Computer Physics Communications 234 (2019) 126–136

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

FOCUS: A full-orbit CUDA solver for particle simulations in magnetized plasmas

C.F. Clauser [a,b,*], R. Farengo [b], H.E. Ferrari [a,b]

[a] *Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina*
[b] *Centro Atómico Bariloche and Instituto Balseiro, Comisión Nacional de Energía Atómica and Universidad Nacional de Cuyo, Av. Bustillo 9500, 8400 Bariloche, Argentina*

ARTICLE INFO

Article history:
Received 16 April 2018
Received in revised form 4 June 2018
Accepted 25 July 2018
Available online 3 August 2018

Keywords:
GPU code
Particle simulation
Magnetized plasma
Nuclear fusion

ABSTRACT

A Full-Orbit CUDA Solver for particle simulations in plasmas, FOCUS, has been developed. The code follows exact particle trajectories by solving Newton's equations with the Lorentz force. The code can use fields calculated analytically or numerically and it is also possible to use the information provided by equilibrium reconstruction and transport codes. FOCUS has an elastic collisions module which covers the whole particle energy range in magnetic fusion devices. Moreover, an atomic collision module was also included to simulate the interaction of the test particles with neutral or partially ionized species. Regarding performance, the main feature is that the code runs on GPUs, allowing the simulation of a large number of particles using moderate computational resources. Different versions of the code have been used in several papers. Here we present a complete description of its capabilities, the basis physics included and detailed information about the numerical algorithms employed. Finally, some examples are presented.

1. Introduction

Particle simulations, were the trajectories of individual particles are followed, have recently become a very common tool in plasma physics. This technique is particularly useful to study the dynamics of a relatively small population of high energy particles under the assumption that their motion does not modify the background plasma. The importance of energetic particles in current and future devices, in particular the production of alpha particles expected in ITER, and the rapid increase in the computing capability has prompted the development of many particle codes in recent years. Some codes solve the guiding center equations [1] while others calculate exact orbits [2] or have both options [3]. FOCUS was originally developed to study energetic particles, it can be used with any small population that satisfies the "tracer" approximation. In particular, it is currently being used to study the effect of MHD modes on impurities [4].

Since it is assumed that the particles do not interact with each other and do not modify the parameters of the background plasma, the problem is naturally amenable to be parallelized. This has led to the use of large CPU clusters to speed up the computations. In recent years, however, the availability of GPUs with large amounts of memory and computing capability has clearly changed the paradigm because each GPU can replace hundreds of CPU cores.

Here were present a full-orbit 3D code written in CUDA language that runs on a GPU. This allows simulations with a large number of particles using moderate computational resources. To the best of our knowledge, only one GPU particle code, LOCUST-GPU, has been reported thus far [5,6]. Our code, named FOCUS (an acronym for Full-Orbit CUda Solver), solves the exact equation of motion including elastic and inelastic collisions. A distinctive feature of our code is the use of a time dependent charge in the equation of motion to account for the effect of charge changing (inelastic) collisions.

The code was initially developed to run in a CPU and was used in several studies [7–9]. The current GPU code was already used in [10,11]. In FOCUS, each particle is identified with one CUDA thread and all simulated particles are considered independent from each other.

This paper presents a complete description of the code and is organized as follows. In Section 2 we present the basic structure of the code and in Section 3 the methods that can be employed to calculate or read the electric and magnetic fields. In Section 4 we present a detailed discussion of the elastic collision operator and in Section 5 we introduce the inelastic collision module. In Section 6, the numerical methods employed to integrate the equations of motion are presented while Section 7 contains test cases

[*] Corresponding author at: Centro Atómico Bariloche and Instituto Balseiro, Comisión Nacional de Energía Atómica and Universidad Nacional de Cuyo, Av. Bustillo 9500, 8400 Bariloche, Argentina.
E-mail address: cesar.clauser@ib.edu.ar (C.F. Clauser).

https://doi.org/10.1016/j.cpc.2018.07.018
0010-4655/© 2018 Elsevier B.V. All rights reserved.

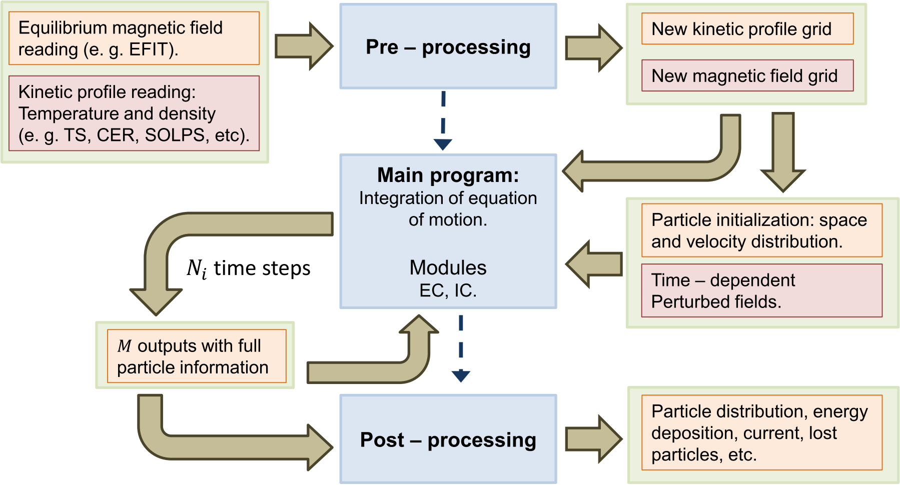
Equilibrium magnetic field reading (e. g. EFIT).
Kinetic profile reading: Temperature and density (e. g. TS, CER, SOLPS, etc).
Pre – processing
New kinetic profile grid
New magnetic field grid
Main program: Integration of equation of motion.
Modules EC, IC.
Particle initialization: space and velocity distribution.
Time – dependent Perturbed fields.
N_i time steps
M outputs with full particle information
Post – processing
Particle distribution, energy deposition, current, lost particles, etc.

Fig. 1. Qualitative diagram of the FOCUS code.

and applications. Finally, we summarize the present work in the Conclusions.

2. Code structure

As we mentioned, FOCUS follows exact particle trajectories by solving Newton's equation in 3D with the Lorentz force and includes the effect of elastic and inelastic collisions. Let us label each test particle with a subscript α. Then, the equations of motion can be written as (in cgs units)

$$\begin{aligned}\frac{d\mathbf{r}_\alpha}{dt} &= \mathbf{v}_\alpha \\ \frac{d\mathbf{v}_\alpha}{dt} &= \frac{q_\alpha(t)}{m_\alpha}\left(\mathbf{E}+\frac{\mathbf{v}_\alpha\times\mathbf{B}}{c}\right)+\dot{\mathbf{v}}_{\alpha,ec}.\end{aligned} \qquad (1)$$

Here, $q_\alpha(t)$ reflects the fact that the particle charge may change due to atomic (inelastic) collisions and $\dot{\mathbf{v}}_{\alpha,ec}$ is the velocity change produced by elastic collisions. Both elastic and inelastic collisions are stochastic processes with typical time scales that are much longer than the cyclotron period, which is related to the Lorentz force.

To solve Eq. (1) we need the following information:

a. The electric and magnetic fields, which could be time independent fields, or include time dependent perturbations.
b. The initial conditions of the particle ensemble.
c. The velocity changes produced by elastic (Coulomb) collisions.
d. The charge changes of the particle, which is determined by inelastic collisions.

Having all this information, we need algorithms to advance the different terms in time. The following sections describe the methods employed to calculate the different contributions appearing in Eq. (1) and the time integration techniques employed.

Fig. 1 shows a qualitative diagram of the FOCUS code, with the main modules and the flow information. Basically, the code has three main sections:

1. The pre-processing section, in which the magnetic field, kinetic profiles and other conditions are loaded and/or processed (i.e. boundary conditions, etc.). This section is required when the numerical equilibrium does not have enough resolution and a more accurate grid is needed (see next section). For analytical equilibria this section is not usually required.
2. The main program section which solves Eq. (1) with (or without) the information provided by additional modules, like elastic or inelastic collisions. In this section the particle initial conditions are provided and, optionally, time-dependent perturbed electromagnetic fields.
3. The post-processing section where the main program results, usually all the information available for an ensemble of particles at M different times, is processed.

3. Electric and magnetic fields

As a test particle code, FOCUS uses an external electromagnetic (EM) configuration. Both the electric and magnetic fields can be functions of space and time $\mathbf{B} = \mathbf{B}(\mathbf{r}, t)$, $\mathbf{E} = \mathbf{E}(\mathbf{r}, t)$. They can be obtained from analytical or numerical calculations or from the reconstruction of experimental measurements. In principle, it is also possible to input the total fields calculated, for example, by a code that solves the non-linear resistive MHD equations. Thus far, however, we have only used the code to study the motion of particles in 2D, static, magnetic fields or in 3D fields that are the sum of a 2D equilibrium magnetic field plus the 3D $\mathbf{B}$ and $\mathbf{E}$ fields produced by MHD perturbations. In the latter case we can write:

$$\mathbf{B}(\mathbf{r}, t) = \mathbf{B}_0(\mathbf{r}) + \mathbf{B}_1(\mathbf{r}, t) \qquad (2)$$

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{E}_1(\mathbf{r}, t) \qquad (3)$$

where $\mathbf{B}_0$ is the equilibrium magnetic field and $\mathbf{B}_1$ and $\mathbf{E}_1$ are the perturbations.

3.1. Equilibrium magnetic field

FOCUS can use analytical equilibria, equilibria obtained by numerically solving the equilibrium equation or equilibria reconstructed from the experimental information (e.g. by EFIT [12]). In particular, an analytical 2D equilibrium with circular cross section was employed in [8] and a numerical solution of the Grad–Shafranov equation in [11].

An accurate calculation of charged particle orbits requires knowing the magnetic and electric fields with enough accuracy along the orbits. Since numerical equilibria are usually calculated on a discrete mesh, it is important to guarantee that the mesh has enough resolution to obtain accurate orbits and the fields satisfy the divergence free condition.

In many important applications concerning axisymmetric devices the equilibrium is reconstructed from the experimental information using equilibrium codes such as EFIT. In these cases, the equilibrium code usually provides the flux function, used to calculate the poloidal field, and the toroidal field on a relatively coarse, 2D (r, z) grid. To interpolate this information to a fine enough grid and guarantee that the fields satisfy the divergence free condition we employ a Chebyshev polynomial expansion. This is the same as the method employed in the SPIRAL code [2]. We stress that the modulus to do the Chebyshev expansion was added to increase the versatility of the code but FOCUS can work with any 3D magnetic field that satisfies the divergence free condition and is accurate enough. When the Chebyshev expansion is employed, the poloidal flux can be written as

$$\Psi(r, z) = \sum_{i,j}^{n} a_{i,j} T_i(x(r)) T_j(y(z)) \tag{4}$$

where

$$x(r) = \frac{r - r_m}{\Delta r/2} \tag{5}$$

$$y(z) = \frac{z - z_m}{\Delta z/2} \tag{6}$$

and $r_m(z_m)$ are the mid values of the grid and $\Delta r(\Delta z)$ the width (height) of the grid (so that x and y are restricted to the $[-1, 1]$ interval). Here, the $T_i(x)$ are Chebyshev polynomials of the first kind and the $a_{i,j}$ coefficients are calculated using the discrete orthogonality relationship

$$a_{i,j} = \left(\frac{2}{n+1}\right)^2 \sum_{k,l}^{n} \Psi_0(r_k, z_l) T_i(x_k) T_j(y_l) \tag{7}$$

where

$$x_k = \cos\left(\frac{(k + 1/2)\pi}{n+1}\right) \tag{8}$$

$$y_l = \cos\left(\frac{(l + 1/2)\pi}{n+1}\right) \tag{9}$$

are the zeros of T_{n+1} and Ψ_0 the value provided by EFIT. As it is mentioned in Ref. [2], a very good convergence is usually reached with $n \sim 25$–30. With the new $\Psi(r, z)$ we generate a new, more refined, grid and the magnetic field is calculated at each node as

$$B_r = -\frac{1}{r}\frac{\partial \Psi}{\partial z} \tag{10}$$

$$B_z = \frac{1}{r}\frac{\partial \Psi}{\partial r}, \tag{11}$$

using the differentiation formula

$$\frac{dT_n(x)}{dx} = \frac{n}{2}\frac{T_{n-1}(x) - T_{n+1}(x)}{1 - x^2}. \tag{12}$$

This process guarantees a divergence-free magnetic field at each node. Then, for a given (r, z) particle position, we use a six-point bivariate interpolation [13]. As an example, the standard poloidal flux grid produced by EFIT for a DIII-D discharge usually has 65 × 65 nodes and the new poloidal flux grid usually has 400 × 400 nodes. This pre-processing task of FOCUS was also implemented in GPU. As with the magnetic field, there are no restrictions for the equilibrium electric field, which can be calculated analytically or numerically.

3.2. Perturbed fields

Different perturbed fields have been used in FOCUS [8]. These include, for instance, ideal first order MHD perturbations:

$$\mathbf{B}_1(\mathbf{r}, t) = \nabla \times [\xi(\mathbf{r}, t) \times \mathbf{B}_0] \tag{13}$$

$$\mathbf{E}_1(\mathbf{r}, t) = -\frac{1}{c}\frac{\partial \xi(\mathbf{r}, t)}{\partial t} \times \mathbf{B}_0 \tag{14}$$

where ξ is displacement field. In addition, ripple and other corrections to an axisymmetric equilibrium can be added.

4. Elastic collisions

The study of relaxation processes in plasmas is based on the Fokker–Planck (FP) theory (see i.e. [14]). This gives an equation for the distribution function f_α of a particle ensemble (again, we use the subscript "α" to refer to the test particles). For microscopic processes such as elastic collisions one can assume a locally homogeneous plasma. In this case the FP equation reads

$$\frac{\partial f_\alpha(\mathbf{v}, t)}{\partial t} = \left[-\frac{\partial}{\partial v_i} F_i(\mathbf{v}, t) + \frac{1}{2}\frac{\partial^2}{\partial \mathrm{v}_i \partial \mathrm{v}_k} D_{ik}(\mathbf{v}, t)\right] f_\alpha(\mathbf{v}, t) \tag{15}$$

where

$$F_i(\mathbf{v}, t) = \lim_{\Delta t \to 0} \frac{\langle \Delta \mathrm{v}_i \rangle_{\Delta t}}{\Delta t} \tag{16}$$

is the friction term and

$$D_{ik}(\mathbf{v}, t) = \lim_{\Delta t \to 0} \frac{\langle \Delta \mathrm{v}_i \Delta \mathrm{v}_k \rangle_{\Delta t}}{\Delta t} \tag{17}$$

is the diffusion tensor. These two quantities are also called the Fokker–Planck coefficients [14]. The FP equation gives the distribution function but in order to implement a particle collision operator we need to associate the FP equation with a Langevin equation, which represents the realization equation of each particle. In general, the relationship between the Langevin equation and the FP equation is not unique. If we assume an isotropic background plasma, and a reference frame where the z direction is parallel to $\mathbf{v}$, the diffusion tensor becomes diagonal

$$\mathbf{D}(\mathrm{v}) = \begin{pmatrix} \frac{1}{2}D_\perp & 0 & 0 \\ 0 & \frac{1}{2}D_\perp & 0 \\ 0 & 0 & D_\parallel \end{pmatrix} \tag{18}$$

and the friction term reduces to

$$\mathbf{F}(\mathrm{v}) = \left(0, 0, F_\parallel\right). \tag{19}$$

For these conditions, it is easy to associate the FP equation with a Langevin equation. The Langevin equation can be defined using the Îto or Stratonovich calculus. In either case, it must be consistent with the FP equation, Eq. (15). Here we use the Îto calculus and thus obtain the following Langevin equation

$$\frac{d\mathrm{v}_i}{dt} = F_i(\mathrm{v}, t) + \sqrt{D_{ii}(\mathrm{v}, t)}\,\xi_i(t), \tag{20}$$

which is the stochastic contribution to the equation of motion, $\dot{\mathbf{v}}_{ec}$ of Eq. (1), for each particle. Here, $\xi_i(t)$ is a Gaussian white noise.

For a Maxwellian background plasma formed by species β (each species with temperature T_β and density n_β) the friction and diffusion coefficients are known [14–16] and can be written in terms of collision frequencies

$$F_\parallel = -\nu_{sd}(\mathrm{v})\mathrm{v} \tag{21}$$

$$D_\parallel = \nu_\parallel(\mathrm{v})\mathrm{v}^2 \tag{22}$$

$$D_\perp = \nu_\perp(\mathrm{v})\mathrm{v}^2 \tag{23}$$

where

$$\nu_{sd} = \sum_\beta \frac{A_D^\beta}{2\mathrm{v}^3}\left(1 + \frac{m_\alpha}{m_\beta}\right)\left(\phi(x_\beta) - x_\beta \phi\prime(x_\beta)\right) \tag{24}$$

$$\nu_\parallel = \sum_\beta \frac{A_D^\beta}{\mathrm{v}^3} G(x_\beta) \tag{25}$$

$$\nu_\perp = \sum_\beta \frac{A_D^\beta}{\mathrm{v}^3}\left[\phi(x_\beta) - G(x_\beta)\right]. \tag{26}$$

Here $x_\beta = \mathrm{v}/\mathrm{v}_{s,\beta}$ with $\mathrm{v}_{s,\beta} = \sqrt{2T_\beta/m_\beta}$, $\phi(x)$ is the error function,

$$G(x) = \frac{\phi(x) - x\phi\prime(x)}{2x^2}, \tag{27}$$

and

$$A_D^\beta = \frac{8\pi q_\alpha^2 q_\beta^2}{m_\alpha^2} n_\beta \ln \Lambda_\beta. \tag{28}$$

The other quantities appearing in the previous equations are the charge state (q), the mass (m) and the Coulomb logarithm, which is defined as

$$\ln \Lambda_\beta = \ln \frac{b_{max}}{b_{min}}. \tag{29}$$

Here b_{max} (b_{min}) is a typical maximum (minimum) impact parameter for the collision. A simplified approximation for the Coulomb logarithm, which is commonly used, has been proposed by Spitzer [15], who considered, as shown below, a classical Coulomb logarithm for electrons (neglecting the ion contribution) and took the static limit

$$\ln \Lambda_{sp} = \ln \frac{3\mathrm{v}_{th,e}^3}{\omega_p q_\alpha}. \tag{30}$$

However, here we propose to keep it in a general form to accurately cover the whole range of energies that could appear during the simulations.

The upper limit b_{max} refers to the plasma screening length. For an ion in rest, this length is equal to the Debye's one [14]

$$\lambda_D = \left(\sum_\beta \frac{4\pi n_\beta q_\beta^2}{T_\beta}\right)^{-1/2} = \left(\sum_\beta \frac{\omega_\beta^2}{\mathrm{v}_{th,\beta}^2}\right)^{-1/2} \tag{31}$$

However, for an ion moving through a plasma it is possible to extend the previous definition by changing $\mathrm{v}_{th,\beta}^2 \to \bar{\mathrm{v}}_\beta^2 = \mathrm{v}_{th,\beta}^2 + \mathrm{v}^2$ [17] in order to include the adiabatic behavior. Hence, we use

$$b_{max} = \left(\sum_\beta \frac{\omega_\beta^2}{\bar{\mathrm{v}}_\beta^2}\right)^{-1/2} \tag{32}$$

where $\omega_p = \sqrt{4\pi n_\beta q_\beta^2/m_\beta}$ and $\mathrm{v}_{th,\beta} = \sqrt{T_\beta/m_\beta}$.

The lower limit b_{min} refers to the minimum impact parameter in the collision. In this case, there are two typical limits [18]: the classical limit

$$b_{min,\beta}^{CL} = \frac{q_\alpha q_\beta}{m_{\alpha\beta}\,\bar{\mathrm{v}}_\beta^2} \tag{33}$$

and the quantum-mechanical limit [19]

$$b_{min,\beta}^{QM} = \frac{\hbar}{2m_{\alpha\beta}\,\bar{\mathrm{v}}_\beta}. \tag{34}$$

Here $m_{\alpha\beta}$ is the reduced mass. The use of one or another limit depends on the parameter

$$\eta = \frac{1}{2}\frac{b_{min,\beta}^{CL}}{b_{min,\beta}^{QM}} = \frac{q_\alpha q_\beta}{\hbar\,\bar{\mathrm{v}}_\beta} \tag{35}$$

which separates the classical ($\eta \gg 1$) and quantum-mechanical ($\eta \ll 1$) regions. As an example, if we are interested in suprathermal particle dynamics (i.e. deuterium–tritium–alpha particles) which are low-charged ions, the proper limits are the quantum one for electrons and the classical one for ions. But, if we consider slow highly-charged impurities like W^{+35} the classical is the appropriate limit for both electrons and ions.

Fig. 2 shows the Coulomb logarithm for collisions between a deuterium test particle and a deuterium plasma with $T_\beta = 10$ keV and $n_\beta = 10^{20}$ m^{-3}, Eq. (29) (here β refers to plasma deuterons and electrons). The dotted line is the classical limit for ions, Eq. (33), and the dashed line is the quantum limit for electrons, Eq. (34). The figure also shows the Spitzer Coulomb logarithm, Eq. (30). As can be noted in this example, the use of the latter can produce an error up to 20% in the collision frequencies, arising mostly from the electronic contribution, which is more important at high energies. Other Coulomb logarithms have also been proposed [18,20].

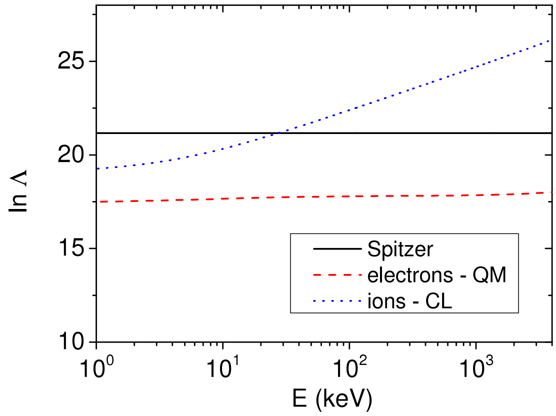

Fig. 2. Coulomb logarithm for a deuterium ion in a deuterium plasma with $T = 10$ keV and $n = 10^{20}$ m^{-3}.

In real devices, kinetic profiles (temperatures and densities) are not uniform. Usually, electron temperature and density can be obtained from Thomson Scattering (TS) and the ion temperature from Charge Exchange Recombination (CER). A common approximation is to use these measurements to generate 1D profiles (as a function of the poloidal flux). In addition, transport codes can usually lead into 1D or 2D kinetic profiles. FOCUS can read these profiles to compute the elastic collisions frequencies.

4.1. Numerical implementation

As we mentioned above, the Langevin equation (20) is related to the FP equation (15) using the Îto calculus. One of the simplest Îto algorithm is the well-known Euler one [21] which is enough to give very accurate results [22]. Thus, for each simulated particle, the elastic collision module updates the particle velocity in the following way

$$\Delta \mathrm{v}_\parallel = -\nu_s \Delta t_{ec}\, \mathrm{v}_0 + \sqrt{\nu_\parallel \Delta t_{ec}}\, \mathrm{v}_0 N_1 \tag{36}$$

$$\Delta \mathrm{v}_{\perp 1,2} = \sqrt{\frac{\nu_\perp \Delta t_{ec}}{2}}\, \mathrm{v}_0 N_{2,3} \tag{37}$$

where v_0 is the velocity before the collision, the N_i are Gaussian random numbers ($\langle N_i \rangle = 0$, $\langle N_i^2 \rangle = 1$) and Δt_{ec} is the time step to evaluate the elastic collision operator. As we mentioned before, this implementation is referred to a local frame where the z axis is parallel to the particle velocity $\mathbf{v}_0$. Let us denote this local system by $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$ where $\hat{e}_1 = \hat{z}$. Then, the final velocity (after a collision evaluation) is

$$\mathbf{v} = \mathbf{v}_0 + \Delta \mathrm{v}_\parallel \hat{e}_1 + \Delta \mathrm{v}_{\perp 1} \hat{e}_2 + \Delta \mathrm{v}_{\perp 2} \hat{e}_3. \tag{38}$$

The accuracy of this algorithm relies on the fact that $\nu \Delta t_{ec} \ll 1$ (for the all frequencies employed). We refer to this integration method as "V–E".

There are several random number generators for GPUs [23]. We use Philox4x32 [24] which guarantees a large stream of unique random numbers with large periods. This generator also offers a good performance. However, the use of other random generators is also possible.

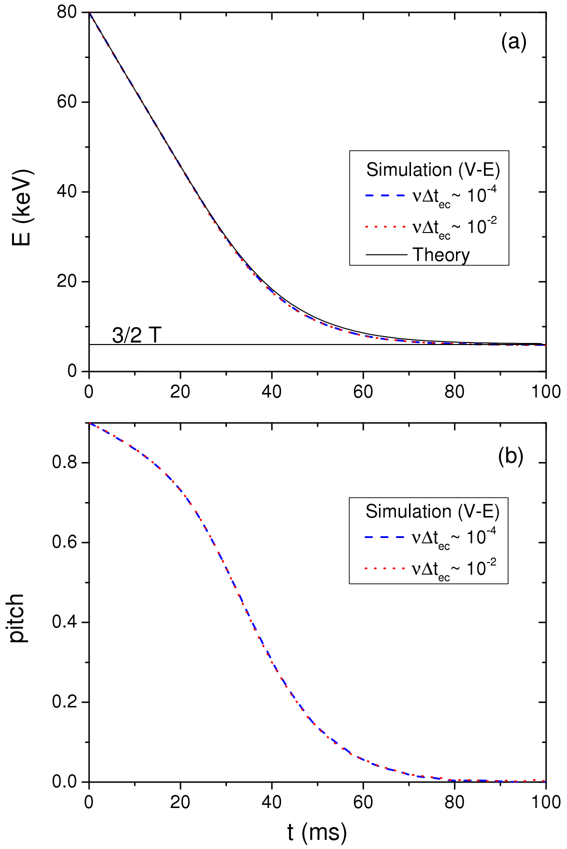

Fig. 3. Mean energy and pitch as a function of time for an ensemble of deuterons in a homogeneous plasma with $T = 4$ keV and $n = 10^{20}$ m^{-3}. The dashed/dotted lines were obtained from particle simulations while the solid lines in (a) was obtained from a FP integration (see the text).

As a test of the elastic collision operator, Fig. 3 shows the mean energy (a) and pitch (b) as a function of time for an ensemble of 10^5 deuterons. The initial energy and pitch are 80 keV and 0.9, respectively. The background plasma is uniform with $T_e = T_i = T = 4$ keV, $n_e = n_i = n = 10^{20}$ m^{-3} and the magnetic field is also uniform, $\mathbf{B} = B\hat{z}$ with $B = 1.9$ T. The pitch is defined as $\lambda = \cos\theta = \hat{\mathrm{v}}\cdot\hat{z}$.

In each figure, two cases with different Δt_{ec} are shown. For the dashed line we have $\nu\Delta t_{ec} \lesssim 10^{-4}$ and for the dotted line $\nu\Delta t_{ec} \lesssim 10^{-2}$. In addition, Fig. 3(a) also includes theoretical curves to compare them with the simulation results. These include the stationary mean energy $\langle E\rangle = 3T/2$ and a time-dependent solution obtained by solving a FP equation (see Appendix). Both curves are shown in Fig. 3(a) with a solid black line. From these comparisons we can observe that the Euler (V–E) method gives an excellent result.

5. Inelastic collisions

Atomic (inelastic) collisions are very important for different processes in plasma physics. In magnetic fusion devices, these processes are important for neutral beam injection, particle losses, diagnostic techniques and impurity dynamics, among others. An inelastic collisions module has been included in FOCUS to take into account charge changing processes (and excitation processes for deuterium test particles).

Even though in FOCUS we were initially interested in the role of atomic processes in the dynamics of alpha particles [10,11], other studies are also possible. Usually the main limitation of this module is the lack of atomic data for the whole range of energies and/or charge states needed for a consistent simulation. Only the case of Hydrogen (and its isotopes) is widely covered. Hence, more and accurate cross section are needed [25].

5.1. Collision frequency

Let us consider a process between a test particle α and a medium β (target particles), which changes the α-particle charge state from q_1 to q_2 (in the case of excitation, the charge remains constant but not the quantum state). The quantity which measures if this process is important is the collision frequency $\nu_{\alpha\beta}^{q_1\to q_2}$, which can be written in the following way [26]

$$\begin{aligned}\nu_{\alpha\beta}^{q_1\to q_2}(\mathbf{r},t) &= n_\beta\langle\sigma\mathrm{v}\rangle_{\alpha\beta}^{q_1\to q_2}\\ &= n_\beta\int d\mathbf{v}_\beta f_\beta(\mathbf{r},\mathbf{v}_\beta,t)\sigma_{\alpha\beta}^{q_1\to q_2}(\mathrm{v}_r)\mathrm{v}_r \qquad (39)\end{aligned}$$

where $\mathrm{v}_r = \mid \mathbf{v} - \mathbf{v}_\beta \mid$, $\sigma_{\alpha\beta}^{q_1\to q_2}(\mathrm{v}_r)$ is the total cross section and f_β is the target distribution function. The quantity $\langle\sigma v\rangle$ is called the *reactivity*. Then, the total collision frequency for changes from charge state q_1 to q_2 is

$$\nu_\alpha^{q_1\to q_2} = \sum_\beta \nu_{\alpha\beta}^{q_1\to q_2}. \qquad (40)$$

On the one hand, for collisions with plasma particles (electrons and ions), we consider that they have a Maxwellian distribution function. In such a case, the reactivity reduces to

$$\langle\sigma\mathrm{v}\rangle_{\alpha\beta}^{q_1\to q_2} = \frac{1}{(2\pi T_\beta/m_\beta)^{1/2}}\frac{1}{\mathrm{v}}\times \int_0^\infty d\mathrm{v}_r \mathrm{v}_r^2\sigma(\mathrm{v}_r)_{\alpha\beta}^{q_1\to q_2}\left[e^{-(\mathrm{v}_r-\mathrm{v})^2/\mathrm{v}_{s,\beta}^2} - e^{-(\mathrm{v}_r+\mathrm{v})^2/\mathrm{v}_{s,\beta}^2}\right]. \qquad (41)$$

On the other hand, for collisions with neutral atoms we consider that they are "cold", in the sense that their thermal energy, usually a few eV's [27,28], is much less than the test particle energy. Thus, in this case, the distribution function can be approximated by a delta function and the reactivity reduces to

$$\langle\sigma\mathrm{v}\rangle_{\alpha\beta}^{q_1\to q_2} = \sigma(\mathrm{v})_{\alpha\beta}^{q_1\to q_2}\mathrm{v}. \qquad (42)$$

Depending on the test particles and the plasma region, different sets of atomic processes should be considered (see i.e. [28,29]). For instance, for alpha particles, charge exchange with neutrals and plasma deuterons as well as electron impact ionizations are the most relevant [10]. It is also clear that the addition of more processes increases the charge change probability.

The cross sections are obtained from existing databases and used to calculate the corresponding collision frequencies, as indicated above. These frequencies are then stored in tables to speed up their use.

5.2. Density and temperature profiles

In addition to cross section availability, we need density and temperature profiles to evaluate collision frequencies. As we mentioned before, inside the plasma, the temperature and density can be obtained from Thomson scattering (TS) or Charge Exchange Recombination (CER) spectroscopy, as well as from transport codes. Outside the plasma, a 2D-model should be employed. For this reason, FOCUS has the capability of reading SOLPS [30,31] or UEDGE 2D profiles. Assuming that the data provided by these codes consist of a set of points distributed over the plasma edge-SOL, FOCUS maps this data once into a (r, z) grid. To do this, we have implemented the following routine: given a (r, z) point, we search for k (usually $k = 4$) neighbors on the provided set of

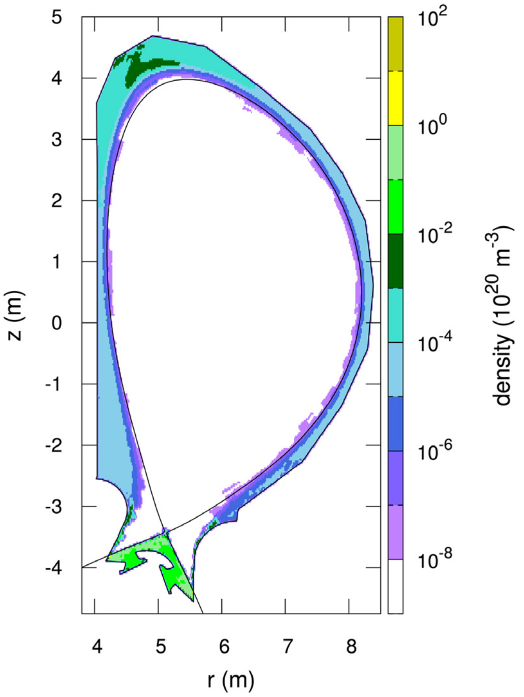

Fig. 4. Cold neutral deuterium density read from the SOLPS-ITER code [32].

data and separate them according to their poloidal flux value, keeping the m-closest (usually $m = 2$) neighbors. Finally, an inverse distance weighted algorithm is employed. Hence, with a rectangular (r, z) grid, we optimize the process of reading the 2D densities or temperatures. As an example, Fig. 4 shows a typical 2D cold neutral deuterium density distribution reconstructed from the data provided by SOLPS-ITER [32].

5.3. Numerical implementation

Defining a time step Δt_{ic}, the probability that a charge change process occurs can be approximated by

$$\text{Prob.}_{\alpha}^{q_1\to q_2} = \nu_{\alpha}^{q_1\to q_2}\Delta t_{ic} \tag{43}$$

which is valid when $\nu_{\alpha}^{q_1\to q_2}\Delta t_{ic} \ll 1$. Hence, we can decide if the charge changes by generating a single uniform random number.

6. Integration of the equation of motion

To reduce the numerical error, the code uses the following dimensionless quantities

$$r \to \tilde{r} = r/a \tag{44}$$

$$\mathrm{v} \to \tilde{\mathrm{v}} = \mathrm{v}/\mathrm{v}_0 \tag{45}$$

$$t \to \tilde{t} = \Omega_0 t \tag{46}$$

$$B \to \tilde{B} = B/B_0 \tag{47}$$

$$E \to \tilde{E} = E/(B_0\mathrm{v}_0/c) \tag{48}$$

The set a, v_0, Ω_0, B_0 should be properly chosen. In our case, a is the minor radius, v_0 is a typical initial velocity of the particle ensemble, B_0 is the vacuum toroidal magnetic field at the major radius R_0 and Ω_0 is the cyclotron frequency calculated with B_0 and taking into account the nuclear projectile charge Z_α.

With these dimensionless quantities, the equations of motion, Eqs. (1), result

$$\frac{d\mathbf{r}_\alpha}{dt} = \gamma\mathbf{v}_\alpha$$

$$\frac{d\mathbf{v}_\alpha}{dt} = \frac{q_\alpha(t)}{Z_\alpha}(\mathbf{E}+\mathbf{v}_\alpha\times\mathbf{B}) + \dot{\mathbf{v}}_{\alpha,ec}, \tag{49}$$

where

$$\gamma = \frac{\mathrm{v}_0}{\Omega_0 a}.$$

All these quantities are dimensionless and the "tilde" has been omitted for simplicity. Note that only one dimensionless parameter, γ, appears in the equations.

FOCUS has two different algorithms to integrate Newton's equations: the well-known Boris algorithm [33,34] and a fourth-order six-stage low diffusive Runge–Kutta (RK46NL) algorithm [35]. In both cases the integrator advances only the Lorentz force term in Eq. (49). The time scale for the Lorentz force term is given by the cyclotron frequency, Ω_0. Usually, the time step employed varies between $0.05\Omega_0^{-1}$ and $0.15\Omega_0^{-1}$ (in the following we will omit the Ω_0^{-1} unit when referring to time steps). As an example, for a deuteron or alpha particle we have $\Omega^{-1} = 1.099 \times 10^{-8}$ s for a 1.9 T magnetic field (DIII-D like condition), and $\Omega^{-1} = 3.94 \times 10^{-9}$ s for a 5.3 T field (ITER like condition).

As we mentioned before, elastic and inelastic collision time scales are determined by their collision frequencies. In magnetic fusion devices, these frequencies are much lower than the cyclotron frequency and thus, the time scales are much longer. As a consequence, we can separate these integrations from the Lorentz force integration. On one hand, the contribution of the elastic collisions term is calculated separately using the Monte Carlo type method described in Section 4.1, with a time step Δt_{ec}. On the other hand, the charge advances with a time step Δt_{ic} using the information provided by the inelastic collisions module (if active), as described in Section 5.3.

Fig. 5 shows the banana orbit of an 80 keV deuteron in a DIII-D equilibrium, calculated with the Boris integrator and a time step of 0.05. There are no collisions in this calculation. For clarity, only a part of the initial full orbit is shown; the remaining curves show the guiding centers, calculated from the full orbit,

$$\mathbf{r}_{gc} = \mathbf{r}_\alpha + \rho_L\frac{\mathbf{v}_\alpha\times\mathbf{B}}{|\mathbf{v}_\alpha\times\mathbf{B}|}, \tag{50}$$

at two different times (here ρ_L is the Larmor radius at the particle position). The black curve is the initial guiding center orbit and the red one is the orbit obtained after 100 ms, which corresponds to 3200 bounce periods. We can see that there is an excellent agreement between both orbits. In fact, the energy is implicitly conserved by the Boris algorithm while the toroidal component of the canonical momentum, defined as (in dimensionless units)

$$p_\theta = r\mathrm{v}_\theta + \frac{\Psi}{\gamma},$$

is also very accurately conserved in this calculation (with a relative error of $\sim 0.25\%$). Taking into account typical DIII-D parameters ($n = 5\cdot 10^{19}\,\mathrm{m}^{-3}$ and $T = 4\,\mathrm{keV}$) the slowing down time for the deuteron is ~ 90 ms. This indicates that the integrator is very accurate for the whole simulation time. For passing particles the accuracy is even better. Although it is not shown, the RK46NL algorithm gave very similar results for the simulated time. However the Boris algorithm is much faster.

In general, full orbit codes have been used for short-time simulations. However, with the high parallelization of GPUs it is possible to simulate larger numbers of particles for longer times. For instance, the simulation of the dynamics of fusion born alpha particles in ITER. This case is more demanding due to the much longer collision times. To test the accuracy of the Boris integrator in these conditions, we followed the evolution of a trapped 3.5 MeV alpha particle in an ITER equilibrium for 600 ms using different time steps, $\Delta t = 0.05$, 0.1 and 0.2. Fig. 6(a) shows a plot of p_θ as a

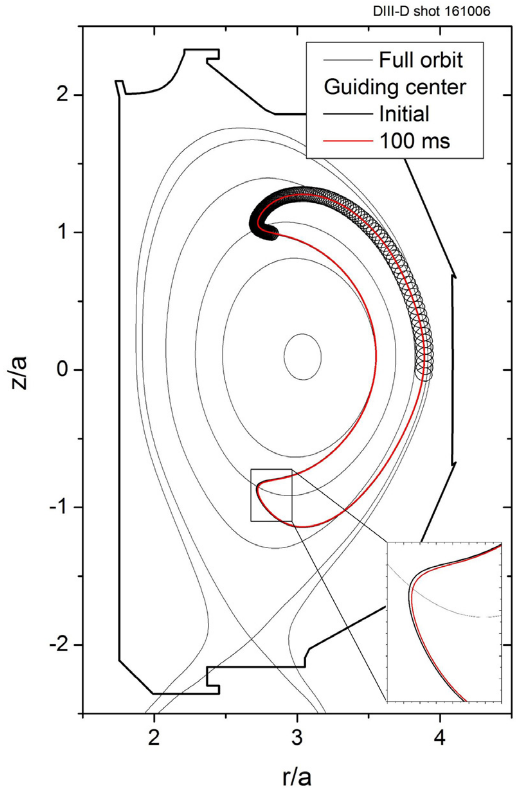

Fig. 5. Banana orbit of an 80 keV deuteron in the DIII-D tokamak. The full-orbit is, only for clarity, partially shown. The guiding center orbits were obtained from the full-orbit and they are shown at $t = 0$ and after 100 ms.

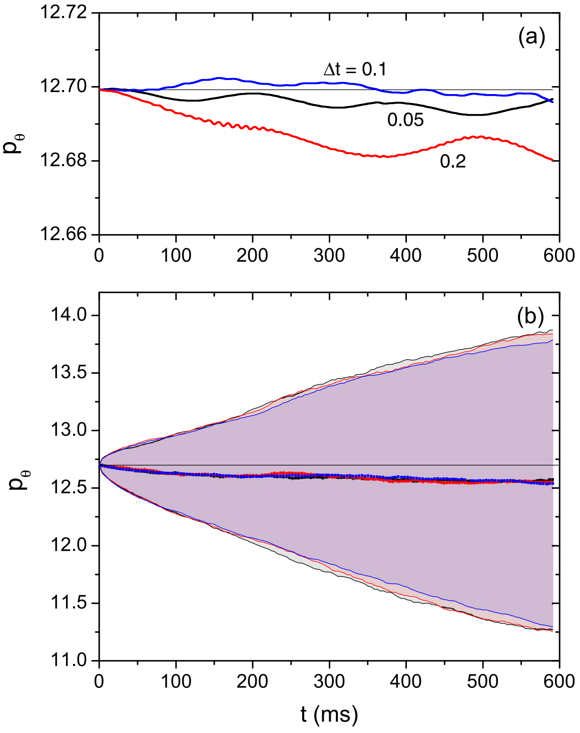

Fig. 6. p_θ time evolution for a 3.5 MeV trapped alpha particle in an ITER equilibrium, for three different time steps: $\Delta t = 0.05$, 0.1 and 0.2 (in Ω_0^{-1} unit). In (a) we followed a particle without elastic collisions (EC) while in (b) we followed an ensemble of 1000 particles with EC.

function of time without elastic collisions. It can be seen that p_θ is very well conserved in the all cases but a small error is introduced.

Although it is clear that the errors remain very small for the whole simulation their relevance can be further assessed by comparing the error with the dispersion produced by elastic collisions. Thus, we included elastic collisions and followed the evolution of 1000 particles with the same initial conditions. Fig. 6(b) shows plots of the average value of p_θ with its corresponding dispersion (given by the square root of the ensemble variance) for the same conditions and time steps as in Fig. 6(a). We can see that in this case the p_θ of the ensembles have a very similar behavior for the three time steps employed and the dispersion is much larger than the errors observed in the case without collisions. We therefore argue that as the random “noise” introduced by elastic collisions is much larger than the small error due to the Boris integrator, a strong convergence of the orbits is not necessary and the weak convergence reached with elastic collisions is enough (here “strong” and “weak” convergence are used as in the context of stochastic differential equations [21]). A similar idea was already discussed in Ref. [22] in the context of Langevin equations collision models.

Electric fields have not been taken into account in previous examples, but the Boris algorithm also offers a very good performance when they are included [34].

6.1. Speed-up

We determined the speed-up obtained when switching from the old (CPU) to the new (GPU) versions of the code. An analytic, circular cross section equilibrium was employed (no fluctuations) and collisions were not included. A GTX 1080 GPU and one 2.20 GHz CPU were used for the comparison.

Fig. 7 shows the computational time t_c as a function of the simulation time t_s for both the CPU and GPU versions. Three cases, with different number of particles, $N_p = 10^3$, 10^4 and 10^5, were simulated. In order to put all the results together, we consider $N_p = 10^5$ as the reference case. The CPU version is strongly-linear scalable with the number of particles, N_p and the number of time steps, N_i. This means that a case with $N_p = 10^3$ and $N_i = 10^7$, takes the same computational time as a case with $N_p = 10^4$ and $N_i = 10^6$, and $N_p = 10^5$ and $N_i = 10^5$. Hence, only the case with $N_p = 10^5$ is shown for the CPU case. The GPU version does not scale linearly for a small number of particles but turns on linear when the number of particles increases enough and exceeds the number of GPU cores (usually a few thousands of particles). This means that doing 100 runs with 10^3 particles (magenta line in Fig. 7) takes considerably longer time doing 10 runs with 10^4 particles or 1 run with 10^5 particles (navy blue line).

For the CPU and GPU used in the simulations, we can observe from Fig. 7 that the current GPU version has a speed-up factor of ∼ 200. This means that we need around 200 similar CPU cores to get the same performance as the one obtained with a single GTX 1080 GPU.

7. Test cases and applications

In this section we show how FOCUS can be used to address different situations. First, we present an example of the calculation of a stationary fast alpha particles profile and then, a summary of other applications, which have been studied or can be addressed.

7.1. Stationary fast alpha particle distribution

Calculating the distribution of high energy alpha particles in configuration and velocity spaces is a necessary first step when studying the effect of different processes (large scale MHD fluctuations, Alfvèn eigenmodes, collisions, etc.) on the confinement

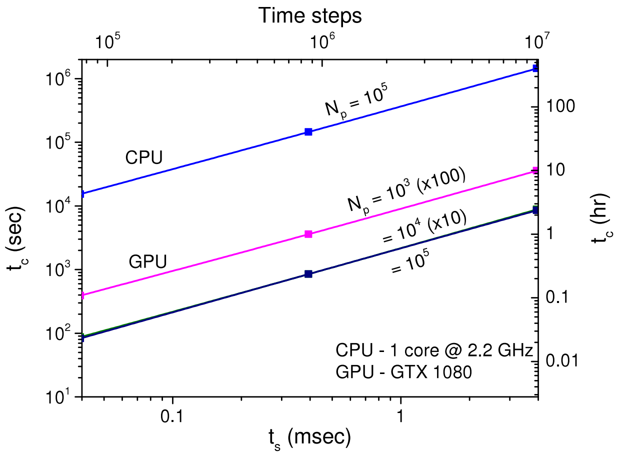

Fig. 7. Comparison of the computational time (t_c) vs. simulation time (t_s) between the GPU version with the old CPU one. For each case, $N_p = 10^3 - 10^5$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

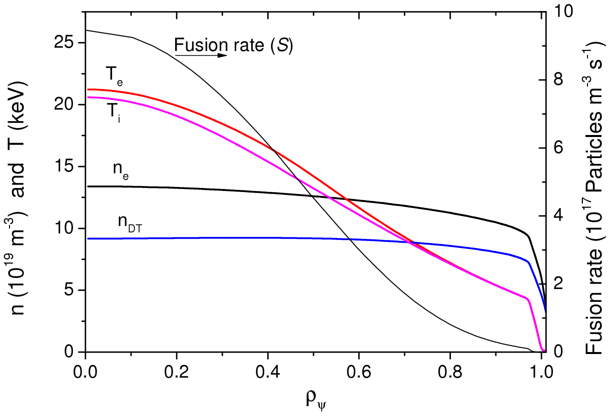

Fig. 8. Plasma profiles and fusion rate of an ITER $Q = 10$ scenario.

of these particles. In addition, this calculation is also useful to test the code and compare the results with those of other well-known codes, such as ASCOT [3].

The simplest approximation for the velocity distribution of fast alpha particles is the well-known slowing down distribution function

$$f_{sd}(\mathrm{v}) = \begin{cases} \dfrac{S}{4\pi}\tau_{E,e}\dfrac{1}{\mathrm{v}_c^3+\mathrm{v}^3} & (\mathrm{v} \le \mathrm{v}_0) \\ 0 & (\mathrm{v} > \mathrm{v}_0) \end{cases} \tag{51}$$

where v_0 is the velocity of 3.5 MeV alpha particles,

$$\tau_{E,e}^{-1} = \nu_{E,e}(0) = \frac{16}{3}\sqrt{\pi}\frac{q_\alpha^2 e^2 n_e \ln \Lambda_e}{m_\alpha m_e \mathrm{v}_{s,e}^3} \tag{52}$$

is an electron collision frequency (see Appendix) and

$$\mathrm{v}_c^3 = \frac{3\pi^{1/2}}{4}\frac{m_e}{m}\left(\sum_{\beta=D,T}\frac{n_\beta}{n_e}\frac{\ln \Lambda_\beta}{\ln \Lambda_e}\frac{m}{m_\beta}\right)\mathrm{v}_{s,e}^3 \tag{53}$$

is the critical velocity. In particular, this solution can be easily obtained from the FP equation in polar coordinates as shown in Appendix. Considering a 50% D–T plasma and approximating $\ln \Lambda_\beta = \ln \Lambda_e$, the critical velocity reduces to [36]

$$\mathrm{v}_c^3 = \frac{5\pi^{1/2}}{4}\frac{m_e}{m}\mathrm{v}_{s,e}^3. \tag{54}$$

The source, S, is the fusion reaction rate per unit volume

$$S = n_D n_T \langle \sigma v \rangle_{DT} \tag{55}$$

where n_D (n_T) is the deuterium (tritium) density and $\langle \sigma v \rangle_{DT}$ is the fusion reactivity [37].

To simplify particle initialization, it is possible to generate a random number with the slowing down distribution starting from a uniform random number $x \in U(0, 1)$. Using the fundamental transformation law of probabilities, we obtain

$$y = \left[\mathrm{v}_c^3\left(1+\frac{\mathrm{v}_0^3}{\mathrm{v}_c^3}\right)^x - \mathrm{v}_c^3\right]^{1/3} \tag{56}$$

which is a random number with a probability distribution function

$$p(y) = 4\pi y^2 \tilde{f}_{sd}(y) \tag{57}$$

where $\tilde{f}_{sd}$ is the normalized slowing down distribution function.

The slowing down distribution function offers a starting point to obtain a stationary distribution of energetic alpha particles. We present here a simulation with the parameters of an ITER $Q = 10$ scenario [38]. The radial electric field was not included in the simulations because it has been shown [39] that for our conditions ($I = 15$ MA, scenario 2 with no error fields) it does not affect the loss of fusion born alpha particles. In the simulation we followed an ensemble of 5×10^4 particles initialized with a spatial distribution given by a probability proportional to the source S, Eq. (55), and a velocity distribution given by the slowing distribution, Eq. (57). The radial profiles employed in this simulation [40] and the source (or fusion rate per volume) are shown in Fig. 8 as functions of the square root of the normalized poloidal flux $\rho_\psi = \sqrt{\psi_n}$, where $\psi_n = (\Psi - \Psi_0)/(\Psi_b - \Psi_0)$, with Ψ_0 and Ψ_b the values at the magnetic axis and at the separatrix, respectively. In this simulation we did not consider the thermal component of the alpha particles (Helium ash). Thus, we replaced the particles whose energy fell below 50 keV by 3.54 MeV particles. The new particles were distributed in space with a probability proportional to the fusion reaction rate and their velocity distribution was isotropic. The particles which escape from the plasma were also replaced using the same method.

Fig. 9 shows global parameters of the simulated ensemble as a function of time. We have simulated approximately 800 ms. Fig. 9(a) shows the mean energy and Fig. 9(b) shows the particle replacement or reposition rate (discussed above). Fig. 9(c) and (d) show the mean normalized poloidal flux of the particles and its variance, respectively. These two provide information about the spatial distribution of the particles. As it can be observed, all these quantities stabilize after $t = 600$ ms. Furthermore, in the stationary regime we can calibrate the reposition rate, Fig. 9(c), with the real total fusion rate and hence identify how many real particles are represented by a simulated particle. For the plasma profiles used, this gives that $\sim$ 97 simulated particles are equivalent to 1.74×10^{20} real particles. This value was employed to calibrate the stationary fast alpha particle density, which is shown in Fig. 10. To produce this plot we performed five independent simulations with 10^4 particles. The black curve in Fig. 10 is the average over the ensemble and the red band represents the standard deviation. This profile is in excellent agreement with those obtained by the well-known ASCOT code for similar kinetic profiles (see i.e. [41,42]).

7.2. Other applications

Since its first version, the versatility and capabilities of the code were continuously increased. Elastic and inelastic collisions were added, and the CUDA version developed. Finally, the possibility of employing the information produced by equilibrium and transport

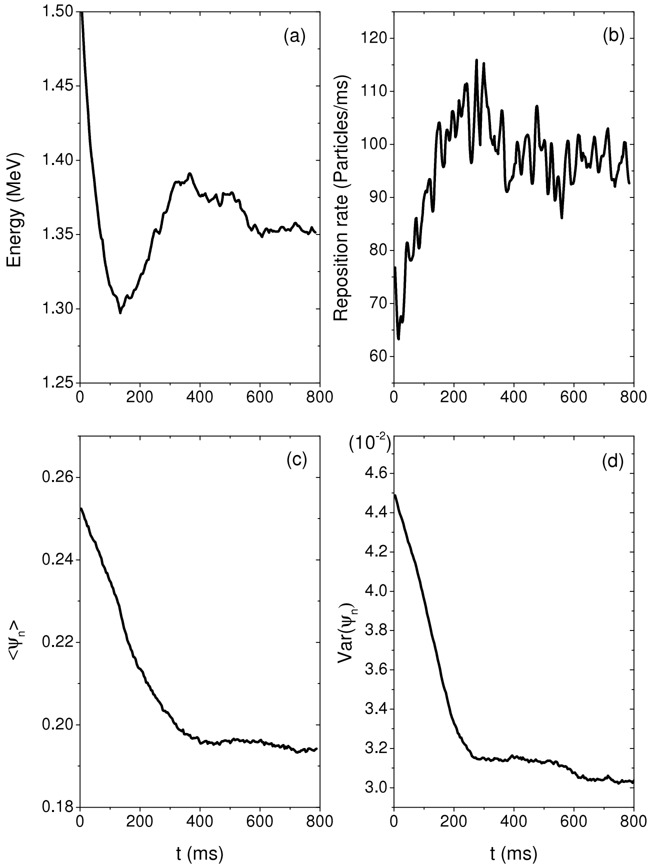

Fig. 9. Time evolution of global parameters of an ensemble of 5 × 10^4 alpha particles, initialized with a slowing down distribution function and with the spatial distribution of the fusion source S, for an ITER $Q = 10$ equilibrium.

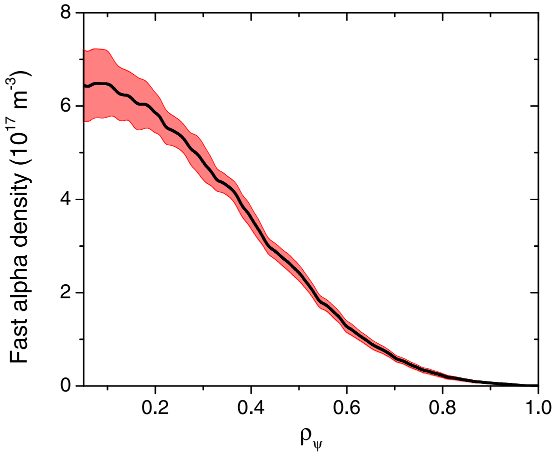

Fig. 10. Stationary fast alpha particle density obtained with FOCUS for an ITER $Q = 10$ equilibrium. The error is shown in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

codes widely used by the fusion community (EFIT, UEDGE, SOLPS, etc.) was added. Some of the studies conducted with the code are listed below.

The code was initially developed to study the effect of kink modes on the redistribution of alpha particles [8]. In that work we employed a simple analytical equilibrium and introduced a formalism that allowed us to use the existing experimental information about the spatial structure of the modes and their amplitude and frequency. The perturbed fields produced by the (1, 1) and (2, 2) modes were included in the calculation of the exact orbits of a large number of particles. Using the information about the individual particle orbits a diffusion coefficient that quantifies the magnitude of the displacement from the initial flux surface produced by the perturbation was introduced. Classical collisions were not included.

Subsequent studies [9] on the redistribution of alpha particles included also the (2, 1) mode. The addition of this mode, which can produce the stochastization of the magnetic field, increased particle redistribution significantly and allowed them to spread beyond the $q = 1$ surface and reach the plasma edge. These studies considered particles with energies between 18 keV and 3.5 MeV, with a slowing down distribution function, and modes with different frequencies and amplitudes. In other studies [4], the effect of kink modes on impurity ions was calculated.

With the addition of inelastic collision it becomes possible to study the effect of charge changing processes on the dynamics of alpha particles [10,11]. These studies showed that charge changing processes can produce significant changes in the transport of alpha particles in the edge-SOL region provided the neutral density is high enough. The addition of inelastic collisions actually reduces the alpha particle loss rate below the level obtained when only elastic (Coulomb) collisions are included. This is due to the inward flux produced by the neutral density gradient. Power losses, on the other hand, remain at approximately the same level because the average energy of the lost particles is higher when inelastic collisions are included. Finally, the spatial distribution of the lost particles changes significantly when inelastic collisions are added, with a larger fraction of the lost particles reaching the wall.

8. Conclusion

We presented a 3D full-orbit code for particle simulations in magnetized plasmas, FOCUS, which was written in CUDA language and runs on GPUs. The code has the capability of performing long time simulations with a large number of particles using moderate computational resources. FOCUS can use analytical as well as numerical equilibria. As an example, it can read EFIT outputs and, in this work, we have presented DIII-D and ITER cases. In addition, time dependent perturbed fields can be added. At present, FOCUS includes elastic and inelastic collisions modules. For this modules, the code can read arbitrary kinetic profiles such as 1D or 2D profiles. In particular, the code can read SOLPS or UEDGE outputs. As far as we know, this is the first full orbit code that includes charge exchange collisions between alpha particles and neutral deuterium.

To show the code's capabilities, we calculated a stationary energetic alpha particle profile for an ITER $Q = 10$ scenario, starting from the slowing down distribution and the fusion source. This case was also useful to compare the code's result with the well-known ASCOT code.

FOCUS is a very flexible code and has already been used to study many different problems. These include the effect of inelastic collisions on alpha particle transport, the redistribution of alpha particles due to ideal MHD perturbations and the effect of kink modes on the dynamics of impurity ions. The possibility of using the information generated by equilibrium and transport codes which are widely used by the fusion community makes it a very useful tool to study different problems under realistic conditions.

Acknowledgments

This work was supported by the Comisión Nacional de Energía Atómica, CNEA (Controlled Nuclear Fusion Program) and Universidad Nacional de Cuyo (Grant 06/C416). C.F. Clauser acknowledges the support of the Fulbright Program and the Ministerio de

Educación y Deportes, Argentina. C. F. C. is also very grateful to J. A. Boedo for fruitful discussions. The authors thank DIII-D for providing the equilibrium data employed in this paper, which is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Award DE-FC02-04ER54698. The authors also acknowledge X. Bonnin, A. Polevoi and A. Loarte for providing the ITER data employed in this work.

Appendix. Fokker–Planck equation for the mean energy calculation

In Section 4 we compared the energy loss obtained from particle simulations using the FOCUS code with a theoretical curve obtained by integrating an energy FP equation. Here we present the details of this calculation. The distribution function produced by elastic collisions between a beam and a homogeneous plasma with a uniform magnetic field, has azimuthal symmetry around the latter. Hence, the FP equation, Eq. (15), can be written in polar coordinates (v, λ) as follows [43]:

$$\frac{\partial f}{\partial t} = \frac{\partial f_v}{\partial t} + \frac{\partial f_\lambda}{\partial t} \tag{A.1}$$

where

$$\frac{\partial f_v}{\partial t} = -\frac{1}{v^2}\frac{\partial}{\partial v}\left[\left(v^2\frac{\partial h}{\partial v}+\frac{\partial g}{\partial v}\right)f\right] + \frac{1}{2v^2}\frac{\partial^2}{\partial v^2}\left(v^2\frac{\partial^2 g}{\partial v^2}f\right) \tag{A.2}$$

and

$$\frac{\partial f_\lambda}{\partial t} = \frac{1}{2v^3}\frac{\partial g}{\partial v}\frac{\partial}{\partial \lambda}\left[(1-\lambda^2)\frac{\partial f}{\partial \lambda}\right]. \tag{A.3}$$

Here $\lambda = \cos\theta$ is the pitch angle and $\theta = 0$ is the symmetry axis. The functions h and g are the Rosenbluth potentials [43] which, for a Maxwellian plasma reduce to [44]

$$h(v) = \sum_\beta \frac{\Gamma_\beta n_\beta}{v}\left(1+\frac{m}{m_\beta}\phi(x_\beta)\right) \tag{A.4}$$

$$g(v) = \sum_\beta \Gamma_\beta n_\beta v\left(\phi(x_\beta)+\frac{\phi(x_\beta)+x_\beta\phi\prime(x_\beta)}{2x_\beta^2}\right), \tag{A.5}$$

where $\Gamma_\beta = 4\pi q_\alpha q_\beta \ln \Lambda_\beta/m_\alpha^2, x_\beta = v/v_{s,\beta}$ with $v_{s,\beta} = \sqrt{2T_\beta/m_\beta}$, and $\phi(x)$ is the error function.

It is important to note that all the velocity and pitch moments reduce to

$$\frac{d}{dt}\langle v^n\rangle \equiv \int v^n\frac{\partial f}{\partial t}d^3v = \int v^n\frac{\partial f_v}{\partial t}d^3v \tag{A.6}$$

and

$$\frac{d}{dt}\langle \lambda^n\rangle \equiv \int \lambda^n\frac{\partial f}{\partial t}d^3v = \int \lambda^n\frac{\partial f_\lambda}{\partial t}d^3v. \tag{A.7}$$

Hence, the FP equation, Eq. (A.1), can be split up in a velocity contribution and a pitch contribution [45,44]. The velocity contribution can be re-written in the form

$$\frac{\partial f_v}{\partial t} = \frac{1}{v^2}\frac{\partial}{\partial v}\left[v^3\nu_E f_v\right] + \frac{1}{2v^2}\frac{\partial}{\partial v}\left[v^2\left(\sum_\beta \nu_{E,\beta}\frac{2T_\beta}{m}\right)\frac{\partial f_v}{\partial v}\right] \tag{A.8}$$

where

$$\nu_{E,\beta} = \frac{2\Gamma_\beta n_\beta}{v_{s,\beta}^3}\frac{m}{m_\beta}\frac{G(x_\beta)}{x_\beta} \tag{A.9}$$

and $\nu_E = \sum_\beta \nu_{E,\beta}$. The function $G(x)$ was defined in Eq. (27).

It is easy to see that, when the different species have the same temperature $T_\beta = T$, the stationary solution of this equation is the Maxwell–Boltzmann distribution function

$$f_v^{MB} = \frac{1}{\left(2\pi v_{th}^2\right)^{3/2}}e^{-v^2/2v_{th}^2}. \tag{A.10}$$

where $v_{th}^2 = T/m$. Then, the final (stationary) mean energy of the ensemble is

$$\langle E\rangle = \int \frac{1}{2}mv^2 f_v^{MB}\, d^3v = \frac{3}{2}T. \tag{A.11}$$

In addition, Eq. (A.8) is a 1D equation that can be easily solved numerically. Thus, we can obtain the mean energy of the ensemble as a function of time,

$$\langle E(t)\rangle = \int \frac{1}{2}mv^2 f_v(v,t)d^3v, \tag{A.12}$$

for the whole energy range.

Eq. (A.8) is also valid for an isotropic situation, where there is no preferred direction in space, like fusion born alpha particles. In particular, the well known stationary slowing down distribution, mentioned in Section 7.1, can be easily obtained by neglecting the second term in the RHS of Eq. (A.8) and taking into account an isotropic and mono-energetic source $S(v) = S\delta(v - v_0)$, and the usual approximations for the velocity, $v_{s,i} \ll v \ll v_{s,e}$, where i = D,T. Using these approximations the collision frequency can be re-written in the following way:

$$\nu_E = \nu_{E,e}(0)\left(1+\frac{v_c^3}{v^3}\right) \tag{A.13}$$

where

$$\nu_{E,e}(0) = \frac{4\Gamma_e n_e}{3\sqrt{\pi}}\frac{m_\alpha}{m_e}\frac{1}{v_{s,e}^3} \tag{A.14}$$

is the electron collision frequency in the limit $v \ll v_{s,e}$ (also given in Eq. (52)) and v_c is a critical velocity for which $\nu_{E,e} = \nu_{E,i}$, defined in Eq. (53). Then, the FP equation, Eq. (A.8), can be approximated by

$$\frac{\partial f_v}{\partial t} = \frac{1}{v^2}\frac{\partial}{\partial v}\left[(v^3+v_c^3)f_v\right] + \frac{S\delta(v-v_0)}{4\pi v^2}, \tag{A.15}$$

where the second term in the RHS is the isotropic source. Finally, the stationary solution, given by Eq. (51), is obtained by setting the LHS equal to zero and integrating in velocity space, with the condition that $f_v(v) = 0$ for $v > v_0$.

References

[1] J.L. Velasco, A. Bustos, F. Castejón, L.a. Fernández, V. Martin-Mayor, A. Tarancón, Comput. Phys. Comm. 183 (9) (2012) 1877–1883. http://dx.doi.org/10.1016/j.cpc.2012.04.005.

[2] G.J. Kramer, R.V. Budny, A. Bortolon, E.D. Fredrickson, G.Y. Fu, W.W. Heidbrink, R. Nazikian, E. Valeo, M.A. Van Zeeland, Plasma Phys. Control. Fusion 55 (2) (2013) 025013. http://dx.doi.org/10.1088/0741-3335/55/2/025013.

[3] E. Hirvijoki, O. Asunta, T. Koskela, T. Kurki-Suonio, J. Miettunen, S. Sipilä, A. Snicker, S. Äkäslompolo, Comput. Phys. Comm. 185 (4) (2014) 1310–1321. http://dx.doi.org/10.1016/j.cpc.2014.01.014.

[4] H.E. Ferrari, R. Farengo, A. Lifschitz, M. Firpo, Effect of ideal kink modes in particle redistribution, in: 44th EPS Conference on Plasma Physics, Belfast, UK., 2017, p2.180 (url: http://ocs.ciemat.es/EPS2017PAP/pdf/P2.180.pdf); H.E. Ferrari, R. Farengo, C.F. Clauser, The effect of a saturated kink on the dynamics of tungsten impurities in the plasma core, Phys. Plasmas (2018) (to be submitted).

[5] R.J. Akers, E. Verwichte, T.J. Martin, S.D. Pinches, R. Lake, GPGPU Monte Carlo calculation of Gyro-phase resolved fast ion and n-state resolved neutral deuterium distributions, in: 39th EPS Conference & 16th Int. Congress on Plasma Physics, 2012, p. P5.088. URL http://ocs.ciemat.es/epsicpp2012pap/pdf/P5.088.pdf.

[6] R.J. Akers, S. Äkäslompolo, B. Colling, J. Hess, Y. Liu, S.D. Pinches, K. Särkimäki, M. Singh, A. Turner, J. Varje, High fidelity simulations of fast ion power flux driven by 3D field perturbations on ITER, in: 26th IAEA Fusion Energy Conference, Kyoto, Japan, 2016, pp. TH/4–1.

[7] H. Ferrari, R. Farengo, Plasma Phys. Control. Fusion 49 (6) (2007) 713–727. http://dx.doi.org/10.1088/0741-3335/49/6/003.

[8] R. Farengo, H.E. Ferrari, M.-C.C. Firpo, P.L. Garcia-Martinez, A.F. Lifschitz, Plasma Phys. Control. Fusion 54 (2) (2012) 025007. http://dx.doi.org/10.1088/0741-3335/54/2/025007.

[9] R. Farengo, H. Ferrari, P. García-Martínez, M.-C. Firpo, W. Ettoumi, A. Lifschitz, Nucl. Fusion 53 (4) (2013) 043012. http://dx.doi.org/10.1088/0029-5515/53/4/043012.

[10] C.F. Clauser, R. Farengo, Phys. Plasmas 22 (12) (2015) 0–7. http://dx.doi.org/10.1063/1.4936875.

[11] C. Clauser, R. Farengo, Nucl. Fusion 57 (4) (2017) 046013. http://dx.doi.org/10.1088/1741-4326/aa5cdf.

[12] L. Lao, J. Ferron, R. Groebner, W. Howl, H. St. John, E. Strait, T. Taylor, Nucl. Fusion 30 (6) (1990) 1035–1049. http://dx.doi.org/10.1088/0029-5515/30/6/006.

[13] M. Abramowitz, I.A. Stegun (Eds.), HandBook of Mathematical Functions: With Formulas, Graphs and Mathematical Tables, Dover Pub., New York, 1695.

[14] S. Ichimaru, Basic Principles of Plasma Physics: A Statistical Approach, W. A. Benjamin, Inc., London, 1973.

[15] L. Spitzer, Physics of Fully Ionized Gases, Interscience Inc., New York, 1956.

[16] B.A. Trubnikov, Rev. Plasma Phys. 1 (1965) 105.

[17] T. Peter, J.J. Meyer-Ter-Vehn, Phys. Rev. A 43 (4) (1991) 1998–2014. http://dx.doi.org/10.1103/PhysRevA.43.1998.

[18] L. De Ferrariis, N.R. Arista, Phys. Rev. A 29 (4) (1984) 2145. http://dx.doi.org/10.1103/PhysRevA.29.2145.

[19] C.F. Clauser, N.R. Arista, Phys. Rev. E 97 (2) (2018) 023202. http://dx.doi.org/10.1103/PhysRevE.97.023202.

[20] D. Mosher, Phys. Fluids 18 (1975) 846. http://dx.doi.org/10.1063/1.861219.

[21] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, Berlin, 1992.

[22] B.I. Cohen, A.M. Dimits, A. Friedman, R.E. Caflisch, P.B.I. Cohen, A.M. Dimits, A. Friedman, R.E. Caflisch, I. Trans, P. Sci, B.I. Cohen, A.M. Dimits, A. Friedman, IEEE Trans. Plasma Sci. 38 (9) (2010) 2394–2406. http://dx.doi.org/10.1109/TPS.2010.2049589.

[23] M. Manssen, M. Weigel, A.K. Hartmann, Eur. Phys. J.: Spec. Top. 210 (1) (2012) 53–71. http://dx.doi.org/10.1140/epjst/e2012-01637-8. arXiv:1204.6193.

[24] J.K. Salmon, M.A. Moraes, R.O. Dror, D.E. Shaw, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis on - SC 11, no. 5, ACM Press, New York, New York, USA, 2011, p. 1. http://dx.doi.org/10.1145/2063384.2063405. URL http://dl.acm.org/citation.cfm?doid=2063384.2063405.

[25] IAEA AMDIS ALADDIN Database. URL https://www-amdis.iaea.org/ALADDIN/.

[26] T.J. Dolan, Fusion Research. Volume I - Principles, Pergamon Press Inc., New York, 1982.

[27] S. Fielding, P. Johnson, M. Forrest, D. Guilhem, G. Matthews, J. Nucl. Mater. 162–164 (C) (1989) 482–488. http://dx.doi.org/10.1016/0022-3115(89)90316-4.

[28] R.K. Janev, W.D. Langer, D.E. Post, K. Evans, Elementary Processes in Hydrogen-Helium Plasmas, Springer Berlin Heidelberg, Berlin, Heidelberg, 1987. http://dx.doi.org/10.1007/978-3-642-71935-6.

[29] R.K. Janev (Ed.), Atomic and Molecular Processes in Fusion Edge Plasmas, Springer US, Boston, MA, 1995. http://dx.doi.org/10.1007/978-1-4757-9319-2.

[30] V. Kotov, D. Reiter, A.S. Kukushkin, Numerical study of the ITER divertor plasma with the B2-EIRENE code package, Tech. rep. (2007). URL http://www.eirene.de/kotov_solps42_report.pdf.

[31] S. Wiesen, D. Reiter, V. Kotov, M. Baelmans, W. Dekeyser, A.S. Kukushkin, S.W. Lisgo, R.A. Pitts, V. Rozhansky, G. Saibene, I. Veselova, S. Voskoboynikov, J. Nucl. Mater. 463 (2015) 480–484. http://dx.doi.org/10.1016/j.jnucmat.2014.10.012.

[32] H. Pacher, A. Kukushkin, G. Pacher, V. Kotov, R. Pitts, D. Reiter, J. Nucl. Mater. 463 (2015) 591–595. http://dx.doi.org/10.1016/j.jnucmat.2014.11.104.

[33] J.P. Boris, R. Shanny, Proceedings of the Conference on the Numerical Simulation of Plasmas (4th), Naval Research Laboratory, Washington D.C., 1970.

[34] H. Qin, S. Zhang, J. Xiao, J. Liu, Y. Sun, W.M. Tang, Phys. Plasmas 20 (8) (2013) 084503. http://dx.doi.org/10.1063/1.4818428.

[35] J. Berland, C. Bogey, C. Bailly, Comput. & Fluids 35 (10) (2006) 1459–1463. http://dx.doi.org/10.1016/j.compfluid.2005.04.003.

[36] C. Estrada-Mila, J. Candy, R.E. Waltz, Phys. Plasmas 13 (11) (2006) 112303. http://dx.doi.org/10.1063/1.2364149. arXiv:1603.08343.

[37] H.-S.S. Bosch, G.M. Hale, Nucl. Fusion 32 (4) (1992) 611–631. http://dx.doi.org/10.1088/0029-5515/32/4/I07.

[38] M. Shimada, D.J. Campbell, V. Mukhovatov, M. Fujiwara, N. Kirneva, K. Lackner, M. Nagami, V.D. Pustovitov, N. Uckan, J. Wesley, N. Asakura, a.E.. Costley, a.J. Donné, E.J. Doyle, A. Fasoli, C. Gormezano, Y. Gribov, O. Gruber, T.C. Hender, W. Houlberg, S. Ide, Y. Kamada, A. Leonard, B. Lipschultz, A. Loarte, K. Miyamoto, V. Mukhovatov, T.H. Osborne, A. Polevoi, a.C. Sips, Nucl. Fusion 47 (6) (2007) S1–S17. http://dx.doi.org/10.1088/0029-5515/47/6/S01.

[39] K. Tani, M. Honda, T. Oikawa, K. Shinohara, Y. Kusama, T. Sugie, Nuclear Fusion 55 (5) (2015). http://dx.doi.org/10.1088/0029-5515/55/5/053010.

[40] A. Polevoi, A. Loarte, N. Hayashi, H. Kim, S. Kim, F. Koechl, A. Kukushkin, V. Leonov, S. Yu. Medvedev, M. Murakami, Y. Na, A. Pankin, J. Park, P. Snyder, J. Snipes, V. Zhogolev, Nucl. Fusion 55 (6) (2015) 063019. http://dx.doi.org/10.1088/0029-5515/55/6/063019.

[41] A. Snicker, S. Sipilä, T. Kurki-Suonio, Nucl. Fusion 52 (9) (2012) 094011. http://dx.doi.org/10.1088/0029-5515/52/9/094011.

[42] A. Snicker, O. Asunta, H. Ylitie, T. Kurki-Suonio, M. Schneider, S.D. Pinches, Nucl. Fusion 55 (6) (2015) 063023. http://dx.doi.org/10.1088/0029-5515/55/6/063023.

[43] M.N. Rosenbluth, W.M. MacDonald, D.L. Judd, Phys. Rev. 107 (1) (1957) 1–6. http://dx.doi.org/10.1103/PhysRev.107.1.

[44] T.S. Chen, A General Form of the Coulomb Scattering Operators for Monte Carlo Simulations and a Note on the Guiding Center Equations in Different Magnetic Coordinate Conventions, Tech. rep. Max Planck Institute for Plasma Physics (1988).

[45] A.H. Boozer, Phys. Fluids 24 (5) (1981) 851. http://dx.doi.org/10.1063/1.863445.