 Test cases ............................................... 99
   3.6.1 Slab geometry .................................. 99
   3.6.2 TCV geometry ................................. 101
   3.6.3 WEST geometry ................................. 103
3.1 Introduction

We concluded the previous chapter with the important result that, in order to speed up Monte Carlo simulations, reducing the number of histories by a factor up to 1000 is a viable strategy as long as the statistical noise realization is refreshed often enough and the solution is averaged at the end of the simulation, producing negligible biases in the solution.

What we did not discuss in the previous chapter is, on the other hand, the fact that the length of the simulations does not depend only on the number of histories, but also on the length of the latter.

By going back to figure 1.14, it is evident that simulating a neutral history in a plasma background with high density and low temperature can take $\sim 10^6$ more time than an history in a hotter region. Because it is precisely these cold and dense plasma regimes that act as the bottleneck of the simulations, and because in the tokamaks of next generation the size of these highly collisional regions is going to increase making it more difficult for neutral particles to escape them, a reduction of a factor 1000 in the number of histories may not be enough to reduce the computational time of plasma-neutral simulations to acceptable levels for, as an example, engineering design of the divertor. Such applications require in fact to explore all the parameter space in order to find the dependence of the engineering outputs (length of divertor targets, their tilt, material, etc...) with respect to the input (pumping speed, upstream density, power crossing the separatrix, etc...): operationally, this amounts either to running a large number of simulations with slightly different value of the inputs for which important speed-ups are desirable. Note that an alternative to this procedure is using adjoint techniques [20], but the introduction of statistical noise due to the Monte Carlo procedure in the adjoint equations increases the complexity of the adjoint techniques.

On the other hand, the Knudsen number of cold and dense plasmas often tends to be smaller than unity and a fluid description then becomes acceptable. Fluid models forgo the precise description of the neutrals as single particles and just consider their macroscopic behavior in terms of volume average quantities like density, fluid velocity and bulk temperature, thus making it possible to obtain substantial speed-ups in the calculations.

In this chapter, we will start in section 3.2 by describing the physical model used in the following with its own approximations; the numerical scheme used for its implementation with the Hybridizable Discontinuous Galerkin method will be shown in section 3.3. The coupling of the new fluid code with Soledge2D will then be described in sections 3.4 and 3.5 in terms of boundary conditions and source terms, and finally results of the full
code will be shown, starting from simplified test cases in 3.6 to simulations in realistic geometry in 3.6.3.

3.2 Physical model

As already said, we will focus on atoms in this manuscript, so let us start from the Boltzmann equation for the atom species derived in chapter 1

\[ \frac{\partial f}{\partial t} + \vec{v} \cdot \vec{\nabla} f = \left( \frac{\partial f}{\partial t} \right)_C \]  

(3.1)

In order to obtain fluid equations, as discussed in chapter 1, we take the moments of equation 3.1. The first two moments of the equation give us the continuity and momentum equations for the density \( n \) and fluid velocity \( \vec{u} \)

\[ \frac{\partial n}{\partial t} + \vec{\nabla} \cdot (n\vec{u}) = S_n \]  

(3.2)

\[ \frac{\partial n\vec{u}}{\partial t} + \vec{\nabla} \cdot \left( n\vec{u} \otimes \vec{u} + \frac{\Pi}{m} \right) = S_{\text{mom}} \]  

(3.3)

Up to this point, the equations are exact; nonetheless, the momentum equation requires information about the second moment of the distribution function, through the pressure tensor \( \Pi \). This would either require a further equation obtained taking the second moment of the Boltzmann equation, namely the energy equation, or the introduction of an approximation to close the system to the two equations obtained up to now.

For this reason, we will assume that the neutrals can be described by a local drifting Maxwellian distribution function with the same temperature of the background ions \( T_n = T_i \)

\[ f(\vec{r}, \vec{v}, t) = n(\vec{r}, t) \left( \frac{2\pi k_B T_i(\vec{r}, t)}{m} \right)^{-3/2} \exp \left( \frac{-m(\vec{v} - \vec{u}(\vec{r}, t))^2}{2k_B T_i(\vec{r}, t)} \right) \]  

(3.4)

This approximation can be explained by the assumption that the plasma is in a charge exchange dominated regime, for which we can expect that the collisions with the background ions drive the neutrals to local thermo-dynamical equilibrium (LTE).

Note also that, in a charge exchange dominated regime, we could also expect the collisions to equalize the momentum of ions and neutrals \( u_i \approx \vec{u}_n \cdot \vec{b} \), at least in the direction parallel to the magnetic field. On the other hand, neutrals generated at the solid walls tend to have an angular distribution different from the one of impacting ions, and the velocities will differ in the vicinity of the boundaries. For this reason, we will retain and solve the
3.2. Physical model

momentum equation 3.3 for the atoms.

Using the assumption of a Maxwellian vdf, we can close the system of equations by calculating the pressure tensor

$$\Pi = m \int d\vec{v}(\vec{v} - \vec{u})(\vec{v} - \vec{u})f(\vec{r}, \vec{v}, t)$$  \hspace{1cm} (3.5)

To show this, it is easier to work with the single components of the tensor, so

$$\Pi_{i,j} = m \int d\vec{v}(v_i - u_i)(v_j - u_j)f(\vec{v}) = m \int d\vec{w}w_iw_jf(\vec{w})$$

where $\vec{w} = \vec{v} - \vec{u}$ is the velocity of the particles relative to the fluid velocity and $v_{i,j,k}$, $u_{i,j,k}$ and $w_{i,j,k}$ are the components of the three velocities in a given coordinate system.

By substituting equation 3.4 in the definition of the pressure tensor, we can rewrite the latter as

$$\Pi_{i,j} = mn(\frac{2\pi k_B T}{m})^{-3/2} \int_{w_i=-\infty}^{\infty} \int_{w_j=-\infty}^{\infty} \int_{w_k=-\infty}^{\infty} w_iw_j e^{-\frac{mw_i^2}{2k_B T_i}} e^{-\frac{mw_j^2}{2k_B T_j}} e^{-\frac{mw_k^2}{2k_B T_k}} dw_i dw_j dw_k$$  \hspace{1cm} (3.6)

The integral in $w_k$ simply gives a factor $\sqrt{\frac{2\pi k_B T_i}{m}}$; for the other two integrals, there are two cases. For $i \neq j$ we get two integrals of the type

$$\int_{s=-\infty}^{\infty} se^{-cs^2} ds = 0$$  \hspace{1cm} (3.7)

which are equal to zero, being integrals of odd functions over a symmetric domain; as such, $\Pi_{i,j} = 0$ for $i \neq j$ ($i$ and $j$ have been taken arbitrarily, so this is true for all components outside of the main diagonal). The second case is the one for $i = j$, so one of the two integrals will be similar to the integral in $w_k$, giving another factor $\sqrt{\frac{2\pi k_B T_i}{m}}$, while the remaining integral is of the form

$$\int_{s=-\infty}^{\infty} s^2 e^{-cs^2} ds = \sqrt{\frac{\pi}{2}} \left( \frac{m}{2k_B T_i} \right)^{-3/2}$$  \hspace{1cm} (3.8)

so that we can obtain $\Pi_{i,i} = \Pi_{j,j} = \Pi_{k,k} = nk_B T_i = p$, or in tensor form $\Pi = pI$. The system of equations, under the assumption of local Maxwellian distribution function for the atoms, finally gives

$$\frac{\partial n}{\partial t} + \vec{\nabla} \cdot (n\vec{u}) = S_n$$  \hspace{1cm} (3.9)

$$\frac{\partial n\vec{u}}{\partial t} + \vec{\nabla} \cdot (n\vec{u}\vec{u} + \frac{p}{m} I) = \vec{S}_{mom}$$  \hspace{1cm} (3.10)
Note that assuming that the distribution function of the atoms is an exact (drifting) Maxwellian equates to a Chapman-Enskog expansion cut at the zero-th order in the Knudsen number \( f(\vec{r}, \vec{v}, t) = f_M + \sum_{i=1}^{\infty} Kn_i f_i(\vec{r}, \vec{v}, t) \). As a consequence, this also amounts to completely neglecting viscosity effects: if we consider a first order expansion, in fact, the pressure tensor would become

\[
\Pi = \Pi_M + \Pi_1 = pI - \zeta \left( \nabla \cdot \vec{u} \right) I - \mu \left( \nabla \vec{u} + (\nabla \vec{u})^T \right) - \frac{2}{3} \left( \nabla \cdot \vec{u} \right) I \quad (3.11)
\]

where \( \zeta \) is the volume viscosity and \( \mu \) is the dynamic viscosity, thus leading to an equation similar to Navier-Stokes momentum equation.

### 3.3 Numerical scheme

From the numerical point of view, equations 3.9 and 3.10 have been implemented in a new code; the vectorial equation 3.10 for the momentum is projected over the \((R, Z, \phi)\) coordinate system, where \(R\) is the major radius, \(Z\) is the vertical position and \(\phi\) is the toroidal angle measured from an initial poloidal plane. The system of equations can then be rewritten as

\[
\frac{\partial n}{\partial t} + \frac{\partial n u_R}{\partial R} + \frac{\partial n u_Z}{\partial Z} + \frac{1}{R} \frac{\partial n u_\phi}{\partial \phi} = S_n \quad (3.12)
\]

\[
\frac{\partial n u_R}{\partial t} + \frac{\partial (n u_R^2 + \frac{p}{m})}{\partial R} + \frac{\partial n u_R u_Z}{\partial Z} + \frac{1}{R} \frac{\partial n u_R u_\phi}{\partial \phi} = S_{mom,R} \quad (3.13)
\]

\[
\frac{\partial n u_Z}{\partial t} + \frac{\partial n u_R u_Z}{\partial R} + \frac{\partial (n u_Z^2 + \frac{p}{m})}{\partial Z} + \frac{1}{R} \frac{\partial n u_Z u_\phi}{\partial \phi} = S_{mom,Z} \quad (3.14)
\]

\[
\frac{\partial n u_\phi}{\partial t} + \frac{\partial n u_R u_\phi}{\partial R} + \frac{\partial n u_Z u_\phi}{\partial Z} + \frac{1}{R} \frac{\partial (n u_\phi^2 + \frac{p}{m})}{\partial \phi} = S_{mom,\phi} \quad (3.15)
\]

where \(n, u_R, u_Z, u_\phi\) and \(S_{mom,R}, S_{mom,Z}\) and \(S_{mom,\phi}\) are the components of the fluid velocity and of the momentum source in the \((R, Z, \phi)\) coordinate system.

An additional simplification has been introduced in the equations, namely the assumption of toroidal symmetry \( \frac{\partial}{\partial \phi} = 0 \) to reduce the problem to only two spatial dimensions; nonetheless, we need to keep equation 3.15 in order to properly evaluate the parallel component of the plasma momentum source \( S_{mom,\parallel}^p = \vec{S}_{mom}^p \cdot \vec{b} = S_{mom,R}^p \frac{B_R}{|B|} + S_{mom,Z}^p \frac{B_Z}{|B|} + S_{mom,\phi}^p \frac{B_\phi}{|B|} \). For completeness, the following derivation of the discretized equations will be done without this simplification even though in the actual code the toroidal symmetry assumption has been made: this amounts to simply substitute some of the elements of the Jacobian, that will be later introduced, with zeros, so...
3.3. Numerical scheme

all the following considerations are still valid.

In order to have a code able to handle unstructured meshes, due to the need to retain the interactions of the neutral particles with the realistic shape of the solid wall and due to the fact that we do not have any restriction to align the mesh elements with the magnetic field lines in contrast with the plasma solver, the code has been written using a finite element method called Hybridizable Discontinuous Galerkin (HDG). Furthermore recent efforts [29] have been made in parallel to create a fluid code for the plasma species with the same numerical tool, being thus able to model the edge plasma even in moving magnetic fields; an HDG fluid code for the neutrals then proves interesting also for such an application, and the development of a monolithic HDG code for both charged and neutral particles is currently on-going.

The HDG method has been first introduced in reference [16], and we refer to that paper for a formal description; in this section, we will try to explain the idea of the method in a more discursive way before showing its implementation in the framework of our specific problem.

Let us start by clarifying the meaning of the name HDG:

- Galerkin methods refer to a set of numerical methods in which we try to find the solution of an operator equation (i.e. finite elements for partial differential equations) in the form of a linear combination of the base functions of a chosen space;

- discontinuous refers to the fact that, given a computational discretization of the physical domain divided in sub-domains, the function space that we choose for the approximation of the solution (typically a sub-set of the space of polynomials of a given order) is defined locally in the sub-domains and the continuity of the solution is thus not guaranteed at the interface of two different sub-domains;

- hybridization, finally, refers to the practice of completely relaxing the continuity condition of the solution by the introduction of an hybrid unknown which represents the approximation of the trace of the solution (the restriction of the latter on the boundary of the domain) on the faces of each element.

Let us draw a comparison between a generic continuous Galerkin (CG) method and a generic discontinuous (DG) one. In both cases, solving the governing equations translates into finding the coefficients of a linear combination of known functions (in the following we will assume polynomial functions), and these coefficients represent the solution of the equations on a finite number of discretized nodes defined in the computational grid elements; the main difference is thus given by the choice of the functions used in the linear
Figure 3.1: Representation of the treatment of two neighbouring grid elements $\Omega_{k1}$ (in red) and $\Omega_{k2}$ (in green) for a generic continuous Galerkin (CG, on the left), discontinuous Galerkin (DG, centre) and hybridizable discontinuous Galerkin (HDG, on the right) method, having assumed a discretization with polynomials of order 4; for the DG and HDG method, the two elements have been purposely split in the figure to emphasize the discontinuity of the approximation of the solution. Blue markers indicate the position of the discretized points $i = 1, ..., N_{\text{elementnodes}}$ in which the equation for variable $x_h$ is to be solved; for the HDG method, the magenta markers indicate the position of the discretized points $j = 1, ..., N_{\text{facenodes}}$ in which the equation for the trace $\hat{x}_h$ for variable $x_h$ is to be solved.

combination. Every couple of neighboring sub-domains of course will have a number of nodes in common, as shown in figure 3.1; in the case of a CG method, the choice of the space dictates that the approximation of the solution in each of the common nodes "seen by" both of the elements will be the same, while in a DG method the value of the solution in each of the common nodes might be different between one element and the other; to ensure that the approximation of the solutions defined in all the sub-domains also approximates the solution in the global domain, fluxes that depend on the solution in each of the elements then have to be introduced to connect the latter.

From the numerical point of view, this means that in the case of CG methods each common node will receive a contribution from the discretized equations from each sub-domains in which it is contained, but at the same time we can "count" each common node only once. In DG methods, instead, each common node might have a different local value for the unknowns and it has to be considered as a different node for each neighboring element; consequently, the global number of degrees of freedom of the problem increases. If we take as an example the small computational domain given by the two triangular elements in figure 3.1, where each triangle contains 15 nodes and each face contains 5 nodes, the CG method will have a total of $(N_{\text{elementnodes}} \times N_{\text{elements}} - N_{\text{commonfaces}} \times N_{\text{facenodes}}) \times N_{\text{unknowns}} = 25 \times N_{\text{unknowns}}$ degrees of freedom due to the fact that one of the faces and all of its 4 nodes are in common between the two elements; the DG method, instead, will have to deal with the whole $N_{\text{elementnodes}} \times N_{\text{elements}} \times N_{\text{unknowns}} =$
3.3. Numerical scheme

30 × N_{unknowns} degrees of freedom. On the other hand, the fact that the solution is considered local in DG methods makes them well suited for parallelization, hp adaptivity (local refinement of the solution by reducing the element size $h$ or by increasing the order of the discretization $p$ in elements where higher accuracy level is required) and the use of unstructured meshes.

The HDG method has thus been developed in order to have a numerical tool with the same advantages of a DG method, while trying to limit the size of linear system, and consequently the computational cost, of the problem. To do this, the hybridization technique comes into play: the continuity of the solution at the interface between neighboring elements is re-written as an additional equation (or system of equations) in terms of a new variable that is defined only on the skeleton of the computational grid. The number of degree of freedom, then, does not strictly depend on the number of elements and of point in each element, but on the number of global faces and of points in each face.

Going back to the example in figure 3.1, the HDG problem is now defined over the 5 faces for a total of $N_{facenodes} \times N_{faces} \times N_{traceunknowns} = 25 \times N_{traceunknowns}$ global degree of freedom, the same as in the CG problem. The actual solution, on the other hand, will still have $N_{elementnodes} \times N_{elements} \times N_{unknowns} = 30 \times N_{unknowns}$ local degrees of freedom, but this is translated into 2 local problems which only require the multiplication of small matrices and are thus quite cheap from the computational point of view.

It is easy to see that the advantage of the HDG formulation over CG or DG methods is linked to the ratio of external to internal nodes: the more internal nodes in each element, the more the HDG method will outperform the other methods at least in terms of number of degrees of freedom. This is strictly true in 2D problems (which we will restrict to in the following) but it must be noted that in three-dimensional applications, where iterative solvers have to be used, the hybridization technique might introduce issues in the conditioning of the problem [40] and pure DG methods might be preferable.

3.3.1 Discretized system of equations

To better understand how the HDG method works, let us go back to the system of equations given by equations 3.12-3.15 that we want to solve. We can rewrite the system in tensorial form as

$$\frac{\partial \vec{e}}{\partial t} + \vec{\nabla} \cdot \vec{F} = \vec{S}$$ (3.16)
having defined

\[
\vec{x} = \begin{bmatrix} n \\ nu_R \\ nu_Z \\ nu_\phi \end{bmatrix}; \quad F = \begin{bmatrix} nu_R & nu_Z & nu_\phi \\ nu_R^2 + \frac{p}{m} & nu_R nu_Z & nu_R nu_\phi \\ nu_Z nu_R & nu_Z^2 + \frac{p}{m} & nu_Z nu_\phi \\ nu_\phi nu_R & nu_\phi nu_Z & nu_\phi^2 + \frac{p}{m} \end{bmatrix}; \quad \vec{S} = \begin{bmatrix} S_n \\ S_{mom,R} \\ S_{mom,Z} \\ S_{mom,\phi} \end{bmatrix}
\]

(3.17)

The system of equations is defined over a domain \( \Omega \subset \mathbb{R}^d \) bounded by \( \partial \Omega \), with \( d \) the number of spatial dimensions of the problem. First of all, let us rewrite the problem in weak form: as such, let us multiply the equation by a test function \( \vec{w} \) defined in the same space as the solution \( \vec{x} \), and let us integrate the equation over the whole domain

\[
\int_{\Omega} \frac{\partial \vec{x}}{\partial t} \cdot \vec{w} dV + \int_{\Omega} (\vec{\nabla} \cdot F) \cdot \vec{w} dV = \int_{\Omega} \vec{S} \cdot \vec{w} dV
\]

(3.18)

Because we want to solve this equation numerically, we need to create a computational domain \( \hat{\Omega} = \bigcup_{i=1}^{N_{el}} \Omega_i \) made of a finite number \( N_{el} \) of disjoint elements \( \Omega_i \); we will call \( \partial \hat{\Omega}_{int} = \bigcup_{i,j=1,i\neq j}^{N_{el}} \Omega_i \cap \Omega_j \) the union of all the internal faces of the elements, while \( \partial \hat{\Omega}_{ext} = \bigcup_{i=1}^{N_{el}} \hat{\Omega}_i \cap \partial \Omega \) the union of the all boundary faces, and \( \partial \hat{\Omega} = \partial \hat{\Omega}_{int} \cup \partial \hat{\Omega}_{ext} \) the set of all faces. Having restricted the domain to a computational one, we define a discontinuous function space over which we want the discretized equations to be solved

\[
W_h^j = \left\{ \vec{w} \in \left( L^2(\hat{\Omega}) \right)^m : \vec{w}|_{\Omega_i} \in \left( P^j(\Omega_i) \right)^m, \forall \Omega_i \in \hat{\Omega} \right\}
\]

(3.19)

with \( m \) the dimensions of the solution vector \( \vec{x} \), \( L^2 \) the Hilbert space of functions with integrable square power, and \( P^j \) the space of all polynomials of degree at most \( j \); that is to say that we want the solution of the discretized system of equations, thus the approximation of the "real" solution, to be defined in each element in terms of polynomial functions of degree smaller or equal than \( j \); furthermore, it is important to note that a solution defined in this space will be discontinuous between the computational subdomains. For completeness, let us also define a second function space

\[
M_h^j = \left\{ \vec{\mu} \in \left( L^2(\partial \hat{\Omega}) \right)^m : \vec{\mu}|_{F_i} \in \left( P^j(F_i) \right)^m, \forall F_i \in \partial \hat{\Omega} \right\}
\]

(3.20)

which, as we will see in a short while, will prove useful for the HDG solution.

Restricting the problem over the computational domain \( \hat{\Omega} \) means that we now seek
an approximation \( \hat{x}_h \) defined in the function space \( W^j_h \) so that

\[
\int_{\hat{\Omega}} \frac{\partial \hat{x}_h}{\partial t} \cdot \hat{w}_h dV + \int_{\hat{\Omega}} \left( \nabla \cdot \hat{F}_h \right) \cdot \hat{w}_h dV = \int_{\hat{\Omega}} \hat{S}_h \cdot \hat{w}_h dV \tag{3.21}
\]

The approximation \( \hat{x}_h \) is solution to equation 3.21 over all the domain \( \hat{\Omega} \) if the function also verifies the equation in each sub-domain \( \Omega_i \) for \( i = 1, \ldots, N_{el} \). So, using also the divergence theorem, we can rewrite the equation as

\[
\int_{\Omega_i} \frac{\partial \hat{x}_h}{\partial t} \cdot \hat{w}_h dV + \int_{\partial \Omega_i} \left( \hat{F}_h \cdot \hat{n} \right) \cdot \hat{w}_h dS - \int_{\Omega_i} F_h \cdot \nabla \hat{w}_h dV = \int_{\Omega_i} \hat{S}_h \cdot \hat{w}_h dV \tag{3.22}
\]

In the integral over the boundary of each element the approximation \( \tilde{F}_h \) of the numerical flux \( F(\hat{x}) \) appear. The hybridization of the method finally comes into play in the definition of this numerical flux, which we will take as

\[
\tilde{F}_h \cdot \hat{n} = F(\hat{x}_h) \cdot \hat{n} + T(\hat{x}_h, \tilde{x}_h) (\hat{x}_h - \tilde{x}_h) \tag{3.23}
\]

where \( T(\hat{x}_h, \tilde{x}_h) \) is a stabilization matrix that has important effects on the stability and accuracy of the scheme, and more importantly \( \hat{x}_h \in M^j_h \) is the approximation of the trace of the solution on the boundary of the element. As it has been introduced, the approximation of the trace is defined in the space \( M^j_h \), so again is in terms of polynomials of order \( \leq j \) but it is now defined only on the faces of the elements. The main consequence of this is given by the fact that each face, or at least the internal ones, is in common between two neighboring elements, thus losing the locality of the definition and linking what happens in the two computational sub-domains.

Substituting the definition 3.23 of the numerical flux in the discretized equation 3.22, we obtain

\[
\int_{\Omega_i} \frac{\partial \hat{x}_h}{\partial t} \cdot \hat{w}_h dV + \int_{\partial \Omega_i} \left( F(\hat{x}_h) \cdot \hat{n} \right) \cdot \hat{w}_h dS + \int_{\partial \Omega_i} (\hat{x}_h - \tilde{x}_h) T \cdot \hat{w} dS - \int_{\Omega_i} F(\hat{x}_h) \cdot \nabla \hat{w}_h dV = \int_{\Omega_i} \hat{S}_h \cdot \hat{w}_h dV \tag{3.24}
\]

Examples of definitions of the stabilization matrix \( T \) can be found, for Euler and Navier-Stokes equations, for example in [48]; in this manuscript we will use a simplified definition

\[
T(\hat{x}_h, \tilde{x}_j) = \tau I \tag{3.25}
\]
where $\tau$ is a positive constant stabilization factor set as input parameter*. Finally, we will rewrite the terms $F(\vec{x}_h)$ and $F(\tilde{\vec{x}}_h)$ in terms of their Jacobian. The Jacobian of the numerical flux is equal to $J(\vec{y}) = \frac{\partial F(\vec{y})}{\partial \vec{y}} = \left( \frac{\partial F(\vec{y})}{\partial n}, \frac{\partial F(\vec{y})}{\partial n_u^R}, \frac{\partial F(\vec{y})}{\partial n_u^Z}, \frac{\partial F(\vec{y})}{\partial n_u^\phi} \right)$ with

\[
\begin{align*}
\frac{\partial F(\vec{y})}{\partial n} &= \begin{bmatrix}
0 & 0 & 0 \\
\frac{T}{m} - u_R^2 & -u_R u_Z & -u_R u_\phi \\
-u_R u_Z & \frac{T}{m} - u_Z^2 & -u_Z u_\phi \\
-u_R u_\phi & -u_Z u_\phi & \frac{T}{m} - u_\phi u_\phi
\end{bmatrix}, \\
\frac{\partial F(\vec{y})}{\partial n_u^R} &= \begin{bmatrix}
1 & 0 & 0 \\
0 & 2u_R & u_\phi \\
u_R & u_Z & 0 \\
u_\phi & 0 & 0
\end{bmatrix}, \\
\frac{\partial F(\vec{y})}{\partial n_u^Z} &= \begin{bmatrix}
0 & 1 & 0 \\
0 & u_R & 0 \\
u_R & 2u_Z & u_\phi \\
0 & u_\phi & 0
\end{bmatrix}, \\
\frac{\partial F(\vec{y})}{\partial n_u^\phi} &= \begin{bmatrix}
0 & 0 & 1 \\
0 & 0 & u_R \\
0 & 0 & u_Z \\
u_R & u_Z & 2u_\phi
\end{bmatrix}
\end{align*}
\]

so that it is easy to demonstrate that $J(\vec{x}_h)\vec{x}_h = F(\vec{x}_h)$ and $J(\tilde{\vec{x}}_h)\tilde{\vec{x}}_h = F(\tilde{\vec{x}}_h)$, and we obtain an explicit relationship between the numerical flux and the approximations of the solution and of its trace respectively. The final discretized equation then becomes

\[
\int_{\Omega} \frac{\partial \vec{x}_h}{\partial t} \cdot \vec{w}_h dV + \int_{\partial \Omega} \frac{\partial F}{\partial \vec{x}_h} \vec{x}_h \cdot (\vec{w}_h \cdot \hat{n}) dS + \int_{\partial \Omega} \tau \vec{x}_h \cdot \vec{w}_h dS - \int_{\partial \Omega} \tau \vec{x}_h \cdot \vec{n} dS - \int_{\Omega} \frac{\partial F}{\partial \vec{x}_h} \vec{x}_h \cdot \vec{\nabla} \vec{w}_h dV = \int_{\Omega} \vec{S}_h \cdot \vec{w}_h dV
\]

\[= \int_{\Omega} \vec{S}_h \cdot \vec{w}_h dV \quad (3.27)\]

The solution that we are looking for, and its trace, can both be written as linear combinations of the base functions of their respective spaces

\[
\vec{x}_h = \sum_{i=0}^{j_1} \alpha_i N_i \\
\tilde{\vec{x}}_h = \sum_{i=0}^{j_2} \beta_i \tilde{N}_i
\]

where $N_i$ and $\tilde{N}_i$ represent the base of the spaces $W_h^{j_1}$ and $M_h^{j_2}$ respectively. Solving equation 3.27 then means finding the $(j_1 + 1) + (j_2 + 1)$ coefficients $\alpha_i$ and $\beta_i$; for the $\alpha_i$ we have to solve $j_1 + 1$ equations similar to 3.27 with $j_1 + 1$ linear independent test

---

* A detailed study on the effect of this choice is provided in reference [28], showing that the convergence behavior of the HDG scheme improves increasing the stabilization factor $\tau$ from a minimum value of $\sim 1$ in dimensionless units to a maximum value specific to the problem being solved.
3.3. Numerical scheme

functions $\vec{w}_i \in W_h^{j_1}$. A good choice for the test functions is thus to use the base functions themselves of the space $W_h^{j_1}$.

$$\sum_{m=0}^{j_1} \frac{\partial \alpha_m}{\partial t} \int_{\Omega_i} N_m \cdot N_k dV + \sum_{n=0}^{j_2} \beta_n \int_{\partial \Omega_i} \frac{\partial F}{\partial x_h} \tilde{N}_n \cdot N_k dS + \tau \sum_{m=0}^{j_1} \alpha_m \int_{\partial \Omega_i} N_m \cdot N_k dS - \tau \sum_{n=0}^{j_2} \beta_n \int_{\partial \Omega_i} \tilde{N}_n \cdot N_k dS - \sum_{m=0}^{j_1} \alpha_m \int_{\partial \Omega_i} \frac{\partial F}{\partial x_h} N_m \cdot \nabla N_k dV = \int_{\Omega_i} \vec{S}_h \cdot N_k dV \quad \forall k = 1, \ldots, j_1 + 1$$

(3.29)

or, in matrix form

$$M \frac{\partial \vec{\alpha}}{\partial t} + H \vec{\beta} + D \vec{\alpha} - E \vec{\beta} - C \vec{\alpha} = \vec{S}$$

(3.30)

where we have defined the vectors $\vec{\alpha} = [\alpha_1, \ldots, \alpha_{j_1}]$ and $\vec{\beta} = [\beta_1, \ldots, \beta_{j_2}]$.

We now need to introduce a time discretization for the first term. In the following, a simple backward Euler scheme is used:

$$M \frac{\vec{\alpha}^t - \vec{\alpha}^{t-1}}{\Delta t} + H \vec{\beta} + D \vec{\alpha}^t - E \vec{\beta}^t - C \vec{\alpha}^t = \vec{S}^t$$

(3.31)

where $t$ represent the current time step and $t - 1$ the previous one. In the following, we will drop the super script $t$ for the current time step, while we will leave $t - 1$ to distinguish the (known) previous values of the unknowns.

As we explained, this problem is equivalent to solving the $j_1 + 1$ equations 3.29, so it is not enough to calculate all the $(j_1 + 1) + (j_2 + 1)$ coefficients $\vec{\alpha}$ and $\vec{\beta}$; nonetheless we can find a relationship between the two, namely:

$$\vec{\alpha} = \left( \frac{1}{\Delta t} M + D - C \right)^{-1} \left( S + \frac{1}{\Delta t} \vec{\alpha}^{t-1} \right) + \left( \frac{1}{\Delta t} M + D - C \right)^{-1} (E - H) \vec{\beta} = U \vec{\beta} + U_0$$

(3.32)

To find $\vec{\beta}$, we are still missing $j_2 + 1$ equations; fortunately, thanks to the hybridization mechanism, we have another relationship that we can impose: the normal component of the numerical flux along faces should be continuous for each face

$$\int_{\partial \Omega_{int}} \vec{F} \cdot \hat{n} \vec{\mu}_h dS + \text{bound} = 0$$

(3.33)

where $\vec{\mu}_h \in M_h^{j_2}$ are the test functions and a term $\text{bound}$ is introduced on the external faces to impose the boundary conditions of the problem. We use again the HDG approximation
of the numerical flux

\[
\int_{\partial \Omega_{\text{int}}} \frac{\partial F}{\partial \tilde{x}_h} \cdot \tilde{\mu}_h \cdot \hat{n} dS + \tau \int_{\partial \Omega_{\text{int}}} \tilde{x}_h \cdot \tilde{\mu}_h dS - \tau \int_{\partial \Omega_{\text{int}}} \tilde{x}_h \cdot \tilde{\mu}_h + \text{bound} = 0 \quad (3.34)
\]

Like before, we use the approximations of the solution and its trace 3.28; to obtain \( j_2 + 1 \) equations for the coefficients \( \beta \), we need to choose \( j_2 + 1 \) linearly independent test functions \( \tilde{\mu}_h \), so we use the base functions \( \tilde{N}_i \) of the space \( M_{j_2} \)

\[
\sum_{n=0}^{j_2} \beta_n \int_{\partial \Omega_{\text{int}}} \frac{\partial F}{\partial \tilde{x}_h} \tilde{N}_n \cdot \tilde{N}_i \cdot \hat{n} dS + \tau \sum_{m=0}^{j_1} \alpha_m \int_{\partial \Omega_{\text{int}}} N_m \cdot \tilde{N}_i dS - \\
- \tau \sum_{n=0}^{j_2} \beta_n \int_{\partial \Omega_{\text{int}}} \tilde{N}_n \cdot \tilde{N}_i dS + \text{bound} = 0 \quad \forall i = 1, \ldots, j_2 + 1 \quad (3.35)
\]

or in matrix form

\[
H_f \tilde{\beta} + D_f \tilde{\alpha} - E_f \tilde{\beta} + \text{bound} = 0 \quad (3.36)
\]

By using the relationship between \( \tilde{\alpha} \) and \( \tilde{\beta} \) from equation 3.32 we finally have an equation for \( \tilde{\beta} \) only

\[
H_f \tilde{\beta} + D_f (U \tilde{\beta} + U_0) - E_f \tilde{\beta} + BC \quad (3.37)
\]

To conclude, with this equation, if the matrix \( H_f + D_f \left( \frac{1}{\Delta t} M + D - C \right)^{-1} (E - H) - E_f \) is invertible, we can thus compute the coefficients \( \tilde{\beta} \) and consequently the approximation of the trace of the solution \( \tilde{x}_h \). Of course we are interested in the approximation of the solution itself, which can be computed starting from the trace using equation 3.32.

It is important to note that the integrals used for equation 3.37 are defined over all \( \partial \Omega_{\text{int}} \), thus the inversion of the matrix \( H_f + D_f \left( \frac{1}{\Delta t} M + D - C \right)^{-1} (E - H) - E_f \) is the most computational intensive step of the whole algorithm: the matrix, in fact, has dimensions \( (N_{\text{internal faces}} \times N_{\text{facenodes}} \times d) \times (N_{\text{internal faces}} \times N_{\text{facenodes}} \times d) \) and this number can quickly become unmanageably large for regularly used computational grids. Nonetheless, the matrix is sparse, making it possible to store only a limited number of elements and easily invert it with sparse matrix solvers like Pastix [47]. Furthermore, like explained in the beginning, depending on the number of nodes on the faces of each
element with respect to the number of total element nodes, the total number of degrees of freedom of the problem can be lower than for other numerical schemes. Finally, the integrals used to obtain equation 3.32 are instead defined only over the single elements: as such the coefficients $\vec{\alpha}$ and the approximation of the solution $\vec{x}_h$ can be computed at the end of each iteration element-by-element, with obvious implications in the parallelization of the scheme.

### 3.3.2 Verification

The method of the manufactured solutions MMS has been applied to the code to verify the implementation of the discretized equations.

The idea behind this method is to choose an analytic function $x_{MS}$ which, in the most generic case, will not be solution of the governing equations $\frac{\partial x}{\partial t} + f(x) = 0$; as such, substituting the chosen analytic function in the governing equations will generate a residual $\frac{\partial x_{MS}}{\partial t} + f(x_{MS}) = E_{MS}$ which we can calculate analytically. The method of manufactured solutions can then be considered as applying the code to solve a new problem $\frac{\partial x_{MS}}{\partial t} + f(x_{MS}) - E_{MS} = 0$.

This amounts to just choose a function (hence the name "manufactured solutions") and set it as initial condition $x^0_h$ for the numerical code; furthermore, a source term is introduced equal to the numerical discretization $E_h$ of the remainder $E_{MS}$ that we can compute analytically; finally, Dirichlet conditions are set imposing the value of the chosen function at the boundary location of the domain.

By running the code in this configuration for one time step, we can compute the norm\(^1\) of the residual $R = ||x^{t+1}_h - x^t_h||$. This error will never be identically zero due to the finite spatial discretization of the computational grid; nonetheless, if the equations have been correctly implemented, the residual should only be a function of the mesh size and of the numerical method used.

In the specific case of HDG methods the order of convergence has been shown to be, for Oseen equations initially [17] and for Euler and Navier-Stokes equation [48] later, $p+1$ when using polynomials of order $p$ for the approximation of the solution; that is, the theoretical residuals should behave as $R_{th} \propto h^{p+1}$ where $h$ is the size of the grid elements of an uniform mesh. In table 3.1 the values of the L2 norm of the residuals for density and the $x$ and $y$ components of the flux are reported for different values of the order of the polynomials and for different mesh refinements; the values of the residuals for density are also shown in figure 3.2 in function of the element size $h$ and of the

---

\(^1\)Note that, having defined the solution of the HDG problem and its trace in subsets of the $L^2$ space, we will take the norm $||x|| = ||x||_2 = \sqrt{\int |x|^2dy}$
order of the polynomials $p$. For simplicity, rectangular grids with uniform triangular elements of size $h$ were used in the simulations; the manufactured solution used to test the implementation of the equations were combinations of trigonometric functions similar to $x_i = A_i \sin(\alpha_{i,1} R + \beta_{i,1}) \cos(\alpha_{i,2} Z + \beta_{i,2})$ for each variable.

![Figure 3.2: Convergence curves for the density for fixed polynomials of order $p$ in function of the refinement of the mesh, on the top, and for fixed mesh refinements in function of the polynomial order on the bottom. Markers indicate the numerical residuals obtained by the code during the MMS tests, and solid lines have been added to help show their slope; dashed lines represent the theoretical residual values obtained as $R^p(h) \propto Ah^{p+1}$, and the text shows the theoretical order of convergence.](image)

As evident, at least for approximations using polynomials of order $p$ up to five, the
### 3.3. Numerical scheme

#### Table 3.1: Residuals and local order of convergence for density and the two components of the flux in function of the order \( p \) of the polynomials used in the simulations and of the dimensionless grid size \( h \).

| \( p \) | \( \frac{1}{h}[a.u.] \) | \( ||t^{p+1} - t^p||_{L^2} \) R | Slope | \( ||\Gamma_{x}^{p+1} - \Gamma_{x}^{p}||_{L^2} \) R | Slope | \( ||\Gamma_{y}^{p+1} - \Gamma_{y}^{p}||_{L^2} \) R | Slope |
|-------|-----------------|-----------------|-------|-----------------|-------|-----------------|-------|
| 1     | 2               | 7.10e-3         | -     | 4.47e-2         | -     | 5.83e-2         | -     |
|       | 4               | 1.54e-3         | 2.21  | 1.06e-2         | 2.08  | 1.32e-2         | 2.14  |
|       | 8               | 3.23e-4         | 2.25  | 2.69e-3         | 1.98  | 3.06e-2         | 2.11  |
|       | 16              | 7.47e-5         | 2.11  | 6.34e-4         | 2.09  | 7.27e-4         | 2.07  |
|       | 32              | 1.80e-5         | 2.05  | 1.52e-4         | 2.06  | 1.77e-4         | 2.04  |
|       | 64              | 4.43e-6         | 2.02  | 3.72e-5         | 2.03  | 4.37e-5         | 2.02  |
| 2     | 2               | 3.84e-4         | -     | 2.49e-3         | -     | 3.71e-3         | -     |
|       | 4               | 4.18e-5         | 3.20  | 2.74e-4         | 3.19  | 5.06e-4         | 2.87  |
|       | 8               | 4.51e-6         | 3.21  | 3.09e-5         | 3.15  | 6.29e-5         | 3.01  |
|       | 16              | 5.19e-7         | 3.12  | 3.69e-6         | 3.07  | 7.80e-6         | 3.01  |
|       | 32              | 6.22e-8         | 3.06  | 4.49e-7         | 3.04  | 9.74e-7         | 3.00  |
|       | 64              | 7.62e-9         | 3.03  | 5.53e-8         | 3.02  | 1.22e-7         | 3.00  |
| 3     | 2               | 3.95e-5         | -     | 3.32e-4         | -     | 5.40e-4         | -     |
|       | 4               | 1.53e-6         | 4.69  | 1.60e-5         | 4.38  | 2.35e-5         | 4.52  |
|       | 8               | 7.84e-8         | 4.29  | 8.95e-7         | 4.16  | 1.32e-6         | 4.15  |
|       | 16              | 4.44e-9         | 4.14  | 4.97e-8         | 4.17  | 7.77e-8         | 4.09  |
|       | 32              | 2.65e-10        | 4.07  | 2.74e-9         | 4.18  | 4.66e-9         | 4.06  |
|       | 64              | 1.63e-11        | 4.02  | 1.53e-10        | 4.16  | 2.83e-10        | 4.04  |
| 4     | 2               | 2.10e-6         | -     | 1.69e-5         | -     | 2.62e-5         | -     |
|       | 4               | 5.49e-8         | 5.26  | 4.60e-7         | 5.20  | 8.34e-7         | 4.97  |
|       | 8               | 1.54e-9         | 5.16  | 1.40e-8         | 5.04  | 2.59e-8         | 5.01  |
|       | 16              | 4.31e-11        | 5.16  | 4.10e-10        | 5.09  | 7.90e-10        | 5.03  |
|       | 32              | 1.26e-12        | 5.10  | 1.27e-11        | 5.01  | 2.46e-11        | 5.01  |
|       | 64              | 1.37e-13        | 3.20  | 3.83e-12        | 1.73  | 3.87e-12        | 2.67  |
| 5     | 2               | 1.47e-7         | -     | 1.56e-6         | -     | 2.14e-6         | -     |
|       | 4               | 1.31e-9         | 6.81  | 1.60e-8         | 6.61  | 2.37e-8         | 6.50  |
|       | 8               | 1.46e-11        | 6.49  | 2.06e-10        | 6.28  | 3.20e-10        | 6.21  |
|       | 16              | 2.07e-13        | 6.14  | 3.03e-12        | 6.09  | 4.74e-12        | 6.08  |
|       | 32              | 9.57e-14        | 1.11  | 2.44e-12        | 0.31  | 2.08e-12        | 1.19  |
|       | 64              | 1.85e-13        | -0.95 | 5.39e-12        | -1.14 | 4.64e-12        | -1.16 |
theoretical order of convergence $p + 1$ is recovered. Note that, even though the residuals seem to saturate for high order polynomials in highly refined meshes, this is only due to the fact that the residuals have reached machine precision (shown by the horizontal dashed line in the figures). Consequently, we can conclude that the implementation of the governing equations in the code has been successfully verified.

3.4 Boundary conditions

Boundary conditions have to be set at the solid walls, at the pump ducts and at the core interface:

- at solid walls, a fraction $R_n$ of the neutral flux impinging on the wall and a fraction $R_i$ of the ion flux on the wall is recycled;
- at pump ducts, a pumping flux of neutrals is imposed;
- at the core interface, we assume that all neutrals that cross this boundary are eventually ionized in the hotter core region, so that the interface acts as a perfectly absorbing surface.

If we write the flux on a surface as $\Gamma = \Gamma^+ + \Gamma^-$, where the superscripts $+$ and $-$ indicate the direction of the flux with respect to the vector normal to the surface, we can effectively impose in all three cases a flux of neutrals that can be expressed as

$$\Gamma_n^+ = R_i \Gamma_i^- + (R_n - 1) \Gamma_n^-$$  \hspace{1cm} (3.38)

Note that throughout this manuscript, we will take the normal to every surface as going from the outside of the computational domain towards the inside, for example the normal to the plasma facing components will always be directed from the wall to the core and the normal to the core boundary will always be directed from the magnetic axis towards the wall.

Equation 3.38 lets us write each of the three boundary conditions:

- at the solid walls, $R_n$ and $R_i$ are set as input parameters for the simulation;
- at the core interface, we impose that $R_n = R_i = 0$ to make the surface perfectly absorbing;
3.4. Boundary conditions

- at the pump ducts, $R_i = 0$ and $R_n$ is either set as an input parameter, or is recalculated in order to keep an input pump flux constant \([50]\)

\[
S = 36.38A(1 - R_n)\sqrt{\frac{T}{m}}
\] (3.39)

where $S$ is the pumping speed \([m^3/s]\), $A$ is the pump duct entrance area \([m^2]\) and $T$ is the neutral gas temperature. Nonetheless while the boundary condition written this way allows to also model the boundary fluxes due to presence of pumps, in the following no pumping flux will be present in the fluid code, as we assume that the neutral gas in the vicinity of the duct entrances mostly behaves kinetically.

The final ingredient missing is the negative-directed part of the neutral flux. If we take a coordinate system $(\hat{x}, \hat{y}, \hat{z})$ centered on the surface $S$ with $\hat{x} = \hat{n}$ directed along the surface to the wall ($\vec{v} \cdot \hat{n} = v_x$), we can write

\[
\Gamma^- = \int f(\vec{r}, \vec{v}, t) \vec{v} \cdot \hat{n} d\vec{v} = \int_{v_x = -\infty}^{v_{lim}} dv_x v_x \int_{v_y = -\infty}^{\infty} dv_y \int_{v_z = -\infty}^{\infty} dv_z f(\vec{r}, \vec{v}, t) \tag{3.40}
\]

By assuming that the distribution function of the neutral particles is a drifting Maxwellian

\[
f(\vec{r}, \vec{v}, t) = n \left(\frac{2\pi k_B T}{m}\right)^{-3/2} e^{-\frac{m(v_x - u_x)^2}{2k_B T}} e^{-\frac{m(v_y - u_y)^2}{2k_B T}} e^{-\frac{m(v_z - u_z)^2}{2k_B T}} \tag{3.41}
\]

with drifting velocity $\vec{u} = (u_x, u_y, u_z)$, we can rewrite the integral in 3.40 as

\[
\Gamma^- = n \left(\frac{2\pi k_B T}{m}\right)^{-3/2} \int_{v_x = -\infty}^{-u_x} dv_x v_x e^{-\frac{m(v_x - u_x)^2}{2k_B T}} \int_{v_y = -\infty}^{\infty} dv_y e^{-\frac{m(v_y - u_y)^2}{2k_B T}} \int_{v_z = -\infty}^{\infty} dv_z e^{-\frac{m(v_z - u_z)^2}{2k_B T}} \tag{3.42}
\]

where the limiting velocity $v_{lim} = -u_x$ is to make sure that the integral is carried over all particles that cross the surface along the negative direction, taking into account the drifting velocity.

The integrals along the $y$ and $z$ directions give each a contribution

\[
\int_{-\infty}^{\infty} e^{-a(t-t_0)^2} dt = \sqrt{\frac{\pi}{a}} \tag{3.43}
\]
giving then

$$
\Gamma^- = n \left( \frac{2\pi k_B T}{m} \right)^{-3/2} \sqrt{\frac{2\pi k_B T}{m}} \int_{v_x=\infty}^{-u_x} dv_x v_x e^{-\frac{m(u_x-a)^2}{2k_B T}}
$$

(3.44)

The remaining integral can be solved with a change of variable

$$
\int_{t_0}^{a} te^{-b(t-t_0)^2} dt = \int_{-\infty}^{a-t_0} se^{-bs^2} ds + t_0 \int_{-\infty}^{a-t_0} e^{-bs^2} ds = \frac{1}{2} t_0 \sqrt{\frac{\pi}{b}} \left( erf(\sqrt{b}(a - t_0)) + 1 \right) - \frac{1}{2b} e^{-\sqrt{b}(a-t_0)^2}
$$

(3.45)

resulting finally in

$$
\Gamma^- = \frac{n\hat{u} \cdot \hat{n}}{2} \left( erf(-2\hat{u} \cdot \hat{n} \sqrt{\frac{m}{2k_B T}} + 1) \right) - n \sqrt{\frac{k_B T}{2\pi m}} e^{-\frac{2m|\hat{u} \cdot \hat{n}|^2}{k_B T}}
$$

(3.46)

As before, the temperature used in the Maxwellian distribution function is taken as the temperature of the background plasma ions. The normal vector is computed using the \((R,Z)\) coordinates of each surface element, and is thus defined only in two dimensions; as such, the boundary values for the three components of the flux are set to

$$
\begin{align*}
\Gamma_R &= \left( R_i \Gamma_i + (R_n - 1) \Gamma_n^- \right) \hat{n}_R \\
\Gamma_Z &= \left( R_i \Gamma_i + (R_n - 1) \Gamma_n^- \right) \hat{n}_Z \\
\Gamma_\phi &= \left( R_i \Gamma_i + (R_n - 1) \Gamma_n^- \right) \hat{n}_\phi = 0
\end{align*}
$$

(3.47)

where \(\hat{n}_R, \hat{n}_Z\) and \(\hat{n}_\phi\) are the three components of the normal vector in the \((R,Z,\phi)\) coordinate system. For the recycling boundary condition, nonetheless, this makes for an approximated treatment of the underlying physical processes: first of all, the coefficients \(R_i\) and \(R_n\) should depend on the mass of the projectile (atom or ion) and of the target substrate, on the energy of the projectile and of the angle of impact against the normal to the surface; secondly, and more importantly, the direction of the recycled or reflected particles have a distribution not centered on the normal to the surface (again depending on the elements, energy and impact angle).

A more refined model for the wall reflection is available in Eirene based on TRIM [42] calculations, providing particle, momentum and energy reflection coefficients for single incident particles with a discretized angle and energy.

Fluid codes, on the other hand, only follow the fluxes impacting on the surfaces integrated
on the whole velocity space. As such, effective reflection coefficients $R_{\text{eff}}$ would need to be constructed by integrating the values $R(E_{\text{in}}, \alpha_{\text{in}})$ obtained by the discretized TRIM tables. For example, for the contribution of the backscattering of neutrals to the neutral particle flux we could write, using the same coordinate system used by TRIM

$$
\Gamma_{n \rightarrow n}^+ = \int_{v_{\text{in}}=0}^{\infty} \int_{\alpha_{\text{in}}=0}^{\pi/2} \int_{\phi_{\text{in}}=0}^{2\pi} R_n(mv_{\text{in}}^2, \alpha_{\text{in}})\Gamma_n^- (v_{\text{in}}, \alpha_{\text{in}}, \phi_{\text{in}}) dv_{\text{in}} d\alpha_{\text{in}} d\phi_{\text{in}} = R_{\text{eff}} \Gamma_n^- \tag{3.48}
$$

While this expression would introduce a further complexity in the code due to the need to re-calculate effective values for the reflection coefficients of particle, momentum and energy for both ions and atoms, all along the wall and at each iteration, it would be a necessary step to guarantee the consistency of the fluid boundary conditions with Eirene [34]. Nonetheless, it is to be reminded that the fluid code developed here is but a building block for an hybrid model: for this reason, as we will explain in the next chapter, the reflection of neutrals and the recycling of ions can be left to Eirene, which already has the TRIM tables implemented. Throughout this chapter, then, the simplified version of the boundary fluxes presented in equation 3.47 will be used in the fluid code.

Finally, it has to be noted that in the previous description we implicitly assumed that all ions and neutrals reflect off the boundary surfaces as atoms but, as explained in chapter 1, this is not true.

Of the recycled ions flux $R_i \Gamma_i^-$, let us call $R_{i\text{backsc}}$ the fraction which is promptly backscattered as energetic atoms and $R_{i\text{des}}$ the remaining fraction which is thermally desorbed as cold molecules; similarly, we will have $R_{n\text{backsc}}$ and $R_{n\text{des}}$ for the reflected neutral flux. As such, the net flux of neutrals at the solid surfaces can be written as

$$
\Gamma_n = R_i R_{i\text{backsc}} \Gamma_i^- + R_n R_{n\text{backsc}} \Gamma_n^- + R_i R_{i\text{des}} \Gamma_i^- + R_n R_{n\text{des}} \Gamma_n^- - \Gamma_n^- \tag{3.49}
$$

Keeping in mind that eventually all the molecules thermally desorbed will dissociate, the impact of the simplification of equation 3.38, that is assuming that $R_{i\text{backsc}} = 1$ and consequently $R_{i\text{des}} = 0$ for both the ions and neutrals fluxes, is thus effectively a shift of the birth of the atoms by molecule dissociation from a volumetric source (near the targets, due to the low kinetic energy of recycled molecules) to a purely surface source, at least from the mass point of view. Assuming that all recycling and reflection is undergone through atoms (with the same temperature of the background ions, remember) completely removes on the other hand several molecular channels allowing to dissipate...
energy and momentum.

While this hints to the fact that this approximation in the recycling process might introduce important errors in the solution, it has to be reminded that this is strictly true only if the fluid code presented here is used on its own to model the neutral gas; as we will see in the next chapter, this restriction will be completely removed by having Eirene handle the molecules.

3.5 Sources and coupling to Soledge2D

To complete the fluid code, sources due to interactions between plasma and neutrals have to be computed; a total of eight source terms are required:

- for Soledge2D, the fluid code has to provide a particle source term \( S_p^n [m^{-3}s^{-1}] \), a parallel momentum source term \( S_G^p [m^{-2}s^{-2}] \) and an energy source term for both electrons \( S_E^{e,n} [W/m^3] \) and ions \( S_E^{i,n} [W/m^3] \);

- for the fluid code itself, we need a source term for the continuity equation \( S_n^n [m^{-3}s^{-1}] \) and a momentum source term for all three components of the neutral flux \( S_{moment,R}^n \), \( S_{moment,Z}^n \) and \( S_{moment,\phi}^n [m^{-2}s^{-2}] \).

To ensure conservation of mass and momentum during plasma-neutral collisions, the following relations have to be verified

\[
S_p^n = -S_n^n \\
S_G^p = -S_{moment,R}^n \frac{B_R}{B} - S_{moment,Z}^n \frac{B_Z}{B} - S_{moment,\phi}^n \frac{B_\phi}{B}
\]

On the other hand, the lack of an energy equation for the neutrals might introduce errors in the calculated ion energy source: as it will be discussed in the next chapter, the assumption of \( T_n = T_i \) equates to assume no net contribution of charge exchange collisions to the source and this might introduce non-negligible deviations when the reaction rate of the neutral-ion collisions becomes important. Nonetheless we can expect that when the two temperatures tend to each other, the errors introduced in the ion energy source term will become negligible.

The source terms are calculated under the assumption that only ionization, recombination and charge exchange events occur between plasma and neutrals.
3.5. Sources and coupling to Spledge2D

\[ S^p_n = n_n n_e \langle \sigma v \rangle_{iz} - n_n n_e \langle \sigma v \rangle_{rc} \]  
\[ S_G^n = n_n u_{n,||} n_e \langle \sigma v \rangle_{iz} - n_i u_{i,||} n_e \langle \sigma v \rangle_{rc} + \left( n_n u_{n,||} n_i - n_i u_{i,||} n_n \right) \langle \sigma v \rangle_{cx} \]  
\[ S^p_{E,x} = n_n n_e \langle \sigma v \Delta E \rangle_{iz} - n_n n_e \langle \sigma v \Delta E \rangle_{rc} \]  
\[ S_{E,i}^n = n_n E_n n_e \langle \sigma v \rangle_{iz} - n_i E_i n_e \langle \sigma v \rangle_{rc} + \left( n_n E_n n_i - n_i E_i n_n \right) \langle \sigma v \rangle_{rc} \]  
\[ S_{n}^n = -n_n n_e \langle \sigma v \rangle_{iz} + n_i n_e \langle \sigma v \rangle_{rc} \]  
\[ S_{mom,R}^n = -n_n u_{n,R} n_e \langle \sigma v \rangle_{iz} + n_i u_{i,R} n_e \langle \sigma v \rangle_{rc} + \left( n_n u_{n,R} n_i - n_i u_{i,R} n_n \right) \langle \sigma v \rangle_{cx} \]  
\[ S_{mom,Z}^n = -n_n u_{n,Z} n_e \langle \sigma v \rangle_{iz} + n_i u_{i,Z} n_e \langle \sigma v \rangle_{rc} + \left( n_n u_{n,Z} n_i - n_i u_{i,Z} n_n \right) \langle \sigma v \rangle_{cx} \]  
\[ S_{mom,\phi}^n = -n_n u_{n,\phi} n_e \langle \sigma v \rangle_{iz} + n_i u_{i,\phi} n_e \langle \sigma v \rangle_{rc} + \left( n_n u_{n,\phi} n_i - n_i u_{i,\phi} n_n \right) \langle \sigma v \rangle_{cx} \]

The reaction rates for the above mentioned collisions are computed with the help of the AMJUEL database [19]. The database contains tables for a number of reactions containing coefficients that are used in numerical fits to quickly compute the rates of the reactions. In this manuscript we will use the following rates:

- H.4 type of rate (as function of both density and temperature of projectile species) for reactions 2.1.5 and 2.1.8 for ionization and recombination respectively;
- H.2 type of rate (as function of only temperature of projectile species) for reaction 3.1.8FJ for charge exchange;
- H.10 type of rate (containing the energy loss averaged reaction rates, as function of both density and temperature) for reactions 2.1.5 and 2.1.8 for ionization and recombination respectively.

For H.4 type of data, the reaction rate can be calculated using the expression

\[ \ln \left( \langle \sigma v \rangle \right) = \sum_{n=0}^{N} \sum_{m=0}^{M} C_{n,m} \left( \ln(n_{proj}) \right)^m \left( \ln(T_{proj}) \right)^n \]  

where the reaction rate \( \langle \sigma v \rangle \) obtained is expressed in \([cm^3/s]\) when the density is in \([10^8 cm^3/s]\) and the temperature in \([eV]\), and \(C_{n,m}\) are the coefficients tabulated in the AMJUEL database. A similar expression is used for H.2 and H.10 \([cm^3/eV/s]\) types of data. Attention has to be paid to make sure that the error on the fit is small, as their range of validity is delimitated by minimum and maximum values for both density and temperature: as an example, fits for ionization and recombination reaction rates provide acceptable precision in the range \([0.1, 20000]eV\) and \([10^{14}, 10^{22}]m^{-3}\), and errors might be introduced for dense and cold plasmas outside of these limits.
To compute these reaction rates, and for the source terms themselves too, density, fluid velocity and temperature of ions, electrons and neutrals are required; furthermore, also the components of the magnetic field are required to compute the parallel component of the neutral fluid velocity and the \((R, Z, \phi)\) components of the ion fluid velocity.

Because Soledge2D and the fluid code use different grids, an interpolation step is required before the computation of the sources to transfer the plasma information to the fluid code or the neutrals information to the plasma solver, and a second interpolation is required after this to either pass the neutral sources to the neutral grid or the plasma sources on its mesh. Nonetheless, remembering that the fluid code is to be used as part of an hybrid code together with Eirene, the same grid is used for both neutral codes and, consequently, interpolation routines of plasma fields and back-interpolation routines of the plasma sources are already present making the communication between plasma and the neutrals easier, effectively using the same architecture built for Soledge2D-Eirene while only switching the neutral code from Eirene to the fluid one.

Figure 3.3: Plasma density \([m^{-3}]\) (top left), parallel ion Mach number (top right), electron temperature \([eV]\) (bottom left) and ion temperature \([eV]\) (bottom right) used as background for the fluid code test case.
3.6 Test cases

Having verified the implementation of the fluid code with the method of the manufactured solutions, we will present now a set of test cases in which we progressively introduced more and more complexity for both the numerical scheme and the transport of the neutral fluid. It should be noted before that the scenarios presented here were all in plasma regimes far from charge exchange dominated regimes. For this reason, we do not expect to recover quantitative agreement with the kinetic simulations and we are only interested in assessing whether the fluid code is able to produce physically reasonable results.

In the following, because the absolute value of the Knudsen number does not play any major role other than letting us assess the validity of the fluid description, we will use a simplified estimation of the Knudsen number

\[ Kn = \frac{v_{th}}{n_i\langle\sigma v\rangle_{ex}} L \]  

(3.59)

where we take the minor radius of the tokamak as a crude estimation of the macroscopic length \( L \). It is reminded that a fluid description is usually considered valid if the Knudsen number is much smaller than unity; as a distinction value for kinetic-fluid in this section, we will use the value \( Kn = 0.01 \).

3.6.1 Slab geometry

The fluid code has been first tested in slab geometry to remove the geometrical complexity. In this simplified geometry, the \( x \) direction mimics the parallel direction in a Tokamak, with the boundaries at \( x = 0 \) and \( x = 1 \) considered to be either divertor targets or limiter walls. The \( y \) direction, instead, is assumed to be along the radial direction, with the boundaries at \( y = 0 \) and \( y = 1 \) simulating the presence of, respectively, the first wall and the interface with the core.

A symmetric plasma background has been analytically generated to provide boundary conditions and sources for the fluid code. Figure 3.3 shows the plasma density, the parallel ion Mach number and the electron and ion temperature used in the test case.
The plasma shows a sheath limited regime behavior, with constant temperature and with a parabolic shape for the density along the parallel $x$ direction. Having assumed that the magnetic field is purely along the $x$ direction, a plasma particle flux arises only on the left $x = 0$ and right $x = 1$ boundaries, as shown in figure 3.4.

The plasma quantities were used as a fixed background to compute the recycling fluxes (with a recycling factor $R = 1$) and the sources for the fluid code [62]. The resulting neutral density is shown in figure 3.5, while the neutral flows are shown in figure 3.6.
As one would expect, the density peaks at the targets due to the recycling source and, in particular, near the upper corners of the domain where the ion particle flux reaches its maximum value; on the other hand, the absorption by the top boundary \( y = 1 \) and the ionization due to the high plasma density near the core interface drives the neutrals to flow from the solid walls towards the latter.

While the recycling flux is imposed to be along the normal to the solid surfaces, so purely horizontal for this simple case, we can see that most of the streamlines in figure 3.6 slightly turn towards the bottom \( y = 0 \) boundary, before accelerating towards the core when they reach the middle of the domain \( x = 0.5 \): this is due to the inhomogeneities in the ion (and thus neutral) temperature and in the magnitude of the recycling flux along the vertical direction, which drives a neutral pressure gradient directed towards the top.

### 3.6.2 TCV geometry

For a more demanding test, realistic TCV geometry and plasma were used. The plasma background was taken from a Soledge2D-Eirene simulation of shot #51325 [26] and, again, the plasma information, shown in figure 3.7, was used to compute the sources for the fluid code.

![Figure 3.7: From left to right, plasma density \( m^{-3} \), electron temperature \( eV \), ion temperature \( eV \) and parallel ion Mach number. The position of the puff is indicated by the red triangle on the bottom, and the entrance of the pump is shown by the white line on the right boundary.](image)

The resulting neutral density is shown in figure 3.8. Due to the openness of the TCV divertor, recycling neutrals flood the whole volume up to the core boundary. For
comparison the atom density computed by Eirene is also shown in the figure, and we can see that a good qualitative agreement is found [62]. The only remarkable difference is that the fluid code slightly underestimates the neutral density in front of both targets, which can be explained by the simplified assumptions introduced in the fluid code. The molecules that come off the walls in Eirene are in fact responsible for sources for the atoms mostly localized near the walls due to their small speed (comparable to the thermal velocity), which is completely missing in the fluid code; furthermore, the puff present in the Eirene simulations is also not present, further reducing the sources of atoms. Consequently, the underestimation of the density at the inner strike point reflects in a smaller influx of neutrals inside the closed flux lines region.

A comparison of the particle source is also shown in figure 3.9. In accord with the lower neutral density in the fluid code, the particle source is also slightly underestimated near the targets; furthermore, it has to be noted that Eirene also predicts some slight plasma source in the region above the core due to recycling in the top of the machine, which is not present in the fluid code.

This in turn shows how the fluid code is intrinsically unable to fully capture the transport features of a kinetic code in the regions far from the dense plasma, and consequently proves the need for a hybrid description in order to model the neutral transport in the whole machine. This is also evident from figure 3.10, which shows the Knudsen number of the atoms computed as 3.59.

Finally, it is interesting to see that the neutral fluxes (figure 3.11) qualitatively behave like in the slab case examined before: the neutrals born off recycling from the walls
3.6. Test cases

macroscopically flow towards the sink caused by the presence of the hot and dense core. The atoms freely stream in the whole divertor area, in accord with the neutral density, except of course along the vertical outer leg where a pressure gradient arises due to the plasma; finally, some geometric effects are visible in the upper outer corner where a recirculation vortex seems to appear due to change of slope of the solid wall.

3.6.3 WEST geometry

Finally, let us address a simulation that makes full use of the penalization technique of the plasma solver. In the following we will use the plasma background generated for realistic WEST Solved2D-Eirene simulations [63]. While the WEST geometry makes for a further step in the geometrical complexity of the wall, this also translates in an increase in difficulty for the underlying physics: the presence of a baffle, for example, receives a non-negligible flux of ions and thus actively modifies the recycling sources, but at the same time it also closes part of the divertor, possibly changing the flow patterns in the vicinity of the targets and in the pump duct below.

As such, we can already envision that the boundary conditions will play a large role in such an application, and due to their limited treatment in the fluid code might lead to important deviations from the kinetic solution.

Nonetheless, in order to seek a pertinent regime for the fluid description, three different cases were explored, with increasing puff rate: $1 \times 10^{21}$, $2 \times 10^{21}$ and $5 \times 10^{21} D/s$ for the
cases that we will call in the following "low", "medium" and "high puff rate" respectively. Except for the puff rate, all the other input parameters were kept constant:

- pure Deuterium plasma;
- magnetic configuration with a lower single null in the so-called far scenario [4];
- $P_{\text{SOL}} = 4 \text{ MW}$, evenly distributed between ions and electrons;
- "self-consistent" core boundary conditions (the plasma particle flux at the core interface is set at each iteration proportional to the neutral influx);
- no sputtering;
- unitary recycling coefficient $R = 1$;
- transport coefficients set to the ones used in [12] for similarity to an H-mode ASDEX upgrade discharge [14];

The puff was set in the private flux region, with $D_2$ molecules injected at a temperature of 0.03 eV. A pump was also set in the bottom of the machine, under the baffle, with an albedo set to 0.95, except for the "high puff rate" case where a value of 0.98 was used to facilitate the onset of detachment.

The results obtained by Soledge2D-Eirene in terms of plasma density, electron temperature and ion temperature for three cases are shown in figure 3.12. A description of Soledge2D-Eirene results for the three cases is beyond the scope of this chapter, but it can be found in reference [63].

The important thing to note is that, by increasing the puff rate we obtain a detached regime with possibly low Knudsen number. The Knudsen number estimated with equation 3.59 is shown in figure 3.13 for the three cases. It is evident that the domain of validity of the fluid description, highlighted with blue shades in the plot, even though almost non-existent in the "low puff rate" case where a value of 0.98 was used to facilitate the onset of detachment.

Nonetheless, even in the "high puff rate" case we see that the Knudsen number is relatively high in most of the computational domain, and a kinetic description should be retained to properly model the neutrals in these less collisional zones.

Finally, remembering that the fluid code assumes $T_n = T_i$ in the pressure gradient and in the calculation of the source terms, figure 3.14 shows the ratio of the atom temperature sampled by Eirene and the ion temperature computed by Soledge2D for the three cases. Going from the "low" to the "high puff rate" case the ratio $T_n/T_i$ gets closer to unity.
3.6. Test cases

Figure 3.12: Plasma density $[10^{20} m^{-3}]$ (top row), electron temperature [eV] (center row) and ion temperature [eV] (bottom row) for the "low", "medium" and "high puff rate" simulations in the left, center and right columns respectively.

Figure 3.13: Knudsen number for the D atoms for the "low", "medium" and "high puff rate" cases on the left, center and right respectively.
as the charge exchange collisions tend to equalize the temperatures of ions and neutrals. This gives us confidence that the assumption, at least nearing detached conditions, can be justifiable.

The plasma information has been used as a fixed background for the fluid code. Because of the lack of puffs in the fluid code contrarily to Eirene, the only source of atoms in these simulations is recombination, either as recycling of the ion flux at the solid walls (plotted in figure 3.15) or as volumetric reaction. However, since these simulations are run in "standalone" and molecules are not taken into account, the puffs should not have a relevant impact: the total ion flux (and thus the recycling atom flux) on the wall is in fact of the order of $\sim 2 \div 4 \times 10^{23} D/s$ while the highest puff rate is "only" $5 \times 10^{21} D/s$, two orders of magnitude smaller.

The results of the fluid simulations are shown in figure 3.16 in terms of neutral density, neutral particle source and neutral parallel momentum source.

For the "low puff rate" case, the particle source is always negative indicating that the predominant process is ionization due to the quite high plasma temperature, peaking at the strike points; a similar behavior is found for the "medium" case, even though a positive source due to volumetric recombination is found on the upper divertor, especially at the high field side; finally in the "high" case volumetric recombination takes over as the predominant process in the lower divertor as it detaches and the neutral density increases.

For the parallel momentum source the "low" and "medium" cases behave in a similar fashion, with neutral particles getting accelerated by the ion flux due to charge exchange collisions in front of the targets; in the "high puff rate" case, instead, it is again the source due to volumetric recombination that predominates, accelerating the neutrals and
3.6. Test cases

![Graph showing ion flux impacting the solid walls for "low", "medium" and "high puff rate" in red, blue and green respectively. A zoom of the lower divertor is also shown.]

Figure 3.15: Ion flux impacting the solid walls for the "low", "medium" and "high puff rate" in red, blue and green respectively. A zoom of the lower divertor is also shown.

![Graph showing neutral density, particle source, and parallel momentum source for the "low", "medium" and "high puff rate" case in the left, center and right columns respectively.]

Figure 3.16: Neutral density $[10^{17} m^{-3}]$ (top row), neutral particle source $[10^{22} m^{-3} s^{-1}]$ (center row) and neutral parallel momentum source $[10^{27} m^{-2} s^{-2}]$ (bottom row) computed by the fluid code for the "low", "medium" and "high puff rate" case in the left, center and right columns respectively.
conversely reducing the ion flux all along the divertor legs from the x-point to the targets.

As already explained, a kinetic model should be retained in these cases, except for small regions that proved highly collisionals; as such, we did not expect a particularly good agreement of the fluid code with the results extracted from Eirene. In fact, the fluid neutral density seems to be about an order of magnitude lower than the neutral density computed by Eirene, especially at the strike points, similar to what was found by [36]. Nonetheless, the code proves to be able to reasonably reproduce the kinetic results even in the high Knudsen number case and in a geometry as complex as the one presented here.

The deviations from the kinetic solutions can, in fact, be explained by the simplifications introduced in the fluid model, mainly in the absence of molecules and in the crude treatment of the recycling boundary condition: the implementation of boundary conditions in accordance with the underlying kinetic description has already been shown to be crucial for the accuracy of the fluid models [34].

All in all, we can argue that the high geometrical flexibility of the fluid code more than compensates the differences introduced in the solution. It is evident that the magnitude of the differences could be alleviated by easing some of the approximations, like for example introducing more detailed boundary conditions or an additional fluid species to take into account molecules, at the expense of the computational cost of the code of course. But is it necessary?
Chapter 4

Hybrid fluid-kinetic description

Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>111</td>
</tr>
<tr>
<td>4.2</td>
<td>Proposed hybrid model</td>
<td>113</td>
</tr>
<tr>
<td>4.3</td>
<td>Evaporation and condensation processes</td>
<td>116</td>
</tr>
<tr>
<td>4.4</td>
<td>Coupling Eirene and the fluid code</td>
<td>121</td>
</tr>
<tr>
<td>4.5</td>
<td>Discussion on the temperature closure</td>
<td>128</td>
</tr>
<tr>
<td>4.6</td>
<td>ITER simulations: hybrid vs fully-kinetic</td>
<td>134</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Simulations setup</td>
<td>134</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Hybrid results</td>
<td>139</td>
</tr>
<tr>
<td>4.6.3</td>
<td>Sources by plasma-neutrals interactions</td>
<td>141</td>
</tr>
<tr>
<td>4.6.4</td>
<td>Particles and energy fluxes on the wall</td>
<td>145</td>
</tr>
<tr>
<td>4.6.5</td>
<td>Computational performance</td>
<td>146</td>
</tr>
</tbody>
</table>
4.1 Introduction

In the previous chapters we have shown the major problem related to the modeling of neutral particles for scrape-off layer plasmas for the larger next generation of Tokamaks, when we access the detached regime.

On one side, the kinetic Monte Carlo procedure often employed proves to be the bottleneck for plasma-neutral simulations of these regimes due to the large number of scattering events occurring in the cold and dense zones in the vicinity of the recombination zones.

On the other, a fluid description is able to speed up the calculations but the atoms behave kinetically in most of the device, so applying a fluid code in the whole computational domain inevitably introduces errors due to the approximations of the model. Of course one might develop more refined fluid models by relaxing some of the approximations: expanding the model to three dimensions, introducing additional fluids for molecules, applying more refined closures (for example by allowing for additional corrections to a Maxwellian), imposing boundary conditions derived from the underlying kinetic physics,... Nonetheless, due to the large range of values of the Knudsen number of the atoms in the scrape-off layer, a fluid description can not be applied for the simulations of the neutral species in the whole Tokamak volume and the approximations introduced in a fluid model can only be valid in limited regions.

Here is where hybrid models come into play. The idea behind these mixed fluid-kinetic descriptions is to synergize the speed of a fluid code with the precision guaranteed by a kinetic one: in a zone of the computational grid where the neutrals undergo a large number of ion-neutral and neutral-neutral collisions we expect the two species to tend towards thermalized equilibrium distribution functions, so the approximations introduced by the fluid code should be valid and a kinetic description is not required; on the other hand, in zones where neutrals only seldom collide with surrounding particles, the approximations of the fluid code might introduce non negligible errors in the solution, and using a kinetic code is imperative.

We can thus already envision a conceptually straightforward implementation of a hybrid model, simply by partitioning the computational mesh in two non-intersecting domains: one in which the fluid code is run, and the other where the kinetic code is applied. This kind of approach has already been applied in fluid-dynamics, shown for example in figure 4.1, where the interface between the continuum domain (fluid areas) and the particle domain (kinetic areas) is determined by a continuum breakdown parameter, like for example the Knudsen number.

It is evident how, in this kind of hybrid models, the interface between the two domains plays a role of the utmost importance. Boundary conditions are thus required for both
Figure 4.1: Example of computational grid used in reference [57] for a subsonic flow over a micro-scale airfoil, showing the two different domains and the interface identified between the particle and continuum domain depending on the continuum breakdown parameter $B$.

The particle-kinetic area and for the continuum-fluid one: while from the fluid side of the boundaries we can just impose quantities (for example temperature, velocity, density, particle flux, etc...) which can be derived from the kinetic side simply as moments of the distribution function, on the kinetic side we can only collect average information and we are missing information related to the distribution function of the discrete particles to be followed. This problem is often solved by assuming that the particles that cross particle-continuum interfaces from the fluid side retain the pdf used to close the fluid equations, so for example a Maxwellian with corrective terms, but it is clear that this introduces another approximation in the model.

Clearly, the numerical performance of this approach depends on the ratio of fluid to kinetic grid elements*, but it has to be taken into account that, as the interface can move during the simulation, there is a non-negligible overhead due to the need to either re-mesh or re-compute the position of the boundaries.

A second approach for a hybrid method is instead able to circumvent this difficulty, by completely removing the need of interfaces between fluid and kinetic regions. To do

*This is strictly true for the fluid part of the hybrid model, as the computational time required by the former will scale, in first approximation, linearly with the number of fluid elements; for the kinetic Monte Carlo part of the code, instead, for a given number of histories this only depends on how large the highly collisional region is.
so, both fluid and kinetic codes need to be solved in the whole domain, and the coupling between the two has to be performed by some other mean. An example of this kind of approach has been presented in [35] with a Delta-f method. The particle distribution function is divided in a fluid and a kinetic part, with the fluid part assumed to be a Maxwellian distribution function. The moments of the fluid part of the distribution function thus evolve following fluid equations, whose closure does not need any assumption but it is rather implemented adding correction terms computed by the kinetic part of the distribution. For example, in the first moment equation (the momentum equation), the second moment of the distribution function has to be closed, so the pressure becomes

\[ p_n = p_{n,\text{fluid}} + p_{n,\text{kinetic}} = n_{\text{fluid}} T_i + m \int v^2 f_{\text{kinetic}} dv. \]

While the computational cost of the fluid code tends to increase with respect to the previous approach, due to the fact that the code has to be now applied to the whole domain, the charge exchange collisions in this hybrid model become pure absorption event for the kinetic code, significantly reducing the computational time required per particle.

### 4.2 Proposed hybrid model

In this section, we will present the hybrid model that we decided to develop and implement.

The idea behind this description is based on [38] and it envisions the distinction of two different populations, or rather phases, for the atom species present in the tokamak: a fully fluid phase, assumed to be in local thermodynamical equilibrium with the plasma ions, and fully kinetic phase, with its own distribution function. These two populations can collide independently with the plasma around them, but they can also "change phase".

To better understand, let us draw a paragon with a closed thermodynamic system, like a sealed beaker, that contains a liquid and the vapor of the same substance. Being the system closed the total number of particles must remain constant, but the liquid phase can freely evaporate into vapor and similarly the gas population can freely condense into the liquid beneath. As we can see from a typical T-s diagram (like the one shown in figure 4.2 for water), an equilibrium point exists in function of the temperature, pressure and volume of the system: if at a given temperature and pressure the liquid fraction is too high with respect to the one at equilibrium, more molecules will pass to the gas phase than the ones that will get stuck to the liquid surface, and the opposite is true in the opposite scenario.

Coming back to the neutral atoms in a tokamak, the liquid phase in the analogy
represents the fluid population of the neutrals while the vapor phase represents the kinetic part (even if both are gases in a tokamak!); the role of the temperature, pressure and volume in shifting the equilibrium towards a phase or the other is played in this case by the Knudsen number of the atoms in the surrounding plasma: if the Knudsen number is low enough, we want the atom species to be populated mostly by the fluid phase, while if the Knudsen number is higher we would like the population to be mostly kinetic.

To do so, in addition to the usual neutral-ion and neutral-electron collisions, the two atom phases can undergo two fictitious processes of condensation and evaporation that we will define as the process of a kinetic atom to become a fluid one and the process of a fluid atom to become a kinetic one respectively. If we go back to the Boltzmann equation for the atoms 1.44, we can write the total distribution function for the atoms as the sum of the distribution functions for the two different phases \( f(\vec{r}, \vec{v}, t) = f_f(\vec{r}, \vec{v}, t) + f_k(\vec{r}, \vec{v}, t) \) (with subscripts \( f \) and \( k \) for fluid and kinetic respectively). Upon noting that the Boltzmann equation is linear in the distribution function, we can thus write two separate equations for the two phases.
4.2. Proposed hybrid model

\begin{equation}
\frac{\partial f_k}{\partial t} + \vec{v}_k \cdot \vec{\nabla} f_k = \alpha f_i n_e \langle \sigma v \rangle_{rc} - f_k n_e \langle \sigma v \rangle_{iz} + \\
( f_i n_e - f_k n_i ) \langle \sigma v \rangle_{ex} + \nu_{evap} f_f - \nu_{cond} f_k
\end{equation}

(4.1)

\begin{equation}
\frac{\partial f_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} f_f = (1 - \alpha) f_i n_e \langle \sigma v \rangle_{rc} - f_f n_e \langle \sigma v \rangle_{iz} + \\
( f_i n_f - f_f n_i ) \langle \sigma v \rangle_{ex} - \nu_{evap} f_f + \nu_{cond} f_k
\end{equation}

(4.2)

where \( \nu_{evap} \) and \( \nu_{cond} \) are the reaction rates for the fictitious processes of evaporation and condensation; of course to satisfy the conservation of mass, momentum and energy the sum of the terms of evaporation and condensation for the two phases is 0. Furthermore, a factor \( \alpha \) (defined as a real number between 0 and 1) is introduced to "decide" whether the neutrals born due to plasma recombination will fall in the fluid population or in the kinetic one; again, to ensure conservation of mass, momentum and energy during a recombination event, the sum of the terms for the fluid and kinetic population must be equal to the term in the plasma equations \( (1 - \alpha) f_i n_e \langle \sigma v \rangle_{rc} + \alpha f_i n_e \langle \sigma v \rangle_{rc} = f_i n_e \langle \sigma v \rangle_{rc} \)

We would like to point that up to this point the global system is still exact, in the sense that we would be able to characterize all the properties of the whole atoms population with an error identically equal to zero (except of course for discretization and/or statistical errors introduced by the solvers) as the choice of the reaction rates for evaporation and recombination is still completely arbitrary.

We now take the moments of equation 4.2, similar to what has been done in the previous chapters, limit ourselves to the first two moments introducing the approximation

\begin{equation}
f_f(\vec{r}, \vec{v}, t) = n_f \left( \frac{2\pi k_B T_f}{m} \right)^{-3/2} \exp \left( -\frac{m(\vec{v} - \vec{u}_f)^2}{2k_B T_f} \right) \end{equation}

(4.3)

that is, the distribution function of the fluid part of the atoms population is a drifting Maxwellian. By doing so, the first two moments of equation 4.2 can be written as

\begin{equation}
\frac{\partial n_f}{\partial t} + \vec{\nabla} \cdot (n_f \vec{u}_f) = S_{iz}^n + (1 - \alpha) S_{rc}^n + S_{evap}^n + S_{cond}^n
\end{equation}

(4.4)

\begin{equation}
\frac{\partial n_f \vec{u}_f}{\partial t} + \vec{\nabla} \cdot (n_f \vec{u}_f \vec{u}_f) + \vec{\nabla} p_f = S_{iz}^{mom} + (1 - \alpha) S_{rc}^{mom} + S_{ex}^{mom} + S_{evap}^{mom} + S_{cond}^{mom}
\end{equation}

(4.5)

where \( p_f = n_f T_f k_B \) is the pressure of the fluid neutrals.

Some information for the fluid part of the population is still missing, its temperature \( T_f \). Here a second approximation is needed, so we assume that the fluid neutrals are
in equilibrium with the background ions and thus $T_f = T_i$ is given by the ion energy equation. The underlying idea is that the fluid description is tailored to be used in charge exchange dominated regimes, where the high number of neutral-ion collisions will tend to thermalize the two species to similar temperatures.

Finally, the only new actors in equations 4.4 and 4.5 are the moments of the evaporation and condensation terms

$$
S_n^{\text{evap}} = - \int \nu_{\text{evap}} f_f d\vec{v} = -\nu_{\text{evap}} n_f \\
S_n^{\text{mom evap}} = - \int \bar{v} \nu_{\text{evap}} f_f d\vec{v} = -\nu_{\text{evap}} n_f \bar{u}_f \\
S_n^{\text{cond}} = \int \nu_{\text{cond}} f_k d\vec{v} = \nu_{\text{cond}} n_k \\
S_n^{\text{mom cond}} = \int \bar{v} \nu_{\text{cond}} f_k d\vec{v} = \nu_{\text{cond}} n_k \bar{u}_k
$$

having assumed that the evaporation and condensation reaction rate $\nu_{\text{evap}}$ and $\nu_{\text{cond}}$ are independent of $\bar{v}$.

From the point of view of the plasma, the ions and electrons will experience collisions with both populations resulting in effective sources for particles, parallel momentum and energy that are simply given by the sum of the source terms due to the fluid part and to the kinetic part of the atoms. Just as an example, the particle source for the plasma can thus be written as

$$
S_n = S_{n,f} + S_{n,k} = [n_e n_f \langle \sigma v \rangle_{iz} - (1 - \alpha)n_i n_e \langle \sigma v \rangle_{rc}] + [n_e n_k \langle \sigma v \rangle_{iz} - \alpha n_i n_e \langle \sigma v \rangle_{rc}] = (n_f + n_k) n_e \langle \sigma v \rangle_{iz} - n_i n_e \langle \sigma v \rangle_{rc} = n_n n_e \langle \sigma v \rangle_{iz} - n_i n_e \langle \sigma v \rangle_{rc}
$$

4.3 Evaporation and condensation processes

The hybrid model described in the previous section heavily relies on the evaporation and condensation processes introduced. The fact that these processes are just fictitious allows us, on the other hand, to tailor their reaction rates for our needs.

We will divide both rates in two terms $\nu = \nu^* \cdot F$: the first one $\nu^*$ contains the magnitude of the rate (units $[s^{-1}]$), while the second one $F$ is a mask factor in the range $[0, 1]$ that lets us scale the rate depending on the local Knudsen number. To better
4.3. Evaporation and condensation processes

Figure 4.3: Mask factor for condensation and evaporation as functions of the Knudsen number, in red and blue respectively.

understand:

- in low Knudsen areas, we would like the fluid phase to predominate on the kinetic phase, so we would like the evaporation rate to be negligible with respect to the condensation;
- in high Knudsen areas, we would like the kinetic phase to predominate, so we would like the condensation rate to vanish.

If we look at the system in function of the Knudsen number, as shown in figure 4.3, in the middle of the above mentioned areas, the two processes of evaporation and condensation can coexist. A risk exists though that an atom in such a region could experience too many phase changes and "oscillate" between the fluid and kinetic populations, possibly creating instabilities; to limit this, we enforce a buffer zone between two limiting values of the Knudsen number.

In the following we will define the mask factors $F_{\text{cond}}$ and $F_{\text{evap}}$ as

$$F_{\text{cond}} = \frac{1 - \tanh\left(\frac{Kn - Kn_{\text{lim}}}{s_{\text{cond}}}\right)}{2}$$

$$F_{\text{evap}} = \frac{1 + \tanh\left(\frac{Kn - Kn_{\text{buf}}}{s_{\text{evap}}}\right)}{2}$$

where $s_{\text{cond}}$ and $s_{\text{evap}}$ are two parameters used to control the sharpness of the hyperbolic tangent functions.
Let us start by focusing on the condensation rate: for Knudsen number tending to zero, the mask factor tends to 1 to ensure that the condensation source has its maximum value; the mask should keep its unitary value as the Knudsen number increases and as soon as the latter gets closer to the first limit, the value of the mask quickly drops so that the condensation of kinetic neutrals to fluid neutrals in the buffer zone is negligible; finally, before reaching the second limit value, the condensation should be identically zero, thus ensuring that in the kinetic region no fluid neutrals are generated and no approximations are made in the following kinetic region. A similar picture can be painted for the evaporation mask: for Knudsen number tending to zero, the mask factor tends also to zero to ensure no kinetic neutral is created in the fluid zone, up to the first limiting value; upon reaching the buffer zone, the evaporation mask quickly increases in the vicinity of the second limit value, so that in the following kinetic region the fluid neutrals will strongly undergo evaporation processes.

As such, the buffer zone acts as a zone where Eirene and the fluid code are only weakly coupled. While the first limiting value $Kn_{lim}$ (the one at smaller Kn) is more or less fixed to control the errors introduced by the fluid approximation, the second value $Kn_{buf}$ can be changed more easily in order to change the size of the buffer at the interfaces between fluid and kinetic areas. As a reference, it was found in reference [5] that continuum Navier-Stokes equations in transitional hypersonic flows tend to fail for gradient length local Knudsen number greater than 0.05; more recent results like reference [53] are more optimistic, showing that hybrid codes that use a breakdown value of the Knudsen number of 0.1 still produce simulation results in almost perfect agreement with fully kinetic codes.

Having defined the mask factors $F$, all that remains to be chosen is the magnitude $\nu^*$ of the condensation and evaporation rates. The only constraint here it is actually a performance related one: as the charge exchange collisions are what mainly slow down Eirene, we can just impose the condensation process to be more probable than a charge exchange collision with an ion. For this reason, we impose $\nu_{cond}^* = \nu_{evap}^* = m \cdot n_i(\sigma v)_{cx}$ where $m$ is multiplicative factor higher than one (the value generally used in the following is $m = 10$). For simplicity, we choose the same magnitude also for the evaporation rate.

Knowing now the full condensation and evaporation rates, we can analytically estimate the quantity of fluid and kinetic neutrals that we can expect at any given value of Knudsen number. If we imagine that there is no transport and that the two atom populations can only change phase, we can write a simplified system
4.3. Evaporation and condensation processes

\[ \frac{\partial n_f}{\partial t} = \nu_{\text{cond}} n_k - \nu_{\text{evap}} n_f \]  
\[ \frac{\partial n_k}{\partial t} = -\nu_{\text{cond}} n_k + \nu_{\text{evap}} n_f \]  
\[ \text{(4.10)} \]
\[ \text{(4.11)} \]

At equilibrium, the two derivatives tend to zero, and we obtain the following system (written in matrix form)

\[ \begin{bmatrix} \nu_{\text{cond}} & -\nu_{\text{evap}} \\ -\nu_{\text{cond}} & \nu_{\text{evap}} \end{bmatrix} \begin{bmatrix} n_k \\ n_f \end{bmatrix} = 0 \]  
\[ \text{(4.12)} \]

Evidently, the two equations are not independent and infinite solutions can be found for the system. Nonetheless, we can extract some information from it. If we consider that the sum of the two densities \( n_n = n_f + n_k \) is constant as the two processes only change the phase of the neutrals, we can take any of the two equations and divide it by \( n_n \):

\[ \nu_{\text{cond}} \frac{n_k}{n_n} - \nu_{\text{evap}} \frac{n_f}{n_n} = \nu_{\text{cond}} c_k - \nu_{\text{evap}} c_f = 0 \]  
\[ \text{(4.13)} \]

where \( c_k \) and \( c_f \) are the concentrations of the kinetic and fluid phases in the atom species. Besides, we know that the sum of the two concentrations must be 1, so that in the end

\[ c_f = \frac{\nu_{\text{cond}}}{\nu_{\text{cond}} + \nu_{\text{evap}}} = \frac{F_{\text{cond}}}{F_{\text{cond}} + F_{\text{evap}}} \]  
\[ \text{(4.14)} \]
\[ c_k = \frac{\nu_{\text{evap}}}{\nu_{\text{cond}} + \nu_{\text{evap}}} = \frac{F_{\text{evap}}}{F_{\text{cond}} + F_{\text{evap}}} \]  
\[ \text{(4.15)} \]

where the choice \( \nu_{\text{cond}}^* = \nu_{\text{evap}}^* \) leaves the concentrations proportional to the mask functions only. It does not surprise that, once equilibrium is reached, the ratio of the two phases is indeed exactly equal to the ratio of the respective processes that generate them. Using the mask functions shown in figure 4.3, the results of equations 4.14 and 4.15 are shown in figure 4.4, and as one would expect, each phase dominates in its own region.

These results should be kept in mind only as an approximation, as in reality the neutrals can be transported away from the point where they are generated; furthermore, evaporation and condensation processes will also compete with neutral-electron and neutral-ion collisions, so that these concentrations may not be exactly verified during a simulation. As such, we could slightly refine the simple analytic system by including ionization and recombination. Let us start by writing the system for the plasma species
Chapter 4. Hybrid fluid-kinetic description

Figure 4.4: Concentrations of fluid and kinetic neutrals as functions of the Knudsen number, in red and blue respectively.

and the "total" atom species given by the sum of the two phases

\[
\frac{\partial n_e}{\partial t} = n_e n_n \langle \sigma v \rangle_{iz} - n_e n_e \langle \sigma v \rangle_{rc} \quad (4.16)
\]

\[
\frac{\partial n_n}{\partial t} = -n_e n_n \langle \sigma v \rangle_{iz} + n_e n_e \langle \sigma v \rangle_{rc} \quad (4.17)
\]

At equilibrium, we obtain again a system of non independent equations, but any of the two lets us write the plasma and total neutral densities in function of the sum of the two \( n_0 = n_e + n_n = n_e + n_f + n_k \), that will be constant

\[
n_e = n_0 \frac{\langle \sigma v \rangle_{iz}}{\langle \sigma v \rangle_{iz} + \langle \sigma v \rangle_{rc}} \quad (4.18)
\]

\[
n_n = n_0 \frac{\langle \sigma v \rangle_{iz}}{\langle \sigma v \rangle_{iz} + \langle \sigma v \rangle_{rc}} \quad (4.19)
\]

Let us now write the equations for the two phases separately:

\[
\frac{\partial n_f}{\partial t} = -n_e n_f \langle \sigma v \rangle_{iz} + (1 - \alpha) n_e n_e \langle \sigma v \rangle_{rc} + \nu_{\text{cond}} n_k - \nu_{\text{evap}} n_f \quad (4.20)
\]

\[
\frac{\partial n_k}{\partial t} = -n_e n_k \langle \sigma v \rangle_{iz} + \alpha n_e n_e \langle \sigma v \rangle_{rc} - \nu_{\text{cond}} n_k + \nu_{\text{evap}} n_f \quad (4.21)
\]

As before, let us look at the equilibrium, and using any of the equations we will obtain
4.4 Coupling Eirene and the fluid code

\begin{align*}
n_f &= \frac{(1 - \alpha)n_e \langle \sigma v \rangle_{rc} + \nu_{\text{cond}} n_n}{\nu_{\text{cond}} + \nu_{\text{evap}} + n_e \langle \sigma v \rangle_{iz}} \quad (4.22) \\
n_k &= \frac{\alpha n_e \langle \sigma v \rangle_{rc} + \nu_{\text{evap}} n_n}{\nu_{\text{cond}} + \nu_{\text{evap}} + n_e \langle \sigma v \rangle_{iz}} \quad (4.23)
\end{align*}

With a bit of algebra and substituting the expressions 4.18 and 4.19, the concentrations of electrons, fluid neutrals and kinetic neutrals can be expressed as:

\begin{align*}
c_e &= \frac{\langle \sigma v \rangle_{iz}}{\langle \sigma v \rangle_{iz} + \langle \sigma v \rangle_{rc}} \\
c_f &= \frac{(1 - \alpha)\langle \sigma v \rangle_{iz} + m F_{\text{cond}}}{\langle \sigma v \rangle_{iz} + m F_{\text{evap}} + \langle \sigma v \rangle_{rc}} \quad (4.24) \\
c_k &= \frac{\alpha \langle \sigma v \rangle_{iz} + m F_{\text{evap}}}{\langle \sigma v \rangle_{iz} + m F_{\text{evap}} + \langle \sigma v \rangle_{rc}} \quad (4.25)
\end{align*}

First of all, we see that the concentrations of fluid and kinetic neutrals are both zero if no recombination occurs; this of course if due to the assumption of no transport, as wall recycling cannot play any role in this simplified description. On the other hand we see that for \( \langle \sigma v \rangle_{iz} \ll \langle \sigma v \rangle_{cx} \), that is when the neutral-electron collisions are negligible with respect to evaporation and condensation processes, the ratio of fluid neutrals to kinetic neutrals tends to the same value \( \frac{F_{\text{cond}}}{F_{\text{evap}}} \) as before; on the other hand, if \( \langle \sigma v \rangle_{iz} \gg \langle \sigma v \rangle_{cx} \), the ratio tends to the value \( \frac{1 - \alpha}{\alpha} \), that is the ratio of the fractions of neutrals born of recombination in the two phases.

4.4 Coupling Eirene and the fluid code

Having described the hybrid model, let us finally talk about its implementation in the Soplege2D-Eirene package.

As explained, we want to have two different populations of atoms: one fully fluid, that evolves with equations 4.4 and 4.5, and one fully kinetic, that evolves with equation 4.1. We will use the fluid code described in chapter 3 for the evolution of the thermalized atoms, while we will use Eirene for the remaining kinetic part of the distribution function of the atoms. Molecules and "test ions" \( (D_2^+) \) will be treated fully kinetically by Eirene in this manuscript. Due to the fact that, at least in theory, the two phases could coexist in any part of the vessel, the two codes will have to be run together in the
whole computational domain. Furthermore, as the two codes have to be coupled to one another, we will use the same triangular grid required by Eirene as shown in figure 4.5.

![Figure 4.5: Zoom on the divertor region of a triangular mesh generated for Eirene and for the fluid code for ITER geometry.](image)

This will have the advantage of not requiring any interpolation between the two codes, while requiring only one interpolation of the underlying plasma.

We would like to point out that this two-phase model could effectively be considered as a "hybrid" between the two hybrid strategies introduced in section 4.1. On one hand, the subdivision of the distribution function for the atoms in two populations is similar to the underlying hypothesis of a delta-f model, with the difference that the \( \delta f \) (the kinetic part of the distribution function) is in our case a proper vdf and, consequently, cannot be negative.

On the other hand as we saw in the previous section, the two additional reactions of condensation and evaporation are effectively used to delimit the two populations in their respective domain of validity, and the model can then also be considered as a penalization technique (immersed boundary): the two codes are run on the same grid, but the boundary of their domains are modeled inside the equations themselves through mask functions, which in our case are the sources by evaporation and condensation.
A schematic diagram of the algorithm of the hybrid model is shown in figure 4.4.

If we do not take into account the first iteration, where most source terms are still uninitialized, any other iteration can be divided in three steps. The first step is the plasma background evolution: Soledge receives 2D maps of source terms for the plasma density, parallel momentum and electron and ion energy equations, defined on the quadrilateral mesh. Furthermore the neutral particles flux at the core boundary is also transferred to Soledge, which can be re-injected as a boundary condition for the plasma density assuming that all neutrals that cross this boundary will get ionized. After the iteration in Soledge, the 2D maps of the plasma fields are frozen and interpolated to the triangular
grid. At this point, the plasma background is used to compute the Knudsen number in order to decide whether each triangle has to be considered kinetic or fluid, or if a buffer zone has to be prescribed.

The Knudsen number in the following is estimated as

\[
K_n = \frac{\lambda}{L} = \frac{v_{th}}{n_e \langle \sigma v \rangle_{cx}} \frac{n_e}{\max(\nabla n_e, \nabla \perp n_e)}
\] (4.27)

The mean free path of the neutrals is approximated as the ratio of the thermal speed \(v_{th} = \sqrt{2kT_i/m}\) having assumed that \(T_n = T_i\) and the charge exchange reaction rate; as the macroscopic length we take an approximation of the gradient length of the plasma density.

In figure 4.6 the Knudsen number computed as equation 4.27 is shown for an ITER simulation, which will be described in the following. Even though the estimated value for the Knudsen number in the divertor region seems to be correct, identifying fluid areas all along the legs due to the detached plasma background, fluid areas are also identified in zones where one would instead expect the fluid description not to hold:

- inside the separatrix, in the confined plasma, both the plasma density and the charge exchange reaction rate are high, resulting in small mean free paths, but equation 4.27 does not take into account the fact that the ionization is more relevant than charge-exchange collisions due to the high temperature;

- near the pump entrances below the dome, where instead almost no plasma is present and as such the gradient length tends to infinity, resulting again in a Knudsen number tending to 0;

- finally, in the far scrape-off layer above the baffle on the high field side, due to a combination of both effects.

To make sure that no condensation occurs in these areas, the condensation and evaporation rates are thus modified multiplying them by some other mask factors, \(\nu_{\text{cond}} = \nu_{\text{cond}}^* \cdot F \cdot F_{\text{dens}} \cdot F_{\text{grad}} \cdot F_{\text{cx}} \cdot F_{\text{iz}}\) for example for condensation. These new mask factors take into account, respectively:

- areas where the plasma density is too low, \(F_{\text{dens}} = 0\) if \(n_e \leq n_{\text{min}}\);

- areas where the gradient length is too low, \(F_{\text{grad}} = 0\) if \(\nabla n_e \leq (\nabla n_e)_{\text{min}}\) (mostly imposed to avoid divisions by zero);

- areas where the charge exchange reaction rate is too low, \(F_{\text{cx}} = 0\) if \(n_i \langle \sigma v \rangle_{\text{cx}} \leq (n_i \langle \sigma v \rangle_{\text{cx}})_{\text{min}}\);
4.4. Coupling Eirene and the fluid code

Figure 4.6: "Unprocessed" Knudsen number computed during an ITER simulation, on the left, and respective fluid, buffer and kinetic areas identified using $K_{lim} = 0.05$ and $K_{buf} = 0.1$, on the right.

- areas where the charge exchange does not predominate over ionization, $F_{iz} = 0$ if

$$\frac{\langle \sigma v \rangle_{ce}}{\langle \sigma v \rangle_{iz}} \leq \left( \frac{\langle \sigma v \rangle_{ce}}{\langle \sigma v \rangle_{iz}} \right)_{min}.$$  

The above pre-factors are shown in figure 4.7, for $n_{min} = 10^{18} \text{m}^{-3}$, $(\nabla n_e)_{min} = 10^{-5} \text{m}^{-4}$, $(n_i \langle \sigma v \rangle_{ce})_{min} = 10^{-10} \text{s}^{-1}$ and $(\langle \sigma v \rangle_{ce})_{min} = 5$ as an example, and a "processed" Knudsen number that takes into account them is shown in figure 4.8.

The minimum values are set as inputs for the simulation, thus increasing the number of free parameters. Nonetheless, as evident, the most important pre-factor is the pre-factor $F_{iz}$ for the ratio between charge exchange and ionization reaction rates. This is in line with the fact that the fluid code has been tailored for charge exchange dominated regimes, and as such this pre-factor lets us ensure the precision of the hybrid model by preventing condensation in not pertinent areas.

Once the "un-processed" Knudsen number and all the pre-factors are computed, the condensation and evaporation reaction rate are calculated. Eirene is then called taking as input from the other codes:

- from the current iteration of Soledge2D, interpolated arrays of plasma density, electron and ion temperature and ion parallel velocity in the whole domain;

- from the current iteration of Soledge2D, the plasma particle flux impacting on the wall elements (or rather their values at the sheath entrance);
Figure 4.7: Pre-factors used to modify the condensation and evaporation rates in order to impose kinetic zones whenever, in order from left to right, the plasma density, the charge exchange rate, the plasma density gradient length or the ratio between charge exchange and ionization rates is too low.

- from the previous iteration of the fluid code, the evaporation source in the whole domain due to fluid neutrals changing phase;

- from the previous iteration of the fluid code, the three components of the fluid velocity and the temperature of the fluid neutrals;

- from the previous iteration of the fluid code, the fluid neutrals particle flux impacting on the wall elements.

The first two inputs for Eirene are the same as in the case of a regular (fully-kinetic) Soledge2D-Eirene simulation, as they are used by Eirene to compute the volumetric and surface sources; the only change here, is the fact that the volumetric recombination source is scaled down by the factor $\alpha$, as already explained previously. The other three inputs, provided to Eirene by the fluid code results of the previous iteration, are used to couple the two codes together. While the first one is self explanatory, Eirene also needs to know the distribution function of the neutrals born of evaporation: because of the underlying approximation for the fluid code, we make the assumption that these kinetic neutrals coming from fluid ones have a Maxwellian distribution function, thus the need for the
4.4. Coupling Eirene and the fluid code

Figure 4.8: "Processed" Knudsen number on the left and respective fluid, buffer and kinetic areas identified using $K_{lim} = 0.05$ and $K_{buf} = 0.1$, on the right.

temperature and the fluid velocity

$$f_{k,\text{evap}} = f_f = Ce^{-\frac{m(v-v_f)^2}{2k_BT_f}}$$ (4.28)

Finally, as shown in the previous chapter 3, the treatment of the recycling and reflection processes in the fluid code has been purposely dealt with in an approximated way. In fact, Eirene is already able to more accurately treat the surface sources through the TRIM database, taking into account the reflection channel passing through both atoms and molecules. Furthermore, it has to be noted that a "first flight" kinetic region is always present right in front of the solid surfaces.

For these reasons, all fluid neutrals impacting on the walls are reflected in Eirene as kinetic atoms (or molecules): from the point of view of the fluid code, the walls are thus completely absorbing surfaces ($R_i = R_n = 0$) and the particle flux driven to the walls is thus transferred to Eirene.

The condensation process in Eirene has been implemented as an additional charge exchange collision between a kinetic neutral and an ion, resulting in a "bulk" fluid neutral and a "bulk" ion

$$D^{0,k} + D^+ \rightarrow D^{0,f} + D^+$$ (4.29)

with the magnitude of the reaction rate taken as a multiple of the local reaction rate of
"ordinary" charge exchange. This way a kinetic neutral that enters a fluid zone will have a higher probability of undergoing a condensation "collision" than a charge exchange one.

Once Eirene finishes tallying the sources, the fluid code is finally called. Like Eirene, it receives the information on the plasma background (with the only change in the eventual volumetric recombination source, which is scaled down by $1 - \alpha$), but it does not receive the plasma fluxes on the walls as Eirene has already taken care of the recycling. On the other hand, the fluid code also receives from Eirene the condensation sources of particles and momentum in the computational grid. The same boundary conditions as in a fully-fluid simulation are used for the core surface; on the solid walls (and for the pumping surfaces, which are effectively treated as segments of wall), instead, the recycling coefficient for the ions and the reflection coefficient for the fluid atoms are set to zero $R_i = R_{n,f} = 0$ since all recycling at the surface is handled by Eirene, as explained.

If we assume that the reflection coefficient for both kinetic and fluid atoms is the same, this then translates in a set of boundary conditions

$$
\Gamma_k^+ = R_i \Gamma_i^- + R_{n,k} \Gamma_k^- + R_{n,f} \Gamma_f^- = R_i \Gamma_i^- + R_n (\Gamma_k^- + \Gamma_f^-) \quad \text{for Eirene}
$$

$$
\Gamma_f^+ = 0 \quad \text{for the fluid code} \quad (4.30)
$$

with the usual choice of positive and negative fluxes with respect to the outgoing normal.

After the iteration of the fluid code, the evaporation source and the flux of fluid neutrals on the wall are stored for the next time step, and the fluid sources for the plasma are added to the kinetic ones computed by Eirene.

### 4.5 Discussion on the temperature closure

As discussed in the previous sections, for the fluid population we made the hypothesis of a Maxwellian distribution thermalized with the surrounding ions.

In figure 4.9 is shown the distribution of the error made in the temperature by assuming $T_{n,f} = T_i$, estimated as the relative error between the temperature of the atoms sampled by Eirene in the fully kinetic simulation with respect to the ion temperature computed by Soledge2D, in the fluid and buffer zones. As it is evident, in both areas the mean error is close to 0\%, with a small spread around this value: by using the limiting values for Knudsen number used up to now ($K_{lim} = 0.05$ and $K_{buf} = 0.1$), the mean value for the fluid and for the buffer areas is respectively $-0.34\%$ and $-0.24\%$, with a standard deviation of $\sim 1.73$ and $\sim 1.16$.

$\sim 83\%$ of the mesh points identified in fluid areas have an error between $-10\%$ and $10\%$. 

while the percentage increases to $\sim 92\%$ in buffer zones; for a smaller error, for example between $-5\%$ and $5\%$, the fraction of fluid and buffer points slightly decrease to $\sim 74\%$ and $\sim 88\%$ respectively.

As a first conclusion, we can see that the assumption $T_{n,f} = T_i$ is correct within a $5 \div 10\%$ range for the majority of the grid elements in the fluid regions, at the very least in the simulation discussed here; we could thus imagine using the relative error on the temperature as an alternative parameter to distinguish fluid and kinetic areas. Nonetheless, small differences are still present, and whether they have any effect will be investigated in a short while.

Secondly, we see that in the buffer region a high number of mesh elements still have a negligible difference between the two temperatures: as such, we can increase the limiting values for the Knudsen number for the fluid and buffer zones. This is even more evident by looking at figure 4.10 that shows the spatial distribution of the error on the temperature in the ITER simulation shown up to now. It has to be noted that, from figure 4.9 there seems to be a number of fluid and buffer elements where the temperature difference between atoms and ions gets values greater than $\pm 10\%$, but by inspecting figure 4.10 these elements result to be on the interface between the fluid/buffer zone and the surrounding kinetic area, and are likely related to high energy neutrals generated by charge exchange.

Figure 4.9: Distribution of the relative error between the atoms temperature computed by Eirene and the ions temperature computed by Soledge2D in the grid elements identified as fluid, in the top plot, and as buffer zones, in the bottom plot. The red solid lines represent a normal distribution with parameters indicated in the figure, normalized to the maximum of the distribution.
with hot ions near the core region. This means that their "condensation mean free path" is still substantial and they are consequently able to penetrate in the fluid regions; for this reason, a possible solution might be to increase the multiplicative factor $m$ in the definition of the reaction rates for evaporation and condensation.

![Figure 4.10: Absolute value of the relative error between the atoms temperature computed by Eirene and the ions temperature computed by Soledge2D on the left, and fluid and kinetic zones as distinguished by the algorithm with limiting values of $K_{lim} = 0.05$ and $K_{buf} = 0.1$ on the right. The white line on the plot on the left helps showing the fluid+buffer area.](image)

While the difference between the atoms temperature and the ions temperature is small, it is not identically zero even in fluid zones; this might thus introduce errors in the fluid contribution to the ion energy source. If we look at how this source term is calculated

$$S_{E,i} = n_e n_i E_n \langle \sigma v \rangle_{iz} - n_e n_i E_i \langle \sigma v \rangle_{rc} + (n_i n_n E_n - n_n n_i E_i) \langle \sigma v \rangle_{cx}$$ (4.31)

we see that the atom energy comes up in the ionization and charge exchange contribution. Using the definition of energy $E_{\text{species}} = \frac{1}{2} m_{\text{species}} u_{\text{species}}^2 + \frac{3}{2} k_B T_{\text{species}}$, we can rewrite the ion energy source term as the sum of a kinetic $S_{E,i,\text{kin}}$ and an internal energy $S_{E,i,\text{int}}$ contribution.
with the usual assumptions \( n_i = n_e \) and \( m_i = m_n \). While no temperature related differences arise in the kinetic contribution, the assumption \( T_n = T_i \) means that the internal energy contribution due to the fluid phase computed in the code is

\[
S_{E,i,int}^* = \frac{3}{2} k_B n_e (T_n n_e \langle \sigma v \rangle_{iz} - T_i n_e \langle \sigma v \rangle_{rc} + \langle \mathcal{F} - \mathcal{F}_i \rangle \langle \sigma v \rangle_{cx}) = \frac{3}{2} k_B T_i n_e (n_n \langle \sigma v \rangle_{iz} - n_e \langle \sigma v \rangle_{rc})
\] (4.34)

so that an error \( \Delta S_{E,i} \) arises whenever \( T_F \neq T_i \)

\[
\frac{\Delta S_{E,i}}{S_{E,i,int}^*} = \frac{S_{E,i,int} - S_{E,i,int}^*}{S_{E,i,int}^*} = \frac{3}{2} k_B n_e (T_n - T_i) (\langle \sigma v \rangle_{iz} + \langle \sigma v \rangle_{cx}) = \frac{T_n - T_i}{T_i} \langle \sigma v \rangle_{iz} - \frac{n_n}{n_n} \langle \sigma v \rangle_{rc} \frac{1 + \frac{\langle \sigma v \rangle_{cx}}{\langle \sigma v \rangle_{iz}}}{1 - \frac{n_e}{n_n} \frac{\langle \sigma v \rangle_{cx}}{\langle \sigma v \rangle_{iz}}}
\] (4.35)

This equation, in turn, lets us estimate the allowable magnitude of the relative temperature difference \( \frac{T_n - T_i}{T_i} \) in order to obtain an error on the ion energy source smaller than a target value \( \varepsilon \), in function of the local values of plasma density, temperature and neutral density.

In particular, the inequality \( \left| \frac{\Delta S_{E,i}}{S_{E,i,int}^*} \right| \leq \varepsilon \) can be rewritten as

\[
\left| \frac{T_n - T_i}{T_i} \right| \leq \varepsilon \frac{1 - \frac{n_n}{n_n} \langle \sigma v \rangle_{rc}}{1 + \frac{\langle \sigma v \rangle_{cx}}{\langle \sigma v \rangle_{iz}}}
\] (4.36)

Unfortunately, this does not give us information on the behavior of the error with respect to easier to understand parameters like temperatures and the densities, as the reaction rates are non linear functions of them and an analytic equation can not be constructed. Nonetheless, by using the AMJUEL database we can estimate the reaction rates for ionization, recombination and charge-exchange, and consequently find the allowable relative temperature deviations \( \frac{T_n - T_i}{T_i} \) to limit the ion energy source error to \( \varepsilon \) in function of three parameters: the plasma density \( n_e \), the electron temperature \( T_e \) and the ratio
Figure 4.11: Maximum relative temperature difference \( \frac{T_n - T_i}{T_i} \) allowable to limit the error on the ion energy source term to a value \( \varepsilon \) of 0.01\% (top left), 0.1\% (top right), 1\% (bottom left) and 10\% (bottom right) in function of the plasma density and electron temperature, and for a density ratio \( \frac{n_e}{n_i} = 1 \).

In reality, the charge exchange reaction rate depends on the ion temperature, but in the following we will assume that \( T_e = T_i \) to avoid adding an additional independent variable.

The inequality 4.36 is shown in figure 4.11, in function of the electron temperature and electron density, for a density ratio \( \frac{n_e}{n_i} = 1 \) and for four values of threshold error on the ion energy source (\([0.01, 0.1, 1, 10]\)%).

We do not show the dependence with the density ratio, as the error on the ion source seems to be quite insensitive with respect to \( \frac{n_e}{n_i} \) in the range explored \( 0.01 \div 100 \), due to the fact that this ratio only enters the equation with the recombination contribution and recombination is always negligible with respect to the other processes (except at very low temperatures, of course).

The first thing we can note is that, as one would expect, the maximum allowable temperature differences increase as we soften the constraints on the maximum error on the ion energy source: for small deviations on the ion energy source term, in most of the temperature and density range a temperature difference smaller than 0.01\% would be required; by allowing errors of 1\%, neutral temperatures \( \sim 1\% \) higher or smaller than the temperature of the ions can be tolerated in a large range.

The plasma density does not seem to play a big role, due to the fact that the reaction rates only show slow dependencies on the plasma density. The remaining actor is then
the electron temperature, which on the other hand is responsible of big variation in the reaction rates. In fact the maximum relative temperature variations can be kept at a relatively high level for high temperatures, but the maximum value allowable quickly decreases when the temperature gets lower than \( \sim 5 \text{eV} \), before increasing again when temperature reaches \( \sim 0.5 \text{eV} \). This is due to the fact that the numerator of equation 4.36, \( 1 - \frac{n_e}{n_n} \frac{\langle \sigma v \rangle_{iz}}{\langle \sigma v \rangle_{iz}} \), goes to zero, as the recombination rate (scaled by \( \frac{n_e}{n_n} \)) exactly balances the ionization process due to the low temperature.

These estimation, on the other hand, should be taken with a grain of salt: first of all, the assumption of \( T_e = T_i \) may not be true in reality and in the simulations, so the relative contribution of charge exchange might be different than what presented here; furthermore, the error presented here only affects the fluid part of the ion energy source term, so that in cases where the fluid contribution is small with respect to the kinetic counter-part we can still expect small deviations of the hybrid source; finally, these conclusions may not be relevant in cases where the internal energy contribution to the source term is negligible with respect to kinetic part.

If we were to consider the relative error to full ion energy source term, we would obtain

\[
\frac{\Delta S_{E,i}}{S_{E,i}} = \frac{S_{E,i} - S_{E,i}^*}{S_{E,i}} = \frac{T_n - T_i}{T_i} \left( (\langle \sigma v \rangle_{iz} + \langle \sigma v \rangle_{ex}) \right) = \frac{T_n - T_i}{T_i} \frac{1}{(1 + \mu_n) - (1 + \mu_i)} \frac{\langle \sigma v \rangle_{ex}}{\langle \sigma v \rangle_{iz}} + (\mu_n - \mu_i) \langle \sigma v \rangle_{ex} \langle \sigma v \rangle_{iz}^{-1} \langle \sigma v \rangle_{ex}^{-1} \langle \sigma v \rangle_{iz}^{-1} (4.37)
\]

where we have defined \( \mu_\alpha = \frac{1}{2} \frac{m_{\alpha} u_{\alpha}^2}{k_B T_{\alpha}} = \frac{1}{2} \frac{m_{\alpha} u_{\alpha}^2}{k_B T_{\alpha}} \) the ratio between the kinetic and internal energy of particle \( \alpha \) (ions or atoms), and we have used the approximation \( T_n = T_i \) for \( \mu_n \).

As before, this can be translated in a condition for the maximum relative difference in temperature between atoms and ions to limit the relative error on the (now total) ion energy source

\[
\left| \frac{T_n - T_i}{T_i} \right| \leq \varepsilon \frac{(1 + \mu_n) - (1 + \mu_i) \frac{n_e}{n_n} \langle \sigma v \rangle_{ex} + (\mu_n - \mu_i) \frac{\langle \sigma v \rangle_{ex}}{\langle \sigma v \rangle_{iz}}}{1 + \frac{\langle \sigma v \rangle_{ex}}{\langle \sigma v \rangle_{iz}}} \]

(4.38)

As we can see, for any value of \( \mu_n \) the ionization contribution increases allowing for a
higher relative difference in temperature. Similarly for any value of $\mu_i$, the recombination contribution increases but its effect is opposite, reducing the maximum relative difference in temperature; note nonetheless that the recombination contribution is often negligible, except for low temperatures.

Finally, a contribution of charge exchange now appears (the contribution of charge exchange to the internal energy for the fluid population is identically null as $T_n - T_i = 0$) and it is proportional to $\mu_n - \mu_i = \frac{1}{2} m (u_n^2 - u_i^2)$. In particular, when $|u_n| \geq |u_i|$ higher differences in the temperature of the atoms can be allowed for a given target error $\varepsilon$ on the ion energy source.

4.6 ITER simulations: hybrid vs fully-kinetic

4.6.1 Simulations setup

In order to assess both the precision and the computational performance of the hybrid code with respect to the kinetic Monte Carlo code alone, a series of Soledge2D-hybrid simulations in ITER geometry have been performed varying the breakdown value of the fluid areas $K_{lim}$ while keeping all other inputs the same, and comparing these results to a simulation performed with the full kinetic model Soledge2D-Eirene.

The parameters used in the simulations, in term of number of $K_{lim}$ and $K_{buf}$ (as a reminder, the value of Knudsen number which interfaces the buffer and kinetic regions) are shown in table 4.1. As a reminder, fluid breakdown Knudsen number of up to 0.1 have been shown to result in negligible errors in hybrid codes used in fluid-dynamics; in this study we decided to venture to values up to 5 as limiting cases, even though we would expect large differences to arise in these simulations.

---

$^1$Note that the same number of particle histories per core (50000) has been used in all cases: while the simulations were performed on the same machine with the same number of cores (13) resulting in the same total number of simulated particles, the hybrid simulations require two additional strata in Eirene (one for the volumetric source due to evaporation, and one for the surface source due to fluid atoms reflecting on the walls as kinetic atoms). This has thus an impact on the partitioning of the particle histories in the various strata: usually, one core is assigned to each stratum and then all remaining cores are assigned to the ion recycling stratum. In the fully kinetic simulation, a total of 4 strata were identified (1 for ion recycling, 1 for volumetric recombination, and 2 for puffs) giving a cores splitting of 10-1-1-1; in the hybrid simulations the two additional strata cause a 8-1-1-1-1 splitting, effectively resulting in 20% less particles used in the sampling of sources due to neutrals born of recycling of ions.
4.6. ITER simulations: hybrid vs fully-kinetic

<table>
<thead>
<tr>
<th></th>
<th>$K_{\text{lim}}$</th>
<th>$K_{\text{buf}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinetic</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Hyb-1</td>
<td>0.01</td>
<td>0.1</td>
</tr>
<tr>
<td>Hyb-2</td>
<td>0.05</td>
<td>0.1</td>
</tr>
<tr>
<td>Hyb-3</td>
<td>0.1</td>
<td>0.5</td>
</tr>
<tr>
<td>Hyb-4</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>Hyb-5</td>
<td>1.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Hyb-6</td>
<td>5.0</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 4.1: Input parameters used for the kinetic and hybrid simulations.

The remaining input parameters were kept the same: pure Hydrogen plasmas, 20MW input power at the core interface, "auto-consistent" density boundary condition (at the core interface, a plasma particle flux equivalent to the total influx of neutral particles from the previous iteration is set), fixed transport coefficients of $D = \nu = 0.3 m^2/s$ and $\chi_e = \chi_i = 1 m^2/s$ without ballooning.

Only a simplified model for the molecular reactions was used to speed up the simulations; all the wall elements were considered made of Beryllium, with a unitary recycling coefficient $R = 1$.

One pump was set below the divertor dome with a fixed albedo of 0.9928, while two puffs were introduced in the top of the machine, each one puffing $H_2$ molecules at a rate of $5.65 \cdot 10^{21} D/s$ with a temperature of $0.03 eV$. The position of the pump, the puffs and the separatrix are shown in figure 4.12.

For reference, the mesh generated to perform these simulations in ITER geometry was made of $\sim 27500$ triangles for the hybrid code, and of $\sim 24400$ quadrilateral elements for Solvedge2D (out of which about half is used for the penalization technique).

Finally, as explained before, the hybrid simulations also require additional inputs, namely $\alpha = 1$ (fraction of recombination in kinetic), $n_{\text{min}} = 5 \cdot 10^{17} m^{-3}$ and \left(\frac{\langle \sigma v \rangle_e}{\langle \sigma v \rangle_i}\right)_{\text{min}} = 5$. As also shown before, the two remaining pre-factors are present mostly to avoid divisions by zero, so the value used for their thresholds are not important in this description.

In figure 4.13 examples of the regions of the computational domain that have been identified as fluid and as buffer for some of the cases described. As one would expect, increasing the limiting value of the Knudsen number increases the number of fluid and

\footnote{Using a more complete model, for example with the set of reactions shown in reference [39], would increase the computational time required by Eirene for the molecules and consequently reduce the impact of the hybrid model, as the latter is currently implemented for atoms only.}
Figure 4.12: On the left, position of the boundaries for the ITER simulations; the red line shows the core interface, the black one the opening of the pumps and the green one the Beryllium plasma-facing components; the blue lines show the position of the two puffs. On the right, the average number of charge exchange collisions undergone by atoms before ionization in the simulations.
4.6. ITER simulations: hybrid vs fully-kinetic

Figure 4.13: Fluid (in blue), buffer (in green) and kinetic (in red) areas identified during the ITER simulations with $Kn_{lim}$ set to 0.05, 0.5 and 5 on the left, in the center and on the right respectively. Only the divertor is shown due to the fact that the rest of the domain is all kinetic.

buffer triangles, while reducing the number of kinetic ones; note nonetheless that still the greater part of the triangles is considered as kinetic, mostly due to the fact that the condition of charge exchange collisions dominating over ionization events

$$\frac{\langle \sigma v \rangle_{cx}}{\langle \sigma v \rangle_{iz}} \leq \left( \frac{\langle \sigma v \rangle_{cx}}{\langle \sigma v \rangle_{iz}} \right)_{\text{min}}$$

is not verified in those areas$^8$.

The results of the fully-kinetic simulation are shown in figures 4.14 and 4.15 in terms of density (for plasma, atoms and molecules), temperature (for electrons, ions, atoms and molecules), and ion Mach number.

In particular we can see from the figures that parallel gradients along the separatrix are able to drop the temperature in front of the targets to $\sim 20\text{eV}$ but not enough to detach the plasma, and the divertors stays in a high recycling regime.

In fact, this case does not exhibit a massive increase in the computational time of Eirene because of exceedingly long trajectories, as charge exchange was not predominant in the vast part of the domain as shown in figure 4.12, and a purely kinetic description might have been required: to give a perspective, condensation occurred in only $\sim 0.1\%$ of mesh elements for the lowest $Kn_{lim}$ case, and the percentage only increased to $\sim 2\%$ for the case with the highest $Kn_{lim}$.

It should be noted that converging full plasma-neutrals simulations with detached conditions for ITER (DEMO, etc...) poses a number of difficulties, especially for the particle

$^8$Note that some parts of the interfaces between fluid and kinetic, especially the ones facing the core boundary, do not show a buffer zone. This is due to the fact that the fluid zone is cut by the pre-factor that imposes a kinetic zone where the ratio between reaction rates of charge exchange and ionization is below the threshold value, which is implemented through a simple Heaviside step function; a refinement of the model by allowing a smoother transition by the pre-factors will be added in the future.
Figure 4.14: From left to right, plasma density ($[m^{-3}]$), electron temperature ([eV]), ion temperature ([eV]) and ion Mach number obtained during the fully-kinetic Soledge2D-Eirene simulation.

Figure 4.15: From left to right, atom density ($[m^{-3}]$, in logarithmic scale), atom temperature ([eV]), molecule density ($[m^{-3}]$, in logarithmic scale) and molecule temperature ([eV]) obtained during the fully-kinetic Soledge2D-Eirene simulation. Remember that for this simulation, all atoms are considered kinetic and there is no fluid population.
4.6. ITER simulations: hybrid vs fully-kinetic

Figure 4.16: Concentration of the kinetic phase, on the left, and of the fluid phase, in the center, for the hybrid simulation with $K_{\text{lim}} = 0.5$. The concentrations are defined here with respect to the total atom density ($c_K = \frac{n_K}{n_K + n_F}$ for example for the kinetic phase). For reference, on the right are shown the fluid, buffer and kinetic zones identified during the simulation.

balance, which go beyond the scope of this manuscript.

Nonetheless, if the hybrid model is able to reproduce the kinetic results with a good agreement and, in the meantime, having a speed-up in such a non-advantageous regime, it will be encouraging for the performances of the code in detached regimes.

4.6.2 Hybrid results

Let us now analyze the results of the hybrid simulations, obtained with the Soledge2D-hybrid code. Note that, in contrast with the simulations shown in chapter 3 which were performed in "standalone" fashion (with only the fluid code running, and the fluid neutrals evolving with respect to a fixed plasma background) these simulations are fully coupled with Soledge2D, and the results shown in the following have been obtained for the full code once it has reached convergence.

In these cases, we now have two different populations of atoms, namely the fluid and kinetic phases. The analysis performed in the previous sections (equations 4.14 and 4.15) tells us that in fluid zones we can expect most atoms to belong to the fluid phase, and conversely mostly kinetic atoms should be present in the kinetic areas of the domain. This may not be identically true, because of the approximation of no transport used to obtain this conclusion; furthermore, to prevent divisions by zero in the fluid code when passing from the fluid atom flux to the fluid atom velocity $u_f = \frac{\Gamma_f}{n_f}$, a lower limit $n_{F,\text{min}}$ (usually taken as $10^{-10}$ in dimensionless units) is set for the fluid atoms density, so that the concentration of fluid atoms will never be identically zero.

Nonetheless, as seen from figure 4.16, the system behaves as expected and the con-
Figure 4.17: Density of the kinetic phase (on the left) and of the fluid phase (in the center) in logarithmic scale for the hybrid simulation with $Kn_{lim} = 0.5$.

Concentration of the fluid atoms is negligible (ranging from 0.01% to $1 \cdot 10^{-9}$%) everywhere except in the triangles identified as fluid, with a strong discontinuity at the interfaces and in the contiguous buffer regions due to the shape of the factors $F_{cond}$ and $F_{evap}$ used to compute the condensation and evaporation sources.

The evaporation and condensation sources for particles are shown in figure 4.18. We can clearly see that for the most part the phase changes happen at the interfaces between fluid and kinetic zones. It is interesting to see how the profile of the condensation source in the fluid zones is completely hollow, showing that no kinetic atom is actually able to cross the buffer zone. While this might give some concern about the statistics in the fluid areas, the deterministic fluid population is largely dominating here (with a concentration ranging from 92% for the lowest $Kn_{lim}$ case to 99.95% for the highest) so that the statistical noise only applies to the negligible kinetic contribution.

We can find some relatively high concentrations of fluid atoms around the core interface: this is not due to high fluid density, but it can rather be easily explained by the fact that in these areas the kinetic density is low enough (as shown in figure 4.17 for the same simulation) to be comparable to the minimum value of the fluid density $n_{F,lim}$; furthermore, these areas do not usually receive a high enough number of histories in Eirene to obtain good enough statistics, so that we can expect high relative deviations caused by statistical noise.
4.6. ITER simulations: hybrid vs fully-kinetic

Figure 4.18: Fluid atoms source due to condensation, on the left, and kinetic atoms source due to evaporation, on the right for the hybrid simulation with $Kn_{lim} = 1$. The white line on the plot helps showing the fluid region.

4.6.3 Sources by plasma-neutrals interactions

While the condensation and evaporation sources are important for the hybrid algorithm, we should concentrate on the sources for the plasma solver $S_{p,n}$, $S_{p,G}$, $S_{p,E,e}$ and $S_{p,E,i}$. As explained, the hybrid code computes separately the contribution of the kinetic phase from Eirene and the contribution of the fluid phase from the fluid code. An example of the plasma particle source is shown in figure 4.19.

Due to the fact that only a relatively small number of mesh elements are treated as fluid and that the volumetric recombination and recycling processes are both fully treated kinetically in these simulations, the kinetic contribution dominates for the most part the sources; nonetheless, the contribution of the fluid population is still important all along the divertor legs, having the same weight in these areas as the kinetic contribution. Of course we can expect that in more pertinent divertor regimes, as the Knudsen number gets lower and the number of fluid mesh elements increases, the fluid contribution to the plasma sources will start to dominate.

More importantly, we can see that the sources computed through the hybrid simulations qualitatively agree with the sources computed by the fully kinetic simulations. This is shown in figures 4.20-4.23 for the plasma particle, plasma parallel momentum, electron energy and ion energy sources along the separatrix; in these figures, we can see that the red dots referring to the kinetic simulation closely match all the solid lines resulting from
Figure 4.19: Particle source for the plasma solver computed by the hybrid simulation with $Kn_{lim} = 1$. The contribution due to the kinetic population is shown on the left, the one due to the fluid phase is at the center, and the total source computed as sum of the two contributions is shown on the right.

the hybrid simulations, even if as the breakdown limit $Kn_{lim}$ increases the match gets worse.

The relative error on the peak value of the four sources between the various hybrid cases and the kinetic one after convergence is reached is reported in table 4.2. As we can see, we get a good agreement between the simulations even quantitatively, at least for low values of the breakdown parameter $Kn_{lim}$. This is in accord with the recent results in other research fields (namely fluid-dynamics) [53], in which values of the Knudsen number up to 0.1 were found to be acceptable as fluid breakdown limits. At least in the particular ITER case just showed, we also find that using a $Kn_{lim} = 0.1$ still produces small variations in the peak value of the sources, and it could be argued that values up to $Kn_{lim} = 0.5$ could still be acceptable. Of course when we go to higher values, the solution diverges more and more with respect to the fully-kinetic case, as one would expect given the fact that a fluid approximation breaks down for Knudsen numbers close to unity.

<table>
<thead>
<tr>
<th>$Kn_{lim}$</th>
<th>$\Delta S_{n,r}^p$ [%]</th>
<th>$\Delta S_{n,i}^p$ [%]</th>
<th>$\Delta S_{E,e}^p$ [%]</th>
<th>$\Delta S_{E,i}^p$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyb-1</td>
<td>0.01</td>
<td>2.9</td>
<td>9.6</td>
<td>3.8</td>
</tr>
<tr>
<td>Hyb-2</td>
<td>0.05</td>
<td>4.1</td>
<td>8.3</td>
<td>4.8</td>
</tr>
<tr>
<td>Hyb-3</td>
<td>0.1</td>
<td>4.3</td>
<td>6.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Hyb-4</td>
<td>0.5</td>
<td>10.8</td>
<td>12.4</td>
<td>12.7</td>
</tr>
<tr>
<td>Hyb-5</td>
<td>1.0</td>
<td>17.8</td>
<td>25.5</td>
<td>20.1</td>
</tr>
<tr>
<td>Hyb-6</td>
<td>5.0</td>
<td>28.3</td>
<td>40.2</td>
<td>31.6</td>
</tr>
</tbody>
</table>

Table 4.2: Relative error on the peak value along the separatrix of the sources for the plasma in the hybrid simulations with respect to the full-kinetic case $\Delta S = \frac{|S_{kin} - S_{hyb}|}{|S_{kin}|}$. 
4.6. ITER simulations: hybrid vs fully-kinetic

Figure 4.20: Plasma particle source profile along the separatrix, for the hybrid simulations (solid lines) compared to the fully kinetic case (red dots). Due to the sources being close to zero along most of the poloidal connection length, only a zoom close to the low field side (on the left) and high field side (on the right) targets are shown.

Figure 4.21: Parallel momentum source profile along the separatrix, for the hybrid simulations (solid lines) compared to the fully kinetic case (red dots). Due to the sources being close to zero along most of the poloidal connection length, only a zoom close to the low field side (on the left) and high field side (on the right) targets are shown.
Figure 4.22: Electron energy source profile along the separatrix, for the hybrid simulations (solid lines) compared to the fully kinetic case (red dots). Due to the sources being close to zero along most of the poloidal connection length, only a zoom close to the low field side (on the left) and high field side (on the right) targets are shown.

Figure 4.23: Ion energy source profile along the separatrix, for the hybrid simulations (solid lines) compared to the fully kinetic case (red dots). Due to the sources being close to zero along most of the poloidal connection length, only a zoom close to the low field side (on the left) and high field side (on the right) targets are shown.
4.6. ITER simulations: hybrid vs fully-kinetic

Something interesting to note is the fact that small variations are found also for the ion energy source, of the same order as for the other sources. As we cautioned in the previous section, the lack of an energy equation for the fluid part of the atom population introduces additional errors in the solution and namely in the ion energy equation 4.35. Nonetheless, at least for the case presented here, the fears might have been exaggerated. This apparent discrepancy between the "analytic" model presented in the previous section and the results shown here is in fact explained by a combination of factors:

- first of all, in kinetic areas the fluid contribution to the sources is small and as such negligible errors are introduced in these zones;
- in fluid zones instead, the parameter $\varepsilon_T$ is close to unity, opening up a window of operation with small errors introduced;
- furthermore by setting the parameter $\alpha = 1$ we are turning off the recombination contribution in equation 4.35;
- finally, the assumption $T_e = T_i$ used to derive the equation 4.35 is not verified in this case.

To clarify, the last two items in the list above effectively shift the position of the discontinuity presented in section 4.5. Furthermore, as shown in figure 4.9 for a simulation with $Kn_{lim} = 0.05$, most of the fluid and buffer cells have a ratio $\varepsilon_T \simeq 0.98$, resulting in errors of the order of $\sim 1 \div 10\%$ as shown in figure 4.11, in line with the errors found in table 4.2. By increasing the value of $Kn_{lim}$ we are effectively broadening the $\varepsilon_T$ range, thus increasing the error.

However, we caution that this might be just a fortuitous coincidence and more detailed studies are necessary to assess the impact of the addition of an equation for the temperature of the fluid phase (would that reduce errors even further?) and actually prove the if such a complication is necessary or not.

4.6.4 Particles and energy fluxes on the wall

Still, what actually interests us, as pertaining the exhaust of particles and heat, are the plasma profiles along the wall. In figures 4.24-4.28 are thus shown the profiles of plasma density, electron temperature, ion temperature, total particle flux and total heat flux along the wall, focusing on the divertor targets. As we can see (or rather not see!), the curves relative to the fully-kinetic simulation and to the various hybrid simulations are in almost perfect agreement, even for high $Kn_{lim}$ values. The peak value of the heat
Chapter 4. Hybrid fluid-kinetic description

flux, its exponential width $\lambda_q$ and spreading factor $S_q$, integral width $\lambda_{int,osp}$ and total heat load $Q_{osp}$ at the outer strike point are reported in table 4.3 for the fully-kinetic and hybrid simulations. The integral width is approximated here as $\lambda_{int} \simeq \lambda_q + 1.64S_q$ [23]; the total heat load is approximated as $Q_{osp} = 2\pi R_{osp}\lambda_{int,target}q_{osp}$, where $R_{osp}$ is the radial coordinate of the outer strike point and $\lambda_{int,target}$ is the value of the integral width computed directly at the target.

As we can see, the maximum error on the integral width of the heat flux is 6.4% resulting in a maximum error on the total heat load of the outer strike point of 6.3% except for the $Kn_{lim} = 5$ case (with a still relatively small error of 12.7%).

<table>
<thead>
<tr>
<th>$Kn_{lim}$</th>
<th>$q_{peak,osp}$ [MW/m²]</th>
<th>$\lambda_{q,osp}$ [mm]</th>
<th>$S_{q,osp}$ [mm]</th>
<th>$\lambda_{q,osp}/S_{q,osp}$</th>
<th>$\lambda_{int}$ [mm]</th>
<th>$Q_{osp}$ [MW]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eirene</td>
<td>6.9</td>
<td>33.9(4.5)</td>
<td>8.4(1.1)</td>
<td>4.1</td>
<td>47.6(6.3)</td>
<td>11.5</td>
</tr>
<tr>
<td>Hyb-1</td>
<td>0.01</td>
<td>6.8</td>
<td>34.5(4.6)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>49.3(6.6)</td>
</tr>
<tr>
<td>Hyb-2</td>
<td>0.05</td>
<td>6.8</td>
<td>35.7(4.8)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>50.6(6.7)</td>
</tr>
<tr>
<td>Hyb-3</td>
<td>0.05</td>
<td>6.8</td>
<td>34.5(4.6)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>49.3(6.6)</td>
</tr>
<tr>
<td>Hyb-4</td>
<td>0.5</td>
<td>6.9</td>
<td>35.7(4.8)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>50.6(6.7)</td>
</tr>
<tr>
<td>Hyb-5</td>
<td>1.0</td>
<td>7.1</td>
<td>34.5(4.6)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>49.3(6.6)</td>
</tr>
<tr>
<td>Hyb-6</td>
<td>5.0</td>
<td>7.3</td>
<td>35.7(4.8)</td>
<td>9.1(1.2)</td>
<td>3.8</td>
<td>50.6(6.7)</td>
</tr>
</tbody>
</table>

Table 4.3: Outer strike point heat flux peak value, width $\lambda_q$, spreading factor $S_q$, ratio between width and spreading factor, integral width and total power deposited. The lengths reported here are calculated at the target, but the values when remapped at the outer mid plane are also shown in parenthesis.

4.6.5 Computational performance

Finally, let us address the computational performance of the hybrid model. As explained, the whole idea of the model is to reduce the length of Eirene histories in low Knudsen number areas by introducing the condensation process, with a reaction rate greater than charge exchange which slows down the Monte Carlo process. In figure 4.29 the average time per process spent in Eirene during the hybrid simulations is shown; as the same machine with the same number of cores has been used for the simulations, this directly correlates to the total time spent in Eirene. As evident, increasing the breakdown value of the Knudsen number $Kn_{lim}$ used in the simulations clearly reduces the time spent as some of the long histories are ended because of the condensation; furthermore, the whole curve is below the time spent in Eirene for the fully-kinetic cases, shown in the plot by
4.6. ITER simulations: hybrid vs fully-kinetic

Figure 4.24: Plasma density profiles along the wall coordinate $s$ for the hybrid simulations (solid lines) compared to the fully kinetic simulation (red dots). Zooms on the inner strike point (top right) and on the outer strike point (bottom right) are also shown.

Figure 4.25: Electron temperature profiles along the wall coordinate $s$ for the hybrid simulations (solid lines) compared to the fully kinetic simulation (red dots). Zooms on the inner strike point (top right) and on the outer strike point (bottom right) are also shown.
Figure 4.26: Ion temperature profiles along the wall coordinate $s$ for the hybrid simulations (solid lines) compared to the fully kinetic simulation (red dots). Zooms on the inner strike point (top right) and on the outer strike point (bottom right) are also shown.

Figure 4.27: Particle flux profiles along the wall coordinate $s$ for the hybrid simulations (solid lines) compared to the fully kinetic simulation (red dots). Zooms on the inner strike point (top right) and on the outer strike point (bottom right) are also shown.
Figure 4.28: Heat flux profiles along the wall coordinate $s$ for the hybrid simulations (solid lines) compared to the fully kinetic simulation (red dots). Zooms on the inner strike point (top right) and on the outer strike point (bottom right) are also shown.

The blue dashed horizontal line. This means that the hybrid architecture results in a positive time gain in Eirene, shown in green in the figure, for the whole range of $Kn_{lim}$ explored here.

The fact that the time spent in Eirene depends on the breakdown parameter $Kn_{lim}$ is due to the fact that, increasing the latter we are increasing the number of mesh elements identified as fluid during the simulation as shown in figure 4.30, effectively increasing the volume where a kinetic atom has a high probability of changing phase. Note that the time spent in Eirene for the hybrid simulations is not exactly monotonically decreasing: the $Kn_{lim} = 0.01$ case spent less time on average than the $0.05$ case, even if the number of fluid mesh elements for the latter is clearly higher. This can be explained by the fact that, on the other hand, the number of buffer elements for the first case was higher than the second, as the Knudsen number in these elements was close to the breakdown limit, and the kinetic atoms entering in these regions were subject to a non-negligible condensation source even if the factor $F_{cond}$ was not identically unitary. Furthermore, the stochasticity of the Monte Carlo procedure can play a role in shortening or lengthening some of the histories, thus slightly modifying the average time between various simulations. Nonetheless we can expect that further decreasing the breakdown parameters $Kn_{lim}$ and $Kn_{buf}$, effectively making sure that no fluid or buffer zones are identified in the computational domain, the time spent on average in Eirene will tend to the same value for a fully-kinetic case; on
Figure 4.29: Average time spent in Eirene per process (in blue) and time gain in Eirene due to the hybrid model (in green) as a function of the chosen input breakdown parameter $Kn_{lim}$.

On the opposite side of the spectrum, for $Kn_{lim} \to \infty$ we can expect that the average time spent in Eirene will saturate to zero, or rather the time needed for Eirene to experience a single condensation event as soon as the particle is born.

It has to be reminded that these simulations did not show extremely long trajectories for the atoms. As such, we can expect the percentage of fluid elements in the computational grid to increase in more pertinent cases for a constant value of $Kn_{lim}$, thus increasing the speed-up in Eirene. Furthermore, ions-molecules elastic collisions might also produce long trajectories for the $D_2$ particles and an extension of the hybrid model to molecules should be envisioned.

While we analyzed the computational performance of Eirene during the hybrid simulations, it has to be noted that the breakdown parameter will not have any effect on the plasma solver Soledge2D nor the fluid code, as their performances are only linked to the numerical scheme, the total number of mesh elements (whether they are considered fluid, kinetic or buffer it has no impact) and their size.

It has to be said that at this time the fluid code has been parallelized with openMP\[45\] only for the inversion of the global matrix (its slowest task) with PaStiX, and the intrinsic parallelism given by the HDG scheme (the fact that the solution of the problem is local to the single mesh elements, as explained in chapter 3) has not been used; as such, some work on the full parallelization of the fluid code, possibly with an MPI architecture, and its optimization will further reduce the wall clock time linked to the fluid part of the
Figure 4.30: Number of grid elements identified as fluid during the hybrid simulations, as function of the chosen breakdown input parameter $Kn_{lim}$.

hybrid simulations.
Conclusions & perspectives

Conclusions

Power exhaust is one of the challenging issues that next generation of magnetized fusion devices (ITER, DEMO) will face and the interactions between neutral and charged particles are of the utmost importance in order to try to reduce the heat loads on the plasma facing components below the engineering limits.

As the interplay of transport of plasma and neutrals with the geometry plays a major role in the power and particles exhaust, two-dimensional coupled plasma-neutrals codes like Soldege2D-Eirene are the current workhorses used to simulate the scrape-off layer of both existing and future Tokamaks.

The plasma is usually described within the fluid approach, while neutral particles often require a kinetic treatment in most of the simulation domain owing to their large mean free paths with respect to the dimensions of the machines. Given the complexity of the geometry and the fact that many neutral species have to be retained (atoms, molecules), this kinetic problem is usually addressed with a Monte Carlo approach. However, for next generation large devices in which the non-linear coupling between the plasma and the neutrals is strong, transport codes can become CPU intensive. The situation is made worse by the fact that some regions of the domain can be considered as fluid for neutrals due to the large number of ion-neutrals collisions, and the Monte Carlo procedure employed for the neutrals requires the computation of long histories.

Improving the coupling between fluid plasma and kinetic neutrals codes has thus been identified as a critical bottleneck in plasma edge/SOL simulations for next step devices. This work attempts at solving the issue related to highly collisional regions by exploring two possible and parallel code operation strategies aimed at accelerating wall clock convergence time.

The first one relies on understanding the behavior of the plasma-neutral system in presence of statistical noise generated by the Monte Carlo procedure of Eirene. This work provides a theoretical framework in which the role of the noise can be analyzed. Even
though we are looking for a stationary solution, the numerical code, which relies on a
time marching scheme, converges towards a Statistical Stationary State (SSS) in which
fluctuations are driven by noise. The time average of the fields in the SSS is solution of
a well defined set of equations and can thus be defined as the stationary solution of the
problem in the presence of noise. The equations obeyed by the mean solution are similar
to the original equations of the problem, but feature additional terms defined as averages
of fluctuating fields. These spurious terms describe e.g. transport related to fluctuations,
showing a clear analogy between turbulent and noise induced transport. The magnitude
of these terms can be estimated from the SSS, as well as the variance of the solution
(which measures the typical deviation between the mean solution and the solution at a
given time step).

If the statistical noise shows small correlations in time, the averaging procedure allows us
to reconstruct the mean solution even at surprisingly high noise level, that is for Monte
Carlo simulations with low numbers of histories followed. In particular, we showed that
(at least in the case presented here) a reduction of the number of histories by a factor
$100 \div 1000$ with respect to the usual practice (guided by an empirical compromise be-
tween computational time and the amplitude of the fluctuations) is possible, obtaining
(mean) solutions that show only small differences with respect to simulations with better
statistics, at the cost of course of having to tally the time average during the end of the
simulations. On the other hand, if strong time correlations are present in the statisti-
cal noise realizations (characteristic of the short cycling procedure often applied to the
plasma-neutrals codes), the plasma is not able to filter out the statistical fluctuations
and is given time to converge to each different realization of the noise, and substantial
biases are introduced in the mean solution.

The most important result is that for a given case and a given noise level, the correlation
time of the statistical noise, which is determined by the way the Monte Carlo is operated,
is the main actor in determining whether the statistical noise effects on the solution are
substantial. In order to establish quantitative criteria to measure the noise-induced bi-
ases, a more detailed study of all the spurious terms introduced in the equations and, at
the same time, of the response of the numerical scheme to the large localized gradients
generated by the statistical noise should be performed. However, we can already conclude
that calling the Monte Carlo code with a low number of histories (but often enough, and
without statistical correlations between successive runs) can be used to accelerate the
convergence of the simulations towards the SSS, since the solution during the relaxation
phase is not of physical interest. The actual gain is likely to be case dependent.

While these conclusions are promising, the computational time of the Monte Carlo
runs depends also on the computational cost per history, and in some regions the neutrals can undergo a large number of collisions with the surrounding ions during their lifetime (generally in cold and dense plasma areas), before getting ionized or escaping the highly collisional region. Furthermore, for bigger fusion devices the highly collisional region is likely to be larger and hence more efficient at trapping neutrals, possibly resulting in long simulations even when using a limited number of histories.

For this reason, a hybrid kinetic-fluid model for the neutral particles has been developed in order to synergize the precision of the kinetic description with the lighter CPU cost of a fluid model. This hybrid model has also to be seen as an intermediate element in a hierarchy of neutral particle transport tolls, with the kinetic Monte Carlo approach being the high-fidelity simulation. The low-fidelity model, in the current state of the hierarchy, is a fluid model in which the atoms are assumed to have a pure Maxwellian velocity distribution with the same temperature of the ions, making the code suitable for charge exchange dominated regimes in local thermodynamical equilibrium with the background plasma. The model is implemented in a fluid code developed specifically for this work. The need for geometrical flexibility has led us to rely on a Hybridizable Discontinuous Galerkin method, which allows the use of unstructured meshes and of high order discretizations.

The hybrid model is based on a two phases approach in which the atoms are divided in two distinct populations, one fully fluid and one fully kinetic, which coexist in the whole simulation volume. Two additional phase change reactions have been introduced to let kinetic atoms condensate into the fluid population and fluid atoms to evaporate in the kinetic one. The introduction of these two reactions, and in particular of the condensation, has the purpose of creating a new mechanism to prematurely end the histories of the kinetic particles entering highly collisional regions in the Monte Carlo solver, thus reducing the computational time; the truncated histories are then translated into a source in the fluid code. In order to make sure that the condensation predominates over evaporation in highly collisional regions, the reaction rates for the two processes have been tailored in function of the local collisionality of the atoms, depending on the rate of the atom-ion collisions and on the length of the gradients of the plasma (that is, on the Knudsen number). Finally, to make sure that the long histories are truncated fast enough in the high collisional regions, the reaction rate of the condensation process in these areas is set as a multiple of the reaction rate of the atom-ion collisions. This hybrid model, initially proposed by Karney and Stotler, can be seen as a variant of models using different equations on different parts of the grid, where the dynamical (non pre-defined) spatial boundaries are enforced with a penalization scheme similar to the
one used in Soledge2D. Operationally, this means coupling the plasma solver Soledge2D, the kinetic Monte Carlo Eirene and the fluid code for the atoms together and running them sequentially at each iteration. Simulations in realistic ITER geometry (but with simplified physics, in particular regarding the molecular processes) show that the full hybrid code is able to reproduce (both qualitatively and quantitatively) the results of a similar kinetic simulation for a surprisingly large range of “threshold collisionality”, even in mostly kinetic conditions. More importantly, the hybrid code is shown to modestly out-performs the kinetic Monte Carlo code, even though the case we investigated here did not present very strongly collisional regions.

The two strategies presented here (relying on a reduced number of particles and using a hybrid model) have thus shown their ability to accelerate convergence of plasma edge simulations, at least in the cases investigated so far. The numerical and/or model uncertainties thus introduced can be checked using the high fidelity model with a large number of Monte Carlo histories. The robustness of these strategies will need further investigation, and will have to analyzed on a case by case basis.

**Future work**

The work described in this manuscript opens up a number of perspectives for future work. The investigation on the effects of the noise should now aim at obtaining quantitative criteria allowing one to assess the bias on the mean solution in a given situation. This requires a deeper analysis of the role and magnitude of the noise induced terms in the equations for the mean solution.

Regarding the hybrid model, the extension of the hybrid description to molecules should take precedence. So far, we have investigated hybrid models for atoms to establish a proof of principle, and treated molecules kinetically but including only a restricted set of molecular processes. In fact, when using a more complete description of the latter (including elastic collisions between molecules and ions, ion conversion, vibrationally resolved molecules, ...), molecules can account for most of the computational time spent in the Monte Carlo code. In fact, elastic collisions between molecules and ions can be another cause for the long histories. Technically, this requires the addition of a new species in the fluid code. It should be noted that the Knudsen number of atoms and molecules might be different, so that fluid and kinetic regions are not necessarily identical for all species. Furthermore, atoms resulting from molecular dissociation introduce a coupling between neutral species, and the corresponding atomic sources have to be treated with care.
Another potentially important refinement of the model is to include an energy equation for the neutral fluid population, thus relaxing the $T_n = T_i$ assumption made throughout this work. This would be especially important when using the neutral fluid model only in the simulations, and could prove essential in some divertor conditions where ion energy sources resulting from charge exchange or elastic collisions play an important role. Conversely, it could also potentially be interesting to introduce a diffusion approximation closure, i.e. the possibility to close the fluid model at the level of the continuity equation. This would allow further investigations on the influence of the closure and possibly speed up simulations further if conditions where such an approximation is justified are found.

While the current hybrid code already obtains a noticeable speed up over the Monte Carlo solver, the real capabilities of the underlying HDG method have not yet been fully employed: the discontinuity of the discretized solution embedded in the numerical method gives us the freedom to apply hp adaptivity techniques, able to locally refine the mesh (h refinement) or the order of the discretization (p refinement) in specific mesh elements, in particular those where the fluid phase dominates. This could part of a strategy to reduce the cost of solving the model on a grid much wider than needed, by dynamically modifying the h and p parameters as the boundary between the neutral phases evolves. Furthermore, the discontinuity of the solution allows straightforward parallelization of the code, which remains to be done.

Finally, it should be noted that an HDG code for the plasma equations has been developed in the same group, and that the latter is currently being merged with the neutral fluid code presented here for the neutrals. This would lead to a new monolithic HDG code for both charged and neutral particles, as a geometrically flexible and fast simulation tool.
Bibliography


[45] The openMP API. www.openmp.org. (Cited in section 4.6.5.)


[47] Pastix. pastixforge.inria.fr. (Cited in section 3.3.1.)


[61] WEST users portal. *westusers.partenaires.cea.fr*. (Cited in section 1.9.)


