



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE



PRINCETON  
PLASMA PHYSICS  
LABORATORY

MASTER PROJECT  
SPRING 2015

---

## Development of a Synthetic Diagnostic for Beam Emission Spectroscopy

---

*Author:*

Loïc HAUSAMMANN

*Supervisor:*

Michael CHURCHILL  
Choong-Seock CHANG  
Stephan BRUNNER

June 19, 2015

# Contents

1	Introduction . . . . .	3
2	XGC1 Code . . . . .	4
2.1	Tokamak Coordinates . . . . .	4
2.2	Gyrokinetic Equations . . . . .	5
2.3	XGC1 Algorithm . . . . .	7
2.4	Construction of the Mesh . . . . .	7
2.5	Output Files . . . . .	9
2.6	Interpolation of the Data . . . . .	10
	The Field Line Equations . . . . .	11
	The Runge-Kutta Methods . . . . .	12
	Benchmarking of the Interpolation . . . . .	14
3	Synthetic Diagnostic for the Beam Emission Spectroscopy . . . . .	18
3.1	Neutral Beam . . . . .	18
	Beam Density . . . . .	18
	Photon Emission . . . . .	23
	Instantaneous Emission . . . . .	23
	Non-Instantaneous Emission . . . . .	24
	ADAS: . . . . .	26
3.2	Optics . . . . .	28
	Sampling Volume . . . . .	28
	Approximation of the Integrals . . . . .	29
	Full Numerical Integration . . . . .	29
	Computation of the Solid Angle . . . . .	31
	Lens and Fiber Cases . . . . .	32
	Mixed Case . . . . .	32
	Filter . . . . .	37
	Point Spread Function . . . . .	38
4	Results . . . . .	39
4.1	Comparison of the Intensity and Density Fluctuations . . . . .	39
4.2	Sensitivity Required for the BES Diagnostic . . . . .	39
4.3	Spatial Correlation . . . . .	43
4.4	Time Correlation . . . . .	45
4.5	Effect of the Different Elements . . . . .	46
5	Conclusion . . . . .	51
6	Annexes . . . . .	52
6.1	Starting Objectives . . . . .	52
6.2	Description of the Code . . . . .	53
	ADAS . . . . .	53
	Collisions . . . . .	53

	XGC1 Loader . . . . .	54
	Beam . . . . .	54
	BES . . . . .	54
	Execution of the Code . . . . .	54
	Toolbox . . . . .	56
6.3	Configuration File . . . . .	56
6.4	Acknowledgments . . . . .	61

# 1 Introduction

In tokamaks and stellarators, turbulent fluctuations have a significant impact on the confinement time and need to be well understood in order to achieve energy production from fusion reactors. In the theoretical part of plasma physics, some large codes are used to simulate these fluctuations but usually the data generated are not directly comparable to experimental observables, which is why synthetic diagnostics are developed.

A synthetic diagnostic consists in simulating the effect of a measuring device using data from a simulation (in this case XGC1). The aim of this kind of measurement is to check if the theoretical model in the simulation describes well the problem and to find the most important effect on the diagnostic.

The measuring device analyzed in this work is the Beam Emission Spectroscopy (BES)[5] that consists in analyzing the Balmer- $\alpha$  light emission<sup>1</sup> of the excited atoms in a neutral beam. These neutral atoms are input into the plasma with a Neutral Beam Injector (NBI), are excited by plasma electrons, then emit photons which are collected by an optical setup (see Figure 1). This diagnostic measures the long wavelength density turbulence in a tokamak, along the radial and poloidal directions ( $k_{\perp}\rho_i \leq 1$  for DIII-D where  $k_{\perp}$  is the wave number perpendicular to the magnetic field, and  $\rho_i$  is the ion gyro radius). The diagnostic can be applied to the edge and the core.

The aim of the synthetic diagnostic code developed for this report is to be easily adaptable for any kind of tokamak (specially DIII-D and NSTX) and to take into account the 3D effect. The usual way to do a BES synthetic diagnostic is by computing the Point Spread Function (PSF)[15][7] that is only applied on one poloidal plane at a time, therefore this code will show the impact of the full 3D computation.

The data presented in this thesis comes from a simulation of a H-mode on DIII-D with an experimental magnetic equilibrium and the other equilibrium data comes from different discharge. This simulation will be called “particle pinch” in this report.

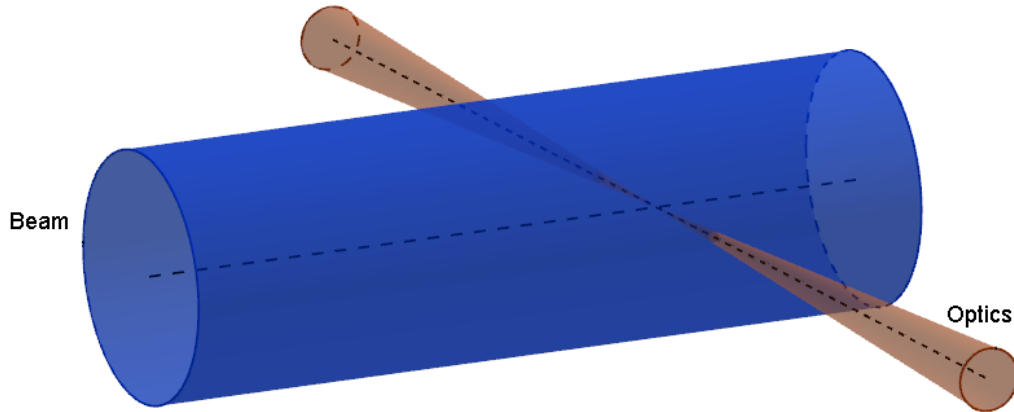


Figure 1: Representation of the BES system

<sup>1</sup>Named after the Swiss mathematician and mathematical physicist Johann J. Balmer born in 1825.

## 2 XGC1 Code

### 2.1 Tokamak Coordinates

In a tokamak, the natural coordinates (in the geometrical point of view) are the toroidal coordinates  $(r, \phi, \theta)$  given in figure 2 and in (1) where  $R_0$  is the major radius of the tokamak, and the cylindrical coordinates  $(R, \phi, Z)$  given in (2), it consists of a Cartesian mesh on each the poloidal planes.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (R_0 + r \cos(\theta)) \cos(\phi) \\ (R_0 + r \cos(\theta)) \sin(\phi) \\ r \sin(\theta) \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} R \cos(\phi) \\ R \sin(\phi) \\ Z \end{pmatrix} \quad (2)$$

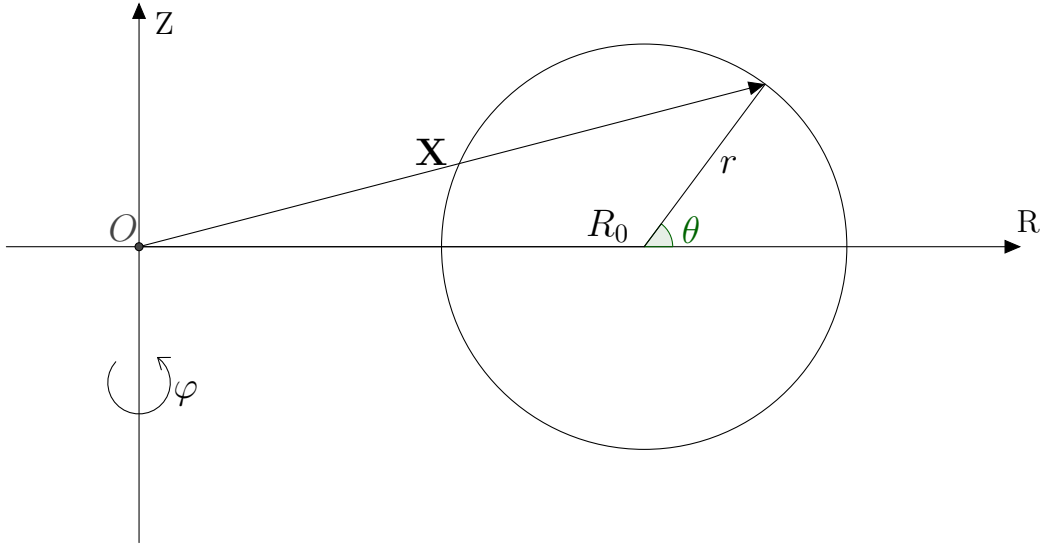


Figure 2: The three toroidal coordinates are shown  $(r, \phi, \theta)$

Theses coordinates are good for the tokamak but do not follow the field geometry. As the particles are affected by the fields and follow the magnetic field (in a constant and homogeneous magnetic field), it seems to be logical to choose some field-line following coordinates. With an electric field and a variable (in space and time) magnetic field, the trajectories are a lot more complicated but the plasma still tends to follow the magnetic field line. The only magnetic coordinate used in this work is the poloidal magnetic flux given in (3) where  $S$  is the disk delimited by the line of constant  $(R, Z)$ . As shown in (4), this coordinate defines magnetic surfaces due to the incompressibility of the magnetic field ( $\nabla \cdot \mathbf{B} = 0$ ). The natural normalization for a tokamak is  $\psi(\mathbf{R}_s)$  where  $\mathbf{R}_s$  is a point on the separatrix<sup>2</sup> and  $\psi_n = \frac{\psi - \psi_0}{\psi(\mathbf{R}_s) - \psi_0}$ , where  $\psi_0$  is the value for a point at the center of the plasma, will be used for indicating this normalization. The advantage of

<sup>2</sup>last closed magnetic surface

this normalization is that the closed magnetic surfaces are the area where  $\psi < 1$  and the opened ones when  $\psi > 1$  (see figure 3).

$$\psi = \frac{1}{2\pi} \int_S \mathbf{B} \cdot d\boldsymbol{\sigma} \quad (3)$$

$$\mathbf{B} \cdot \nabla \psi = 0 \quad (4)$$

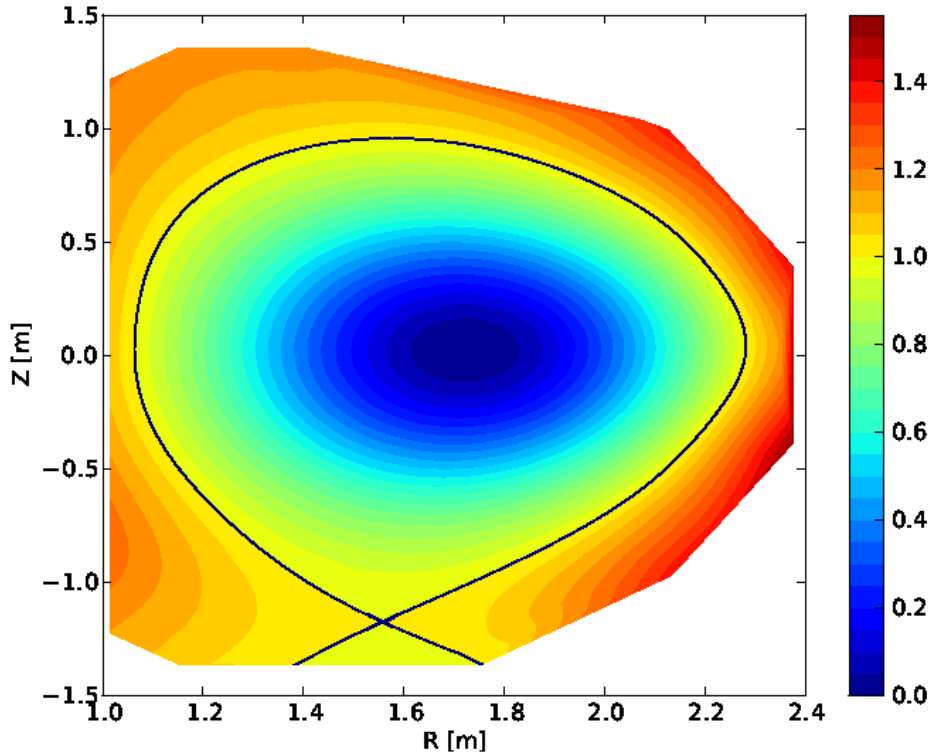


Figure 3:  $\psi_n$  is shown and the values are taken from the particle pinch simulation.

## 2.2 Gyrokinetic Equations

The particle approach (using the Newton's laws and the Lorentz force) is too heavy to compute due to the large amount of particles inside a plasma. The Fokker-Planck equation (6) can be derived with a statistical approach. This method consists in defining a distribution function ( $f$ ) that describes the charged particles (only one population is assumed here). By integrating over all the phase space (space and velocity), we obtain the number of particles at a specific time[17]:

$$\int f(\mathbf{x}, \mathbf{v}, t) d^3v d^3x = N_p \quad (5)$$

where  $\mathbf{x}$  is the position,  $\mathbf{v}$  the velocity and  $N_p$  the total number of particles in the space defined by the integral.

By taking the time derivative of (5), we obtain the Fokker-Planck equation that is usually used for describing a plasma[17]:

$$\left. \frac{df}{dt} \right|_{\text{collisionless traj.}} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \frac{\partial f}{\partial \mathbf{x}} + \frac{q}{m} (\mathbf{E} + \mathbf{v} \wedge \mathbf{B}) \cdot \frac{\partial f}{\partial \mathbf{v}} = C[f] + S[f] \quad (6)$$

where  $q$  is the charge of a particle,  $m$  its mass,  $\mathbf{E}$  the electric field and  $\mathbf{B}$  the magnetic field.  $C$  and  $S$  are the collision and source operator.

From the definition given above, we can define two quantities that will be used for Maxwell's equations<sup>3</sup>:

$$n(\mathbf{x}, t) = \int f(\mathbf{x}, \mathbf{v}, t) d^3v, \quad (7)$$

$$\mathbf{j}(\mathbf{x}, t) = q \int \mathbf{v} f(\mathbf{x}, \mathbf{v}, t) d^3v. \quad (8)$$

where  $n$  is the density of particles (useful for computing the charge density) and  $\mathbf{j}$  is the electric current.

In theory, we can stop here, analytically this model is good enough to work with, but in order to solve the equations on a computer, we need an other approximation that will give the gyrokinetic equations. A good way to decrease the computation time is to reduce the number of dimensions. In this case, it can be done by averaging the Larmor motion<sup>4</sup> (also called the gyro-motion), thus the dimension will go from 6 (3 in space and 3 in velocity) to 5 (3 in space and 2 in velocity). The reason why this approximation can be made in a tokamak comes from the external magnetic field created by the coils that gives a very strong total magnetic field, thus the cyclotron frequency is much higher than any characteristic time scale relevant for the turbulence. Therefore, we do not need to know where the particle is on the Larmor motion, but we need to keep the effects of this rotation. To do so, we can see the particles as a small ring (that carries a current) perpendicular to the magnetic field and with a radius equal to the Larmor radius. With this approximation, the necessity of resolving the Larmor motion is removed and the time steps in the simulation can be a lot larger.

The gyrokinetic equations are given by [2]:

$$\dot{\mathbf{X}} = \frac{1}{D} \left[ v_{\parallel} \hat{b} + \left( \frac{v_{\parallel}^2}{B} \right) \nabla B \wedge \hat{b} + \frac{\mathbf{B} \wedge (\mu \nabla B - \bar{\mathbf{E}})}{B^2} \right], \quad (9)$$

$$\dot{v}_{\parallel} = -\frac{1}{D} \left( \mathbf{B} + v_{\parallel} \nabla B \wedge \hat{b} \right) \cdot (\mu \nabla B - \bar{\mathbf{E}}), \quad (10)$$

$$-\nabla_{\perp} \frac{\rho_i^2}{\lambda_{Di}^2} \nabla_{\perp} \Phi = e (1 - \nabla_{\perp} \rho_i^2 \nabla_{\perp}) (\bar{n}_i - n_e), \quad (11)$$

where  $\mathbf{X}$  is the ions gyro-center position,  $v_{\parallel}$  its velocity along the magnetic field.  $\bar{\mathbf{E}}$  is the gyro-averaged electric field,  $\hat{b} = \mathbf{B}/B$ ,  $\mu = v_{\perp}^2/2B$  is the magnetic moment.  $D = 1 + (v_{\parallel}/B) \hat{b} \cdot (\nabla \wedge \hat{b})$ ,  $\rho_i$  is the ions gyroradius,  $\lambda_{Di}$  the ion Debye length,  $\bar{n}_i$  is the gyro-center ion density (electrons have a smaller gyroradius, therefore  $\bar{n}_e \approx n_e$ ). The two first equations describe the gyro-averaged motion of the particles and the last one is the gyrokinetic Poisson equation.

<sup>3</sup>Named after the famous British mathematical physicist James C. Maxwell born in 1831.

<sup>4</sup>Named after the British physicist and mathematician Joseph Larmor born in 1857.

## 2.3 XGC1 Algorithm

XGC1 is an electrostatic gyrokinetic ion-electron particle code using the particle-in-cell (PIC) method and the cylindrical coordinate system[2]. The code is a total-f code that uses the benefits of the  $\delta f$  and full-f methods. The full-f method consists in evolving the full distribution of the particles. It is usually opposed to the  $\delta f$  method that consists in evolving only the perturbation from the equilibrium. The  $\delta f$  method has trouble resolving the edge of the plasma where the perturbations attain the size of the equilibrium and the full-f has difficulties resolving the small perturbation due to the noise. The total-f creates the particles for the PIC method as in  $\delta f$  but the background is evolved too. The particle distribution is initialized as Maxwellian, and therefore is not an equilibrium solution of the gyro-kinetic equations (9-11). A few time steps are needed before having a realistic simulation.

A PIC code simulates particles by computing the electric field created by the particles and not by computing the force between two particles. Therefore a charge deposit has to be done and is usually the bottleneck of this kind of code. The steps of the simulation are the following[2]:

- Apply the charge deposit method,
- Compute the electric potential with the gyrokinetic Poisson equation,
- Calculate the electric field,
- Write the output,
- Evolve the particles.

## 2.4 Construction of the Mesh

The mesh of XGC1 consists of a number of poloidal planes ( $n_p$ ) where an unstructured triangular mesh is constructed and with a constant distance between them. In order to simplify the construction of the mesh and to decrease the amount of memory requested by the code, all the poloidal planes are equipped with the same mesh.

In equation (12), the field-line following poloidal variable is introduced, we denote  $B_T$  the toroidal component of the magnetic field,  $\mathbf{B}_P = \nabla\phi \wedge \nabla\psi$  the poloidal component of the magnetic field with  $\psi = RA_\phi$ ,  $\mathbf{A}$  the magnetic potential vector, and let  $ds$  being along the surface of constant  $\phi$  and  $\psi$ [2].

$$\lambda := \int \frac{B_T}{RB_P} ds \quad (12)$$

The magnetic field can be written  $\mathbf{B} = \nabla\psi \wedge \nabla(\lambda - \phi)$  which implies  $\lambda - \phi$  is constant along the B-field and thus  $\Delta\lambda = \Delta\phi = \frac{2\pi}{n_p}$ . The integral needs a reference line and is computed using the method of characteristics along with an ordinary differential equation solver. The mesh is constructed by using the intersections between the field lines and the poloidal planes (see figure 4), therefore  $\lambda$  is approximately used for the field-line following property. The vertices of the mesh are situated on a few magnetic flux surfaces (lines in a poloidal plane) and are separated by a distance of  $\Delta\lambda = \frac{2\pi}{n_p}$ .



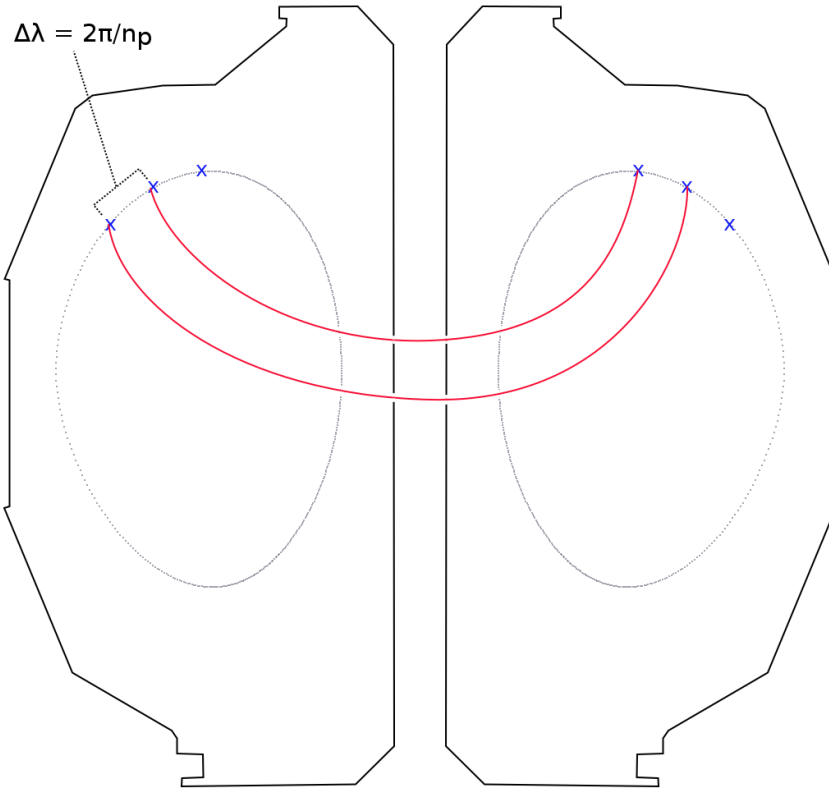


Figure 4: The grid (blue crosses) is constructed with the intersections between the poloidal plane and the field-line (red lines). The gray lines represent a magnetic flux surface. With this construction, the mesh respects the field-line following property.

Close to the X-point, the distance between two points becomes too small (due to the fact that  $B_P \rightarrow 0$ ) and a minimal value for  $B_P$  (in the formula) is set:  $\lambda := \int \frac{B_T}{R \max(B_P, B_{p0})} ds$ . An other problem comes from the part inside the separatrix, since the magnetic flux surfaces are closed and we want a periodic mesh, the following requirement is made:  $\Delta\lambda = \frac{\lambda_{2\pi}}{N}$  where  $\lambda_{2\pi}$  is the variation in  $\lambda$  over one loop[2].

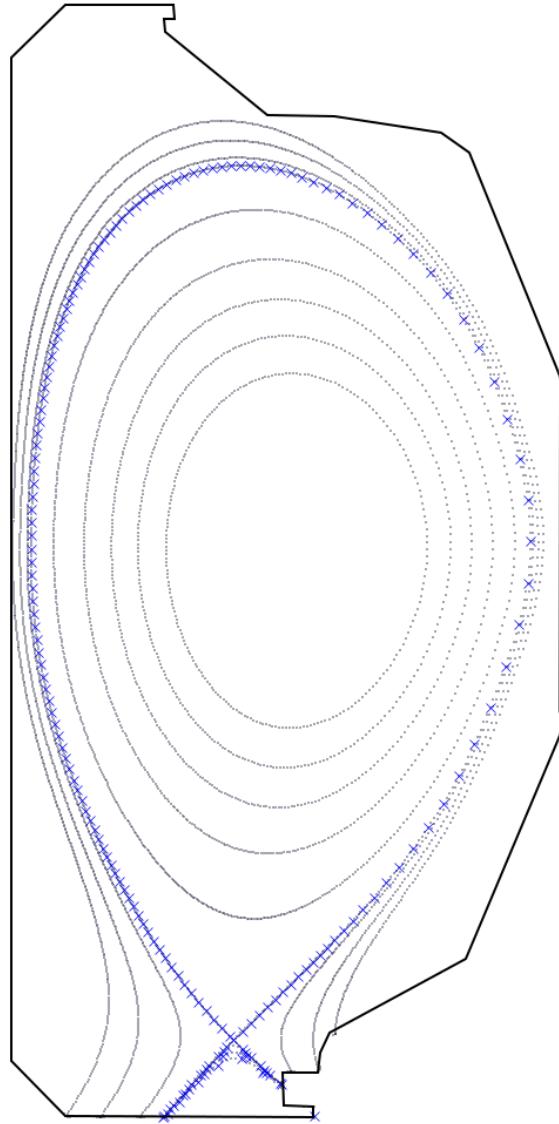


Figure 5: Example of a poloidal mesh for the XGC1 code (reduced mesh shown for clarity). The complete mesh is composed by  $n_p$  poloidal plane, and each one has the same mesh. The vertices of the mesh are situated on a few different magnetic flux surfaces (only a small sample of flux surface is shown in this figure), and the distance between two vertices of the same magnetic flux surface is given by  $\lambda$  (except inside the separatrix and around the X-point [approximately done]). The blue crosses are taken from a simulation, but only 20% of the vertices on a magnetic flux surface are shown. The full simulation (in the case of the particle pinch simulation) uses around 57k vertices on each poloidal plane. The X-point is shown at the bottom where the separatrix has an X-shape.

## 2.5 Output Files

One of the big problems XGC1 faces is the size of the outputs that a simulation can generate. The output is composed of a several different files. Each turbulence file can reach more than 10GB and to restart the code, the file has a size that can reach 1TB.

First, the file called 'unit.m' contains some information like the time step used in

the simulation, the number of time steps between two output writing, the position and the poloidal magnetic flux of the magnetic axis, the ion mass and charge, the thermal velocity, etc.

The file called 'xgc.mesh' contains all the information about the mesh: the position of the vertices on a poloidal plane, the equilibrium poloidal magnetic flux on the mesh, the number of poloidal planes, the next node for the field line following property, etc.

The magnetic field information is contained in the file 'xgc.bfield' and are given on the mesh.

Flux-surface averaged moments of the particle distribution function (e.g. electron density, temperature, etc.) are saved in the file 'xgc.oneddiag' as a function of the time step and the poloidal magnetic flux. For the equilibrium data, they are usually given as a function of  $\psi_n$  (or  $\psi$ ), therefore the size of theses quantities is small. The file contains more than 100 entries, therefore the file cannot be fully described and the reader should go look at them if he wants more information.

The last files are split by time steps 'xgc.3d.\*' and they contain all the fluctuation information including the electric potential, ion density and electron density fluctuations. Theses files are the only ones that are read during the synthetic diagnostic computation (the others are read once at the beginning and never called again).

## 2.6 Interpolation of the Data

The BES synthetic diagnostic requires density fluctuations at points in between the XGC1 poloidal planes, therefore an interpolation is done on each planes and along the magnetic field lines. A particular spatial point between XGC1 poloidal planes is mapped along the magnetic field lines to the poloidal plane toroidally in front and behind the point. An interpolation of the density fluctuations on the unstructured mesh on each plane is done onto the mapped point using some SciPy[10] methods, and the choice is given between a linear one (**LinearNDInterpolator**) or a cubic one (**CloughTocher2DInterpolator**). Then a linear interpolation is done between the two points along the field line, giving the final interpolated value of the density fluctuations at that point. In figure 6, an example of density interpolation is shown. Around  $R = 2.2$ , some fluctuations are present and the aim of the BES diagnostic is to analyze them.

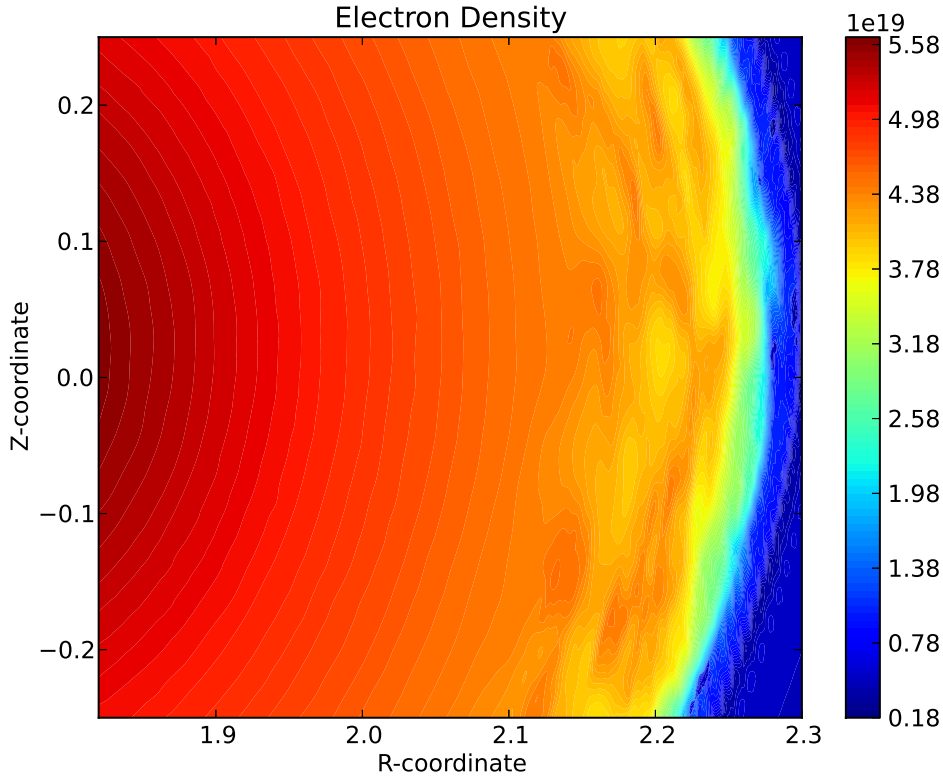


Figure 6: Interpolation of the electron density with a linear interpolation on each toroidal planes,  $\Delta\phi = 0.05$ , onto the first plane of XGC1 and from the bisector plane of the second one (they are separated by  $\frac{2\pi}{16} \approx 0.4$ ).

### The Field Line Equations

The equation of the field lines can be found by using (13) where  $d\mathbf{l}$  is an infinitesimally small vector along the magnetic field.

$$d\mathbf{l} \wedge \mathbf{B} = \begin{pmatrix} dR \\ dZ \\ R d\phi \end{pmatrix} \wedge \begin{pmatrix} B_R \\ B_Z \\ B_\phi \end{pmatrix} = \mathbf{0} \quad (13)$$

The differential equation is given in (14) and uses the relation obtained in (13). The equation (15) gives the length of the field line and is used to compute the weight of the interpolation method.

$$\frac{d\mathbf{l}}{d\phi} = \begin{pmatrix} \frac{RB_R}{B_\phi} \\ \frac{RB_Z}{B_\phi} \\ R \end{pmatrix} \quad (14)$$

$$\left\| \frac{d\mathbf{l}}{d\phi} \right\| = R \sqrt{\left( \frac{B_R}{B_\phi} \right)^2 + \left( \frac{B_Z}{B_\phi} \right)^2 + 1} \quad (15)$$

## The Runge-Kutta Methods

The computation of these equations is done with a Runge-Kutta method<sup>5</sup> (RK) of order two (Heun's method), but a few others are implemented (Euler, generalized RK2, RK3 and RK4)[6]. The choice of using a Runge-Kutta method of order two is due to the fact that the relative error for a simulation with 16 planes is already at  $10^{-4}$  with a few step (see figure 9). A common way to represent the Runge-Kutta methods is with the Butcher tableau<sup>6</sup> (see (16) with (17) for making the link to the RK methods[3]). In a Butcher tableau for an explicit method, the matrix A is always lower triangular with a null diagonal. An easy way to compute the order condition is by using the tree diagrams that avoid doing Taylor expansion to high order, but this method will not be presented in this thesis[3][6].

The interpolation code allows only the use of explicit Runge-Kutta methods<sup>7</sup>, but it should be enough to integrate due to the small size of the interval and the problem should not be too stiff. In figure 1, the implemented methods are shown in the form of the Butcher tableau.

$$\begin{array}{c|ccc} c_1 & a_{11} & \dots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & a_{ss} \\ \hline & b_1 & \dots & b_s \end{array} \quad (16)$$

$$\begin{cases} \frac{dy}{dt} = f(t, y) \\ y_0 = y(0) \\ k_i = f\left(t_n + c_i h, y_n + h \sum_{j=1}^{s-1} a_{i,j} k_j\right) \\ y_{n+1} = y_n + h \sum_{i=1}^s b_i k_i \end{cases} \quad (17)$$

In order to prove the order of the Heun's method (the others will not be proven in this report, but can be found in [6]), we will use two Taylor expansions and assume for simplicity that the problem does not depend explicitly on the time: the first one on the exact solution (18) and the other one on the second step of the method (19) where  $\frac{dy}{dt} = f(y)$ . The notation used is  $h$  for the time step,  $y(t)$  for the exact solution and  $y_i$  for the approximation at the time step  $t_i$ .

$$y(h) - y(0) = f(0)h + \frac{1}{2}f(0)f'(0)h^2 + \mathcal{O}(h^3) \quad (18)$$

$$f(y_0 + hf(y_0)) = f(y_0) + hf(y_0)f'(y_0) + \mathcal{O}(h^2) \quad (19)$$

The truncation error is given by the difference between the exact Taylor expansion and the numerical scheme. The error is given in (20) where  $y_0 = y(0)$  (initial condition of the scheme) and shows that the convergence order is 2.

<sup>5</sup>Named after the German mathematician and physicist Carl D. T. Runge, born in 1856 and the German mathematician Martin W. Kutta born in 1867.

<sup>6</sup>Named after John C. Butcher born in 1933, a New Zealander mathematician specialized in numerical methods for solving ordinary differential equation.

<sup>7</sup>In the case of an implicit method, the code is putting all the implicit values to zero.

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

(a) The Euler method has a convergence order of 1.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \alpha & \alpha & 0 \\ \hline & 1-\frac{1}{2\alpha} & \frac{1}{2\alpha} \end{array}$$

(b) The RK2 methods have a convergence order of 2.  $\alpha = 1$  is the Heun's method and  $\alpha = 0.5$  the midpoint rule.

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & -1 & 2 & 0 \\ \hline & 1/6 & 2/3 & 1/6 \end{array}$$

(c) The Kutta's third order method has a convergence order of 3.

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 \\ & 1/6 & 1/3 & 1/3 & 1/6 \end{array}$$

(d) The classic RK4 method has a convergence order of 4 and is the last RK method with the same number of step than the order.[3]

Table 1: Butcher Tableau of the different Runge-Kutta methods implemented.

$$\begin{aligned} y_1 - y(h) &= y_0 - y(h) + \frac{h}{2} [f(y_0) + f(y_0 + hf(y_0))] \\ &= \frac{h}{2} [f(y_0) + f(y_0) + hf(y_0)f'(y_0) + \mathcal{O}(h^2)] - y(h) \\ &= \mathcal{O}(h^3) \end{aligned} \tag{20}$$

In figure 7, the convergence of the different methods is shown on a test function ( $f(y) = y^4$ ) and clearly follows the theoretical convergence. For the RK2 methods, only the Heun's one is shown.

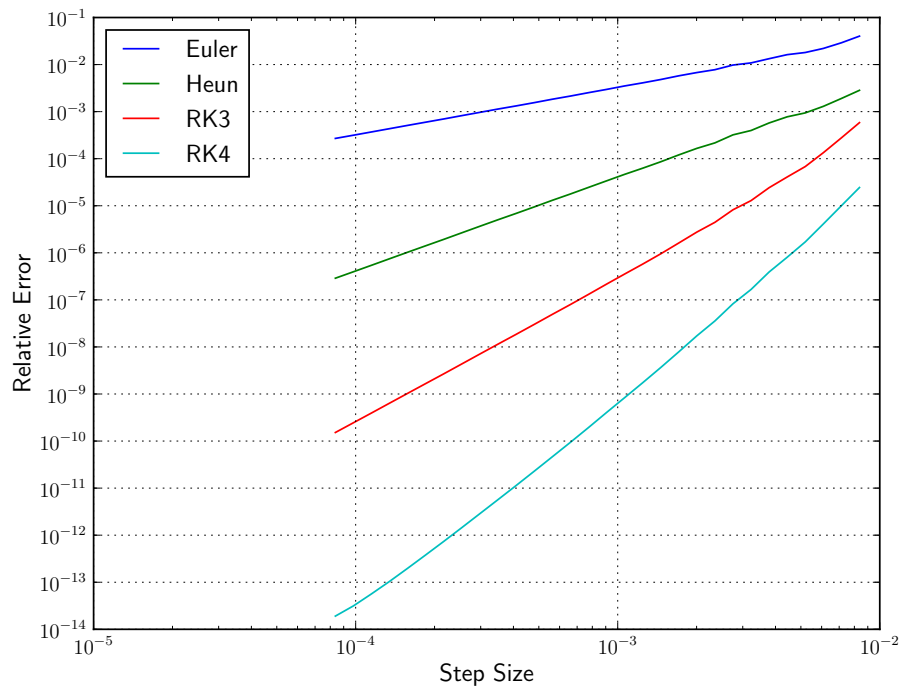
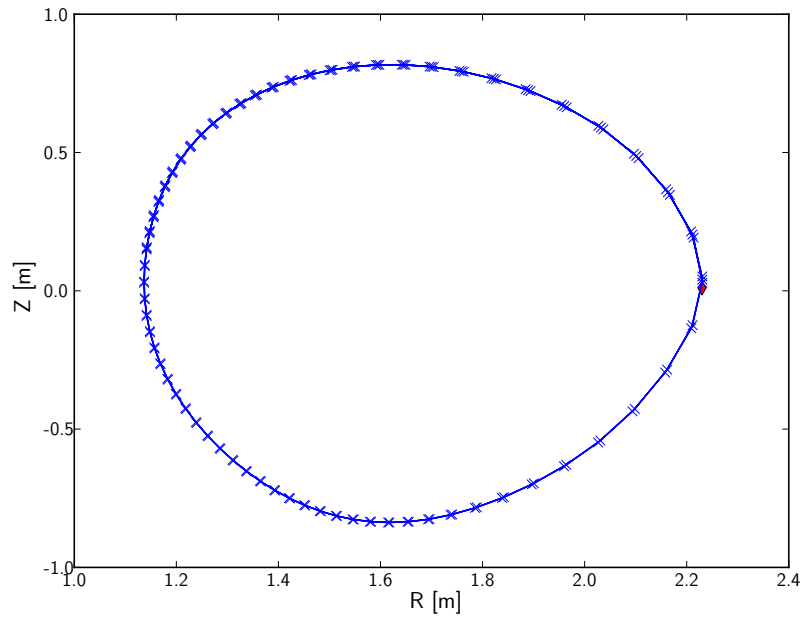


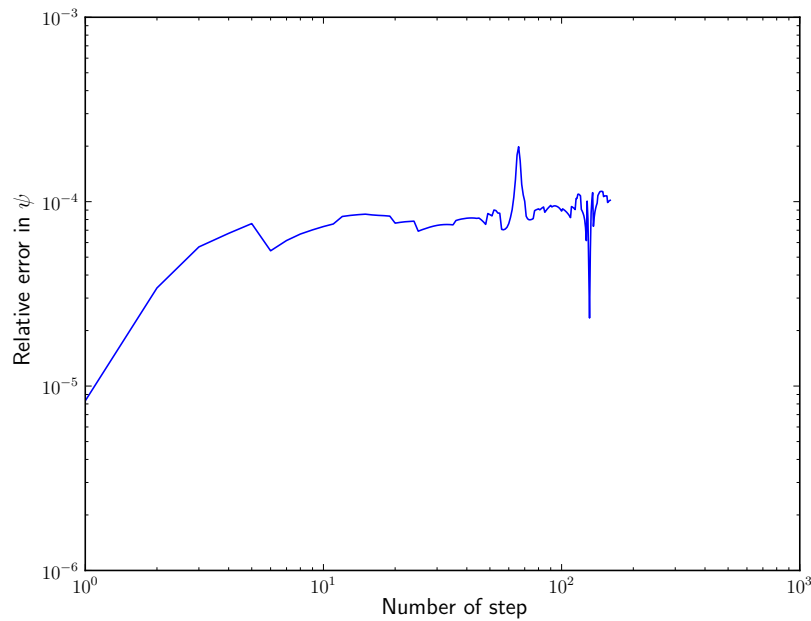
Figure 7: Convergence of the different Runge-Kutta methods as a function of the number of steps on a test function ( $f(y) = y^4$ ). The relative error is computed with a reference value computed on 3000 steps (step size of around  $3 \cdot 10^{-6}$ ). The convergence follows well the theoretical ones.

### Benchmarking of the Interpolation

From (4),  $\psi$  is a conserved quantity along the field line and a good way to check if the interpolation is well done. Figure 8 shows the evolution of  $\psi$  and the position during 10 toroidal rotation (16 steps is equivalent to one rotation). The interpolation seems to follow exactly the magnetic surface (except for the small error due to the numerical approximation) and keeps a poloidal magnetic flux close to a constant.



(a) Position computed with the field line integration. The positions seems to stay on the same magnetic surface. The red diamond is the initial position (around  $R = 2.2$  and  $Z = 0$ ).



(b)  $\psi$  as a function of the number of step. In theory, it should be constant, but due to the numerical approximation, a small error is introduced.

Figure 8: Field line integration along 10 toroidal rotations with  $\Delta\phi = 0.05$  and a linear interpolation on the toroidal planes.

Figure 9 and 10 show the convergence of the electron density and the coordinate computed with the field line integration as a function of the step size of the RK method. Figure 9 shows a convergence order of 2 (as the Runge-Kutta method should) and around



10 intervals ( $\Delta \approx 0.04$ ) is enough for having a good precision. Figure 10 shows the convergence of the position (R,Z) and the length of the field lines and one can see an order 2 convergence. The reason between the difference of accuracy between the R,Z,s and the data interpolation convergence comes from a few computation between the two of them and the toroidal plane interpolation. In the convergence of the length of the field lines, one can see a small discontinuity. It comes from the linear interpolation of the field that makes the function non differentiable at a few points. This will be clearer in figure 11.

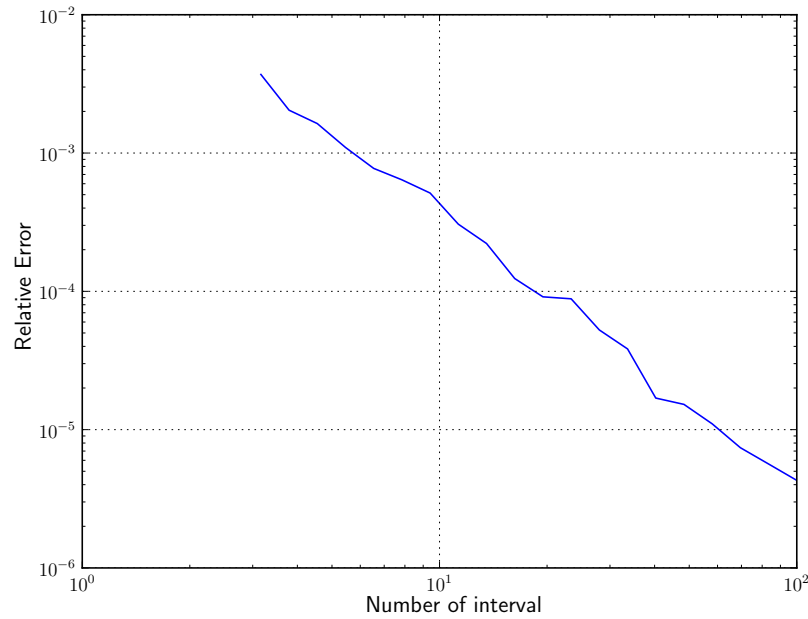


Figure 9: Convergence of the electron density as a function of the number of step in the Runge-Kutta method. The linear interpolation on the toroidal plane is used. The reference value is computed with 1000 steps.

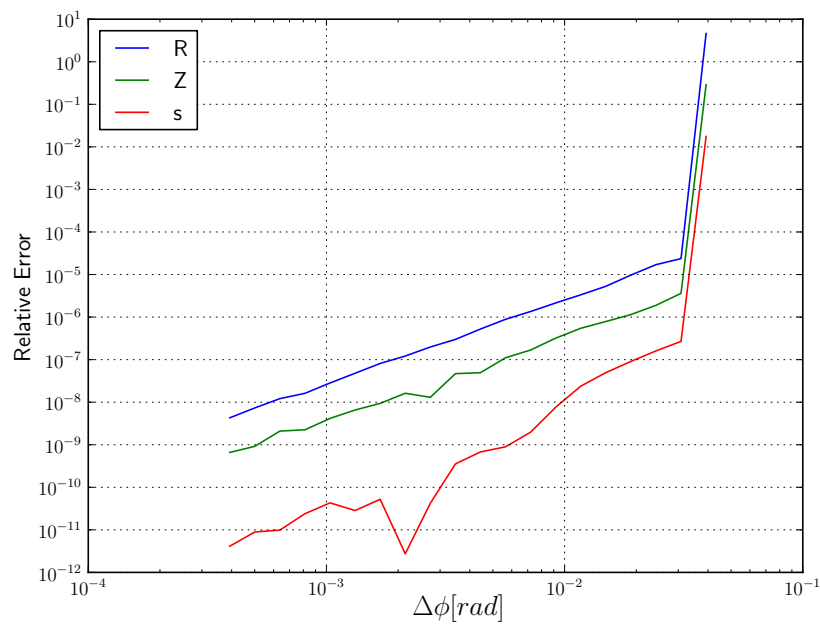


Figure 10: Convergence of the R-Z coordinates and the distance along the field lines (s) as a function of the step size. The reference value is computed with 5000 steps.

### 3 Synthetic Diagnostic for the Beam Emission Spectroscopy

The synthetic diagnostic is composed by two main part: the neutral beam and the optics. They will be discussed separately and verified by some convergence studies.

#### 3.1 Neutral Beam

The neutral beam is the key element of the BES diagnostic, as it is the way information about the plasma turbulence characteristics is obtained by this diagnostic. Usually neutral beams used are for heating the plasma, therefore the diagnostic does not take an important volume around the tokamak. They are designed for having a large power (but small enough for not damaging the wall) and depositing it in the core of the plasma. A lot of diagnostics use theses beams and in the case of the BES, it uses them for measuring the density fluctuations. A realistic beam model is used for the BES synthetic diagnostic, including multiple beam energy components, each one with a fraction of the total energy and their own mass and energy (mono-energetic components). The density of the beam is assumed to be a Gaussian perpendicular to the direction of propagation with a constant width (no beam divergence), therefore, only the central line is computed. The BES emission of the beam is computed with the beam density, thus a first step consists in computing it.

#### Beam Density

As the beam propagates into the plasma, its density decreases exponentially due to both the ionization of the neutral particles and to the charges exchange with the plasma ions. The beam density can be computed by two different ways, a fast one that uses only the equilibrium density and temperature, and a slow one that computes it with the equilibrium plus fluctuation. In the case of the equilibrium data, the computation is done once at the beginning, otherwise it should be done at each time steps. As it will be seen later, the effect of the turbulent eddies is not very important on the density of the beam. In the case of the fluctuation computation, the beam velocity is assumed to be large enough for depending only on one time step ( $v_b > L_B/dt$  where  $v_b$  is the beam velocity,  $L_B$  is the distance traveled by the beam and  $dt$  is the time step). By making this assumption, we are also assuming one on the blob velocity should be small enough to be resolved by the time step. Usually the particles in the main component of the beam have an energy around 100[keV], therefore the restriction on the time step is of the order of  $dt \geq 10^{-7}$ [s].

The number of collisions ( $n_{col}$ ) per unit of volume and distance traveled by a neutral particle is computed with the help of the stopping coefficient ( $S_{cr}$ )[9] obtained in the ADAS database and gives the number of collision in an interval  $dz$  (21). From it, the differential equation for the beam density can be computed (22) where  $n_e S_{cr}$  gives the number of plasma electrons per unit of time met by an incident neutral particles, the multiplication by  $n_b$  is taking into account all the particles in the beam and  $dt = \frac{dz}{v_b}$  is the required time for a beam particle to travel a distance  $dz$ .

$$n_{\text{col}} = n_e n_b S_{\text{cr}} \frac{dz}{v_b}, \quad (21)$$

$$\frac{dn_b}{dz} = -\frac{n_{\text{col}}}{dz}, \quad (22)$$

The solution of the differential equation is given in (23) where  $n_{b,0}$  is the initial density of the beam,  $z_p$  is the distance from the source (along the central line),  $E$  is the energy of a particle in the beam,  $n_e$  is the electron density of the plasma, and  $m$  is the mass of a beam's particle. The ADAS beam stopping coefficient takes into account the following effects: spontaneous emission, stimulated emission, photoabsorption, collisions with electrons and ions, radiative and three body recombination[16].

$$n_b(z_p) = n_{b,0} \exp \left( - \int_0^{z_p} n_e(z) S_{\text{cr}}(E, n_e(z), T_e(z)) \sqrt{\frac{m}{2E}} dz \right) \quad (23)$$

To reduce the computation time, the beam is assumed to always have a Gaussian profile perpendicular to the propagation direction, thus only the simulation of the central line is done. The code creates a mesh between the source of the beam and the farthest point used for the optical system, and the number of points used in the discretization ( $\mathbf{Nz}$ ) is specified in the configuration file (given in annexes 6.3). In the current version, the beam density is computed on the mesh by doing a Gauss-Legendre quadrature formula with one point (also called midpoint rule)[1]. The convergence order of this method is 2 and the degree of exactness is 1. The proof is given in (24) where  $z = \frac{b+a}{2}$ ,  $h = b - a$  and  $c \in [a, b]$  by using a Taylor expansion with the Lagrange remainder.

$$\begin{aligned} & \left| \int_a^b f(x) dx - (b-a)f(z) \right| \\ &= \left| (b-a)f(z) + \left[ \frac{(a-z)^2}{2} - \frac{(z-b)^2}{2} \right] f'(z) + \frac{h^3 f''(c)}{24} - (b-a)f(z) \right| \\ &= \left| \frac{h^3 f''(c)}{24} \right| \end{aligned} \quad (24)$$

In figure 11, the convergence of the last point of the beam is shown with the exact computation and with a test function ( $f(x) = x^3$ ). The test function shows the exact convergence order, but for the XGC1 data, the convergence is worse (still seems to be second order, but with a lot of noise). As seen in the proof of the convergence order, the error is proportional to the second derivative of the integrand. When interpolating a function, some non-differentiable points can appear (e.g. in the linear case, between two intervals the first derivative is not continuous) and produce a non-smooth convergence of the quadrature formula. In this version of the code, nothing is done to remove this problem due to the fact that some oscillations can appear when using higher order interpolation.

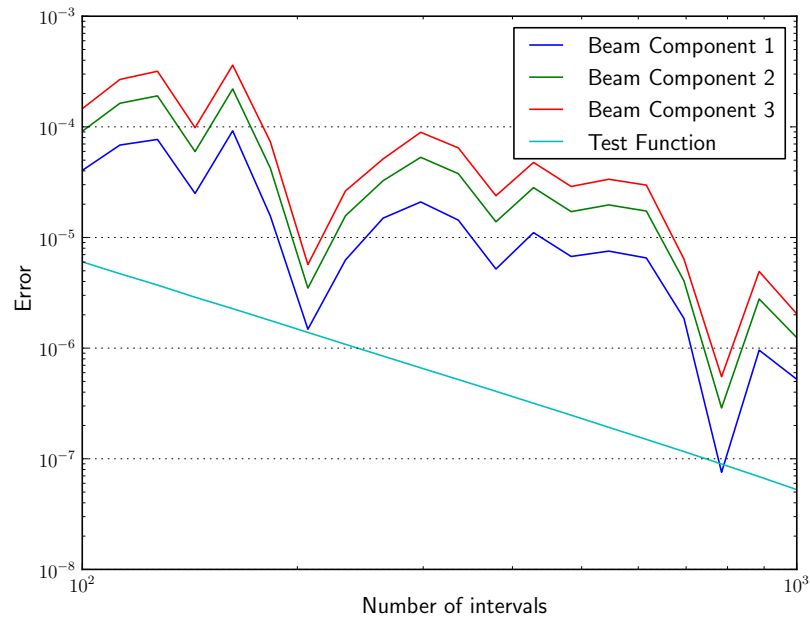


Figure 11: Convergence of the last point on the beam's mesh for all the component as a function of the number of step. The test function is using the same code, but replace the integrand by  $f(x) = x^3$ . In the real cases, the convergence is noisy, which is due to the interpolation that create some discontinuities in the first derivative of the integrand.

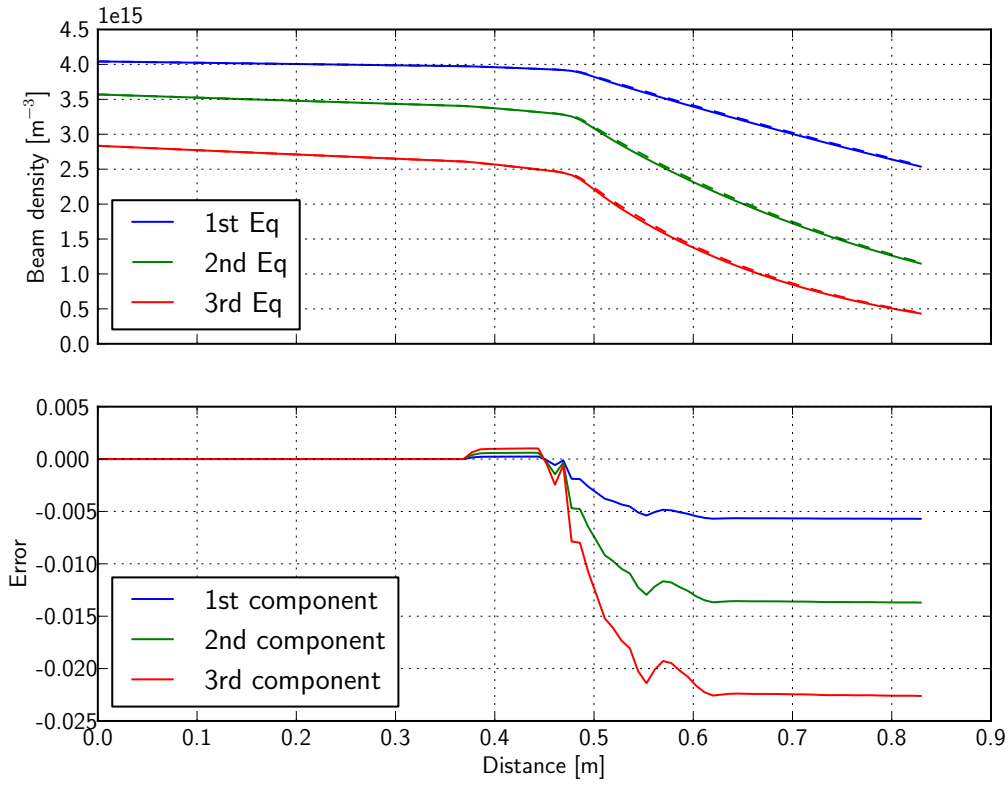


Figure 12: Beam density as a function of the distance for the case using equilibrium plasma profiles only (solid) and also for including the fluctuations in the plasma profiles (dashed). Different colors correspond to different beam components. The data comes from the particle pinch simulation. The number in the labels are for the component and 'Eq' is for the equilibrium. It can be seen that the error depends on the fluctuation only in the pedestal (i.e. near distance = 0.4 – 0.5, otherwise, it is a constant).

In the first graph of figure 12, the density of the beam for each component and for each cases (equilibrium or equilibrium plus fluctuations data) are shown. The second one shows the relative error ( $\frac{n_{b,eq} - n_{b,fluc}}{n_{b,eq}}$ ) between the two cases. The difference between the two comes from the fact that fluctuations are very important only at the edge (around the pedestal). Figure 13 shows this and, as expected, that the difference between the equilibrium and the fluctuation computation are really different only in the pedestal area (near distance = 0.4 – 0.5).

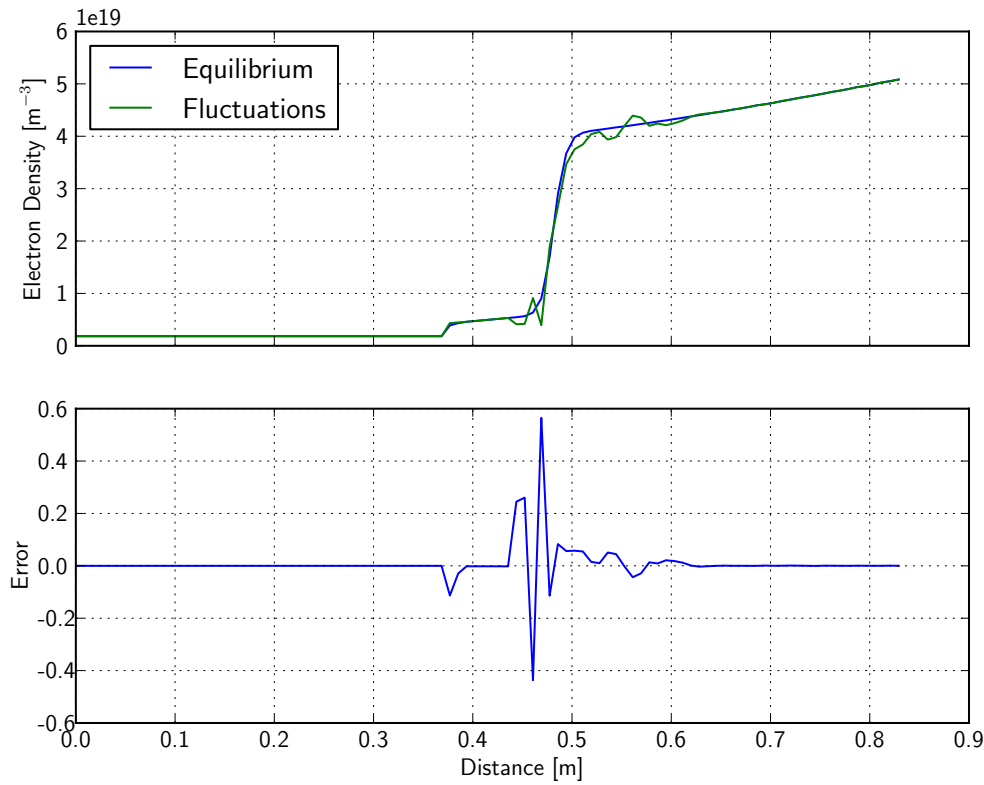


Figure 13: Electron density as a function of the distance (at the time step 150) and error between the two cases . The data comes from the particle pinch simulation.

In figure 14, the effect of the distance on the influence of each components of the beam on the beam density is shown. It shows that the lower energy components are lost sooner than the higher energy components as expected for this beam energy (75keV). The optical filter will have an importance later, but if we choose one that is centered on the wavelengths of the main component (see section 3.2), we can see that the secondary components are only useful in the area where the beam enter the plasma (an analysis on the full diagnostics will be done in section 4).

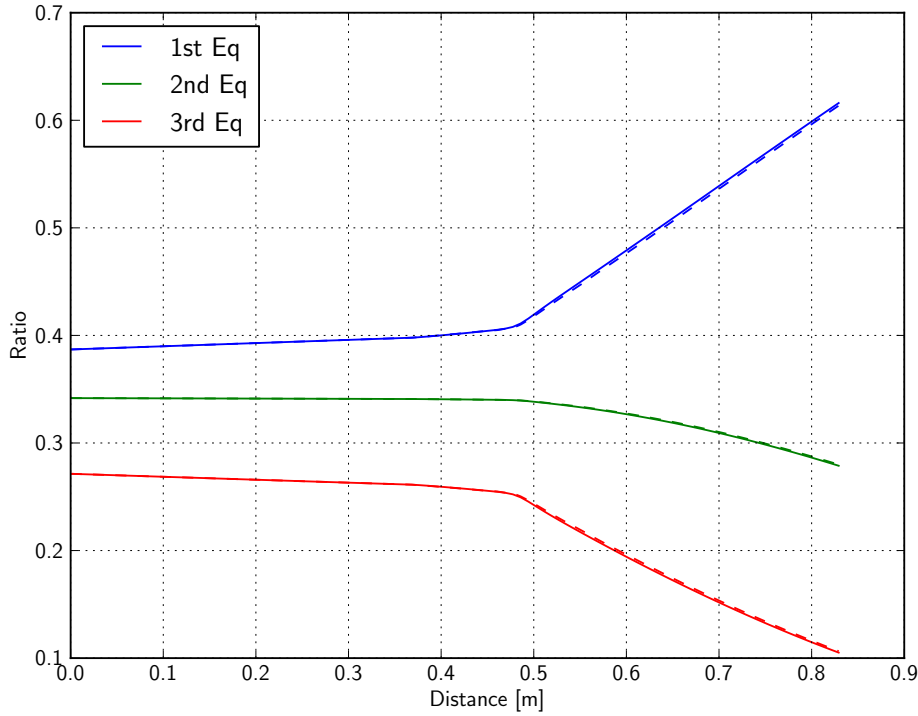


Figure 14: Ratio of the beam component density to the total density as a function of the distance. The data comes from the particle pinch simulation. Different colors correspond to different beam components and solid (dashed) lines are for the computation based on the equilibrium (equilibrium plus fluctuations). This graph shows how the different components influence the beam density.

### Photon Emission

The number of collisions per unit of time and volume ( $n_{\text{col}}$ ) that will give an excited particle at level  $n = 3$  is obtained with the ADAS database (ADF22)[4]:

$$n_{\text{col}}(z) = \langle \sigma v \rangle_{\text{ex}} n_e(z) n_b(z), \quad (25)$$

where  $\langle \sigma v \rangle_{\text{ex}}$  is the effective beam emission coefficient given by the ADAS database. The electron density is around the same order in all the core of the plasma, but for the beam density, a Gaussian distribution is assumed and will have a huge impact on the localization of the emission. As a first approximation, the number of collisions can be estimated as a Gaussian in the plane perpendicular to the central line.

**Instantaneous Emission** In this case, the lifetime of the excited particles is assumed to be null, and, therefore  $n_{\text{col}} = \varepsilon$  where  $\varepsilon$  is the photon radiance. This simplification is made in order to analyze the impact of the lifetime on the data given by the BES diagnostic.<sup>8</sup>

The instantaneous emission is shown in figure 15 and, as predicted, the emission is close to a Gaussian. The emission starts to be large around 0.4 [m] from the source,

<sup>8</sup>To choose this method, the configuration file parameter `t_max` should be put to 0.



which is due to the increased electron density in the pedestal region (this is further verified by the "starting line" following the contour of the toroidal curve of the tokamak). The fluctuations are not very well seen because their amplitude is not large enough in comparison with the scale.

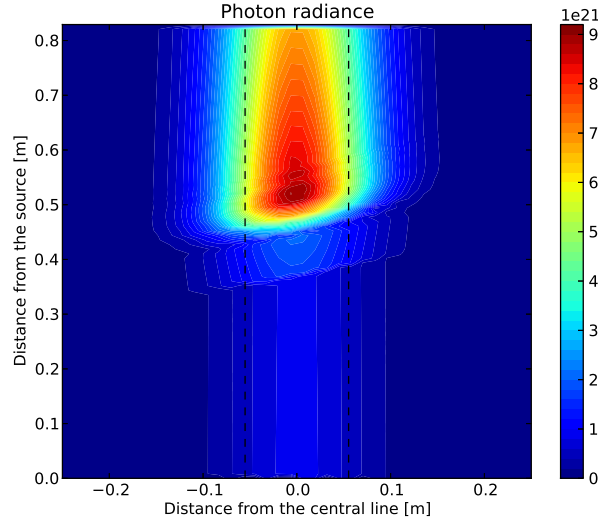


Figure 15: Instantaneous emission at the time step 81, with a mesh of 100 points and with the fluctuations for the beam density. The shape of the emission is limited by the pedestal (starting point) and the beam density (Gaussian profile). The two dashed black lines are at one standard deviation (for the beam density) of the central line.

**Non-Instantaneous Emission** The emission of photons is not an instantaneous process, the de-excitation is a process following a decay law ( $n(t) = n_0 e^{-t/\tau}$  where  $n(t)$  is the density of excited particles) with a characteristic time  $\tau$ . For common neutral beam energies, the excited neutral can move a few centimeters before emitting the de-excitation photon. The decay time  $\tau$  has a plasma density dependence due to collisions above density of circa  $10^{19} [m^{-3}]$ [8], which is around the usual value of density in a tokamak. We neglect this dependency in the code, but it is made flexible enough to upgrade it very easily. Neglecting this density dependence in the lifetime should modify a little bit the correlation length measured by the diagnostic, but should not be very important because for an order of magnitude difference in density the lifetime change less than half an order of magnitude[8, Figure 4].

In order to compute the emission, the number of collisions ( $n_{\text{col}}$ ) per unit of time and unit of volume is computed [as in (25)]:

$$n_{\text{col}} = n_e n_b \langle \sigma v \rangle \quad (26)$$

where  $\langle \sigma v \rangle$  is the excitation rate.

The excited population at a certain point is given by the integral on the particles path of the number of collisions multiplied by the exponential decay:

$$\begin{aligned} n_{\text{ex}}(z) &= \int_0^{\alpha\tau} n_{\text{col}}(z - tv_b) e^{-t/\tau} dt \\ &= \frac{1}{v_b} \int_0^{\alpha\tau v_b} n_{\text{col}}(z - z') \exp\left(-\frac{z'}{v_b\tau}\right) dz' \end{aligned} \quad (27)$$

where  $z$  is the position of the emission along the beam line,  $v_b$  is the beam speed,  $z' = tv_b$  and  $\alpha$  is the decay time multiplier used in the code (in the analytical case  $\alpha \rightarrow \infty$ ).

Now the density of particles de-excited at the position  $z_p$  during a unit time can be computed and gives the photon emissivity.

$$\varepsilon(z_P, t = 0) = -\frac{d}{dt} \left( n_{\text{ex}}(z_P) \exp \left[ -\frac{t}{\tau} \right] \right) \approx \frac{n_{\text{ex}}}{\tau} \quad (28)$$

In figure 16, the convergence of the non-instantaneous emission is shown as a function of the number of intervals (for (28) with the second line of (27)). The convergence order is 2, but for the real case, the convergence is very noisy, due to the interpolation. In figure 17, the non-instantaneous emission is shown, which is much smoother than the instantaneous case (Figure 15) especially around the pedestal. Figure 18 shows the relative difference between the two cases. As seen, the largest difference is in the pedestal region and one can clearly see the circular shape of the “starting line”.



Figure 16: Relative error of the lifetime integral as a function of the number of intervals for a test case (in green) and the real case (in blue). The test case uses exactly the same code but with the function  $f(x) = x^4$ .

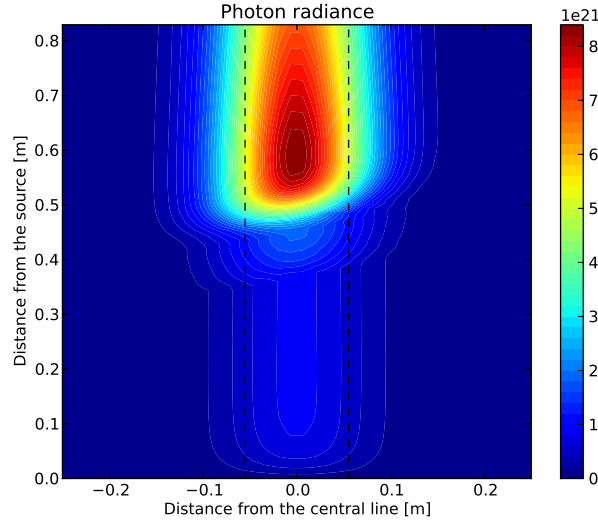


Figure 17: Non-instantaneous emission at the simulation time step 81, with a mesh of 100 vertices, 50 points for the lifetime and using the equilibrium plus fluctuations for the beam density. The shape of the emission is set by the pedestal (starting point) and the beam density (Gaussian profile). The two dashed black lines are at one standard deviation (for the beam density) from the beam central line.

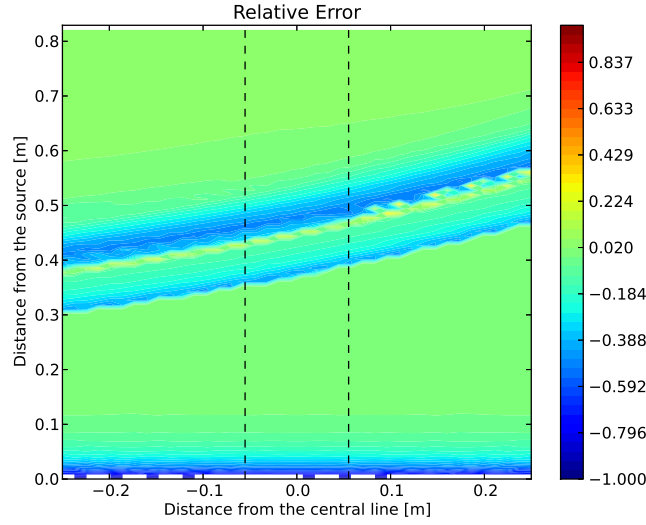


Figure 18: Relative difference between the instantaneous and non-instantaneous emission. The effect of the fluctuations is not seen due to the difference in the pedestal (the two blue lines). The two dashed black lines are at one standard deviation (for the beam density) of the central line.

**ADAS:** The ADAS files used (beam stopping and emission coefficients) contain two arrays of coefficients: the first one a two-dimensional array as a function of the electron density and beam energy, and the second array a one dimensional array as a function of the electron temperature (the dependency on this one is a lot smaller[16]). For the two-dimensional array, the temperature is kept at a reference value and for the one-

dimensional array, the beam energy and the electron density are kept at a reference value. In order to put the two arrays together, we are doing a first interpolation (linear) on the 2D array (that depends on the electron density and the beam energy) and multiply it by an interpolation of the coefficient normalized by the value at the reference temperature. In (29), the computation is shown where  $c_i$  is the array depending on  $i$  (the index of the quantities are dropped for avoiding a confusion with the arguments),  $\text{IN}(a, b)$  is the interpolation of  $a$  evaluated at  $b$ ,  $c_T(T_{ref})$  is the coefficient from the temperature array at the reference temperature.

$$c(n_e, E_b, T_i) = \text{IN}(c_{n,E}, [n_e, E_b]) \cdot \text{IN}\left(\frac{c_T}{c_T(T_{ref})}, T_i\right) \quad (29)$$

### 3.2 Optics

The aim of the code is to be usable for different kinds of tokamak (or even a stellerator), therefore the optics considered is not very complex. The effects of the optics taken into account are the sampling volume, the Doppler<sup>9</sup> effect and the filter.

The optics consists of a lens that will focus the light onto a fiber; the limits of the collected light is given by the size of the focus point (that has the shape of a disk, see figure 19). The focus point is the image of the fiber and the fiber is considered to completely fill the lens, therefore no information about the fiber is needed. In this thesis, the object called "focus point" is not the focus point of the lens, but the position where the sampling volume is the thinnest perpendicular to the optical path.

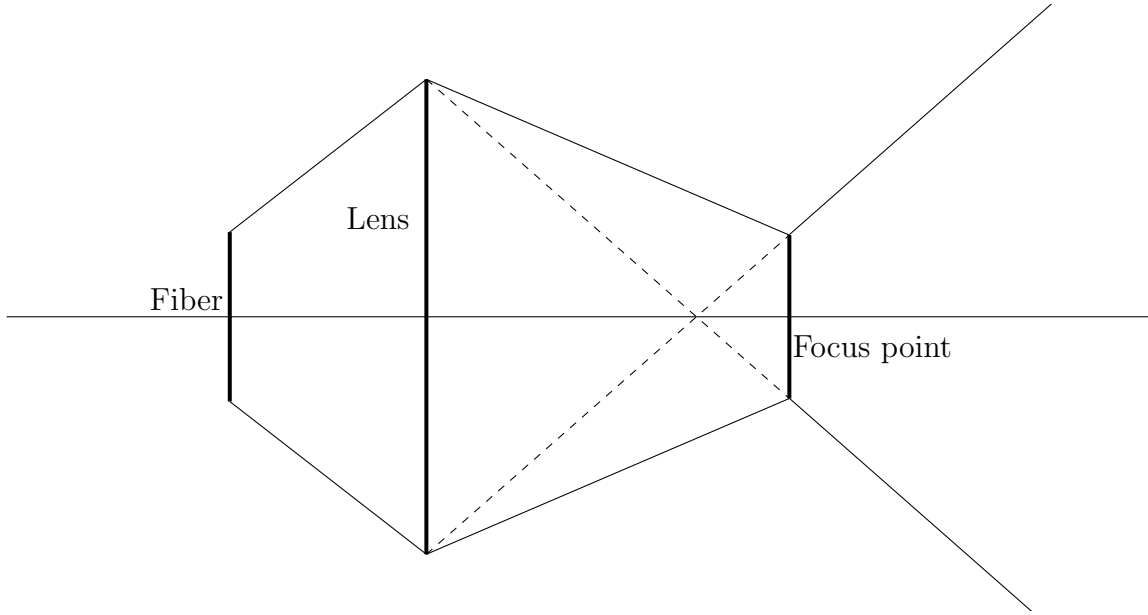


Figure 19: Optics considered by the code (not drawn to scale). The thick lines are the objects and image considered and the thin ones are the limits of the collected light.

#### Sampling Volume

If the emission is isotropic, the intensity of light received by a fiber is given by the following equation:

$$I = \frac{1}{4\pi} \int_z \frac{\iint_{\Sigma} \varepsilon \Omega d\sigma}{\iint_{\Sigma} \Omega d\sigma} dz, \quad (30)$$

where  $z$  is the central line of the optics,  $\Sigma$  is the plane perpendicular to  $z$  and inside the sampling volume and  $\Omega$  is the solid angle. The denominator of the fraction is called the etendue (or throughput) and is conserved along the trajectory of the light as long as only perfect reflections and refraction are applied.

The integral along the  $z$ -axis is made with a Gauss-Legendre quadrature formula[1] (of order 2) and should be done between  $\pm\infty$ , which is why a cut-off has to be made. A natural characteristic length is the beam width along the optical axis and it has been chosen as a normalization. The choice of the cut-off and the precision (number of points

<sup>9</sup>Named after the Austrian mathematician and physicist Christian A. Doppler, born in 1803.

that define the intervals) are given to the user with the variables `int` and `Nint` in the configuration file (section 6.3). The choice of the mid point of the integrals are the fiber focus points given by the variables `R,Z,phi` in the same file. As seen in section 3.1, the variable that will have the most influence on the emission is the beam density, therefore the integral should be close to the integral of a Gaussian. Due to this expected shape of the emission, a message is displayed if the accuracy is too low (the code asks at least 2 points per standard deviation).

Two different ways of computing the integrals perpendicularly to the optical path are available, and will be described in the next section: a first one that uses an approximation and a second one that does a numerical computation of the integrals.<sup>10</sup>

**Approximation of the Integrals** A very easy way to compute the perpendicular integral is by assuming that the emission depends only on the  $z$  coordinate inside the sampling volume, therefore the intensity integral simplifies to:

$$I \approx \frac{1}{4\pi} \int_z \varepsilon dz. \quad (31)$$

This approximation can be seen as taking into account only the emission from points along the central line and is motivated by the computation of the solid angle in Figure 28.

**Full Numerical Integration** The 2D integration can be done with different 2D quadrature formula of order 4,6 and 10 over a disk. The methods are given in Table 2[1] and in Figure 20. For this method, the precision is not changeable by the user, therefore the choice was to put a very accurate quadrature formula.

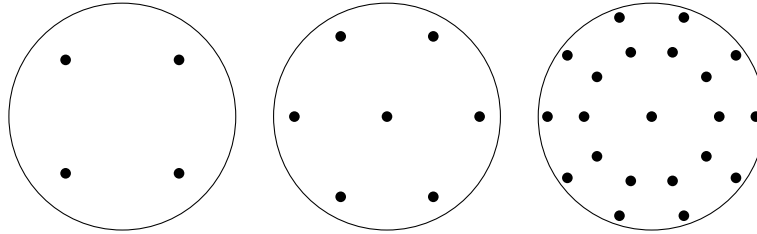


Figure 20: Points of the quadrature formula used for the computation of the photon radiance with the full computation.

In Figure 21, the convergence of the  $z$ -integral is shown and the convergence order is clearly 2. The relative error of both cases is very similar.

<sup>10</sup>The choice is made by the variable `type_int` (in the input file).

Index	Points	Weights
0 – 3	$(\pm \frac{h}{2}, \pm \frac{h}{2})$	$\frac{1}{4}$

(a) Method of order 4

Index	Points	Weights
0	$(0, 0)$	$\frac{1}{4}$
1 – 2	$(\pm \sqrt{\frac{2}{3}}h, 0)$	$\frac{1}{8}$
3 – 6	$(\pm \sqrt{\frac{1}{6}}h, \pm \frac{\sqrt{2}}{2}h)$	$\frac{1}{8}$

(b) Method of order 6

Index	Points	Weights
0	$(0, 0)$	$\frac{1}{9}$
1 – 10	$(hA_- \cos(\frac{2\pi k}{10}), hA_- \sin(\frac{2\pi k}{10}))$	$\frac{16+\sqrt{6}}{360}$
11 – 20	$(hA_+ \cos(\frac{2\pi k}{10}), hA_+ \sin(\frac{2\pi k}{10}))$	$\frac{16-\sqrt{6}}{360}$

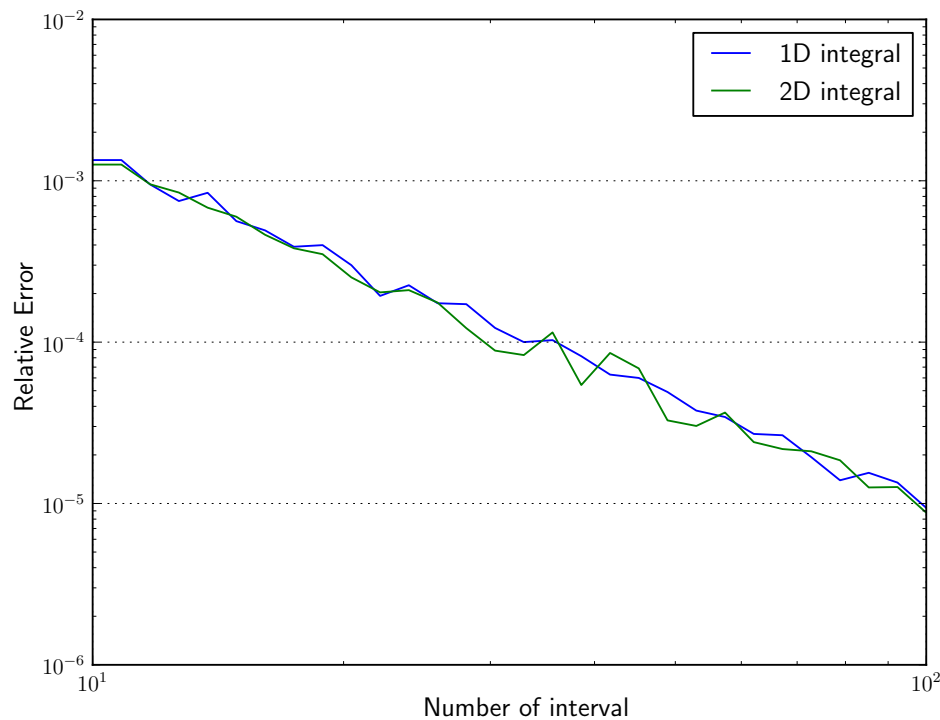
(c) Method of order 10 where  $k$  is the index and  $A_{\pm} = \sqrt{\frac{6 \pm \sqrt{6}}{10}}$ .Table 2: Points and weights for the quadrature over a circle of order 4,6 and 10 where  $h$  is the radius of the circle.

Figure 21: Convergence of the integral along the central line as a function of the number of step for the two cases.

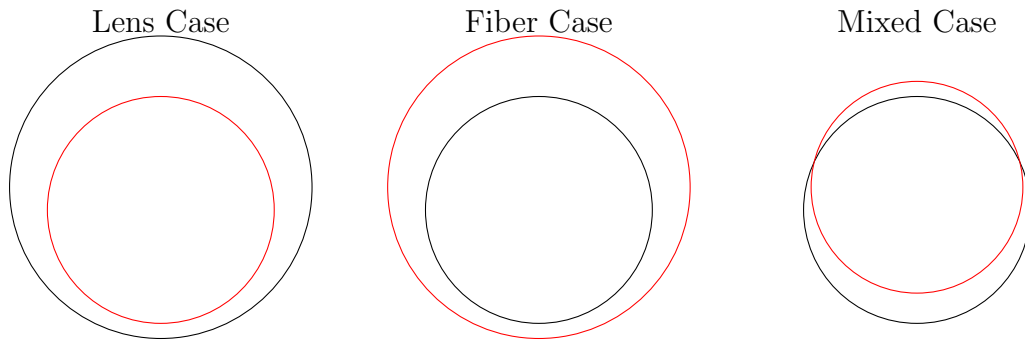
## Computation of the Solid Angle

To compute the solid angle occupied by the lens from a point (P) in the sight line, an integration of  $\Omega = \iint_S \frac{\mathbf{r} \cdot \hat{\mathbf{n}}}{r^2} dS$ , where S is the surface collecting the light,  $\mathbf{r}$  is the vector from P to the surface, and  $\hat{\mathbf{n}}$  is the unit vector normal to the surface, has to be done.

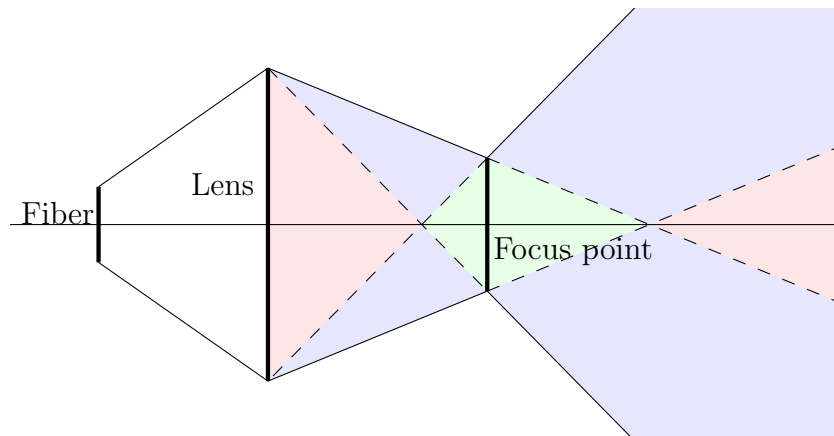
Depending on the position of the emission, three different cases can arise, the two first where the computation is more or less the same and a last one where the computation is far more complicated:

- Lens case
- Fiber case
- Mixed case

The three case are shown in Figure 22. The first one corresponds to the case where only the lens restricts the integrated area of the solid angle, the second where only the fiber restricts it, and the last one is a mixed case between the two.



(a) The three different views possible. In red, the lens is drawn and in black the focus point.



(b) Area of the three different cases. In green, it is the lens case, in red, the fiber one and in blue the mixed case.

Figure 22: The computation of the solid angle has three different cases: one where the lens is limiting the solid angle, on where it is the focus point and one where it is both of them.



**Lens and Fiber Cases** The solid angles in these cases can be computed analytically, as was done in [13]. The result is given in (32) and Figure 23 where  $L$  is the distance between the plane containing the disk and the emission position  $z_0$ ,  $R_{\max}$  ( $R_1$ ) is the maximal (minimal) distance between the point and the disk,  $K$  is the complete integral of the first kind,  $k^2 = 1 - \frac{R_1}{R_{\max}}$ ,  $\Lambda_0$  is the Heuman's lambda function,  $\xi = \arctan \frac{L}{|r_0 - r_m|}$ ,  $r_0$  is the distance between  $P$  and the axis perpendicular to the disk and passing at its center, and  $r_m$  is the radius of the disk.

$$\Omega = \begin{cases} 2\pi - \frac{2L}{R_{\max}}K(k) - \pi\Lambda_0(\xi, k) & r_0 < r_m \\ \pi - \frac{2L}{R_{\max}}K(k) & r_0 = r_m \\ -\frac{2L}{R_{\max}}K(k) + \pi\Lambda_0(\xi, k) & r_0 > r_m \end{cases} \quad (32)$$

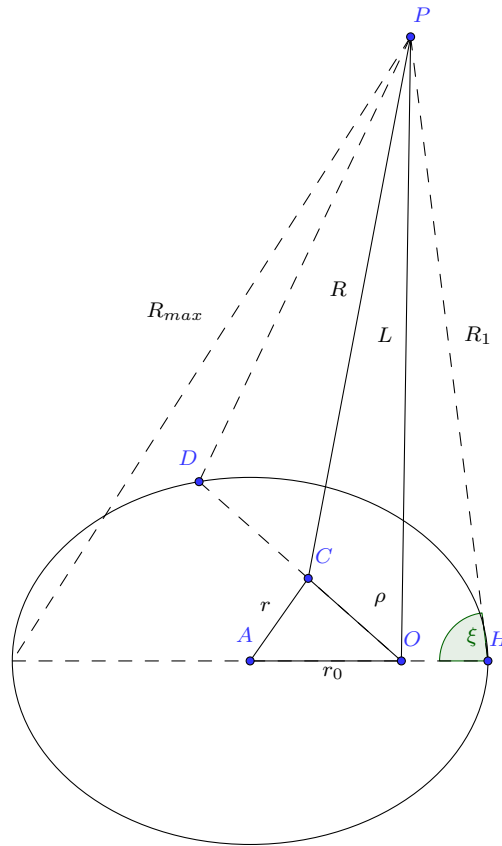


Figure 23: Copy of the drawing in [13] showing the situation for the computation of the solid angle in the fiber/lens case.

**Mixed Case** For the last case, the integration is more complicated, therefore a numerical integration has been used. In order to simplify the computation, the surface is split into two parts (by the blue line in figure 24): the first on the focus point and the second one on the lens. Another simplification is done by always computing the largest part of the surface and then removing it from the computation on the full disk using the analytical formula from the lens and fiber cases (for example in the figure, the lens computation will not be done between the solid red line and the blue one, but between the

blue and the red dashed line). Since the numerical integration can lead to negative values when the segment is very small, a threshold is chosen and the negative values are put to zero (the error is really small in the computation of the diagnostic, because the main contribution is from the lens case as it will be shown latter). The formula for the largest area of the disk is given in (33) where  $S$  is the surface of the object,  $S_i$  is a circle sector of the object,  $r_{\max}$  is the maximal radius at the angle  $\theta$ ,  $\omega$  are the weights for the radius and the angle, and  $\Delta\theta$  is the angle covered by the circle sector,  $f(x) = \frac{\mathbf{R} \cdot \hat{\mathbf{n}}}{\|\mathbf{r} + \mathbf{R}\|^{3/2}}$  and  $\mathbf{R}$  is the vector from the emission point to the center of the disk. The quadrature formulas used are Gauss-Legendre of order 4 for both the radius and the angle integration.

$$\begin{aligned}
 \iint_S f(r, \theta) dr d\theta &= \sum_i \iint_{S_i} f(r, \theta) dr d\theta \\
 &= \sum_i \int \frac{1}{2} \sum_j r_{\max}(\theta) \omega_{r,j} f(r_j, \theta) d\theta \\
 &= \frac{\Delta\theta}{4} \sum_{i,j,k} r_{\max}(\theta_k) \omega_{\theta,k} \omega_{r,j} f(r_j, \theta_k)
 \end{aligned} \tag{33}$$

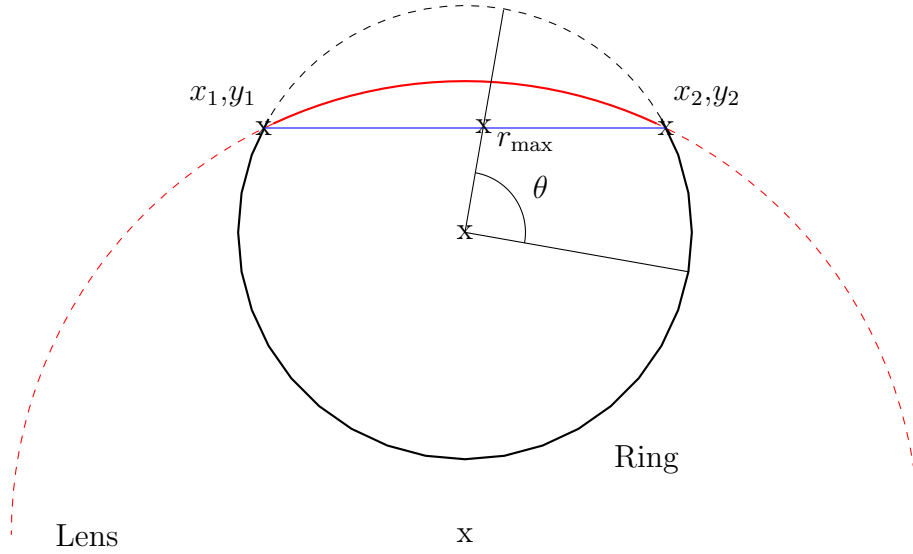


Figure 24: View from a point of the ring and the lens. The surface  $S$  receiving the photons is inside the black and red thick lines.

In Figure 25, the mixed case on the lens or the ring is computed with a different size of the segment removed and the relative difference between the analytical formula for the full case is shown. When the segment tends to 0, the analytical and the numerical formulas give exactly the same value. In Figure 26, a convergence study has been performed on the accuracy of the method as a function of the number of radial and angle intervals. As expected the radial case has a fourth order convergence but not the angle case. This difference is due to the domain of integration that is not smooth enough everywhere. The two intersection points lead to a discontinuity in the derivative of the perimeter. In the radial case, this non smooth domain is not seen (as the method is always using the same

number of angle), but for the angle integration, the choice of which angles are computed is very important.

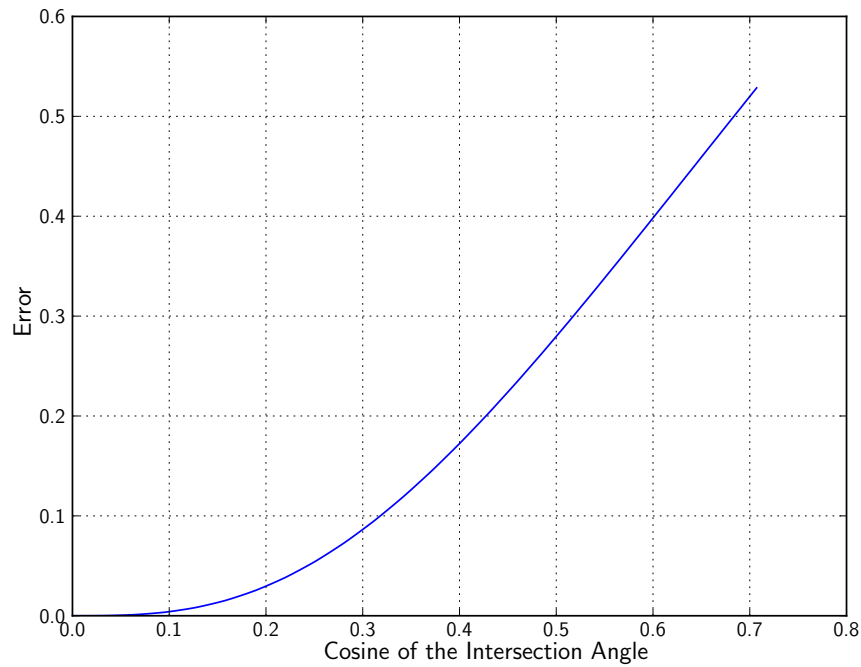


Figure 25: The relative difference to the analytical formula is shown as a function of the angle between  $x_1$  and the line L in Figure 27. The numerical integration converges to the analytical formula.

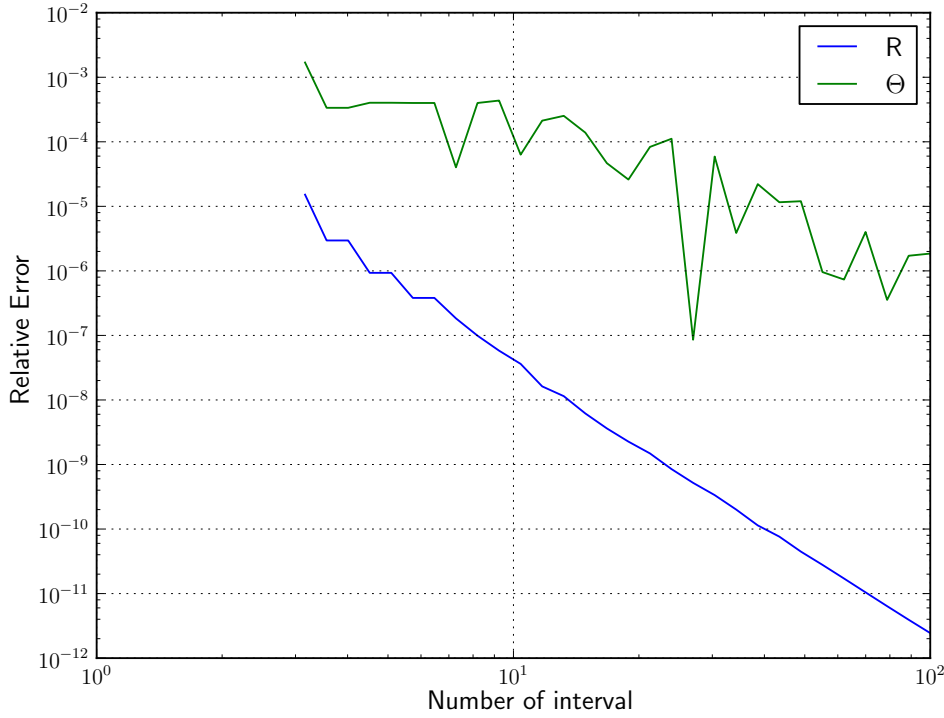


Figure 26: Convergence of the numerical integration of the solid angle as a function of the number of intervals for the radial and angle integral. For the angle case, the convergence is not smooth, because the integration domain is not enough regular.

**Computation of the Intersections** To compute the “intersections” between the lens and the focus point (intersections between the red and the black circles in figure 24), the equations (34) have to be solved where  $L$  is the distance between the lens and the focus point,  $z$  is the distance between the point  $\mathbf{P}$  and the lens,  $\mathbf{x}/\mathbf{y}$  is the coordinates (2D) of the intersection points on the ring<sup>11</sup>/lens. They come from the fact that  $x$  ( $y$ ) is a point on the perimeter of the focus point (lens) and the last one from the fact that  $x, y$  and  $P$  are on the same line.

$$\begin{cases} x_1^2 + x_2^2 &= r_r^2 \\ y_1^2 + y_2^2 &= r_l^2 \\ \frac{\mathbf{y}-\mathbf{P}}{z} &= \frac{\mathbf{y}-\mathbf{x}}{L} \end{cases} \quad (34)$$

First the last equation is used to have a linear relation between  $\mathbf{x}$  and  $\mathbf{y}$ .

$$\mathbf{y} = \left( \mathbf{P} - \frac{z}{L} \mathbf{x} \right) \beta \quad (35)$$

where  $\beta = \frac{1}{1-\frac{z}{L}}$ <sup>12</sup>. In the second part, we use the last equation in the second equation of (34) for having a relation between  $x_1$  and  $x_2$ :

$$x_1 = \frac{1}{2P_1} \left( \frac{L}{z} \left[ \mathbf{P}^2 - \left\{ \frac{r_l}{\beta} \right\}^2 \right] + \frac{z}{L} r_r^2 - 2P_2 x_2 \right) = A + Bx_2. \quad (36)$$

<sup>11</sup>The term of ring is sometimes used in this thesis and in the code for the focus point.

<sup>12</sup>In the code, this variable is called  $f$ .

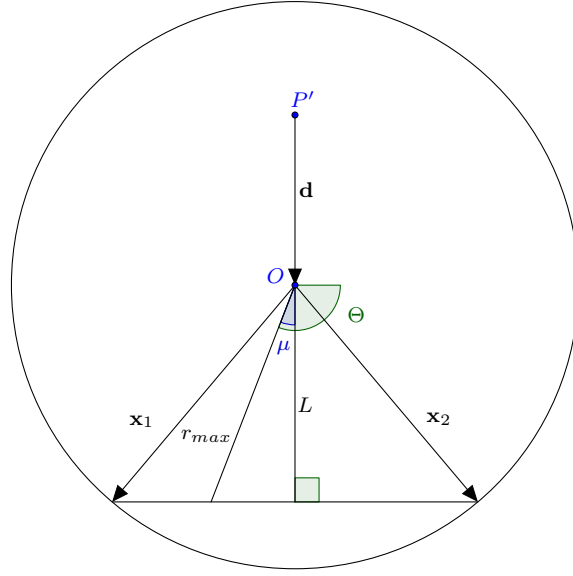


Figure 27:  $r_{\max}$  is computed with the help of the geometry.

The last step is to compute the value of  $x_2$  with the help of the first equation of (34):

$$x_2 = \frac{-2BA \pm \sqrt{4B^2A^2 - 4(A^2 - r_r^2)(B^2 + 1)}}{2(B^2 + 1)}. \quad (37)$$

**Computation of  $r_{\max}$**  In Figure 27, the situation is shown and will be used to explain the computation. The notations are the following:  $P'$  is the projection of  $P$  on the plane,  $x_i$  are the two intersections,  $\Theta$  is an angle of the quadrature formula and  $O$  is the center of the disk ( $O$  is supposed to be at  $(0, 0)$ ).

First the vector  $\mathbf{x}_i$  are used to know if  $\Theta$  is on the  $x$  side or on the  $P'$  side (for the  $x$  side, the computation of the distance between  $O$  and the line between  $x_1$  and  $x_2$  is for every angle and then we use the min between the disk radius and this distance). Then  $\mu$  is computed by using the scalar product between  $\mathbf{d}$  and  $\mathbf{r}_{\max}$ . The next step is to compute  $L$  and using it for computing  $r_{\max} = \frac{L}{\cos(\psi)}$ ,  $L$  can be easily computed with the formula of the distance between a line (given by two points  $[x_i]$ ) and a point ( $O$ ):  $L = \|(\mathbf{x}_1 - \mathbf{O}) - ((\mathbf{x}_1 - \mathbf{O}) \cdot \mathbf{n}) \mathbf{n}\|$  where  $\mathbf{n} = \frac{\mathbf{x}_1 - \mathbf{x}_2}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$ .

In Figure 28, the solid angle is shown as a function of the distance to the focus point and to the central line. As expected, the area where the solid angle is the highest is around the focus point. One can see three different cases due to the strong dependency of the mixed case on the distance from the central line (the two other are quite close to a constant along the distance from the central line).

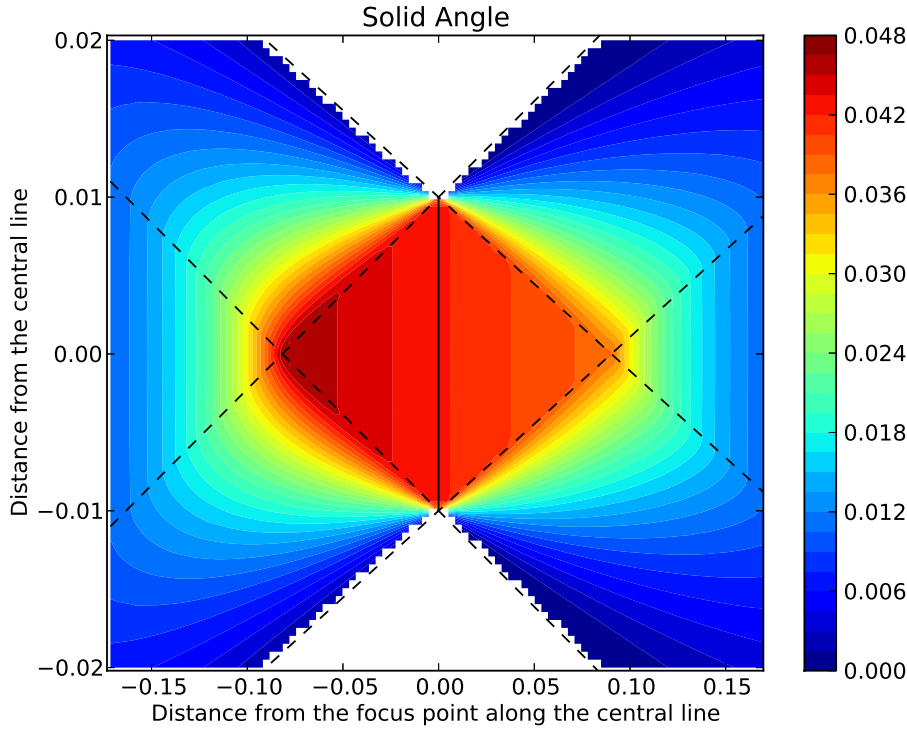


Figure 28: Solid angle as a function of the position. The three different cases are well shown in this picture. In the radial direction, the fiber/lens cases are close to a constant and the mixed case decrease rapidly. The lens is situated on the left and the dashed lines are the ones in Figure 22.

### Filter

The filter is modeled with an input file that gives the transmission coefficient as a function of the wavelength and by a linear interpolation of this file. In order to decrease the memory used for the computation, the filter is applied at the moment of the emission computation and not at the end of the diagnostic. The advantage of this method is that the wavelength dependency of the light collected by the lens is removed quickly and it allows to add the intensity of different wavelength together.

The beam is not going in the direction of the optics, therefore the Doppler shift is not constant and depends on the position. The Doppler shift allows to see the difference between the light emitted by the plasma and the one emitted by the beam. The computation is done with the law given in (38) where  $\lambda$  is the wavelength seen by the optic,  $\lambda_0$  is the one in the referential of the particle,  $v = v_b \cos(\alpha)$  is the velocity in the direction of the optic,  $\alpha$  is the angle between the optics line of sight and the beam direction and  $c$  is the speed of light.

$$\lambda = \lambda_0 \left(1 - \frac{v}{c}\right) \quad (38)$$

## Point Spread Function

This code is not computing the Point Spread Function (PSF)<sup>13</sup> yet but as it is the usual way to compute the diagnostic, it should be presented quickly in this thesis. For an explanation of the computation done in PSF method, see [15]. The aim of the PSF is to be able to reverse the diagnostic effects and give the possibility obtain the density fluctuation directly from the measured emission by the experimentalists. The PSF is the picture of a point source viewed by the optics and its relation to the image and the real object is given in (39) where  $M(x)$  is the image,  $S(x)$  is the source,  $*$  is the convolution operator and  $P(x)$  is the PSF. Figure 29 shows an example of the effect of a PSF (this is not an actual BES synthetic diagnostic). In the case of the BES diagnostic, it includes not only the optical effect but the physical aspect too (like the non-instantaneous emission). To calculate the PSF ( $P(x)$ ) from the synthetic diagnostic in this thesis, the inverse problem of deconvolving  $S(x)$  from  $M(x)$  needs to be solved, and can be non trivial.

$$M(x) = S(x) * P(x) \quad (39)$$

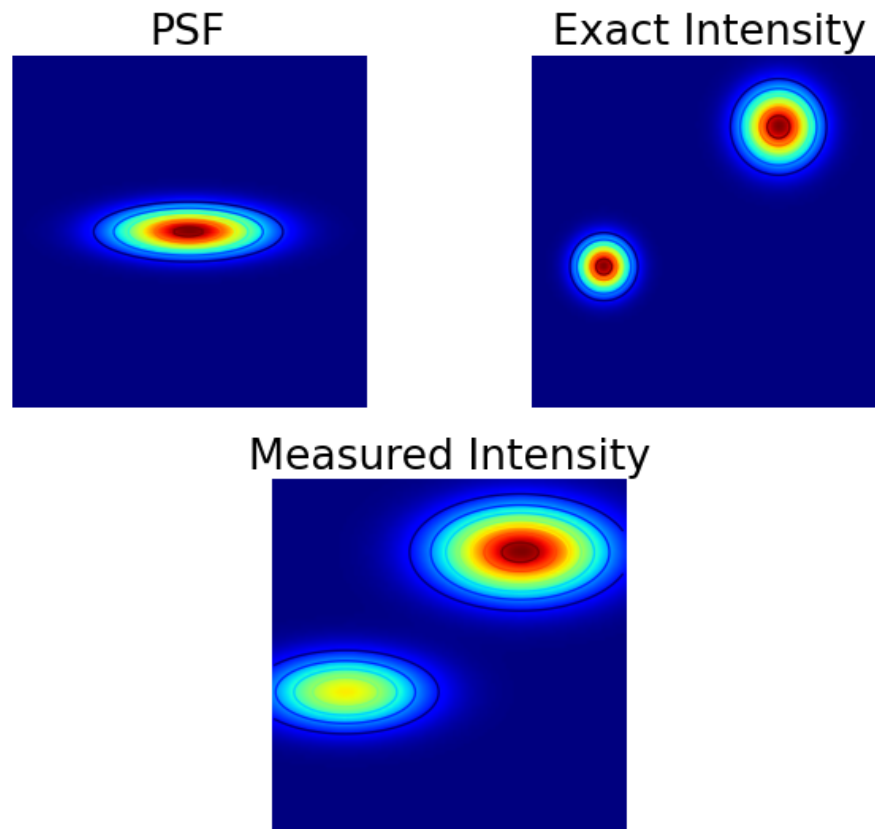


Figure 29: Example of picture for the PSF, an exact intensity and the intensity measured by the optics described by the PSF.

<sup>13</sup>It will do it with the full 3D effects.

## 4 Results

### 4.1 Comparison of the Intensity and Density Fluctuations

Usually in a tokamak, the optical axis is aligned with the magnetic field to have a better view of the fluctuations (the effect of the sampling volume is lower by doing this)[15], but in these results we have not done this<sup>14</sup>. The lens is positioned on the midplane and therefore has an angle of about  $5^\circ$  with the magnetic field[15]. In Figure 30, an example of fluctuation is shown (density and intensity of light collected by the BES diagnostic). The turbulence eddies in the BES picture are smoother and wider due to all the instrumental effects (specially the non-instantaneous emission and the sampling volume). The non-instantaneous emission extends the shape of the instabilities towards of the tokamak center (the beam comes from the exterior).

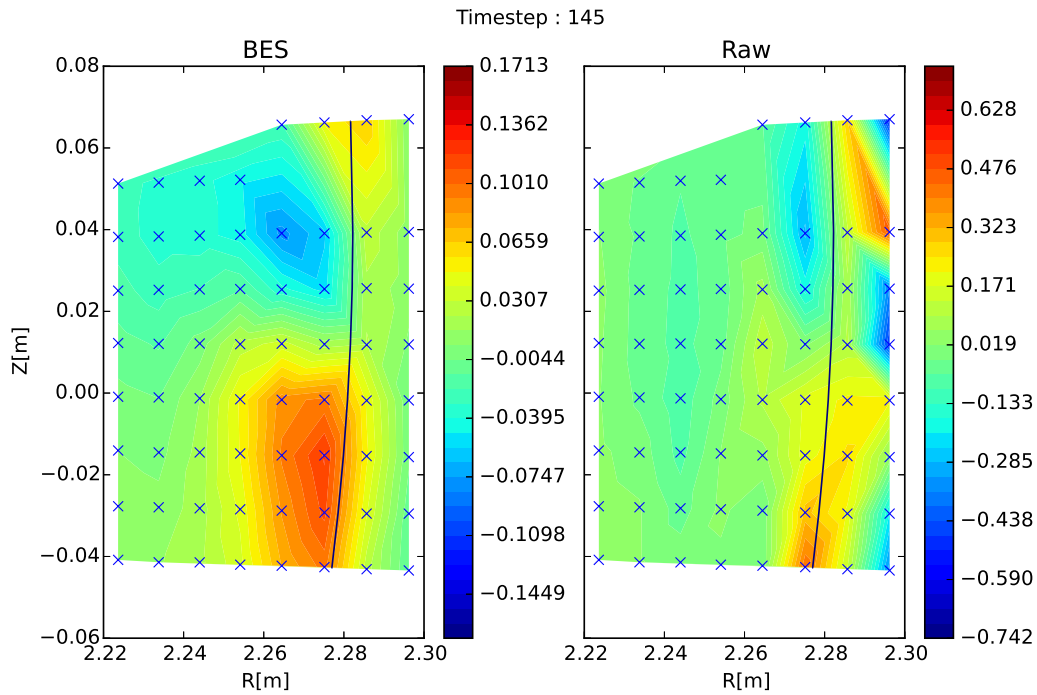


Figure 30: Example of fluctuations. On the left side, the fluctuations of the diagnostic are shown and on the right side, the raw density fluctuations from the XGC1 simulation. The blue crosses are the focus points and the black line is the separatrix.

### 4.2 Sensitivity Required for the BES Diagnostic

The sensitivity required by the diagnostic is often computed on the basis of atomic consideration and given in the form of the following scaling coefficient  $C$  in  $C \frac{\tilde{I}}{I} = \frac{\tilde{n}_e}{n_e}$ [12]. This coefficient is ill defined for the case without fluctuations (in density or in intensity). A toolbox has been created (see section 6.2) where it is possible to compute it for different points. The computation is realized with the equilibrium beam, the central line approximation for the optical integral and the expected fluctuations (in temperature and in density) as input. The choice of not taking the full sampling volume into account

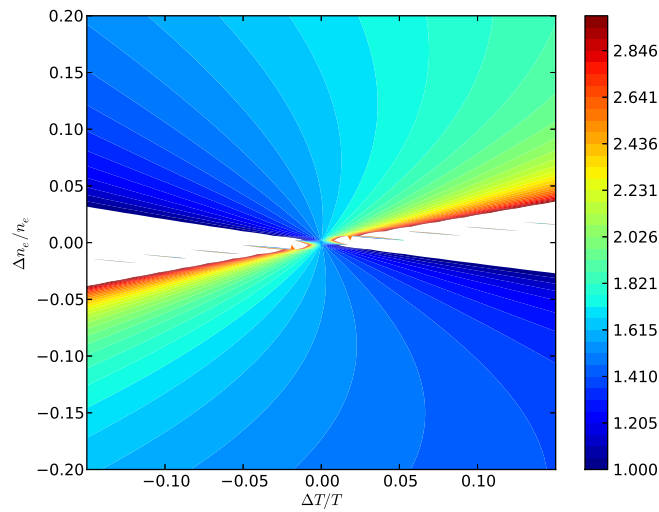
<sup>14</sup>To change it, only the variable `pos_lens` has to be changed.



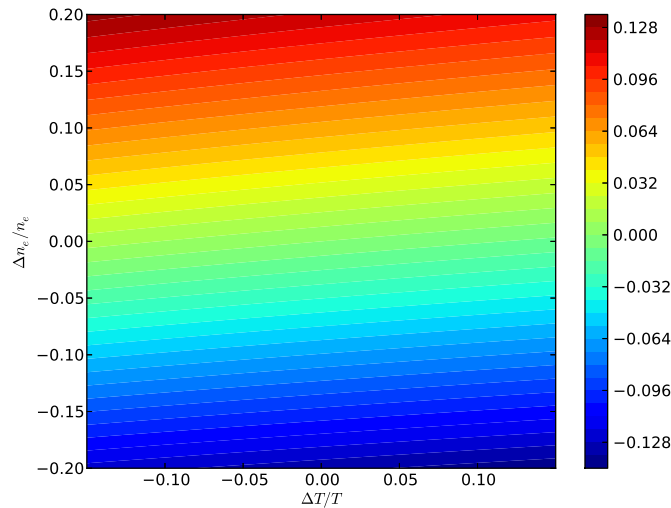
was made due to the small effect of it on this computation and the assumption on the shape of the turbulence. As we assumed for this computation that the ratio between the density fluctuation and the density is constant if the radius of the focus point is large, the sampling volume can give some turbulent eddies with a non reasonable size. The total intensity including fluctuation ( $I + \tilde{I}$ ) is given in (40) where  $\tilde{n}_e$  ( $\tilde{T}_e$ ) denotes the density (temperature) fluctuation and  $n_b$  the beam density (taken at the equilibrium). To compute it, we assumed that the ratio  $\frac{\tilde{n}_e}{n_e}$  (and we make the same assumption for the temperature) does not depend on the position (when the optics are aligned with the magnetic field, this approximation is close to reality).

$$I + \tilde{I} \propto \int (n_e + \tilde{n}_e) n_b \langle \sigma v \rangle |_{T_e + \tilde{T}_e, n_e + \tilde{n}_e} dz \quad (40)$$

Usually  $C$  is evaluated around 2–3 for a typical fluctuation range in the core (smaller than 5% in density and 1% in temperature)[5][12] but as seen in Figure 31 and 32 the scaling factor highly depends on the temperature fluctuation and, in Figure 33, on the position in the plasma of the focus point. The difference between the scaling factor given in [5][12] and figure 32 comes from their simplification that does not take into account the low density and high fluctuations of the pedestal (see figure 33). The core of the plasma has usually a higher value for the scaling factor than the pedestal and some smaller fluctuations in density, therefore a higher sensitivity is required. The discontinuities depend more on the density than the temperature fluctuation as seen in their equations (given in the caption of figures 31 and 32) and a significant difference is seen between the edge and the core. In the intensity fluctuation pictures (figures 31 and 32), one can see that the temperature dependency is more or less nonexistent (specially in the pedestal), validating the common assumption that  $C \frac{\tilde{I}}{I} = \frac{\tilde{n}_e}{n_e}$ , without a significant temperature fluctuation dependency.

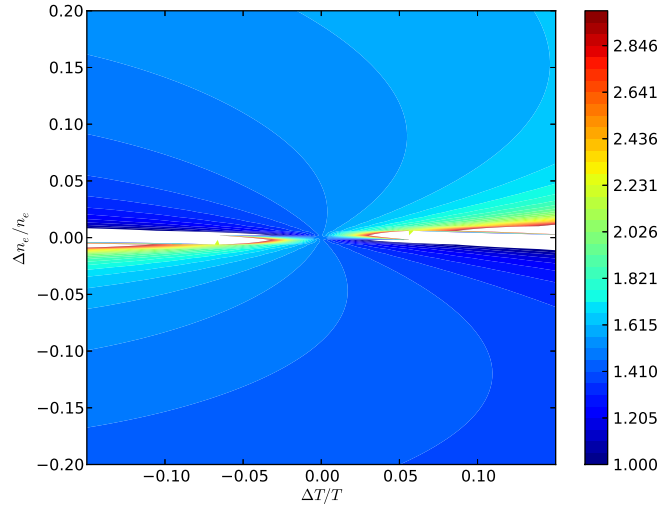


(a) Scaling factor: A discontinuity is present due to the ill-defined area of the scaling factor (where the intensity fluctuation is null). Its equation is given by:  $\tilde{n}_e/n_e = 0.113\tilde{T}_e/T_e$ . A threshold has been applied to the color map (1 for the lower boundary and 3 for the upper).

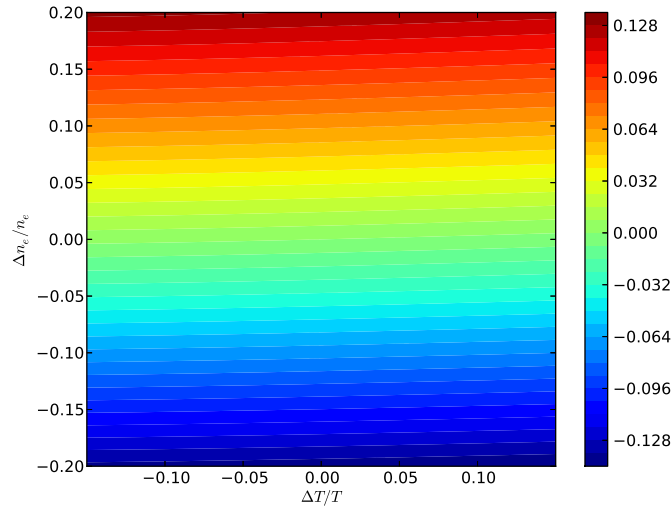


(b) Relative intensity fluctuation.

Figure 31: Scaling coefficient and ratio of the intensity fluctuations as a function of the density fluctuations and the temperature fluctuations for the core of the plasma ( $R = 1.67$  for DIII-D). The values of the electron density and temperature are taken from the equilibrium. In the core of the plasma, the scaling factor depends only weakly on the density fluctuation and the intensity fluctuation depends mainly on it.



(a) Scaling factor: A discontinuity is present due to the ill-defined area of the scaling factor (where the intensity fluctuation is null). Its equation is given by:  $\tilde{n}_e/n_e = 0.04\tilde{T}_e/T_e$ . A threshold has been applied to the color map (1 for the lower boundary and 3 for the upper).



(b) Relative intensity fluctuation.

Figure 32: Scaling factor and ratio of the intensity fluctuations as a function of the density fluctuations and the temperature fluctuations for the pedestal of the plasma ( $R = 2.25$  for DIII-D). The value of the electron density and temperature are taken from the equilibrium. In this case, both the scaling factor and the intensity fluctuation depend on the density fluctuation but only the scaling factor depends strongly on the temperature fluctuation. The scaling factor is weaker than in the core.

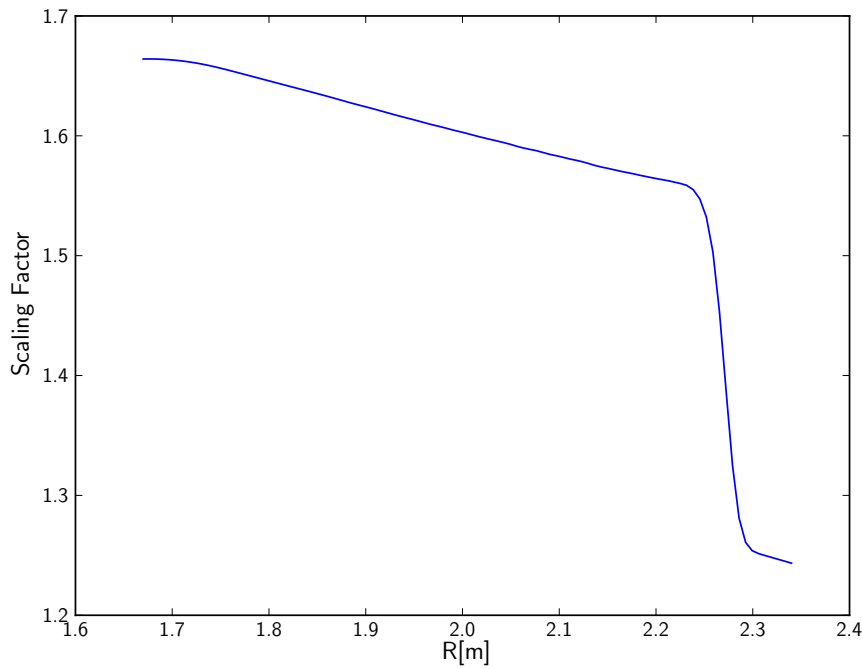
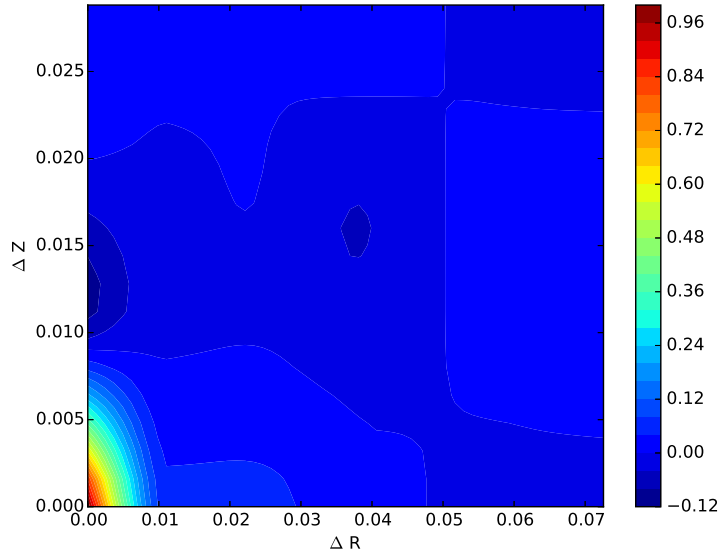


Figure 33: Scaling factor as a function of the radial coordinate (the points are situated on the mid plane). The ratio of the fluctuation is assumed to be constant along the radial axis and with the following ratio  $\tilde{n}_e/n_e = 0.1$  along with  $\tilde{T}/T = 0.01$ .

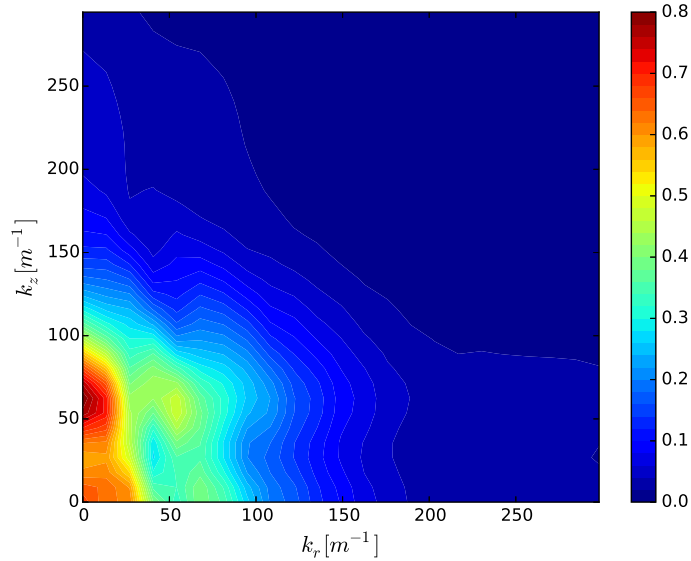
### 4.3 Spatial Correlation

The cross correlation is a common computation for characterizing the shape of the turbulent eddy and has been done in [7], [14], [11] and [12]. The computation is done with the function `correlate2d` in SciPy (using a convolution)[10].

In figure 34, the cross correlation shape is shown and the shape is close to the one obtained by G. McKee[12, Figure 4]. In the real space, the vertical direction has an oscillation and in the radial one, an exponential decay is observed. For the Fourier space, a small difference is present around the null wavelength, but it seems to come from the different types of condition (L-mode for [12] and H-mode for theses results)[18].



(a) Example of spatial correlation for the diagnostic . In the radial direction, a decay without oscillation can be seen and in the Z direction, an oscillation is present.



(b) Example of wave number spectrum of the spatial correlation (using an FFT). In the radial direction, the exponential decay is still present and, in the vertical direction, the peak is localized around  $k_z = 60[m^{-1}]$  and is much larger. It shows clearly that the radial direction is non-oscillatory and the vertical direction is oscillatory. A windowing has been applied on it.

Figure 34: Correlation done for  $\psi_n = 0.86$ ,  $R = 2.223[m]$  and  $Z = -0.014[m]$ .

Figure 35 shows the radial and vertical correlation length (at  $\Delta Z = 0$  for the radial one and  $\Delta R = 0$  for the vertical one). In the radial direction, a small flat area is present just before reaching a 0-correlation. From a few BES simulations with a less accurate resolution, this flat area seems to converge to a null value. In order to remove it, more

computation time is required. In [7, Figure 11-12], a similar computation is made and except for the flat area, the shape is the same. The difference of characteristic length comes from the difference between a L-mode and a H-mode and the area of the diagnostic (in the article, the position is deeper in the plasma)[18].

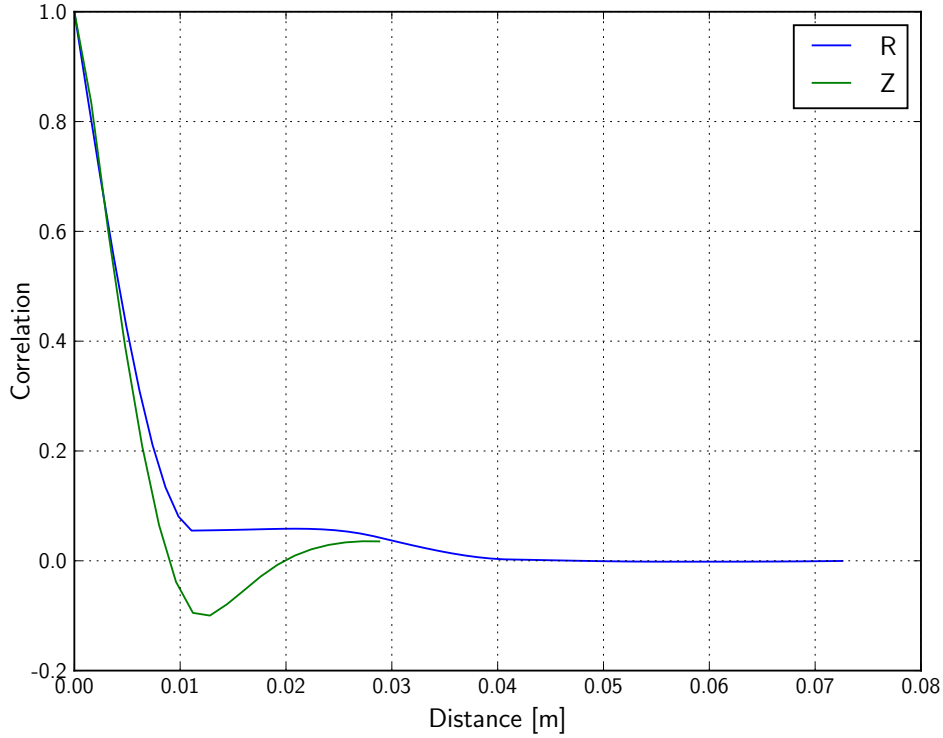


Figure 35: Correlation along the R and Z direction (the other one is set to 0) for the focus point situated at  $\psi_n = 0.86$ ,  $R = 2.223$  and  $Z = -0.014$ .

#### 4.4 Time Correlation

In figure 36, the time correlation is presented. The result looks like [14, Figure 4] with a characteristic time quite close to it. In this figure, it can be seen that the time step of the output of XGC1 ( $dt = 1.56 \cdot 10^{-6}$  [s]) is not good enough to have a good temporal resolution in the time correlation (only three steps are really related), therefore the time analysis is subject to a lot of noise.

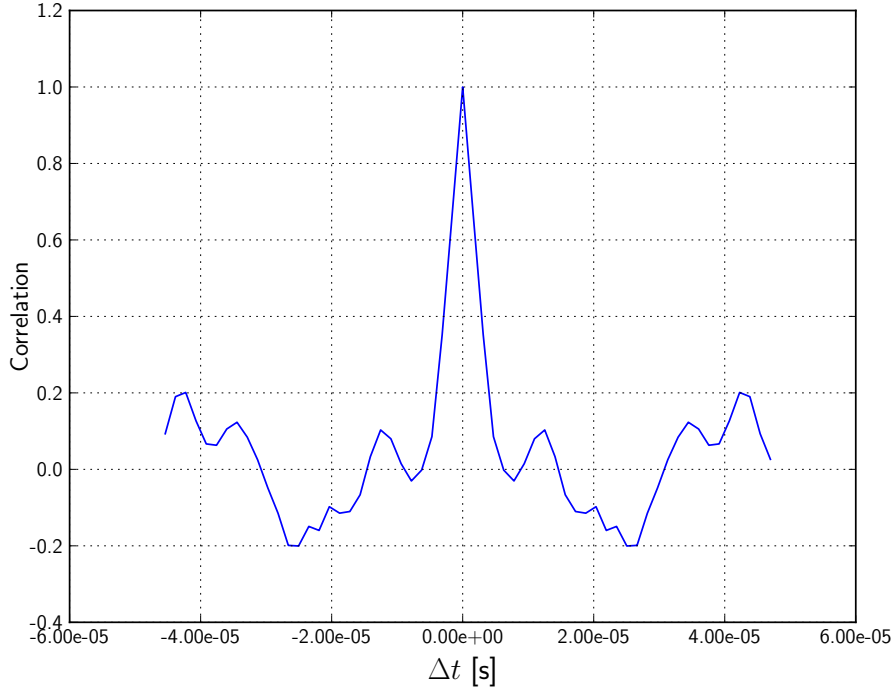


Figure 36: Time correlation for the focus point situated in  $\psi_n = 0.937$ ,  $R = 2.254[m]$  and  $Z = -0.0148[m]$

#### 4.5 Effect of the Different Elements

In this part, we analyze the effect of the different approximations possible in the code. The two references cases are the raw density fluctuations and the full synthetic diagnostic. The three approximations applied are the instantaneous emission (see section 3.1), the computation of the beam with the equilibrium data (see section 3.1) and the central line approximation for the optical integral (see section 3.2). All the approximation cases use the beam equilibrium approximation, or otherwise the full diagnostic (except for their approximation). In Table 3, the approximations are shown.

	Lifetime effect	Optics	Full optical integral	Beam eq.
Raw	<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>
BES	✓	✓	✓	✓
Central line ap.	✓	✓	<b>X</b>	<b>X</b>
Equilibrium beam	✓	✓	✓	<b>X</b>
Instantaneous	<b>X</b>	✓	✓	<b>X</b>

Table 3: Approximations made for the different cases. The label Optics is for if the optical axis integral is made or not.

The synthetic BES diagnostic does not have a significant effect on the time correlation (as seen on figure 37). In principle, the only cause of a change in the time correlation is a beam with a slower velocity than turbulent eddy motion and the sampling time effect from the measuring instrument, but in this synthetic diagnostic, we have supposed that the beam velocity is much larger than the blob velocity and we have not taking

into account the sampling time effects (e.g. photo-diodes). The instruments can be designed (and should be) in a way to minimize the sampling time effects, so that this approximation is close to the reality. The non-instantaneous emission is smoothing and increasing the size of turbulent eddies, therefore, the correlation can change a little bit in the area where two different kind of blob are meeting (a small time correlated blob with a long time correlated one).

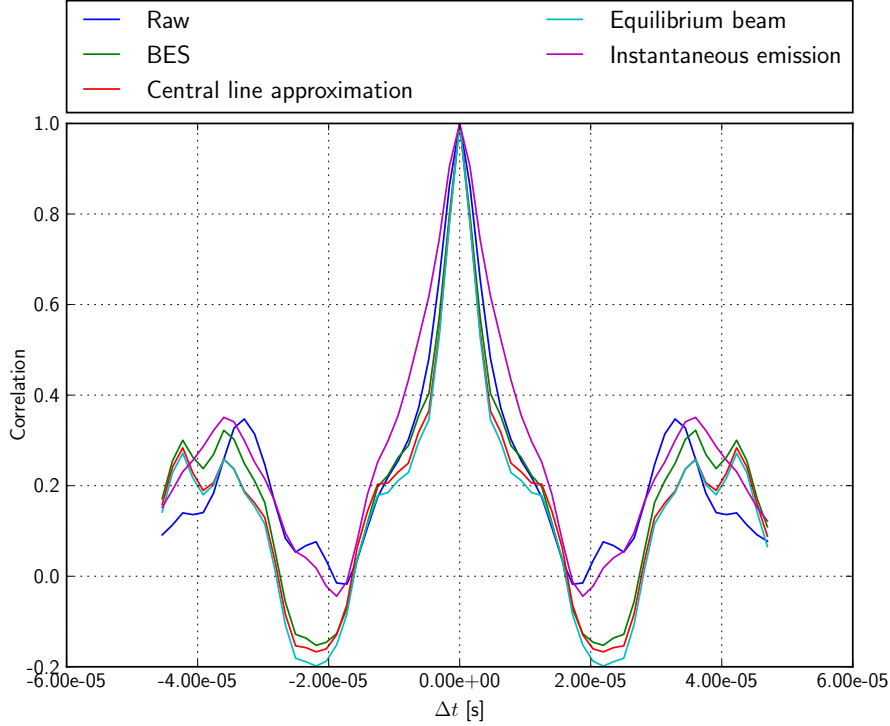


Figure 37: Comparison of the time correlation for the focus point situated in  $\psi_n = 0.937$ ,  $R = 2.254[m]$  and  $Z = -0.0148[m]$ . The time correlation is not really influenced by the diagnostic.

The vertical correlation of the different cases is given in Figure 38. Turbulent eddies are following the poloidal direction, therefore the vertical usually has no real meaning, but since the BES diagnostic is focused at the outboard midplane, the vertical is very close to the poloidal direction (only a very small shift for the lowest and highest fibers). As it is more easy to compute the vertical one, only this one is used. In the vertical direction, we still have a dependency on the kind of emission (instantaneous or not) which can seem strange (as the beam is in the perpendicular direction), but in fact it is not. The lifetime effect is shifting the turbulent eddies by about 3 [cm] inward, therefore the diagnostic sees a stronger turbulent area (and less subject to noise) with a different correlation length. Therefore, only the correlation of the diagnostic with the non-instantaneous emission is comparable to real experiment (and accurate enough). The optical integral along the optical axis [eq (30)] becomes important (difference between the instantaneous emission and the raw one) and is due to the turbulent eddies that follows the magnetic field lines (which have a vertical component with respect to the optical axis used). The integral takes some points before (and after) the focus plane that can still be in the blob even if the focus point is outside of it. Also, the equilibrium beam approximation is introducing a non negligible error at long distance ( $\Delta z \approx 0.06$  [cm]). The difference between the raw



density fluctuation and the BES one is close to the one in [7].

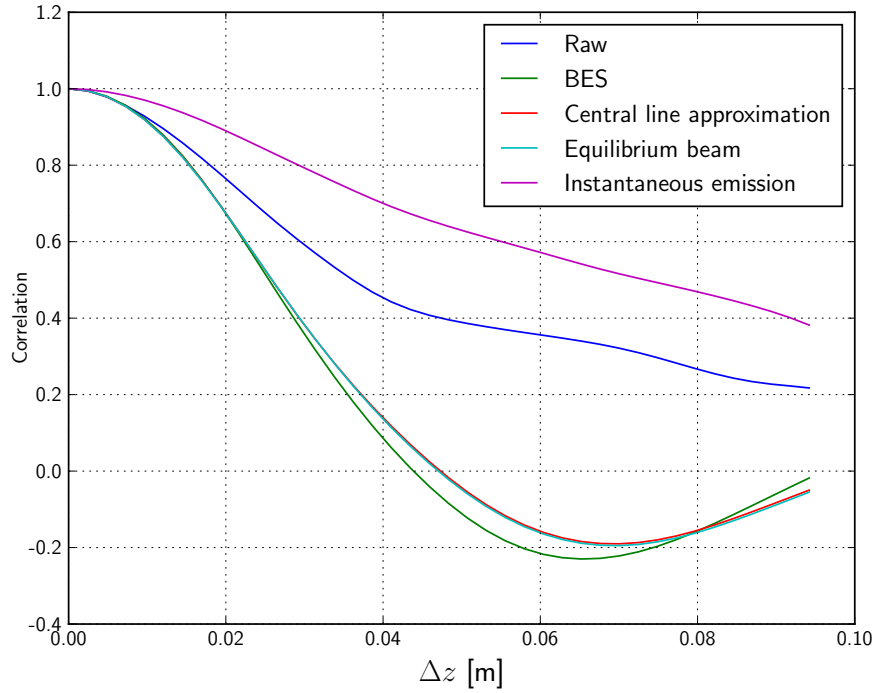


Figure 38: Comparison of the vertical correlation for  $\psi_n = 1.02$ ,  $R = 2.285[m]$  and  $Z = -0.043[m]$ . The difference with Figure 35 is due to the noise. The two most important effects are the sampling volume and the non-instantaneous emission.

The difference between the BES correlations in Figures 35, 38 and 39 is due to the method of computation. In the second figure, only 8 fibers are used but for the first Figure all of them are used (64), therefore the noise in the second figure is more important.

Figure 39 shows the radial correlation. One can make the same observation than for the vertical case, and see a small difference between the central line approximation and the full diagnostic at the outermost side of the diagnostic and between the equilibrium beam approximation and the full diagnostic at the innermost side of the diagnostic. For the lifetime effect, the difference in this direction is more important than before and it is quite logical due to the direction of the beam (it stays at the mid-plane). The effect of the optical integral along the optical axis is the same, but now it is due to the toroidal component of the magnetic field. As it can be seen in Figure 39, the central line approximation is not good in the pedestal region (positive  $\Delta R$ ) but seems to work well when the density equilibrium gradient is small. The opposite case is seen in the beam equilibrium approximation, that is good for small distance from the source (for example the first time that the beam is in the pedestal) and becomes worse when the distance becomes large (for example in the core). The difference between the radial correlation length using the raw density fluctuation and the BES one is close to the one in [7].

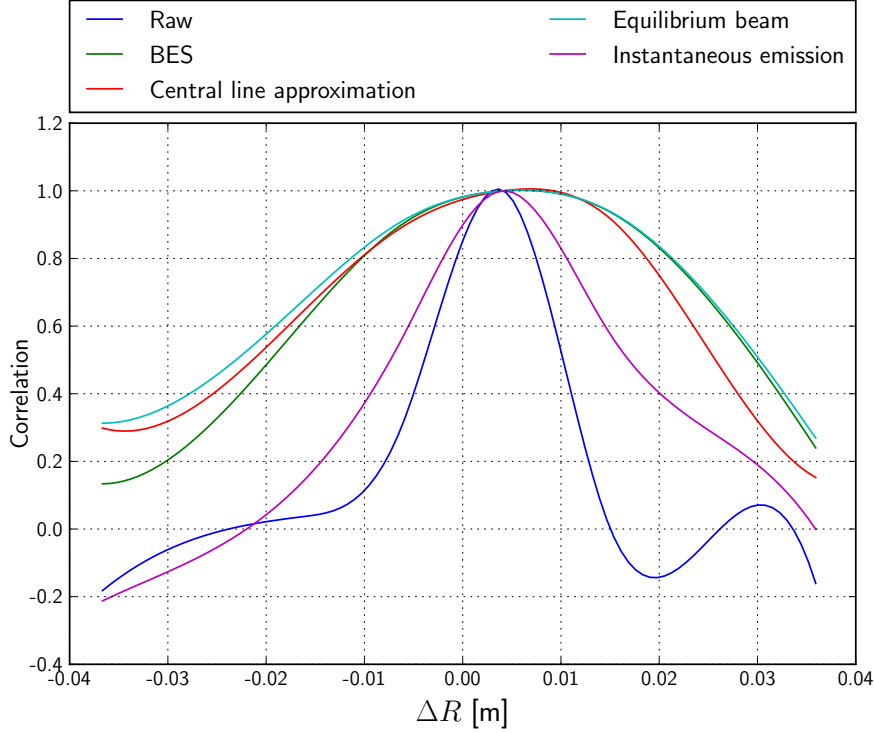


Figure 39: Comparison of the radial correlation for  $\psi_n = 0.958$ ,  $R = 2.264[m]$  and  $Z = 0.012[m]$ . The difference with Figure 35 is due to the noise. In this case, everything has an effect, but the most important are the optical integral along the optical axis and the finite lifetime. For the central line approximation/the equilibrium beam approximation, only outermost/innermost (positive/negative  $\Delta R$ ) side of the plasma is affected.

In order to compute the vertical correlation length, an exponential decay is assumed ( $C(\Delta z) = \exp(-\Delta z/\xi)$  where  $\xi$  is the correlation length). In a first part, the computation takes only the part of the array that is decreasing monotonically and, then, two different cases are possible. The first one is when the smallest value is smaller than  $\exp(-1)$  and the length is given by the distance such that  $C(\Delta z) = \exp(-1)$ . The second one does a fitting on the monotonic part of the data and uses the fitted value of  $\xi$ . The result is given in Figure 40 and two points are not well computed (the peaks of the raw case and the instantaneous case). This is due to the insufficient assumption of monotonic data as the correlation is monotonic but an oscillation is preventing the data to decrease exponentially. Thus the correlation becomes too large. As usual, the most important effect on the diagnostic is the kind of emission, the instantaneous emission is close to the raw density fluctuation and the non-instantaneous is close to the full computation. The effect of shifting due to the lifetime is clearly seen in Figure 40. The lowest part of the non-instantaneous emission is shifted of about 3 cm in comparison to the instantaneous one. This distance is the one given by  $v_b\tau$  where  $v_b$  is the beam velocity (of the main component) and  $\tau$  is the lifetime.

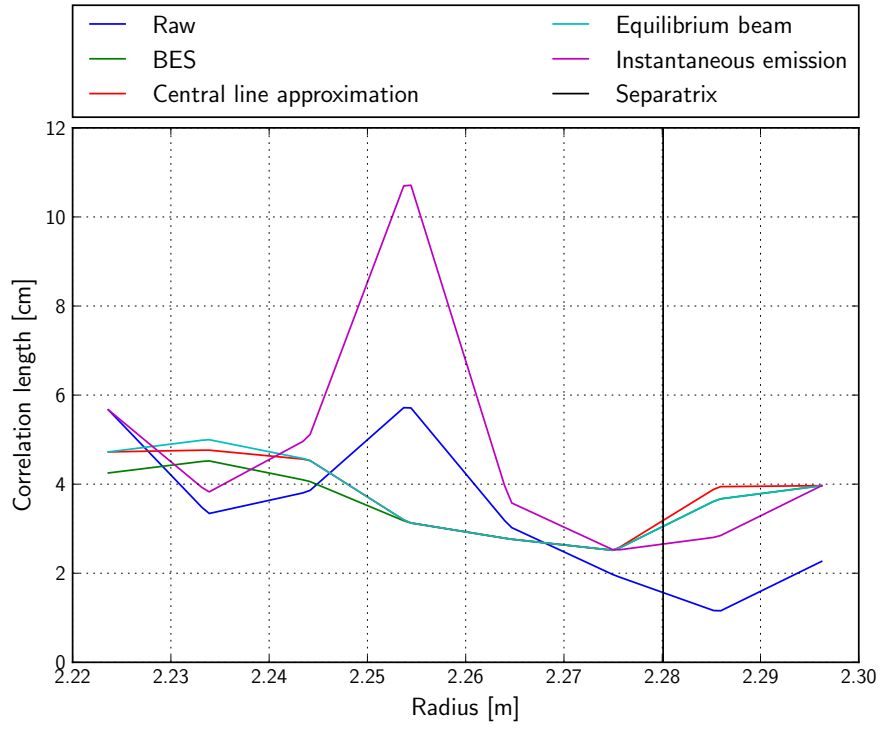


Figure 40: Comparison of the radial dependency on the vertical correlation for  $Z = -0.042[m]$ . The computation assumes an exponential decay. Two peaks around  $R = 2.255[m]$  are not well computed due to this assumption. The real value is smaller. The shift due to the lifetime can be seen in the minimum of each curve. the non-instantaneous cases are shifted of about  $3[cm]$  ( $\approx v_b\tau$ ) from the instantaneous cases.

## 5 Conclusion

A synthetic BES diagnostic code has been developed to be used with turbulence codes, namely XGC1. It includes some new features such as the full 3D computation, the choice of the approximation and the computation of the scaling factor<sup>15</sup>. The full 3D computation is very important when the optics are not perfectly aligned with the magnetic field and when two different turbulent eddies (positive and negative density fluctuation) are following each other, therefore it should be taken into account in order to have an accurate synthetic diagnostic. The different approximations (instantaneous emission, equilibrium beam and central line approximation) shows the difference of importance of these computations. The non-instantaneous emission should always be present when trying to reproduce an experiment. The equilibrium beam is a good approximation close to the source (outside the separatrix) of the beam and at short correlation length, and the central line approximation is good in the area where the equilibrium density has a small gradient (e.g. in the core).

Due to some oscillation induced by the higher order methods for the integrals, the 1.0 version of the code<sup>16</sup> has a low order of convergence. However the order can be increased with the slope limiter, . The choice of the quadrature method is hard coded with a string, therefore it is easy to change it to any other order and, the version 2.0 is aims at having the order as an input of the code.

A number of improvements are still planned:

- an MPI Version,
- a blob detection algorithm,
- using a 2D beam or a code already existing,
- allowing multiple beams,
- using a slope limiter for the integrals in order to remove the oscillations,
- a higher order toroidal interpolation (using more planes),
- a higher order radial and poloidal interpolation in order to have a smoother convergence for the integral and for increasing their order,
- an adaptive integration to increase the number of step in the pedestal,
- computing a point spread function that can be used in the experiment,
- using only the useful nodes by taking into account the next node information,
- giving the possibility of choosing the accuracy of the full sampling integral,
- computing the blob velocity,
- loading other simulation results than XGC1,
- adding a GUI to facilitate its use,
- more accurate optics modeling.

<sup>15</sup>in the toolbox.

<sup>16</sup>version written for this report.

## 6 Annexes

### 6.1 Starting Objectives

- Understanding the XGC1 code
  - Understanding the general characteristics,
  - Understanding the grid, the fluctuation in density and the data outputs.
- Understanding the synthetic BES IDL code
  - Generalize the code for different reactors,
  - Modify the IDL code for using the XGC1 outputs,
  - (Optional) Translating the code in python,
  - Extend to include further physics if needed.
- Interpolation of  $\tilde{n}_e$  along a field line.
- Apply synthetic BES to previous XGC1 simulations and compare to experiment as possible.
- (Optional) Compare specific DIII-D simulations with edge BES coverage.

## 6.2 Description of the Code

The program versions used for this thesis are: SciPy 0.14.0, Numpy 1.8.1, h5py 2.0.1, python 2.7.3, dot 2.38 (for the graph) and memory\_profiler 0.32 (this one is not used for running the diagnostic). The code was commented as much as possible, but explaining the idea behind the code will probably give the reader a better understanding of it. Let us start with the most basic classes and finish with the main class. A documentation has been done with sphinx (version 1.2.3) and sphinxcontrib-tikz 0.4.1 in HTML (available on the website of the PPPL).

Figure 41 shows the relation between all the classes.

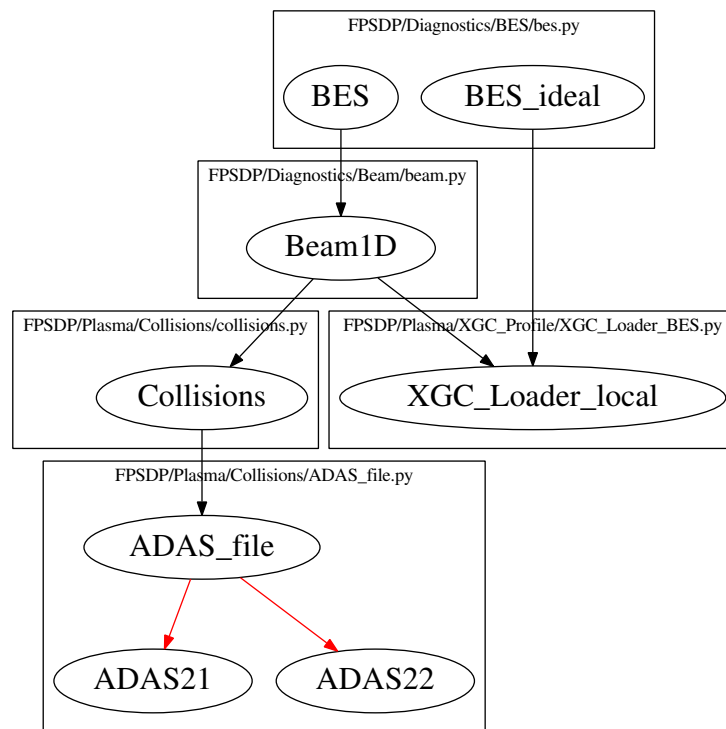


Figure 41: Diagram of the connections between the different classes. The black lines represents attributes, the red ones inheritances, the ellipses classes and the boxes files.

### ADAS

The first classes are the ADAS readers (that all inherit from an abstract class [**class** `ADAS_file`]). They are created to read the ADAS database files and to simplify the access to them. The code reads only two kind of ADAS files, the 21 (**class** `ADAS21`) and the 22 (**class** `ADAS22`). The first one consists of the beam stopping coefficient and is used to compute the density of the beam, the second one is for the emission coefficient and is used when computing the emission in a specific place.

### Collisions

The **class** `collisions` is used to interpolate the data from ADAS and computing a few quantities that depends on the collisions (for example the lifetime of the excited state or the wavelength of the emitted photons).

## XGC1 Loader

The **class** `XGC1 Loader local` reads the data from XGC1, stores them, and interpolates them with a field line interpolation.

## Beam

The **class** `Beam1D` simulates the beam using the equilibrium data. It has the data from XGC1 as an attribute (under the shape of an instance of `XGC1 Loader local`). The emission is computed in this class too.

## BES

The **class** `BES` uses `Beam1D` to compute the light received by the optical system. It is the highest level of the problem. It reads the data from a configuration file and uses the method `get_bes()` to compute the diagnostics.

## Execution of the Code

Two methods from **class** `BES` are called when using the diagnostic: the initialization (see Figure 42) and `get_bes` (see Figure 43). In the figures, the function `Integration.integration_points` gives the quadrature formula. The initialization consists in creating the beam (`Beam1D.__init__` reads the collisions files and the beam's values in the configuration file), then in computing the limits (`BES.compute_limits` computes the limits by taking into account all the effect except the field line interpolation). The initialization of the loader class (`XGC Loader local.__init__` loads the equilibrium data and the first time step) is done afterward, then the setup of the simulation output in the beam class (`Beam1D.set_data` copies the instance of `XGC Loader local` and the creation of the beam mesh by using the limits given by `BES.compute_limits` and computes the density for the first time step) are made. At the end the loading of the filter (`BES.load_filter` reads the filter file and interpolate it) and the computation of the emission's wavelength (`Collisions.get_wavelength` computes the wavelength by assuming an hydrogen atom) are done.

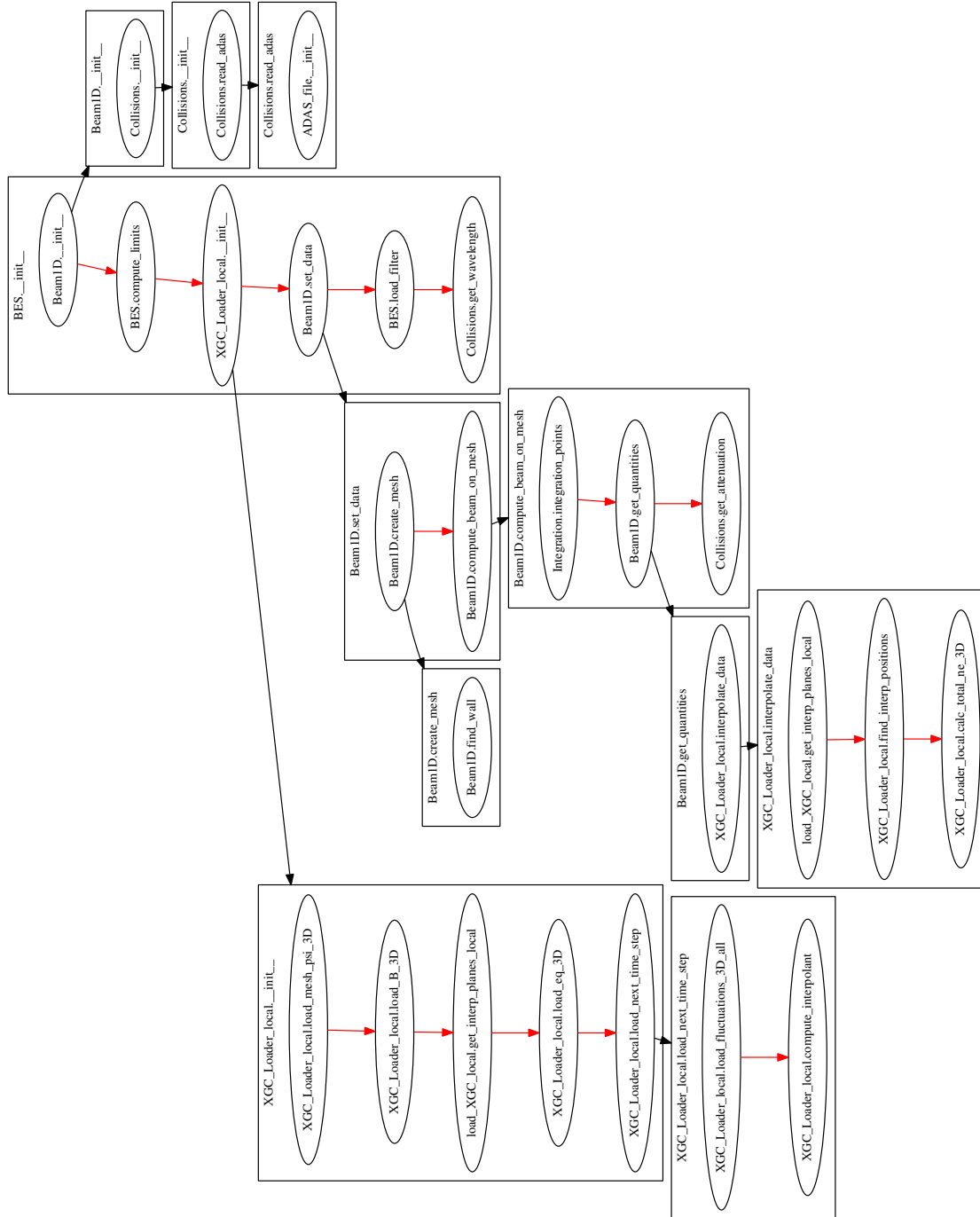


Figure 42: Methods called for the initialization of the diagnostic. The red lines are for successive calls and the black ones to go inside the function.

The second method consists in a loop on each time step and each fiber. The code loads the next time step (`XGC_Loader_local.load_next_time_step` loads and interpolates the data from XGC1), computes the beam (if the beam is computed with the fluctuation, `BeamID.compute_beam_on_mesh`) and the intensity of each fiber (`BES.intensity`). The method `BES.intensity` computes the integral along the optical axis, calls the method `BES.light_from_plane` that computes the integral on the perpendicular plane by taking into account the filter (thus the Doppler effect too) and uses the functions `BES.get_emis_from` (to compute the emission (instantaneous or not)) and `BES.`



`get_solid_angle` (to compute the solid angle if needed).

## Toolbox

A toolbox has been created (file `FPSDP/Diagnostics/BES/tools.py`). It consists of a class that reads the output of the diagnostic and with a few methods creates all the figures given in the results section 4 or making some video clips. After the declaration of this class, some functions draw all the other figures given in this thesis (such as the convergence). The only three methods that will be described are `put_two_files_together`, `check_geometry` and `compute_beam_config`. The first one can be used when splitting a simulation in different time intervals (e.g making a first simulation between the time steps 0 and 100 and a second one between 101 and 180). The method takes the two output files and merges them in a single file.

The second one uses the initialization of the diagnostic to plot the beam, focus points, limits and the lens as understood by the code. It can be used to check if the fibers are on the beam or if everything is well computed.

The last one helps to create the input file. It computes the direction of the beam (from the position of the source) to have it tangent to the circle with a given radius and can give the angles such that the fibers (described with their radial position) are exactly on the beam.

## 6.3 Configuration File

```
# For double input, put .0 for assuring that the data is
    read as
# a float

[Beam geometry]
# position of the beam (X,Y,Z) (in meter)
# Z is for the height
position = [-2.36511538, -1.3655, 0.0]
# direction of the beam (X,Y,Z), it will be normalized in
    the code
direction = [0.96910869, 0.24663402, 0.0]
# beam width (FWHM in meter)
# horizontal
beam_width_h = 0.13
# vertical
beam_width_v = 0.27
# number of points for the discretization along the beam
    direction
# the ending point is computed at the edge of the mesh
# (the beam line goes from 'position' until the farrest
    point used
# for all the computation)
Nz = 100

[Beam energy]
# total power of the beam (in W)
```

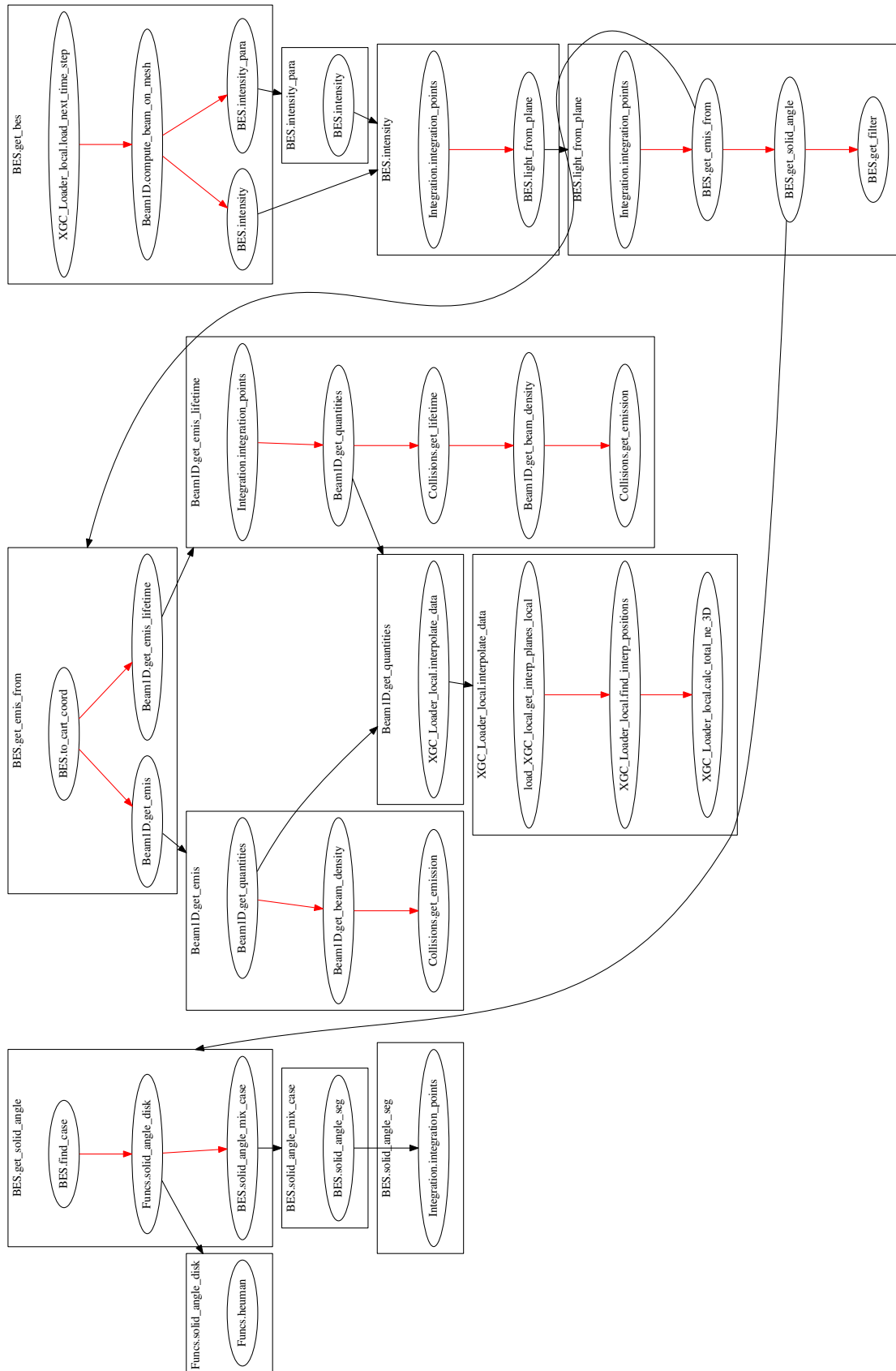


Figure 43: Methods called for the evaluation of the intensity collected by the fibers. The red lines are for successive calls and the black for going inside the function.

```

power = 1.2e6
# fraction of power between the different beam components (
  in percent)
f = [77,17.0,6.0]
# Energy of the different components (keV)
E = [75.0,37.5,25.0]
# mass of each components of the beam (in amu)
mass_b = [1.0,2.0,3.0]

[Collisions]
# ADAS files (separated by ,)
# attenuation
adas_atte = ["FPSDP/Plasma/Collisions/bms10#h_h1.dat"]
# emission
adas_emis = ["FPSDP/Plasma/Collisions/bme10#h_h1.dat"]
# collisions considered (index start to 0)
# first index is for a file (in adas_file) and second is
# for the beam component
# not checked in the code, but one beam component should
  have
# only one file (usually only one file is used for
# all beam components)
coll_atte = [[0,0], [0,1], [0,2]]
coll_emis = [[0,0], [0,1], [0,2]]

# quantum number of the low energy state and the excited
  one
# the code assume an hydrogen atom
n_low = 2
n_high = 3
# number of points (=> intervals) for the integration of
  the lifetime effect
# Used in Beam1d.get_emis_lifetime()
Nlt = 50
# t_max*lifetime = upper value of the integrale for the
  lifetime effect
# cutoff value
# if t_max = 0, the lifetime will not be
# used in Beam1d.get_emis_lifetime()
t_max = 8.0
# Lifetime of the excited particles
# (used for computing the limits of the mesh)
tau = 8e-9
# true for using the equilibrium for the attenuation
# and false for using the equilibrium + fluctuation
# Using the fluctuation increase the size of the mesh
# used for the interpolation, therefore all the
  interpolaion
# on the data will be a little bit longer
eq_atte = true

```

```

[Optics]
# position of the lens (X,Y,Z) in meter
pos_lens = [-0.81236,-2.58433,0]
# position of the focus point for each fiber (R,Z,phi)
# Z is assumed to be an array of the good size,
# R and phi can be a float or an array
# For the cartesian X = R*cos(phi) and Y = R*sin(phi)

R = [2.22364, 2.23378, 2.24402, 2.25411, 2.26448, 2.27514,
      2.28568, 2.29622, 2.22364, 2.23378, 2.24402, 2.25411,
      2.26448, 2.27514, 2.28568, 2.29622, 2.22364, 2.23378,
      2.24402, 2.25411, 2.26448, 2.27514, 2.28568, 2.29622,
      2.22364, 2.23378, 2.24402, 2.25411, 2.26448, 2.27514,
      2.28568, 2.29622, 2.22364, 2.23378, 2.24402, 2.25411,
      2.26448, 2.27514, 2.28568, 2.29622, 2.22364, 2.23378,
      2.24402, 2.25411, 2.26448, 2.27514, 2.28568, 2.29622,
      2.22364, 2.23378, 2.24402, 2.25411, 2.26448, 2.27514,
      2.28568, 2.29622, 2.22364, 2.23378, 2.24402, 2.25411,
      2.26448, 2.27514, 2.28568, 2.29622]

Z = [-0.0140560, -0.0145320, -0.0145320, -0.0147840,
      -0.0152600, -0.0152600,
      -0.0154120, -0.0156640, -0.000916000, -0.00106800,
      -0.00129200, -0.00151600,
      -0.00166800, -0.00166800, -0.00182000, -0.00179200,
      -0.0408400, -0.0414160,
      -0.0414440, -0.0420200, -0.0422720, -0.0425960,
      -0.0431000, -0.0434240,
      -0.0277000, -0.0279520, -0.0282040, -0.0284560,
      -0.0287800, -0.0292560,
      -0.0295080, -0.0295080, 0.0122520, 0.0120280,
      0.0121280, 0.0119760,
      0.0120760, 0.0119520, 0.0118000, 0.0119000,
      0.0250680, 0.0252400,
      0.0253680, 0.0254680, 0.0253440, 0.0254720,
      0.0256720, 0.0255480,
      0.0382080, 0.0383360, 0.0385360, 0.0387360,
      0.0391160, 0.0390920,
      0.0392920, 0.0394200, 0.0512760, 0.0515040,
      0.0519560, 0.0521840,
      0.0656800, 0.0662320, 0.0667840, 0.0670400]

phi = -2.55726044

# index of the simulation toroidal plane
# will be align to the focus points
plane = 0

# radius of the lens
rad_lens = 0.2

```

```
# radius of the focus points (size sampling plane at the
# focus point)
# can be an array of the size of Z or a float
rad_foc = 0.01

# cut off for the integral along the sightline
# [pos_foc-int*w, pos_foc+int*w] where w is the
# width (standard deviation) of the beam in the direction
# of the optics
# (more detail in the code)
# we are assuming that the focus point are
# on the central line of the beam
int = 2.5
# number of points for the integration along the
# sightline
Nint = 100
# number of section for the integral of the solid angle
Nsolid = 80
# number of interval for the radial integration of the
# solid angle
Nr = 10

# type of integration for the emission
# 1D for taking into account only the emission
# of the central line or 2D for doing an integration
# over the plane perpendicular to the central line
type_int = 2D

[Data]
# name of the filter file
filter = FPSDP/Diagnostics/BES/filter.dat
# path to the simulation data
data_path = /global/project/projectdirs/m499/jlang/
particle_pinch/
# method of the interpolation (linear or cubic)
interpolation = cubic
# torroidal step for the field line interpolation (in
# radians)
dphi = 0.01
# simulation time steps wanted
timestart = 1
timeend = 182
timestep = 1
# Limits for the loader (with a low toroidal resolution, it
# should be false)
# read doc about the XGC1 loader for more information
limits = false
```

## 6.4 Acknowledgments

First, thank you Michael Churchill for your help and patience through out all this thesis. Your were always available to help me understand the problems that I met, to check my code in order to make it the most readable and logical possible and correcting my report. It was a pleasure to meet you.

A special thanks to Stephan Brunner who accepted the project and helped me through out my master. Thank you for spending a lot of time teaching me plasma physics, helping me with the administration for the project, correcting my reports, etc. It was an honor to work with you and I hope to do it again.

Thank you Choong-Seock Chang for proposing the project and accepting me.

I want to thank you Seung-Hoe Ku for his help with XGC1 and Ahmed Diallo for his explanations about the experimental part of the BES diagnostic and his friendship.

The last person that I wish to thank is Christèle Zbinden for her help in correcting this thesis.

# Bibliography

- [1] M. Abramowitz and I. A. Stegun. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications, New York, 0009-revised edition edition, June 1965.
- [2] M. F. Adams, S.-H. Ku, P. Worley, E. D’Azevedo, J. C. Cummings, and C.-S. Chang. Scaling to 150k cores: Recent algorithm and performance engineering developments enabling XGC1 to run at scale. *Journal of Physics: Conference Series*, 180(1):012036, July 2009.
- [3] J. C. Butcher. On fifth and sixth order explicit Runge-Kutta methods: order conditions and order barriers. *The Canadian Applied Mathematics Quarterly*, 17(3), 2009.
- [4] A. R. Field, D. Dunai, N. J. Conway, S. Zoletnik, and J. Sárközi. Beam emission spectroscopy for density turbulence measurements on the MAST spherical tokamak. *The Review of Scientific Instruments*, 80(7):073503, July 2009.
- [5] R. J. Fonck, P. A. Duperrex, and S. F. Paul. Plasma fluctuation measurements in tokamaks using beam-plasma interactions. *Review of Scientific Instruments*, 61(11):3487–3495, Nov. 1990.
- [6] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*, volume 31 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin/Heidelberg, 2006.
- [7] C. Holland, A. E. White, G. R. McKee, M. W. Shafer, J. Candy, R. E. Waltz, L. Schmitz, and G. R. Tynan. Implementation and application of two synthetic diagnostics for validating simulations of core tokamak turbulence. *Physics of Plasmas (1994-present)*, 16(5):052301, May 2009.
- [8] I. H. Hutchinson. Excited-state populations in neutral beam emission. *Plasma Physics and Controlled Fusion*, 44(1):71, Jan. 2002.
- [9] K. Ikeda, M. Osakabe, A. Whiteford, K. Ida, D. Kato, S. Morita, K. Nagaoka, Y. Takeiri, K. Tsumori, M. Yokoyama, M. Yoshinuma, and L. Experiment. *J. Plasma Fusion Res. SERIES, Vol. 8 (2009) Optical Observation of Neutral Beam Attenuation in Hydrogen Discharge at LHD*. 2008.
- [10] E. Jones, T. Oliphant, P. Peterson, and others. SciPy: Open source scientific tools for Python, 2001.

- [11] G. McKee, R. Ashley, R. Durst, R. Fonck, M. Jakubowski, K. Tritz, K. Burrell, C. Greenfield, and J. Robinson. The beam emission spectroscopy diagnostic on the DIII-D tokamak. 1999.
- [12] G. R. McKee, C. Fenzi, R. J. Fonck, and M. Jakubowski. Turbulence imaging and applications using beam emission spectroscopy on DIII-D (invited). *Review of Scientific Instruments*, 74(3):2014–2019, Mar. 2003.
- [13] F. Paxton. Solid Angle Calculation for a Circular Disk. *Review of Scientific Instruments*, 30(4):254–258, Apr. 1959.
- [14] M. W. Shafer, R. J. Fonck, G. R. McKee, C. Holland, A. E. White, and D. J. Schlossberg. 2d properties of core turbulence on DIII-D and comparison to gyrokinetic simulations. *Physics of Plasmas (1994-present)*, 19(3):032504, Mar. 2012.
- [15] M. W. Shafer, R. J. Fonck, G. R. McKee, and D. J. Schlossberg. Spatial transfer function for the beam emission spectroscopy diagnostic on DIII-D. *Review of Scientific Instruments*, 77(10):10F110, Oct. 2006.
- [16] H. P. Summers. The ADAS User Manual, version 2.6, 2004.
- [17] T. Vernay. *Collisions in global gyrokinetic simulations of tokamak plasmas using the delta-f particle-in-cell approach: Neoclassical physics and turbulent transport*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2013.
- [18] F. Wagner. A quarter-century of H-mode studies. *Plasma Physics and Controlled Fusion*, 49(12B):B1, Dec. 2007.