Representation and modeling of charged particle distributions in tokamaks

Andreas Bierwage1, Michael Fitzgerald2, Philipp Lauber2, Mirko Salewski4, Yevgen Kazakov3, and Žiga Štancar6,2
1QST, Rokkasho and Naka Fusion Institutes, Japan
2Culham Centre for Fusion Energy of UKAEA, Culham Science Centre, Abingdon, United Kingdom
3Max-Planck-Institut für Plasmaphysik, Garching, Germany
4Department of Physics, Technical University of Denmark, Kgs. Lyngby, Denmark
5Laboratory for Plasma Physics, LPP-ERM/KMS, Partner in the Trilateral Euregio Cluster (TEC), Brussels, Belgium
6Jožef Stefan Institute, Ljubljana, Slovenia

Experimental diagnostics, analysis tools and simulations represent particle distributions in various forms and coordinates. Algorithms to manage these data are needed on platforms like the ITER Integrated Modelling & Analysis Suite (IMAS), performing tasks such as archiving, modeling, conversion and visualization. A method that accomplishes some of the required tasks for distributions of charged particles with arbitrarily large magnetic drifts in axisymmetric tokamak geometry is described here. Given a magnetic configuration, we first construct a database of guiding center orbits, which serves as a basis for representing particle distributions. The orbit database contains the geometric information needed to perform conversions between arbitrary coordinates, modeling tasks, and resonance analyses. Using that database, an imported or newly modeled distribution is mapped to an exact equilibrium, where the dimensionality is reduced to three constants of motion (CoM). The orbit weight is uniquely given when the input is a true distribution: one that measures the number of physical particles per unit of phase space volume. Less ideal inputs, such as distributions estimated without drifts, or models of particle sources, can also be processed. As an application example, we reconstruct the drift-induced features of a distribution of fusion-born alpha particles in a large tokamak, given only a birth profile, which is not a function of the alpha’s CoM. Repeated back-and-forth transformations between CoM space and energy-pitch-cylinder coordinates are performed for verification and as a proof of principle for IMAS.

1. INTRODUCTION AND WORKFLOW

1.1. Motivation, purpose and scope

The ITER Integrated Modelling & Analysis Suite (IMAS) offers various ways to store particle distributions for the study of magnetically confined fusion plasmas. Different representations include different choices of coordinates and different discretization methods (mesh grids or marker particles). The Energetic Particle Topical Group of the International Tokamak Physics Activity (ITPA) is currently driving an action to equip IMAS with tools to model and convert distributions of fast particles between different representations that arise in experimental and computational work.

In experimental work, the observed position space of an energetic particle diagnostic is often given by the line-of-sight of the diagnostic in cylinder coordinates (R, z, ζ), where R is the major radius, z the height and ζ the toroidal angle. In tokamaks, ζ is often an ignorable symmetry coordinate. For many diagnostics, the observed velocity-space is given by weight functions that are derived from energy and momentum conservation. The diagnostic weights are often expressed as functions of kinetic energy E = Me^2/2 and velocity pitch λ = ε_θ/ε, where ε is the magnitude of the velocity vector, ε_θ its component parallel to the magnetic field vector B, and M the particle mass. Diagnosticians hence have a good understanding of the observation regions of diagnostics in phase space (E, λ, R, z). Synthetic diagnostics calculating expected measurements based on numerical simulations are usually based on such representations of local velocity distributions.

Given the uncertainties of experimental measurements, no distinction is usually made between the gyro-averaged distri-
distribution of physical particles and the distribution of their guiding centers (GC), at least in the plasma core. The modeling of distribution functions is therefore often limited to GC statistics (except near the wall where the geometry of gyro-orbits matters for the interpretation of loss diagnostics and for the estimation of heat loads on plasma-facing hardware).

If one assumes that the plasma is not only toroidally symmetric but also quasi-cess, the distribution functions of confined charged particles and their GCs can be taken to depend only on three constants of motion (CoM). Analyses of resonant instabilities are usually performed with respect to such an equilibrated reference state, where the orbit time \( r \), gyrophase \( \xi \) and toroidal angle \( \zeta \) are ignorable symmetry coordinates. The respective conserved actions — namely, the kinetic energy \( E \), magnetic moment \( \mu \), and canonical toroidal angular momentum \( P_\zeta \) — combined with an index \( \sigma \) specifying the sign of \( v_\parallel \) on passing orbits, constitute a natural set of CoM coordinates \((E, \mu, P_\zeta; \sigma)\) that is common in theoretical analyses, although other useful (and equivalent) sets of CoM exist. Recently, an experimental measurement of a fast ion distribution in three-dimensional (3-D) CoM space was demonstrated for fast-ion D-\( \alpha \) spectroscopy [6]. Efforts are underway to apply this orbit tomography method [7] to other fast-ion diagnostics [8].

In order to make predictions of a measurement based on a stability calculation or carry out stability analyses based on measured data, we must be able to transform distribution functions between 3-D CoM space and 4-D representations given in various coordinates as illustrated in Fig. 1. Related problems have been tackled to various degrees by different research groups with different codes and methods. For instance, TRANSP/NUBEAM [9] and LIGKA/HAGIS [10] employ binning and smoothing techniques to map distributions represented by marker particles onto a mesh and compute their gradients in CoM space. While TRANSP/NUBEAM uses Monte-Carlo sampling, LIGKA/HAGIS uses a database of GC orbits for all particle species. Related activities were reported by the ASCOT [11] and MEGA [12] groups. These methods have been used for many years, but detailed documentations are rarely published. The above references [9–11] all point to presentations given at technical meetings. Useful elements can be found in many papers, but it is necessary to assemble that information into a complete goal-oriented workflow.

Thus motivated, this paper documents the workflow used in our computer code VisualStart [12, 13]. Since 2009, we have used this code to initialize hybrid simulations with modeled or numerically computed beam ion distributions [13–17] and to analyze their resonant interactions with Alfven waves [18, 20]. VisualStart represents GC distributions using marker particles, which can be directly used in full-f simulation [1] and yield a quiet start [1, 13]. The representation in CoM space with an underlying GC orbit database will also allow us to directly embed the distribution functions in general phase-space transport theories [21–24], both formally and technically.

Here we propose the orbit-based representation and modeling technique as a solution for ITER IMAS and other platforms that need to store and process distributions of charged particles in magnetically confined plasmas. The subject of smoothing is discussed briefly in the following section, but the documentation and verification of smoothing algorithms is left for future work, as is the associated problem of initializing delta-\( f \) simulations that require the evaluation of gradients. We note that a similar and in many ways complementary effort has been undertaken by S. Benjamin et al. [25].

### 1.2. Basic rules for practical situations

The theoretical foundations for transforming distribution functions are relatively straightforward, but it is also clear that compromises and creativity are needed when tackling real-world data with numerical techniques, where accuracy has to be traded for computational performance and where data may have varying degrees of quality and completeness. For instance, the singularities and topological boundaries of CoM space [26] require attention when cutting a mesh to define phase space volume elements. Experimentally measured distributions and numerically computed distributions are often incomplete and noisy, so some amount of modeling is needed, including smoothing, interpolation and extrapolation.
In view of these challenges, some basic rules should be followed in order to maintain the physical and numerical integrity of the data, independently of the particular workflow used to process distribution functions:

(i) Smoothing, interpolation and extrapolation are best applied at the preparation stage, where a new distribution function is first constructed or imported. Such potentially irreversible manipulations should be minimized in all subsequent operations.

(ii) Pay attention to the fact that, in toroidal geometry, resolution and noise levels inevitably vary in space and in time, because the mirror force and magnetic drifts cause GCs to be distributed nonuniformly along their orbits. In particular:

- Inherent topological discontinuities and singularities, as found in CoM space, should be preserved; e.g., through mesh accumulation and by smoothing only around, not across trapped-passing boundaries and loss cones.
- The optimal representation method (maximizing accuracy) may differ between archiving a distribution function and using it in a simulation. While archiving merely requires adequate resolution, simulations should also minimize signal-noise correlations [13].

These issues are most pronounced in but not limited to cases with large magnetic drifts.

(iii) Ensure that the distribution to be transformed is an exact equilibrium in the given magnetic configuration; i.e., its shape must not vary in time when evolved using unperturbed equations of motion.

The equilibrium constraint (iii) is crucial, so it deserves further explanation: Any distribution function $f$ can be mapped to CoM space, where the result $f_{\text{orb}}$ measures the time-averaged particle densities on unperturbed GC orbits. This operation is indicated by a blue arrow in Fig. 1. However, when the original $f$ is not an equilibrium, this mapping operation is time-dependent and it is impractical to store the information needed to invert it. The equilibrium constraint reduces the effective dimensionality to three CoM by determining the longitudinal distribution of GCs along their orbit, which is now time-independent and with it all coordinate transformations as well. The coordinate transformations $f \leftrightarrow f_{\text{orb}}$ can then be readily performed in any direction, as indicated by a red double arrow in Fig. 1. Nonetheless, distribution functions are very likely to be encountered in IMAS, where one can expect to receive particle data from a measurement for which the magnetohydrodynamic (MHD) equilibrium is not exactly known or from a simulation where waves or magnetic islands were present. Enforcing the equilibrium constraint means that the input data are transformed into a distribution that is consistent with a modeled magnetic configuration, which generally differs somewhat from the conditions where the input data originated from.

1.3. Orbit-based representation and modeling

The above considerations suggest that it is useful and advantageous to represent distribution functions in CoM space with an underlying database of unperturbed GC orbits. With that geometric information, it is straightforward to reconstruct a 4-D distribution in arbitrary coordinates by loading a suitable number of markers uniformly in time along each GC orbit [13]. Projections to various coordinates and views of user-defined sub-spaces with user-defined resolution can then be readily obtained by binning the markers on demand. This minimizes the amount of transformations and associated data corruption, as demanded in items (i) and (ii) above. Marker particles are a natural choice for representing fast particle distributions since the latter are often computed using particle codes. Moreover, by organizing the markers on orbit contours, one can identify and deal with inherent singularities and discontinuities as required in item (ii). Last but not least, by representing a distribution function in CoM orbit space, the equilibrium constraint (iii) is automatically enforced.

We note that the implementation and enforcement of the equilibrium constraint only requires a conversion to CoM coordinates: $f \rightarrow f_{\text{orb}}$. The Jacobians required for such transformations can be determined using the Monte-Carlo method as implemented in TRANSPIRUBMAR by Breslau & Liu [9]. In this way, it is possible to avoid integration along GC orbits.

However, the computational cost associated with building an orbit database is more than repaid by the information one gains about the shape of the orbits, the longitudinal particle distribution, and the transit times. The transit times are needed for resonance analyses and their variation near trapped-passing boundaries can be used for mesh accumulation. The information about the orbit geometry is also useful for reduced models, for instance to capture finite-orbit-width effects [27], and it allows to define and make use of a variety of specialized GC coordinates, like the radii $R_m$ where an orbit crosses the midplane. Last but not least, the information contained in the orbit database is also valuable for carrying out modeling tasks, such as the following example:

Given a model function $G_{\text{mdl}}$ that may not be an exact solution of a kinetic equation for the GCs of physical particles, we wish to construct an equilibrium GC orbit distribution $f_{\text{orb}}$. A common example are models of the form $G_{\text{mdl}}(E, \lambda, \Psi_p)$ that contain measured radial profiles as functions of flux $\Psi_p$ and a velocity distribution computed by a bounce-averaged Fokker-Planck code. In such a case, the orbit database allows to incorporate the effect of magnetic drifts a posteriori [12]. There are, of course, various possibilities to perform this operation; in other words, there is some freedom to interpret the values of $G_{\text{mdl}}$ as weights in an orbit distribution $f_{\text{orb}}$. An intuitive and physically meaningful choice is to perform an orbit-time average $\langle G_{\text{mdl}} \rangle_{\text{orb}}$ followed by a suitable normalization: $f_{\text{orb}} = \text{const.} \times \langle G_{\text{mdl}} \rangle_{\text{orb}}$.

2 Functions such as $G_{\text{mdl}}(Z)$ described in this example will here be called “quasi-distributions“, because they do not measure the true number of objects of interest (here GCs) in a volume element $dZ$ but require more elaborate transformations than a mere conversion of coordinates. See Section 3.
1.4. Workflow

The workflow described in this paper is shown schematically in Fig. 2 and consists of five steps:

• **Step 1**: Define an axisymmetric magnetic field $B$, e.g., using a dedicated MHD equilibrium code.

• **Step 2**: Define samples in 3-D CoM orbit space, with cell indices $(i, j, k)$. The blue and red workflow branches in Fig. 2 begin with different inputs:
  - (blue) Import marker particles from another code as in earlier works [15–17]. The weights of the imported samples are reinterpreted as orbit weights: $w_{\text{import}} \rightarrow W_{\text{orb}}(i, j, k)$.
  - (red) Use a meshing algorithm to divide the CoM space into cells of size $\Delta V_{\text{orb}}(i, j, k)$. This path is described in detail in this paper.

• **Step 3**: Integrate along GC orbits starting from the samples defined in Step 2. Store in a database.

• **Step 4**: Represent an equilibrium distribution function $f_{\text{orb}}$ using weighted markers loaded on the orbits in the database. When loading $N_{r}$ markers uniformly in time, their weight is an $N_{r}$’s fraction of the orbit weight: $w = W_{\text{orb}}/N_{r}$. In the case of the red workflow in Fig. 2 $W_{\text{orb}} = \Delta V_{\text{orb}} \times f_{\text{orb}}$ is the product of the volume element $\Delta V_{\text{orb}}$ and the desired phase space density function $f_{\text{orb}}$. The latter may be imported or constructed from a model.

• **Step 5**: Visualize and verify. Iterate if necessary; e.g., changing the resolution in CoM space (Step 2), the number of markers on the orbits, or the weight assignment (Step 4). Export when ready.

Note that MHD wave spectra can be obtained from the data available in Step 1, and resonance maps can be computed from the orbit database built in Step 3. Procedures for smoothing the distribution, computing phase space gradients and analyzing the stability of resonant modes remain to be added in or around Step 5.

1.5. Outline

Clear definitions of symbols and terminology are a prerequisite for successful conversion operations. Thus, we begin our treatise in Section 2 with a discussion of different classes of distributions functions and how they are transformed. **Step 1** of our workflow is part of Section 3 where we describe the tokamak plasma that serves as a working example, define coordinates and normalizations. **Step 2**, the construction of a phase space mesh, is covered in Section 4 where we also discuss orbit classes and the phase space topology in our scenario. The GC equations of motion we solve to construct the orbit database in **Step 3** are given in Section 5 and the formula for the orbit volume elements is derived in Section 6. **Steps 4 & 5** are covered in Sections 7 and 8 where we construct a distribution function $f$ from a model, visualize and analyze the result, and verify the integrity and accuracy of our scheme by transforming $f$ back and forth between two different representations. Modeling is a rich subject on its own, and our example is only meant to highlight the utility of an orbit database and to serve as a proof-of-principle for IMAS applications. The Appendices contain supplementary information on binning procedures and numerical accuracy.

2. CLASSES OF DISTRIBUTIONS AND TRANSFORMATIONS

There are not only different coordinates and different numerical representations of distributions; the latter also come in different types whose identity must be clearly defined since it determines how they are transformed. Here, we distinguish...
between three types of distributions: (i) density functions, (ii) histograms, and (iii) quasi-distributions.

The first two can be characterized as follows:

(i) The symbols \( f(Z) \) represent phase space densities, which have units of \([m^{-8}s^2}]\) and whose values are independent of the coordinates \(Z\) appearing in the argument, so \( f(Z(u,x)) = f(u,x) \). The argument merely specifies the representation space. The same is true for velocity and volume integrals:

\[
\begin{align*}
  n(x) & = \int d^3v f :\text{density field,} \\
  v(\tau) & = \int d^3x f : \text{velocity distribution in a volume } V.
\end{align*}
\]

(ii) The symbols \( h(Z) \) represent histograms, which have the units of the inverse volume element \(|dZ|\) of the coordinates used for “binning” (counting the objects of interest). Histograms are convenient to construct and integrate since no Jacobian is required (except for coordinate conversions).

The phase space density function \( f \) is related to histogram functions \( h, h' \), etc., as

\[
f(Z) = \mathcal{J}_Z^{-1} h(Z) = \mathcal{J}_Z^{-1} h'(Z') = ...
\]

where \( \mathcal{J}_Z \) is the Jacobian for the transformation from Cartesian coordinates \( Z_c = (v, x) = (v_1, v_2, v_3, x, y, z) \) to another arbitrary set of phase space coordinates \( Z \).

A phase space density function \( f \) is an exact solution of a kinetic equation for the objects of interest; in our case, guiding centers (GC). In this work, we assume that all GC coordinates are defined such that they preserve the form of the kinetic equation

\[
0 = \left( \frac{\partial}{\partial \tau} + Z_c \cdot \frac{\partial}{\partial Z_c} \right) f = \left( \frac{\partial}{\partial \tau} + \mathcal{J}_{Zc} \cdot \frac{\partial}{\partial Z_{gc}} \right) f = \left( \frac{\partial}{\partial \tau} + \hat{Z}_{gc} \cdot \frac{\partial}{\partial Z_{gc}} \right) f.
\]

(4)

This is the case when the Jacobian \( \mathcal{J}_{gc} \) for the transformation \( Z_c \rightarrow Z_{gc} \) from Cartesian to GC coordinates satisfies the phase space conservation law

\[
\mathcal{J}_{gc} \equiv \partial_\tau \mathcal{J}_{gc} + \partial_{Z_{gc}} \cdot \left( \mathcal{J}_{gc} \cdot Z_{gc} \right) = 0.
\]

Equation (5) is trivially satisfied for canonical coordinates, since their volume elements differ only by a constant factor, so \( \mathcal{J}_{gc} = \text{const} \). All our CoM coordinates — namely, \((E, \mu, P, \sigma)\) and any equivalent set of GC orbit coordinates, such as \((v, \theta, \phi, \tau, \rho)\) — are, by definition, conserved by the GC equations of motion that we solve. This means that, even when the equations of motion are not written in Hamiltonian form, our orbit coordinates combined with the triplet of ignorable angle variables \((r, \xi, \zeta)\) — orbit time, gyrophase, toroidal angle — can be considered to be canonical coordinates, whose Jacobians are constants satisfying Eq. (5) down to the numerical accuracy of our particle-pushing algorithm.

A well-known set of noncanonical GC coordinates is \((\mu, v_1, x_{gc})\), whose volume elements transform as

\[
d^3v = \frac{2\pi B_\parallel}{M} d\phi d\sigma d^2x_{gc}.
\]

The Jacobian factor \( B_\parallel \) satisfies Eq. (5) as shown by Littlejohn [29]. Equation (6) is very useful, because it can be easily converted to other velocity coordinates that are defined locally in position space — such as the pitches \( \lambda = v_1/v \) and \( \Lambda = \mu B_\parallel/E \) since their Jacobian factors can be derived analytically [X]

Both (i) densities and (ii) histograms are true distributions in the sense that, when integrated over a phase space volume element \( dV \),

\[
\int dV f(Z_{gc}) = \int dV f = \mathcal{N}(dV),
\]

they yield the true number \( \mathcal{N} \) of objects contained inside that volume element. Functions or data that do not satisfy this condition for the specified kind of objects constitute the third type in our classification scheme:

(iii) The symbol \( G(Z) \) represents quasi-distributions, which do not yield the true number of objects by mere volume integration in the configuration at hand, but require more complex transformations.

Quasi-distributions often arise in experimental or modeling work. A typical example in experimental measurements is the line-of-sight density, where the transformation \( G \rightarrow f \) takes the form of an inverse problem.

In fact, since the magnetic configuration in an experiment is rarely known accurately, any experimentally measured particle statistics should be treated as quasi-distributions when used as input for modeling distributions in a numerically constructed MHD equilibrium. Conversely, it should be kept in mind that the spatial distribution of GCs differs from that of physical particles since the latter is broadened by gyration around the GCs. This means that, strictly speaking, the GC distributions that we deal with in this work constitute quasi-distributions from the viewpoint of physical particles. Even if our GC phase space density function \( f \) is normalized to give the correct number of physical particles in the entire plasma volume, \( \mathcal{N}(V) = \mathcal{N}_{phys}(V) \), the values of \( \mathcal{N}(dV) \) and \( \mathcal{N}_{phys}(dV) \) given by Eq. (7) for an arbitrary subvolume \( dV \) may differ in the case of fast ions with a steep density gradient.

In general, the process of “modeling” essentially consists of choosing a physically meaningful transformation \( G \rightarrow f \). We will return to this topic in Sections 6 and 8 where we apply our methods to model a distribution of fusion-born alpha particles, based on a quasi-distribution \( G(\Psi_p) \) that represents their birth profile.

Finally, we emphasize that when a GC orbit distribution function is given in the form \( f_{orb}(E, \mu, P, \sigma) \), or an equivalent
set of CoM coordinates, the Jacobian $J_{\text{CoM}}$ must be accompanied by instructions concerning how to deal with the index $\sigma$ that specifies the sign of the parallel velocity. First, it must be clarified whether $\sigma$ is the sign relative to the magnetic field or the plasma current if their directions differ. Second, it must be clarified how particles trapped in a magnetic mirror are to be counted, because $\sigma$ is redundant for such trapped particle orbits. Depending on the choice made, the integral in Eq. (7) can be written in at least three ways:

$$N = \sum_{\sigma=\pm 1} \int dE d\mu dP_z J^{\text{up}}_{\text{CoM}} (\text{trap/pass.}) f_{\text{orb}},$$  \hspace{1cm} (8a)

$$= \sum_{\sigma_{\text{pass}}=0, \pm 1} \int dE d\mu dP_z J_{\text{CoM}} f_{\text{orb}},$$  \hspace{1cm} (8b)

$$= \sum_{\sigma_{\text{pass}}=\pm 1} \sum_{\sigma_{\text{pass}}=\pm 1} \int dE d\mu dP_z \frac{J_{\text{CoM}}}{2} f_{\text{orb}}.$$  \hspace{1cm} (8c)

In the first case (8a), the summation over $\sigma$ counts each trapped orbit twice, which means that the histogram $h^{\text{up}}_{\text{orb}}$ has two sets of identical entries each representing half of the particles in the domain of trapped orbits. In the second case (8b), the summation index $\sigma_{\text{pass}} = \{0, \pm 1\}$ is three-valued, identifying trapped (0), co- and counter-passing orbits ($\pm 1$), so that $h_{\text{orb}}$ represents the full number of particles in all regions of GC phase space without duplicate entries. In mathematical form:

$$h_{\text{orb}}(\sigma_{\text{pass}} = 0) = 2 \times h^{\text{up}}_{\text{orb}} (\text{trap.}),$$  \hspace{1cm} (9)

$$J_{\text{CoM}} = J^{\text{up}}_{\text{CoM}} (\text{pass.}) = 2 \times J^{\text{up}}_{\text{CoM}} (\text{trap.}).$$  \hspace{1cm} (10)

The values of the density function $f_{\text{orb}}$ itself are, of course, independent of the convention used. The point we want to make is that it is crucial to ensure consistency between the summation over $\sigma$ and the Jacobian when counting marker particles by integrating $f_{\text{orb}}$ in $(E, \mu, P_z; \sigma)$ space as in Eq. (8).

In this paper, we adopt the third option (8c), with the single-valued constant Jacobian $J_{\text{CoM}}/2$. Our CoM phase space mesh defined in Section 4 will double-count all orbits — passing and trapped — when they cross the plasma midplane: once on the high-field side (HFS) and once on the low-field side (LFS). The factor 2 dividing the Jacobian in Eq. (8c) compensates this.

3. GEOMETRY, COORDINATES AND NORMALIZATION

a. System geometry The magnetic field vector is written

$$B = \nabla \times A = \nabla \Psi \times \nabla P + B_T \nabla \theta,$$  \hspace{1cm} (11)

$$= \nabla \zeta \times \nabla \Psi + \nabla \Psi \times \nabla \theta,$$  \hspace{1cm} (12)

where $2\pi \Psi = \frac{1}{2\pi} \int d\Psi d\varphi d\zeta J_\Psi^* B \cdot \nabla \zeta$ is the toroidal flux and $2\pi \Psi = -2\pi A_T = \frac{1}{2\pi} \int d\Psi d\varphi d\zeta J_\Psi^* B \cdot \nabla \varphi$ the poloidal flux. These fluxes are related via the tokamak safety factor $q = d\Psi/d\Psi_p$ measuring the mean magnetic field line pitch.

b. Magnetic flux labels The normalized toroidal and poloidal fluxes $(\psi, \varphi)$ or their square roots $(\rho_T, \rho_P)$ can serve as convenient minor radial coordinates:

$$\rho_T^2 = \psi = \frac{1}{2\pi} \int_{\partial \Omega} V(\Psi) \frac{dV}{d\Psi} J^*_{\Psi} d\Psi, \quad \rho_P^2 = \varphi = \frac{\Psi_P - \Psi_{P,0}}{\Psi_{P,a} - \Psi_{P,0}},$$  \hspace{1cm} (13)

where $V(\Psi)$ is the volume within the flux surface labeled $\Psi$. $\Psi_{P,0}$ and $\Psi_{P,a}$ are the values of the poloidal flux at the mag-
netic axis \((r = 0)\) and boundary \((r = a)\). For the toroidal flux
we have \(\Psi_a - \Psi_b = \int_{\Omega_b}^{\Omega_a} \frac{d\Psi}{B} q\). In some occasions, we use a
volume-averaged minor radius \(0 \leq r(\Psi) \leq a\), which is defined here as
\[
r^2 = \frac{R_0}{\pi} \int_0^{\psi_{\text{max}}} \frac{dV}{J^2} B_x = \frac{R_0}{\pi} \int_0^{\psi_{\text{max}}} \frac{d\Psi}{B_x} \approx a^2 \psi.
\]
Our \(r\) reduces the geometric minor radius in the limit of a
cylindrical plasma with circular cross-section. For \(B_c \approx R_0 B_0\), we have
\(r/a \approx \rho_r\), and \(a \approx [2(\Psi_a - \Psi_0)/B_0]^{3/2}\) is the minor radius
of the plasma boundary.

c. Particle and orbit coordinates In the guiding center
(GC) description, the velocity space consists of the igno-
rable gyrophase \(\xi\), the perpendicular particle velocity \(v_\perp\), and the parallel GC velocity \(v_\parallel = v_x \hat{b} \parallel \) relative to the mag-
netic field vector \(\mathbf{B}\) with unit vector \(\hat{b} \parallel \equiv \mathbf{B}/|\mathbf{B}|\) and magnitude
\(B = |\mathbf{B}|\). The canonical momentum can be written in the
form \(\mathbf{P} = \mathbf{P}_{gc} + \Omega_0^{-1}v\) for a particle with gyrofrequency
\(\Omega_0 = Q_e B_0/M\), charge \(Q_e\) and mass \(M\). In GC coordinates,
itso covariant toroidal component,
\[
P_{\zeta} = \mathbf{P}_{gc} \cdot \partial_{\xi} \mathbf{x} = \frac{\Psi}{B_0} \hat{B}_\zeta + \frac{\psi}{\Omega_0} \hat{B}_\zeta,
\]
is called the canonical toroidal angular momentum and is chosen here to have units of area per radian, \((2\pi)^{-1}[\text{m}^2]\). Together with
the lowest-order magnetic moment \(\mu\) and the signed ki-
netic energy \(E = \sigma_e E\),
\[
\mu = \frac{Mv_r^2}{2B}, \quad E = \sigma_e \frac{M}{2} v^2, \quad \sigma = \frac{v_J}{|v_J|}.
\]
this yields a set of coordinates \((E, \mu, P_\zeta)\) that belongs to the
class of constants of motion \((\text{CoM})\), which are conserved by
our equations of motion (presented in Section 5) when they
are solved in a time-independent field \(\mathbf{B}\) that is symmetric
along \(\zeta\) (or “axisymmetric” around \(\hat{r}\)), satisfying \(\partial_{\xi} \mathbf{B}_\zeta =
\partial_{\zeta} \mathbb{P}_\zeta = 0\).

The sign \(\sigma\) of the parallel velocity is measured relative to
the parallel current density \(J_p = J \cdot \mathbf{B}/|\mathbf{B}|\). While redundant
or unnecessary for orbits trapped in a magnetic mirror
(both signs are present), \(\sigma\) identifies whether a passing orbit
with coordinates \((E, \mu, P_\zeta)\) is co- or counter-going. In ma-
thematical treatments, \(\sigma\) usually appears as a separate index,
\((E, \mu, P_\zeta; \sigma)\), but this is inconvenient for numerical represen-
tations. In our orbit database, we define \(\sigma_{\text{ref}}\) at a certain refer-
cence point; namely, the starting point used for the orbit calcu-
lation. This or any equivalent convention allows us to attach
\(\sigma_{\text{ref}}\) to \(E\) as in Eq. (16) and obtain a compact representation
where all CoM arrays in the computer code have the same
dimensionality: three.

It is also useful to define several different coordinates mea-
suring the velocity pitch:
\[
\Lambda \equiv \frac{\mu B_0}{E} = \frac{\cos^2 \alpha}{B}, \quad \sin \alpha = \lambda \equiv \frac{v_\parallel}{v} = \sigma B \sqrt{1 - \mathbf{B}},
\]
with \(\hat{B} = \mathbf{B}/B_0 \approx R_0/R\) and \(\sigma_B \equiv v_\parallel/v\). Each pitch coordi-
nate has its advantages in practical applications:

- The coordinate \(\lambda = v_\parallel/v\) is widely used together with \(E\) or velocity \(v = \sqrt{E/M}\). The local velocity space
Jacobi of this set is a CoM \((J_{E,1} \propto E)\), Eq. (A12),
and the Fokker-Planck equation can be integrated analyti-
cally in these coordinates [34].

- The coordinate \(\alpha\) is a true angle, so its value mea-
sures the velocity pitch in a very intuitive form. More-
over, the domains of co- and counter-passing particles
are enlarged along the \(\alpha\)-axis, making this coordinate
a good choice for sampling the velocity distribution of
predominantly injected beam ions [17] and distributions
produced by central RF heating using a 3-ion scheme
[31,32,55].

- The coordinate \(\Lambda = \hat{B}^{-1} \cos^2 \alpha\) has the advantage of
being a constant of motion with \(J_{E,1} \propto \sqrt{E}\) and distributions
produced by central RF heating using a 3-ion scheme
[31,32,55].

Unlike the conserved quantities \(E, \mu\) and \(\Lambda\), the pitch coordi-
nates \(\alpha\) and \(\lambda\) vary along a GC orbit. The latter can be turned
into orbit coordinates (i.e., CoM), \(\sigma_{\text{ref}}\) and \(\lambda_{\text{ref}}\), by evaluating
them at a well-defined reference point, as we did earlier for the
sign \(\sigma_{\text{ref}}\).

d. Reference points for orbit coordinates In cases where
we import computed particle distributions (blue workflow in
Fig. 2), the initial reference point is taken to be the position of
an imported marker particle, which may be located anywhere inside the plasma or in the surrounding vacuum.

In cases where we need to construct a new CoM mesh (red workflow in Fig. 2) the reference points lie in the midplane, which (at least in a usual tokamak plasma) contains all O-type stagnation points and is defined as the curve \( z_m(X) \) along which the magnetic field is perpendicular to its gradient,

\[
z_m(R) : \mathbf{B} \cdot \nabla \mathbf{B} = 0, \tag{18}\]

The midplane is shown as a dash-dotted green line in Fig. 4. It contains the magnetic axis, \((R, z) = (R_0, z_0)\), but deviates from the horizontal plane \(z = z_0\) in up-down asymmetric plasmas like ours. Every GC orbit crosses the midplane twice, once in a region with higher field strength and once in a region with lower field strength. Figure 5 shows an example, where the orbit’s high- and low-field side crossings of the midplane are labeled “HFS” and “LFS”, respectively.

e. Normalization Spatial positions and lengths are given in meters unless stated otherwise. Velocities are normalized by a reference value \(v_0\). Energy is normalized by \(M_0^2\) (for each particle species) and the magnetic field by its on-axis value \(B_0\) (here 3.7 Tesla):

\[
\hat{b} = \frac{v}{v_0}, \quad \hat{E} = \frac{E}{M_0^2}, \quad \hat{\mu} = \frac{\mu B_0}{M_0^2}, \quad \hat{\nu} = \frac{v_0^2}{2B}. \tag{19}\]

4. COM MESH AND DRIFT ORBIT TYPES

In this section, we describe how we sample the GC orbit space. The example in Fig. 5 is used for illustration. A low resolution is chosen in order to make all grid points clearly visible. All coordinates shown in Fig. 5 are evaluated at the height \(z_m(X)\) of the midplane as defined in Eq. (18). This means that all coordinates appearing in this section are constants of motion (CoM). The set \((E, \alpha, X)\) used in panels (a) and (b) yields particularly compact and, in our opinion, intuitive plots. Another view of pitch-position space in \((\Lambda, X)\) coordinates is shown in panel (c). The plasma boundary is also taken to be the loss boundary for the GC orbits, so we do not include the vacuum here and our midplane mesh covers only the width of the plasma itself: \(-1 \text{ m} \leq X \leq 0.84 \text{ m}\).

The reference velocity used for normalization is chosen to be \(v_0 = 1.3 \times 10^7 \text{ m/s}\), which corresponds to \(E_0 = M_0^2/2 \approx 3.5 \text{ MeV}\) for fusion-born alpha particles (\(^{3}\text{He}^{++}\)) or 1.75 MeV for deuterons (\(^{2}\text{He}^{+}\)). Both species have the same charge-to-mass ratio \(qE/M\) and their characteristic gyroradius in this example is \(\rho_0 = v_0/\Omega \approx 0.07 \text{ m} = 0.02 \times R_0\). We consider the full energy range \(0 \leq E \leq \hat{E}_0\) with \(\hat{E}_0 = 0.5\) and define a mesh consisting of \(N_E^{(\Lambda)}\) grid points that are the boundaries of \(N_E = N_E^{(\Lambda)} - 1\) cells. The value of energy at the center of cell \(i\) is denoted \(\hat{E}_i\). The example in Fig. 6(a) has \(N_E = 15\) cells with equal sizes \(\Delta \hat{E}_i = \hat{E}_0/N_E = \text{const.}\)

The remaining phase space is divided along the coordinate...
The classes of potato and stagnation orbits arise from magnetic drifts. The magnitude of the magnetic drift velocity $v_{\text{drift}}$ (see Eq. (23) below) is proportional to the inverse aspect ratio $a/R_0$ and the kinetic energy. Furthermore, its $(R,z)$-component is inversely proportional to the poloidal field $B_p \propto I_p$, which tends to divert it into the toroidal direction. This means that magnetic drifts tend to be large for energetic particles in tokamak plasmas with relatively low plasma current and in compact tori. In such cases, the overall fraction of potato and stagnation orbits can be significant.

The mesh in Fig. 6(b) is rather sparse, and our automatic classification algorithm identified only 3 samples as potato orbits (see Fig. 8(a) for a denser mesh and higher energy). The $a > 0$ legs of potato orbits are found around the upper rim of the trapped-passing boundary on the $X > 0$ side of Fig. 6(b). On the $X < 0$ side, they are located slightly below the $a = 0$ line.

It is important to note that the trapped-passing (t-p) boundary on the $X < 0$ side also lies below the $a = 0$ line. This is due to the magnetic drifts and has the implication that the turning points — where the magnetic mirror force causes a sign reversal in $v_{\text{drift}}$ — are near but not identical to the V-type stagnation points of trapped particle orbits. The example in Fig. 7(a) illustrates this for the case of an orbit very close to a t-p boundary.

The t-p boundary refers to a separatrix in the orbit topology; i.e., it marks the topological transition of two V-type stagnation points via an X-type stagnation point as illustrated schematically in the box at the top left of Fig. 4. The marginally trapped orbit in panel (a) has two V-type stagnation points that nearly form an X. Across the t-p boundary, this orbit decomposes into a counter-passing orbit and a trapped orbit, each with a single V-type stagnation point that turns into a smooth part of the orbit contour as one departs from the t-p boundary.

We use the word “stagnation” to indicate that GCs spend a relatively large amount of time in a relatively small region of the poloidal plane. Around O-type stagnation points, this is the case for the entire orbit. In the case of X- and V-type stagnation points as in Fig. 7 GCs spend a large amount of time in a small portion of the orbit contour. Spatial stagnation also means that $v_{\text{drift}} = \sigma_B \sqrt{1 - \Lambda B}$ stagnates as shown in

\[\begin{array}{|c|c|c|}
\hline
\text{Encircle magnetic axis at } (X, Y) = (0, 0) & \text{Yes} & \text{No} \\
\hline
\text{Trapped:} & \text{Potato orbits} & \text{Banana orbits} \\
\hline
\text{Passing:} & \text{Circulating orbits} & \text{Stagnation orbits} \\
\hline
\end{array}\]

\footnote{The discussion below Eq. (19) in Ref. [36] is related to this.}
we are able to avoid these degenerate $\eta_l = 0$ points entirely by locating them along the $X_0$ cell boundary on $(E, \Lambda) = \text{const.}$ lines. All cell centers are, thus, ensured to have $\eta_l \neq 0$.

5. EQUATIONS OF MOTION

As indicated by the red and blue arrows between Steps 2 & 3 in our workflow Fig. 2, the GC phase space samples that are provided by

(a) the mesh constructed in Section 4 or

(b) particle codes like OFMC [38, 39] and MEGA [40, 41] (when operated in full-f mode [14, 42])

serve as initial conditions for the GC equations of motion, which we solve once around each orbit.

VisualStart solves the same equations with the same numerical scheme as its current target code MEGA. Following the formalism introduced by Littlejohn [29] and reviewed by Cary & Brizard [43], we use the effective electromagnetic fields $B^* \equiv \nabla \times A^*$ and $E^* \equiv -\nabla \Phi - \partial_t A^*$, with the potentials $A^* \equiv A + \frac{\rho_0}{c} B$ and $Q e \Phi \equiv Q e \Phi + \mu B$. Here, we consider the motion of GCs in an unperturbed magnetic field and in the absence of any electric field ($\partial_t B = \frac{\rho_0}{c} = 0$), so the effective fields reduce to

$$B^* = B + \frac{\rho_0}{\Omega} B^* \times \hat{b}, \quad E^* = -\frac{\mu}{Q e} \nabla B, \quad \text{(20)}$$

and the equations for the GC velocity $v_{GC} = \dot{x}_{GC}$ and its parallel acceleration become

$$\dot{x}_{GC} = v^*_{GC} + B^* \cdot \hat{b}_{GC} = v^*_{DB} + B^* \cdot \hat{b}_{GC}, \quad \text{(21)}$$

$$\dot{v}_{||} = \left( \frac{\Omega}{B} \right) B^* \cdot E^*, \quad \text{(22)}$$

$\hat{b} = B/B$ is the unit vector along $B$, and $B^* \equiv B \cdot \hat{b}$ is Littlejohn’s GC Jacobian for the noncanonical set of GC coordinates $(\mu, u, \xi, x_{GC})$ with $u \equiv \hat{b} \cdot (x_{GC})$ ($x_{GC} \equiv \eta_l$ (see also the discussion on p. 704 of Ref. [43]). The GC velocity is composed of a parallel component (modified by curvature effects) and magnetic drifts due to $\nabla B$ and curvature $\nabla \times \hat{b}$:

$$v^*_{||} = \frac{\mu}{Q e} \hat{b} \times \nabla B + \frac{\rho_0}{\Omega} B^* \cdot \nabla \hat{b}, \quad \text{(23)}$$

In the normalization of VisualStart (Section 4):

$$\hat{b}^0 \equiv \left( \frac{\Omega}{B} \right) \hat{b}_{||}, \quad \hat{v}^*_{DB} = \rho_0 \hat{B}_{||} \times \hat{b} \times \nabla \hat{b} + \frac{\rho_0}{\Omega} B^* \cdot \nabla \hat{b}, \quad \text{(24)}$$

with $\rho_0 = \frac{\eta_l}{\Omega}$. The computed GC orbits are recorded in a database, following accuracy checks [15].

The example in Fig. 6 gives a total of about 8000 orbit samples, 52% of which lie inside the region bounded by the red curve ($\Lambda B^0 \leq 1$) for the present setup. About 1100
orbits (14%) are lost as they hit the (artificial) boundary of the simulation domain due to magnetic drifts. Using 3 processes on an 8th Generation Intel CORE i7-8565U processor (1.80 GHz × 8), it took about 35 minutes to compute these 8000 orbits with our simple Matlab code in which there is much room left for optimization. A proper orbit database for the entire plasma volume may require at least 100 times more samples, which would take 150 core-hours with the present implementation. An optimized code in a language such as Fortran, C or Julia is expected to need only a small fraction of this time. Making use of GPUs, the time required to compute such an orbit database should be reasonably short for routine work.

6. VOLUME ELEMENTS AND WEIGHTING

6.1. Derivation

Having computed the GC orbits using the unperturbed equations of motion in Section 4, the next step is to represent the distribution function \( f \) in discretized form using an ensemble of marker particles labeled \( n = 1...N_{\text{mk}} \), which sample the orbits in our database. In other words, we seek the Klimontovich representation

\[
f(Z) \approx \sum_{n=1}^{N_{\text{eq}}} w_n \delta(Z_{\text{gc},n} - Z).
\]

At this point, \( Z_{\text{gc},n} \) is the marker position in an arbitrary set of GC coordinates including ignorable angles, for instance \((v^x_1,v^y_1,\xi,\eta)\). The Dirac \( \delta \) distribution has units of inverse phase space volume \([\text{m}^{-6}\text{s}^3]\) and may be viewed as a proxy for any sort of particle shape factor needed to map the weights \( w_n \) onto a discrete mesh. Each marker follows the trajectory of a GC as indicated by the subscript in \( Z_{\text{gc},n} \). Its weight factor \( w_n \) represents a certain number \( N_n \) of GCs of physical particles:

\[
w_n = [\Delta N]_n = \Delta V_n \times N(\Delta V_n)/\Delta V_n = \Delta V_n \times f.
\]

where \( \Delta V_n \) is a 6-D volume element of GC phase space in Cartesian coordinates that is attached to marker \( n \).

Since we are considering only exact equilibrium distributions, the orbit time \( \tau \) is an ignorable angle coordinate with period \( \tau_{\text{pol}} \) and all factors in Eq. (26) are independent of the three angle coordinates \((\tau,\xi,\zeta)\). These distributions depend only on three CoM coordinates, whose values we will represent by cell indices \((i,j,k)\). When we load \( N \) markers uniformly in time along an unperturbed GC orbit, neighboring markers are indistinguishable. Each marker carries precisely \( N_n \)’s fraction of their orbit’s volume \( \Delta V_{\text{orb}} \) and weight \( w \):

\[
\Delta V_n \rightarrow \Delta V_{i,j,k} = \left[ \Delta V_{\text{orb}} / N_n \right]_{i,j,k} = \left[ \Delta V_{\text{orb}} \tau_{\text{pol}} \right] / \tau_{\text{pol}}_{i,j,k}. \tag{27}
\]

\[
w_n \rightarrow w_{i,j,k} = \frac{W}{N_n} \quad \text{for } l = 1,...,N_{\tau,j,k}. \tag{28}
\]

As indicated in Eq. (27), the markers also represent equal fractions \( \Delta \tau \) of an orbit’s poloidal period \( \tau_{\text{pol}} \):

\[
\Delta \tau_{i,j,k} = \frac{\tau_{\text{pol}}}{N_n}_{i,j,k}. \tag{29}
\]

Recalling from Eqs. (20) that the volume element is eventually applied to a distribution function \( f \), it is clear that Eq. (29) actually implies an orbit time average:

\[
\sum_{i=1}^{N_{\tau,j,k}} \Delta \tau_{i,j,k} f(Z_{\text{gc},i,j,k}) \approx \int_{\tau_{\text{pol}}} d\tau f \equiv \tau_{\text{pol}} \int f_{\text{orb}}. \tag{30}
\]

The average is trivial for true equilibrium distributions, which are independent of \( \tau \), so that \( f = f_{\text{orb}} \). However, \( f \) in Eq. (30) may also be replaced by an arbitrary quasi-distribution \( G \), so this equation also constitutes a recipe for transforming arbitrary models into true equilibria. In summary, Eq. (26) becomes

\[
w_{i,j,k} = \frac{W}{N_n} \times \left[ \Delta V_{\text{orb}} / N_n \right]_{i,j,k}. \tag{31}
\]

In applications where we merely import particle data computed by another code (blue workflow in Fig. 2), the weights \( w_i \) are provided with the data. Each sample is interpreted as the initial position of a new GC orbit, so the input weight is interpreted as the weight of an orbit: \( w_{\text{import}} \rightarrow w_{i,j,k} \). This step enforces the equilibrium constraint, and its first equality of Eq. (31) yields the marker weights. The volume element \( \Delta V \) and distribution function \( f \) do not appear explicitly in such a case.

In applications where we construct a new distribution function from a model or where we reexpress a mesh-based distribution function in the orbit-based representation (red workflow in Fig. 2), it is necessary to determine the GC orbit volume element \( \Delta V_{\text{orb}}_{i,j,k} \) for a given mesh in CoM space. In our case, the mesh has the form shown in Fig. 6 and the associated volume elements \( \Delta V_{\text{orb}}_{i,j,k} \) in units of \([\text{m}^6\text{s}^{-3}]\) can be readily obtained when written in terms of the canonical action coordinates \((E,\mu,\nu)\) associated with the ignorable angles \((\tau,\xi,\zeta)\), taking advantage of the fact that the Jacobian for a canonical transformation is constant in both space and time (see, for instance, Ref. [36]).

With our definition \( P_\xi = -\Psi_\nu / B_0 + v_i B_\xi / (\Omega_0 B) \), which has units of \((2\pi)^{-1}[\text{m}^2]\), the transformation between Cartesian coordinates \((\eta,\xi)\) for the phase space of physical particles and canonical coordinates \((E,\tau,\mu,\zeta; P_\xi,\zeta)\) for the phase space of

---

4 An experimental Julia version of our orbit solver runs 50 times faster.
5 Describes in detail how we evaluate Eq. (25).
The index $\sigma = \frac{v_0 J_y}{|\psi J_0|}$ determines in which domain (co- or counter-going) a volume element lies, irrespective of the orbit type (trapped or passing). Using the fact that canonical volume elements are identical anywhere on a GC orbit $(E, \mu, P_c)$, Eq. (32) adds the volume elements at the high- and low-field side (HFS, LFS) midplane crossings and divides the result by 2, in accordance with our double-counting convention in Eq. (30). Integrating over the angles $\xi$ and $\zeta$, using the pitch coordinate $\Lambda = \mu B_0/E$ instead of the magnetic moment, and applying our normalizations $\hat{v} = v/v_0$ and $\hat{E} = E/(\Lambda v_0^2)$, we obtain

$$\left(2\pi^2\hat{\xi}\hat{E}\hat{\Lambda}\Delta\xi \right) \left[ \frac{dP_c d\rho}{M d\zeta} \right]_{\sigma_{\tau \rho}} + \left[ \frac{dP_c d\rho}{M d\zeta} \right]_{\sigma_{\tau \rho}} \times \frac{v_0}{2}$$

$$\approx \left(2\pi^2\hat{E}\hat{\Delta}\hat{E}\hat{\Lambda}\Delta\xi \Delta\Lambda \Delta\rho \right) \left[ \frac{dP_c d\rho}{M d\zeta} \right]_{\sigma_{\tau \rho}} \times \frac{v_0}{2}$$

for our CoM mesh with indices $(i, j, k)$ as defined in Section 6, where the cell index $k$ covers positive and negative signs of $\sigma$ for both the HFS and LFS midplane crossings. It remains to specify the increment $\Delta P_c$.

Since our CoM mesh in Fig. 6 is defined in the midplane, the canonical toroidal momentum $P_c$ for fixed $E$ and $\Lambda$ is a function of major radius only, which is here expressed in terms of $X = R - R_0$:

$$P_{c,i}^{\text{EA}}(X) = -\frac{\Psi_0(X)}{B_0} + \frac{\hat{v}_0(X)}{B(X)} B_{c,i}(X).$$

Its increment can then be evaluated as

$$\Delta P_{c,i} \approx P_{c,i} \left( X_{k+1}^{(i)} \right) \hat{E}_i \hat{\Lambda}_i - P_{c,i} \left( X_{k}^{(i)} \right) \hat{E}_i \hat{\Lambda}_i,$$

where $X_{k+1}^{(i)}$ and $X_{k}^{(i)}$ are the grid points adjacent to cell $X_k$. Alternatively, using $\eta_0 = \sigma_0 \sqrt{2E(1 - \Lambda B)}$, the increment $\Delta P_c$ can be expressed in terms of $\Delta X$ as

$$\Delta P_c = \left[ \frac{\Psi_0}{B_0} - \frac{\rho_0}{\hat{v}_0} \left( \hat{E} + \frac{\hat{v}_0^2}{2} \right) \right] \frac{B_{i} B_{i}'}{B^2}$$

where radial derivatives like $P_{c,i}^{\text{EA}}$ are taken along the midplane with fixed $E$ and $\Lambda$, and they are evaluated at cell centers $X_k$.

In summary, substituting Eq. (29) into (33) and multiplying by the number $N_c$ of markers used to represent an orbit, we obtain the GC phase space volume element represented by the orbit sample in cell $(E_i, \Lambda_i, X_i)$ in a form that can be readily evaluated numerically.\(^6\)

$$\left[ \Delta V_{\text{orb}} \right]_{i,j,k} = 4\pi^2 \hat{v}_0 \hat{E}_i \hat{\Delta} \hat{E}_i \hat{\Delta} \Lambda_j \left[ \frac{\rho_0}{\hat{v}_0} \left( \hat{E} + \frac{\hat{v}_0^2}{2} \right) \right] \times \frac{1}{2}.\quad (37)$$

The final factor $1/2$ originates from Eq. (30) via Eq. (32) and compensates our double-counting of all orbits in the domain $X_{\text{min}} \leq X \leq X_{\text{max}}$ in Fig. (6b) via index $k$.

6.2. Discussion

The formula for the volume element in Eq. (37) may look harmless, but the factor $\tau_{\text{pol}} \Delta P_c$ should be evaluated with care so as to minimize numerical inaccuracies. We have already seen in Fig. (7c,d) that $\tau_{\text{pol}}$ possesses singularities at t-p boundaries, although not all of them are problematic as we will see shortly. The existence of zeros and singularities in the increment $\Delta P_c$ can be readily anticipated from the derivative $[P_{c,i}^{\text{EA}}]_{X}$ in Eq. (36), which can be approximated at leading order as

$$P_{c,i}^{\text{EA}}(X) \approx -\frac{\Psi_0(X)}{B_0} + \frac{\rho_0}{\hat{v}_0(X)} \left( \hat{E} + \frac{\hat{v}_0^2}{2} \right).\quad (38)$$

Clearly, $\eta_0 = 0$ constitutes a singularity and there are at least two points where the terms on the right-hand side cancel, giving $[P_{c,i}^{\text{EA}}]_{X} = 0$. Figure 8 shows an example for 3.5 MeV alpha particles. We consider orbits along the black parabolic line in Fig. 8(a), where $\Lambda = 0.865$.

Figure 8(b) shows the form of the increment $\Delta P_c(X)$ evaluated using Eq. (35) (solid curves) and Eq. (36) (dotted). The values agree nicely nearly everywhere, except in the vicinity of the $\eta_0 = 0$ singularity labeled (E).\(^7\) A verification exercise similar to that performed later in Section 8.4 showed systematic errors when using Eq. (36). These errors seemed to be effectively absent when $\Delta P_c$ is evaluated using Eq. (35), which is therefore our default choice.

The zeros of $\Delta P_c(X)$ correspond to stagnation points and imply that these and nearby orbits have a small weight factor $W_{\text{pol}}$. In the example in Fig. 8, there are three such points: (C) and (F) are the O-type stagnation points in the domains of counter- and co-passing particles; whereas (D) is an X-type stagnation point situated on the separatrix of the t-p boundary. The zero of $\Delta P_c(X)$ at point (D) plays an important role: it cancels one of the three singularities of the poloidal transit time $\tau_{\text{pol}}$ in Fig. 8(c). In fact, a glance at the nearby barely trapped orbit in Fig. 8(e) shows that points (B,D,G) actually refer to the same transit time singularity; i.e., the same t-p

\(^6\) Eq. (37) differs from the formula for the volume element given in Eq. (40) of Ref. [13]. We consider the present derivation to be more rigorous. This affects mainly trapped particle orbits, which were effectively absent in previous applications of the code VisualStart.

\(^7\) We expect that the red and blue curves in Fig. 8(b,c) connect smoothly near $X = -0.4$ m if one adds grid points closer to $\eta_0 = 0$. 
a black curve in (a). In panel (b), solid lines represent \( \Lambda = \text{total number of } \) by Eq. (35) and the dotted lines are the alternative form \( j \). Blue and red lines in (b)–(d) represent orbits stemming with \( \tau \) of trapped orbit near the t-p separatrix. Panel (f) shows the time trace of point at (D) is unlikely to be perfect, so some amount of numerical inaccuracy must be expected.

The remaining transit time singularities — namely points (B) and (G) in our example — have the consequence that orbits representing those and nearby portions of GC phase space may accumulate a relatively large weight. One could compensate these singularities by demanding that the value of the distribution function \( f_{\text{orb}} \) vanishes at the t-p boundary; e.g., by including a factor \( \tau_{\text{pol}}^{1/2} \) in \( f_{\text{orb}} \). However, this may not be physically meaningful as the following arguments show.

The particles populating the barely trapped orbit in Fig. 8(e) as well as nearby potato and counter-passing orbits spend a relatively large amount of time in a small portion of pitch-position space \((v_z/v_1, R, \zeta)\) near V-type stagnation points. This fact is highlighted by the bunching of the small black circle symbols in Fig. 8(c,f), which represent 16 marker particles distributed uniformly in time. However, unlike the tiny O-type stagnation orbits around points (C) and (F), the large orbits near t-p boundaries like (B) and (G) can actually become densely populated in real plasmas. This becomes evident if one thinks in terms of particle sources: particles deposited on orbits located near a t-p boundary, like our example in Fig. 8(e), will be conveyed to the V-type stagnation points and spend a long time in that region, producing a local spike in the particle density.

Of course, the finite particle supply rate along with instabilities and collisions will prevent the formation of a true singularity in a real plasma. Nevertheless, the above arguments show that, within certain limits, the spikes in the transit times in Fig. 8(c) and resulting spikes in the orbit volume elements may adjust the weights as mentioned a few paragraphs earlier, e.g., by imposing an upper limit on their values (or their gradients) as in the flux limiter scheme of fluid dynamics. Although this trick may ensure convergence, it is likely to converge to a somewhat inaccurate value. On the other hand, one may argue that the correct value is unknown as it depends on physical mechanisms that have been ignored, such as scattering by collisions and field fluctuations. Perhaps the most elegant solution in the collisionless limit is to use an analytical estimate of a volume element near the singularity, provided that a well-behaved solution exists. Anyhow, we believe that the optimal approach to dealing with singularities depends on the application at hand, so we leave it for future research. It seems meaningful to deal with this problem in the context of smoothing techniques that will be needed to evaluate phase singularity in a real plasma. Nevertheless, the above arguments show that, within certain limits, the spikes in the transit times in Fig. 8(c) and resulting spikes in the orbit volume elements and marker weights, are physical meaningful.

Such spikes can however cause numerical problems. In some cases, the resulting inaccuracies may be tolerable. In the worst case, the presence of singularities may prevent the attainment of numerical convergence, since a finer mesh will have grid points closer to the singularity. If necessary, one may adjust the weights as mentioned a few paragraphs earlier, e.g., by imposing an upper limit on their values (or their gradients) as in the flux limiter scheme of fluid dynamics. Although this trick may ensure convergence, it is likely to converge to a somewhat inaccurate value. On the other hand, one may argue that the correct value is unknown as it depends on physical mechanisms that have been ignored, such as scattering by collisions and field fluctuations. Perhaps the most elegant solution in the collisionless limit is to use an analytical estimate of a volume element near the singularity, provided that a well-behaved solution exists. Anyhow, we believe that the optimal approach to dealing with singularities depends on the application at hand, so we leave it for future research. It seems meaningful to deal with this problem in the context of smoothing techniques that will be needed to evaluate phase
space gradients for instability analyses and may be based on physical scattering mechanisms.

7. MODELING COM DISTRIBUTIONS WITH LARGE DRIFTS

A computed or modeled equilibrium distribution $f_{\text{eq}}$ is needed to determine the weight factor in Eq. (31). A model that is a function of CoM only, such as $G_{\text{mdl}}(E, \mu, P_\parallel)$, would directly yield an exact equilibrium distribution $f_{\text{eq}}$. While simple models may suffice for physics studies at a fundamental level, predictive simulations and simulations used to interpret observations require more realism. It would be convenient to have $G_{\text{mdl}}$ given in an explicit, purely analytical form. However, for fast ions with significant magnetic drifts, a large number of control parameters is usually necessary to mimic realistic distribution functions, which tends to make the modeling job cumbersome, at least for non-artificial intelligence.

For instance, consider the problem of constructing a model for fusion-born alpha particles. Even if one assumes that their pitch angle distribution is uniform at birth time, strong pitch angle nonuniformities can promptly arise within one poloidal transit time. The loss cone, which contains orbits whose trajectories intersect plasma-facing components, is an obvious example. The magnetic drifts that cause those losses in toroidal geometry also affect the pitch angle distribution of confined orbits. For instance, for fusion-born alpha particles in ITER, the anisotropy associated with drift orbit topology was estimated to be around 10% (see Section 7 of Ref. [45]). Pitch anisotropies and bump-on-tail structures along the energy axis are also a well-known cause of velocity space instabilities that can be observed as ion cyclotron emissions (ICE) (e.g., see Refs. [47, 48]).

Let us consider a more concrete example. Figure 9 shows the GC orbit of a barely trapped 3.5 MeV alpha particle, which traverses both the hot plasma core and the cooler plasma periphery. If one uses a slowly varying function $G_{\text{mdl}}(E, \mu, P_\parallel)$ to model the overall distribution of alpha particles in this range of energies and pitch angles, the density will be similar on the barely trapped orbit shown in Fig. 9 and on the neighboring large potato and small counter-passing orbits indicated by dashed lines, because their coordinates have similar values. In other words, the particle density will be similar on both sides of the t-p boundary.

However, this may not be an accurate representation of a real distribution of fusion-born alphas, in particular when the fusion reactivity varies substantially in different portions of the orbit. To illustrate this, the 3.5 MeV alpha particle orbit in Fig. 9 has been underlaid with the contours of the sharply peaked MHD pressure field from of our working example (c.f., Fig. 5), which is characteristic for the bulk component of a JET plasma where 3-ion RF heating was applied near the magnetic axis [41, 42]. In such a case, fusion-born alphas from the plasma center will populate the entire orbit shown in Fig. 9 albeit in a nonuniform manner as indicated by the small black circles. Just across the t-p boundary, however, most of the newly born alphas will be concentrated on the small counter-passing orbit, while the large potato orbit would be populated only sparsely, since it is topologically disconnected from the particle source. In principle, this can produce a steep gradient in the distribution function $f_{\text{eq}}(E, \mu, P_\parallel)$ across the t-p boundary. In reality, collisions or instabilities can be expected to have a smoothing effect, but depending on the relevant time scales, a strong nonuniformity may be sustained. Such information may be necessary for interpreting experimentally observed signals or for reproducing them in simulations, and it is advantageous to have the ability to construct suitable models.

A viable recipe for doing so can be found in Eq. (30). Such an orbit time average can be readily realized using our orbit database and constitutes a physically meaningful way to convert an arbitrary model function $G_{\text{mdl}}(Z)$ into a true equilib-
rium distribution $f_{orb}$:

$$f_{orb} = \langle G_{mdl}(P_P) \rangle_{orb} \equiv \frac{1}{N_{orb}} \int d\tau G_{mdl}(Z(\tau))$$

$$\approx f_{orb}(i, j, k) = \frac{1}{N_{i,j,k}} \sum_{l=1}^{N_{i,j,k}} G_{mdl}(Z_{xc,j,k})$$

where $Z_{xc,j,k}$ is the position of marker $l$ on orbit $(i, j, k)$. For instance, in the example discussed in the previous paragraphs, one may let $G_{mdl} \propto P(\Psi_p)$, so that the MHD pressure field in Fig. 9 acts as a particle source.

This approach can be viewed as taking the path of “self-organization”, where we leave it to the equations of motion to determine a distribution function that is consistent with the magnetic geometry and system boundaries at hand, and the constraints encoded in $G_{mdl}$ determine the particular spatial profile and velocity distribution for a certain case. This idea also underlies Monte-Carlo codes that follow GC orbits many times around the torus in the presence of sources and collisions. We mimic a part of this process with reduced computational effort: the GC orbits in our database are followed for only one poloidal turn in the absence of any perturbations (field fluctuations and collisions).

8. APPLICATION AND VERIFICATION

In this section, we demonstrate how the methods that we presented in this paper can be applied to a practical example. Section 8.1 describes the setup and how we construct — from a simple model $G_{mdl}$ — an equilibrium distribution $f_{orb}$ of alpha particles that are assumed to be born near the center of a large tokamak plasma with an isotropic distribution of pitch angles at birth time. The results are discussed in Sections 8.2 and 8.3. In the final Section 8.4 we demonstrate how we use our orbit database to convert a binned 4-D distribution function $f^{(0)}(E, A, R, z)$ (or any other coordinates) back to an orbit-based representation $f^{(1)}$ in Com space. The result is binned again to give $f^{(1)}$ for comparison with the original $f^{(0)}$. The procedure is then iterated one more time, yielding $f^{(2)}$, in order to reveal systematic errors. The contents of this section may be summarized in diagrammatic form like this:

\[
\begin{array}{ccc}
\text{Modeling} & \rightarrow & \text{Com space} \\
G_{mdl} & \rightarrow & f^{(0)}_{orb}(i, j, k) \\
& \rightarrow & f^{(0)}(E, A, R, z) \\
& \rightarrow & f^{(1)}(E, A, R, z) \\
& \rightarrow & f^{(2)}(E, A, R, z) \\
\end{array}
\]

Strictly speaking, we transform only between 2-D and 3-D spaces, $(A, P(T)) \leftrightarrow (\lambda, R, z)$, since both sets share the energy coordinate $E$, which is a conserved quantity, so the mapping in that direction is identical. We will work here with an arbitrary normalization (henceforth indicated by hats), so the values of $f_{orb}$ of $f$ must match but bear no particular meaning.

Our distribution functions will be inspected here in the same way as an experimental diagnosticist would: measure spatial distributions by integrating over certain portions of velocity space, and measure velocity distributions by integrating over certain portions of position space. The required binning operations and Jacobians are described in [A]

8.1. Setup and parameters

In order to highlight how toroidal geometry affects the form of the constructed equilibrium distribution function via the mirror force and magnetic drifts, we start from a quasi-distribution $G_{mdl}$ that is independent of pitch and has a relatively sharp peak in minor radius:

$$G_{mdl}(P_P) = C \times \hat{h}_{mdl}(P_P) \times H_{mdl}(E),$$

where $C$ is a normalization constant that is arbitrary here.

The radial density profile $\hat{h}_{mdl}(P_P)$ is shown as a solid black line in Fig. 10(a). Although this is primarily meant to be a toy model for testing our algorithms with a localized particle source and large magnetic drifts, we note that this profile is physically feasible if one considers scenarios with a sharply peaked ion temperature profile as produced in 3-ion-heated JET plasmas [51, 52].

The energy dependence of particle distributions in a real plasma is governed by processes such as collisional drag and instabilities (e.g., see Fig. 4 in Ref. [17]). These processes are not considered here. We use energy only to exemplify the
TABLE II. Phase space sampling parameters in our five test cases. $N_c$, $N_x$, $N_y$ determine the number of cells in $(E, \alpha, X)$-space defined in Section 4. The number $N_i$ of markers sampling an orbit is (with a lower bound of 16) chosen to be proportional to the orbit length $L_{orb}$, the reference value being the number of markers per plasma boundary length, here $L_{orb} \approx 7.50$ m. There are $N_{mk}$ markers sampling $N_{orb}$ confined orbits.

| $E$ [keV] | Circ. | Stagn. | Banana | Potato | Lost $\approx$ |
|-----------|-------|--------|--------|--------|-------------|
| 35        | 25,780| 359    | 7,158  | 38     | 780         |
| 350       | 24,430| 1,066  | 6,191  | 154    | 2,280       |
| 3500      | 20,531| 3,165  | 3,505  | 483    | 6,430       |
| 0...35    | 87,368| 612    | 24,456 | 47     | 1,710       |
| 0...3500  | 75,178| 7,208  | 15,861 | 1,061  | 14,880      |

TABLE III. Partition of orbit types (circular, stagnation, banana, potato, lost) as defined in Section 4. These numbers are not weighted by particle densities, so they only characterize the partition of orbit samples on our mesh. The actual prompt losses of real particles are negligible in our large plasma with centrally peaked density profile.

Table II shows the parameters for five test cases that we will consider here: three mono-energetic alpha distributions with $E = E_0 = (35, 350, 3500)$ keV and two distributions that are uniform in the energy range $0 \leq E \leq E_0$ with $E_0 = (35, 3500)$ keV. Table III summarizes the numbers of circulating, stagnation, banana, potato and lost orbit samples (CoM mesh points) in each case. Panels (b) and (c) of Fig. 10 show how the contours of a banana orbit (A) and a circulating orbit (B) change due to the effect of magnetic drifts when the kinetic energy is increased from 35 keV to 3.5 MeV.

While the model in Eq. (44) has a simple appearance, the actual form of the alpha particle distributions we construct here is determined by the combination of Eq. (44) and the equations of GC motion (21) and (22), whose solutions are folded into the simple model via the integral (59). In other words, the geometric effects are captured by the GC orbits in our database, whereas the model $G_{rad}$ effectively describes the structure of the particle sources that populate those orbits, and the integral (59) combines all that information.

The geometric effects that we expect to see in this setup may be summarized as follows. These properties of a toroidally confined and radially bounded plasma are well-known and are listed here only for completeness and to simplify the discussion of the results:

- The peripheral plasma will be populated by particles on orbits that also pass through the core as in Fig. 10(c). Such orbits are found in a certain range of pitch angles around the t-p boundary.
- Flux surface-averaged profiles tend to be smoothed on the scale length of the magnetic drifts. The gyroradii have a similar effect and may be included via satellite particles [49] (not done here).
- Magnetic drifts shift co-passing orbits outward in $R$ and counter-passing orbits inward.
- The mirror force confines some particles to the low-field side of the plasma and causes them to accumulate near $V$-type stagnation points.
- The energy- and pitch-dependence of the above-mentioned spatial nonuniformities gives rise to net flows both poloidally (due to radial nonuniformity) and toroidally (due to a local imbalance between co- and counter-going particles).

8.2. Properties of mono-energetic distributions

Figure 10(a) shows the computed density profiles $\hat{n}(\rho_p)$ on a mesh that is uniformly spaced in normalized poloidal flux $\psi_p = \rho_p^2$. The profile for 35 keV alphas is similar to the model $\hat{n}(\rho_p)$. The small discrepancy at $\rho_p = 0$ can be eliminated with a finer mesh as shown in the inset. The magnetic drifts increase with increasing energy, causing the profile $\hat{n}(\rho_p)$ to broaden and the central value $\hat{n}(0)$ to drop. The total number of particles in the plasma is nearly the same in all three cases, since prompt losses are relatively small: less than 1% for 350 keV and about 2.5% for 3.5 MeV.

Figures 11 and 12 show the form of various moments of the alpha distributions with kinetic energies 35 keV and 3.5 MeV in the poloidal plane $(X, Y)$:

- Particle density (a): Even for 3.5 MeV, our model yields an alpha density that is peaked at the axis, but the peak is wider than in the 35 keV case, especially in the horizontal $(X = R - R_0)$ direction.
FIG. 11. Moments of the mono-energetic alpha distribution with 35 keV. We integrated over velocity space using Eq. (A5) with the following weight factors: (a) $w$ for particle density $\hat{n}(X,Y)$; (b) 1 for the number of markers per cell; (c) $w \times E/3$ for alpha particle pressure $\hat{P}_a$, then divided by density to yield temperature $T_a = \hat{P}_a/\hat{n}$; (d)–(g) $w \times (j_R, j_z, j_\zeta, j_\parallel)$ for horizontal, vertical, toroidal and parallel current densities $(j_R, j_z, j_\zeta, j_\parallel)$. The amplitude is normalized to $\hat{n}_0 = 1$.

- **Markers/cell (b):** Since we have set a lower bound of 16 markers per orbit, the smallest stagnation orbits stand out in the plots showing the number of markers per cell, while the surroundings are fairly uniform. At 35 keV, the co- and counter-passing stagnation points effectively coincide with the magnetic axis. At 3.5 MeV, their mean separation along $X$ is about 0.14 m (cf. Fig. 8(b)).

- **Temperature (c):** With the density nonuniformity divided out, the 3.5 MeV temperature field clearly shows that our alpha particle distribution effectively consists of two partially overlapping co- and counter-going mono-energetic beams, which are shifted radially out- and inward, respectively.

- **Flows (d–g):** The relative shift of the co- and counter-going beams causes net local currents. The poloidal current field (consisting of $j_R$ and $j_z$) forms two vortices, one on the high-field side (HFS) of the magnetic axis rotating clockwise, and one on the low-field side (LFS) rotating counter-clockwise. Toroidal and parallel currents $j_\zeta$ and $j_\parallel$ are very similar here, since the magnetic pitch is relatively uniform ($q \approx 1$) in the plasma center. The structure and spatial extent of the currents is similar at lower and higher energies, but the magnitude should be proportional to $\sqrt{E}$ and, thus, differ by about a factor of 10. This seems to be the case here if one accounts for the fact that the 35 keV case has a smaller signal-to-noise ratio.

FIG. 12. Moments of the mono-energetic alpha distribution with 3.5 MeV. Arranged as Fig. 11.

FIG. 13. Pitch angle distributions for the mono-energetic cases. Panel (a) shows pitch profiles $\hat{\nu}(\alpha|E_0)$ as defined in Eq. (A6). The black dotted curve shows the 3.5 MeV result for a reduced domain size, where the artificial plasma boundary has been placed at 0.25% of the flux space [33], removing many large orbits (scaled to match the blue curve outside the loss cone). Panels (b) and (c) show contours of the local pitch distribution $\hat{\nu}(X,\alpha|E_0)$ obtained by integrating over the height of the spatial domain in radial bins of size $\Delta X = 0.04$ m. The two crosses and two plus symbols (sometimes overlapping) in panels (c) and (d) mark the starting points of four alpha particle orbits with $E_0 = 3.5$ MeV whose poloidal contours are shown in panel (d). Blue and orange orbits start with $\alpha = -0.09\pi$ and $+0.09\pi$, respectively. Solid orbits start from $Y = 0$ (crosses) and dashed orbits from $Y = 0.56$ m (pluses). Panel (d) also shows the color contours of the 3.5 MeV alpha particle density field from Fig. 12(a), $\Psi_p$ contours (white), the midplane (dark green), and the boundary (light green).
Figure 13 shows the pitch angle distribution of our monoengetic alpha particles. The spatially integrated distributions \( \hat{\nu}(\alpha|E_0) \) in panel (a) show that a pitch nonuniformity exists around \( \alpha = 0 \), up to intermediate pitch angles, \( |\alpha| \lesssim 0.2\pi \). The deviation from the mean increases from about 10\% for 35 keV to about 50\% for 3.5 MeV. At larger pitch angles, \( |\alpha| \gtrsim 0.2\pi \), the distributions are flat (except for aliasing noise, to be discussed later). More detailed views of the nonuniformities in the region \( |\alpha| \lesssim 0.2\pi \) of the cases with \( E_0 = 35 \) keV and 3.5 MeV are provided in panels (b) and (c), where we plot the contours of \( \hat{\nu}(X, \alpha|E_0) \) to show the radial dependence of the pitch angle distribution in the range \(-0.48 \text{ m} < X < 0.48 \text{ m}\). Here, the summation over markers was performed over the \( Y \) coordinate for radial bins of size \( \Delta X = 0.04 \text{ m} \).

The structures seen in Fig. 13 can be readily explained on the basis of our model function \( G_{mdl} \), the effects of toroidal geometry (mirror force and magnetic drifts), and the loss boundary. Rather than saying anything new, the purpose of the following discussion is to show that our procedures work reliably and produce the expected results. We focus on the 3.5 MeV case.

Figure 13(c) clearly shows the opposite radial shift of co- and counter-going particles due to their opposite magnetic drifts: the peak in the region \( \alpha > 0 \) is shifted towards the right \( (X > 0) \) and the peak in the region \( \alpha < 0 \) is shifted to the left \( (X < 0) \). The wedge-shaped structure of the pitch anisotropy around \( \alpha = 0 \) represents the widening of the domain of trapped particles towards the low-field side as expected from Fig. 8(a).

In order to explain the structure of the pitch anisotropy in the region \( |\alpha| \lesssim 0.2\pi \), it is helpful to consider the trajectories of a few GC orbits in that region. For this purpose, Fig. 13(d) shows the GC orbits of four simulation particles with energy \( E = 3.5 \) MeV. Two of them (solid lines) are launched near the midplane \( (Y_{start} \approx 0) \): the small orange orbit starts from \( (X, \alpha)_{start} = (0.05 \text{ m}, 0.09\pi) \), and the blue orbit starts from \( (X, \alpha)_{start} = (-0.05 \text{ m}, -0.09\pi) \), as indicated by the crosses in panels (c) and (d). The blue orbit is a large potato orbit that passes both near the center of the plasma and through the perriphery. The small orange orbit is a stagnation orbit. Now recall Eq. 19, where we defined the orbit distribution \( f_{orb} \) to be the orbit time-average of the model \( G_{mdl}(\rho P) \) that represents our alpha particle birth profile. Since the latter is sharply peaked near the plasma center, the values of \( G_{mdl}(\rho P) \) can differ substantially on the above-mentioned pair of orbits.

Generally speaking, orbits that spend most of their time inside the peak of the source profile \( G_{mdl}(\rho P) \) are densely populated with physical particles, so our GC markers representing them have larger weights \( w_j \). Orbits that spend most of the time outside are sparsely populated. The mirror force and magnetic drifts have the consequence that particles on counter-passing orbits spend less time inside the central peak of \( G_{mdl}(\rho P) \) than co-passing orbits. In addition, the losses of co- and counter-going particles are asymmetric, especially in the vicinity of the trapped-passing boundary. These effects are manifested in the minima and maxima of \( \hat{\nu}(\alpha|E_0) \) and \( \hat{\nu}(X, \alpha|E_0) \) in Fig. 13, which are most pronounced in the high-energy case due to its larger drifts.

Note that the distribution \( \hat{\nu}(X, \alpha|E_0) \) in Fig. 13(c) was computed by integrating along the vertical coordinate \( Y \), so it also includes particles on orbits like those shown by dashed lines in panel (d). The dashed orbits were launched from \( (X, \alpha)_{start} = (0.05 \text{ m}, \pm 0.09\pi) \), but more than half a meter above the midplane, from \( Y_{start} = 0.56 \text{ m} \), as indicated by pluses in panels (c) and (d). The orbit starting with a positive pitch (orange) passes through the central plasma, where it acquires a relatively large weight due to the centrally peaked source profile \( G_{mdl}(\rho P) \). In contrast, the orbit starting with a negative pitch travels outward and nearly hits the boundary (light green curve).

In fact, the entire peak of \( \hat{\nu}(\alpha|E_0) \) around \( \alpha/\pi \approx 0.1 \) can be explained in that way. We have verified this with a test case where the plasma boundary had been placed at \( \phi_P = 0.25 \) (\( \rho_P = 0.5 \)) instead of \( \phi_P = 1 \), so that orbits are constrained to the inner 25\% of the present flux space [33]. In that case, the plasma boundary is located around \( X = 0.38 \text{ m} \) at the height of the outer midplane, so that the three large orbits in Fig. 13(d) are all lost. The resulting pitch angle profile is shown as a black dotted line in Fig. 13(a). It has a deeper and wider loss cone, and there is no peak at \( \alpha/\pi \approx 0.1 \).

8.3. Properties of flat-energy distributions

Realistic alpha particle distributions are nonuniform in energy. The reason for our use of a model that is flat in energy is that the \( E \)-dependence of geometrically induced nonuniformities can be easily seen, and it is also easier to verify that our method works as expected.

The densities and current fields are very similar to those in the mono-energetic cases shown in Figs. 11 and 12 above. A few different features can be observed in the high energy case, \( E = 0...3.5 \) MeV, whose moments are shown in Fig. 13.
presence of particles with a wide range of energies can be inferred primarily from the blurring of the stagnation points in panel (b), and from the nonuniformity of the temperature field in (c). The latter resembles somewhat the letters “IO” as fast particles dominate outside the central density peak; especially the co-passing ones (forming the “O” shifted to the right) for plots of \( \hat{v}(E, \lambda) \), see Fig. 16. Note that different color scales are used in the upper and lower row.

Our orbit database is relatively sparse \((N_E \times N_\alpha \times N_\kappa = 24 \times 48 \times 48)\), and the bins used in Figs. 14 and 15 have effectively the same resolution: \( N_E \times N_\alpha \times N_\kappa = 24 \times 48 \times 48 \). The binned results are smooth along energy, because the orbit samples coincide with the bins. Most of the noise in Fig. 15 is in the pitches \( \alpha \) and \( \Lambda \), where the distribution of bins is similar but not identical to the distribution of orbit samples (CoM space). In other words, the noise in the range \( 0.3 \leq |\alpha|/\pi \leq 0.5 \) and \( \Lambda \leq 0.7 \) is effectively an aliasing effect. The noise is largely suppressed in the poloidal plane \((X, Y)\) in Fig. 14 and in \((\hat{h}_p, \mu)\)-space shown in the central column of Fig. 15 since the bins in these coordinates have a shape very different from the mesh we used to sample the CoM space (cf. Figs. 6 and 16).

8.4. Verification of reversibility and accuracy of coordinate conversions via an orbit-based representation

The results discussed in the previous sections were obtained by binning the marker weights of our orbit-based representation in CoM space to a mesh in arbitrary coordinates. The reverse transformation is also easily performed as we will demonstrate in this section. On platforms like ITER IMAS, we expect this to be an essential part of the transformation toolbox, especially if an orbit-based representation is used as a reference. Here, this operation serves us primarily as a ver-

Figure 16 also highlights potential difficulties in attempts to convert the binned distributions from one set of coordinates to another. In particular, the small region around the origin \((\mu, v_0) = (0, 0)\) in the central column of Fig. 15 is expanded to the full length of an axis in \((E, \alpha)\) and \((\Lambda, \bar{E})\) spaces on the left and right sides of Fig. 15 so some features of the distribution functions are likely to get lost in direct conversions unless a polar mesh is used in \((\mu, v_0)\) coordinates, which is equivalent to using another set of coordinates, such as \((E, \alpha)\). We emphasize that we have not performed direct conversions between binned distributions in Fig. 15. They were all obtained by binning, on the respective mesh, the marker particles of the orbit-based representation \( \hat{f}_{orb} \).

The binning in Figs. 10-15 was performed using a 1st-order (linear) interpolation scheme as commonly used in particle-in-cell (PIC) codes, where each marker is assumed to have the shape of a top-hat function whose width is equal to the bin size. Naturally, the noise is larger with a 0th-order (binary) binning scheme, where markers resemble \( \delta \) functions. The difference between 1st- and 0th-order binning can be seen by comparing panels (a) and (b) of Fig. 16 where we show the spatially integrated velocity distribution \( \hat{v}(E, \lambda) \) with \( \lambda = v_0/v = \sin^{-1}(\alpha) \). Note also that the result in Fig. 16(a) appears smoother than that in \( \hat{f}_{orb} \) (c) simply by using \( \lambda \) instead of \( \alpha \), because this change in coordinates reduces the above-mentioned aliasing effect. Another difference is that Fig. 16 was obtained with 10 times more markers per orbit, but this does not have a strong influence on the aliasing in the domain of deeply passing particles, since those markers rarely cross the bin boundaries.
ification exercise to test with what degree of accuracy we can perform and reverse coordinate conversions without applying any smoothing algorithms.

We consider the flat-energy case with \( E_0 = 3.5 \) MeV and follow the procedure that was outlined in Eq. \( 40 \). Let us now define the relevant steps in detail:

1. Using Eqs. \( 39 \) and \( 41 \), we construct a distribution function \( f_{orb} \) in the orbit-based representation:
   \[
   G_{\text{mdl}} \rightarrow f_{orb}(i, j, k). \tag{43}
   \]

   The CoM mesh that underlies our orbit database has the same number of cells as specified in Table \( \| \) \( N_x \times (N_y \times N_k) = 24 \times 48^2 \). The number of markers per boundary length is increased by a factor 10 to \( N_m = 480 \times L_{\text{orb}}/L_{\text{bin}} \) (min. \( 16 \)), giving a total of \( N_{\text{mark}} = 18 \) M markers.

2. The orbit-based CoM distribution \( f_{orb} \) is binned on a uniform 4-D mesh in the widely used coordinates \((E, \lambda, R, z)\). In compact form, this operation can be written as
   \[
   f_{orb}^{(q)}(i, j, k) \rightarrow f^{(q)}(E, \lambda, R, z) \tag{44}
   \]

   where \( \eta = 0, 1, 2 \) is the iteration number. The initial result \( f^{(0)} \) is referred to as the “input” distribution for the subsequent back-and-forth transformations. More precisely, the binning operation we perform is given by Eq. \( 43 \) in the form
   \[
   f(E_i, \lambda_j, R_k, z_m) = \sum_{n=1}^{N_{\text{mark}}} w_n \cdot \xi(I(Z_{\text{gc,n}} = Z_{ijkm})) \tag{45}
   \]

   where the shape factor \( 0 \leq \xi \leq 1 \) is a top-hat function, which means that the markers resemble \( \delta \) functions. That is, we use the 0th-order binning scheme as in Fig. \( \ref{fig:16} \) (b) and the number of bins (= cells) is \( N_x \times N_y \times (N_k \times N_z) = 24 \times 24 \times 48^2 \). The pitch resolution was reduced from \( N_{\text{pitch}} = 48 \) to 24 in order to reduce noise and aliasing (cf. Section \( \ref{sssec:res} \)).

3. For each orbit in the database, we compute new weight factors \( W^{(q+1)}_{ijk} \) by integrating \( f^{(q)}(E, \lambda, R, z) \) in time \( \tau \) along the orbit contour as
   \[
   W^{(q+1)}_{ijk} = \left| V_{\text{orb}} \sum_{l=1}^{N_z} f^{(q)}(E, \lambda(\tau_l), R(\tau_l), z(\tau_l)) \right|_{ijk}, \tag{46}
   \]

   which gives a new orbit-based distribution:
   \[
   f^{(q+1)}(E, \lambda, R, z) \rightarrow f^{(q+1)}_{orb}(i, j, k). \tag{47}
   \]

4. Repeat step 2 and compare the result with the previous iteration:
   \[
   f^{(q+1)} \approx f^{(q)}. \tag{48}
   \]

We have then iterated steps 3 and 4 one more time, looking for increasing deviations that may hint at systematic errors. The results are summarized in Figs. \ref{fig:17} and \ref{fig:18}.

Figure \ref{fig:17} shows that the overall shape of the radial density profile is reproduced well. A closer inspection of the relative errors in Fig. \ref{fig:17} reveals that the first iteration changes the particle density by about \(+2\%\) in the densely populated re-
gion \( r/a \lesssim 0.3 \). Around \( r/a \approx 0.7 \) the increase is about +10\%, and near the boundary we have a reduction by about −20\%. The relative error roughly doubles in the second iteration, indicating that it is not random but systematic. As indicated in the legend of Fig. 17(a), the total number of GCs \( N \) increases by 1.9\% in the first and 2.4\% in the second iteration, where the error accumulates to 4.3\%.

Figure 17(b) shows that our procedure has the tendency to increase the particle density in the plasma interior and reduce it near the plasma boundary. We have not investigated why the systematic errors have this particular form and it may, in fact, depend on the particular magnetic configuration at hand. However, the larger errors in the outer region of the plasma are likely related to the fact that, in our test case, this region is populated almost exclusively by particles on large orbits near the trapped-passing boundary (cf. Fig. 10(c)), where the poloidal transit time \( \tau_{pol} \) exhibits singular behavior (see Fig. 8) and inaccuracies can be expected. In the present case, the magnitude of these errors can be reduced by increasing the number of phase space samples as shown in Appendix B.3.

Figure 18 shows the results for the velocity space distributions \( \tilde{v}^{(0)}(E, \alpha) \) and \( \tilde{v}^{(0)}(E, \lambda) \) for iterations \( \eta = 0, 1, 2 \). No significant differences can be seen in the energy direction, which is to be expected since it is a conserved quantity and our energy bins are identical in both representations. But even along the pitch coordinates \( \alpha, \lambda \), excellent agreement is obtained after dividing out the systematic increase of \( N \). The noise at large pitch values retains its form as expected for an aliasing effect, and its magnitude increases systematically from one iteration to the next. The minimum associated with the loss kernel tends to deepen with each iteration, which is also to be expected due to the irreversibility of losses. As was noted in connection with Fig. 16 we see again in Fig. 18 that binning on a uniform mesh in \( \lambda \) instead of \( \alpha \) reduces aliasing and, thus, yields smoother results in the deeply passing domain \( (|\alpha| \gtrsim 0.3\sigma \text{ or } |\lambda| \gtrsim 0.8) \).

Before proceeding to the conclusion of this paper, we shall add a few encouraging words for readers who may find the results in this section somewhat unsatisfactory. First, it should be noted that we have chosen a fairly challenging setting for our application example: Our density peak is localized deeply in the plasma core, where derivatives of the poloidal magnetic flux \( \Psi_p \) are small so that \( P_i \) is dominated by drift terms. This means the so-called “non-standard” orbits are actually prevalent here. Second, we have sampled the CoM space only relatively sparsely (Table 11) and used no smoothing procedure whatsoever. Even the binning of particles was done using a 0th-order scheme, which is why the noise in Fig. 18 is larger than in Fig. 13.

In other words, one may say we have been looking here at a worst-case scenario with the least degree of sophistication. This should be taken in a positive way, because it means that there is much room for improvement even with well-established techniques. Higher resolution in CoM space, higher-order binning schemes and other smoothing techniques are obvious solutions to reduce noise to a level where the distribution becomes sufficiently smooth to measure local phase space gradients of a particle distribution. The application of such techniques in combination with the orbit-based representation method described here is left for future work.

9. SUMMARY AND CONCLUSION

This paper was motivated by the need to model and process distributions of charged particles in tokamak plasmas in a unified framework like the ITER Integrated Modelling & Analysis Suite (IMAS), which has to address the needs of a diverse community of users dealing with experimental and numerical data. One of the tasks IMAS will need to perform is the conversion of distribution functions between various coordinates. Our proposal is to construct a database of unperturbed GC drift orbits and use it as a basis for representing distribution functions in constants-of-motion (CoM) space. A working prototype of a concrete workflow as implemented in the code VisualStart [12, 13] is described here.

It should be noted that dedicated IMAS data structures for storing GC orbit properties and quantities along GC trajectories of marker distributions have recently been added into the ‘distributions’ IDS (Interface Data Structures) [50], making IMAS readily compatible with the representation and methods proposed here.

An orbit-based representation maps any input distribution to an exact equilibrium that is consistent with the specified magnetic field configuration. While the result may generally differ from the original input, the equilibrium constraint ensures that all subsequent mappings from 3-D CoM orbit space back to arbitrary 4-D sets of coordinates become unique and straightforward. In fact, further operations like visualization or exporting are reduced to binning marker particles on user-defined meshes in arbitrary coordinates and arbitrary subspaces.

A new orbit database needs to be set up for each MHD equilibrium, and one may even have to set up one for different energy ranges (say, 1...tens keV, and 0.1...few MeV). The required computational effort should be manageable if the code is optimized and capable of exploiting state-of-the-art computing technology. Once the database has been constructed, it can be used to represent and model various populations of charged particles that may be present in the same plasma, such as beam ions, RF-heated ions and fusion products. One orbit database may be shared between particle species with identical charge-to-mass ratio \( Qe/M \) by scaling their energies.

The time spent on preparing the orbit database is also justified by the fact that the acquired information can be used for other important tasks, such as resonance analyses and modeling, which must also be part of the IMAS toolbox. The degrees of freedom for modeling are vast, since each orbit can be assigned an independent weight. Rather than prescribing parametric functions of CoM — whose limitations were briefly discussed in Section 7 — it is worth exploring unconventional ways that make better use of the degrees of freedom available and take advantage of the fact that our orbit-based representation automatically enforces the equilibrium condition and physical constraints associated with toroidal geometry; namely, the mirror force, magnetic drifts and the resulting
velocity space nonuniformities and boundary losses. As an example, we have demonstrated in Section 8 how the orbit database can be used to model a distribution of alpha particles with a core-localized birth profile that is not a function of the fast alpha’s CoM. This method to design distribution functions was inspired by orbit-following Monte-Carlo simulations, but is much cheaper computationally. We expect that similar strategies can be used to model distribution functions based on various types of experimentally measured data or from results of bounce-averaged Fokker-Planck codes that ignore magnetic drifts [28].

Finally, we demonstrated in Section 8.4 how a distribution of energetic alpha particles with large magnetic drifts can be converted back and forth between a 4-D mesh in arbitrary coordinates and the orbit-based representation in 3-D CoM space. Besides being an essential part of the coordinate transformation toolbox needed by platforms like IMAS, this procedure served us as a method to verify the numerical accuracy and logical integrity of the workflow. In fact, this procedure enabled us to identify and correct subtle problems that arose in early versions of our implementation of the algorithm. The lessons we learned are documented in this paper.

The methods presented here can serve as a starting point that already provides much of the required functionality required by IMAS. One of the next steps should be the adoption and development of smoothing techniques with the goal to mitigate the effect of discretization noise, while minimizing systematic errors associated with such irreversible data manipulations. This is a prerequisite for performing stability analyses of resonant modes and for preparing initial conditions for delta-f simulations, which require explicit information about gradients in the phase space density.

We suggest the reader to also follow related and complementary developments in orbit tomography [4, 8, 25]. For instance, those studies demonstrate an alternative way to sample the orbit space and employ methods for adaptive meshing and reducing binning errors.

ACKNOWLEDGMENTS

One of the authors (A.B.) acknowledges insightful discussions with Yasushi Todo concerning the handling of the Jacobian factor $B^i_l$ and with Stuart Benjamin and David Pfefferle concerning methods for phase space discretization and mapping techniques. On a broader note, A.B. would like to thank Liu Chen, Zhihong Lin, Fulvio Zonca as well as Sergio Briguglio, Claudio DiTroia and Gregorio Vlad for their support during the early development stages of VisualStart in 2008–2010 for the hybrid code HMGC. Subsequent extensions enabling VisualStart to initialize long-time simulations performed for JT-60U with the code MEGA benefited from support by Kouju Shinohara, Yasushi Todo and Masatoshi Yagi, to whom A.B. would also like to express his gratitude. Finally, A.B. is grateful to Jeronimo Garcia and Shunsuke Ide for encouraging and supporting a fruitful collaboration between JET and QST. The working example used in the present paper was inspired by plasmas produced in recent JET experiments with central 3-ion RF heating, which posed a suitably challenging and practically relevant scenario for testing our numerical algorithms and workflow. We would like to thank all JET contributors for making these interesting experiments possible and allowing us to participate.

The workstation used for the numerical calculations reported here was funded by QST President’s Strategic Grant (Creative Research). This work has been partially carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 and 2019-2020 under Grant Agreement No. 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A: Binning operations and Jacobians

In this section, we describe how we evaluated the Klimontovich representation [25] to visualize our results in Section 8. For clarity, we break the procedure up into (i) the computation of a histogram $h$ and (ii) its conversion to a density function $f$. As a concrete example, we use the set of noncanonical coordinates $(E, \lambda, R, Z)$. The cell center positions are $Z_{i,j,k,m} = (E_i, \lambda_j, R_k, Z_m)$ and the volume elements are $\Delta V_{i,j,k,m} = \Delta E_i \Delta \lambda_j \Delta R_k \Delta Z_m$. The corresponding histogram of GCs is given by

$$h(Z_{i,j,k,m}) = \sum_{l=1}^{N_{\text{mark}}} w_l S(Z_{gc,l} - Z_{i,j,k,m}) / \Delta V_{i,j,k,m}.$$  \hfill (A1)

Marker $l$ is located at GC position $Z_{gc,l}$, and its weight $w_l$ determines how many physical particles this marker is meant to represent (cf. Eq. (25)). The shape function $S$ determines how many physical particles this marker is meant to represent (cf. Eq. (25)). The shape function $S$ is bounded such that $0 \leq S \leq 1$ and

$$f = \frac{h}{f^{(\text{orb})}} \approx \frac{h(Z_{i,j,k,m})}{\int_E f^{(\text{orb})} f_e^{(\text{orb})} \Delta E}.$$  \hfill (A2)

where $f^{(\text{orb})}$ is the Jacobian for the conversion from Cartesian to noncanonical GC coordinates. Substituting Eq. (A1) into (A2) and assuming, within the accuracy limits of our discrete mesh, that $[\langle l \rangle]_{ijk,m} \approx \lfloor l \rfloor$, the formula for mapping the phase

---

11 In contrast to the canonical 6-D phase space volume elements $\Delta V_{\text{orb}}$ and $\Delta V$ of GC orbits and their markers as defined in Section 6 the symbol $\Delta V$ appearing here denotes an element in an arbitrary mesh in an arbitrary number of dimensions.
space density function $f$ to our 4-D mesh becomes\footnote{Conventional PIC codes sample the phase space by loading marker particles in a uniformly randomized manner at the beginning of a simulation ($t=0$). In that case, the initial marker weights (via their volume elements) carry a factor $B_i'(t=0)/B(t=0)$. When evaluating moments of the distribution function during a simulation, that factor is effectively multiplied by the inverse factor $B(t)/B_i'(t)$ that appears in Eq. (A3). These two factors cancel approximately and may, thus, be omitted, if one ignores corrections of order $O(\sigma_0^2/\alpha^2)$.}

$$f(E_r, A_p, R_i, z_m) = \sum_{n=1}^{N_{\text{vol}}} w_l \times S(z_e - z_i, \cdots) \Delta V_{\text{vol}}.$$ (A3)

Based on Eq. (A3), we compute density fields in 1-D and 2-D by binning marker weights $w_l$ as follows:

$$n(\psi, \varphi) = \sum_{n=1}^{N_{\text{vol}}} w_l \frac{S(\psi - \psi_i)}{\Delta \psi},$$ (A4)

$$n(R, z) = \sum_{n=1}^{N_{\text{vol}}} w_l \frac{S(z - z_i)}{\Delta R_i \Delta z_j}.$$ (A5)

The spatially integrated velocity distributions are

$$v(a_i, b_j) = \sum_{n=1}^{N_{\text{vol}}} w_l \frac{S(z_e - z_i)}{\Delta a_i \Delta b_j}.$$ (A6)

In Sections 8.2 and 8.3 we used 1st-order shape functions $S$ for binning (linear interpolation), and in Section 8.4 we used 0th-order (binary) binning.

Finally, we present expressions for the Jacobian factors multiplying $B_i'(B)$. Converting the right-hand side of Eq. (8) to polar GC velocity coordinates by substituting $d\mu \rightarrow B^{-1} d\Omega$.

$$d\tilde{\psi}_i = 2\tilde{E} \tilde{B},$$ (A7)

$$\tilde{E} = 2\tilde{E} \cos \theta = 2\tilde{E}(1 - \hat{\theta}^2),$$ (A8)

$$\tilde{B} = \sqrt{2\tilde{E} \sin \theta} = \sigma_B \sqrt{2\tilde{E}(1 - \hat{\theta}^2)},$$ (A9)

with $\sigma_B = v_i/|v_i|$, one readily obtains the Jacobians

$$\frac{\partial \tilde{\psi}_i}{\partial \hat{\theta}} = 2\tilde{B},$$ (A10)

for locally (in $x$) transforming velocity space elements $2\pi v_i d\psi_i d\theta_i$ to other sets $(a, b)$, such as

$$\tilde{J}_{\psi_\alpha} = \frac{\psi}{|\psi|}, \quad \tilde{J}_{\psi} = 2\pi \tilde{E} \frac{\tilde{B}}{|\tilde{B}|},\quad \tilde{J}_{\psi \psi} = 2\pi \tilde{E} \frac{\tilde{B}}{|\tilde{B}|}.$$ (A11)

$$\tilde{J}_{\psi \hat{\theta}} = \tilde{J}_{\psi \hat{\theta}}, \quad \tilde{J}_{\psi \hat{\theta}} = 2\pi \tilde{E} \frac{\tilde{B}}{|\tilde{B}|}.$$ (A12)

Appendix B: Numerical accuracy and data storage

Researchers tend to invest significant effort in the preparation of their simulation scenarios, balancing accuracy against speed. However, the data on platforms like IMAS are likely to be of variable quality, since some of it originates from automated or semi-automated workflows. The tools used to process those data should hence have a certain degree of tolerance with respect to inaccuracies and be able to detect and handle exceptions. The workflow in VisualStart also contains such sensibility checks and ways to mitigate the effects of inaccuracies. Some of the inaccuracies arising in VisualStart could be avoided with more sophisticated techniques, while others are practically inevitable due to numerical resolution being necessarily finite. This Appendix contains notes concerning some issues that are relevant for the subject of this paper or, at least, for the current implementa-
FIG. 19. Sampling of potato orbits. The two examples in (a) and (b) correspond to 2.52 MeV alpha particles with different pitch angles in the scenario of Fig. 8 that is partly based on JET. The coordinate values listed are those at the starting point of an orbit contour, which is indicated by a large yellow circle. The small black circles represent values listed are those at the starting point of an orbit contour, which is indicated by a large yellow circle. The small black circles represent 1538 samples that have been stored for various purposes, such as the distribution of samples: (i) uniform in space, (ii) uniform in time, or (iii) a 50:50 combination of (i) and (ii). We usually choose option (i) for beam-like distributions that consist only of passing particles. Option (iii) is preferred when trapped particles are present in order to ensure good resolution both near and far away from V-type stagnation points as the examples in Fig. 19 show.

The recorded time arrays must include at least $R(t)$, $Z(t)$, $v_r(t)$ and the time $t$ itself. Our databases contain many other arrays that have been stored for various purposes, such as $\xi(t)$, $\nu(t)$, $v_{\phi}(t)$, $v_{\psi}(t)$ and $\psi(t)$. The database file from our low-resolution example in Fig. 6 has a size of 33 Mbyte in NetCDF format. The databases with the parameters in Table II occupy about 150 Mbyte each in the mono-energetic cases and 500 Mbyte in the flat-energy cases. These files also contain useful auxiliary information, such as the numerical parameters, poloidal and toroidal transit times, the orbit length, bounce angle, the orbit type, and some measures of the computational accuracy: energy conservation, and spatial mismatch between start- and end-point.

Orbits that have not been completed within a certain upper bound of time steps are labeled as “incomplete”. Before proceeding to Step 4 of our workflow in Fig. 2, we deal with such incomplete orbits and those that have insufficient accuracy. Bad orbits can be corrected if necessary, or simply discarded if sufficient accuracy is difficult to achieve and unnecessary (as may happen for tiny orbits very close to an O-type stagnation point).

saves all orbits in discretized form, typically using 32–64 samples per orbit. The user can choose between three options for the distribution of samples: (i) uniform in space, (ii) uniform in time, or (iii) a 50:50 combination of (i) and (ii). We usually choose option (i) for beam-like distributions that consist only of passing particles. Option (iii) is preferred when trapped particles are present in order to ensure good resolution both near and far away from V-type stagnation points as the examples in Fig. 19 show.

The recorded time arrays must include at least $R(t)$, $Z(t)$, $v_r(t)$ and the time $t$ itself. Our databases contain many other arrays that have been stored for various purposes, such as $\xi(t)$, $\nu(t)$, $v_{\phi}(t)$, $v_{\psi}(t)$ and $\psi(t)$. The database file from our low-resolution example in Fig. 6 has a size of 33 Mbyte in NetCDF format. The databases with the parameters in Table II occupy about 150 Mbyte each in the mono-energetic cases and 500 Mbyte in the flat-energy cases. These files also contain useful auxiliary information, such as the numerical parameters, poloidal and toroidal transit times, the orbit length, bounce angle, the orbit type, and some measures of the computational accuracy: energy conservation, and spatial mismatch between start- and end-point.

Orbits that have not been completed within a certain upper bound of time steps are labeled as “incomplete”. Before proceeding to Step 4 of our workflow in Fig. 2, we deal with such incomplete orbits and those that have insufficient accuracy. Bad orbits can be corrected if necessary, or simply discarded if sufficient accuracy is difficult to achieve and unnecessary (as may happen for tiny orbits very close to an O-type stagnation point).

Data related to marker weighting are also present; in particular, the volume element sizes $\Delta V_{\text{orb}}$. There is an option to undo the factor 1/2 in the volume element given by Eq. (37) for stagnation orbits that are smaller than the cell size and, hence, cannot be double-counted. Their volume elements may then be rescaled in proportion to the orbit’s radial diameter, but the effect is usually negligible since such orbits carry little weight (and one should use a finer mesh if they are of interest).

Interpolation is necessary when converting the raw data from the particle pushing algorithm to the samples used to represent an orbit in the database. Further interpolation is necessary when loading markers on an orbit that has been retrieved from the database. As shown in Fig. 20, we found that a tiny discontinuity in an orbit contour (mismatch between start and end points) can cause relatively large “ringing” in spline interpolations. Subsequent interpolations in that region — namely, when we load marker particles — can produce corrupt samples that may even lie outside the plasma. When applying the 1-D interpolation function interp1 of Matlab for processing our orbits (and other potentially “noisy” data), we avoid such problems by using the pchip option instead of spline.

FIG. 20. Example of interpolation errors caused by a small mismatch between the start and end points of an orbit, which can be seen in the zoomed inset in panel (a). Panel (b) shows the results of various interpolation algorithms using the Matlab function interp1: linear interpolation, piecewise cubic spline (spline), spline without the last sample (“$N - 1$”), and shape-preserving piecewise cubic interpolation (pchip). The magnetic field in this example is based on a JT-60U plasma as used, for instance, in Ref. [17].

B.3. Marker loading

In order to avoid crowding at the midplane, the markers on each orbit are loaded with a random time offset $0 \leq \tau_0 < \tau_{\text{pol}}$ relative to the starting point at the midplane. When using the markers in a simulation, they are also spread (randomly) along the toroidal angle $\zeta$. 
Since kinetic energy is conserved and our 3-D and 4-D distributions both share the same energy mesh, the parameter $N_\eta$ has no impact on the numerical accuracy in our verification exercise.

B.4. Convergence test

Figure [17b] showed indications of relatively large errors that increased systematically with each iteration of our back-and-fourth transformation between a 4-D mesh in the coordinates $(E, A, R, z)$ and the orbit-based representation in 3-D CoM space. In our discussion in Section 8.4, we attributed this problem to inaccuracies in the estimation of the volume elements for orbits sampling the region near the trapped-passing boundary, where some factors exhibit singular behavior.

Further tests show that these errors can be reduced by simply increasing the number of phase space samples via the parameters $N_x, N_y$ and $N_t$ in Table II. In Fig. 21, we demonstrate this for the case where we increased the number of radial grid points $N_x$. While the flat-energy case with $E = 0...3.5$ MeV was used in Fig. 17, we have chosen only mono-energetic alpha particles with $E = 3.5$ MeV for the convergence test in Fig. 21. For an identical mesh, this results in larger values of the relative errors in Fig. 17, presumably because the signal-to-noise ratio is reduced.

The comparison between panels (a) and (b) of Fig. 21 shows a significant reduction of the numerical errors in both the local values of the alpha particle density profile $\hat{n}^\eta(r)$ and the total number of GCs $N$. As before, the errors accumulate with each iteration, which confirms their systematic character. Large errors persist only at the artificial loss boundary $(r/a \approx 1)$, which is to be expected and requires dedicated solutions that are not covered here.

Besides increasing the number of samples, there may exist other ways to correct or otherwise suppress the inaccuracies associated with singular volume elements. This is left for future work. A preliminary discussion was given in Section 6.2 of the main text.
A. Bierwage, M. Toma, and K. Shinohara. MHD and resonant
instabilities in JT-60SA during current ramp-up with off-axis N-
NB injection. *Plasma Phys. Control. Fusion*, 59(12):125008, 2017.

E. Tholerus and M. Fitzgerald. Calculating the fast ion distribution
function in constant-of-motion space for MHD instability
analysis. *Joint T7-07 Fast Ion Physics Group & M15-24 meet-
ing*, May 31, 2017.

A. Bierwage. VisualStart: GUI-Aided Unified Initialization
Tool for Hybrid (MHD + Particle) Simulations. *17th NEXT
Meeting*, Kashiwa, Japan, March 15—16, 2012.

A. Bierwage, C. Di Troia, S. Briguglio, and G. Vlad. Orbit-
based representation of equilibrium distribution functions for
low-noise initialization of kinetic simulations of toroidal plas-
as. *Comp. Phys. Comm.*, 183:1107, 2012.

A. Bierwage, K. Shinohara, N. Aiba, and Y. Todo. Role of
convective amplification of $n = 1$ energetic particle modes for
N-NB ion dynamics in JT-60U. *Nucl. Fusion*, 54(10):104001, 2014.

A. Bierwage, K. Shinohara, Y. Todo, N. Aiba, M. Ishikawa,
G. Matsunaga, M. Takechi, and M. Yagi. Self-consistent
long-time simulation of chirping and beating energetic particle
modes in JT-60U plasma. *Nucl. Fusion*, 57(1):016036, 2017.

A. Bierwage, K. Shinohara, Y. Todo, N. Aiba, M. Ishikawa,
G. Matsunaga, M. Takechi, and M. Yagi. Simulations tackle
abrupt massive migrations of energetic beam ions in a tokamak
plasma. *Nature Comms.*, 9:3282, 2018.

A. Bierwage and K. Shinohara. Orbit-based analysis of reso-
nant excitations of Alfvén waves in tokamaks. *Phys. Plasmas*,
21(11):112116, 2014.

A. Bierwage and K. Shinohara. Orbit-based analysis of non-
linear energetic ion dynamics in tokamaks. I. Effective mode
closest to resonance and resonant frequency tracking. *Phys.
Plasmas*, 23(4):042511, 2016.

A. Bierwage and K. Shinohara. Orbit-based analysis of non-
linear energetic ion dynamics in tokamaks. II. Mechanisms for
rapid chirping and convective amplification. *Phys. Plasmas*,
23(4):042512, 2016.

F. Zonca, L. Chen, S. Briguglio, G. Fogaccia, G. Vlad, and
X. Wang. Nonlinear dynamics of phase space zonal structures
and energetic particle physics in fusion plasmas. *New J. Phys.*, 17(1):013052, 2015.

L. Chen and F. Zonca. Physics of Alfvén waves and energetic
particles in burning plasmas. *Rev. Mod. Phys.*, 88:15008, 2016.

M.V. Falessì and F. Zonca. Transport theory of phase space
zonal structures. *Phys. Plasmas*, 26(2):022305, 2019.

M. V. Falessì F. Zonca, L. Chen and Z. Qiu. Nonlinear radial
envelope evolution equations and energetic particle transport in
tokamak plasmas. *J. Phys.: Conf. Ser.*, 1785:012005, 2017.
Varenna-Lausanne 2020, 12-16 October, Lausanne, Switzerland.

S. Benjamin. *Distribution transforms for orbit tomography*,
Bachelor Thesis, The Australian National University, 2021. Pa-
er in preparation.

J.A. Rome and Y-K.M. Peng. The topology of tokamak orbits.
*Nucl. Fusion*, 19(9):1193, 1979.

Ph. Lauber and Z. Lu. Analytical finite-larmor-radius and finite-
orbit-width model for the LIGKA code and its application to
KGAM and shear Alfvén physics. *J. Phys.: Conf. Ser.*,
1125:012015, 2018. Varenna-Lausanne 2018, 27-31 August,
Varenna, Italy.

A. Bierwage, M. Toma, and K. Shinohara. MHD and resonant
instabilities in JT-60SA during current ramp-up with off-axis N-
T. Kurki-Suonio, A.W. Larsen, F. Leipold, D. Moseev, S.K. Nielsen, S.D. Pinches, J. Rasmussen, M. Rebai, M. Schneider, A. Shevelev, S. Sipilä, M. Stejner, and M. Tardocchi. Alpha-particle velocity-space diagnostic in ITER. *Nucl. Fusion*, 58(9):096019, 2018.

[47] S. Sumida, K. Shinohara, R. Ikezoe, M. Ichimura, M. Sakamoto, M. Hirata, and S. Ide. Characteristics of fast $^3$He ion velocity distribution exciting ion cyclotron emission on JT-60U. *Plasma Phys. Control. Fusion*, 61(2):025014, 2019.

[48] S. Sumida, K. Shinohara, M. Ichimura, T. Bando, A. Bierwage, and S. Ide. Identification of slow-wave ion cyclotron emission on JT-60U. *Nucl. Fusion*, 61(11):116036, 2021.

[49] A. Bierwage, Y. Todo, N. Aiba, and K. Shinohara. Sensitivity study for N-NB-driven modes in JT-60U: Boundary, diffusion, gyroaverage, compressibility. *Nucl. Fusion*, 56(10):106009, 2016.

[50] ITER physics data model documentation (restricted access). [https://sharepoint.iter.org/departments/POP/IMDesign/DataModel/CI/imas-3.33.0/html_documentation.html](https://sharepoint.iter.org/departments/POP/IMDesign/DataModel/CI/imas-3.33.0/html_documentation.html). Status: Aug. 2021.

[51] R.B. White and M.S. Chance. Hamiltonian guiding center drift orbit calculation for plasmas of arbitrary cross section. *Phys. Fluids*, 27(10):2455, 1984.

[52] Y. Chen, R.B. White, G.-Y. Fu, and R. Nazikian. Numerical study of the nonlinear evolution of toroidicity-induced Alfvén eigenmodes. *Phys. Plasmas*, 6:226, 1999.

[53] R.B. White, V.N. Duarte, N.N. Gorelenkov, E.D. Fredrickson, M. Podesta, and H.L. Berk. Modeling of chirping toroidal Alfvén eigenmodes in NSTX. *Phys. Plasmas*, 26:092103, 2019.

[54] S.D. Pinches, L.C. Appel, J. Candy, S.E. Sharapov, H.L. Berk, D. Borba, B.N. Breizman, T.C. Hender, K.I. Hopcraft, G.T.A. Huysmans, and W. Kerner. The HAGIS self-consistent nonlinear wave-particle interaction model. *Comp. Phys. Comm.*, 111:133, 1998.

[55] S.D. Pinches, H.L. Berk, M.P. Gryaznevich, S.E. Sharapov, and JET-EFDA Contributors. Spectroscopic determination of the internal amplitude of frequency sweeping TAE. *Plasma Phys. Control. Fusion*, 46(7):S47, 2004.