Real-Time Simulation of the NBI Fast-Ion Distribution

M. Weiland[1], R. Bilato[1], R. Dux[1], B. Geiger[1], A. Lebschy[1], F. A. A. Felici[2], R. Fischer[1], W. W. Heidbrink[3], W. Hu[4], D. Rittich[1], M. Scheffer[2], B. Sieglin[1], and M. A. Van Zeeland[5]

The ASDEX-Upgrade Team and EUROfusion-MST1 Team

[1]*Max-Planck-Institut für Plasmaphysik, Garching, Germany*
[2]*Eindhoven University of Technology, Eindhoven, Netherlands*
[3]*University of California Irvine, CA 92697, USA*
[4]*Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, Anhui, People's Republic of China*
[5]*General Atomics, San Diego, CA 92186, USA*
Corresponding Author: M. Weiland, markus.weiland@ipp.mpg.de

Knowledge of the fast-ion distribution arising from neutral beam injection (NBI) is important for transport analysis and magnetic equilibrium reconstruction. For sophisticated plasma control, which will be essential for the success of future fusion devices, it is very beneficial to know this distribution function already in real-time during the discharge. Then, the relevant quantities (e.g., heating profiles, current-drive etc.) can be fed to real-time transport and equilibrium codes like RAPTOR, which estimate kinetic and current density profiles in real-time. Beyond real-time applications, such fast models are essential for optimization problems, e.g., reactor design studies or discharge planning.

Several sophisticated models exist, that can calculate this beam ion distribution in good agreement with experimental data, such as the Monte Carlo code NUBEAM. The high accuracy of these codes has, however, to be paid with relatively intensive numerical efforts, which compromises their use in real-time applications. In this contribution, we present the novel code RABBIT (Rapid Analytically Based Beam Injection Tool). RABBIT currently takes ≈ 25 ms per time step, which is roughly a factor of 1000 faster than the NUBEAM code. The approximations needed to arrive at this goal are discussed. Benchmarks are carried out with the more accurate but also much slower NUBEAM code, indicating a good agreement.

Several applications of the model on different machines are carried out. RABBIT is run in real-time in the discharge control system of ASDEX-Upgrade to improve active plasma control. In addition, RABBIT is being used for accurate equilibrium reconstructions (with the IDE code) in between shots. This facilitates the development of advanced scenarios, where a fine-tuning of the q-profile is desired.

On DIII-D, RABBIT is foreseen to be used in experiments with the goal to demonstrate real-time control of Alfvén eigenmodes (AE). Here, the neutron rate prediction from RABBIT is compared to the measured neutron rate to detect appreciable fast-ion transport. In conjunction with direct AE detection with ECE diagnostics, when detrimental conditions are observed, countermeasures to stabilize AEs can be activated during the discharge. This could be of great importance for future fusion reactors, where strong AE activity is expected.

Published as a journal article in Nuclear Fusion
http://iopscience.iop.org/article/10.1088/1741-4326/ab1edd

REAL-TIME SIMULATION OF THE NBI FAST-ION DISTRIBUTION

M. WEILAND[1], R. BILATO[1], R. DUX[1], B. GEIGER[1], A. LEBSCHY[1], C. COLINS[2], F. FELICI[3], R. FISCHER[1], W. W. HEIDBRINK[4], W. HU[5], D. LUI[4], D. RITTICH[1], M. SCHEFFER[3], B. SIEGLIN[1], M. A. VAN ZEELAND[2], THE ASDEX UPGRADE[1] AND EUROFUSION MST1 TEAMS

[1]Max-Planck-Institut für Plasmaphysik, Garching, Germany
[2]General Atomics, San Diego, California, USA
[3]Technische Universiteit Eindhoven, Netherlands
[4]Department of Physics and Astronomy, University of California, Irvine, California, USA
[5]Institute of Plasma Physics, Chinese Academy of Sciences, Hefei, China

Email of corresponding author: markus.weiland@ipp.mpg.de

Abstract

The RABBIT code is presented, which is capable of computing the NBI fast-ion distribution in real-time. We discuss the approximations needed to arrive at this goal: A simplified beam geometry is used for calculating the beam attenuation. Finite-orbit-width effects are taken into account by an orbit average of the beam deposition. The time-dependent solution of the Fokker-Planck equation (2D in velocity) is then calculated based on analytic expressions. This code currently takes ≈25 ms per time step, which is roughly a factor of 1000 faster than the more sophisticated NUBEAM code. Nevertheless, good agreement between both codes is found in a comprehensive benchmark.

1. INTRODUCTION

Knowledge of the fast-ion distribution arising from neutral beam injection (NBI) is important for transport analysis and magnetic equilibrium reconstruction. For sophisticated real-time plasma control, which will be essential for the success of future fusion devices, this distribution function needs to be known already during the discharge. Then, the relevant quantities (e.g. heating profiles, current-drive etc.) can be coupled to real-time transport and equilibrium codes like RAPTOR [1, 2], which has already been implemented and tested in the discharge control systems of present-day machines like TCV and ASDEX Upgrade. Beyond real-time applications, such fast models are essential for optimization problems, e.g. reactor design studies or discharge planning [2].
The NBI fast-ion distribution f can be calculated by solving the kinetic equation:

$$\frac{\partial f}{\partial t} + (\vec{v}\cdot\nabla_{\vec{x}} f + \vec{a}\cdot\nabla_{\vec{v}} f) = \hat{C}(f) + \sigma \tag{1}$$

Here, the terms correspond, from right to left, to the source term (e.g. due to NBI), the collision operator (e.g. slowing down and pitch angle scattering), the orbit effects (where $\vec{a} = \frac{Ze}{m}(\vec{E} + \frac{\vec{v}}{c}\times\vec{B})$ is the acceleration due to the Lorentz force on ions with charge Ze, mass m, velocity vector $\vec{v}$ in an electric and magnetic field $\vec{E}$ and $\vec{B}$) and the partial time derivative. Several sophisticated models exist, that can calculate this beam ion distribution in good agreement with experimental data [3–5], such as the Monte-Carlo code NUBEAM [6, 7]. The high accuracy of these codes has, however, to be paid with relatively intensive numerical efforts, which compromises their use in real-time applications. Reduced models like PENCIL [8] and the NBI block in ASTRA [9, 10] are faster, but still not fast enough for real-time applications. In addition, they usually make quite strong approximations, e.g. PENCIL neglects finite-orbit-width effects entirely and calculates only steady-state solutions.
In this proceeding, we give a short overview of the novel code RABBIT (Rapid Analytical Based Beam Injection Tool) [11]. RABBIT currently takes ≈25 ms per time step, which is roughly a factor of 1000 faster than the NUBEAM code. This execution time is comparable with the energy confinement time (e.g. ≈100 ms on ASDEX Upgrade), which makes real-time applications possible. The approximations needed to arrive at this goal are discussed, following the terms of the kinetic equation (1) from the right to the left: A strongly simplified beam geometry is used for calculating the beam attenuation (section 2). The collisions are treated by an analytic solution of the steady-state Fokker-Planck equation in each radial cell, locally in the uniform-plasma limit (section 3). A correction for finite orbit width effects is included by an orbit average of the source term over the first fast-ion orbit (section 4). Further details of the code can be found in [11]. In the conclusions (section 5) we discuss several applications of the code.

2. BEAM ATTENUATION - SOURCE TERM

The calculation of the source term – the fast-ion birth rate – is equivalent to calculating the beam attenuation: Each neutral atom, which is lost from the beam through ionization, becomes a newly born fast ion. For the calculation

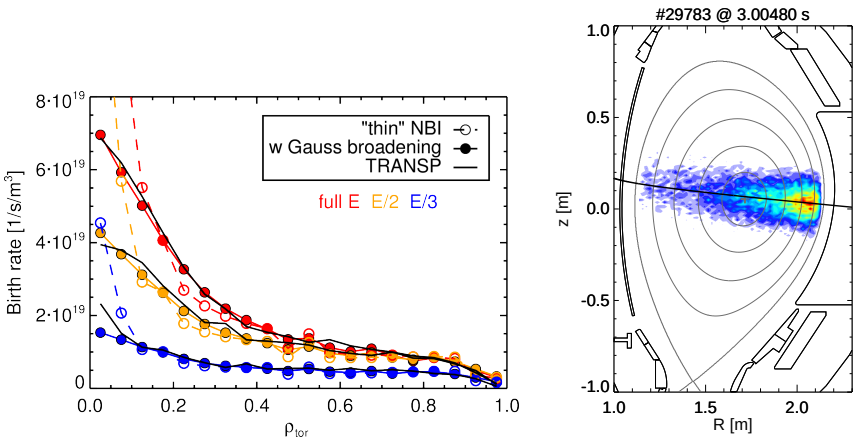


FIG. 1. Beam deposition. Left: Comparison of radial profiles. Right: 2D beam deposition calculated by NUBEAM (color scale) with the beam center-line.

of the beam attenuation, we use BESFM (beam emission forward-model), which was developed for the analysis of beam emission charge exchange spectra [12]. The code solves a collisional radiative model, including excitation and ionization reactions with electrons, main and impurity ions and charge exchange reactions with main and impurity ions. The code calculates the neutral density evolution, resolved for different atomic states, on straight beamlets. The geometry and number of beam-lets representing one source can be modified. For our purposes, we choose the simplest geometry, which is the thin center-line of the source.

As a result of BESFM, we get the flux Γ of neutrals along the thin center-line (coordinate l) of the NBI source (in [1/s]). The fast-ion birth (or deposition) rate can be calculated from the derivative along the line. To get the fast-ion birth-rate (in [1/s]) between two grid points separated by Δl, we simply take the difference $\tilde{S}_{\rm line}(l+\Delta l/2) = \Gamma(l+\Delta l)-\Gamma(l)$. We can get a birth profile S (in [1/m^3/s]) by rebinning the grid points along the line into a radial grid (e.g. in terms of $\rho_{\rm tor}$), and by dividing with the volume of each radial grid cell ($S(\rho_i) = \tilde{S}(\rho_i)/\Delta V_i$). If we would continue to consider only the thin center-line, this would lead to bad agreement with NUBEAM calculations. This is illustrated by the dashed line in fig. 1: In reality, a neutral beam has a quite significant broadening in the poloidal plane, perpendicular to the center-line. In the outer plasma, this is not so crucial for the birth profile, because the flux surfaces are tangential to this broadening. In the plasma center, however, poor agreement is observed with realistic NUBEAM calculations. E.g. in this case, the "thin beam" model shows an almost divergent behavior, because the considered NBI source (AUG Q3) lies very close to the magnetic axis.

In order to improve the birth-profile calculation, we need to consider the poloidal spreading of the beam. The geometry of our model is sketched in fig. 2: We assume a Gaussian spreading locally perpendicular to the beam-centerline. Its standard deviation $\sigma = \sigma(l)$ is allowed to be a function of the coordinate along the center-line l. We discuss now our beam-width model at a given point B (with given R,z coordinates $R_{\rm b}$, $z_{\rm b}$) on the center-line. In order to be fast, we assume locally circular flux surfaces: We calculate the geometric minor radius $r_{\rm b}$ and the radial flux coordinate ρ_b corresponding to ($R_{\rm b}$, $z_{\rm b}$) based on the realistic (numeric) equilibrium. For all other points in the poloidal plane (away from the beam center-line), we assume a linear relation $r(\rho) = \rho \cdot r_{\rm b}/\rho_{\rm b}$.

In that sense, it is important to use a flux coordinate ρ which behaves very similar to a radius, and we have chosen $\rho = \rho_{\rm tor}$ (the square root of normalized toroidal flux) for this reason. If we want to calculate the contribution of the birth rate in point B $\tilde{S}_{\rm line}(l_{\rm b})$ into a given radial cell ρ_i, we need to consider the crossing-points between the orange line (perp. to the beam) and the flux surface boundaries of that radial cell: h_{1l}, h_{2l}, h_{1u} and h_{2u}. Due to our assumption of circular flux surfaces, these crossing-points can be calculated analytically by elementary geometrical considerations. The contribution into the i-th radial cell ρ_i is then given by:

$$\tilde{S}(\rho_i) = \tilde{S}_{\rm line}(l_{\rm b}) \cdot w(\rho_i)/\textstyle\sum_i w(\rho_i) \qquad (2)$$

$$w(\rho_i) = \frac{1}{2}\left(\mathrm{erf}\,\frac{h_{2u}}{\sqrt{2}\sigma} - \mathrm{erf}\,\frac{h_{1u}}{\sqrt{2}\sigma}\right) + \frac{1}{2}\left(\mathrm{erf}\,\frac{h_{2l}}{\sqrt{2}\sigma} - \mathrm{erf}\,\frac{h_{1l}}{\sqrt{2}\sigma}\right) \qquad (3)$$

since the integral of our assumed Gaussian broadening is given by the error function.

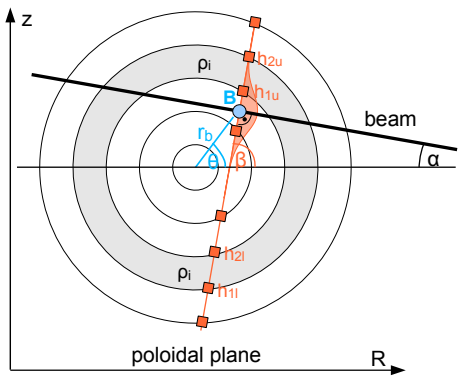


FIG. 2. Illustration of the analytic beam-width correction.

As discussed above, this beam-width correction is most important close to the plasma center. Here, the flux surfaces are also in realistic equilibria very close to a circular shape, which justifies our assumptions. Nevertheless, we do a correction for elongation of the flux surfaces. Elongation effectively means, that the length scales in fig. 2 are different along the vertical and horizontal axes. This means also that the length scales along the blue and orange line are different. We can take this into account by rescaling the beam standard deviation to an "effective" value (on each point on the grid along the beam). We use the following rescaling:

$$\sigma \to \sigma \cdot \sqrt{\frac{(a\cos\theta)^2 + (b\sin\theta)^2}{(a\cos\beta)^2 + (b\sin\beta)^2}} \tag{4}$$

This is motivated by the equation of an ellipse with the two axis a and b, defined on a given flux surface by:

$$a = \begin{cases} \max(R) - R_0 & \text{for: } R > R_0 \\ R_0 - \min(R) & \text{for: } R < R_0 \end{cases}; \qquad b = \frac{\max(z) - \min(z)}{2} \tag{5}$$

where R_0 is the major radius of the magnetic axis.
Fig. 1 shows the results of this beam width correction for the birth profile. We get a very good agreement to NUBEAM simulations, which do take a fully realistic NBI geometry into account with a computationally much more expensive Monte-Carlo approach.

3. THE FOKKER-PLANCK EQUATION AND ITS ANALYTIC SOLUTION

In this section we describe and review the analytic solution of the Fokker-Planck (FP) equation, which we solve in RABBIT. In order to make the kinetic equation 1 analytically solvable, we will neglect the orbit term in this section. A correction for orbit effects will be discussed later in section 4. Locally we assume an uniform plasma for the solution of the FP equation, namely the FP equation is solved on each radial cell with the local density and temperature, and independently from the other radial cells. The distribution function f depends then only on two velocity-space dimensions, the speed v and the pitch $\xi = v_\parallel/v$. The third dimension would be given by the gyro-angle, but f is supposed to be symmetric with respect to that. If we write out the collision term in the limit of (thermal electron velocity) $\gg$ (fast-ion velocity) $\gg$ (thermal ion velocity), we get the Fokker-Planck equation: [6]

$$\frac{1}{\tau_s v^2}\frac{\partial}{\partial v}[(v^3 + v_c^3)f] + \frac{\beta}{\tau_s}\frac{v_c^3}{v^3}\frac{\partial}{\partial \xi}(1-\xi^2)\frac{\partial f}{\partial \xi} + \frac{1}{\tau_s v^2}\frac{\partial}{\partial v}\left[\left(\frac{T_e}{m_{fi}}v^2 + \frac{T_i}{m_{fi}}\frac{v_c^3}{v}\right)\frac{\partial f}{\partial v}\right] = \frac{\partial f}{\partial t} - \sigma \tag{6}$$

with the Spitzer time:

$$\tau_s = 6.32 \cdot 10^8 \cdot \frac{A_{fi}}{Z_{fi}^2 \ln\Lambda_e} \cdot \frac{(T_e\,[\mathrm{eV}])^{3/2}}{n_e\,[\mathrm{cm}^{-3}]}\ \mathrm{s} \tag{7}$$

the critical velocity:

$$v_c = 5.33 \cdot 10^4 \cdot \sqrt{T_e\,[\mathrm{eV}]} \cdot \left\langle \frac{Z_i^2}{A_i} \right\rangle^{1/3} \mathrm{m/s} \tag{8}$$

and:

$$\beta = \frac{\left\langle Z_i^2 \right\rangle}{2\left\langle \frac{Z_i^2}{A_i} \right\rangle A_{fi}}, \quad \left\langle \frac{Z_i^2}{A_i} \right\rangle = \frac{\sum_i n_i (Z_i^2/A_i)\ln\Lambda_i}{n_e \ln\Lambda_e}, \quad \left\langle Z_i^2 \right\rangle = \frac{\sum_i n_i Z_i^2 \ln\Lambda_i}{n_e \ln\Lambda_e} \tag{9}$$

where the sums with index i go over the ion species. The subscript fi stands for the fast-ion (i.e. beam) species. $\ln\Lambda_e$ and $\ln\Lambda_i$ denote the Coulomb logarithms for fast-ion collisions with the electrons and the i-th ion species, respectively. We calculate them with the same formulas that are also implemented in NUBEAM (as described in [11]).
Finally, σ is the NBI source term in the FP equation. We parametrize it as

$$\sigma = \frac{S}{2\pi v^2}\delta(v - v_0)K(\xi) \tag{10}$$

which assumes a mono-energetic injection velocity v_0. The typically three energy components of a beam can be modeled by summation, then. A broad pitch distribution $K(\xi)$ (e.g. due to the broadening of the beam) can be taken into account. The normalization is defined such, that $\int K(\xi)\mathrm{d}\xi = 1$ and thus $\iiint \sigma \mathrm{d}^3\vec{v} = S$. Hence S is the fast-ion birth (or deposition) per volume and time (e.g. in units of $[1/\mathrm{m}^3/\mathrm{s}]$).

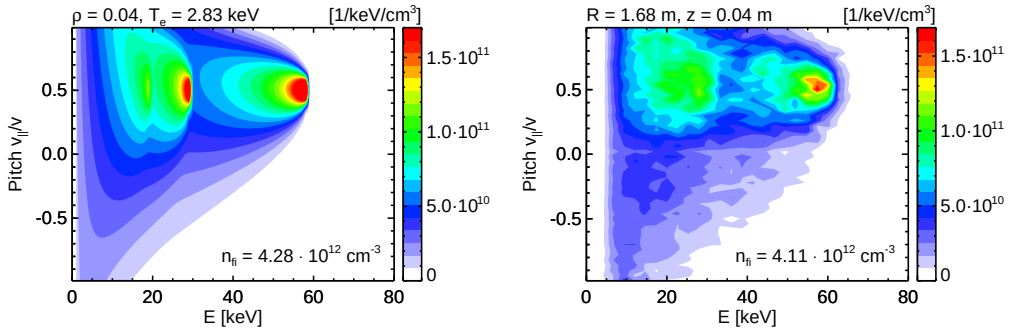


FIG. 3. Left: Solution of the Fokker-Planck equation (6) using three injection energies. Right: Calculated TRANSP fast-ion velocity distribution in the plasma center.

The steady-state solution to the FP equation for such a source term has been found in [13, 14]. Since Legendre polynomials $P_l(\xi)$ are eigenfunctions of the pitch-angle scattering operator, the solution is given as a series of Legendre polynomials:

$$f(v,\xi) = \frac{1}{2\pi}\frac{S\cdot\tau_s}{v^3+v_c^3}\cdot\sum_{l=0}^{\infty}\left(l+\tfrac{1}{2}\right)u^{l(l+1)}P_l(\xi)K_l\cdot H(v_0-v) \tag{11}$$

with:

$$K_l = \int K(\xi)P_l(\xi)\,\mathrm{d}\xi, \quad u = \left(\frac{v_0^3+v_c^3}{v^3+v_c^3}\frac{v^3}{v_0^3}\right)^{\beta/3} \tag{12}$$

and H being the Heaviside step function. Fig. 3(a) shows an example of this solution in the plasma core. We have calculated the solutions for the full, half and third energy component of the NBI independently of each other, and added them together. This is possible because in our model (and also in NUBEAM), collisions between fast ions are neglected. 3(b) shows the result of a NUBEAM calculation as comparison and the shape of the solutions agree very well.

In the end we are interested in integrals of the distribution function (e.g. fast-ion current, pressure, heating profiles etc.). Many of these integrals can be solved analytically, which is ideal for real-time calculations. For example, the fast-ion density is given by:

$$n_{\mathrm{fi}} = \int_0^{v_0}\int_{-1}^{+1} f\cdot 2\pi v^2\mathrm{d}v\mathrm{d}\xi = \frac{S\tau_s}{3}\ln\left(\frac{v_0^3+v_c^3}{v_c^3}\right) \tag{13}$$

In these integrals, the orthogonality relation of the Legendre polynomials is exploited:

$$\int_{-1}^{1} P_n(\xi)P_m(\xi)\,\mathrm{d}\xi = \frac{2}{2n+1}\delta_{nm} \tag{14}$$

Since $P_0(\xi)=1$, we can interprete the above integration over ξ as $\int \mathrm{d}\xi = \int P_0(\xi)\mathrm{d}\xi$, which lets only survive the $l=0$ term of the infinite sum. A subsequent integration over v leads then to the final result.

The first few Legendre polynomials are given by:

$$P_0(\xi)=1,\ P_1(\xi)=\xi,\ P_2(\xi)=\frac{1}{2}(3\xi^2-1),\ \ldots \tag{15}$$

In general, $P_l(\xi)$ is a polynomial of order l. Since the integrals of interest contain usually only low powers of ξ (e.g. the fast-ion current $\propto \iint v\xi f\cdot 2\pi v^2$), we need in fact only very few summands of the infinite sum in eq. 11 - e.g. RABBIT considers currently only $l=0,1$.

For a first comparison with NUBEAM, we consider the fast-ion density and heating to electron and ions. The results are shown by the dashed lines in fig. 9. While the overall agreement is reasonable, the profile shapes do not yet agree very well. All three profiles show an under-estimation in the plasma core and an over-estimation at the plasma edge. These deviations stem from the orbit effects, which we have neglected so far and which we will take into account in the next section.

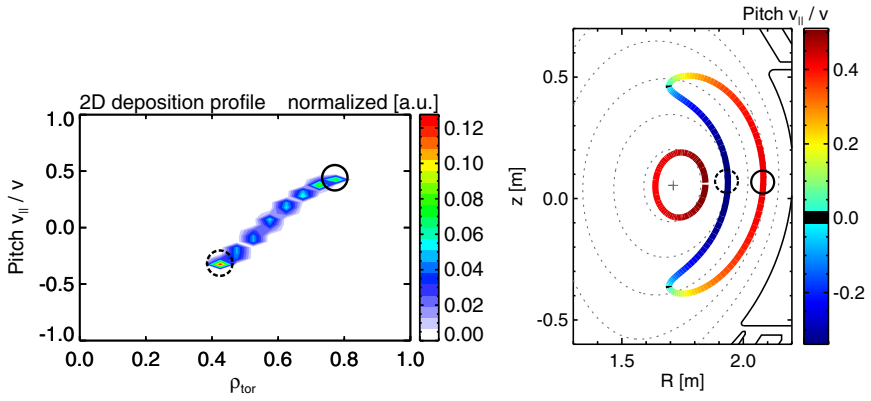


FIG. 4. Left: 2D Deposition profiles $\tilde{S}(\rho_{tor}, \xi)$ for an individual fast-ion ($\xi = v_\parallel/v$). It is calculated by averaging over the first orbit, shown on the right. In both plots, the birth position is indicated by the full circle.

4. ORBIT-AVERAGE OF THE SOURCE TERM

Solving the full kinetic equation including the orbit term is a much more difficult task, and analytically not possible. This is the reason, why many codes such as NUBEAM rely on Monte-Carlo methods. Here, the source term is represented by a set of Monte-Carlo markers. For each marker, the orbit trajectory is calculated and during each orbit step, a Monte-Carlo collision operator is applied (changing the velocity vector). In a real-time application, such a complete treatment is not possible. We therefore have to consider an ad-hoc correction for these orbit effects.

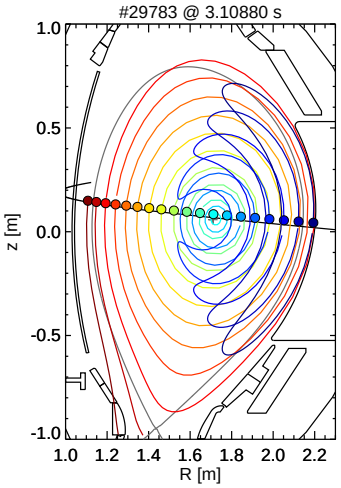
A way to do this is to average the NBI source term over the first (unperturbed) orbit. This is illustrated in fig. 4 for a trapped fast-ion. Up to now, we have only considered the birth position and assumed that the ion stays on that initial flux surface. In reality, the ion travels towards other flux surfaces, and thereby also changes its pitch. In the concrete example, which is typical for co-current NBI, the birth position (marked with a circle) is almost on the furthest outside part of the orbit, and the ion moves further inside on the orbit. This leads effectively to a broadened birth distribution of this individual ion, which is broadened towards the plasma center with respect to its initial position. We can take this into account by mapping the orbit into a 2D deposition function $\tilde{S}(\rho, \xi)$, where the weight for each bin is given by the fraction of time the ion spends in the given bin (relative to the duration of a full poloidal orbit turn). If an orbit is unconfined (i.e. promptly lost), $\tilde{S}(\rho, \xi)$ is set to zero. In order to use $\tilde{S}(\rho, \xi)$ in our analytic solution of the FP equation, we further decompose it into Legendre polynomials according to eq. (12) (again we need only very few moments, currently $l = 0, 1$).

The brute force approach to calculate such an orbit average for the whole NBI source would be then to take a MC representation of the source, calculate such a 2D deposition function (i.e. an orbit) for every marker, and sum over all markers [15]. To get reasonable statistics, $\approx$ 5000 markers are needed. Using a conventional 4th order Runge-Kutta integrator, roughly 1 s would be needed to calculate these orbits, which is clearly too slow for real-time applications.

To overcome this one could, in principle, either use approximation formulas for the orbits [16] or try to reduce the number of necessary orbits. We initially tried the first approach, which however gave not fully satisfactory results. Although there are quite precise approximation formulas for e.g. the orbit widths, it is still difficult to estimate the 2D deposition function, i.e. the radial profile and the pitch evolution. In addition, a MC approach does not work well with our approach for calculating the beam attenuation along a single line (the beam center-line).

Ultimately, we decided to implement the second approach, by calculating only very few orbits and doing an appropriate interpolation in between. Fig. 5 shows an example of all (19) guiding center orbits calculated for the highest energy component. We calculate only orbits along the beam attenuation grid (i.e. the beam centerline), and even here we calculate an orbit only for every n-th grid point. The guiding center starting position is slightly elevated from the beam-centerline because we take the Larmor radius into account. In doing so, we need to calculate $3 \cdot 19 = 57$ orbits per injector. This is possible with a conventional 4th order Runge-Kutta integrator in ≈ 10 ms (without parallelization) and hence fast enough for real-time applications. In order to get the deposition function also for the grid-points along the center-line in between (e.g. at l_b), we need to do an interpolation, as illustrated in fig. 6. Assuming that we have calculated the orbit deposition functions $\tilde{S}_1$ and $\tilde{S}_2$ for l_1 and l_2 with $l_1 < l_b < l_2$ we use:

$$\tilde{S}_b(\rho) = \tilde{S}_1(\rho - (\rho_b - \rho_1)) \cdot \frac{l_2 - l_b}{l_2 - l_1} + \tilde{S}_2(\rho - (\rho_b - \rho_2)) \cdot \frac{l_b - l_1}{l_2 - l_1} \tag{16}$$



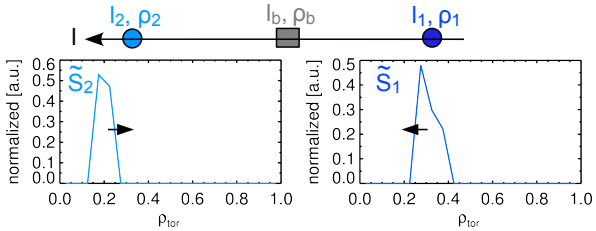


FIG. 5. (left) All calculated orbits for the orbit-average of the highest energy-component source term. The guiding-center birth position is shown with a dot in the same color as the corresponding orbit.

FIG. 6. (top) Illustration of the interpolation of the deposition function according to eq. (16). The two blue points correspond to the same colored points in fig. 5 (they are the two passing orbits closest to the low field side).

This corresponds to a shift of $\tilde{S}_1$ and $\tilde{S}_2$ to the flux coordinate ρ_b of the new location and a subsequent linear interpolation with respect to the beam center-line coordinate l. The shift ensures, that the width of $\tilde{S}_\mathrm{b}$ is equal to an interpolation of the widths of $\tilde{S}_1$ and $\tilde{S}_2$, whereas a pure interpolation without shifting would lead to an over-estimation of the width of $\tilde{S}_\mathrm{b}$.

With this method, we are able to calculate the orbit averaging for ions born on the beam center-line. Now, we have to make this compatible with our beam-width correction (introduced in section 2). This is illustrated in fig. 7. We start on the center-line at point b, which is in the radial cell ρ_{ref}. The beam-width correction is applied on the orange straight line perpendicular to the beam center-line and we need now to extrapolate the orbit distribution function from ρ_ref to other radial cells ρ_i along the orange line. We do this in a similar fashion as for the interpolation technique: We shift the distribution function from ρ_ref to the new position ρ_i:

$$\tilde{S}_i(\rho) = \tilde{S}_b(\rho - (\rho_i - \rho_\mathrm{ref})) \tag{17}$$

This is based on the assumption, that the orbits at ρ_i and ρ_ref are similar: same width (in terms of ρ), same topology, same pitches and same starting position (e.g. in terms of poloidal angle θ). Most of these assumptions should be fulfilled quite well. Topology changes may occur at the trapped-passing boundary, and with this method we effectively "smooth" over this boundary, which is done in reality by collisions anyway. The last assumption is well fulfilled in the outer plasma, since the width of the beam imposes only small changes of θ here. In the plasma center, this assumption may be not fulfilled well, for example if the beam center-line approaches closely the magnetic axis as in the example which we have considered earlier in fig. 1. Since the orbit widths are usually small close to the magnetic axis, this should however be tolerable.

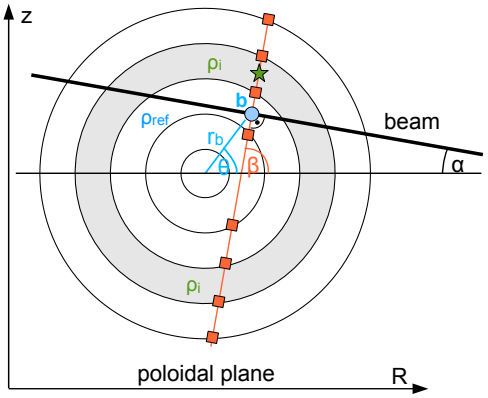


FIG. 7. Illustration of the combination of orbit-average and beam-width correction: Orbits are only calculated along the center-line (Point b in radial cell ρ_{ref}), and are extra-polated perpendicular to the center-line (=orange line) according to eq. (17).

In order to test the accuracy of the RABBIT-orbit averaging technique, we compare it in fig. 8 to a conventional Monte-Carlo orbit average of a FIDASIM [17, 18] birth distribution as described in section 2 of [15]. We get very good agreement in the birth profile, and also a good agreement in the average pitch (which is the first Legendre component). Also, the RABBIT-orbit average produces smoother profiles (due to the interpolations), while the MC profiles contain Monte-Carlo noise. At the same time, the RABBIT-orbit average is orders of magnitude faster than the Monte-Carlo approach, since it requires only ≈ 60 orbit calculations compared to the 5000 orbit calculations in the MC calculation.

As final step, we plug the orbit-averaged source term in the FP equation and calculate the profiles (figure 9). The orbit averaging leads to a very good agreement with the profiles from NUBEAM. Slight deviations remain in the plasma center, affecting only a small fraction of the plasma volume. In fig. 10 we compute the volume-integrated heating power to electrons and ions. It is interesting to note, that the orbit average has also an effect on these volume-integrated quantities and improves the agreement with NUBEAM. This is due to the (mostly) linear dependence of the critical energy on T_e: Since we consider co-current NBI in this exampe, the orbits shift

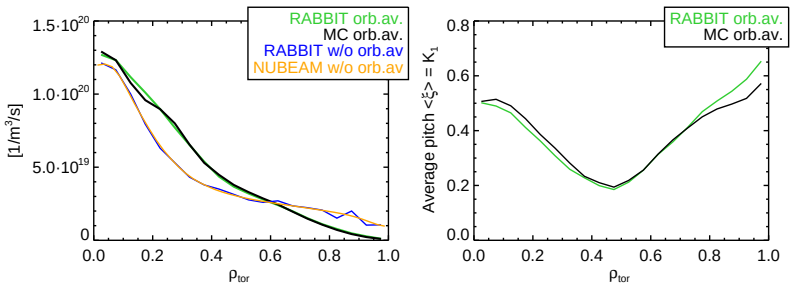
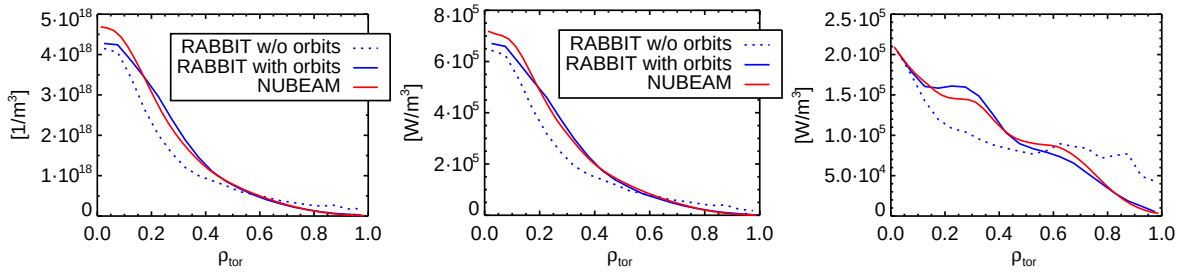


FIG. 8. Comparison of the RABBIT orbit-average with a Monte-Carlo orbit-average as described in [15]. The MC source markers for the latter are calculated with the FIDASIM [17, 18] code, which includes a fully realistic 3D beam geometry. Left: Birth profile S summed over all energy components. Right: Average pitch K_1 for the full energy component.



(a) Fast-ion density **(b)** Heating to ions **(c)** Heating to electrons

FIG. 9. Calculated profiles from RABBIT with and without orbit-averaging of the source, in comparison to NUBEAM results. This case is based on the AUG discharge #29783 at 3.11 s.

the fast-ion distribution further inwards - i.e. away from the cold plasma edge (where the critical energy is lower and electron heating more dominant) towards the hot plasma core (where the critical energy is higher and electron heating is thus weaker).

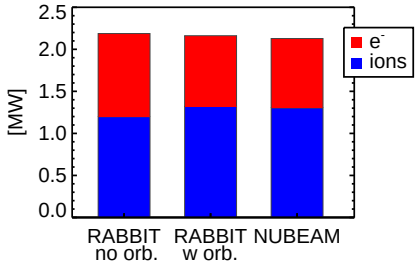


FIG. 10. Comparison of volume-integrated heating power to electrons and ions.

5. CONCLUSION AND OUTLOOK

The RABBIT code consists of three parts: In the first part, the beam attenuation and fast-ion birth profile is calculated. We use a realistic collisional radiative model resolved for atomic states, which is evaluated in a simplified beam geometry (only along the center-line of the beam). The effect of finite beam-width is included analytically. In the second part, the orbit average of the source term is calculated. This is done with a conventional Runge-Kutta integrator using arbitrary numerical axisymmetric equilibria. To speed up the calculation, only a very low number of orbits are calculated ($\approx$ 20 per beam energy component) and an optimized interpolation is used in between. The third part calculates the solution of the Fokker-Planck equation. In this proceeding we have described the steady-state solution, the analytic treatment of the time-dependence is described in [11]. Since the FP-solution is calculated analytically, this part is clearly the fastest part, while the other two parts take approximately 12 ms each. In total, the RABBIT code needs $\approx$25 ms per time-step and beam (serial), which is roughly a factor of 1000 faster than NUBEAM. The calculation time could be further reduced, e.g. by faster hardware, further code optimizations or higher parallelization. The reported 25 ms have been estimated on an Intel Xeon E5-2680 v3 CPU with 2.5 GHz clockrate, using a parallelizing scheme with one thread per beam. A higher parallelization could, e.g., be achieved trivially by using one thread per beam energy component. A benchmark is carried out with the more accurate NUBEAM code, indicating very good agreements for heating profiles as well as profiles of fast-ion density.

Several applications of the code are planned and carried out. An interface with the discharge control system of ASDEX Upgrade and with the RAPTOR code has been created, and it will allow an improved real-time control of the plasma discharges in the upcoming 2018 experimental campaign of ASDEX Upgrade. Here, the kinetic input profiles will be provided by estimates from RAPTOR, which are constrained by the real-time diagnostics

(when available) and complemented by model-based estimates. Currently, real-time T_e and n_e diagnostics are used, and there are plans to make the charge-exchange diagnostics (T_i and rotation) real-time capable as well. The plasma equilibrium is reconstructed in real-time by the JANET code [19]. An extension of these activities to other tokamaks such as JET and TCV is already in preparation.

RABBIT is also well suited for non-real-time applications, where fast models are desired. For example, it can be used for fast intershot analysis which are needed to facilitate the decision-making in the control room. To this end, RABBIT is being used for accurate equilibrium reconstructions at ASDEX Upgrade (AUG) with the IDE code [20, 21], which allow sophisticated estimates of the safety factor (q) profile directly after a discharge. This will aid the development of advanced scenarios, where a fine-tuning of the q-profile is desired. Another application is the ASDEX Upgrade flight simulator [22, 23], which is currently being developed to check planned discharges prior to their execution, e.g. if they will meet the experimental goals or to ensure that machine limits are not exceeded. It uses the ASTRA code as physical model for AUG, and RABBIT has been coupled to provide the NBI-related inputs. In addition, RABBIT could also be used for scenario optimization or reactor design studies, where the short calculation time will allow large parameter scans.

On DIII-D, RABBIT is forseen to be used in experiments with the goal to demonstrate real-time control of Alfvén eigenmodes (AE). To this end, RABBIT has been extended to calculate neutron rates, which can be compared to the measured neutron rate to detect appreciable fast-ion transport. In conjunction with direct AE detection with ECE diagnostics, when detrimental conditions are observed, counter-measures to stabilize AEs can be activated during the discharge. The RABBIT neutron rate prediction has already been tested during a range of DIII-D discharges for intershot analysis. A comparison with TRANSP was later carried out for the whole database of shots and indicated very good agreement between the codes. With RABBIT essentially the same information is available much faster, which allows to detect fast-ion confinement degradation directly after the discharge and to include this information for planning of the subsequent discharge.

Several extensions of the code are possible in the future. The inclusion of charge-exchange losses could be relevant for smaller machines such as TCV. In addition, radial transport could be included within our time-dependence numerical scheme, which could allow to model radial transport, e.g. due to MHD activity or plasma turbulence.

ACKNOWLEDGMENTS

This work was partly performed within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

REFERENCES

[1] Felici F. et al., *Nuclear Fusion*, **51** (2011) 083052.
[2] Felici F. et al., *Plasma Physics and Controlled Fusion*, **54** (2012) 025002.
[3] Salewski M. et al., *Nuclear Fusion*, **50** (2010) 035012.
[4] Weiland M. et al., *Plasma Physics and Controlled Fusion*, **58** (2016) 025012.
[5] Weiland M. et al., *Nuclear Fusion*, **57** (2017) 116058.
[6] Goldston R. et al., *Journal of computational physics*, **43** (1981) 61.
[7] Pankin A. et al., *Computer Physics Communications*, **159** (2004) 157.
[8] Challis C. et al., *Nuclear Fusion*, **29** (1989) 563.
[9] Polevoi A. et al., *Benchmarking of the NBI block in ASTRA code versus the OFMC calculations*, Japan Atomic Energy Research Institute, 1997.
[10] Pereverzev G. V. et al., ASTRA. Automated System for TRansport Analysis in a tokamak, (2002).
[11] Weiland M. et al., *Nuclear Fusion*, **58** (2018) 082032.
[12] Lebschy A., Master thesis, LMU München, 2014.
[13] Cordey J. G. et al., *The Physics of Fluids*, **17** (1974) 1626.
[14] Gaffey J. D., *Journal of Plasma Physics*, **16** (1976) 149.
[15] Weiland M., PhD thesis, LMU München, 2016.
[16] Feng Y. et al., *Computer physics communications*, **88** (1995) 161.
[17] Heidbrink W. et al., *Comm. Comp. Physics*, **10** (2011) 716.
[18] Geiger B., PhD Thesis, LMU München, 2013, also IPP report 10/46.
[19] Giannone L. et al., *Fusion Engineering and Design*, **100** (2015) 519 .
[20] Fischer R. et al., *40th EPS Conference on Plasma Physics*, (2013).
[21] Fischer R. et al., *Fusion Science and Technology*, **69** (2016) 526.
[22] Treutterer W. et al., *Concepts of the new ASDEX Upgrade flight simulator*, SOFT 2018.
[23] Janky F. et al., *AUG flight simulator development*, SOFT 2018.