

# Geometric integrator for charged particle orbits in axisymmetric fusion devices

S. V. Kasilov<sup>1,2</sup>, A. M. Runov<sup>3§</sup>, and W. Kernbichler<sup>2</sup>

<sup>1</sup>*Institute of Plasma Physics,*

*National Science Center “Kharkov Institute of Physics and Technology”,*

*61108, Kharkov, Ukraine,*

<sup>2</sup>*Fusion@ÖAW,*

*Institut für Theoretische Physik - Computational Physics,*

*Technische Universität Graz*

*Petersgasse 16, A-8010 Graz, Austria and*

<sup>3</sup>*Max-Planck-Institut für Plasmaphysik, D-17491 Greifswald, Germany*

## Abstract

A semi-analytical geometric integrator of guiding center orbits in an axisymmetric tokamak is described. The integrator preserves all three invariants of motion up to computer accuracy at the expense of reduced orbit accuracy and it is roughly an order of magnitude more efficient than a direct solution of the equations of guiding center motion with a standard high order adaptive ODE integrator.

*Key words:* plasma physics; kinetic modelling; numerical integrators

---

§ Corresponding author, runov@ipp.mpg.de, tel. +493834882432

## I. INTRODUCTION

An evaluation of the distribution function and/or its moments by direct modelling of particle orbits is widely used in plasma physics (see, e.g., codes like EUTERPE<sup>1,2</sup> or ASCOT<sup>3</sup>). An efficient algorithm for calculation of trajectories of charged particles in complex (quasi-)stationary magnetic and electric fields is one of the key issues in such codes because of the high number of test particle orbits required to minimize the statistical error of such calculations, which scales inversely with the square root of the number of test particles. This issue is especially important for global transport modelling (e.g., see Ref. 4) where the profiles of plasma parameters are calculated self-consistently from test particle trajectories, which have to be traced over the profile relaxation (confinement) time.

Within transport modelling, computation of stochastic test particle orbits<sup>5</sup> requires the solution of guiding center equations<sup>6,7</sup>, which is usually performed with help of general-purpose ODE integrators. In case of axisymmetric systems (tokamaks) the guiding center motion is fully integrable, because there exist three integrals of motion, which fully determine each orbit in the 5D phase space: The total energy  $w$ , magnetic moment  $\mu$  and the canonical toroidal angular momentum  $p_\varphi$ , respectively given by

$$w = \frac{m(v_\perp^2 + v_\parallel^2)}{2} + e\Phi, \quad \mu = \frac{mv_\perp^2}{2B}, \quad p_\varphi = mv_\parallel \frac{B_\varphi}{B} + \frac{e}{c}A_\varphi, \quad (1)$$

with electrostatic potential  $\Phi$ , magnetic field module  $B$ , co-variant toroidal component of the magnetic field  $B_\varphi$ , co-variant toroidal component of the vector potential  $A_\varphi$ , speed of light  $c$ , particle charge  $e$ , mass  $m$ , perpendicular velocity  $v_\perp$ , and parallel velocity  $v_\parallel$ . An accurate conservation of the invariants (1) is of primary importance for transport modelling in drift kinetic or drift fluid approximation while other accuracy requirements related to orbits can be significantly relaxed. Algorithms with such exact (up to computer accuracy) conservation of invariants are called *geometric integrators* (see, e.g., Ref. 8). These integrators preserve the geometry of the exact phase space flow (in particular, orbits resulting from the integrator of this paper stay exactly closed in the poloidal plane  $\varphi = \text{const}$  unless they cross the boundary of the computation domain), but the orbits do not necessarily satisfy Hamiltonian equations of motion with some (slightly modified) Hamiltonian as in the case of the symplectic integrators<sup>9</sup> being a sub-class of geometric integrators.

In the following sections we will introduce and study such an integrator suitable for

transport modelling of axisymmetric fusion devices. In a comparison with commonly used general-purpose ODE integrators one can expect two advantages: First, in numerical efficiency; and second, such an algorithm should be less sensitive to the accuracy of the representation of the electromagnetic field allowing also for numerical inaccuracies resulting, in particular, from the statistical noise in the data.

## II. DERIVATION OF THE INTEGRATOR

In general magnetic field geometry, equations of guiding center motion with invariants  $w$  and  $\mu$  used as velocity space variables are<sup>7</sup>

$$\dot{\mathbf{r}} = \frac{v_{\parallel} \mathbf{B}^*}{B h_{\parallel}^*}, \quad \mathbf{B}^* = \nabla \times \left( \mathbf{A} + \frac{m c v_{\parallel}}{e B} \mathbf{B} \right), \quad h_{\parallel}^* = \frac{\mathbf{B} \cdot \mathbf{B}^*}{B^2}, \quad (2)$$

where  $\mathbf{B}$  and  $\mathbf{A}$  are magnetic field and vector potential, respectively, and  $v_{\parallel} = v_{\parallel}(\mathbf{r}, w, \mu, \sigma)$  is determined by the first two Eqs in (1) and parallel velocity sign  $\sigma = \pm$ . In axisymmetric geometry using cylindrical variables  $(R, \varphi, Z)$ , equations of motion omitting the symmetry variable  $\varphi$  take the form

$$\begin{aligned} \dot{R} &= -\frac{1}{R h_{\parallel}^*} \left( \frac{\partial A_{\varphi}}{\partial Z} \frac{1}{m B_{\varphi}} \left( p_{\varphi} - \frac{e}{c} A_{\varphi} \right) + \frac{m c B_{\varphi}}{2e} \frac{\partial}{\partial Z} \left( \frac{v_{\parallel}}{B} \right)^2 \right), \\ \dot{Z} &= \frac{1}{R h_{\parallel}^*} \left( \frac{\partial A_{\varphi}}{\partial R} \frac{1}{m B_{\varphi}} \left( p_{\varphi} - \frac{e}{c} A_{\varphi} \right) + \frac{m c B_{\varphi}}{2e} \frac{\partial}{\partial R} \left( \frac{v_{\parallel}}{B} \right)^2 \right). \end{aligned} \quad (3)$$

The varying part of  $B_{\varphi} = B_{\varphi}(\psi)$  as a function of poloidal flux  $\psi = -A_{\varphi}$  is of the order of plasma beta or of the square of the ratio of the poloidal and toroidal field strengths. In most tokamaks this variation is only a few percent and can be safely ignored. With this assumption, Eqs. (3) are rewritten as

$$\dot{R} = -\frac{1}{R h_{\parallel}^*} \frac{\partial}{\partial Z} \mathcal{H}, \quad \dot{Z} = \frac{1}{R h_{\parallel}^*} \frac{\partial}{\partial R} \mathcal{H}, \quad (4)$$

where a Hamiltonian like function  $\mathcal{H}$  is

$$\mathcal{H} = \frac{c B_{\varphi}}{e} \left( \frac{w}{B^2} - \frac{\mu}{B} - \frac{e \Phi}{B^2} \right) - \frac{e}{2 m c B_{\varphi}} \left( A_{\varphi} - \frac{c}{e} p_{\varphi} \right)^2. \quad (5)$$

Particle orbits are then determined by the condition  $\mathcal{H} = 0$ . A numerically efficient low order approximation of these orbits can be obtained if one uses in (5) a linear interpolation for the following functions of coordinates  $\mathbf{x} = (x^1, x^2) \equiv (R, Z)$ ,

$$f_A(\mathbf{x}) \equiv A_{\varphi}, \quad f_B(\mathbf{x}) \equiv \frac{1}{B}, \quad f_{\Phi}(\mathbf{x}) \equiv \frac{\Phi}{B^2}, \quad (6)$$

discretized on a triangular mesh required for the continuous piecewise linear interpolation. As a result,  $\mathcal{H}$  (5) becomes a continuous piecewise quadratic function,

$$\mathcal{H} = \frac{1}{2} \sum_{i,j=1}^2 a_{ij} (x^i - x_a^i)(x^j - x_a^j) - \frac{1}{2} \sum_{i,j=1}^2 a_{ij} (x_0^i - x_a^i)(x_0^j - x_a^j), \quad (7)$$

where the coefficients  $a_{ij}$  are constant within a given triangle and are determined by initial values of the particle coordinates  $\mathbf{x}_0$  and the velocity components  $v_{\perp 0}$  and  $v_{\parallel 0}$  in this triangle as follows,

$$a_{ij} = \frac{1}{\alpha} \left( v_{\perp 0}^2 + v_{\parallel 0}^2 + \frac{2ef_{\Phi}(\mathbf{x}_0)}{mf_B^2(\mathbf{x}_0)} \right) \frac{\partial f_B}{\partial x^i} \frac{\partial f_B}{\partial x^j} - \alpha \frac{\partial f_A}{\partial x^i} \frac{\partial f_A}{\partial x^j}. \quad (8)$$

Here,  $\alpha = e(mcB_{\varphi})^{-1}$ , and the constants  $x_a^i$  are the solution to the following linear equation set,

$$\sum_{j=1}^2 a_{ij} (x_0^j - x_a^j) = v_{\parallel 0} f_B(\mathbf{x}_0) \frac{\partial f_A}{\partial x^i} + \left( v_{\perp 0}^2 + 2v_{\parallel 0}^2 + \frac{4ef_{\Phi}(\mathbf{x}_0)}{mf_B^2(\mathbf{x}_0)} \right) \frac{f_B(\mathbf{x}_0)}{2\alpha} \frac{\partial f_B}{\partial x^i} - \frac{e}{m\alpha} \frac{\partial f_{\Phi}}{\partial x^i}. \quad (9)$$

The spatial particle trajectories,  $\mathcal{H} = 0$ , are then continuous piecewise second order curves. Note that vector potential and magnetic field module in (6) are treated as independent functions within the above linear interpolation, which does not hold the relation  $B = |\nabla \times \mathbf{A}|$  anymore. Consequently, gradient- $B$  drift is retained in the orbits while it would be absent if  $B$  would be computed from that relation. Due to continuous interpolation of all functions of the coordinates in Eq. (1), orbits are continuous also in the velocity space where preferable variables are  $(v_{\perp}, v_{\parallel})$ . Those are more convenient for modelling of collisions and of anomalous transport, both required in transport simulations. A parametric (time-dependent) form of orbit segments is obtained from the equations of motion (4), which are further simplified by ignoring the Larmor radius correction,  $h_{\parallel}^* \rightarrow 1$  and by replacing  $R$  with a constant  $\bar{R}$  being the radial coordinate of the center of mass of a given triangle,

$$\dot{x}^i = \frac{(-1)^i}{\bar{R}} \sum_{j=1}^2 a_{ij} (x^j - x_a^j). \quad (10)$$

This approximation does not affect the orbit shape. Namely, particles move along exactly the same orbits in the phase space but time dependence of phase space coordinates is slightly changed. Depending on the sign of the determinant  $D = \det(a_{ij})$ , orbits are either ellipses ( $D > 0$ ) or hyperbolas  $D < 0$ , respectively given by

$$\begin{aligned} x^i &= x_a^i + x_c^i \cos(\omega\Delta t) + x_s^i \sin(\omega\Delta t), & D > 0, \\ x^i &= x_a^i + x_c^i \cosh(\omega\Delta t) + x_s^i \sinh(\omega\Delta t), & D < 0, \end{aligned} \quad (11)$$

where  $\Delta t$  is the integration time step,

$$\omega = \frac{|D|^{1/2}}{R}, \quad x_c^i = x_0^i - x_a^i, \quad x_s^i = \frac{(-1)^i}{|D|^{1/2}} \sum_{j=1}^2 a_{3-i,j} x_c^j. \quad (12)$$

The numerical implementation of this integrator is quite straightforward: One follows the test particle for the time step  $\Delta t$  (usually determined by collisions or anomalous transport) using Eqs. (11) and modifies its velocities  $(v_\perp, v_\parallel)$  according to the conservation laws (1). If  $\Delta t$  exceeds the time to reach a boundary of the triangle, one stops the trajectory at this point of intersection. Its coordinates,  $\mathbf{x}_b$ , satisfy a quadratic equation, and the corresponding time can be found from (11) for  $\mathbf{x} = \mathbf{x}_b$ . Then one follows the particle in the next triangle using the point of intersection as a new starting point  $\mathbf{x}_0$  and its local velocities as new  $v_{\perp 0}$  and  $v_{\parallel 0}$ .

### III. BENCHMARKING

We now compare the results obtained by the geometric integrator derived in the previous section and a conventional adaptive ODE integrator, *odeint* (Ref. 10). The magnetic configuration we use is ITER-like (the same had been used in Ref. 4) with zero electric field everywhere.

The triangular mesh required for the geometric integrator has been produced from an nearly orthogonal field-aligned quadrangular mesh used for fluid modelling by the B2 code<sup>11</sup>. The *odeint*<sup>10</sup> solver has been employed here for the full 3D system of guiding center equations<sup>7</sup> in the covariant representation<sup>12</sup> for cylindrical coordinates. A very accurate divergence-free representation of the magnetic field based on 2D spline interpolation (5th order) of the poloidal flux function has been used for this integrator, which acts as a reference case.

As an example, a few collisionless trajectories calculated by the geometric integrator for trapped deuterium and iron ions are presented in Fig. 1 together with pertinent orbits from the reference case. In addition, the used triangular mesh is also shown. All orbits almost coincide in the overview figure (a), but the magnification of the zones (b) and (c) shows a noticeable difference in some details of the trajectories, especially in the region of the banana tips shown in (c). This shift is related to the crude representation of the field-related values, Eq. (6), on the mesh. The integrator preserves all constants of motion exactly, therefore, an initial error introduced by a linear interpolation at the start point (arrow in Fig. 1, b)

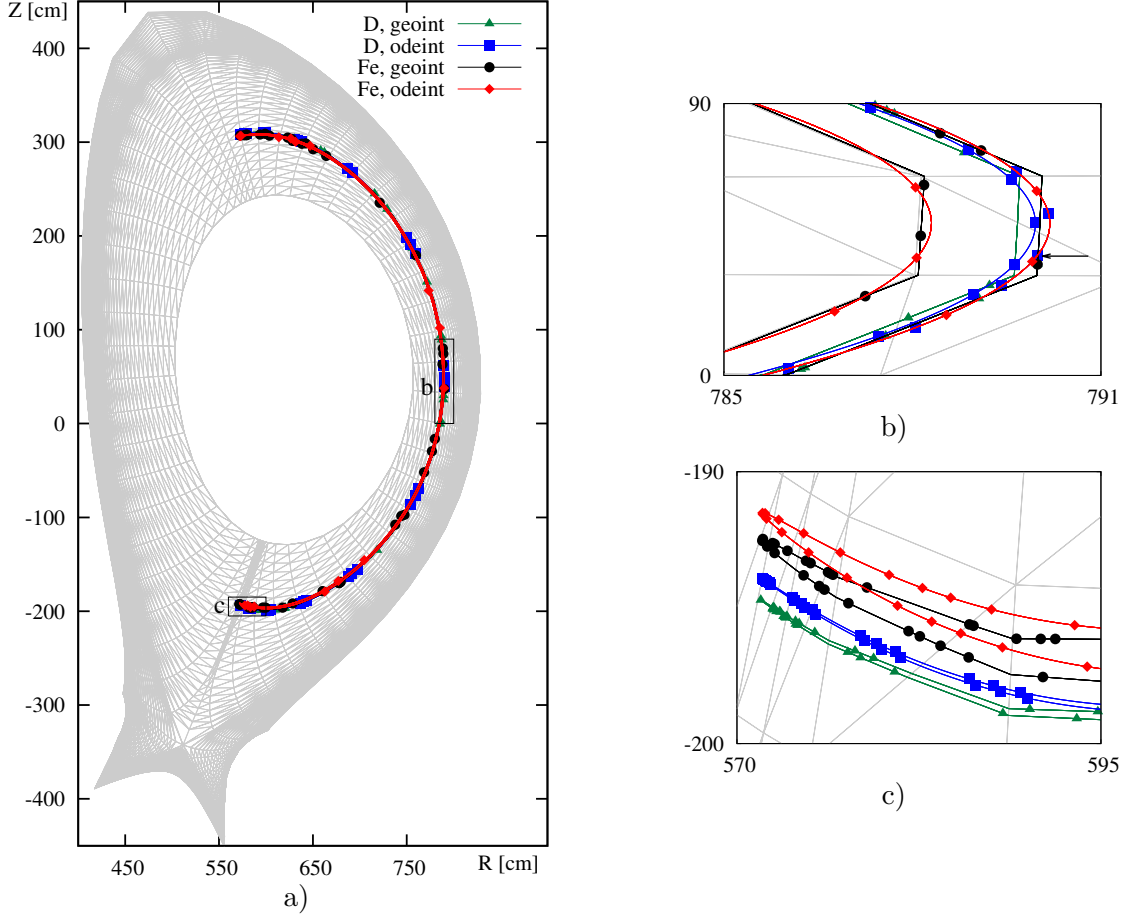


FIG. 1: a) – Geometry of an ITER-like configuration and example orbits (triangles, green – D ions, geometric integrator (Sec. II); squares, blue – D ions, *odeint*<sup>10</sup>; circles, black – Fe ions, geometric integrator; diamonds, red – Fe ions, *odeint*. b) and c) – magnification of the pertinent zones in a). The arrow points to the starting point of all four trajectories depicted.

propagates along the whole trajectory. Indeed, if one starts the trajectories near the corner of a cell (where this error vanishes), the shift between the banana tips vanishes, too. It should be noted that the B2 mesh is optimized to represent the hydrodynamical plasma parameters. For transport studies, this interpolation can be essentially improved without suffering from larger costs. The strategy of grid refinement is to improve the accuracy of linear interpolation of the poloidal flux function, which is mainly responsible for the error in Fig. 1. Such a refinement is needed only in the regions where accurate orbits are required, namely, in pre-separatrix and pedestal regions but not in the plasma core where, as shown below, errors in the orbits have small effect on the radial transport.

The crucial point is how, if at all, inaccuracies of the orbits affect the overall transport

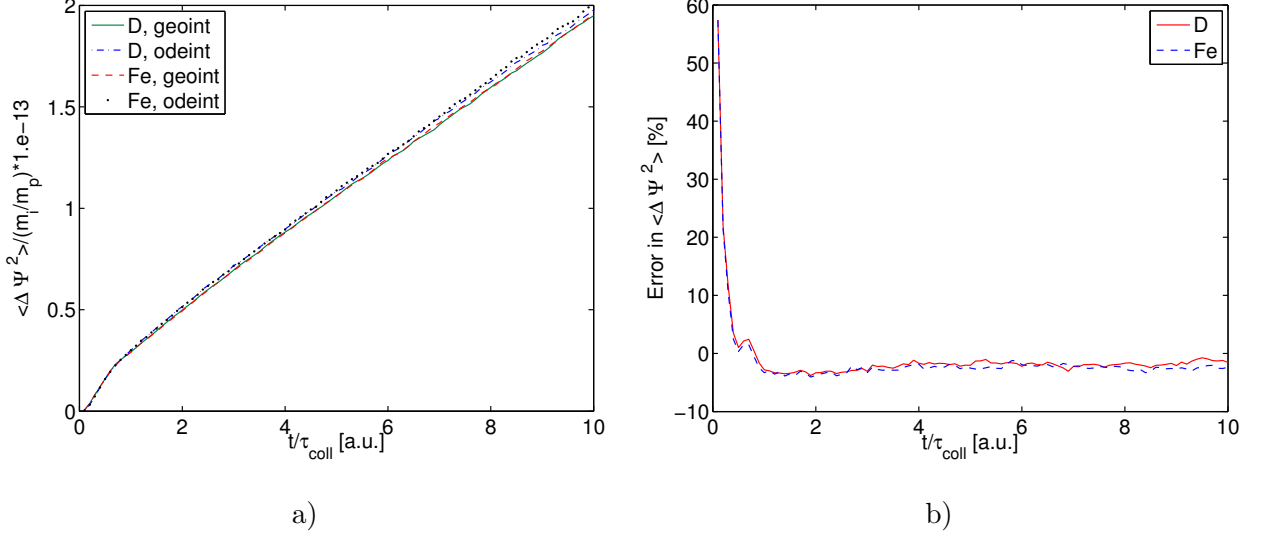


FIG. 2: a) – Variance of the poloidal flux deviation  $\langle(\psi - \psi_0)^2\rangle$  normalized to the mass ratio as a function of normalized time: solid line – D, geometric integrator, dash-dot line – D, conventional integrator, dash line – Fe, geometric integrator, dotted line – Fe, conventional integrator. b) – the relative error of the variance,  $(\langle(\psi - \psi_0)^2\rangle_{geoint} - \langle(\psi - \psi_0)^2\rangle_{odeint})/\langle(\psi - \psi_0)^2\rangle_{odeint}$  in percent: solid line – D, dash line – Fe.

properties of the system. To check this, we add collisions to our model according to Ref. 5, i.e. introducing specific random changes of pitch parameter and velocity module between the integration steps of the guiding center equations. Test particles are started at the same point as before, but with random, evenly distributed pitch parameters  $v_{\parallel 0}(v_{\perp 0}^2 + v_{\parallel 0}^2)^{-1/2}$ , and are followed for 10 collisional times  $\tau_{coll}$ . During this process, the time evolution of the poloidal flux at the particle location is stored. The variance of the poloidal flux deviation  $\langle(\psi - \psi_0)^2\rangle$  ( $\psi_0$  is the flux at the start point) serves as a measure of the neoclassical radial diffusion coefficient of the system. The results of the Monte Carlo computations of this variance (normalized to the mass ratio for D and Fe), obtained with help of the geometric integrator and of the reference case integrator are presented as functions of time in Fig 2, a). Again, the curves almost coincide. To emphasize the difference, we plot the relative error,  $(\langle(\psi - \psi_0)^2\rangle_{geoint} - \langle(\psi - \psi_0)^2\rangle_{odeint})/\langle(\psi - \psi_0)^2\rangle_{odeint}$  in percent, see Fig 2, b). A significant relative difference is present only at the start of the trajectories,  $t \leq \tau_{coll}$ , where the change of  $\psi$  due to collisions is small compared to the orbit width and thus is of no importance. After one collision time the difference converges to an acceptable value of a few percent. As it was mentioned above, this value can be reduced further by optimization of the mesh.

The estimation of the relative numerical efficiency of the integrators is a subtle matter: The systems which are solved are not quite the same. Moreover, the CPU time ratio strongly depends on details of the implementation of both integrators, even on the realization of the compiler intrinsic functions (see Eq. (11)). One can estimate the overall effect of our effort by a rule of thumb comparing the computing times for calculations pertinent to Fig 2 where, of course, the CPU time spent on collisions was excluded from the measurement. For the same statistical error, one gains a factor of ten when using the geometric integrator. This value would be even higher if one compares the efficiency for a self-consistent transport problem, like the one described in Ref. 4. A frequently used tool in transport studies is a so-called track length estimator (see, e.g., the textbook Ref. 13), where the knowledge of the dwell time of each particle per cell is required. This means, that when using a conventional integrator also the computations of any intersection of trajectories with cell boundaries are required. To find all these intersection points makes the usage of the conventional integrator even more expensive relative to our algorithm, because those points are already automatically obtained when using the geometric integrator.

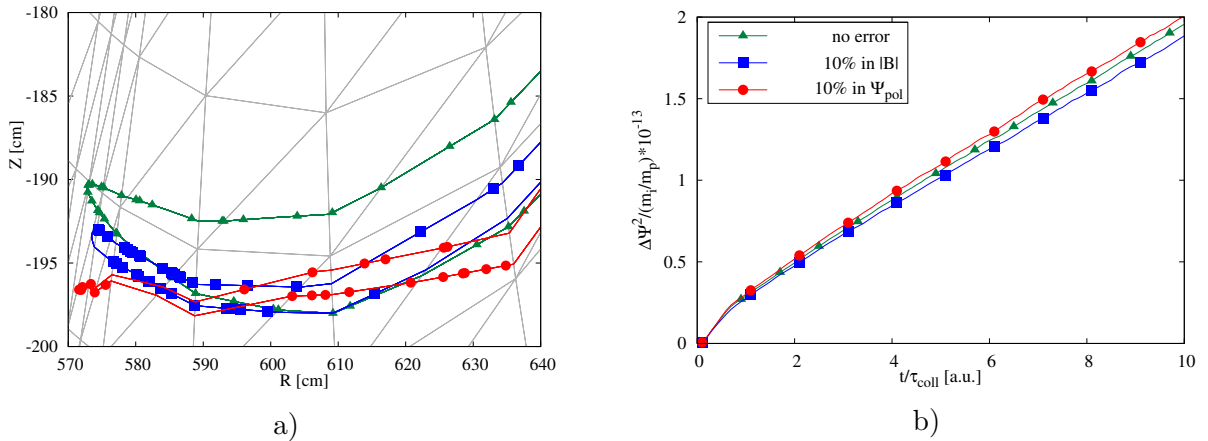


FIG. 3: Test of the geometric integrator for Fe ions moving in the field with errors: Squares – 10% error introduced in  $|B|$ , circles – 10% error introduced in  $\psi_{pol}$ , triangles – case without error. a) – orbits of Fe ions, magnification of the zone “c” (see Fig. 1). b) – Variance of the poloidal flux deviation  $\langle (\psi - \psi_0)^2 \rangle$  normalized to the mass ratio as a function of normalized time.

Another aspect is the quality of the fields entering Eq. (2). The magnetic field used in these test calculations had been represented very accurately and is smooth, and the electric field was set to zero – this allows us to use a high-order adaptive integrator, at least for



purposes of this test. However, in a realistic case the electric field should be calculated self-consistently. This electric field is represented piecewisely by constants within mesh cells and, therefore, it is discontinuous at cell boundaries. In addition, this field has a significant numerical noise in case of Monte Carlo modelling and cannot be represented by smooth dependencies in the whole space. This makes the usage of high-order numerical schemes problematic and would require filtering of the statistically obtained electric field. This might introduce artifacts and introduces further CPU expenses. However, the low-order representation of all field-related values in Eq. (6) allows one to use the noisy field quantities directly within the geometric integrator.

Effect of the noise in field quantities on the orbits and transport is demonstrated in Fig. 3. In this test, random errors with relative value of 10 % have been introduced in the values of the magnetic field module  $B$  and poloidal flux  $\psi$  at the grid nodes. Despite the visible local change of the orbits, effect of the errors on the variance  $\langle(\psi - \psi_0)^2\rangle$  is not dramatic and is of the order of the noise level. Note that change of this quantity can be both, positive and negative depending on the noise pattern. In particular, larger radial orbit excursions from the exact flux surfaces induced by the noise in the poloidal flux  $\psi$  do not lead indefinitely to the increase in radial transport because this transport is computed with respect to flux surfaces approximated by linear interpolation (their cross sections are polygons of high order) which are affected by the noise in the same way as the orbits.

#### IV. SUMMARY

The semi-analytical geometric integrator described here has roughly an order of magnitude higher efficiency than a conventional method for guiding center orbit integration. Essentially this is due to the fact that all analytical results employed by this integrator are expressed in terms of elementary functions, which are intrinsic functions of FORTRAN compilers with a pertinent CPU cost of the order of a single algebraic operation. In transport modelling, the efficiency of this geometric integrator is even higher because the track length estimator frequently used for the evaluation of macroscopic parameters from Monte Carlo test particle distributions does not require any additional search of orbit intersections with the given mesh. These data are obtained by the geometric integrator as a by-product of orbit tracing. The integrator has been already employed in 2D kinetic transport modelling

in a tokamak<sup>4</sup> and can also be applied for the modelling of kinetic effects in combination with 2D fluid and neutral transport codes.

## V. ACKNOWLEDGMENTS

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

- 
- <sup>1</sup> G. Jost, T. M. Tran, W. A. Cooper, L. Villard, and K. Appert, *Phys. Plasmas* **8**, 3321 (2001).
- <sup>2</sup> V. Kornilov, R. Kleiber, R. Hatzky, L. Villard, and G. Jost, *Phys. Plasmas* **11**, 3196 (2004).
- <sup>3</sup> J. A. Heikkinen, T. P. Kiviniemi, T. Kurki-Suonio, A. G. Peeters, and S. K. Sipilä, *J. Comput. Phys.* **173**, 527 (2001).
- <sup>4</sup> A. M. Runov, S. V. Kasilov, and P. Helander, *J. Comput. Phys.* **300**, 605 (2015).
- <sup>5</sup> A. H. Boozer and G. Kuo-Petravic, *Phys. Fluids* **24**, 851 (1981).
- <sup>6</sup> A. I. Morozov and L. S. Solov'ev, in *Reviews of plasma physics*, edited by M. A. Leontovitch (Consultants Bureau, New York, 1966), Vol. 2.
- <sup>7</sup> R. J. Littlejohn, *Journ. Plasma Phys.* **29**, 111 (1983).
- <sup>8</sup> E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, Berlin Heidelberg New York, 2006).
- <sup>9</sup> J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems, Applied Mathematics and Mathematical Computation* (Chapman & Hall, London - Glasgow - New York - Tokyo - Melbourne - Madras, 1994).
- <sup>10</sup> H. W. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran 77: The Art of Scientific Computing* (Press Syndicate of the University of Cambridge, The Pitt Building, Trumpington Street, Cambridge CB2 1RP 40 West 20th Street, New York, NY 10011-4211, USA 10 Stamford Road, Oakleigh, Melbourne 3166, Australia, 1992).
- <sup>11</sup> R. Schneider, X. Bonnin, K. Borrass, D. P. Coster, H. Kastelewicz, D. Reiter, V. A. Rozhansky, and B. J. Braams, *Contrib. Plasma Phys.* **46**, 3 (2006).

- <sup>12</sup> S. V. Kasilov, W. Kernbichler, V. V. Nemov, and M. F. Heyn, *Phys. Plasmas* **9**, 3508 (2002).
- <sup>13</sup> J. Spanier and E. M. Gelbard, *Monte Carlo principles and neutron transport problems* (Dover Publications, New York, 2008).