

Computational Physics

A framework for synthetic diagnostics using energetic-particle orbits in tokamaks[☆]H. Järleblad^{a,*}, L. Stagner^b, M. Salewski^a, J. Eriksson^c, M. Nocente^d, B.S. Schmidt^a, M. Rud Larsen^a^a Department of Physics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark^b General Atomics, P.O. Box 85608, San Diego, CA 92186-5608, USA^c Department of Physics and Astronomy, Uppsala University, 751 20 Uppsala, Sweden^d Department of Physics, University of Milano-Bicocca, 20126 Milano, Italy

ARTICLE INFO

Article history:

Received 4 November 2022

Received in revised form 3 August 2023

Accepted 5 September 2023

Available online 9 September 2023

Keywords:

Nuclear fusion

Fast ions

Orbits

Weight functions

ABSTRACT

In fusion plasma physics, the large-scale trajectories of energetic particles in magnetic confinement devices are known as orbits. To effectively and efficiently be able to work with orbits, the Orbit Weight Computational Framework (OWCF) was developed. The OWCF constitutes a set of scripts, functions and applications capable of computing, visualizing and working with quantities related to fast-ion (FI) orbits in toroidally symmetric fusion devices. The current version is highly integrated with the DRESS code, which enables the OWCF to compute and analyze the orbit sensitivity for arbitrary neutron- and gamma-diagnostics. However, the framework is modular in the sense that any future codes (e.g. FIDASIM) can be easily integrated. The OWCF can also compute projected velocity spectra for FI orbits, which play a key role in many FI diagnostics. Via interactive applications, the OWCF can function both as a tool for investigative research but also for teaching. The OWCF will be used to analyze and simulate the diagnostic results of current and future fusion experiments such as ITER. The orbit weight functions computed with the OWCF can be used to reconstruct the FI distribution in terms of FI orbits from experimental measurements using tomographic inversion.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In magnetic confinement fusion devices, such as tokamaks, the population of particles can be categorized into two sets; thermal and energetic [1]. In tokamaks of size similar to the Joint European Torus (JET, major radius ~ 3 meters) [2], the following distinction can be made. If we take three times the thermal velocity as a threshold, $3v_{th}$, the energetic fast ions are ~ 10 times more energetic than the thermal (bulk) ions. For a typical plasma temperature of $1 - 10$ keV ($\approx 1 - 11 \times 10^7$ K), the lower limit for energetic particles should arguably be placed around 10 keV. In reality, the transition from thermal to fast is naturally continuous. Despite their relatively small population [3], fast ions can cause damage to first-wall components [4] and drive plasma instabilities [5,6]. Fast ions also need to be confined for a sufficient length of time (several slowing-down times, i.e. slowing down from fast

to slow speeds) so as to heat the thermal ions and enable a self-sustained plasma burn [4].

Because of their relatively low energy, the thermal ions stay mostly confined to the magnetic flux surfaces, while the fast ions have more exotic large-scale motion [7,3]. This motion is commonly referred to as a *fast-ion orbit* (FIO). Depending on the energy, starting position and angle of the particle velocity with respect to the magnetic field, the path of the FIOs can vary substantially. Therefore, it is of great importance to know how sensitive a diagnostic is to different FIOs, since various diagnosable particles or quanta (e.g. fusion-born neutrons and γ -rays) might originate more strongly from certain FIOs [3]. Depending on the heating scheme of the tokamak plasma, certain FIOs will be more populated than others [3] and the exact shape of the diagnostic signal is thus dependent on a multitude of factors, including the magnetic equilibrium (which determines the shape of the topological FIO regions in phase space) [8] and the heating scheme, as well as the density and temperature profiles of the fusion plasma [9].

By knowing the sensitivity of a plasma diagnostic in phase space, the diagnostic signal can be directly related to the fast-ion (FI) distribution, given the assumption of a linear relationship be-

[☆] The review of this paper was arranged by Prof. David W. Walker.

* Corresponding author.

E-mail address: henrikj@dtu.dk (H. Järleblad).

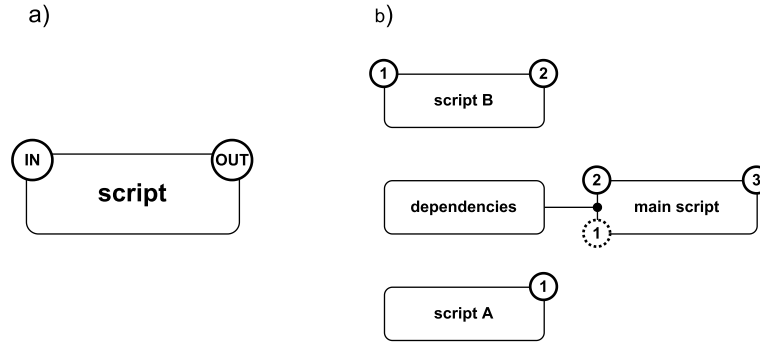


Fig. 1. a) A basic building block of the OWCF. The script has some input data visualized as an 'IN' circle in the top-left corner. The script has some output data visualized as an 'OUT' circle in the top-right corner. b) An illustration of script inter-dependency. Dotted circle means 'optional'. Black dot means 'depends on', and the black line shows which file that contains required functions. These schematics are similar to the schematics of e.g. the Unified Modeling Language [29], but with the inter-block arrow schematics replaced by number 'bubbles' for clarity.

tween the signal and the FI distribution [10]. This can be expressed as the matrix equation

$$S = W F \quad (1)$$

where S is the diagnostic signal, W is the weight matrix modeling the phase-space sensitivity for each measurement bin and F is the fast-ion distribution in phase space. As will be shown, equation (1) is one of the problems that our framework will help solve. The sensitivity of the FI diagnostic can be expressed with varying dimensionality depending on the phase space of interest. For example, in velocity-space tomography [11–15], the sensitivity is two-dimensional and can be expressed as a so-called *weight function* of the parallel and perpendicular velocity components ($v_{||}$, v_{\perp}) of the ion with respect to the magnetic field (or, equivalently, via the energy E and pitch $p = v_{||}/v$ [16]). In terms of computational resources, 2D weight functions are usually possible to compute on a regular laptop, for any phase-space grid resolutions of interest; in terms of both RAM and CPU power. For example, on an Intel(R) Core(TM) i7-8665U CPU @ 1.90 GHz laptop, single-thread computing 2D weight functions using the DRESS code [9] for the neutron emission spectroscopy diagnostic TOFOR [17] with assumed 80 measurement bins and a 400x200 (E, p) grid takes about 10 minutes and requires roughly 1 GB of RAM (including all required input data). Assume we add another dimension that we discretize into N grid points. Then the computation time would be roughly N times as long. The RAM requirement would increase due to several factors, e.g. the need to hold more synthetic diagnostic signals in memory as well as modeling the sightline of the fast-ion diagnostic in position space. This is because 2D weight functions can relate a diagnostic measurement to the FI distribution only at a single point (R, z) in position space [18,19,15,20–22], where R is the major radius cylindrical coordinate and z the vertical coordinate. To be able to relate a measurement to the FI distribution at *all* (R, z) points via FIOs, three-dimensional so-called *orbit weight functions* (OWs) can be used [8,3,19,23]. The (E, p_m, R_m) so-called orbit space is the 3D coordinate space of choice for the current version of the OWCF, due to its semi-bounded space and favorability for tomographic reconstructions [3]. E is the energy of the fast ion and p_m is the pitch at the maximum major radius position R_m of the fast ion as it traverses its orbit. All quantities computed with the OWCF can be mapped to the standard $(E, \mu, P_\phi; \sigma)$ coordinates (where μ is the magnetic moment, P_ϕ is the toroidal canonical angular momentum and σ is a binary coordinate). This provides the user with the flexibility to analyze and export computed results in either (E, p_m, R_m) or $(E, \mu, P_\phi; \sigma)$ coordinates. This is useful for e.g. stability analysis, which is often performed in the $(E, \mu, P_\phi; \sigma)$ coordinates. It should be mentioned that, in addition to OWs, other ways of relating a diagnostic signal to the full

FI distribution exist. These include expressing the FI distribution in terms of a basis of slowing-down distribution functions [24,25]. For orbit-space grid resolutions of interest, 3D OWs usually need to be computed using a computational cluster to be able to complete the calculation within a reasonable timeframe, due to both RAM and CPU requirements. Naturally, to be able to calculate, analyze and work with these 3D OWs, a speed-optimized framework written in a high-performance programming language is useful. To this end, the orbit weight computational framework (OWCF) was developed.

In short, the OWCF is a collection of scripts, utilities, structures and functions written in the Julia programming language [26]. The efficient computation of guiding-center FIOs (where the finiteness of the Larmor radius is not taken into account), made possible by the Julia language, serves as the foundation for the framework. The OWCF provides answers to several questions, such as:

- How to compute orbit weight functions efficiently,
- How to analyze orbit weight functions effectively and intuitively,
- How to streamline transformations of arbitrary fast-ion distributions into fast-ion orbits,
- How to investigate populatable fast-ion orbits for every (R, z) point,
- How to compute synthetic diagnostic signals using orbit weight functions with an exceptionally large amount of orbit-space grid points ($> 100 \times 100 \times 100$),
- How to decompose diagnostic signals into the contributions from fast-ion orbit types,
- How to streamline the possibility of reconstructing fast-ion orbit distributions from measurements.

Before being structured into a shareable framework, the tools of the OWCF have been used to develop and analyze orbit weight functions for neutron emission spectroscopy [8] and one-step gamma-ray spectroscopy diagnostics [3]. The tools have also been used to facilitate discussion of an optimal fast-ion diagnostics setup [27], and to analyze the most likely orbit-type constituents of a decrease in neutron measurements during a discharge in the JET deuterium-tritium campaign [28].

In addition, even though the OWCF is currently heavily integrated with the DRESS [9] code for computing synthetic diagnostic neutron and γ -ray spectra, the OWCF maintains a modular approach towards synthetic diagnostics codes in general. This enables future codes to be easily integrated with the OWCF, for computing OWs via synthetic diagnostic FIO spectra.

This paper is structured as follows. In section 2, an overview of the OWCF is given. In sections 3 and 4, discussions on topological maps and orbit weight functions are provided, respectively. In

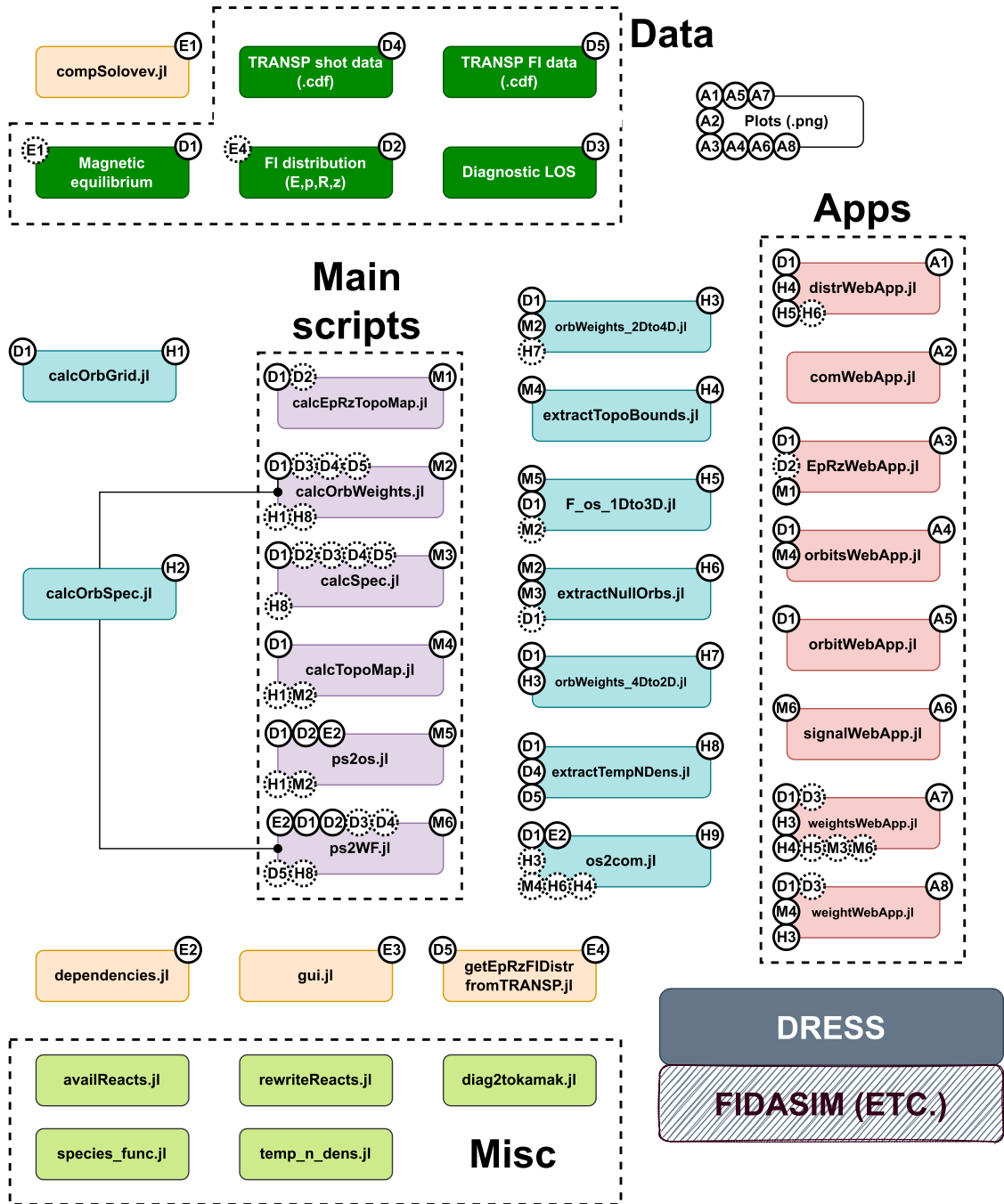


Fig. 2. A graphical overview of the OWCF. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

5, the capability of the OWCF to transform between (E, p_m, R_m) and (E, p, R, z) is outlined and in section 6 the tools of the OWCF for decomposing diagnostic signals in terms of orbit types are presented. Finally, a conclusion follows in section 7.

It should be mentioned that the OWCF contains more utilities and tools than can be included in this paper. For a complete and detailed breakdown of the whole OWCF, the reader is referred to the manual included with the OWCF.

2. Overview of the OWCF

The OWCF comprises many scripts and file structures, and it is written almost entirely in the Julia programming language. It was chosen as a suitable language for the OWCF since it is high-level,

yet maintains high-performance via efficient code compilation and execution [26].

To help the reader understand the structure of the OWCF, a basic building block of the framework has been visualized in Fig. 1a, and a basic relationship between different scripts has been visualized in Fig. 1b. We can observe how the inputs and outputs form an I/O chain between the scripts, including optional inputs and integrated usage of dependencies. Together, this simple example serves as a basic illustration of the much larger system that is the full OWCF.

A graphical overview of the full OWCF is given in Fig. 2. There are four main groups of code (inside dotted lines): main scripts, apps, data and misc. The **main scripts** (colored purple) constitute the bulk of the computational tools of the OWCF. The apps pro-

vide interactive analysis of computed quantities using web-based plotting. We shall explore the user interface and capabilities of the OWCF main scripts and apps in later sections. Furthermore, as we can observe in Fig. 2, there are five forms of basic input data for the OWCF: a magnetic equilibrium, a fast-ion (FI) distribution in (E, p, R, z) coordinates, a diagnostic line-of-sight (LOS), a .cdf-file with TRANSP [30] discharge (shot) data and a .cdf-file with TRANSP NUBEAM [31] FI data. TRANSP is a 1.5D equilibrium and transport solver for interpretation and prediction of tokamak discharges. Its simulations and results are widely used in the plasma physics community, thus making its output a natural basic data input for the OWCF. NUBEAM is the FI module of TRANSP.

Because of its versatile set of tools, the OWCF can be used in a number of ways depending on the needs of the user. One integral part of the framework that is likely to be used by almost all users is the easy and efficient computation of topological maps.

3. Topological maps

Topological maps enable the user to easily interpret and analyze computed quantities in 3D (E, p_m, R_m) orbit space [7,32,23,19,8] and 4D (E, p, R, z) guiding-center phase space. For each (E, p_m, R_m) or (E, p, R, z) point, the corresponding guiding-center drift orbit is computed by solving the equations-of-motion [33]

$$\dot{\mathbf{X}} = \frac{p_{\parallel}}{\gamma m} \frac{\mathbf{B}^*}{B_{\parallel}^*} + \epsilon \mathbf{E}^* \times \frac{\hat{\mathbf{c}}\mathbf{b}}{B_{\parallel}^*} \quad (2)$$

$$\dot{p}_{\parallel} = q \mathbf{E}^* \cdot \frac{\mathbf{B}^*}{B_{\parallel}^*} \quad (3)$$

where $\dot{\mathbf{X}}$ is the time derivative of the guiding-center particle position, p_{\parallel} is the particle momentum parallel to the magnetic field, γ is the Lorentz factor, m is the particle mass, ϵ is an ordering parameter, c is the speed of light, $\hat{\mathbf{c}}\mathbf{b}$ is the unit vector in the direction of the magnetic field and q is the particle charge. \mathbf{B}^* and \mathbf{E}^* are the effective magnetic and electric fields, respectively, defined as

$$\mathbf{B}^* \equiv \mathbf{B} + \epsilon \frac{cp_{\parallel}}{q} \nabla \times \hat{\mathbf{b}} \quad (4)$$

$$\mathbf{E}^* \equiv \mathbf{E} - \frac{\epsilon}{q} \left(p_{\parallel} \frac{\partial \hat{\mathbf{b}}}{\partial t} + mc^2 \nabla \gamma \right) \quad (5)$$

where \mathbf{B} is the magnetic field, \mathbf{E} is the electric field and $\frac{\partial \hat{\mathbf{b}}}{\partial t}$ is the time derivative of the unit vector in the direction of the magnetic field. The equations-of-motion are relativistic to enable users of the OWCF to work with particles with speeds close to c , e.g. runaway electrons [34,35]. Once the equations-of-motion have been integrated once poloidally, we can identify the FIO type for our phase-space point of interest. By identifying which parts of phase space correspond to the six basic types of FIOs (co-passing, counter-passing, trapped, stagnation, counter-stagnation and potato) [8,3,36], we can identify topological regions and use them to gain further insight about the phase-space quantities. Topological maps also serve as a ‘discrete prism’ when splitting synthetic diagnostic signals into their orbit-type constituents, as we shall discuss in later sections.

3.1. (E, p_m, R_m) orbit space

The computation of topological maps in orbit space is done by integrating equations (2), (3) for all of (E, p_m, R_m) orbit space until the particle has completed its orbit poloidally and then inferring the orbit type. The vertical starting point of the integration, z_m is found by finding the minimum poloidal flux (or maximum, if the

poloidal flux is decreasing towards the edge, depending on definition) for a given $R = R_m$. That is

$$z_m = \max_z(\psi_p(R_m, z)) \quad \forall z \in [z_{axis} - dz, z_{axis} + dz] \quad (6)$$

where $\psi_p(R, z)$ is the poloidal flux as a function of R and z , z_{axis} is the vertical coordinate value of the magnetic axis and dz is an arbitrary length within which the maximum ψ_p value is to be found with certainty. dz naturally varies depending on machine.

In addition to the six basic orbit types mentioned in the previous section, the OWCF also defines ‘lost’, ‘incomplete’ and ‘invalid’ orbits. Lost orbits are particle trajectories that, when obtained via integration of equations (2), (3), intersect the wall of the machine. Incomplete orbits are orbits that the integration algorithm cannot complete (for any reason e.g. numerical errors). Invalid orbits are (E, p_m, R_m) coordinates that do not correspond to physically possible orbits. When examining the orbit-space topology, it makes most sense to look at slices of constant energy E , and not p_m or R_m . This is because only slices of constant E contain *all* orbit types [8,3]. An overview of how the OWCF computes topological maps in (E, p_m, R_m) orbit space is shown in Fig. 3.

In addition to topological maps, the OWCF also allows the user to compute maps of the poloidal and toroidal transit time in (E, p_m, R_m) orbit space. This can be useful for e.g. examining magnetic equilibria in detail or resonances between particles and Alfvén eigenmodes [6,36].

The **orbitsWebApp.jl** (A4 in Fig. 2) can be used to interactively visualize the (E, p_m, R_m) topological maps, poloidal and toroidal transit times. A screenshot of the web application can be seen in Fig. 4. We can see how the web application also visualizes the current p_m , as well as a top-view and a poloidal projection of the orbit trajectory for the current (E, p_m, R_m) coordinate. In the plot of the toroidal transit time τ_t (middle-left plot), it is interesting to note the narrow curved region of relatively large values of τ_t . This corresponds to trapped orbits that, due to the specific magnetic equilibrium, precess toroidally approximately as much in the co-current as in the counter-current direction. With the **orbitsWebApp.jl** web application, the user can change the values of the E , p_m and R_m sliders interactively and analyze (E, p_m, R_m) orbit space in real-time. $(E, \mu, P_{\phi}; \sigma)$ constants-of-motion space (where μ is the magnetic moment, P_{ϕ} is the toroidal canonical angular momentum and σ is a binary coordinate) can be examined via a toggle button. This switch has been illustrated in Fig. 5.

3.2. (E, p, R, z) phase space

The OWCF can also compute maps for the orbit topology, poloidal and toroidal transit time in (E, p, R, z) phase space, via the **calcEpRzTopoMap.jl** main script (M1 in Fig. 2). Topological maps for (E, p, R, z) phase space are useful when the user would like to know which orbit types pass through a certain (R, z) point, what the (E, p) topological map looks like and how that changes from one (R, z) point to another.

Data computed in (E, p, R, z) phase space can be visualized using the **EpRzWebApp.jl** (A3) web application, as illustrated in Fig. 6. We can observe how, at the (R, z) point of interest, the valid orbits consist mostly ($> 80\%$) of co- and counter-passing orbits. At pitch values close to zero, the valid orbits are exclusively trapped orbits for all energies.

The **EpRzWebApp.jl** web application is envisioned to assist fast-ion diagnostics that observe small (compared with minor radius) measurement volumes, e.g. fast-ion D- α diagnostics, in determining which orbit types are able to produce signals at given (R, z) points. Several other usage areas can also be imagined, such as for teaching and a deeper insight into the populated orbits for a fast-ion distribution given in (E, p, R, z) coordinates. For example, it

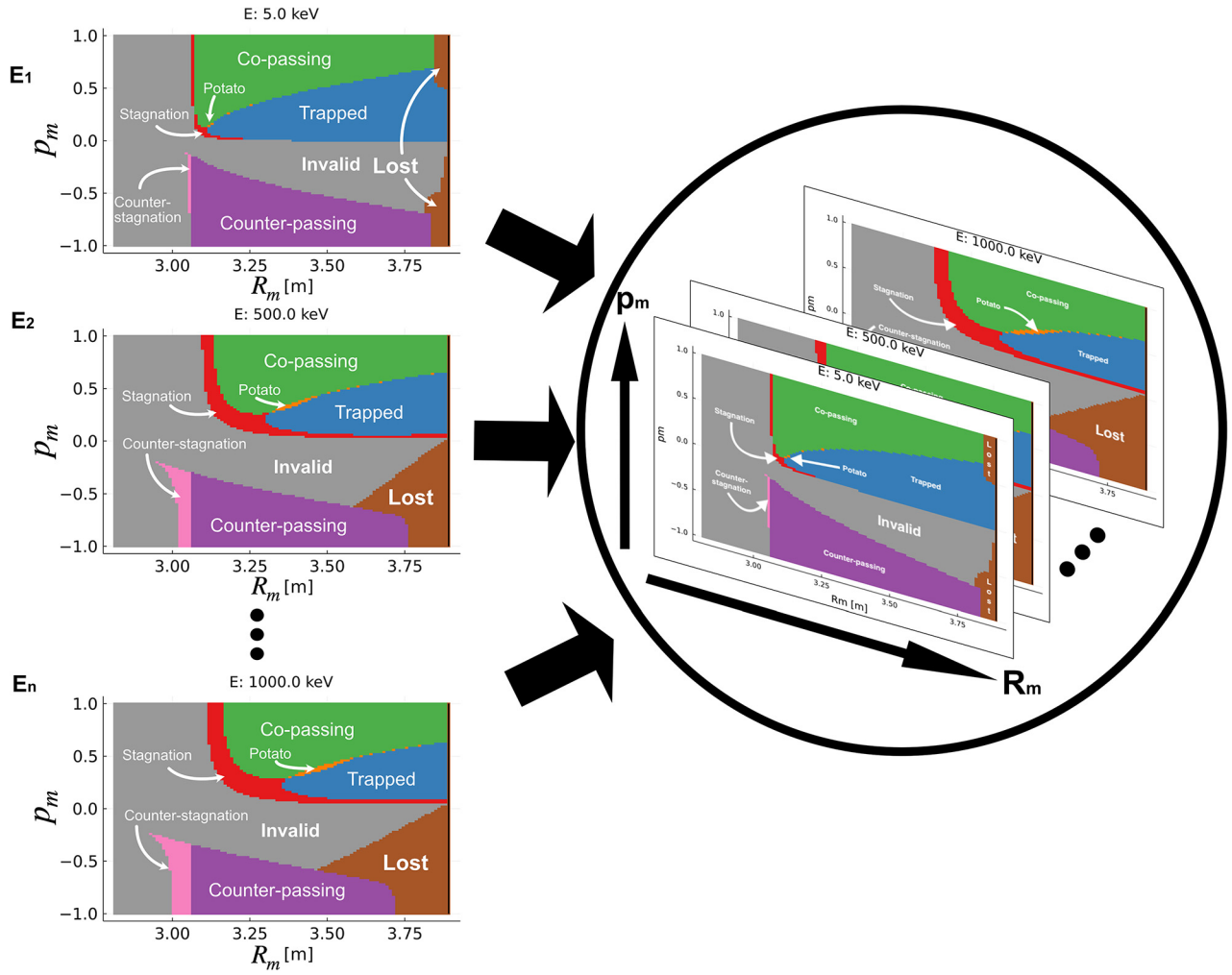


Fig. 3. A graphical overview of the process by which the OWCF computes topological maps for the (E, p_m, R_m) coordinate space. All the energy slices (E_1, E_2, \dots, E_n) are computed sequentially and then put together into a 3D topological map of the whole (E, p_m, R_m) coordinate space.

can be imagined how students can use the OWCF apps to perform numerical experiments by modifying the orbit parameters interactively. This could provide a quick and effective way of teaching FIOs and orbit-space topologies, and help the students faster develop an intuition for FIOs and FIO-related quantities.

4. Orbit weight functions

Orbit weight functions (OWs) quantify how sensitive a diagnostic is to different fast-ion drift orbits. OWs are arguably one of the main features of the OWCF. Hence, the discussion concerning OWs will be given extra care, and it has therefore been split into two subsections: computation and visualization.

4.1. Computation

The computation of OWs is done via the **calcOrbWeights.jl** main script (M2 in Fig. 2). As has been discussed in [8,3,23], the process of computing orbit weight functions in (E, p_m, R_m) orbit space can be mathematically expressed as

$$w(E_{1,d}, E_{2,d}) = \frac{1}{4\pi^2 \tau_p} \int_0^{2\pi} \int_0^{2\pi} \int_0^{\tau_p} S(E_{1,d}, E_{2,d}, E, p_m, R_m, t, \gamma, \phi_0) dt d\gamma d\phi_0 \quad (7)$$

where $w(E_{1,d}, E_{2,d})$ is the orbit weight function for the diagnostic measurement bin with lower and upper limits $E_{1,d}$ and $E_{2,d}$ respectively. τ_p is the poloidal transit time of the orbit, S is the expected measurement signal of the orbit, t is time, γ is the gyro-angle and ϕ_0 is the initial toroidal angle of the orbit. The orbit weight function is the expected signal of an orbit averaged over all reducible coordinates [23,19].

The **calcOrbWeights.jl** main script computes orbit weight functions using equation (7) while also taking the geometry of the diagnostic line-of-sight (LOS) into account. When computing OWs for e.g. neutron emission spectroscopy [37] or gamma-ray spectroscopy [38,39] by coupling the OWCF to the DRESS code, there are several fusion reactions for the user to choose between. The following fusion reactions are currently supported by the OWCF (via DRESS) [40,41]:

- $D + D \rightarrow n (2.45 \text{ MeV}) + {}^3\text{He} (0.82 \text{ MeV})$
- $D + T \rightarrow n (14.1 \text{ MeV}) + {}^4\text{He} (3.5 \text{ MeV})$
- $D + {}^3\text{He} \rightarrow p (14.7 \text{ MeV}) + {}^4\text{He} (3.6 \text{ MeV})$
- $T + p \rightarrow \gamma (19.8 \text{ MeV}) + {}^4\text{He}$

Please note however, that any other fusion reaction can be added to the list, by simply implementing the required cross-sectional data. Also, please note that the fusion-product proton from the $D({}^3\text{He}, p){}^4\text{He}$ reaction has a non-zero charge. The pro-

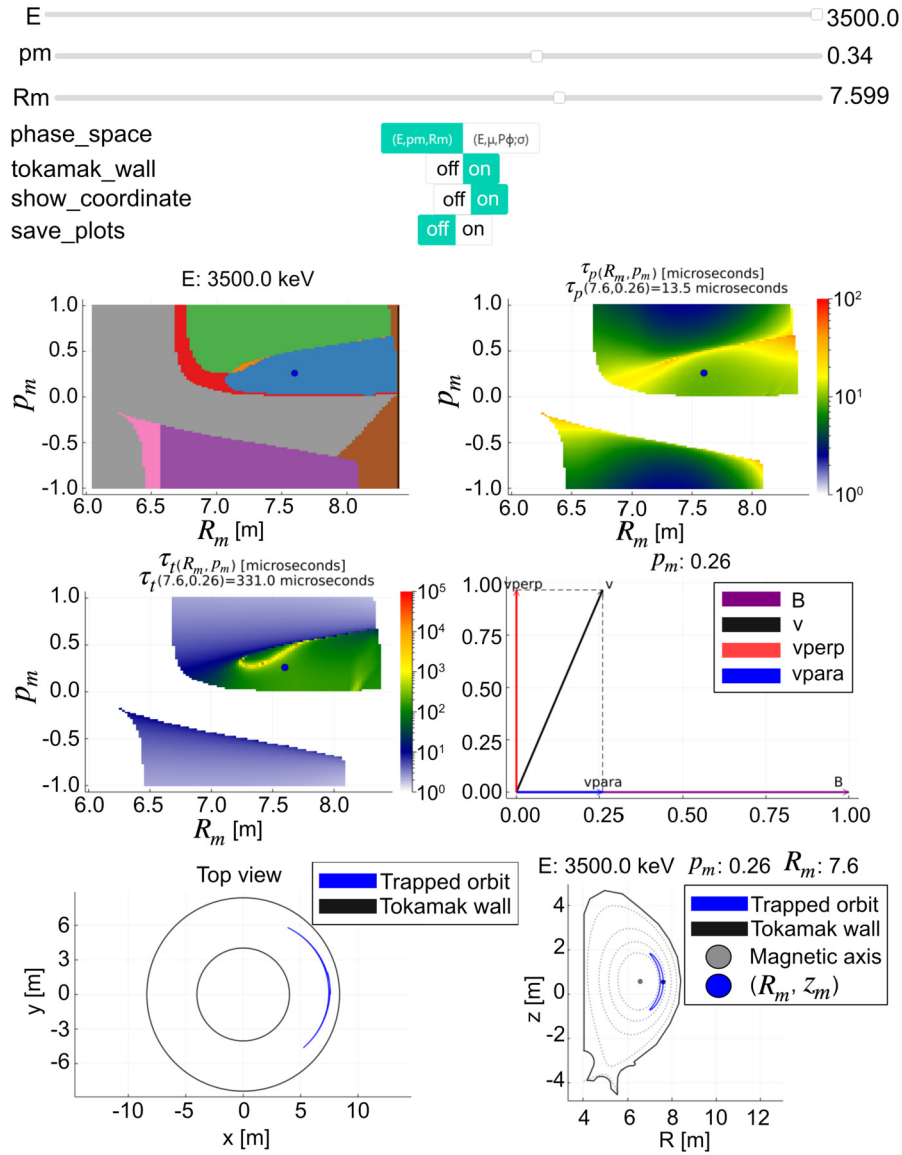


Fig. 4. A screenshot of the orbitsWebApp.jl web application for interactively visualizing FIOs, topological maps, poloidal and toroidal transit time maps in (E, p_m, R_m) orbit space and $(E, \mu, P_\phi; \sigma)$ constants-of-motion space. ' $\tau_p(7.6, 0.26) = 13.5$ microseconds' means that τ_p at $(R_m, p_m) = (7.6 \text{ m}, 0.26)$ is 13.5 microseconds.

ton trajectory would therefore have to be followed post-fusion to enable diagnostic investigation. As of the current version of the OWCF, this is not implemented. Instead, the resulting proton energy distribution for the plasma as a whole is returned (assuming 4π emission, i.e. ignoring any diagnostic LOS input). In future versions of the OWCF, implementation could utilize e.g. the methods developed in [42].

When computing OWs numerically, the temperature and density profiles of the thermal (slow) particle species in the fusion reaction must be specified. With the OWCF, they are specified as functions of ρ_{pol} ,

$$\rho_{pol} = \left(\frac{\psi_p - \psi_{p,axis}}{\psi_{p,sep} - \psi_{p,axis}} \right)^{1/2}, \quad (8)$$

where ψ_p is the poloidal magnetic flux function, $\psi_{p,axis}$ is the value of ψ_p at the magnetic axis and $\psi_{p,sep}$ is the value of ψ_p at the plasma separatrix. If no profiles are specified, the OWCF will use default (re-scalable) profiles. These are illustrated in Fig. 7.

Continuing, calcOrbWeights.jl also gives the user the option of computing weights binned into projected velocities u instead of

e.g. neutron or gamma-ray energies. The projected velocity of a fast-ion can be written as [11]

$$u = v_{||} \cos \phi + v_{\perp} \sin \phi \cos \Gamma, \quad (9)$$

where $v_{||}$ and v_{\perp} are the fast-ion velocity components parallel and perpendicular to the magnetic field, respectively. ϕ is the observation angle between the line-of-sight and the magnetic field vector and Γ is the gyrophase [43]. The projected velocity is a good proxy for spectral fast-ion diagnostics since it reflects essential features of the spectrum formation, due to the conservation of energy and momentum in a fusion reaction [11]. u is an analytic measure of the sensitivity of a diagnostic in the sense that u does not require any thermal density and temperature data. When u has been computed, a value of w is added to its corresponding velocity bin. The binning weights w are

$$w = \frac{\Delta \tau_p}{\tau_p} \frac{\Omega \Delta \phi}{2\pi}, \quad (10)$$

where $\Delta \tau_p / \tau_p$ is the fraction of the total poloidal transit time for the FIO spent at the point of interest, Ω is the solid angle of the

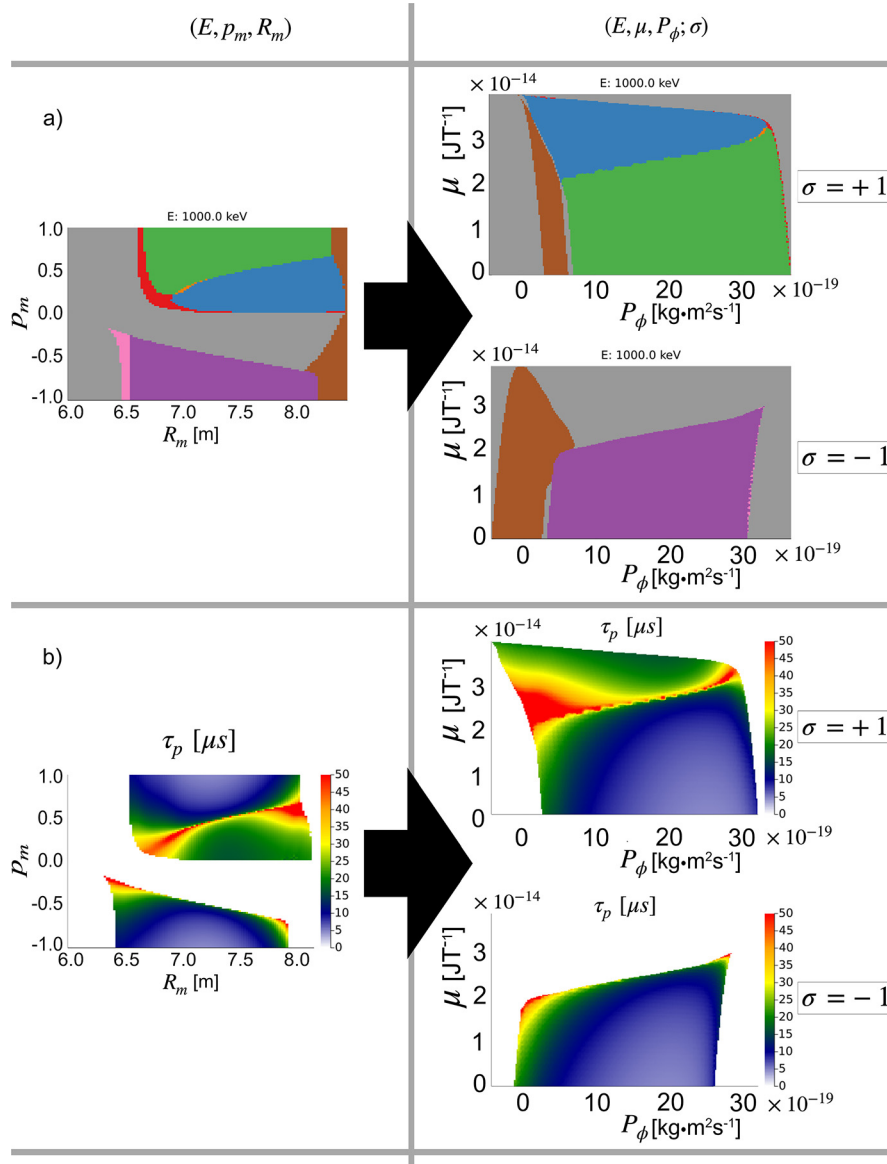


Fig. 5. An example of how the OWCF can be used to map a) topological regions and b) poloidal transit times (τ_p) from (E, p_m, R_m) orbit coordinates to $(E, \mu, P_\phi; \sigma)$ constants-of-motion coordinates. The slices in b) correspond to the same energy slices as in a). We can observe how the apparent size of the topological regions change between the phase spaces. $\sigma = -1$ corresponds to counter-going (counter-current) orbits and $\sigma = +1$ corresponds to co-going (co-current) orbits. The lost region (brown) was not included for the poloidal transit times.

diagnostic viewing cone voxel and $\Delta\phi/2\pi$ is the toroidal angle fraction that the diagnostic viewing cone voxel of interest occupies. As mentioned earlier, the finiteness of the Larmor radius is not taken into account by the OWCF. The process of creating the resulting projected velocity spectrum has been illustrated in Fig. 8. Let $u_{i,j}$ denote the projected velocity of sample j from gyro-center i . In Fig. 8: **A**) Two samples (yellow), with projected velocities $u_{1,1}$ and $u_{1,2}$, along the gyro-orbit of the same guiding-center (red) end up in the same u -bin, even though $u_{1,1} \neq u_{1,2}$. This is because of the finite discretization of u , and $u_{1,1} \approx u_{1,2}$ so the samples end up in the same bin. **B**) A sample with $u = v_{\parallel,1} \cos \phi$. Because of $\cos \Gamma$ in eq. (9), not many samples end up here. **C**) Samples from both gyro-centers can end up in this bin. From the first point because of up-shift from $v_{\parallel,1} \cos \phi$ and from the second point because of down-shift from $v_{\parallel,2} \cos \phi$, where $v_{\parallel,1} < v_{\parallel,2}$. **D**) Again, because of $\cos \Gamma$ a lot of samples will end up here, from both gyro-centers. **E**) A sample with $v_{\parallel,2} \cos \phi$. **F**) Even though this sample is outside of the LOS, the OWCF still includes it in the binning, since its guiding-center is inside of the LOS.

4.2. Visualization

The orbit weight functions can be visualized with the **weightsWebApp.jl** web application (A7 in Fig. 2). With the OWCF, the orbit weight functions can be visualized both as a (E_d, E, p_m, R_m) 4D matrix or as an equivalent $(E_d, E, \mu, P_\phi; \sigma)$ 5D matrix. In Fig. 9, we can see a screenshot of the **weightsWebApp.jl** (A7) web application. We can observe how the S and WF signals match, verifying that the orbit weight functions are an accurate model of the diagnostic [11]. We can also observe how the product of the weight function (top-right plot) and the fast-ion distribution (middle-left) result in a detailed view of where the diagnostic signal is likely to have originated in orbit space (middle-right).

Another way of thinking about OWs is: given an orbit with the coordinate (E, p_m, R_m) , what is the expected signal? What signal is that particular orbit likely to produce? To answer these questions, the user can employ the **weightWebApp.jl** web application (A8 in Fig. 2, also please note the singular form *weight* in the name). The user can scan the valid orbits of orbit space and examine what

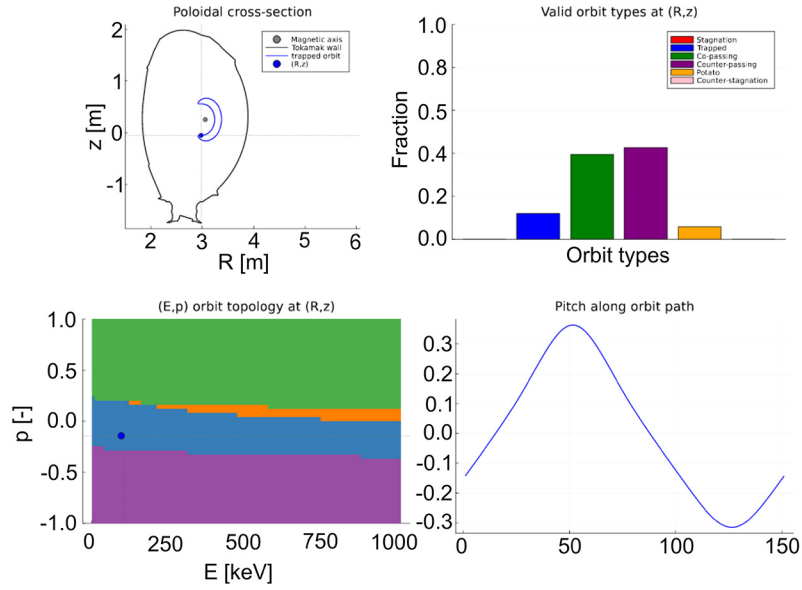


Fig. 6. A screenshot of the EpRzWebApp.jl web application for interactively visualizing FIOs and topological maps in (E, p, R, z) orbit space. The app controls have been omitted for brevity. The top right bar plot shows the populatable orbits at (R, z) as fractions. The bottom right plot shows the evolution of pitch as the fast ion traverses the FIO. JET discharge 96100 at 13 s.

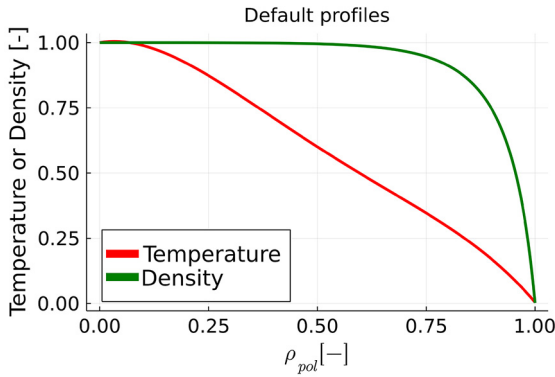


Fig. 7. The default bulk temperature and density profiles used by the OWCF, in case the user provided none. ρ_{pol} is the normalized poloidal flux coordinate as defined in (8). The profiles are based on the temperature and density profiles of JET shot No 96100 at 13 seconds.

their expected signals look like. In Fig. 10, we can observe how the user can examine the expected signals for various orbits and orbit types by changing the (E, p_m, R_m) coordinate with the sliders. In Fig. 10, the trapped orbit in focus has both its ‘banana tips’ inside the vertical sightline of the diagnostic, resulting in a double-humped expected signal, plotted in the top-left plot.

5. Transforming to orbit space

A guiding-center fast-ion distribution f can be equivalently parameterized in several different sets of coordinates. The fast-ion data from TRANSP (D5 in Fig. 2) is given on an irregular grid spiraling outwards from the magnetic axis, as shown in Fig. 11. To ultimately transform the fast-ion distribution to (E, p_m, R_m) orbit space, the OWCF first transforms the data onto a rectangular grid in (E, p, R, z) coordinates by creating a Delaunay tessellation for the spiral (R, z) grid [44,19] and then utilize linear barycentric interpolation [45]. The barycentric weights used by the OWCF has been chosen as suggested by [46]. That is,

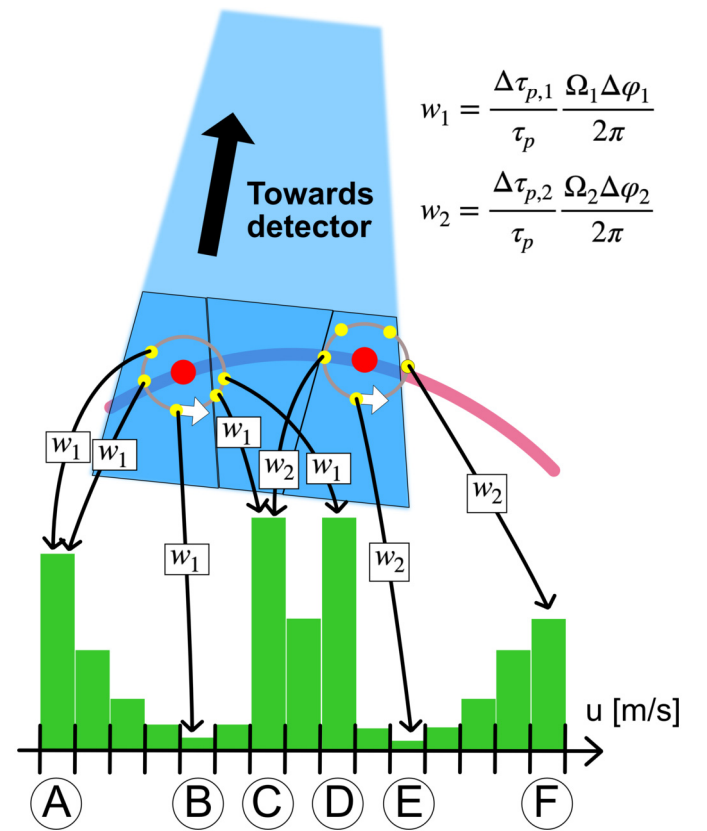


Fig. 8. An illustration of the computation of projected velocity u spectrum by the OWCF. The orbit trajectory is depicted as a pink line.

$$\mu(R, z, R_j, z_j) = \frac{\lambda(R, z, R_j, z_j)}{\sum_{j=1}^3 \lambda(R, z, R_j, z_j)} \quad (11)$$

with

$$\lambda(R, z, R_j, z_j) = \frac{1}{(R - R_j)^2 + (z - z_j)^2}, \quad (12)$$

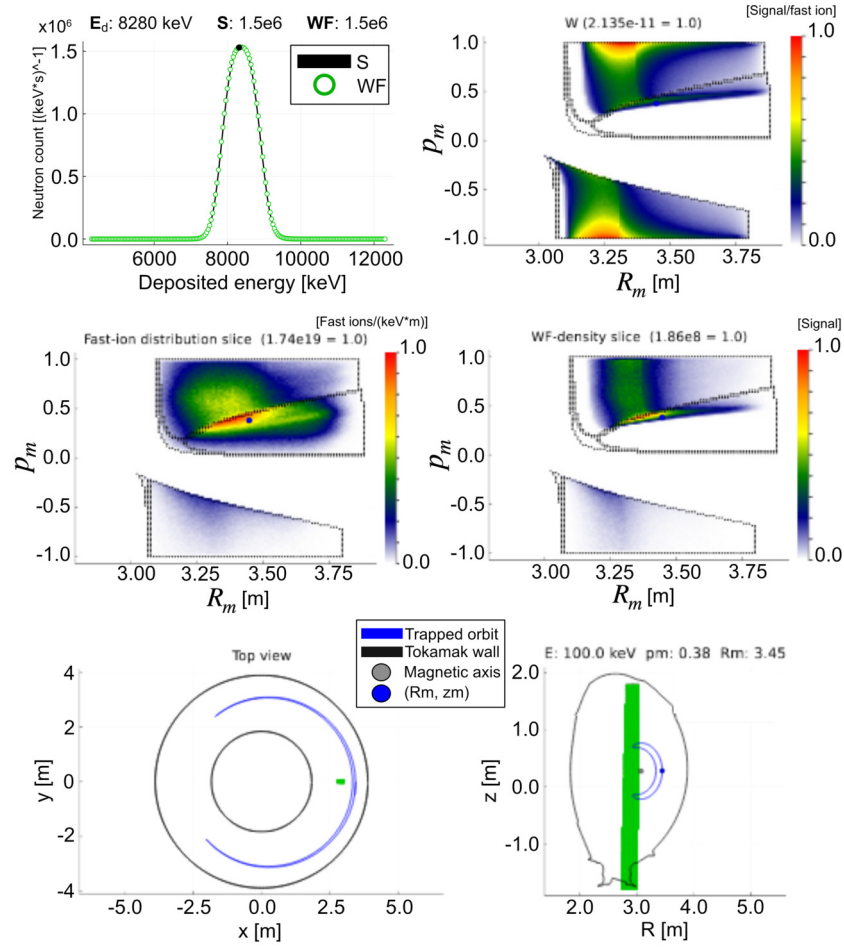


Fig. 9. A screenshot of the `weightsWebApp.jl` web application for interactively visualizing orbit weight functions. The app controls have been omitted for brevity.

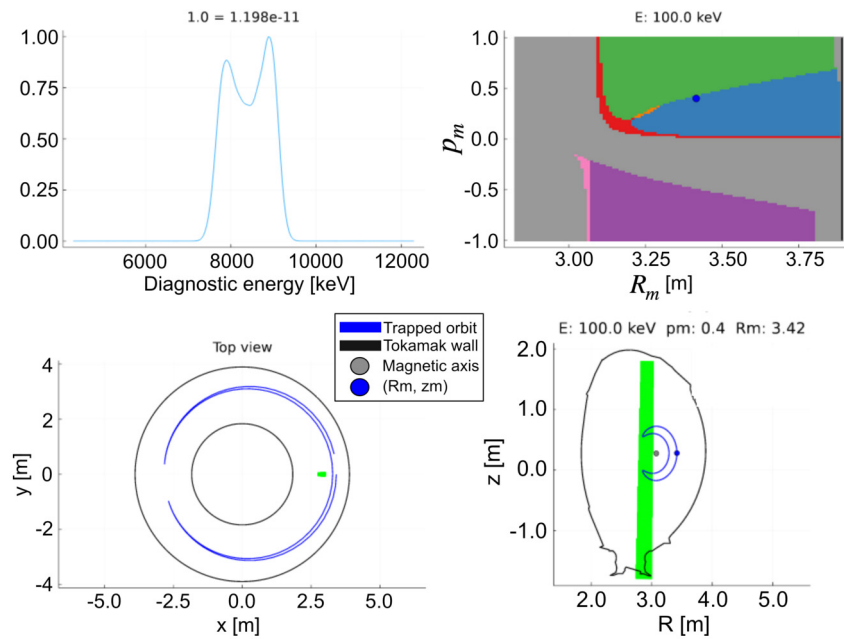


Fig. 10. A screenshot of the `weightWebApp.jl` web application for interactively visualizing orbit weight functions. The top left-hand plot shows the expected signal for the orbit visualized in the bottom plots. The app controls have been omitted for brevity.

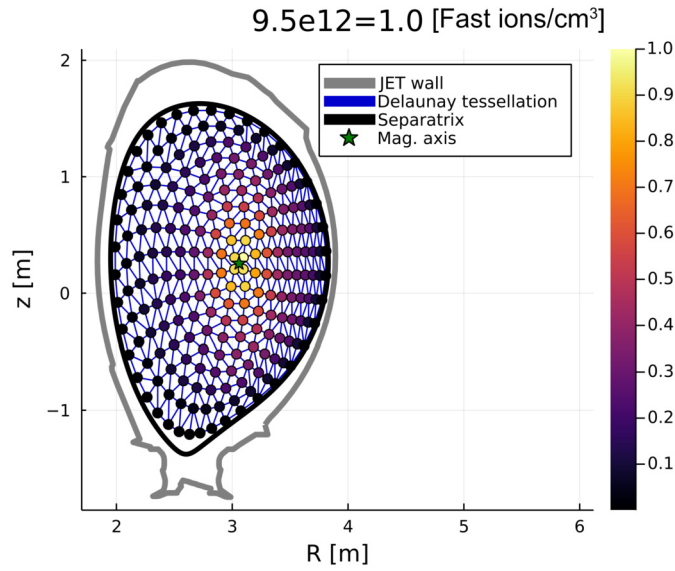


Fig. 11. An example of a Delaunay tessellation performed on TRANSP fast-ion distribution data known only at discrete, irregular (R, z) points. The energy E and pitch p dependency of the distribution was integrated out to enable illustrative plotting, i.e. $f(R_i, z_i) = \int f(E, p, R_i, z_i) dE dp$. $f(R_i, z_i)$ was then normalized so that $\max(f(R_i, z_i)) = 1.0$ to enable clean plotting. The true maximum value of $f(R_i, z_i)$ is shown in the plot title as $\max(f(R_i, z_i)) = 9.5 \times 10^{12}$.

where (R_j, z_j) are the individual coordinates of the three vertices of the Delaunay triangle containing the query point (R, z) . Query points outside of the Delaunay tessellation will be treated with nearest neighbor extrapolation, followed by forcing all $f(E, p)$ outside of the separatrix to be identically zero.

With $f(E, p, R, z)$ obtained from the TRANSP data, the fast-ion distribution in (E, p_m, R_m) orbit space can be obtained via Monte-Carlo sampling. This is done as follows. The 4D (E, p, R, z) array is ‘spaghettified’ into a 1D vector. The cumulative sum vector is then computed and an evenly distributed random number between 0 and the last cumulative sum vector element is drawn. The corresponding element in the vector is identified, as well as the pertaining (E, p, R, z) coordinate. The FIO is then computed and the (E, p_m, R_m) coordinate of the FIO is identified. This is done for a sufficient number of Monte-Carlo samples and the $f(E, p_m, R_m)$ distribution is obtained. The $f(E, p, R, z) \rightarrow f(E, p_m, R_m)$ process has been visualized in Fig. 12. Other methods of transforming $f(E, p, R, z) \rightarrow f(E, p_m, R_m)$ might be integrated in the OWCF in updated versions of the framework. This is discussed in the future work section of this paper.

6. Orbit constituents of diagnostic signals

With the OWCF, the user can also decompose synthetic diagnostic signals into orbit type constituents, to investigate what fractions of orbit types are most likely to have produced a certain diagnostic measurement. Mathematically, this can be expressed as [3]

$$WF = \sum_h W_h F_h, \quad (13)$$

where ‘ h ’ stands for the different orbit types (co-passing, trapped etc). To investigate the dependency on (E, p_m, R_m) in terms of orbit types, we can further expand (13) as

$$\sum_h W_h F_h = \sum_h \sum_E W_{h,E} F_{h,E} \quad (14)$$

$$= \sum_h \sum_{p_m} W_{h,p_m} F_{h,p_m} \quad (15)$$

$$= \sum_h \sum_{R_m} W_{h,R_m} F_{h,R_m} \quad (16)$$

where e.g. $F_{h,E}$ is the fast-ion distribution for a particular orbit type h at a certain fast-ion energy E .

To compute these kinds of signal splits, the user can employ the **ps2WF.jl** (M6 in Fig. 2) main script. In addition to being able to compute orbit split signals, the **ps2WF.jl** tool can also compute WF signals for ultra high-resolution grids in orbit space. That is, grids that consist of tens of millions of valid orbits and that require a computational cluster to compute within a reasonable timeframe (minutes instead of hours, assuming one orbit takes ~ 1 ms to compute). These kinds of grids can be useful when the $S = WF$ identity needs to be validated. S is the synthetic signal from e.g. a regular Monte-Carlo code and the validity of the OWs depends on the perfect match between the WF vector and the S vector. For too coarse grids, the identity $S = WF$ will likely not hold. This is because accurate representation of the orbit-space sensitivity (i.e. W) of the diagnostic requires a discretization of (E, p_m, R_m) orbit space with a sufficiently high resolution.

The resulting output of the **ps2WF.jl** script can be visualized using the **signalWebApp.jl** (A6 in Fig. 2) web application. This has been illustrated in Fig. 13. We can observe how the OWCF predicts that about 60% of the measurement signal in the 2500 keV measurement bin (indicated by a red dot in the top row left-hand plot) is likely to have originated from co-passing orbits (top row right-hand plot) and about 20% from stagnation orbits. The OWCF also shows that stagnation orbits are more likely to have contributed to the measurement signal than trapped orbits, for fast-ion energies above approximately 250 keV (second row plot). This is because the fast-ion distribution is likely to consist of more stagnation orbits than trapped orbits above that energy (third row plot), and because this particular diagnostic sightline (oblique sightline in JET) is more sensitive to stagnation orbits than trapped orbits above the same energy level, approximately (bottom row plot).

Furthermore, if a simple overview of the orbit-type constituents of a diagnostic signal is sought, the OWCF provides quick and easy-to-use tools in **gui.jl** (E3 in Fig. 2) that can decompose a synthetic diagnostic signal into orbit-type origin, as shown in Fig. 14. We can observe how the diagnostic signal is likely to have originated mostly from co-passing orbits and about a third from trapped orbits. To know the likely orbit-type origin of diagnostic signals can be useful when e.g. verifying certain heating schemes, such as the three-ion heating scheme [47]. It can also be useful when investigating the likely orbit types of the birth distribution of alpha particles in future burning plasmas.

7. Conclusion

The OWCF is a framework of scripts, functions, routines and apps that combine to enable computation, visualization and analysis of fast-ion orbits and quantities in (E, p_m, R_m) (orbit space) $(E, \mu, P_\phi; \sigma)$ (constants-of-motion) and (E, p, R, z) (guiding-center) coordinates, such as orbit weight functions and fast-ion distribution functions. It is written in the Julia programming language to enable efficient computation of fast-ion orbits, pushing the computation time down to ~ 1 ms (deeply passing ~ 1 ms, marginally trapped ~ 3 ms). With the OWCF, the user can compute topological maps of the different orbit types and view them in any of the supported coordinate spaces. The topological boundaries can then be superimposed onto e.g. orbit weight functions to analyze the sensitivity of fast-ion diagnostics to different orbit types. The orbits and orbit-space quantities can be visualized

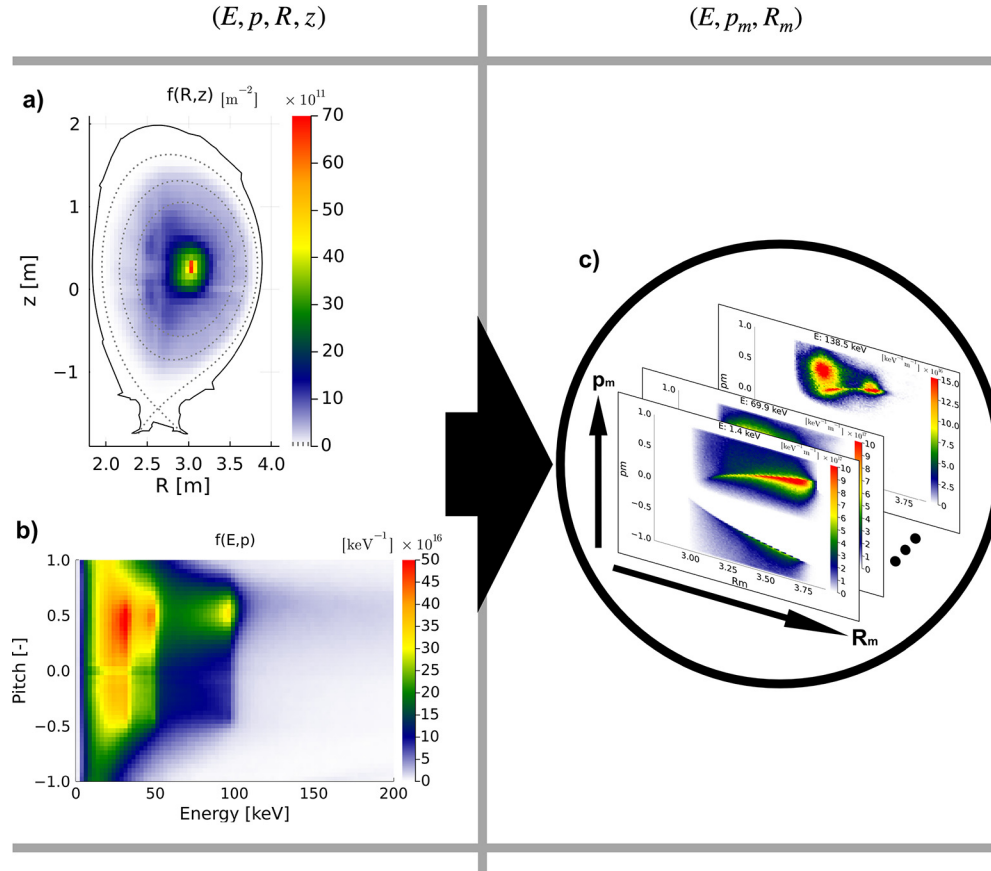


Fig. 12. An example of $f(E, p, R, z) \rightarrow f(E, p_m, R_m)$. In a) the $f(R, z)$ distribution has been illustrated, where the E - and p -dependences have been integrated out. In b) the opposite is shown, i.e. $f(E, p)$. In c) the resulting fast-ion distribution in orbit-space $f(E, p_m, R_m)$ has been split into slices of constant E .

interactively via the OWCF web applications, where fast-ion orbits are computed in real-time based on the user's controls, made possible by highly optimized guiding-center codes [44,48].

The OWCF has been designed to be modular in terms of synthetic diagnostics codes. In addition to the already integrated DRESS code, future codes such as FIDASIM can be added to the OWCF framework by simply altering a few lines of code in the OWCF.

Furthermore, the OWCF also contains tools for splitting synthetic diagnostic signals into their likely orbit-type constituents. This can be analyzed at varying levels of detail, depending on the needs of the user.

Tools that allow for transforming quantities between (E, p_m, R_m) , $(E, \mu, P_\phi; \sigma)$ and (E, p, R, z) are incorporated into the OWCF. This enables the user to compute orbit-related quantities in one coordinate space and analyze them in any of the other supported spaces.

Future work for the OWCF includes continuing to make the scripts and apps more user-friendly by improving their efficiency and speed, making the OWCF compatible with additional forward models and computational cluster architectures, as well as adding further transformation capabilities to be able to transform more efficiently back and forth between (E, p_m, R_m) , $(E, \mu, P_\phi; \sigma)$ and (E, p, R, z) . Such capabilities include the methods developed by [49] and [50], that transforms via grid-optimized automatic differentiation and marker distributions, respectively.

Finally, the versatility and standard of the OWCF will provide a useful and easy-to-use toolbox for understanding the complexity of fast-ion orbits in tokamaks. It will help pave the way for future tomographic reconstructions of the fast-ion distribution using orbit

weight functions, and contribute to the realization of fusion as a virtually limitless source of sustainable energy.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

The authors report financial support was provided by Royal Danish Academy of Sciences and Letters. The authors report financial support was provided by European Consortium for the Development of Fusion Energy.

Data availability

Raw data were generated at the JET large scale facility. Derived data supporting the discussions of this study are available from the corresponding author upon reasonable request.

Acknowledgements

This work has received support from the Niels Bohr Foundation which is a merger of The Niels Bohr Grant, The Emil Herborg Grantpart, The Grant of MA Marcus Lorenzen, The Ole Rømer Foundation and The Julie Marie Vinter Hansen Travel Grant. The views and opinions expressed herein do not necessarily reflect those of The Royal Danish Academy of Sciences and Letters.

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No.

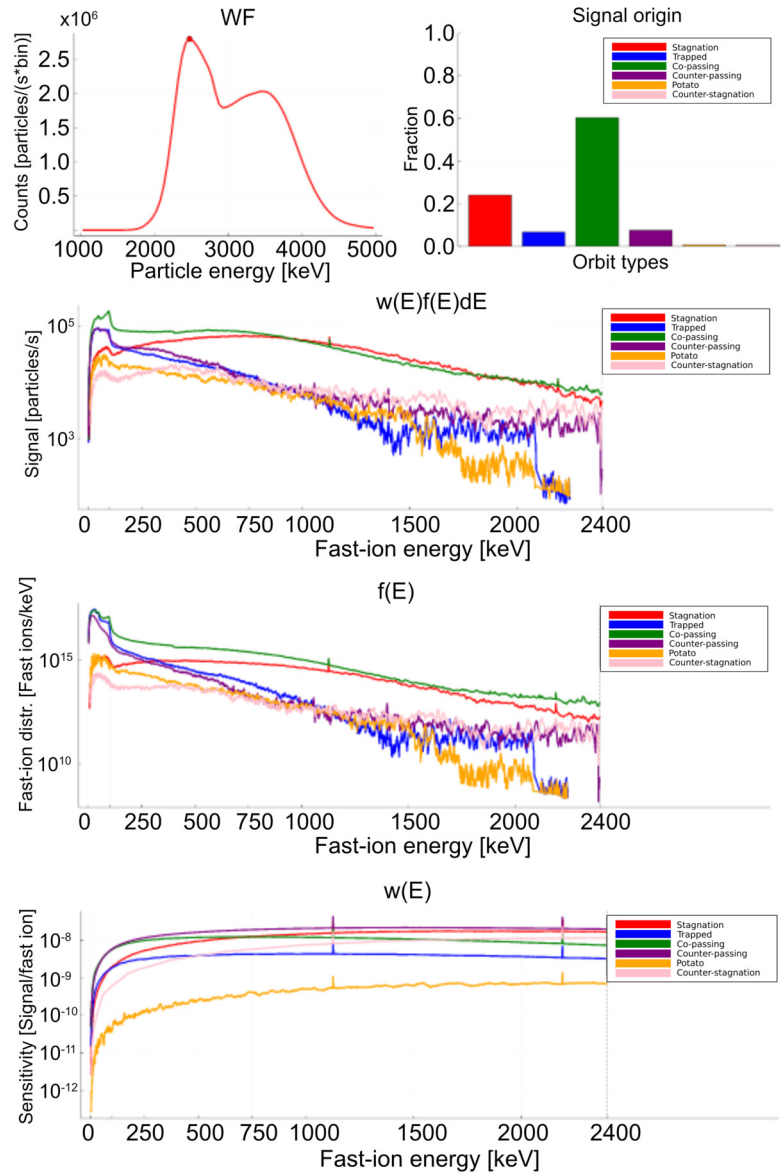


Fig. 13. An example screenshot of the signalWebApp.jl (A6) web application. The user controls have been omitted for brevity. JET shot 94701 at 10.8 seconds, TRANSP ID V01.

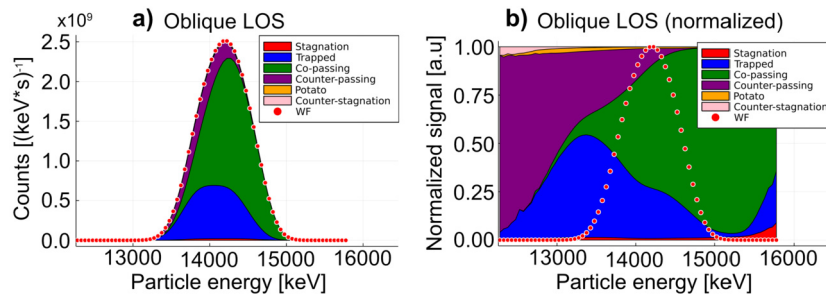


Fig. 14. An example of a) a diagnostic signal WF split into its orbit-type constituents. In b) the signal amplitude has been set to 1.0 for all diagnostic measurement bins, to allow the orbit-type fractions to be examined in detail. The absolute and normalized diagnostic signal has been superimposed for reference in a) and b), respectively.

101052200 – EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

References

- [1] B. Madsen, Reconstructing the fast-ion velocity distribution in fusion plasmas from sparse datasets, Ph.D. thesis, Technical University of Denmark, 2020.
- [2] J. Mailloux, et al., Nucl. Fusion 62 (2022) 042026, <https://doi.org/10.1088/1741-4326/ac47b4>.

- [3] H. Järleblad, et al., Nucl. Fusion 62 (2022) 112005, <https://doi.org/10.1088/1741-4326/ac63d3>.
- [4] S.D. Pinches, et al., Plasma Phys. Control. Fusion 46 (2004) B187, <https://doi.org/10.1088/0741-3335/46/12B/017>.
- [5] S.E. Sharapov, et al., Nucl. Fusion 53 (2013) 104022, <https://doi.org/10.1088/0029-5515/53/10/104022>.
- [6] W.W. Heidbrink, Phys. Plasmas 15 (2008) 055501, <https://doi.org/10.1063/1.2838239>.
- [7] J.A. Rome, Y.-K.M. Peng, Nucl. Fusion 19 (1979) 1193, <https://doi.org/10.1088/0029-5515/19/9/003>.
- [8] H. Järleblad, et al., Rev. Sci. Instrum. 92 (2021) 043526, <https://doi.org/10.1063/5.0040696>.
- [9] J. Eriksson, et al., Comput. Phys. Commun. 1996 (2016) 40–46, <https://doi.org/10.1016/j.cpc.2015.10.010>.
- [10] M. Salewski, et al., Nucl. Fusion 56 (2016) 106024, <https://doi.org/10.1088/0029-5515/56/10/106024>.
- [11] M. Salewski, Fast-ion diagnostic in fusion plasmas by velocity-space tomography, Dr. Techn. Thesis, January 2020.
- [12] M. Salewski, et al., Nucl. Fusion 52 (2012) 103008, <https://doi.org/10.1088/0029-5515/52/10/103008>.
- [13] M. Salewski, et al., Nucl. Fusion 57 (2017) 056001, <https://doi.org/10.1088/1741-4326/aa60e9>.
- [14] M. Salewski, et al., Fusion Sci. Technol. 74 (1-2) (2018) 23–36, <https://doi.org/10.1080/15361055.2017.1380482>.
- [15] M. Salewski, et al., Nucl. Fusion 58 (2018) 096019, <https://doi.org/10.1088/1741-4326/aace05>.
- [16] B. Madsen, et al., Nucl. Fusion 60 (2020) 066024, <https://doi.org/10.1088/1741-4326/ab82b5>.
- [17] M. Gatú Johnson, et al., Nucl. Instrum. Methods Phys. Res., Sect. A 591 (2008) 417–430, <https://doi.org/10.1016/j.nima.2008.03.010>.
- [18] W.W. Heidbrink, et al., Plasma Phys. Control. Fusion 49 (2007) 1457–1475, <https://doi.org/10.1088/0741-3335/49/9/008>.
- [19] L. Stagner, et al., Nucl. Fusion 62 (2022) 026033, <https://doi.org/10.1088/1741-4326/ac3ed2>.
- [20] M. Salewski, et al., Nucl. Fusion 54 (2014) 023006, <https://doi.org/10.1088/0029-5515/54/2/023005>.
- [21] M. Salewski, et al., Plasma Phys. Control. Fusion 56 (10) (2014) 105005, <https://doi.org/10.1088/0741-3335/56/10/105005>.
- [22] M. Salewski, et al., Nucl. Fusion 53 (6) (2013) 063019, <https://doi.org/10.1088/0029-5515/53/6/063019>.
- [23] L. Stagner, W.W. Heidbrink, Phys. Plasmas 24 (2017) 092505, <https://doi.org/10.1063/1.4990391>.
- [24] B. Madsen, et al., Plasma Phys. Control. Fusion 62 (2020) 115019, <https://doi.org/10.1088/1361-6587/abb79b>.
- [25] B.S. Schmidt, et al., Nucl. Fusion 63 (7) (2023) 076016, <https://doi.org/10.1088/1741-4326/acd6a6>.
- [26] J. Bezanson, et al., SIAM Rev. 59 (1) (2017) 65–98, <https://doi.org/10.1137/141000671>.
- [27] H. Järleblad, Orbit-space sensitivity of fast-ion diagnostics, Ph.D. thesis, Technical University of Denmark, Kgs. Lyngby, 2022.
- [28] H. Järleblad, et al., Fast-ion orbit origin of neutron measurements in the JET DT campaign, to be submitted to Nucl. Fusion.
- [29] I.J. Grady Booch, James Rumbaugh, Unified Modeling Language User Guide, 2nd edition, Addison-Wesley, Boston, 2005.
- [30] J. Breslau, M. Gorenkova, F. Poli, J. Sachdev, A. Pankin, G. Perumpilly, Transp. [Computer Software], <https://doi.org/10.1157/dc.20180627.4>, jun 2018.
- [31] A. Pankin, D. McCune, R. Andre, et al., Comput. Phys. Commun. 159 (3) (2004) 157–184, <https://doi.org/10.1016/j.cpc.2003.11.002>.
- [32] L. Stagner, Inference of the fast-ion distribution function, Ph.D. thesis, University of California, Irvine, 2018.
- [33] X. Tao, A.A. Chan, A.J. Brizard, Phys. Plasmas 14 (2007) 092107, <https://doi.org/10.1063/1.2773702>.
- [34] J. Connor, R. Hastie, Nucl. Fusion 15 (3) (1975) 415, <https://doi.org/10.1088/0029-5515/15/3/007>.
- [35] M. Rosenbluth, S. Putvinski, Nucl. Fusion 37 (10) (1997) 1355, <https://doi.org/10.1088/0029-5515/37/10/I03>.
- [36] W.W. Heidbrink, R.B. White, Phys. Plasmas 27 (2020) 030901, <https://doi.org/10.1063/1.5136237>.
- [37] J. Eriksson, et al., Plasma Phys. Control. Fusion 61 (1) (2018) 014027, <https://doi.org/10.1088/1361-6587/aad8a6>.
- [38] M. Nocente, et al., Rev. Sci. Instrum. 81 (2010) 10D321, <https://doi.org/10.1063/1.3501386>.
- [39] M. Nocente, et al., Nucl. Fusion 57 (7) (2017) 076016, <https://doi.org/10.1088/1741-4326/aa6f7d>.
- [40] M. Kikuchi, K. Lackner, M.Q. Tran, Fusion Physics, International Atomic Energy Agency, Vienna, Austria, 2012.
- [41] J.E. Perry, S.J. Bame Jr., Phys. Rev. 99 (5) (1955) 1386.
- [42] W.W. Heidbrink, et al., Plasma Phys. Control. Fusion 63 (2021) 055008, <https://doi.org/10.1088/1361-6587/abeda0>.
- [43] M. Salewski, et al., Nucl. Fusion 55 (9) (2015) 093029, <https://doi.org/10.1088/0029-5515/55/9/093029>.
- [44] L. Stagner, Inference of the fast-ion distribution function, Ph.D. thesis, University of California, Irvine, 2018.
- [45] J.P. Berrut, G. Klein, J. Comput. Appl. Math. 259 (2014) 95–107, <https://doi.org/10.1016/j.cam.2013.03.044>.
- [46] F.F. Little, et al., Surf. Comput. Aided Geom. Des. (1983) 99–107.
- [47] Ye.O. Kazakov, et al., Phys. Plasmas 28 (2021) 020501, <https://doi.org/10.1063/5.0021818>.
- [48] L. Stagner, H. Järleblad, <https://github.com/JuliaFusion/GuidingCenterOrbits.jl>, 2022.
- [49] S. Benjamin, Distribution transforms for orbit tomography, Master's thesis, The Australian National University, July 2021.
- [50] A. Bierwage, et al., Comput. Phys. Commun. 275 (2022) 108305, <https://doi.org/10.1016/j.cpc.2022.108305>.