



**HAL**  
open science

# Transport and losses of energetic particles in tokamaks in the presence of Alfvén activity using the new full orbit TAPaS code coupled to FAR3d

H Betar, David Zarzoso, Jacobo Varela, Diego Del-Castillo-Negrete, Luis  
Garcia

## ► To cite this version:

H Betar, David Zarzoso, Jacobo Varela, Diego Del-Castillo-Negrete, Luis Garcia. Transport and losses of energetic particles in tokamaks in the presence of Alfvén activity using the new full orbit TAPaS code coupled to FAR3d. 2024. hal-04541528v1

**HAL Id: hal-04541528**

**<https://hal.science/hal-04541528v1>**

Preprint submitted on 10 Apr 2024 (v1), last revised 15 Oct 2024 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Transport and losses of energetic particles in tokamaks in the presence of Alfvén activity using the new full orbit TAPAS code coupled to FAR3d

**Betar, H.**

Aix-Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340 Marseille, France

E-mail: homam.BETAR@univ-amu.fr

**Zarzoso, D.**

Aix-Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340 Marseille, France

**Varela, J.**

Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain

**Del-Castillo-Negrete, D.**

Oak Ridge National Laboratory, Oak Ridge, TN 37831-8071, United States of America

**Garcia, L.**

Universidad Carlos III de Madrid, 28911 Leganes, Madrid, Spain

**Du, X.**

General Atomics, P.O. Box 85608, San Diego, California 92186-5608, USA

**Abstract.** Recent developments and tools integrated into the TAPAS code are presented, enabling realistic scenario simulations of particle dynamics within experimental tokamak magnetic equilibria. In particular, the enhanced capabilities of TAPAS enable seamless coupling with external simulations, provided the metric and equilibrium magnetic field of the external code are known. Coupling TAPAS with the gyro-fluid code FAR3d, the transport and losses of energetic particles (EPs) in the presence Alfvén eigenmodes (AEs) in DIII-D plasma discharge #159243 were investigated using simple initialization of energetic particles in phase space. Detailed analyses of prompt losses in the presence of collisions (and without it) were performed. Then, further analysis that involves including the total electromagnetic perturbations resulting from AEs activities was performed. The results indicate that, for the energies and the initial conditions considered here, the presence of AEs enhances the particle losses at the end of the different simulations conducted in this study.

## 1. Introduction

In current and future fusion devices, energetic particles (EPs) play a crucial role. These particles are generated either externally from sources such as neutral beam injection (NBI) and ion cyclotron resonance heating (ICRH), or internally through nuclear fusion reactions, resulting in 3.5 MeV alpha particles. It is imperative to adequately confine EPs to facilitate the transfer of their energy to the thermal population via Coulomb collisions, thereby sustaining fusion reactions. However, the presence of EPs in burning plasmas can induce instabilities, specifically EP-driven instabilities, which can have a detrimental impact on overall energy confinement.

EP-driven instabilities involve various phenomena, such as the fishbone instability initially observed in the PDX tokamak [1], later analyzed in DIII-D plasma [2], and shear Alfvén eigenmodes destabilized by super-Alfvénic EPs. These modes can lead to major EP losses towards the plasma-facing components, as demonstrated for energetic beam ions and byproducts of deuterium-deuterium fusion reactions in the DIII-D tokamak [3, 4]. Shear Alfvén waves usually experience strong continuum damping in slab and cylindrical geometries in the presence of non-uniform magnetic field, with the exception of global Alfvén eigenmodes (GAE) in cylindrical geometry arising below the minimum value in the Alfvén continuum [5, 6, 7, 8]. However, in realistic toroidal geometry with noncircular flux surfaces, gaps in the Alfvén continuum can be formed, leading to the existence of eigenmodes that are not damped by the continuum. Those include toroidicity induced Alfvén eigenmodes (TAE) [9, 10], ellipticity induced Alfvén eigenmodes (EAE) [11, 12], noncircularity induced Alfvén eigenmodes (NAE) [13, 14], reversed-shear Alfvén eigenmodes (RSAE) [15, 16], and beta-induced Alfvén eigenmodes (BAE) [17] -for more details, see Refs. [18, 19] and references therein. Since modes inside these gaps are not damped by the continuum, any external drive at the corresponding frequencies can provide enough energy to trigger an instability. More specifically, the motion of EPs can resonate with these modes and the gradients of the EP distribution function in phase-space can result in their destabilization. This is why they can be excited by alpha particles resulting from fusion reactions or by EPs generated through NBI or ICRH -see Refs [20, 21, 22, 23, 24] and references therein. Moreover, the presence of stochastic orbits resulting from turbulence and/or the onset of other MHD or kinetic instabilities [25, 26, 27], such as tearing modes [28], ion-temperature gradients (ITG) [29], or other interacting phenomena [30], can significantly alter the transport and losses of EPs.

The dynamics of particles and more precisely EPs can be numerically analyzed following two approaches. The first one implies solving in a self-consistent way the equations for the EPs, the background plasma and the electromagnetic fields. This can be accomplished using gyro-fluid models with corrections to take into account the effects of EPs (see [31, 32, 16, 33, 34] and references therein), hybrid-MHD models, where the background plasma is described using fluid equations and the effect of EPs is included by means of a gyro-kinetic [35, 36] or a kinetic [?], and fully gyro-kinetic

models [37, 38], where all the species are evolved using the gyro-kinetic formalism. The second approach involves the use of passive-tracers codes, where the time-dependent electromagnetic perturbations are externally imposed and the trajectories of EPs are integrated but do not impact the evolution of the perturbations. Some of these codes are ASCOT [39], LOCUST [40] or KORC [41]. TAPAS code [42] falls in this category and can be easily coupled to solvers that evolve the dynamic of the background plasma. The paper focuses on detailing the new updates implemented in TAPAS, and the application of the new version of the code to conduct in-depth investigation of EP transport and losses by considering the full-orbit trajectory of particles in the presence of arbitrary 3D electromagnetic perturbations and experimental equilibria. In particular, the emphasis is put on the impact of Alfvén Eigenmodes on the transport and losses of energetic particles in DIII-D tokamak. In that context, EP generated by NBI are considered, with energies of the order of  $\sim 20$  keV and the electromagnetic perturbation is obtained by means of an external code (FAR3d [33, 43] in this work).

The remainder of the paper is structured as follows. Section 2 details the last upgrades that have been implemented in TAPAS code, namely the full-orbit integration, the use of generalized coordinates and magnetic geometry, the inclusion of collisions, the initialization in 5D when considering a full-orbit approach and the general workflow. In section 2.6, we discuss some numerical issues encountered next to the last magnetic surface and near the magnetic axis. Section 3 is devoted to presenting the results of the simulations using FAR3d code, which provides the electromagnetic perturbations to be used in TAPAS in section 4. Finally, we give a summary and directions for future research in section 5.

## 2. The new TAPAS code

The Toroidal Accelerated PArticle Simulator (TAPAS) code has undergone significant updates since its initial version [42]. The previous version only integrated trajectories in a 5-dimensional guiding-center coordinate system, whereas the current version now possesses the capability to track particle dynamics in a 6-dimensional phase space  $(\mathbf{x}, \mathbf{v})$ , where  $\mathbf{x} = (x, y, z)$  and  $\mathbf{v} = (v_x, v_y, v_z)$  represent the position and velocity components of the particle in the phase space, respectively. This enhancement allows for the resolution of the particle gyro-motion, enabling the code to capture phenomena occurring on timescales comparable to the gyro-motion time and on spatial scales smaller than the particle Larmor radius.

Moreover, unlike the initial version of TAPAS, which assumed simplified concentric and circular flux surfaces for numerical convenience and analytical solution for electromagnetic perturbation, the updated version is capable of handling realistic scenarios. This includes both 2D axisymmetric equilibria found in tokamak devices and 3D equilibria present in stellarator devices. Consequently, magnetic equilibria constructed using equilibrium codes such as VMEC [44, 45], DESC [46], and SIESTA [47] can be seamlessly incorporated into TAPAS by reading the equilibrium fields from

an external file. In this manner, the code accommodates a broader range of magnetic equilibria.

Equilibrium codes and non-linear simulations employed to study plasma dynamics in toroidal devices, such as FAR3d, often solve the relevant equations in Boozer coordinates [48, 49]. However, the full-orbit version of TAPAS solves Newton's equations for each particle in a Cartesian coordinate system. Therefore, coordinate transformations are applied to convert data between different coordinate systems utilized by various codes. This allows the integration of TAPAS with other codes that employ generalized toroidal coordinates, enabling comprehensive investigations of plasma dynamics.

### *2.1. Integration of the equations of motion with the Boris algorithm*

Assuming no collisions, the equations solved by the full-orbit version of TAPAS are

$$\frac{dx_i}{dt} = v_i \tag{1}$$

$$m_s \frac{dv_i}{dt} = eZ_s (E_i + \epsilon_{ijk} v_j B_k) \tag{2}$$

where  $e$  the elementary charge,  $Z_s$  and  $m_s$  the atomic number and the mass of the particle, respectively,  $x_i = x, y, z$ , and  $v_i = v_x, v_y, v_z$  are the position and velocity components of the particle, and  $\epsilon_{ijk}$  is the Levi-Civita symbol.  $E_i$  and  $B_i$  are the  $i^{th}$  components of the electric and magnetic fields at the particle position obtained, in this paper, using FAR3d code. It should be noted that in the Cartesian coordinate system, as used in TAPAS to integrate the previous equations, there is no distinction between co-variant and contra-variant representations due to the orthogonality property of this coordinate system. Therefore, the equations of motion presented in Eqs. (1) and (2) do not differentiate between the two. Furthermore, the non-relativistic Newton equations are employed in TAPAS due to the assumption that the velocities of particles under consideration are much smaller than the speed of light. This assumption holds for most phenomena observed in toroidal devices, except for the case of runaway electrons, where relativistic effects must be taken into account as done in KORC [41].

The Boris method can be employed to numerically solve the equations of motion. It is a second-order accurate leapfrog scheme generally used owing to its excellent long-term accuracy and its ability to effectively resolve the gyro-motion of particles. One notable advantage of the Boris method is its energy conservation property, particularly in cases where only magnetic fields are present. It also exhibits good energy conservation even in more general scenarios involving both electric and magnetic fields. This attractive feature stems from its ability to conserve the phase space volume, making the Boris algorithm a standard choice for particle pushing in particle-in-cell (PIC) codes utilized in solving the Boltzmann-Maxwell system of equations in plasma physics [50]. It is to be noted that both electric and magnetic fields can be obtained from any external code

in any general coordinate system. Hence, before updating the position and velocity of the particle, one must transform these fields to the Cartesian coordinate system.

To understand this scheme, let us begin by discretizing Eqs.(1) and (2) in time using a central stencil

$$\frac{x_i^{n+1} - x_i^n}{\Delta t} = v_i^{n+\frac{1}{2}} \quad (3)$$

$$\frac{v_i^{n+\frac{1}{2}} - v_i^{n-\frac{1}{2}}}{\Delta t} = eZ_s E_i(\mathbf{x}^n) + eZ_s \epsilon_{ijk} \frac{v_j^{n+\frac{1}{2}} + v_j^{n-\frac{1}{2}}}{2} B_k(\mathbf{x}^n) \quad (4)$$

where  $\Delta t$  is the time step. Eqs.(1) and (2) are approximated at times  $t_{n+1/2}$  and  $t_n$ , respectively. It is worth noting that in the Boris method, the electromagnetic fields are evaluated at the particle position at time  $t_n$ .

The Boris method, along with its variations such as the Vay [51] and Higuera-Cary methods [52, 53], focuses on finding accurate approximations for the velocity at time  $t_n$  in the right-hand side of Eq.(2). In the Boris scheme, the velocity is taken to be the average of its values at  $t_{n+1/2}$  and  $t_{n-1/2}$ . The electric field in Eq.(4) can be eliminated by splitting the calculation into three steps: two acceleration steps due to the electric field and one pure rotation due to the magnetic field. This is achieved by defining  $v_j^{n-\frac{1}{2}}$  and  $v_j^{n+\frac{1}{2}}$  as follows

$$v_j^{n-\frac{1}{2}} = v_j^- - \frac{eZ_s E_j \Delta t}{m}, \quad v_j^{n+\frac{1}{2}} = v_j^+ + \frac{eZ_s E_j \Delta t}{m} \quad (5)$$

which leads, after substituting in Eq.(4), to

$$v_j^+ = v_j^- + \epsilon_{jmn} \tilde{v}_m s_n \quad (6)$$

where

$$\tilde{v}_m = v^- + \epsilon_{mnl} v_n^- Q_l, \quad Q_l = \frac{eZ_s B_l \Delta t}{m}, \quad s_n = \frac{2Q_n}{1+Q^2} \quad (7)$$

In the presence of both electric and magnetic fields, the Boris method consists of the following steps:

- (i) First half of electric acceleration, i.e. find  $v_j^-$  by solving the first equation of (5), where  $v_j^{n-\frac{1}{2}}$  is known from the previous time steps.
- (ii) Rotation step due to magnetic field. This step involves calculating  $v_j^+$  using Eq.(6), where  $\tilde{v}_m$  and  $s_n$  are obtained from Eq.(7).
- (iii) Second half of electric acceleration which involves finding  $v_j^{n+\frac{1}{2}}$  using the second equation of (5), where electric field is estimated at the position of the particle at  $t_n$ .

## 2.2. Collision operator

Collisions are introduced in TAPAS by means of an operator acting either on particle velocity (for the full-orbit version) or on guiding-centre velocity (for guiding-centre

version). In both cases, the collision operator is decomposed into stochastic and deterministic operators, which model the diffusion and the Coulomb drag, respectively, for test particles colliding with background particles of mass  $m_b$ , charge  $eZ_b$ , density  $n_b$  and temperature  $T_b$ . For the case of background electrons, we have  $b = e$ , whereas for background ions  $b = i$ . For the full-orbit version, the equations governing the evolution of the velocity in the presence of collisions are

$$d\mathbf{v}_{\text{coll,det}} = -\nu_s dt \quad (8)$$

$$d\mathbf{v}_{\text{coll,stoch}} = \left[ \sqrt{D_{\parallel}} \frac{\mathbf{v} \otimes \mathbf{v}}{v^2} + \sqrt{D_{\perp}} \left( \mathbf{I} - \frac{\mathbf{v} \otimes \mathbf{v}}{v^2} \right) \right] \cdot d\mathbf{W}^{\mathbf{v}} \quad (9)$$

where  $\mathbf{W}^{\mathbf{v}}$  is a Wiener process in velocity space and  $\mathbf{I}$  is the unit tensor in 3D.

In the deterministic part of the collision operator,  $\nu_s$  represents the Coulomb drag given by the expression

$$\nu_s = \sum_b 4n_b C_b \left( \frac{m_b}{2k_B T_b} \right)^{3/2} \frac{\psi(x)}{x} \quad (10)$$

with  $k_B$  the Boltzmann constant,  $C_b$  given by the expression

$$C_b = \frac{e^2 Z_s^2 e^2 Z_b^2 \log \Lambda}{8\pi \epsilon_0^2 m_s^2} \quad (11)$$

the variable  $x$  is defined by

$$x = \frac{v}{\sqrt{T_b / (2m_b)}} \quad (12)$$

and the function  $\psi$  is defined as

$$\psi(x) = \frac{\Phi(x) - x\Phi'(x)}{2x^2} \quad (13)$$

with  $\Phi$  the error function

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du \quad (14)$$

In the stochastic part,  $D_{\parallel}$  and  $D_{\perp}$  are the diffusion coefficients, calculated as follows

$$D_{\parallel} = \sum_b 4n_b C_b \left( \frac{m_b}{2k_B T_b} \right)^{1/2} \frac{\psi(x)}{x} \quad (15)$$

$$D_{\perp} = \sum_b 2n_b C_b \left( \frac{m_b}{2k_B T_b} \right)^{1/2} \frac{\Phi(x) - \psi(x)}{x} \quad (16)$$

The deterministic part is solved after each update of the velocity in the Boris algorithm, using the same strategy as the one used in [41] for the radiative term of runaway electrons. In practice, the deterministic equation is decomposed into the effect of the Lorentz force  $d\mathbf{v}_{\text{L,det}} = eZ_s/m_s (\mathbf{E} + \mathbf{v} \times \mathbf{B})$  and the effect of collisions  $d\mathbf{v}_{\text{coll,det}} = -\nu_s dt$ . At each time step  $i$ , it is computed the velocity due to the Lorentz

force at time step  $i + 1$ . The result of this computation is called  $\mathbf{v}_L^{i+1}$ . The effect of the Coulomb drag is then computed as follows

$$\frac{\mathbf{v}_{\text{coll,det}}^{i+1} - \mathbf{v}^i}{\Delta t} = \nu_s(\mathbf{x}^{i+1/2}, \mathbf{v}^{i+1/2}) \mathbf{v}^{i+1/2} \quad (17)$$

with  $\mathbf{v}^{i+1/2} = (\mathbf{v}_L^{i+1} + \mathbf{v}^i) / 2$ . The new velocity is then computed as

$$\mathbf{v}^{i+1} = \mathbf{v}_L^{i+1} + \mathbf{v}_{\text{coll,det}}^{i+1} - \mathbf{v}^i \quad (18)$$

The stochastic part is solved using the Euler scheme, with a time step larger than the one used for the integration of the deterministic equations in order to reduce the computational time. The Wiener process is generated taking into account that it must have independent and Gaussian increments. Therefore, the increment  $d\mathbf{W}^v$  is generated using a Box-Muller algorithm for each direction  $(x, y, z)$ .

The collision operator has been tested initializing an ensemble of  $10^6$  particles at the same position in real space with velocities localized around  $v_{\parallel} = 2$  and  $v_{\perp} = 2$ . The initialization in velocity space is illustrated in Fig.1a. The colormap represents the distribution function computed from the initial particles and the dashed lines represent for the reference the Maxwellian distribution function. After  $\omega_{c,0}t = 10^7$ , the distribution function is recomputed. The result is illustrated in Fig.1b. A very good agreement is obtained. Such agreement becomes evident when plotting the distribution function for a given value of  $\mu$  and for a given value of  $v_{\parallel}$ , which is shown in figures 1c and 1d, respectively.

### 2.3. Initialization of particles

With respect to the previous version of the code, particles are not any longer described by their weights. This is to avoid numerical noise when computing macroscopic quantities such as density, temperature or fluxes. Therefore, all particles have the same statistical weight. Nonetheless, in order to account for the initialization following distribution functions in 6D, the possibility exists to initialize particles non uniformly in phase space by means of the Metropolis-Hastings algorithm based on the Markov chain Monte-Carlo method [54, 55, 56]. However, in some cases it is useful to initialize particles in specific regions or uniformly in velocity and/or real spaces. In that case, the Metropolis-Hastings algorithm is not employed and particles are simply initialized in an equidistant grid. The initialization in real space occurs always in toroidal coordinates, namely  $(\rho, \theta, \varphi)$ , with  $\rho$  a generalized coordinate with length dimensions,  $\theta$  the intrinsic poloidal angle and  $\varphi$  the toroidal angle. The initialization in velocity space occurs in 3D using the cartesian coordinates  $(v_x, v_y, v_z)$ . This choice is sometimes convenient for specific studies. However, it does not provide enough physical insight. For this reason, another possibility is to initialize particles in 5D using the initialization that was already implemented in the previous version of TAPAS, depending on the set of selected couple of variables in velocity space:  $(v_{\parallel}, \mu)$ ,  $(E, \Lambda)$ ,  $(E, \lambda)$  and  $(E, P_{\varphi})$ , where  $E$  is the

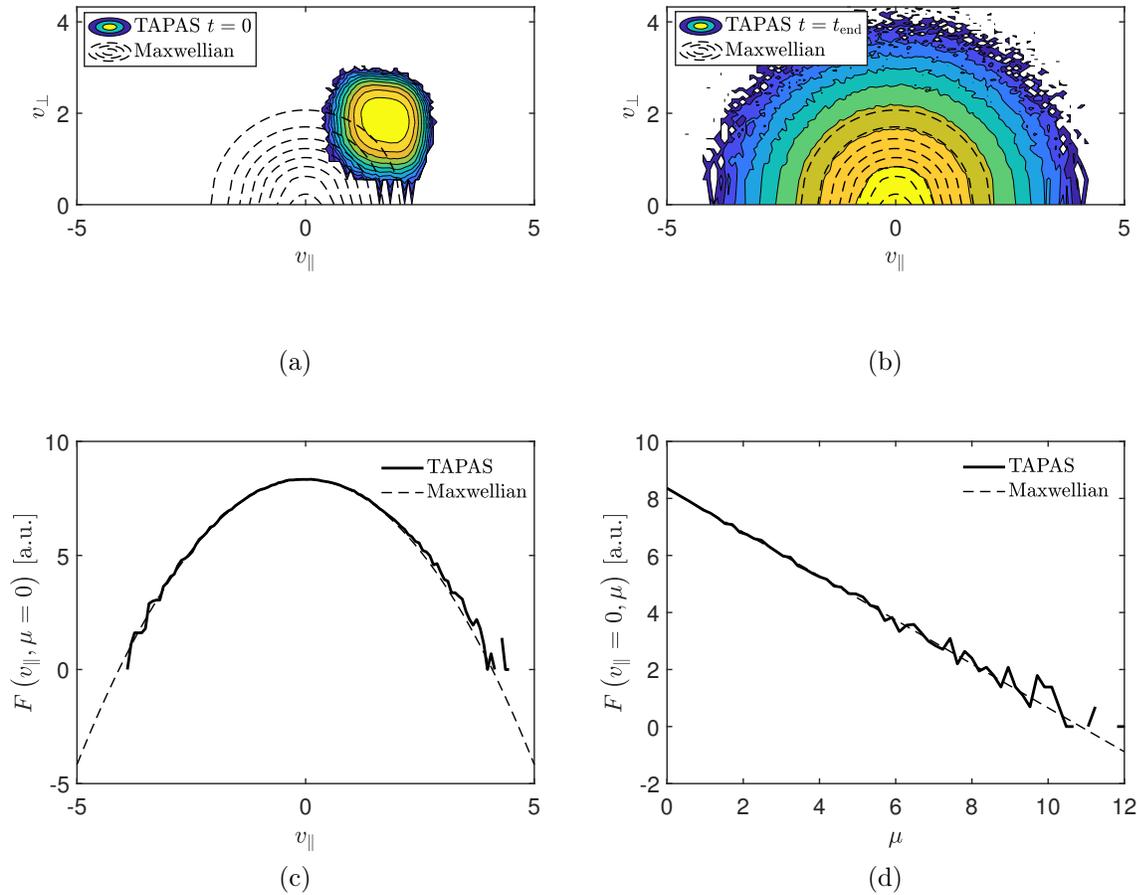


Figure 1: Initial (top left) and final (top right) distribution function in  $(v_{\parallel}, v_{\perp})$  computed from TAPAS in the presence of collisions. As a reference, the Maxwellian distribution is provided using contour plot. Also, in the same figure, the final distribution function for a given value of  $\mu$  (bottom left) and for a given value of  $v_{\parallel}$  (bottom right) are given as a function of  $v_{\parallel}$  and  $\mu$ , respectively. The Maxwellian distribution is given using dash line.

kinetic energy,  $\Lambda = \mu B_0/E$  and  $\lambda = v_{\parallel}/v$ . For each of these choices, the computation of  $(v_{\parallel}, \mu)$  is done first (this step is obviously not required when particles are initialized using parallel velocity and magnetic moment). Then, a gyro-phase  $\varphi_c$  is selected using a uniform distribution function  $\mathcal{U}_{[0,2\pi[}(\varphi_c)$ . The unit vector in the direction of the magnetic field is computed at the real position  $(\rho, \theta, \varphi)$  where the particle is initialized and then transformed into cartesian coordinates  $\mathbf{b} = b_x \mathbf{e}_x + b_y \mathbf{e}_y + b_z \mathbf{e}_z$ . The angle between  $\mathbf{b}$  and  $\mathbf{e}_z$  is computed and called  $\alpha_b$ . Afterwards, a rotation of the basis  $\{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$  in

order to align  $\mathbf{e}_z$  with  $\mathbf{b}$  is performed. The components of the new basis are simply

$$\hat{\mathbf{e}}_x = \mathbf{R}_{11}\mathbf{e}_x + \mathbf{R}_{21}\mathbf{e}_y + \mathbf{R}_{31}\mathbf{e}_z \quad (19)$$

$$\hat{\mathbf{e}}_y = \mathbf{R}_{12}\mathbf{e}_x + \mathbf{R}_{22}\mathbf{e}_y + \mathbf{R}_{32}\mathbf{e}_z \quad (20)$$

$$\hat{\mathbf{e}}_z = \mathbf{b} \quad (21)$$

with  $\mathbf{R}$  the rotation matrix

$$\mathbf{R} = \begin{bmatrix} b_y^2(1 - \cos \alpha_b) + \cos \alpha_b & -b_y \cdot b_x \cdot (1 - \cos \alpha_b) & b_x \sin \alpha_b \\ -b_y \cdot b_x \cdot (1 - \cos \alpha_b) & b_x^2(1 - \cos \alpha_b) + \cos \alpha_b & b_y \cdot \sin \alpha_b \\ b_x \sin \alpha_b & b_y \cdot \sin \alpha_b & \cos \alpha_b \end{bmatrix} \quad (22)$$

In the new basis, the vectors  $\hat{\mathbf{e}}_x$  and  $\hat{\mathbf{e}}_y$  constitute a basis of the plane perpendicular to the magnetic field at the position  $(\rho, \theta, \varphi)$  and therefore, the perpendicular velocity can be expressed as

$$\mathbf{v}_\perp = v_\perp (\cos \varphi_c \hat{\mathbf{e}}_x + \sin \varphi_c \hat{\mathbf{e}}_y) \quad (23)$$

Taking into account that the vector expression of the parallel velocity is  $v_\parallel \mathbf{b}$ , the initial velocity in cartesian coordinates reads

$$v_x = v_\parallel b_x + v_\perp (\cos \varphi_c \mathbf{R}_{11} + \sin \varphi_c \mathbf{R}_{12}) \quad (24)$$

$$v_y = v_\parallel b_y + v_\perp (\cos \varphi_c \mathbf{R}_{21} + \sin \varphi_c \mathbf{R}_{22}) \quad (25)$$

$$v_z = v_\parallel b_z + v_\perp (\cos \varphi_c \mathbf{R}_{31} + \sin \varphi_c \mathbf{R}_{32}) \quad (26)$$

#### 2.4. Generalized toroidal coordinates and coordinate transformations in TAPAS

TAPAS makes use of generalized coordinates  $(x^1, x^2, x^3)$ . For the case of toroidal coordinates,  $x^1$  is usually a distance-like label for the magnetic surface,  $x^2$  represents the position in the poloidal direction and  $x^3$  represents the position in the toroidal direction. In a simplified geometry where flux surfaces are reduced to nested circular and concentric torii,  $x^1$  would represent the minor radius or the distance from the surface to the magnetic axis. However, one could also use a cylindrical coordinate system, in which case  $x^1 = R$ ,  $x^2 = \varphi$  and  $x^3 = Z$ , or even a Cartesian coordinate system, where  $x^1 = x$ ,  $x^2 = y$  and  $x^3 = z$ . The choice of the coordinate system will depend on the physics to be analyzed. In practice, this will be given by the external code that provides the 3D electromagnetic perturbations. In that sense, TAPAS has been developed in the most general way so that the choice of the coordinates used by the external code (FAR3d in the present work) is transparent to TAPAS and everything is handled by modules applying the required transformations between three coordinate systems

$$(x, y, z) \longleftrightarrow (R, Z, \varphi) \longleftrightarrow (x^1, x^2, x^3) \quad (27)$$

In the present work, TAPAS is coupled to FAR3d, where Boozer coordinates are used. Throughout the remainder of the paper,  $(\rho, \theta, \zeta)$  refer to this Boozer coordinate system.

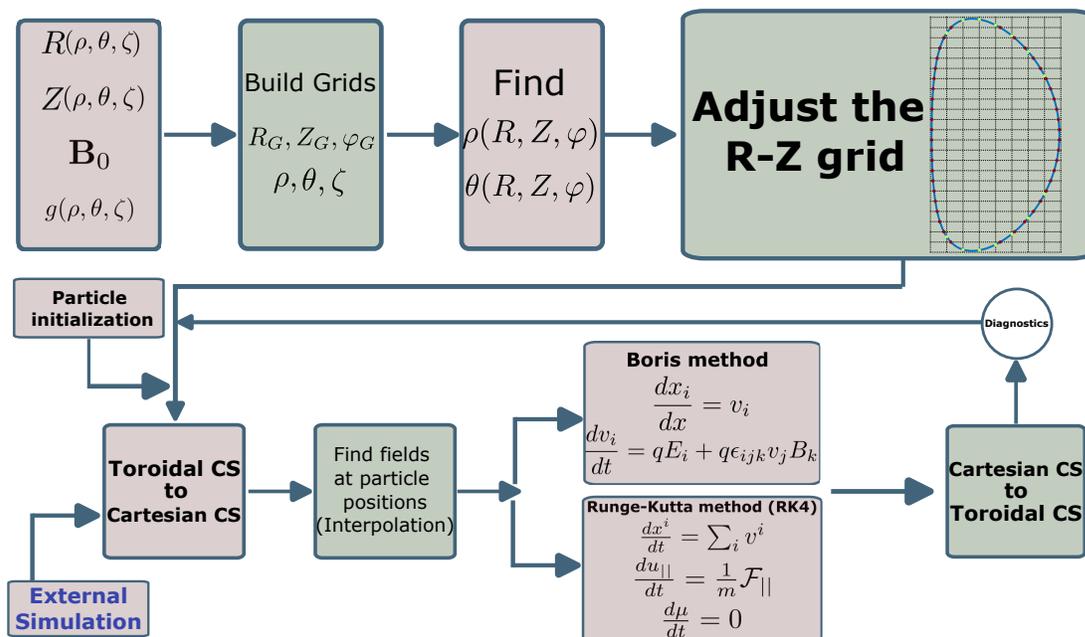


Figure 2: An illustration of the main steps implemented in TAPAS code.

In TAPAS, the equations of motion are solved in Cartesian coordinates. However, to accurately update the coordinates of the particle in the generalized toroidal coordinate system, suitable coordinate transformations are required. For this purpose, TAPAS provides two grids: the toroidal grid and the cylindrical grid. The toroidal grid is used to interpolate the fields at the particle position in toroidal geometry, while the cylindrical grid is employed to transform the particle coordinates from Cartesian to toroidal coordinates after pushing it in time with the Boris algorithm. Details of these transformations are given in [Appendix A](#). Defining the toroidal grid in TAPAS is straightforward, as it involves specifying the number of points and the intervals in the different directions. On the other hand, the cylindrical grid needs to be defined in a manner that ensures an accurate approximation of the toroidal coordinates at its nodes. Equilibrium codes, such as VMEC, are used to compute the flux surfaces represented by  $R = R(\rho, \theta)$  and  $Z = Z(\rho, \theta)$  (assuming 2D equilibria). These flux surfaces can then be inverted to determine the toroidal coordinates  $\rho(R, Z)$  and  $\theta(R, Z)$  in the cylindrical grid using the Newton-Raphson method.

### 2.5. TAPAS workflow

TAPAS is developed in FORTRAN and employs a hybrid MPI-OpenACC paradigm for parallelization across multi-GPU nodes. The code utilizes MPI for domain decomposition across nodes and OpenACC directives to offload computations onto GPU accelerators within each node. LAPACK and parallel HDF5 libraries are required for TAPAS. LAPACK is essential to solve eigenvalue problems, including the calculation of parallel magnetic potential in certain cases. Meanwhile, parallel HDF5 libraries facilitate

input/output operations. The primary steps in TAPAS can be summarized as follows:

- (i) Initialization of parallel environment.
- (ii) Reading input file.
- (iii) The simulation first checks if it is a restart. If it is indeed a restart, the code proceeds to read an HDF5 file containing all the necessary data. However, if it is not a restart, the simulation initializes itself based on the provided input parameters.
- (iv) Calculating all the required quantities to solve Eqs. (1-2).

- If the magnetic equilibrium and electromagnetic perturbations are obtained using another code, such as FAR3d, an external HDF5 file containing this information is read. The HDF5 file includes the parametric form of the magnetic surfaces given in generalized coordinates, represented by  $R(\rho, \theta, \zeta)$  and  $Z(\rho, \theta, \zeta)$ . Additionally, the file contains the contra-variant components of the equilibrium magnetic field. In the cylindrical coordinate system, denoted by  $(R, Z, \varphi)$ , while in the Boozer coordinate system, denoted by  $\rho, \theta, \zeta$ , as described in the next section.

If the magnetic equilibrium and all the required quantities are computed within TAPAS, such as in the case of circular or analytically derived geometries, the values of  $R(\rho, \theta, \zeta)$  and  $Z(\rho, \theta, \zeta)$  are determined by applying the relevant analytical formulas.

- In order to perform various calculations and coordinate transformations, TAPAS computes the contra- and covariant metrics ( $g^{ij}$  and  $g_{ij}$ ), the Jacobian of the transformations, and the gradients of different quantities. Additionally, depending on the chosen equilibrium, TAPAS determines the functions  $\rho(R, Z)$  and  $\theta(R, Z)$ . These functions are crucial for coordinate transformations from Cartesian coordinate system to Boozer coordinate system. By knowing the Cartesian coordinates  $(x_p, y_p, z_p)$  of a particle, one can easily obtain its cylindrical coordinates  $(R_p, Z_p, \varphi_p)$ . Subsequently, by interpolating the values of  $\rho(R, Z)$  and  $\theta(R, Z)$  at the corresponding cylindrical coordinates  $(R_p, \varphi_p, Z_p)$ , the particle's Boozer coordinates  $(\rho_p, \theta_p, \zeta_p)$  can be determined.
- In the case where the equilibrium is obtained from an external data source, TAPAS undertakes a crucial task of identifying the intersection points between the cylindrical grid lines, represented by  $R$  and  $Z$ , and the last available magnetic surface. These intersection points are essential for the subsequent computations.

Then, a corrective algorithm is employed to address a specific concern arising from points that lie in proximity to the last available closed surface but reside outside the physical domain. This algorithm ensures the proper handling of such points, mitigating any potential issues and preserving the integrity of the simulation. The challenge of determining the coordinates  $(\rho_p, \theta_p, \zeta_p)$  within the cylindrical coordinate system will be discussed in a dedicated sub-section (see Section 2.6). This sub-section will provide a detailed analysis of the methods

employed to address this challenge, presenting the intricacies involved and offering solutions to overcome the associated complexities.

- When the equilibrium is read from an external file, TAPAS implements an algorithm to address interpolation issues that may arise near the last available closed surface and the magnetic axis. To ensure accurate trajectory calculations for charged particles in close proximity to this surface, slight adjustments are made to the  $\rho(R, Z)$  and  $\theta(R, Z)$  functions, as will be discussed in the next section.
- (v) The main loop is started.
- If the used version is the full orbit one, the following steps are applied for each particle:
    - The electromagnetic fields are estimated at the particle positions by applying the recently implemented 3D bspline interpolation.
    - The particle coordinates are transformed to Cartesian CS.
    - Eq.(3) is solved as discussed in Section 2.1.
    - The particle coordinates are transformed back to Boozer CS to test if the particle remains inside the toroidal device.
    - If the particle crossed the last available magnetic surface, it is marked as lost, and its trajectory is no longer integrated. However, in the gyro-kinetic version of the code where circular or analytically derived equilibria were used, the code allows for the possibility of following the particle dynamics outside the toroidal device.
    - The particle’s velocity is updated by solving Eq.(4) with Boris method.
    - Magnetic moment and the parallel velocity are calculate for the particle.
    - Every  $n_{\text{diag}}$ , the chosen 1D, 2D, and 3D diagnostics are applied and stored in HDF5 file.
  - Otherwise, the guiding-centre algorithm, detailed in [42], is applied.

(vi) At the end of the main loop, a restart file is produced.

The main steps implemented in TAPAS are schematically indicated in Fig.2.

### *2.6. Numerical challenges faced in TAPAS*

Interpolation plays a crucial role in TAPAS as it is frequently applied to estimate fields at particle positions in the toroidal coordinate system or determine particle toroidal coordinates after solving Eqs (1)-(2) in the Cartesian coordinate system.

Two interpolation methods available in the current extended version of the code are Lagrange and B-spline interpolation. Lagrange interpolation is well-suited for interpolating quantities in 1D and 2D, while B-splines offer greater stability and accuracy in 3D. Lagrange interpolation can exhibit oscillations between data points, particularly when using high-degree polynomials or closely spaced data (known as the Runge phenomenon), which becomes more pronounced in 3D cases. In contrast, B-spline interpolation tends to be more stable and less prone to oscillations owing to the

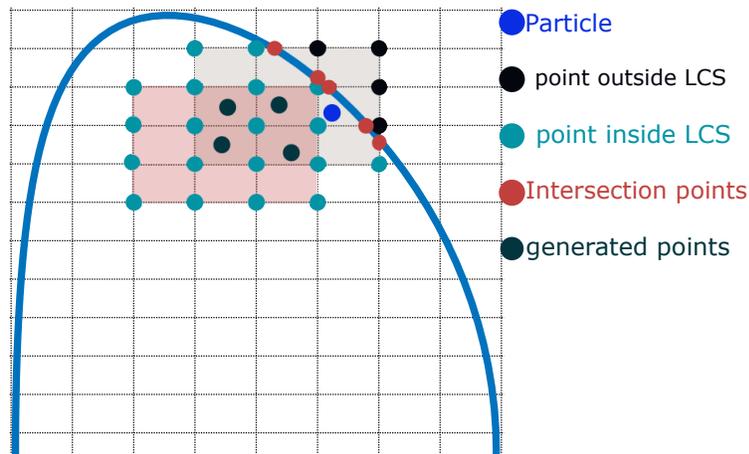


Figure 3: The upper part of a poloidal cross-section of an axisymmetric equilibrium. The blue curve represents the last available surface, while the mauve-shaded region corresponds to the complete interpolation rectangle and the gray-shaded region represents a boundary interpolation rectangle.

local support of the basis functions, which allows for better control of the influence of nearby data points. Both methods can be extended to higher-dimensional spaces ( $nD$ ), but they come with a significant computational cost as they involve tensorial products of 1D interpolation functions.

However, the specific implementation of B-spline interpolation in TAPAS is more computationally and resource-intensive compared to Lagrange interpolation. This is because, in the case of B-splines, an additional linear system of equations must be solved each time the electromagnetic fields are updated. This additional step arises from the fact that the B-spline projection coefficients of a function do not directly correspond to its values at the grid points (*aka* modal interpolation). Therefore, these coefficients need to be determined by solving the resulting linear system, using the values of the function at the grid points. The details of the interpolation methods are provided in [Appendix B](#).

Problems can also arise when the values needed for the interpolation fall outside the last available magnetic surface, where no data exists. Also, numerical errors in some equilibrium codes can significantly increase near the magnetic axis and in the vicinity of the last available surface, leading to inaccurate integration of particle trajectories in these regions. Therefore, in TAPAS we have to face this kind of numerical issues. In addition, independently of the accuracy of the magnetic equilibrium, another issue that has been addressed in TAPAS is related to the inherent complexity of realistic fusion device geometries. Interpolating fields accurately in such complicated geometries presents additional challenges, and appropriate techniques must be employed to mitigate potential errors. In this section, we discuss these two issues separately.

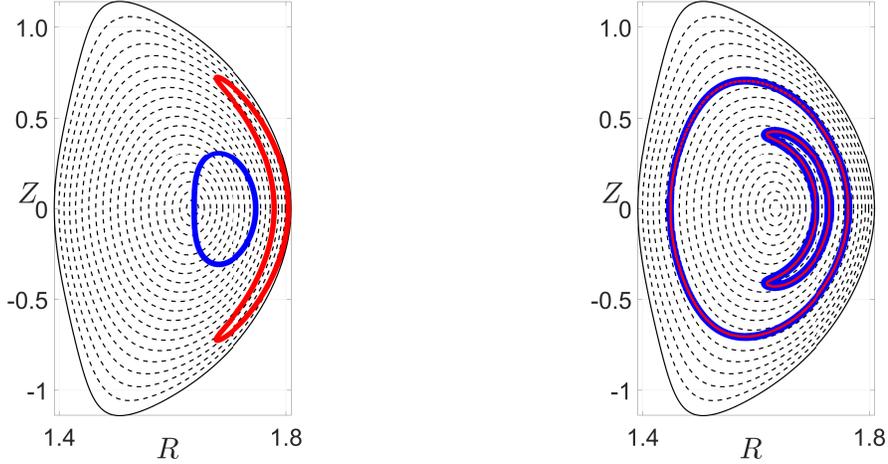


Figure 4: Left Frame: two examples of particle trajectories: one traversing the magnetic axis (blue curve) and another in close proximity to the last closed surface (red curve). Right Frame: trajectories of a passing particle and a trapped banana particle, obtained using both the full orbit (blue curves) and guiding center versions of TAPAS (red curves).

### 2.6.1. Interpolation of fields in the vicinity of the last available magnetic surface

In order to interpolate the fields at the particle position, it is necessary to obtain the toroidal coordinates of the particle, since the fields are calculated in toroidal geometry. This process involves calculating the cylindrical coordinates  $(R_p, Z_p, \varphi_p)$  corresponding to the new Cartesian position  $(x_p, y_p, z_p)$ , and then the corresponding toroidal coordinates  $(\rho_p, \theta_p, \zeta_p)$  can be obtained using Lagrange interpolation.

Nevertheless, the process of interpolation can introduce certain challenges. In the case of a 2D axi-symmetric equilibrium, when applying 2D Lagrange interpolation to approximate a quantity at a specific location, it is necessary to know its values at 16 points within the neighborhood of the interpolation point. This particular region, depicted as the mauve-shaded and gray-shaded rectangles in Fig.3, will be referred to as the *interpolation rectangle* throughout the discussion.

In some cases, certain nodes of the interpolation rectangle may lie outside the last closed surfaces, as illustrated by the gray-shaded rectangle in Fig.3. In this figure, a particle that follows a trajectory that brings it close to the last magnetic surface is shown as a blue circle. Two interpolation rectangles are shown: the first one (mauve-shaded) does not contain the particle, and all its nodes are located within the physical domain. However, although the second interpolation rectangle (gray-shaded) contains the particle, four of its nodes (represented by black circles at the top right) exist outside the physical domain. This can be problematic as it introduces inaccuracies in the interpolation process and affect the particle trajectory calculation since the corresponding toroidal coordinates  $(\rho, \theta, \zeta)$  are not defined at those nodes. To accurately solve this problem, we apply the following steps:

- (i) Initially, an attempt is made to find an interpolation rectangle where all its nodes

lie within the physical domain. This rectangle is preferred as it guarantees that the toroidal coordinates  $(\rho, \theta, \zeta)$  are defined for all the nodes, allowing for accurate interpolation. If it is not possible to find an interpolation rectangle that satisfies that criterion, the focus shifts to identifying the interpolation rectangle with the fewest nodes located outside the physical domain. This approach aims to minimize the impact of nodes falling outside the physical domain on the accuracy of the interpolation.

- (ii) If  $N$  nodes of the interpolation rectangle fall outside the physical domain, an inverse problem of Lagrange interpolation is solved to determine the field values at these nodes. This is done as follows
- We search for the complete interpolation rectangle that shares the largest area with the rectangle containing the particle while being entirely situated within the physical domain (e.g., the mauve-shaded rectangle in Fig.3).
  - We arbitrarily generate  $N$  points that exist inside both the complete rectangle, which is the mauve-shaded rectangle in Fig.3, and the particle rectangle, which is the gray-shaded rectangle in Fig.3. These points are represented as dark-green circles in Fig.3. It is worth noting that while the intersection points between cylindrical grid lines and the last closed surface can be used, they can be problematic as they may be located very close to the nodes that exist outside the physical domain. Consequently, this leads to ill-conditioned Lagrange basis functions, where  $L_i \approx 0$  and  $f_i \rightarrow \infty$ , where  $f_i$  is the value to be calculated (see Appendix B).
  - We approximate the coordinates  $(\rho, \theta, \zeta)$  at generated points using the complete interpolation rectangle.
  - Knowing the values of  $(\rho, \theta, \zeta)$  at the generated points, we then solve the inverse Lagrange interpolation problem to assign physical values to the nodes outside the physical domain. In other words, we find the values of  $(\rho, \theta, \zeta)$  at the nodes outside the domain such that they interpolate those given at the randomly generated points.
- (iii) Once the interpolation rectangle is completely determined, we can employ it to approximate the values  $(\rho_p, \theta_p, \zeta_p)$  corresponding to the particle position.

We observed that the choice of interpolation order used to calculate the values of the function at the generated points required to fully determine the interpolation rectangle can have a significant impact on the trajectory of the particle. This is because low-order interpolations can introduce numerical drifts and thereby result in poor accuracy. For example, in certain cases, we observed that linear interpolation may introduce numerical drifts, resulting in particle losses at the mid-plane of the device. Hence, in TAPAS, we consistently employ cubic interpolation. As an example of the strategy outlined above, the red curve in the left frame of Fig.3 shows the trajectory of a trapped particle in DIII-D tokamak moving in the vicinity of the last closed surface, and whose trajectory is determined using the algorithm described above to find its coordinates in toroidal

geometry.

*2.6.2. Interpolation near the magnetic axis* TAPAS relies on magnetic equilibrium, which is obtained analytically or numerically using equilibrium codes, to construct the toroidal geometry specific to a particular device. The external simulation coupled with TAPAS solves the dynamic equations within this coordinate system. However, depending on the external simulation and/or equilibrium code used, TAPAS may sometimes define equilibrium quantities starting from a closed magnetic surface located at a distance  $\Delta r$  from the magnetic axis. As a consequence, equilibrium quantities and/or electromagnetic perturbations are not defined within the first closed surface: for instance, this situation arises when TAPAS is coupled with GYSELA5D, which solves the gyrokinetic equation by excluding the center  $r = 0$  to prevent the divergence of certain quantities like the metric, gradient, and others. Therefore, it becomes necessary to develop a method for extrapolating fields when particles enter this region.

Several strategies can be employed to address this issue. For example, one approach is to approximate the first closed surface around the magnetic field as a circle and use the corresponding analytical formula to calculate equilibrium quantities at the particle's position. A more realistic approximation involves treating the first closed surface as an ellipse or utilizing a formula derived from solving the Grad-Shafranov equation analytically to extrapolate different quantities when particles enter the first closed surface.

However, when employing any of these strategies, we observed two distinct effects that can occur separately or simultaneously. The first effect is that the particle motion exhibits an arbitrary unphysical diffusion behavior when it approaches the magnetic axis, followed by a numerical drift comparable to (or larger than) its Larmor radius upon exiting that region. Although the errors arising from this behavior can be quantified, the second effect, which involves unphysical particle trapping near the magnetic axis, is more subtle. Identifying this effect becomes challenging during simulations, making it difficult to measure, yet it can significantly impact the simulation, particularly in cases involving the injection of energetic particles near the magnetic axis.

Following the previous discussion, we have developed a straightforward procedure that utilizes a second-order Taylor expansion in  $2D$  ( $3D$  expansion for  $3D$  equilibriums, such as in the case of stellarator devices), to approximate equilibrium fields within the first closed surface. Subsequently, if we intend to calculate a quantity  $f$  at the particle's position  $(\rho_p, \theta_p, \zeta_p)$  when  $\rho_p < \rho_1$  (where  $\rho_1$  represents the radius of the first closed surface in the generalized toroidal coordinate system), we can apply the following relationship:

$$f(\rho_p, \theta_p, \zeta_p) = f(\rho_i, \theta_i, \zeta_i) + \sum_{j=1}^3 \frac{\partial f}{\partial x^j} \Big|_{\rho_i, \theta_i, \zeta_i} (x^j - x_p^j) + \sum_{j=1}^3 \sum_{k=1}^3 \frac{\partial^2 f}{\partial x^j \partial x^k} \Big|_{\rho_i, \theta_i, \zeta_i} (x^j - x_p^j)(x^k - x_p^k) \quad (28)$$

where  $x^j$  and  $x^k$  refer to  $\rho$ ,  $\theta$ , and  $\zeta$ .

Problems near the magnetic axis can be addressed by using another equilibrium code, such as DESC [46], known for its higher accuracy in this region. But this choice

is not always practical since it will also require replacing the equilibrium code in the external simulation coupled with TAPAS.

The impact of employing the Taylor expansion method on the particle trajectory is illustrated in the left frame of Fig.4. Here, we plot with blue color the projection onto the poloidal cross-section of the trajectory of a trapped particle in the DIII-D tokamak, where equilibrium quantities close to the magnetic axis were estimated via Eq.(4). The simulation lasted for  $10^6 \tau_{\text{TAPAS}} \approx 10^{-3}$  sec, where  $\tau_{\text{TAPAS}} \approx 10^{-9}$  sec is the characteristic time of TAPAS. The extended duration was selected to ensure that the particle traverses the first closed surface multiple times. As anticipated, the resulting trajectory, represented by the blue curve in the left frame of Fig.4, follows closely to the expected physical path, exhibiting negligible numerical drift over the entire simulation period.

### *2.7. Guiding Center vs full orbit*

As a benchmark for the full orbit version of the code, we performed numerical experiments to compute particle trajectories using both the full orbit version and the previously implemented guiding center version for circular geometry, and upgraded here to account equilibriums derived from numerical codes or an analytical solution of the Grad-Shafranov equation using expansion methods.

An essential assumption of gyro-kinetic theory is the adiabatic conservation of the magnetic moment, made possible by the presence of a strong background guide field. Given the validity of this assumption, which depends on several factors such as the strength and profile of the guide field, electromagnetic perturbations, and particle energy, trajectories calculated with the guiding center version accurately approximate those obtained via the full orbit version, which is confirmed in Fig.4.

In the right frame of Fig.4, trajectories for two particles —passing and trapped— are plotted. Both particles have equal initial energies:  $E_{\text{EP}} = 4E_{\text{th}}$ , where  $E_{\text{th}}$  is the reference energy. While both particles were initialized at  $\theta = 0$ , their radial initializations differ, with the passing particle at  $\rho = 170\rho_r$  and the trapped particle at  $\rho = 50\rho_r$ , where  $\rho_r$  denotes the reference length. The blue and red curves represent trajectories for the full orbit and guiding center versions, respectively. As illustrated, the guiding center approximation accurately captures the particle’s trajectory in this scenario. Additionally, this test provides also a benchmark for particle initialization in 6D phase space starting from an initialization in 5D phase space, as discussed in Section 2.3.

### *2.8. Performance of the parallel algorithm of TAPAS*

The simulations presented in this work were performed on the French Jean-Zay supercomputer (HPE SGI 8600 system), which comprises two partitions: CPU and GPU partitions. The CPU partition employed for this paper consists of 1528 compute nodes, each equipped with 2 Intel Cascade Lake processors (20 cores per processor and

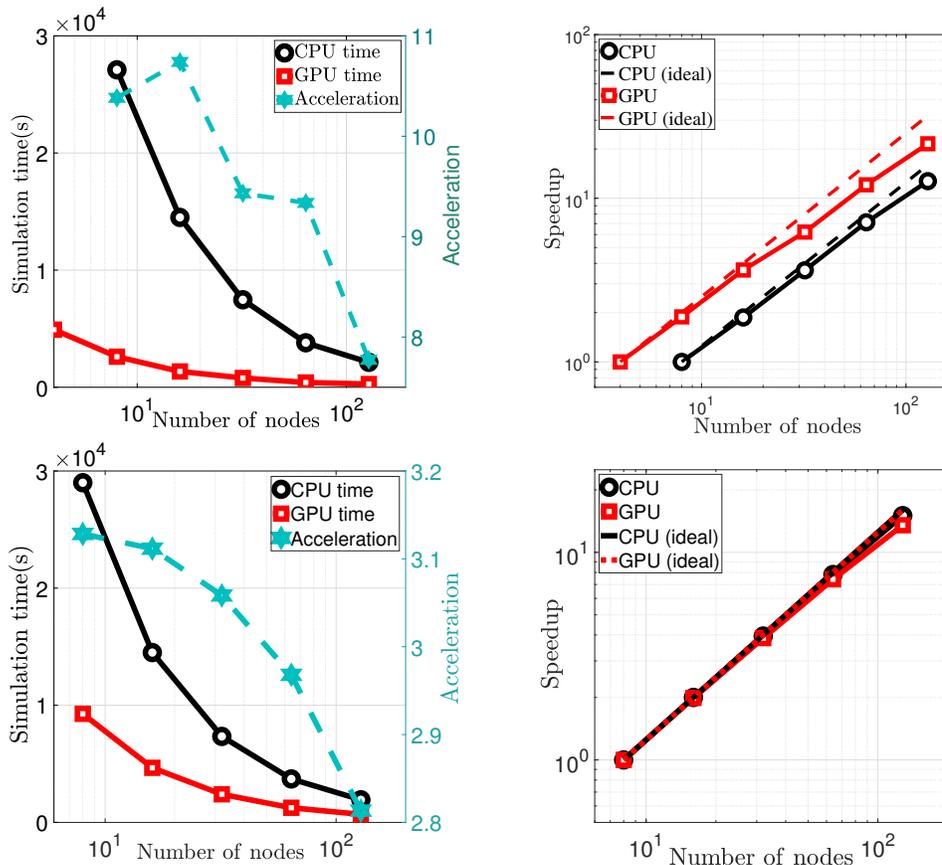


Figure 5: Simulation time (left frames) and speedup (right frames) for circular geometry (top frames) and reconstructed magnetic equilibrium of DIII-D tokamak (bottom frames) as a function of the number of nodes for CPU (black curves) and GPU (red curves) partitions on Jean-Zay supercomputer.

192 GB of shared memory per node). While the GPU nodes used here consist of 2 processors and 4 Nvidia Tesla V100 GPUs, each equipped with 32GB of memory. The compute nodes are interconnected by an Intel Omni-Path network (OPA).

In this section, we discuss the scaling analysis of the new full-orbit version of the code on Jean-Zay supercomputer. In addition to evaluating the performance of the full-orbit scheme, we focus also on the impact of bspline interpolation on overall performance, particularly when integrating the 3D electromagnetic perturbation calculated by an external code. We also investigate the effects resulting from considering realistic D-shape equilibrium (here the magnetic equilibrium for DIII-D tokamak), obtained by fitting experimental data and reconstructing magnetic equilibrium using numerical codes such as VMEC; an important step as particles along their trajectory are checked at every time step if they become close to the magnetic axis or the last available closed surface, introducing more if-conditions into the algorithm that might affect code performance.

Two criteria were employed to measure performance. The first criterion is the acceleration defined as the ratio between the time on CPU nodes (with one MPI process

per code) and the time on an equal number of GPU-accelerated nodes (one MPI process per GPU). The second criterion, speedup, is defined as  $t_N/t_{N_0}$ , where  $N$  represents the number of nodes in the corresponding simulation and  $N_0$  is the number of nodes in the reference simulation. Here,  $t$  denotes the time required to complete the relevant simulation. Further, deviations from the ideal speedup were also estimated.

To investigate the parallel implementation performance of bspline interpolation, a number of simulations were performed using the full-orbit version of TAPAS, initializing particles in circular geometry. The 3D electromagnetic perturbations were obtained from an electrostatic turbulence simulation performed using the gyro-kinetic code GYSELA5D. In these simulations,  $N_p \sim 2.7 \times 10^8$  particles were uniformly initialized in 6D phase space.

The top-left frame of Fig.5 shows simulation time on the left-y axis as a function of the number of nodes for both CPU (black curves) and GPU (red curves) implementations, alongside the acceleration on the right-y axis. Interestingly, an acceleration factor within the range of [7.75 – 11] was gained. The decrease in acceleration with increasing the number of nodes can be attributed to having smaller number of particles per node, leading to faster calculations on the CPU node. The top-right frame of the figure illustrates speedups and their deviations from the ideal speedup for both CPU and GPU implementations. Despite a larger deviation in the GPU implementation from the ideal speedup, the GPU implementation enables achieving a large acceleration factor when the full-orbit version with bspline interpolation and circular geometry is employed.

The performance analysis of the code when using the full-orbit version with realistic geometry, requiring special treatment of particles near the magnetic axis and the last available closed surface, involved initializing TAPAS with approximately  $\sim 1.34 \times 10^8$  particles uniformly distributed in 3D spatial space. Particle initialization in velocity space start with initializing them in  $(E, \Lambda)$  space (see Section 2.3). These particles were mono-energetic ( $E = 16.93$  keV) and uniformly initialized in pitch angle ( $-1 \leq \lambda \leq 1$ ). The initial coordinates of the particles in velocity space  $(v_x, v_y, v_z)$  were calculated applying coordinate transformation as outlined in Section 2.3.

The bottom-left frame of Fig.5 shows the simulation time (left-y axis) and the acceleration factor (right-y axis) as a function of the number of nodes for both CPU (black curves) and GPU (red curves) implementations. In this scenario, an acceleration factor within the range of [2.8–3.15] was reached, notably smaller than that achieved for circular geometry. This discrepancy largely results from interpolation issues arise when particles become close to the magnetic axis and the last available closed surface, implying the need for further optimizations to reach better acceleration factor. Moreover, the speedups in the full-orbit version with numerically/experimentally calculated geometry, depicted in the bottom-right frame of Fig.5, are almost identical for both CPU and GPU implementations. Nevertheless, a slightly larger deviation from the ideal speedup is evident in the GPU implementation as the number of nodes increases.

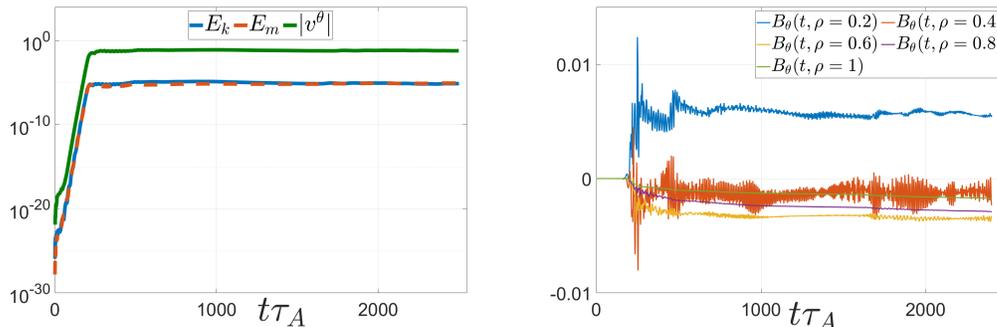


Figure 6: Left frame: the temporal evolution of magnetic energy (orange curve), kinetic energy (blue curve), and zonal flow (green curve). Right frame: the variations in the amplitude of the poloidal magnetic field at predefined radial positions.

### 3. FAR3d simulation of high- $\beta$ NBI-induced Alfvén Eigenmode activities

In this section, we briefly outline the results of Far3d simulations that have been employed in TAPAS to investigate the dynamics of energetic particles in the presence of Alfvén eigenmodes. A nonlinear simulation has been performed with FAR3d [34]‡ to investigate the behavior of Alfvén eigenmodes in the presence of energetic particles (Deuterium) resulting from NBI heating. Magnetic equilibrium was computed via the VMEC code, incorporating up to 15 poloidal mode numbers. Initial profiles for energetic particle density and temperature were obtained from the TRANSP code. In the FAR3d simulation, the energetic particle temperature peaks at the magnetic axis ( $T_{EP} = 21$  keV) with  $\beta_{EP} = 3\%$ , while the thermal plasma temperature remains around  $T_{th} \approx 2.1$  keV. During this nonlinear simulation, 139 Fourier mode  $m/n$  pairs were chosen:  $n = 0$  ( $m = -14 \rightarrow 14$ ),  $n = 3$  ( $m = -20 \rightarrow 20$ ), and  $n = 6$  ( $m = -52 \rightarrow 52$ ). These mode numbers were determined through linear stability analysis conducted using the linear version of FAR3d. This analysis scanned the growth rates and frequencies of different toroidal modes numbers. It was shown [43] that the dominant modes span  $n = 1 \rightarrow 6$  for various linear simulations with different values of  $\beta_{EP}$ . In this context, the  $n = 3$  toroidal mode family exhibited the largest growth rates, hence its selection. Meanwhile, the choice for the  $n = 6$  mode family stems from the instability of its associated modes for large values of  $\beta_{EP}$ , which the case of the non-linear simulation we performed ( $\beta_{EP} = 3\%$ ).

Fig.6 shows the temporal evolution of electromagnetic and kinetic energies of the thermal plasma, represented by the dashed-orange and blue curves, respectively. Additionally, it illustrates the behavior of the zonal flow structure, indicated by the green curve, in the left frame. As expected, the evolution of thermal plasma energies unfolds in three distinct stages: the initial phase, which corresponds to very small amplitude perturbations before the emergence of various unstable modes, extends until

‡ (See Appendix Appendix C for a brief description of the code

$t = 100\tau_A$ , an intermediate linear phase until  $t = 224\tau_A$ , characterized by the exponential growth of unstable modes, implying energy exchange between energetic particles and the background thermal plasma, and a third phase showing the non-linear saturation of unstable modes and persists until the end of the simulation at  $t = 2500\tau_A$ . Throughout both the linear and non-linear phases of these unstable modes, the interaction between Alfvén eigenmodes and energetic particles may result in significant losses of energetic particles, and thus degrade the confinement performance which is discussed later. Furthermore, as demonstrated in the left frame of Fig.6, the evolution of zonal flows also follows a similar pattern in terms of energies and experiences these same three distinct phases. These zonal flows are characterized by the poloidally and toroidally averaged poloidal velocity, i.e.  $\langle v^\theta \rangle_{\theta,\zeta} \approx \frac{\partial \bar{\Phi}_{m=0,n=0}}{\partial \rho}$ , and it can play a key role in the transport of energetic particle.

The right frame of Fig.6 illustrates the behavior of the amplitudes of the poloidal component of the magnetic perturbation over time at various pre-selected radial positions. As depicted in the figure, these amplitudes increase as the system enters the saturation phase, but the rate of growth differs at different radial locations, with the most significant increase occurring near the magnetic axis at  $\rho = 0.2$ . Furthermore, we observe that the behavior in regions beyond  $\rho = 0.4$  is quasi-steady, indicating that the amplitudes do not exhibit sudden changes over time, i.e. MHD bursts.

In contrast, this quasi-steady state behavior is not observed in the inner plasma, as shown by the blue and orange curves in the right frame of Fig.6 for  $\rho = 0.2$  and  $\rho = 0.4$ , respectively. At these specific locations, we observe quasi-steady behavior for extended time intervals, followed by MHD burst events characterized by abrupt fluctuations in the amplitudes. During these events, the amplitudes increase and subsequently return to their values prior to the event. Indeed, the occurrence of these MHD burst events at these specific locations is closely correlated with the concentration of Alfvén eigenmode activities in those regions, as shown in Fig.7 and Fig.8. However, the investigation of the dependence of these burst events on  $\beta_{EP}$  [43] have shown that they become less pronounced for energetic particles with smaller values of  $\beta_{EP}$ , and they can even disappear entirely when  $\beta_{EP} \leq 1\%$  [43].

Fig.6 highlights two essential considerations regarding coupling Far3d with TAPAS. First, the energy evolution curves show all characteristic times involved in the dynamics. On the another hand, the fluctuation evolution in the amplitude of the poloidal component of the magnetic field (i.e., the right frame of Fig.6) gives access to the shortest characteristic time scales in the studied scenario, given that the amplitude values are saved at each time step, set here as  $dt = 0.025\tau_A$ . TAPAS time step, therefore, should be smaller than the shortest timescale ( $\tau_s$ ) observed in the external code. In this paper, this condition is satisfied, as the characteristic frequency in TAPAS is the gyro-frequency time  $\Omega_c$ , approximately  $10^3\tau_s^{-1}$ . Then, at time  $t_n$ , the electromagnetic perturbation are linearly interpolated using their values at  $t_{prev}$  and  $t_{next}$ , satisfying  $t_{prev} \leq t_n \leq t_{next}$ , which are computed externally by Far3d.

Fig.7 presents the perturbations of the electrostatic potential (top frames) and the

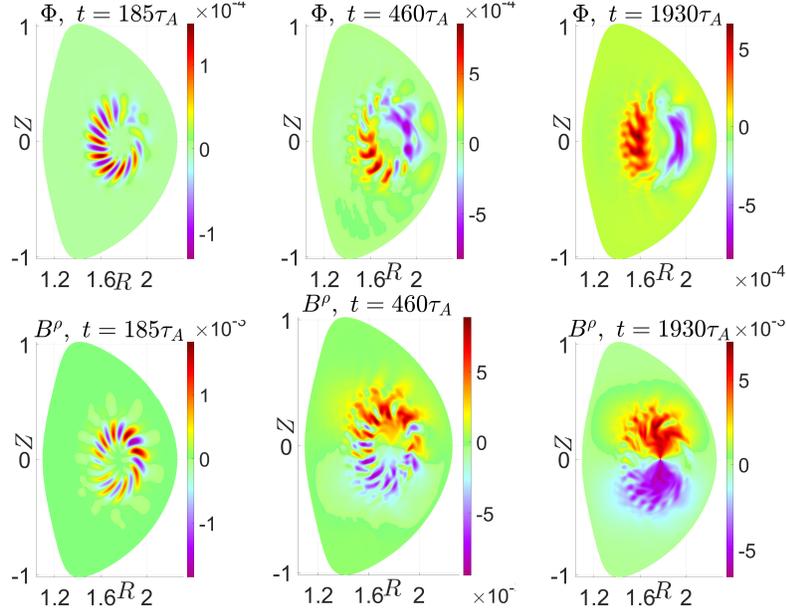


Figure 7: Electrostatic Perturbation (top frames column), and radial component of the magnetic field (bottom frames) at three different times: the linear phase (left), the initial stage of saturation phase (middle), and a later stage of saturation phase (right), as obtained from FAR3d simulations of Alfvén eigenmodes.

radial component of the magnetic field (bottom frames). These fields are computed at three distinct times:  $t = 185\tau_A$  (top row),  $t = 460\tau_A$  (middle row), and  $t = 1930\tau_A$  (bottom row). These time instances were deliberately selected during the linear growth phase of unstable modes, the early stages of the non-linear phase, and late stage in the saturation phase, coinciding with an MHD burst event occurring around  $t = 1930\tau_A$ .

During the linear phase of the simulation, the dominant mode pairs were  $9/3$ ,  $-9/-3$ ,  $10/3$ , and  $-10/-3$ , all of which were localized near the magnetic axis within the inner region of the plasma. This concentration is evident in the profiles of  $\Phi$  and  $B^\rho$  presented in Fig.7. However, it is challenging to distinguish between these different modes in these profiles because they are localized in the same region, with their extrema occurring nearly simultaneously. This mode localization explains why MHD burst events occur at radial positions within the inner plasma, as demonstrated in the right frame of Fig.6.

As we transition into the non-linear phase, new Alfvén eigenmodes start to contribute to the dynamics. To highlight this point, we plot in Fig.9 the kinetic energy of thermal plasma (top-left), its magnetic energy (top-right), and the energetic particles energy (bottom) for the strongest 10 Alfvén eigenmodes (i.e. excluding modes with  $n = 0$ ). As shown in the figure, although the mode  $9/3$  appears as the dominant one, the contributions of other modes become evident and cannot be neglected.

Moreover, during the early stages of the saturation phase (i.e., at approximately  $t\tau_A = 460$ ), we observe contributions from other modes, including  $11/3$ ,  $-11/-$

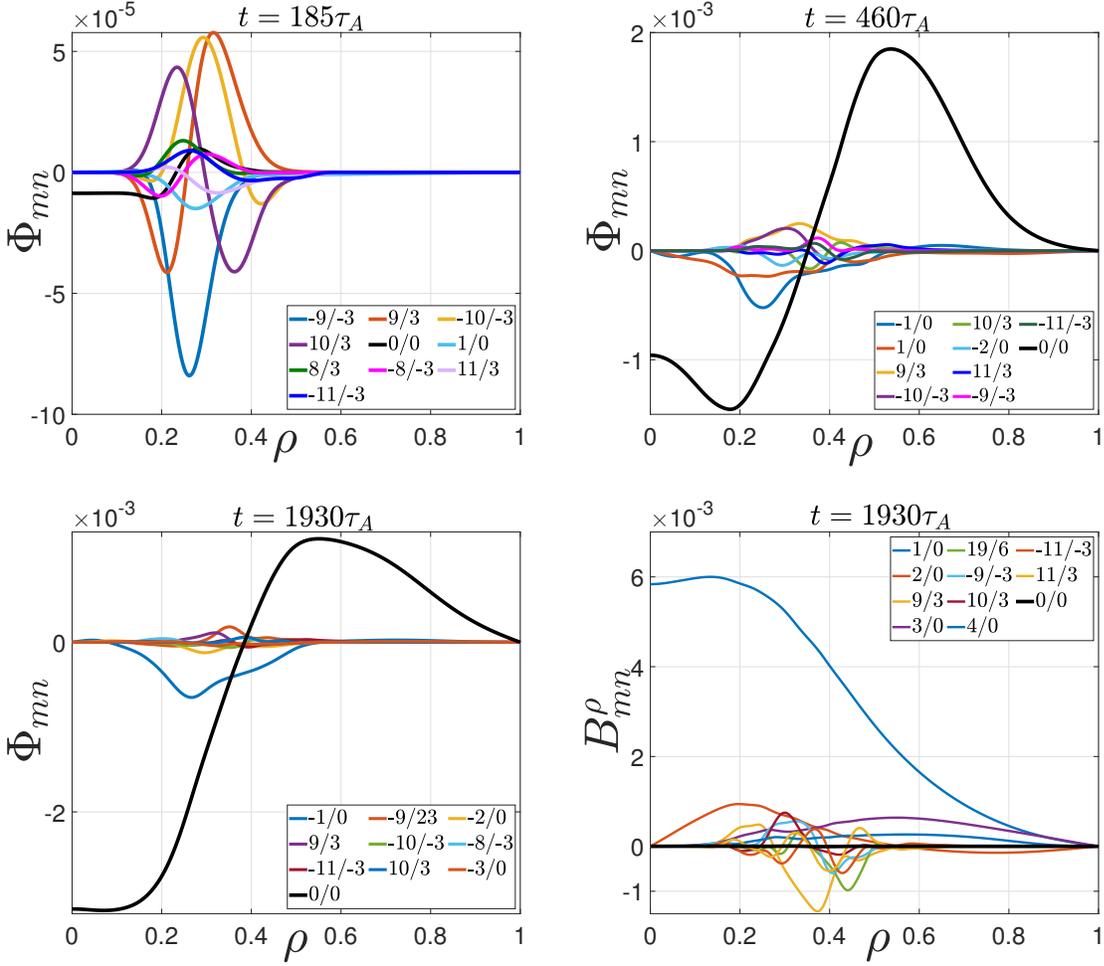


Figure 8: The first ten dominant Fourier mode pairs ( $m/n$ ) of the electrostatic potential as obtained from FAR3d simulation at three distinct time points: the linear phase (top-left), the initial stage of saturation phase (top-right), and a later stage of saturation phase (bottom), as obtained from FAR3d simulations of Alfvén eigenmodes.

3, 1/0, -1/0. Fig.8 displays the first ten dominant modes in electrostatic perturbations in the top and bottom-left frames where we also show the contribution resulting from the modes  $n = 0$ . This component dominates during the non-linear phase, except for the magnetic stream function, where interestingly, it is not the dominant one. This holds true for magnetic perturbations  $B^\rho$  and  $B^\theta$  as well. To emphasize this point, the bottom-right frame of Fig.8 shows the first ten dominant modes, including the 0/0 mode. It becomes evident that this mode is not among the dominant ones; instead, in this case, the dominant mode is 1/0, as shown in the bottom-right frame of Fig.7, and which extends over the entire plasma region forming an envelope for the remaining modes [43].

In summary, Far3d simulations performed in this study to investigate the nonlinear

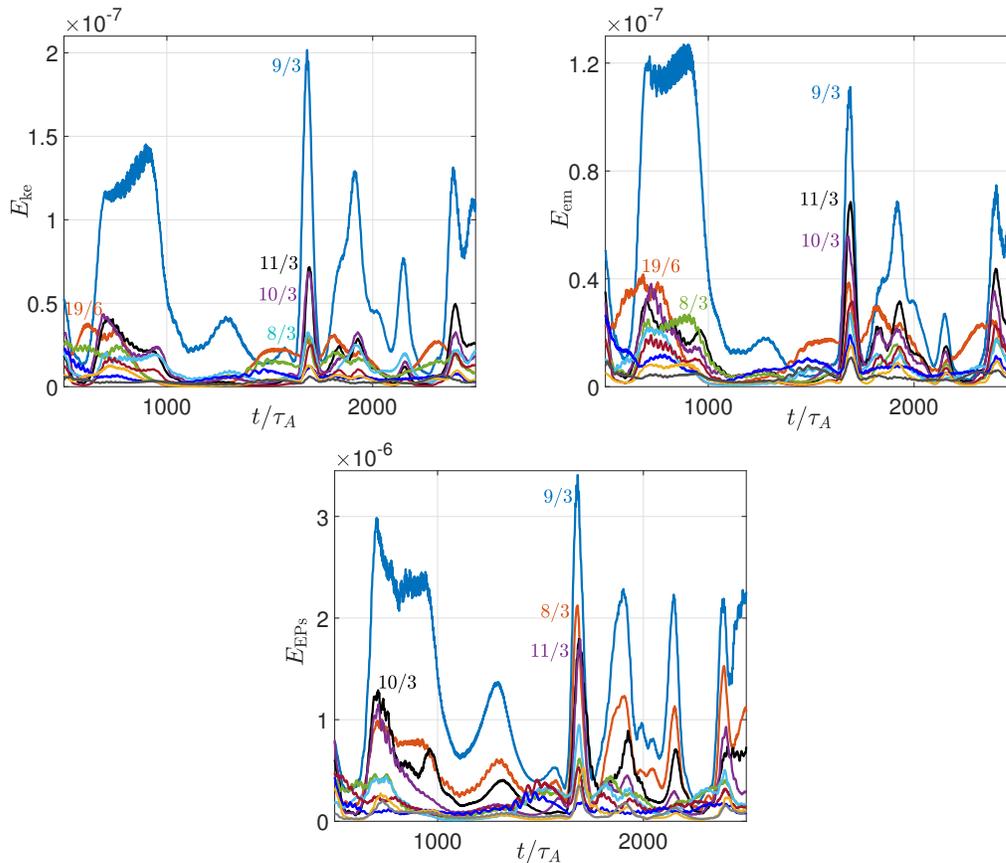


Figure 9: Kinetic (top-left) and magnetic (top-right) energies for the first 10 dominant Alfvén eigenmodes in the thermal plasma, and energies of the energetic particles for the first 10 dominant modes (bottom). These curves were plotted by averaging the original curves sampled at every Alfvén time using a time window of width  $32\tau_A$  to smooth out the high frequencies while keeping an accurate approximation of the mode energy.

behavior of Alfvén eigenmodes, utilizing the DIII-D equilibrium of discharge #159243 with  $\beta_{EPs} = 3\%$ , indicate that both the linear and nonlinear phases are mainly dominated by modes  $i/3$  where  $i = 8, 9, 10, 11$ . Furthermore, the temporal evolution of fluctuations in the radial magnetic field reveals the emergence of MHD-burst activities characterized by an abrupt increase in amplitude. This increase is found to be associated, as will be discussed elsewhere, with an increase in the radial flux of energetic particles toward the last available closed magnetic surface; implying a larger particle losses compared to scenarios where MHD-burst events do not excited, as observed when  $\beta_{EP}$  is below a critical value, as will be detailed in a forthcoming paper.

#### 4. Transport and losses of energetic Deuterium with TAPAS

The 3D electromagnetic perturbations obtained with FAR3d have been introduced in TAPAS to investigate the impact of Alfvén eigenmode activities on the dynamics of

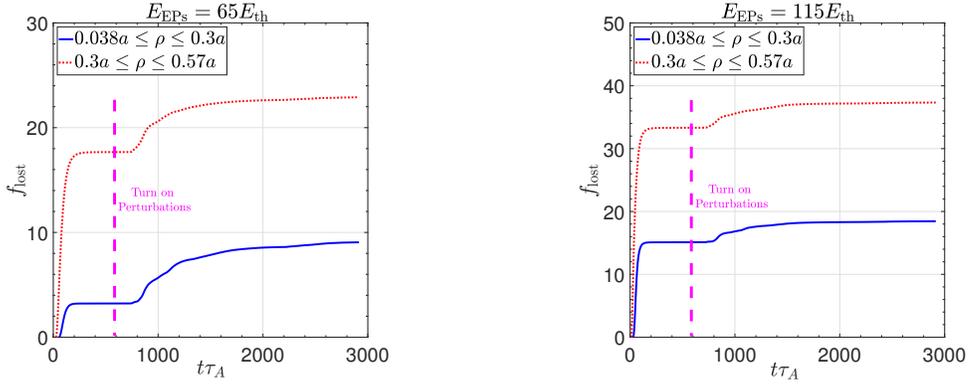


Figure 10: Fraction of lost particles, defined as  $f_{\text{lost}} = \frac{N_{\text{p,lost}}(t_{\text{end}})}{N_p}$ , for various simulations considered in this paper. The vertical line indicates the moment electromagnetic perturbations were initiated, prior to which only prompt losses were present.

energetic particles initialized at different energies and radial positions. The study conducted with TAPAS involved mono-energetic particle beams uniformly initialized in poloidal and toroidal directions, as well as in pitch angle (i.e.,  $-1 \leq \lambda \leq 1$ ). While the radial initialization was also uniform, the particles were initialized in two different radial intervals for different simulations:  $0.038a \leq \rho \leq 0.3a$  (hereafter, the inner region) and  $0.3a \leq \rho \leq 0.57a$  (hereafter, the outer region). These radial initializations were selected to involve regions where AEs activities are more pronounced, while also expanding two radial intervals: one in proximity to the magnetic axis and another situated farther from it.

To study the effects of the co-existence of the magnetic equilibrium and the electromagnetic perturbations on the energetic particles, we scanned two energies. These simulations covered two energy levels ( $E_{\text{EPs}} = 65E_{\text{th}}$  and  $E_{\text{EPs}} = 115E_{\text{th}}$ ), where  $E_{\text{th}} \approx 2$  keV is the thermal energy of the background plasma. For each level of energy, we performed a total of 8 simulations, each involving a particle count of  $N_p = 1.6 \times 10^6$ . Furthermore, for both investigated scenarios—when considering only the equilibrium and when it is coupled with electromagnetic perturbations—two sets of simulations were performed: one considering collisions and the other omitting them. In simulations that include collisions, the characteristic time of the stochastic part of collision operator is  $\tau_c = 10^3 \tau_{\text{TAPAS}} \approx 10^{-6} \text{sec}$ . Therefore, the time step in Eq.(9) is 1000 times larger than that used for integrating particle trajectories.

In total, 16 simulations were performed. In the following subsections, we start by studying the prompt losses (i.e., losses of charged particles in the absence of electromagnetic perturbations). Subsequently, we focus on the effects of including the electromagnetic perturbation resulting from Alfvén eigenmodes, as discussed in Section 3.

#### 4.1. Prompt Losses

In the presence of a non-uniform magnetic equilibrium (e.g., a toroidal magnetic field proportional to  $1/R$ ), particles can be categorized as co-passing, counter-passing, or trapped. Losses in the presence of only the equilibrium magnetic field strongly depend on particle type. These losses occur when certain particles have orbits that intersect with the device wall, leading to their loss during their first orbit within the device. To analyze and characterize these losses, we performed a total of 8 simulations, each involving a particle count of  $N_p = 1.6 \times 10^6$ . These simulations covered two energy levels ( $E_{\text{EPs}} = 65E_{\text{th}}$  and  $E_{\text{EPs}} = 115E_{\text{th}}$ ). For each energy level, two sets of simulations were performed—one considering collisions and the other omitting them. In simulations that include collisions, the characteristic time of the stochastic part is  $\tau_c = 10^3 \tau_{\text{TAPAS}} \approx 10^{-6} \text{sec}$ . Therefore, the time step in Eq.(9) is 1000 times larger than that used for integrating particle trajectories. Additionally, the particles were radially initialized within two specific radial intervals:  $0.038a \leq \rho \leq 0.3a$  (hereafter, the inner region) and  $0.3a \leq \rho \leq 0.57a$  (hereafter, the outer region).

Fig.10 shows, up to the moment indicated by the vertical line, the cumulative distribution function of the fraction of lost particles in the presence of only the magnetic equilibrium, denoted as  $f_{\text{lost}} = \frac{N_{\text{p,lost}}(t)}{N_p}$ , over time for the set of simulations where collisions were omitted (i.e. prompt losses). It is observed that for both simulated energies, almost all losses occur before  $t = 300\tau_A$ . Throughout this paper, blue curves represent the case where particles initialized in the inner region (i.e. closer to magnetic axis), while red curves correspond to particles initialized in the outer region. As a consistent observation for the scanned energies, we observe that particles initialized in the outer region exhibit larger losses. For  $E_{\text{EPs}} = 65E_{\text{th}}$  ( $E_{\text{EPs}} = 115E_{\text{th}}$ ), the total prompt losses (i.e. before the vertical line) were 3.3% and approximately 17.8% (15.9% and 33.4%) for inner and outer regions, respectively.

To better understand these losses, the top frames of Fig.13 show the fraction of lost particles at the end of various simulations, defined by  $f_{\text{lost}} = \frac{N_{\text{p,lost}}(t_{\text{end}})}{N_p}$ , as a function of the initial parallel velocity of the particles. As shown by the blue curves in the figure, for both simulated energies, the average value of  $v_{\parallel}$  for lost particles peaks at a value of 5, normalized to the reference velocity in TAPAS. However, while lost particles at higher energy ( $E_{\text{EPs}} = 115E_{\text{th}}$ ) explore a wider interval in  $v_{\parallel}$  for the initialization in the inner region, this interval remains comparatively small compared to the case when particles are initialized in the outer region, as illustrated by the red curves in the bottom frame of Fig.13. Two factors contribute to this behavior. Firstly, the main part of losses results from trapped particles with orbit widths intersecting with the wall. On the other hand, with an increase in the energy of charged particles, there is a growing contribution to losses from co-passing and counter-passing particles.

To gain insight into this behavior, we present the prompt losses exhibited by each particle type for two energetic particle beams ( $E_{\text{EPs}} = 65E_{\text{th}}$  and  $E_{\text{EPs}} = 115E_{\text{th}}$ ) in Fig.11. The left and right frames correspond to the low and high-energy cases,

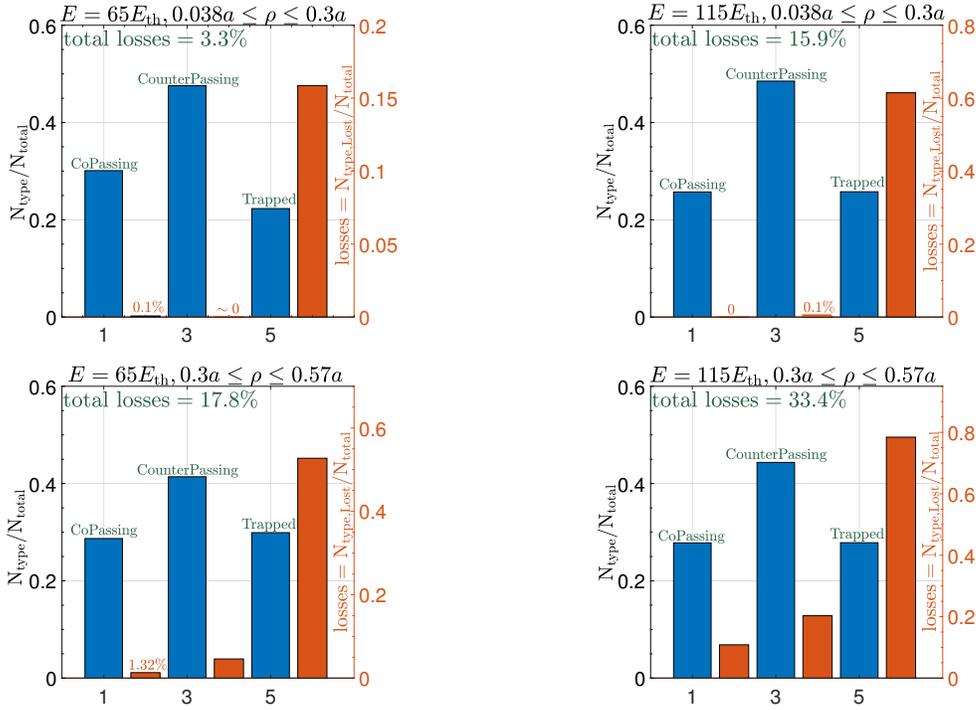


Figure 11: Fractions of different particle types are represented by the blue bars: left bars for co-passing particles, middle bars for counter-passing particles, and right bars for trapped particles. The corresponding prompt losses of each type are shown by the orange bars -notice these losses are calculated as the number of lost particles of a specific class divided by the total number of particles of that class. The top frames correspond to radial initialization in the inner region, while the bottom frames represent radial initialization in the outer region. The left frames are associated with particle energy  $E_{EPs} = 65E_{th}$ , and the right frames correspond to particle energy  $E_{EPs} = 115E_{th}$ .

respectively, while the top and bottom frames represent radial initialization of the particles in the inner and outer regions, respectively.

As shown in the left frames of Fig.11, in the case of low energy ( $E_{EPs} = 65E_{th}$ ), most losses originate from trapped particles when particles are initialized closer to the magnetic axis. Losses from co-passing and counter-passing particles are completely negligible (e.g., the contribution of co-passing particles to total losses is approximately 0.7%). However, when particles are initialized in the outer region, losses from co-passing and counter-passing particles amount to about 2.2% and 10.5%, respectively, as shown in the bottom-left frame of the figure.

As the energy of the particles increases ( $E_{EPs} = 115E_{th}$ ), losses resulting from trapped particles also increase for both radial initializations. As indicated in the right frames of Fig.11, 62% of trapped particles are lost when they are initialized in the inner region, while this ratio increases to 79% for the outer region. Moreover, as shown in the top-right frame, the contribution of co-passing and counter-passing particles to the losses remains negligible, while for radial initialization in the outer region shown in

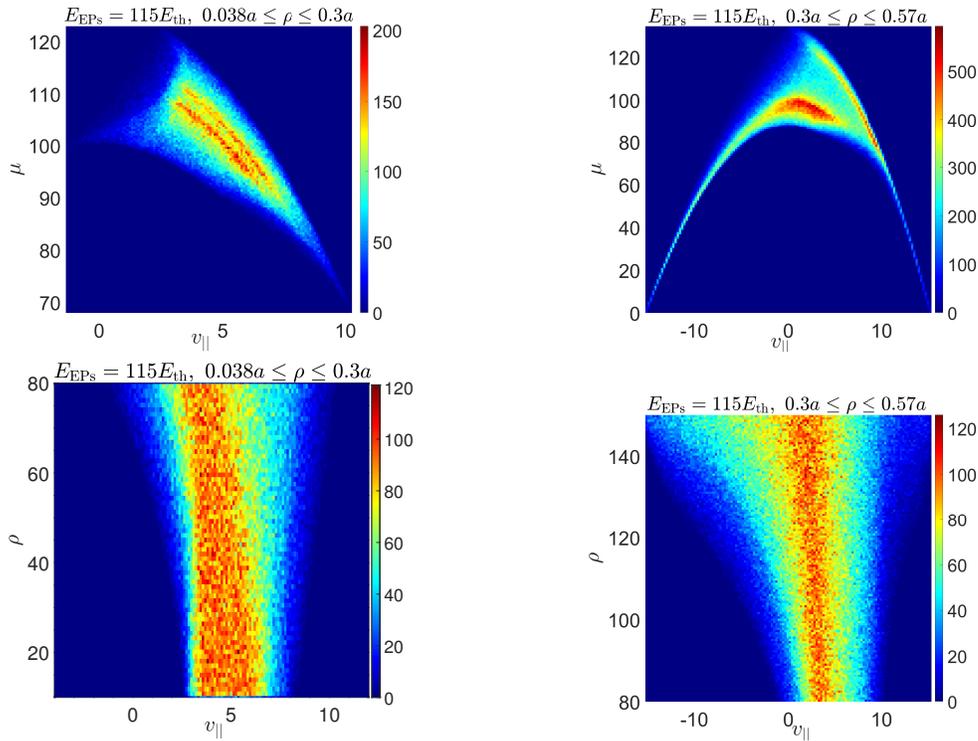


Figure 12: Top frames: Number of lost particles as a function of  $v_{\parallel}$  and  $\mu$  for initialization in the inner and outer regions in the left and right frames, respectively. Bottom frames: Number of lost particles as a function of  $v_{\parallel}$  and  $\rho$  for initialization in the inner and outer regions in the left and right frames, respectively.

the bottom-right frame, co-passing and counter-passing particles are now responsible for approximately 9% and 26.6% of the total losses, respectively. As expected, this is further supported by the results illustrated in the top frames of Fig.13 showing the fraction of lost particles as a function of their initial parallel velocity for both simulated energies.

A straightforward implication of assuming a strong guide field is the conservation of the magnetic moment. Since this quantity is proportional to the perpendicular kinetic energy of charged particles (i.e.,  $\mu = E_{k,\perp}/B$ ), plotting the 2D distribution of the number of lost particles as a function of their initial  $v_{\parallel}$  and  $\mu$  completes the picture. In the top frames of Fig.12, we plot, for  $E_{\text{EPs}} = 115E_{\text{th}}$ , the number of lost particles in the initial  $v_{\parallel} - \mu$  plane of the phase space for both radial initializations. The top-left frame corresponds to the initialization in the inner region, while the top-right frame illustrates the initialization in the outer region. In both frames, the majority of losses arise from trapped particles, and as expected, from those trapped particles with large perpendicular energy. This is easily understood since higher perpendicular energy implies a larger perpendicular velocity of particles, resulting in more drifts and wider orbit widths. Finally, the bottom frames of Fig.12 display the number of lost particles as a function of  $\rho$  and  $v_{\parallel}$ . These figures reveal an interesting property for both radial initializations:

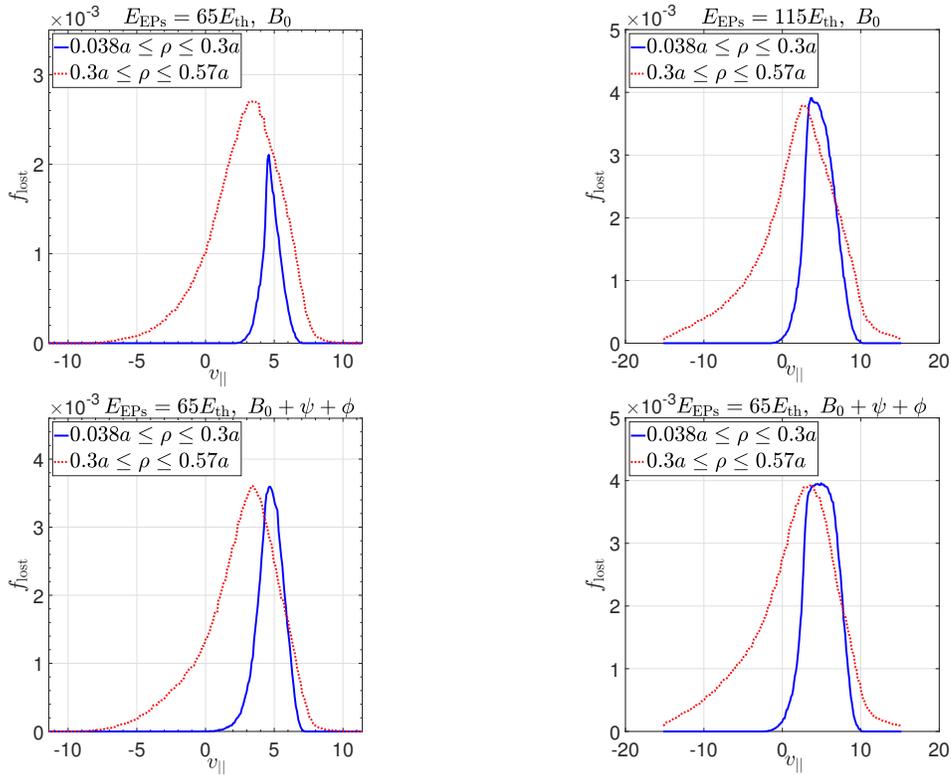


Figure 13: Fraction of lost particles at the end of various simulations, defined as  $f_{\text{lost}} = \frac{N_{\text{p,lost}}(t_{\text{end}})}{N_{\text{p}}}$ . The left frames correspond to lower energy, while the right ones highlights the results for the higher energy. In the top frames, only the magnetic equilibrium is considered, while in the bottom frames, electromagnetic perturbations were included.

particles initialized farther from the magnetic axis explore wider intervals along the  $v_{\parallel}$  direction in phase space.

In simulations involving collisions, we observed their overall impact on dynamics to be negligible for the simulated time window, as the collision frequency considered is three order of magnitude smaller than the reference frequency of TAPAS. Additionally, during extended simulations lasting for  $t = 2.6$  seconds with only the equilibrium included, the detrapping and trapping effects of collisions on particles were also observed. In other words, collisions change the type of particles by, for example, allowing for a diffusion-like region to exist close to the head of loss cone in velocity space, and thus detrapping the charged particle resulting in a co-passing or counter-passing particle. However, due to the time limit imposed by the nonlinear simulations presented in the previous section, exploring the long-term effects of collisions in the presence of electromagnetic perturbation will require performing longer nonlinear simulations with Far3D code.

In conclusion, prompt losses exhibit dependence on particle type, energy, and radial initialization. Lower-energy prompt losses are dominated by trapped particles for both radial initializations. Conversely, for higher energies, the contribution of co-passing

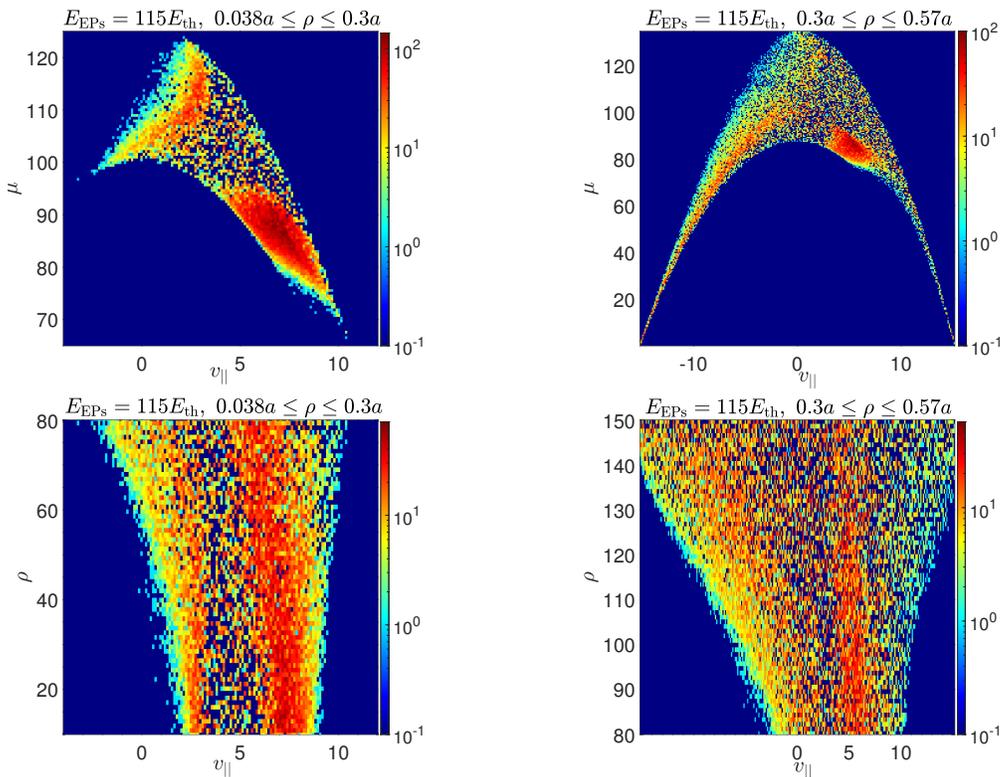


Figure 14: The impact of electromagnetic perturbations on particle losses at  $E_{\text{EPs}} = 115E_{\text{th}}$  presented as a function of  $v_{\parallel}$  and  $\mu$  in the top frames, and of  $v_{\parallel}$  and  $\rho$  in the bottom frames.

and counter-passing particles to the losses becomes more pronounced (see Fig.11) when particles are initialized farther from the magnetic axis. On the other hand, the effects of collisions in the investigated regime were found to be negligible due to their longer characteristic time and the duration of the time window simulated by the non-linear simulation.

#### 4.2. Transport and losses of energetic particles in the presence of Alfvén eigenmodes

In this section, our primary focus is to analyze the effects of including electromagnetic perturbations resulting from AEs activities on the overall losses of energetic particles. In particular, we investigate the losses that occur after the prompt losses are saturated, where first-orbit losses are no longer dominant; i.e., the electromagnetic perturbations are activated during this stage. To achieve this, a series of simulations was performed for two radial initializations:  $0.038a \leq \rho \leq 0.3a$  (inner region) and  $0.3a \leq \rho \leq 0.57a$  (outer region). These simulations involved a particle number  $N_p = 1.6 \times 10^6$  and a total duration of  $3000\tau_A$ . The beam energies and other initial parameters were maintained as described in the previous subsection 4.1, and electromagnetic perturbations were introduced in all simulations at  $t = 502\tau_A$ .

The effects of including electromagnetic perturbations on particle losses over time

are illustrated in Fig.14 for  $E_{\text{EPs}} = 65E_{\text{th}}$  and  $E_{\text{EPs}} = 115E_{\text{th}}$  in the left and right frames, respectively. For  $E_{\text{EPs}} = 65E_{\text{th}}$ , the total losses at the end of the simulations were 9.1% (blue curve) for the radial initialization in the inner region and 23% (red-dashed curve) for that in the outer region. In comparison to prompt losses, this implies that in this case, the increase in losses due to the presence of AEs activities is 5.8% and 5.2% for the radial initializations in the inner region and the outer region, respectively. On the another hand, the losses at the end of simulations for  $E_{\text{EPs}} = 115E_{\text{th}}$  (right frame of the figure) are 18.4% (blue curve) for  $0.038a \leq \rho \leq 0.3a$  with an increase of around 2.5% compared to prompt losses, and 37.3% (red-dashed curve) for  $0.3a \leq \rho \leq 0.57a$  with an increase of around 3.9% compared to the prompt losses. Another interesting feature is that the losses resulting from including electromagnetic perturbations for the case of low energy (left frame of Fig.14) are larger compared to prompt losses than in the case of higher energy (right frame of Fig.14).

To understand this result, the fraction of particle losses, defined as  $f_{\text{lost}} = \frac{N_{\text{p,lost}}(t_{\text{end}})}{N_{\text{p}}}$ , is plotted in the bottom frames of Fig.(13) as function of the initial values of  $v_{\parallel}$ . For the case of  $E_{\text{EPs}} = 65E_{\text{th}}$  shown in the bottom-left frame, it is evident that the values of  $v_{\parallel}$  around which the largest losses occur remain approximately equal to the case when only prompt losses exist (top-left frame). However, as shown in the bottom-left frame, the losses are both larger compared to those shown in the top-left frame and expand wider intervals of  $v_{\parallel}$ . The differences are more challenging to recognize for the  $E_{\text{EPs}} = 115E_{\text{th}}$  case shown in the right frames, where the largest values remain centered around the same values for both cases when including electromagnetic perturbations (bottom-right frame) and when they are absent (top-right frame). Moreover, the values of  $f_{\text{lost}}$  are slightly larger in the bottom-right frame, mainly for the radial initialization close to the magnetic axis (blue curve).

To further quantify the differences in losses due to the presence of AEs, we plot in Fig.14 on a log scale the number of lost particles resulting only from the inclusion of the perturbations at the end of simulations for  $E_{\text{EPs}} = 115E_{\text{th}}$  (i.e.  $N_{\text{p,lost EMs}} = N_{\text{p,total losses}} - f_{\text{p,prompt losses}}$ ). In the top frames, we show the projection  $N_{\text{p,lost EMs}}(v_{\parallel}, \mu)$  for  $0.038a \leq \rho \leq 0.3a$  and  $0.3a \leq \rho \leq 0.57a$  in the left and right frames, respectively, while in the bottom frames, we plot the corresponding  $N_{\text{p,lost EMs}}(v_{\parallel}, \rho)$ .

For the initialization closer to the magnetic axis (left frames), we notice that the main contribution to the losses in the presence of AEs is due to the trapped particles, as evident in the top-left frame. This result is consistent with the losses of different types of particles shown in the top-left frame of Fig.11, where only 15% of the trapped particles were lost (compare also with Fig.13). On the other hand, passing particles have a larger contribution to losses when initialized in the outer region, as illustrated in the top-right frame of Fig.14. This result can be understood for the scenarios discussed in this paper since lost particles explore a wider range of  $v_{\parallel}$  when initialized further from the magnetic axis, as shown in the bottom frames of Fig.14. The additional losses can then be partially attributed to passing particles whose first orbits intersect with the last available closed magnetic surface, a result that also applies when only the prompt losses

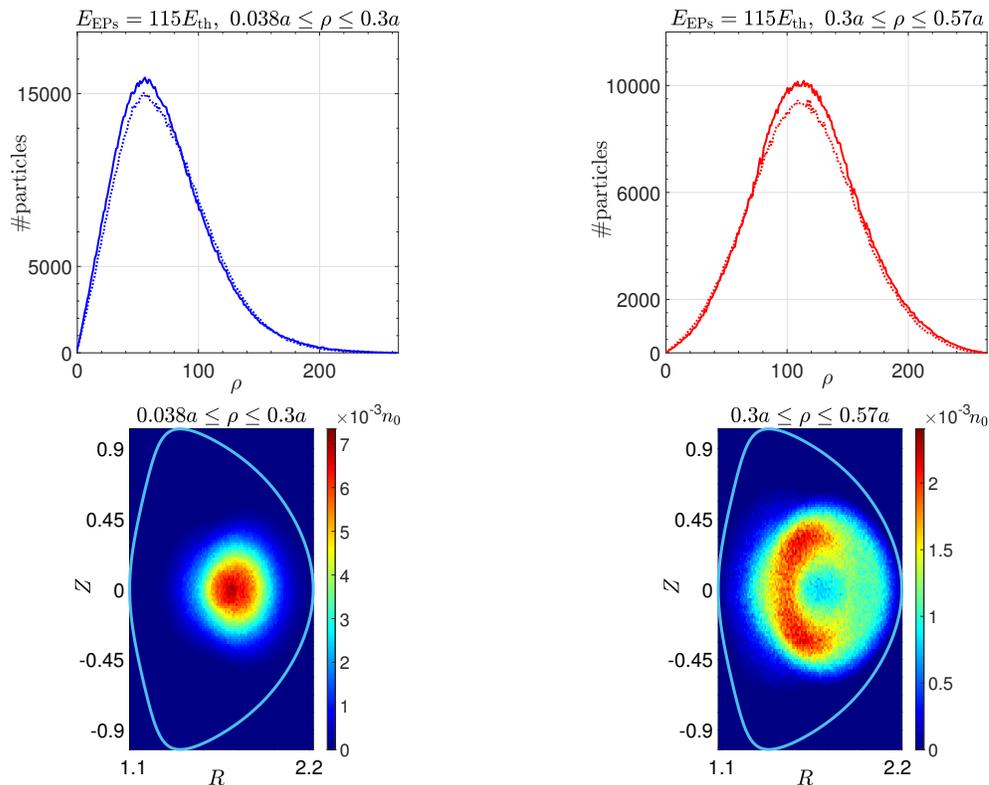


Figure 15: Top frames: the number of confined particles as a function of  $\rho$  for  $E_{\text{EPs}} = 115E_{\text{th}}$  (solid curves and dashed curves represent cases where only the magnetic equilibrium and when AEs are included, respectively). Bottom frames: the number density of confined particles at the end of the simulations in the presence of electromagnetic perturbation resulting from AEs activities.

are considered, as shown in Fig.11.

The influence of AEs on the spatial distribution of particles is illustrated in Fig.15. In the top frames, we show the number of confined particles at the end of the simulations as a function of the radius-like variable  $\rho$  for  $E_{\text{EPs}} = 115E_{\text{th}}$  and the two radial initializations considered in this paper. Solid lines represent the case where only the magnetic equilibrium is considered, while the dashed curves correspond to the simulations where electromagnetic perturbations were included. In the investigated scenarios of this paper, we notice that the effect of magnetic equilibrium is to redistribute the particles in the radial direction. It is evident then that the presence of AEs results in more losses, as shown by the dashed curves in the top frames. However, although these losses are small compared to the prompt losses (solid curves) as discussed above (c.f. right frame of Fig.10), we notice that the interval around the region with the maximum number of particles exhibits the largest losses since the fraction of particles with the largest energies exists in this region. Additionally, we recall that the duration of the AEs simulations performed here is fixed by the external simulation (i.e., in our case, Far3D code as discussed in section 3). The results reported in Fig.10 correspond

to Far3D simulations extending over  $2500\tau_A = 0.7$  msec, where in these simulations  $\tau_A = 2.97 \times 10^{-7}$ sec, which can be very short compared to the duration of an AEs pulse that can last for a fraction of second [19]. Therefore, with longer simulations, it is expected that the losses of energetic particles will possibly further increase due to the wave-particle interactions between energetic particles and the AEs.

Another way to visualize the spatial distribution of confined particles is by plotting the normalized energetic particle number density (i.e.,  $n_{EPs}(R, Z)$ ) in the poloidal plane at the end of simulations in which electromagnetic perturbations were included, as shown in the bottom frames of Fig.15. In the bottom-left frame of the figure,  $n_{EPs}$  is displayed for the radial initialization in the inner region, while the bottom-right frame presents  $n_{EPs}$  for the initialization in the outer region. These frames illustrate that the particles are redistributed in the poloidal plane, with the maximum value of the density, occurring approximately at the average value of the initial radial interval with larger values of density on the high-field side of the poloidal cross-section, a trend that becomes more pronounced when particles are initialized further from the magnetic axis.

## 5. Conclusions and future work

In this paper, we discuss the recent upgrades implemented in the TAPAS code. These updates involved developing a full-orbit version of the code to resolve particle dynamics in the  $6D$  phase space by implementing the Boris scheme to solve the equations of motion for charged particles. Additionally, the code was extended to handle realistic geometries with coordinate systems based on magnetic equilibrium obtained through approximate analytical solution (e.g., analytical solutions of the Grad-Shavranov equation using expansion methods [57]), numerical methods (e.g., employing the VMEC code[58]), or experimental data (e.g., fitting experimental data with TRANSP[59] and EFIT[60]). The new updates also included the implementation of the Metropolis-Hastings sampling method for initializing particles. Furthermore, collisions were introduced through an operator acting on particle velocity in phase space. These updates, associated with the newly implemented multi-dimensional bspline interpolation and a correction algorithm near the last available magnetic surface, enabled us to couple TAPAS with other codes, such as the gyro-fluid code Far3D[33] or the full-f gyrokinetic code GYSELA[37].

By coupling TAPAS with the Far3D code, we studied the transport and losses of EPs in the presence of Alfvén eigenmodes in the DIII-D tokamak. Far3D simulation was performed for  $\beta_{EP} = 0.03$ . Linear analysis revealed that within this phase, the  $n = 3$  EAE family at  $f \approx 160$  exhibited the maximum growth rate and prominently dominated by  $9/3$ ,  $10/3$ , and  $11/3$  modes; while the nonlinear phase, characterized by MHD burst events, possibly resulting from modes overlap, was primarily dominated by the  $9/3$  RSAE mode [43].

In TAPAS, the particles were mono-energetic and uniformly initialized in both toroidal and poloidal angles for two radial initializations:  $0.008a \leq \rho \leq 0.0285a$  and  $0.3a \leq \rho \leq 0.58a$ . We have shown that the prompt losses (i.e. first orbit losses) of

energetic particles are larger for higher energies and when initialized further from the magnetic axis. Moreover, as expected we have shown that the prompt losses strongly depend on the particle type, while the effect collisions on the losses was negligible for the time window simulated in this paper.

In the presence of AEs activities, we once again observed that the effects of collisions remain negligible. On the other hand, the losses of energetic particles increase with the inclusion of electromagnetic perturbations for both simulated beam energies. This rise in losses is primarily attributed to the enhancement in trapped particle losses (as classified with respect to equilibrium) and contributions resulting from passing particles, mainly in scenarios where particles are initialized in the outer region. Moreover, the ratio between losses resulting from AEs activities and the prompt losses was larger for energetic particles with lower energy since the prompt losses of trapped particles in this case were smaller compared to energetic particles with higher energy (c.f. Fig.11).

The simulations performed in this paper, aimed at testing the new tools developed in TAPAS to account for realistic geometries and to investigate energetic particle transport in 6D phase space by adopting simple uniform initialization of mono-energetic particles in phase space. Further simulations with more realistic particle initialization are necessary to model experiments. This includes simulating parallel and perpendicular Neutral Beam Injection (NBI) at specific toroidal angles, which will require employing the Metropolis algorithm to sample radial profiles of density and temperature for different species. Another crucial aspect to consider longer non-linear simulations to study the effect of collisions with the background plasma on the transport of energetic particles, as collisions play a important role in slowing down these particles and consequently affect their losses and transport, depending on their energy and the length of Alfvén eigenmode events. This will be explored in a forthcoming paper.

## **Acknowledgments**

The authors want to thank W.W. Heidbrink, M. A. Van Zeeland, and D. Del Sarto for useful discussions. This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them. This work has received financial support from the AIM4EP project (ANR-21-CE30-0018), funded by the French National Research Agency (ANR), from the Comunidad de Madrid under the project 2019-T1/AMB-13648, and from the Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under Contract No. DE-AC05-00OR22725. TAPAS simulations were performed on HPC resources of IDRIS under the allocations 2021-A0100512455, 2022-AD010512455R1 and 2023-A0140514165 made by GENCI. TAPAS development, optimization and deployment on GPU accelerator

were done within the framework of an advanced support program funded by IDRIS and GENCI.

## Appendix A. Coordinate transformations

The co- and contra-variant representations of a vector  $\mathbf{Q}$  read

$$\mathbf{Q} = Q^i \mathbf{e}_i \quad (\text{A.1a})$$

$$\mathbf{Q} = Q_i \nabla x^i \quad (\text{A.1b})$$

where  $\nabla x^i = \frac{\partial x^i}{\partial \mathbf{X}}$ ,  $\mathbf{e}_i = \frac{\partial \mathbf{X}}{\partial x^i}$ , and the co- and contra-variant components of  $\mathbf{Q}$  are linked to each other through the metric tensor  $\mathbf{g}$  by the relations

$$Q^i = g^{ij} Q_j \quad (\text{A.2a})$$

$$Q_i = g_{ij} Q^j \quad (\text{A.2b})$$

with the contra-variant and co-variant representations of the metric tensor are, respectively,

$$g^{ij} = \nabla x^i \cdot \nabla x^j \quad (\text{A.3a})$$

$$g_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j \quad (\text{A.3b})$$

where the Einstein's notation over the repeated indexes is implicit. The norm of  $\mathbf{e}_i$  does not necessarily equal one. Therefore, we use the notation  $\hat{\mathbf{e}}_i$  to denote the unit vector  $\mathbf{e}_i / \|\mathbf{e}_i\|$ . Knowing the metric tensor and one of the two representations of  $\mathbf{Q}$ , one can use Eqs. (A.1) to find the other representation. In the most general case where the equilibrium magnetic field is 3D and  $\varphi$  does not necessarily coincide with  $x^3$ , the Jacobian matrix of the transformation between cylindrical and generalized coordinates read

$$[J] = \begin{bmatrix} \partial_{x^1} R & \partial_{x^2} R & \partial_{x^3} R \\ \partial_{x^1} Z & \partial_{x^2} Z & \partial_{x^3} Z \\ \partial_{x^1} \varphi & \partial_{x^2} \varphi & \partial_{x^3} \varphi \end{bmatrix} \quad (\text{A.4})$$

The Jacobian matrix of the inverse transformation straightforwardly reads

$$[J]^{-1} = \frac{1}{J} \begin{bmatrix} \partial_{x^2} Z \partial_{x^3} \varphi - \partial_{x^3} Z \partial_{x^2} \varphi & \partial_{x^3} R \partial_{x^2} \varphi - \partial_{x^2} R \partial_{x^3} \varphi & \partial_{x^2} R \partial_{x^3} Z - \partial_{x^3} R \partial_{x^2} Z \\ \partial_{x^3} Z \partial_{x^1} \varphi - \partial_{x^1} Z \partial_{x^3} \varphi & \partial_{x^1} R \partial_{x^3} \varphi - \partial_{x^3} R \partial_{x^1} \varphi & \partial_{x^1} Z \partial_{x^3} R - \partial_{x^3} Z \partial_{x^1} R \\ \partial_{x^1} Z \partial_{x^2} \varphi - \partial_{x^2} Z \partial_{x^1} \varphi & \partial_{x^2} R \partial_{x^1} \varphi - \partial_{x^1} R \partial_{x^2} \varphi & \partial_{x^1} R \partial_{x^2} Z - \partial_{x^2} R \partial_{x^1} Z \end{bmatrix} \quad (\text{A.5})$$

where  $J = \det [J]$ . From the chain rule, one easily finds

$$\begin{bmatrix} dR \\ dZ \\ d\varphi \end{bmatrix} = [J] \begin{bmatrix} dx^1 \\ dx^2 \\ dx^3 \end{bmatrix}, \quad \begin{bmatrix} dx^1 \\ dx^2 \\ dx^3 \end{bmatrix} = [J]^{-1} \begin{bmatrix} dR \\ dZ \\ d\varphi \end{bmatrix} \quad (\text{A.6})$$

Assuming  $(x^1, x^2, x^3) = (\rho, \theta, \varphi)$  and a 2D equilibrium, we have  $\partial_{x^3} = 0$ ,  $\partial_{x^1}\varphi = \partial_{x^2}\varphi = 0$  and  $\partial_{x^3}\varphi = 1$ . The Jacobian matrix of the inverse transformation therefore reduces to

$$[J]^{-1} = \begin{bmatrix} [J_{\text{red}}]^{-1} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (\text{A.7})$$

with

$$[J_{\text{red}}]^{-1} = \frac{1}{J} \begin{bmatrix} \partial_\theta Z & -\partial_\theta R \\ -\partial_\rho Z & \partial_\rho R \end{bmatrix} \quad (\text{A.8})$$

where  $J = \partial_\rho R \partial_\theta Z - \partial_\rho Z \partial_\theta R$ . Owing to the relation  $dx^i = \nabla x^i \cdot d\mathbf{X}$ , one can write

$$\nabla \rho = \frac{1}{J} (\partial_\theta Z \nabla R - \partial_\theta R \nabla Z) \quad (\text{A.9})$$

$$\nabla \theta = \frac{1}{J} (-\partial_\rho Z \nabla R + \partial_\rho R \nabla Z) \quad (\text{A.10})$$

In the cylindrical coordinate system, the contra-variant and co-variant basis vectors coincide with each other along the  $R$  and  $Z$  directions, i.e.  $\mathbf{e}_R = \nabla R$ ,  $\mathbf{e}_Z = \nabla Z$ , whereas  $\mathbf{e}_\varphi = R \nabla \varphi$ . In TAPAS, when the geometry  $(R(\rho, \theta, \varphi)$  and  $Z(\rho, \theta, \varphi))$  is obtained from another numerical code such as VMEC, the contra-variant and co-variant metrics can be calculated using Eqs. (A.3) and Eqs (A.9)-(A.10). However, it is important to note that for 3D equilibria, such as in the case of stellarators, the  $\partial_{x^3}$  terms do not vanish. As a result, the inverse Jacobian is no longer a block-diagonal matrix as shown in Eq.(??). Eqs (A.9)-(A.10) would need to be modified accordingly to account for this additional dependence on the  $x^3$  coordinate. This modification ensures accurate calculations and consistent representation of the magnetic equilibrium in the 3D case.

Since in TAPAS, the equations of motion are solved in Cartesian coordinates, one has to obtain the  $j$ -th Cartesian component of a vector  $\mathbf{Q}$ . Eq.(A.1b) must be multiplied by the corresponding unit vector, denoted as  $\mathbf{e}_j$ , where  $j$  can be either  $x$ , or  $y$ , or  $z$ . The transformation can be expressed as follows:

$$Q_j = Q_i \nabla x^i \cdot \mathbf{e}_j \quad (\text{A.11})$$

where  $x^i = \rho, \theta, \varphi$ . The Cartesian unit vectors are given in cylindrical coordinates by

$$\mathbf{e}_x = \sin \varphi \mathbf{e}_R + \cos \varphi \mathbf{e}_\varphi, \quad \mathbf{e}_y = \cos \varphi \mathbf{e}_R - \sin \varphi \mathbf{e}_\varphi, \quad \mathbf{e}_z = \mathbf{e}_Z \quad (\text{A.12})$$

where  $\mathbf{e}_\varphi = R \nabla \varphi$ . Therefore, substituting Eq.(A.12) in Eq.(A.11), and using Eqs. (A.9)-(??), one finds, for 2D equilibria, the following transformation equations

$$Q_x = \frac{\sin \varphi}{J} \partial_\theta Z Q_\rho - \frac{\sin \varphi}{J} \partial_\rho Z Q_\theta + \frac{\cos \varphi}{R} Q_\varphi \quad (\text{A.13})$$

$$Q_y = \frac{\cos \varphi}{J} \partial_\theta Z Q_\rho - \frac{\cos \varphi}{J} \partial_\rho Z Q_\theta - \frac{\sin \varphi}{R} Q_\varphi \quad (\text{A.14})$$

$$Q_z = -\frac{\partial_\theta R}{J} Q_\rho + \frac{\partial_\rho R}{J} Q_\theta \quad (\text{A.15})$$

The aforementioned transformations have been implemented in TAPAS, enabling the computation of the Cartesian components of the electromagnetic fields necessary for solving Eq.(4) .

## Appendix B. Interpolation methods

### Appendix B.1. Multi-dimensional Lagrange interpolation

In TAPAS, we implement both linear and cubic Lagrange interpolation for both 1D and 2D cases. A general formula for Lagrange interpolation of order  $q$  in an  $n$ -dimensional space can be written as follows:

$$f(x_{1,p}, x_{2,p}, \dots, x_{n,p}) = \sum_{i_1=1}^{q+1} \dots \sum_{i_n=1}^{q+1} \mathcal{L}_{i_1}(x_{1,p}) \dots \mathcal{L}_{i_n}(x_{n,p}) f(x_{1,i_1}, \dots, x_{n,i_n}) \quad (\text{B.1})$$

where

$$\mathcal{L}_{i_l}(x_{l,p}) = \prod_{k \neq i_l}^q \frac{x_{l,p} - x_{l,k}}{x_{l,i_l} - x_{l,k}} \quad (\text{B.2})$$

For example, in 2D space and for cubic interpolation, we substitute  $q = 3$  and  $n = 2$  into the previous equation, yielding:

$$f(x_p, y_p) = \sum_{i=1}^4 \dots \sum_{j=1}^4 \mathcal{L}_i(x_p) \dots \mathcal{L}_j(y_p) f(x_i, y_j) \quad (\text{B.3})$$

with

$$\mathcal{L}_i(x_p) = \prod_{k \neq i}^q \frac{x_p - x_k}{x_i - x_k} \quad (\text{B.4})$$

In TAPAS, Lagrange interpolation is employed for both 1D and 2D cases, primarily for interpolating equilibrium quantities at the particle position. Additionally, it is used to adjust the grid in the cylindrical coordinate system to avoid interpolation errors arising from the presence of some interpolation nodes outside the physical domain, specifically beyond the last closed magnetic surface, where certain fields are not defined. This approach guarantees accurate interpolation of the necessary quantities across the entire computational domain, enhancing the overall reliability of the simulations. However, it should be noted that the accuracy of the interpolation scheme can have a significant impact on the trajectory of the particle. Inaccuracies introduced by using a low-order interpolation method can lead to numerical drift, which in certain scenarios can result in particle losses close to the last closed surface. Therefore, choosing an appropriate interpolation method with sufficient accuracy is crucial for maintaining the integrity of particle trajectories in the simulation.

### Appendix B.2. Multi-dimensional bspline interpolation

As mentioned earlier in this section, in addition to cubic Lagrange interpolation (given by Eq.(B.1)), TAPAS also offers the option of using B-spline interpolation in 1D, 2D, and 3D. B-spline interpolation is primarily employed for interpolating electromagnetic perturbations obtained through the coupling of TAPAS with another code. This choice ensures numerical stability and accurate integration of particle trajectories.

Similar to Lagrange interpolation, B-spline interpolation can be formulated in a tensorial form for a space of  $n$  dimensions. However, for the purposes of this discussion, we will focus on the 2D case as it provides a compromise between the simplicity of 1D and the complexity of 3D, while still covering the various aspects of the method. In 2D space, the interpolated field  $f$  at the particle position  $(x_p, y_p)$  can be expressed as follows:

$$f(x_p, y_p) \approx \sum_{i=-3}^{N_x-2} \sum_{j=-3}^{N_y-2} \eta_{i,j} \mathcal{B}_{i,3}(x_p) \mathcal{B}_{j,3}(y_p) \quad (\text{B.5})$$

where  $\mathcal{B}_{i,3}$  and  $\mathcal{B}_{j,3}$  are the B-spline basis functions along  $x$  and  $y$  directions, respectively, and  $\eta_{i,j}$  are the  $(N_x + 2)(N_y + 2)$  B-spline coefficients that need to be calculated beforehand. It is important to note that the previous definition does not assume any specific order of interpolation, and all points in the computational domain can in principle be used when appropriate boundary conditions are considered. In TAPAS, we implement cubic B-spline interpolation. The basis function along a given direction, such as  $x$ , is expressed as follows:

$$\mathcal{B}_{i,3}(x) = \frac{1}{(\Delta x)^3} \begin{cases} (x - x_i)^3, & x_i \leq x < x_{i+1} \\ (\Delta x)^3 + 3(\Delta x)^2(x - x_{i+1}) + 3(\Delta x)(x - x_{i+1})^2 - 3(x - x_{i+1})^3, & x_{i+1} \leq x < x_{i+2} \\ (\Delta x)^3 + 3(\Delta x)^2(x_{i+3} - x) + 3(\Delta x)(x_{i+3} - x)^2 - 3(x_{i+3} - x)^3, & x_{i+2} \leq x < x_{i+3} \\ (x_{i+4} - x)^3, & x_{i+3} \leq x < x_{i+4} \\ 0, & \text{otherwise} \end{cases} \quad (\text{B.6})$$

The tensor  $\eta_{i,j}$  represents the projection coefficients tensor of rank  $2^d$  that needs to be computed during the update of the electromagnetic fields. To determine the elements of this tensor, a total of  $(N_x + 2) \times (N_y + 2)$  equations are required. Among these equations,  $(N_x \times N_y)$  are obtained from the values of the function at the grid points. The remaining equations arise from the boundary conditions, which can be either periodic (e.g., in the toroidal and poloidal directions) or natural (e.g., in the radial direction). In the case of natural boundary conditions, the second derivative of the approximated function vanishes at the boundaries.

In TAPAS, interpolation methods used for time and space are different. While cubic B-spline interpolation is available for temporal interpolation, linear interpolation is employed for the fields at the particle position in time. On the other hand, cubic B-spline interpolation is utilized for spatial interpolation. In TAPAS, the electromagnetic fields are read from external simulation at two time steps  $t_s$  and  $t_e$ , while the particle exists at time  $t_n$  with  $t_s < t_n < t_e$ . To correctly interpolate the fields at the particle position at  $t_n$ , the following procedure is followed:

- The fields at the particle position  $\mathbf{x}$  are interpolated using cubic B-spline interpolation at both  $t_s$  and  $t_e$ . This step ensures that the fields are accurately represented at those time instances.

- Once the fields are interpolated at  $\mathbf{x}$  for  $t_s$  and  $t_e$ , they are linearly interpolated at  $t_n$  using the field values at  $t_s$  and  $t_e$ . This linear interpolation allows for estimating the fields at the specific time  $t_n$  when the particle exists.

By performing these interpolation steps, TAPAS effectively combines cubic B-spline interpolation in space and linear interpolation in time to obtain the fields at the particle's position and time.

### Appendix C. Brief overview of FAR3d

The nonlinear version of FAR3d code solves a system of equations for both the thermal plasma and an arbitrary number of energetic particle populations. Regarding the thermal plasma, the code evolves nonlinear equations for the perturbed fields: Ohm's law for the perturbed magnetic stream function  $\psi$ , an equation for the parallel component of the vorticity  $U$ , and another two for both the perturbed pressure  $p$  and the parallel velocity of the plasma  $v_{\parallel}$ , where the model assumes  $\mathbf{v}_{E \times B}$  to be the dominant plasma velocity. For the energetic particles, the code solves two additional equations for each particle population. These equations are derived by taking the first two moments of the gyrokinetic equation, allowing for the inclusion of wave-particle interactions *via* a two-pole energetic ion closure model [31], resulting in equations for the parallel velocity  $v_{EP,\parallel}$  and density  $n_{EP}$  of the energetic particles.

FAR3d makes use of Fourier decomposition of various fields on rational magnetic surfaces in both toroidal and poloidal directions within the Boozer coordinate system [48, 49]. Subsequently, a nonlinear evolution equation is solved for each mode using an implicit multi-step finite difference method in the radial direction. The nonlinearity inherent in the original system of equations in real space is incorporated through the convolution terms present in the equations for each mode. These convolution terms account for the interaction of the corresponding mode with all the other remaining modes.

The equilibrium fields, including the magnetic field and magnetic flux surfaces, are obtained through numerical calculations by solving the ideal magnetohydrodynamic (MHD) equation using the VMEC code [58] that implements a variational method to minimize the total energy of the plasma, resulting in the determination of the magnetic flux surfaces. FAR3d then establishes the Boozer coordinate system and evolves various perturbed quantities according to the chosen equilibrium profiles. The selection of poloidal and toroidal mode numbers ( $m/n$ ) that determine the modes evolved by the code is done in a way that is compatible with the safety factor profile involved in the equilibrium calculation. Additionally, the code has the capability to read initial and equilibrium values for different fields from an external file. These values can be obtained through analytical methods, experimental measurements, or using codes such as TRANSP [59] that allows for the determination of profiles and transport coefficients by combining experimental results with the solution of particle and energy balance equations.

## References

- [1] K. McGuire, R. Goldston, M. Bell, M. Bitter, K. Bol, K. Brau, D. Buchenauer, T. Crowley, S. Davis, F. Dylla, H. Eubank, H. Fishman, R. Fonck, B. Grek, R. Grimm, R. Hawryluk, H. Hsuan, R. Hulse, R. Izzo, R. Kaita, S. Kaye, H. Kugel, D. Johnson, J. Manickam, D. Manos, D. Mansfield, E. Mazzucato, R. McCann, D. McCune, D. Monticello, R. Motley, D. Mueller, K. Oasa, M. Okabayashi, K. Owens, W. Park, M. Reusch, N. Sauthoff, G. Schmidt, S. Sesnic, J. Strachan, C. Surko, R. Slusher, H. Takahashi, F. Tenney, P. Thomas, H. Towner, J. Valley, and R. White. Study of high-beta magnetohydrodynamic modes and fast-ion losses in pdx. *Phys. Rev. Lett.*, 50:891–895, Mar 1983.
- [2] W.W. Heidbrink and G. Sager. The fishbone instability in the diii-d tokamak. *Nuclear Fusion*, 30(6):1015, jun 1990.
- [3] H.H. Duong, W.W. Heidbrink, E.J. Strait, T.W. Petrie, R. Lee, R.A. Moyer, and J.G. Watkins. Loss of energetic beam ions during tae instabilities. *Nuclear Fusion*, 33(5):749, may 1993.
- [4] H.H. Duong and W.W. Heidbrink. Confinement of fusion produced mev ions in the diii-d tokamak. *Nuclear Fusion*, 33(2):211, feb 1993.
- [5] K Appert, R Gruber, F Troyuon, and J Vaclavik. Excitation of global eigenmodes of the alfvén wave in tokamaks. *Plasma Physics*, 24(9):1147, sep 1982.
- [6] T. E. Evans, P. M. Valanju, J. F. Benesch, Roger D. Bengtson, Y. M. Li, S. M. Mahajan, M. E. Oakes, D. W. Ross, X. Z. Wang, J. G. Watkins, and C. M. Surko. Direct observation of the structure of global alfvén eigenmodes in a tokamak plasma. *Phys. Rev. Lett.*, 53:1743–1746, Oct 1984.
- [7] G. Y. Fu and J. W. Van Dam. Title of the article. *Phys. Fluids B*, 1:1919, 1989.
- [8] N.N. Gorelenkov, E. Fredrickson, E. Belova, C.Z. Cheng, D. Gates, S. Kaye, and R. White. Theory and observations of high frequency alfvén eigenmodes in low aspect ratio plasmas. *Nuclear Fusion*, 43(4):228, mar 2003.
- [9] C. Z. Cheng and M. S. Chance. Low-n shear Alfvén spectra in axisymmetric toroidal plasmas. *The Physics of Fluids*, 29(11):3695–3701, 11 1986.
- [10] H. Kimura, Y. Kusama, M. Saigusa, G.J. Kramer, K. Tobita, M. Nemoto, T. Kondoh, T. Nishitani, O. Da Costa, T. Ozeki, T. Oikawa, S. Moriyama, A. Morioka, G.Y. Fu, C.Z. Cheng, and V.I. Afanas’ev. Alfvén eigenmode and energetic particle research in jt-60u. *Nuclear Fusion*, 38(9):1303, sep 1998.
- [11] R. Betti and J. P. Freidberg. Ellipticity induced Alfvén eigenmodes. *Physics of Fluids B: Plasma Physics*, 3(8):1865–1870, 08 1991.
- [12] R. Betti and J. P. Freidberg. Stability of Alfvén gap modes in burning plasmas. *Physics of Fluids B: Plasma Physics*, 4(6):1465–1474, 06 1992.
- [13] G. J. Kramer, M. Saigusa, T. Ozeki, Y. Kusama, H. Kimura, T. Oikawa, K. Tobita, G. Y. Fu, and C. Z. Cheng. Noncircular triangularity and ellipticity-induced alfvén eigenmodes observed in jt-60u. *Phys. Rev. Lett.*, 80:2594–2597, Mar 1998.
- [14] A. D. Turnbull, E. J. Strait, W. W. Heidbrink, M. S. Chu, H. H. Duong, J. M. Greene, L. L. Lao, T. S. Taylor, and S. J. Thompson. Global Alfvén modes: Theory and experiment\*. *Physics of Fluids B: Plasma Physics*, 5(7):2546–2553, 07 1993.
- [15] W. Deng, Z. Lin, I. Holod, Z. Wang, Y. Xiao, and H. Zhang. Linear properties of reversed shear alfvén eigenmodes in the diii-d tokamak. *Nuclear Fusion*, 52(4):043006, mar 2012.
- [16] D.A. Spong. Simulation of alfvén frequency cascade modes in reversed shear-discharges using a landau-closure model. *Nuclear Fusion*, 53(5):053008, apr 2013.
- [17] M. S. Chu, J. M. Greene, L. L. Lao, A. D. Turnbull, and M. S. Chance. A numerical study of the high-n shear Alfvén spectrum gap and the high-n gap mode. *Physics of Fluids B: Plasma Physics*, 4(11):3713–3721, 11 1992.
- [18] WW Heidbrink. Basic physics of alfvén instabilities driven by energetic particles in toroidally confined plasmas. *Physics of Plasmas*, 15(5), 2008.

- [19] S. Sharapov. *Energetic Particles in Tokamak Plasmas*. CRC Press, 1st edition, 2021.
- [20] W.W. Heidbrink, E.J. Strait, E. Doyle, G. Sager, and R.T. Snider. An investigation of beam driven alfvén instabilities in the diii-d tokamak. *Nuclear Fusion*, 31(9):1635, sep 1991.
- [21] S.E. Sharapov, D. Borba, A. Fasoli, W. Kerner, L.-G. Eriksson, R.F. Heeter, G.T.A. Huysmans, and M.J. Mantsinen. Stability of alpha particle driven alfvén eigenmodes in high performance jet dt plasmas. *Nuclear Fusion*, 39(3):373, mar 1999.
- [22] R. J. Dumont, J. Mailloux, V. Aslanyan, M. Baruzzo, C.D. Challis, I. Coffey, A. Czarnecka, E. Delabie, J. Eriksson, J. Faustin, J. Ferreira, M. Fitzgerald, J. Garcia, L. Giacomelli, C. Giroud, N. Hawkes, Ph. Jacquet, E. Joffrin, T. Johnson, D. Keeling, D. King, V. Kiptily, B. Lomanowski, E. Lerche, M. Mantsinen, L. Meneses, S. Menmuir, K. McClements, S. Moradi, F. Nabais, M. Nocente, A. Patel, H. Patten, P. Puglia, R. Scannell, S. Sharapov, E. R. Solano, M. Tsalas, P. Vallejos, H. Weisen, and JET contributors. Scenario development for the observation of alpha-driven instabilities in jet dt plasmas. *Nuclear Fusion*, 58(8):082005, jun 2018.
- [23] A. Fasoli, C. Gormenzano, H.L. Berk, B. Breizman, S. Briguglio, D.S. Darrow, N. Gorelenkov, W.W. Heidbrink, A. Jaun, S.V. Konovalov, R. Nazikian, J.-M. Noterdaeme, S. Sharapov, K. Shinohara, D. Testa, K. Tobita, Y. Todo, G. Vlad, and F. Zonca. Chapter 5: Physics of energetic ions. *Nuclear Fusion*, 47(6):S264, jun 2007.
- [24] M. Fitzgerald, S.E. Sharapov, P. Siren, E. Tholerus, M. Dreval, G. Szepesi, P. Vallejos, T. Johnson, N. Fil, J. Ferreira, P. Rodrigues, A. Figueiredo, D. Borba, R. Coelho, F. Nabais, J. Mailloux, H.J.C. Oliver, C. Di Troia, F. Napoli, Ž. Štancar, R. Dumont, D. Keeling, and JET Contributors. Toroidal alfvén eigenmode stability in jet internal transport barrier afterglow experiments. *Nuclear Fusion*, 62(10):106001, aug 2022.
- [25] E.M. Carolipio, W.W. Heidbrink, C.B. Forest, and R.B. White. Simulations of beam ion transport during tearing modes in the diii-d tokamak. *Nuclear Fusion*, 42(7):853, jul 2002.
- [26] M. García-Muñoz, P. Martin, H.-U. Fahrbach, M. Gobbin, S. Günter, M. Maraschek, L. Marrelli, H. Zohm, and the ASDEX Upgrade Team. Ntm induced fast ion losses in asdex upgrade. *Nuclear Fusion*, 47(7):L10, jun 2007.
- [27] H. Cai and D. Li. Recent progress in the interaction between energetic particles and tearing modes. *Natl Sci Rev*, 9:nwac019, 2022.
- [28] Harold P. Furth, John Killeen, and Marshall N. Rosenbluth. Finite-Resistivity Instabilities of a Sheet Pinch. *The Physics of Fluids*, 6(4):459–484, 04 1963.
- [29] P. N. Guzdar, Liu Chen, W. M. Tang, and P. H. Rutherford. Ion-temperature-gradient instability in toroidal plasmas. *The Physics of Fluids*, 26(3):673–677, 03 1983.
- [30] Y. Todo. Introduction to the interaction between energetic particles and alfvén eigenmodes in toroidal plasmas. *Reviews of Modern Plasma Physics*, 3(1):1, 2018.
- [31] C. L. Hedrick, J.-N. Leboeuf, and D. A. Spong. Alpha-Alfvén local dispersion relation and solutions. *Physics of Fluids B: Plasma Physics*, 4(12):3869–3882, 12 1992.
- [32] D. A. Spong, B. A. Carreras, and C. L. Hedrick. Linearized gyrofluid model of the alpha-destabilized toroidal Alfvén eigenmode with continuum damping effects. *Physics of Fluids B: Plasma Physics*, 4(10):3316–3328, 10 1992.
- [33] J. Varela, D.A. Spong, and L. Garcia. Analysis of alfvén eigenmodes destabilization by energetic particles in tj-ii using a landau-closure modela. *Nuclear Fusion*, 57(12):126019, sep 2017.
- [34] D.A. Spong, M.A. Van Zeeland, W.W. Heidbrink, X. Du, J. Varela, L. Garcia, and Y. Ghai. Nonlinear dynamics and transport driven by energetic particle instabilities using a gyro-landau closure model\*. *Nuclear Fusion*, 61(11):116061, oct 2021.
- [35] Y. Todo and T. Sato. Linear and nonlinear particle-magnetohydrodynamic simulations of the toroidal Alfvén eigenmode. *Physics of Plasmas*, 5(5):1321–1327, 05 1998.
- [36] Y. Todo. Properties of energetic-particle continuum modes destabilized by energetic ions with beam-like velocity distributions. *Physics of Plasmas*, 13(8):082503, 08 2006.
- [37] V. Grandgirard, J. Abiteboul, J. Bigot, T. Cartier-Michaud, N. Crouseilles, G. Dif-Pradalier, Ch. Ehrlacher, D. Esteve, X. Garbet, Ph. Ghendrih, G. Latu, M. Mehrenberger, C. Nordscini,

- Ch. Passeron, F. Rozar, Y. Sarazin, E. Sonnendrücker, A. Strugarek, and D. Zarzoso. A 5d gyrokinetic full-f global semi-lagrangian code for flux-driven ion turbulence simulations. *Computer Physics Communications*, 207:35–68, 2016.
- [38] E. Lanti, N. Ohana, N. Tronko, T. Hayward-Schneider, A. Bottino, B.F. McMillan, A. Mishchenko, A. Scheinberg, A. Biancalani, P. Angelino, S. Brunner, J. Dominski, P. Donnel, C. Gheller, R. Hatzky, A. Jocksch, S. Jolliet, Z.X. Lu, J.P. Martin Collar, I. Novikau, E. Sonnendrücker, T. Vernay, and L. Villard. Orb5: A global electromagnetic gyrokinetic code using the pic approach in toroidal geometry. *Computer Physics Communications*, 251:107072, 2020.
- [39] Eero Hirvijoki, Otto Asunta, Tuomas Koskela, Taina Kurki-Suonio, Juho Miettunen, Seppo Sipilä, Antti Snicker, and Simppa Äkäslompolo. Ascot: Solving the kinetic equation of minority particle species in tokamak plasmas. *Computer Physics Communications*, 185(4):1310–1321, 2014.
- [40] SH Ward, Rob Akers, AS Jacobsen, P Ollus, SD Pinches, E Tholerus, RGL Vann, and MA Van Zeeland. Verification and validation of the high-performance lorentz-orbit code for use in stellarators and tokamaks (locust). *Nuclear Fusion*, 61(8):086029, 2021.
- [41] L Carbajal, D del Castillo-Negrete, D Spong, S Seal, and L Baylor. Space dependent, full orbit effects on runaway electron dynamics in tokamak plasmas. *Physics of Plasmas*, 24(4):042512, 2017.
- [42] David Zarzoso, Diego del Castillo-Negrete, Rémi Lacroix, Pierre-Eric Bernard, and Stanislas Touzet. Transport and losses of fusion-born alpha particles in the presence of tearing modes using the new toroidal accelerated particle simulator (tapas). *Plasma Physics and Controlled Fusion*, 64(4):044003, 2022.
- [43] J Varela, D A Spong, L Garcia, Y Ghai, D Zarzoso, D del Castillo-Negrete, H Betar, J Ortiz, D C Pace, M A Van Zeeland, X Du, R Sanchez, V Tribaldos, and J M Reynolds-Barredo. Effect of the neutral beam injector operational regime on the alfvén eigenmode saturation phase in diii-d plasma. *Plasma Physics and Controlled Fusion*, 65(12):125004, oct 2023.
- [44] HIRSHMAN S. P. and Whitson J. C. *The Physics of fluids*, 26(12):3553–3568, 1983.
- [45] PERAZA-RODRIGUEZ H. et al. *Physics of Plasmas*, 24(8), 2017.
- [46] PANICI D. et al. *Journal of Plasma Physics*, 89(3):955890303, 2023.
- [47] HIRSHMAN S. P. et al. *Physics of Plasmas*, 18(6), 2011.
- [48] A H Boozer. Establishment of magnetic coordinates for a given magnetic field. 4 1981.
- [49] Allen H. Boozer. Establishment of magnetic coordinates for a given magnetic field. *The Physics of Fluids*, 25(3):520–521, 03 1982.
- [50] Julien Derouillat, Arnaud Beck, Frédéric Pérez, Tommaso Vinci, M Chiaramello, Anna Grassi, M Flé, Guillaume Bouchard, I Plotnikov, Nicolas Aunai, et al. Smilei: A collaborative, open-source, multi-purpose particle-in-cell code for plasma simulation. *Computer Physics Communications*, 222:351–373, 2018.
- [51] Jean-Luc Vay, Irving Haber, and Brendan B Godfrey. A domain decomposition method for pseudo-spectral electromagnetic simulations of plasmas. *Journal of Computational Physics*, 243:260–268, 2013.
- [52] Adam V Higuera and John R Cary. Structure-preserving second-order integration of relativistic charged particle trajectories in electromagnetic fields. *Physics of Plasmas*, 24(5), 2017.
- [53] Takayuki Umeda. A new integrator for relativistic e-cross-b motion of charged particles. *Journal of Computational Physics*, 472:111694, 2023.
- [54] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.
- [55] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [56] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.
- [57] J. P. Hans Goedbloed and Stefaan Poedts. *Principles of Magnetohydrodynamics: With*

- Applications to Laboratory and Astrophysical Plasmas.* Cambridge University Press, 2004.
- [58] S. P. Hirshman and J. C. Whitson. Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria. *The Physics of Fluids*, 26(12):3553–3568, 12 1983.
- [59] Joshua Breslau, Marina Gorelenkova, Francesca Poli, Jai Sachdev, Alexei Pankin, Gopan Perumpilly, Xingqiu Yuan, and Laszlo Glant. *Transp*, jun 2018.
- [60] O. Meneghini, S.P. Smith, L.L. Lao, O. Izacard, Q. Ren, J.M. Park, J. Candy, Z. Wang, C.J. Luna, V.A. Izzo, B.A. Grierson, P.B. Snyder, C. Holland, J. Penna, G. Lu, P. Raum, A. McCubbin, D.M. Orlov, E.A. Belli, N.M. Ferraro, R. Prater, T.H. Osborne, A.D. Turnbull, G.M. Staebler, and The ATOM Team. Integrated modeling applications for tokamak experiments with omfit. *Nuclear Fusion*, 55(8):083008, jul 2015.