# Simulating Collisions with GORILLA

J. Schatzlmayr[1], C.G. Albert[1], S.V. Kasilov[1,2,3]

[1] Fusion@ÖAW, Institut für Theoretische Physik - Computational Physics, Technische Universität Graz, Petersgasse 16, 8010 Graz, Austria
[2] Institute of Plasma Physics, National Science Center "Kharkov Institute of Physics and Technology", 61108 Kharkiv, Ukraine
[3] Department of Applied Physics and Plasma Physics, V. N. Karazin Kharkiv National University, 61022 Kharkiv, Ukraine

## Introduction

One of the goals of the TSVV6 work package of Eurofusion is to investigate the effects of impurity atoms (mainly emanating from the reactor wall) onto the edge plasma. This poster presents an important step towards this goal by discussing the introduction of a collision operator into the guiding center tracing code GORILLA (see below). First results are shown on the distribution of tungsten ions.

## GORILLA

The name **GORILLA**[1][2] is an acronym for

**G**uiding center

**OR**bit

**I**ntegrator with

**L**ocal

**L**inearisation

**A**pproach.

It is a Fortran based code used to trace particles within fusion reactors. Its outstanding feature is that it divides the whole computation domain into small tetrahedral cells and within each such cell the electric field, the magnetic field and other quantities are approximated by piecewise linear functions.

This makes the guiding center **equations of motion linear** and they can be **solved exactly** with a computational speed up to **ten times faster** than conventional Runge Kutta methods.

Due to its high computational efficiency and the nature of its setup, the code lends itself ideally for calculating arbitrary moments of the distribution function.

## Theory on Collisions

When modelling collisions, we are actually dealing with diffusion problems. In 1D, the diffusion equation reads

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(\underbrace{fV - D\frac{\partial f}{\partial x}}_{fV_{\text{eff}}}) = 0, \tag{1}$$

with $f$ the distribution function and $D$ the diffusion coefficient.

Lifting Eq. 1 to 6D phase space, transforming to curvilinear coordinates and then only concentrating on velocity space (which produces the dominant effects), the diffusion matrix ends up being diagonal. In terms of the momentum $p$ and the pitch parameter $\lambda = v_\parallel/v$, the diffusion equation then reads

$$\frac{\partial f}{\partial t} = \frac{1}{p^2}\frac{\partial}{\partial p}\left(p^2(D^{pp}\frac{\partial f}{\partial p} - V^p f)\right) + \frac{\partial}{\partial \lambda}\left(D^{\lambda\lambda}\frac{\partial f}{\partial \lambda}\right). \tag{2}$$

## Numerical Implementation

Eq. 1 can be cast into the form of the Fokker Planck equation

$$\frac{\partial f}{\partial t} + \frac{\partial}{\partial x}(Af - \frac{\partial}{\partial x}(Bf)) = 0. \tag{3}$$

The fundamental solution to this equation is

$$G(x, x_0, t) = \frac{1}{\sqrt{4\pi Bt}}\exp\left(-\frac{(x - x_0 - At)^2}{4Bt}\right). \tag{4}$$

Numerically, this can be approximated by tracing test particles whose trajectory evolves according to

$$\begin{aligned} x(t + \Delta t) =& x(t) + \Delta x(x(t)), \\ \Delta x =& \sqrt{2\Delta t B}\xi + A\Delta t, \end{aligned} \tag{5}$$

where $\xi$ are random variables with mean $\overline{\xi} = 0$ and standard deviation $\overline{\xi^2} = 1$.

Eq. 2 can be similarly manipulated leading to

$$\begin{aligned} \Delta\lambda =& \sqrt{2\Delta t D^{\lambda\lambda}}\xi_\lambda + V^\lambda_{\text{coll}}\Delta t, \\ \Delta p =& \sqrt{2\Delta t D^{pp}}\xi_p + V^p_{\text{coll}}\Delta t, \end{aligned} \tag{6}$$

where $D^{\lambda\lambda}$, $V^\lambda_{\text{coll}}$, $D^{pp}$ and $V^p_{\text{coll}}$ are now dependent on the particles phase space position. They are evaluated according to [3].

## Implementation in GORILLA

When tracing a particle with collisions in GORILLA, three steps are executed in a loop:

- The local collision frequency is computed to evaluate $\Delta t$ from Eq. 6
- $\lambda$ and $p$ are updated according to Eq. 6
- The particles trajectory is followed for the time $\Delta t$.

The collision module consists of five subroutines:

- `collis_init` is called before the particle tracing begins. All the non phase space dependent parts of $D^{\lambda\lambda}$, $V^\lambda_{\text{coll}}$, $D^{pp}$ and $V^p_{\text{coll}}$ are evaluated for all tetrahedra.
- `getran` evaluates $\xi_\lambda$ and $\xi_p$ from Eq. 6.
- `coleff` and `onseff` evaluate the phase space dependent parts of $D^{\lambda\lambda}$, $V^\lambda_{\text{coll}}$, $D^{pp}$ and $V^p_{\text{coll}}$.
- `stost` is called whenever a collision should be performed. It then calls the subroutines `coleff`, `onseff` and `getran`, evaluates $\Delta t$ and updates $\lambda$ and $p$ according to Eq. 6.

## References

[1] M. Eder *et al*, Physics of Plasmas **27**, 122508 (2020)
[2] M. Eder *et al*, *"GORILLA: Guiding-center ORbit Integration with Local Linearization Approach"* submitted to Journal of Open Source Software
[3] I.N. Dnestrovskii and D.P.Kostomarov, Springer Berlin Heidelberg, 1986. OCLC: 851371970.

## Results

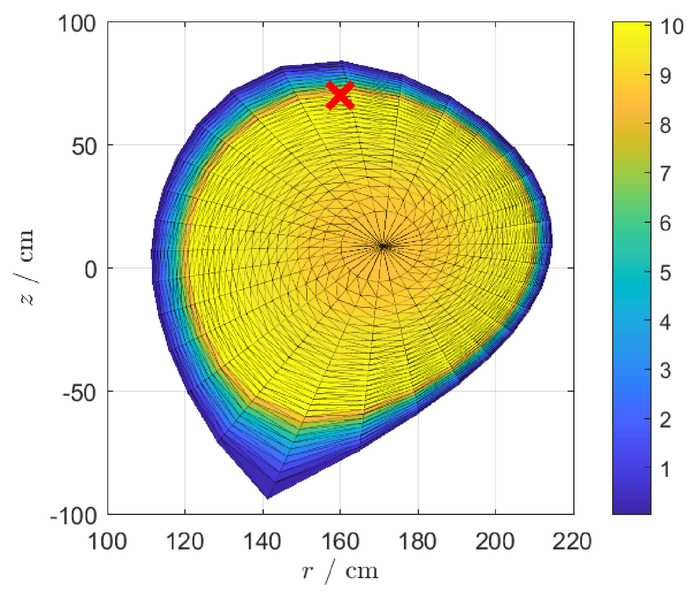


**Figure 1:** *Rescaled particle density produced by $10^5$ ionised tungsten particles (15 fold) with a temperature of 3.5 keV and deuterium as background particles. Each particle was initiated at the red x and traced until it left the computation domain (ASDEX Upgrade grid). CPU time ≈ 400 hours.*

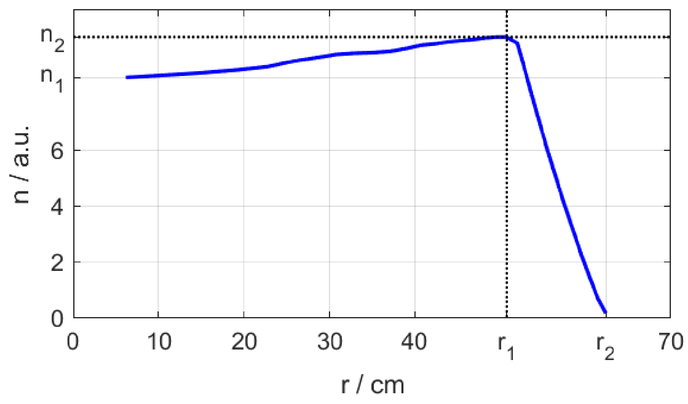


**Figure 2:** *Density in Fig. 1 averaged over poloidal angle*

Similarly to Eq. 5, the tracing time of the tungsten particles can be estimated according to

$$\Delta r \approx \sqrt{2\Delta t D_\perp} \Rightarrow \tau_{tr} \approx \frac{r^2}{2D_\perp} \approx 2\text{ s}, \tag{7}$$

agreeing well with a simulated median of 2.3 s.
As the computation domain has a tiny hole in the center, 0.7 percent of particles got lost there. Here, there is still a discrepancy to

$$\Gamma = -D_\perp\frac{\partial n}{\partial r} \Rightarrow \frac{\Gamma_{\text{int}}}{\Gamma_{\text{ext}}} \approx \frac{(r_2 - r_1)(n_2 - n_1)}{n_2 r_1} \approx 0.03. \tag{8}$$

## Summary and outlook

- GORILLA can trace impurity atoms all the way until they hit the device wall using a statistical operator to simulate collisions
- Because of its high computational efficiency, it can do so in the fraction of a time needed by other codes
- In the future, more complicated profiles of temperature, density and electric field are intended to be tested to gain more insight into the overall behaviour of impurity atoms.

QR code to our working group:

## Acknowledgments

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.