ARTICLE IN PRESS

Computer Physics Communications ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

ISDEP: Integrator of stochastic differential equations for plasmas

J.L. Velasco [a,b], A. Bustos [a,b,*], F. Castejón [a,b], L.A. Fernández [b,c], V. Martin-Mayor [b,c], A. Tarancón [b]

[a] National Fusion Laboratory, EURATOM-CIEMAT, 28040 Madrid, Spain

[b] Instituto de Biocomputación y Física de los Sistemas Complejos (BIFI), Universidad de Zaragoza, 50018 Zaragoza, Spain

[c] Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain

ARTICLE INFO

Article history:
Received 5 October 2011
Received in revised form
16 February 2012
Accepted 12 April 2012
Available online xxxx

Keywords:
MC code
Plasma transport

ABSTRACT

In this paper we present a general description of the ISDEP code (Integrator of Stochastic Differential Equations for Plasmas) and a brief overview of its physical results and applications so far. ISDEP is a Monte Carlo code that calculates the distribution function of a minority population of ions in a magnetized plasma. It solves the ion equations of motion taking into account the complex 3D structure of fusion devices, the confining electromagnetic field and collisions with other plasma species. The Monte Carlo method used is based on the equivalence between the Fokker–Planck and Langevin equations. This allows ISDEP to run in distributed computing platforms without communication between nodes with almost linear scaling. This paper intends to be a general description and a reference paper in ISDEP.

1. Introduction

ISDEP (Integrator of Stochastic Differential Equations for Plasmas) is a code devoted to solve the dynamics of a minority population of ions in a complex 3D fusion device. It was developed under the CIEMAT[1]-BIFI[2]-UCM[3] collaboration in 2007 and is in continuous improvement. From a physical point of view, ISDEP avoids several common approximations in neoclassical transport codes. As an example, it makes use of the Cartesian coordinates instead of the Boozer or magnetic coordinates [1]. Boozer coordinates are specific coordinates for magnetically confined plasmas, but they do not allow the representation of magnetic islands, ergodic zones or points in space outside the plasma boundary where the field lines are open. Other approximations avoided are related to the size of the orbits in the device and the diffusive nature of the transport processes. Finally, the kinetic energy of the studied particles does not need to be conserved, which allows the inclusion of strong electric fields in the dynamics. The minority population can be thermal particles, obtaining then information from the plasma bulk. It can also be fast particles coming from heating systems or impurities. The code considers collisions of the minority with all the plasma species. In addition, this code was designed to run in distributed computing platforms, including desktop grids and computing grids, motivating the development of these computing architectures.

The layout of this paper is as follows: in Sections 2 and 3 we illustrate the Monte Carlo method used and the ion equations of motion. In Section 4 we describe the architecture of the code and the numerical techniques used. Section 5 summarizes the physical results obtained up to date and in Section 6 we present our conclusions and summary.

2. Monte Carlo method

The basis of the Monte Carlo method used in ISDEP relies on the equivalence between the Fokker–Planck and Langevin equations [2]. The Fokker–Planck equation is a linear 2nd order partial differential equation for a distribution function of a minority population of particles (called test particles) that interact with a static background. The Langevin approach is equivalent to this description, but providing Stochastic Differential Equations (SDEs) [2] for a single test particle motion. Integrating many test particle trajectories and analyzing the results it is possible to obtain the solution to the original Fokker–Planck equation.

ISDEP integrates the trajectories of test particles, taking into account collisions with ions and electrons from the background, the confining magnetic field and the electric fields existing in the plasma. In order to reduce computational requirements, a very common approximation in Plasma Physics, called the guiding center (GC) approximation [3,4], is used in the code. The motion of a charged particle in a magnetic field may be divided into the

* Corresponding author at: National Fusion Laboratory, EURATOM-CIEMAT, 28040 Madrid, Spain. Tel.: +34 91 346 6685.
E-mail address: andres.debustos@ciemat.es (A. Bustos).

1 Centro de Investigaciones Energéticas, Medio Ambientales y Tecnólogicas, Madrid, Spain.

2 Instituto de Biocomputación y Física de los Sistemas Complejos, Zaragoza, Spain.

3 Complutense University of Madrid, Madrid, Spain.

0010-4655/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.cpc.2012.04.005

fast gyration around a magnetic field line and the motion of the gyration center along this line. The time scale of the fast gyration is typically 10^4–10^5 times faster than the motion along the field line. If the gyroradius, i.e. the Larmor radius, is much smaller than any other characteristic length of the system, an average in the gyromotion can simplify substantially the dynamics of the test particle. In this way the gyro-angle coordinate disappears and the phase space is reduced from 6 to 5 dimensions. The guiding center coordinates chosen are $(\mathbf{r}_{gc}, v^2, \lambda)$, where $\mathbf{r}_{gc}$ is the guiding center position, v^2 is the normalized particle kinetic energy and $\lambda = \mathbf{v}\cdot\mathbf{B}/(B\,v)$ is called *pitch*. In most cases the ion Larmor radius in fusion devices is $r_L \sim 1$ mm, much smaller than any other characteristic length.

Denoting by x the point in the 5D phase space, the time evolution of the distribution function $f(x,t)$ in the Fokker–Planck description is given by the convective $(F^i(x,t))$ and the diffusive transport $(G^i_j(x,t))$ tensors:

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial}{\partial x^i}\left(-F^i(x,t) + \frac{1}{2}\frac{\partial}{\partial x^j}G^i_k(x,t)G^{kj}(x,t)\right)f(x,t). \tag{1}$$

The explicit form of F_i and G_{ij} will be discussed below. The linear nature of the Fokker–Planck equation is manifested in the independence of F_i and G_{ij} with respect to $f(x,t)$. The equivalent set of Stochastic Differential Equations (SDEs) in Itô's sense [2] (i.e. Langevin equations) is:

$$\mathrm{d}x^i = F^i(x,t)\ \mathrm{d}t + G^i_j(x,t)\ \mathrm{d}W^j. \tag{2}$$

Now the coordinates in phase space x^i refer to the motion of a single particle, whose trajectory is determined by the background via F^i and G^i_j. The Wiener process, $\mathrm{d}W^j$ [2], is a stochastic process with independent increments and Gaussian distribution such that: $\mathrm{d}W^j(0) = 0$, $\langle \mathrm{d}W^j(t)\rangle = 0$, $\langle \mathrm{d}W^j(t)\ \mathrm{d}W^k(t)\rangle = \delta^{jk}\,\mathrm{d}t$. In our case, the problem consists of an SDE system of five equations with two Wiener processes. They represent the random effect of the collisions with the plasma background. The SDE's can be transformed to the Stratonovich [2] convention because it is suitable for additional numerical methods. An SDE in Stratonovich form is indicated by $\circ\ \mathrm{d}W$:

$$\mathrm{d}x^i = \hat{F}^i(x,t)\ \mathrm{d}t + G^i_j(x,t)\circ \mathrm{d}W^j. \tag{3}$$

Under very general assumptions [2] the Stratonovich and the Itô formulations are equivalent. Usually the Itô formulation is employed in developing the theory of SDE, while the Stratonovich convention is used in practical calculations. The Stratonovich convective term is given by:

$$\hat{F}^i(x,t) = F^i(x,t) - \frac{1}{2}\frac{\partial G^{ik}(x,t)}{\partial x^j}\,G^j_k(x,t), \tag{4}$$

while the diffusion tensor G^{ij} does not change in this convention. ISDEP usually employs the Stratonovich convention to solve the equations of motion presented in the next section.

3. Equations of motion

In this section the complete equations of motion of a single particle are shown. Let the guiding center coordinates be x^i, $i = 1\ldots5$. The notation of the physical quantities used in this section is summarized in Table 1. The position of the particle in real space evolves according to the GC equations of motion [4]. The differential equations for v^2 and λ are obtained from the energy and magnetic moment conservation in the absence of collisions [4]. The derivation of the Fokker–Planck drift and diffusion tensors (F^C_i and G_{ij}, $i,j = v^2, \lambda$) that account for the collisions can be found in Refs. [5–7]. For the sake of completeness, we recall them in Eqs. (13) and (14), below. It can be seen that the evolution of the position is deterministic and the diffusion in (v^2,λ) space is diagonal: $G^x_i = G^y_i = G^z_i = G^{v^2}_\lambda = G^\lambda_{v^2} = 0$, $G^{v^2}_{v^2} = G_{v^2}$, $G^\lambda_\lambda = G_\lambda$. Finally, the equations of motion are:

Table 1
Notation of the physical quantities in the equations of motion.

$\mathbf{r}_{gc}$	Guiding Center position
$\mathbf{v}$	GC velocity, normalized to c
λ	Particle pitch $= \mathbf{v}\cdot\mathbf{B}/vB$
v^2	Particle velocity square
$\mathbf{v}_D$	GC drift velocity
$\mathbf{v}_\parallel$	GC parallel velocity
$\mathbf{B}$	Confining magnetic field
$\mathbf{R}_c$	Curvature radius of $\mathbf{B}$
ρ	Effective radius
m	Proton mass
c	Speed of light
e	Elementary charge
$V, \mathbf{E}$	Plasma potential and electric field
n	Plasma density
$T_i,\ T_e$	Ion and electron temperatures
v_{th}	Plasma thermal velocity

$$\mathrm{d}\mathbf{r}_{gc} = \left(\mathbf{v}_\parallel + \mathbf{v}_D\right)\,\mathrm{d}t, \tag{5}$$

$$\mathbf{v}_\parallel = v\lambda\frac{\mathbf{B}}{B} + \frac{mv^2c^2(1-\lambda^2)}{eB^3}\mathbf{B}\cdot\left(\nabla\times\frac{\mathbf{B}}{B}\right)\mathbf{B}, \tag{6}$$

$$\mathbf{v}_D = \frac{mv^2c^2(1-\lambda^2)}{2eB^3}\mathbf{B}\times\nabla B + \frac{\mathbf{E}\times\mathbf{B}}{B^2} + \frac{mv^2c^2\lambda^2}{eB^3}\left(\frac{\mathbf{B}\times\mathbf{R}_c}{R_c^2}\right), \tag{7}$$

$$\mathrm{d}v^2 = \left(\frac{2e}{m}(\mathbf{E}\cdot\mathbf{v}_D) + F^C_{v^2}(x)\right)\,\mathrm{d}t + G_{v^2}(x)\,\mathrm{d}W^{v^2}, \tag{8}$$

$$\mathrm{d}\lambda = \left(\frac{1-\lambda^2}{2}\left(\frac{2ev_\parallel}{mv^3}\mathbf{E}\cdot\mathbf{v}_\parallel - \frac{\lambda}{B^3}\mathbf{E}\cdot(\mathbf{B}\times\nabla B) + \frac{2}{B^3}\mathbf{E}\cdot\left(\frac{\mathbf{B}\times\mathbf{R_c}}{R_c^2}\right)\right) - \frac{v_\parallel}{vB}\nabla B\cdot\mathbf{v}_\parallel - \frac{m\lambda v^2}{eB^3}\nabla B\cdot\left(\frac{\mathbf{B}\times\mathbf{R_c}}{R_c^2}\right)\right)\mathrm{d}t. + F^C_\lambda(x)\,\mathrm{d}t + G_\lambda(x)\,\mathrm{d}W^\lambda. \tag{9}$$

These equations require the introduction of the Braginskii deflection and energy slowing down frequencies for the two background species: electrons and ions. Let $b = e, i$, then:

$$\nu_B(e) = \frac{4}{3}\sqrt{\frac{2\pi}{m_e}}\frac{e^4 n\ln\Lambda}{T_e^{\frac{3}{2}}},$$
$$\nu_B(i) = \frac{4}{3}\sqrt{\frac{\pi}{m_i}}\frac{e^4 n\ln\Lambda}{T_i^{\frac{3}{2}}}, \tag{10}$$

$$\nu_d(b) = \frac{3}{2}\sqrt{\frac{\pi}{2}}\nu_B(b)\frac{m_b^2}{m_i^2}\frac{\hat{\Phi}(x_b)-\hat{\Psi}(x_b)}{x_b^3},$$
$$\nu_E(b) = 3\sqrt{\frac{\pi}{2}}\nu_B(b)\frac{m_i}{m_b}\frac{\hat{\Psi}(x_b)}{x_b}, \tag{11}$$

where $\ln\Lambda$ is the Coulomb logarithm for ions [7], $x_b = v/v_{th}(b)$ and

$$\hat{\Phi}(x) = \int_0^x \mathrm{d}y\,\frac{2}{\sqrt{\pi}}\mathrm{e}^{-y^2},$$
$$\hat{\Psi}(x) = \frac{\hat{\Phi}(x) - x\,\hat{\Phi}'(x)}{2x^2}. \tag{12}$$

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005

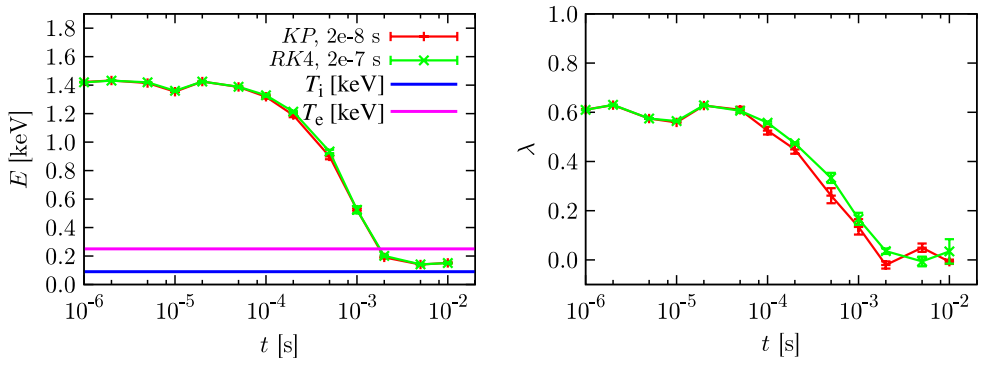

Fig. 1. Two different numerical methods applied to the same physical problem in TJ-II. We plot the evolution of the average energy and the average pitch. The background temperature is also depicted. Both methods arise to equal results, but the performance of the RK4 is better than the KP method (400 vs. 2000 s in the EULER cluster at CIEMAT).

Finally, F^C and G are:

$$F^C_{v^2} = -\sum_b \nu_E(b)\left(x_i^2 - \frac{x_b}{\sqrt{\pi}}\frac{e^{-x_b^2}}{\hat{\Psi}(x_b)}\frac{T_b}{T_i}\right),$$
$$F^C_{\lambda} = -\sum_b \lambda \nu_d(b). \tag{13}$$

$$G_{v^2} = 2x_i\sqrt{\sum_b \nu_E(b)\frac{T_b}{T_i}},$$
$$G_{\lambda} = \sqrt{\sum_b (1-\lambda^2)\nu_d(b)}. \tag{14}$$

This system of 5 SDE's formed by Eqs. (7)–(9) describes the dynamics of ions embedded in a thermal bath in arbitrary 3D geometry using the GC approximation. The physical properties of the test particle population can be extracted from a set of trajectories. In a Monte Carlo code the statistical errors scale as $1/\sqrt{N_t}$. Usually, $N_t \sim 10^5 - 10^7$ trajectories are needed.

3.1. Numerical algorithms

ISDEP has several numerical methods to solve the SDE system, with different properties and convergence orders. When dealing with numerical solution of SDE's one has to distinguish two types of convergence, strong and weak convergence [2]. Strong convergence is a concept similar to the usual convergence in ordinary differential equations, whereas weak convergence is related to averages and statistical quantities of a set of solutions. Since most of the ISDEP results are moments of the particle distribution function, weak convergence order is the lead criteria for the numerical methods used. As an example, an order 1 weak algorithm is the Euler–Maruyama scheme [2]:

$$x^i_{n+1} = x^i_n + F^i(x_n)\Delta t + G^i_j(x_n)\Delta W^j. \tag{15}$$

An order 2 weak algorithm is the Klauder–Petersen [KP] method [2] (note that $dW = \sqrt{dt}\,\eta$, $\eta = N(0,1)$, being $N(0,1)$ the normal distribution):

$$x^i_{n+1} = x^i_n + \frac{1}{2}\left(F^i(x_n) + F^i(x^1)\right)\Delta t + \frac{1}{2}\left(G^i_j(x^2) + G^i_j(x^3)\right)\sqrt{\Delta t}\,\eta^j, \tag{16}$$

$$x^{1i}_{n+1} = x^i_n + F^i(x_n)\Delta t + G^i_j(x_n)\sqrt{\Delta t}\,\eta^j, \tag{17}$$

$$x^{2i}_{n+1} = x^i_n + G^i_j(x_n)\sqrt{\Delta t/2}\,\eta^j_1, \tag{18}$$

$$x^{3i}_{n+1} = x^i_n + F^i(x_n)\Delta t + G^i_j(x_n)\sqrt{\Delta t/2}\,\eta^j_0, \tag{19}$$

$$\eta^j = N(0,1), \tag{20}$$

$$\eta^j_0 = N(0,1). \tag{21}$$

The numerical method may be chosen according to the parameters of the Fokker–Planck equation. In a high collisionality regime a high order method in the stochastic part should be used. In low collisionality problems an order 1 method in the stochastic part combined with a fourth order Runge–Kutta (RK4) for the deterministic part can produce excellent results.

This is the case of the simulations shown in Fig. 1: we calculate the evolution of a supra-thermal population of ions in the TJ-II stellarator. We choose the initial state $f(x,t) \sim \delta(\mathbf{x} - \mathbf{x}_0)\cdot\delta(E - E_0)$, $E_0 = 1.4$ keV and calculate the average total energy $\langle E(t)\rangle$ during the thermalization process.

The problem is solved with two numerical methods: the KP method (order 2 deterministic, order 2 stochastic) and the RK4 (order 4 deterministic, order 1 stochastic). A time step of $\Delta t = 2\cdot 10^{-8}$, s is needed for KP while only $\Delta t = 2\cdot 10^{-7}$ s is needed for the RK4. In both cases 500 trajectories are integrated. The total CPU time needed is 2000 s for KP and 400 s for the RK4, making the latter more suitable for this problem.

4. Architecture

ISDEP is programmed in C to maximize performance and portability and is automatically designed to scale perfectly in distributed computing platforms such as grid or volunteer computing. Consequently, the performance in massive parallel computers is close to 100%. In this section we show the execution steps of the code together with the data analysis tools and the computing platforms employed.

4.1. Initialization

A copy of the executable and the input files are sent to each computing node, or copied into a file-system common to all nodes. The input files contain the plasma background data, the confining magnetic field and technical details (time step, numerical algorithm chosen, etc.). The first stages of the execution of the code are invested in initializing the random number generator, the magnetic field array and the trajectory itself. The magnetic field array contains all the information related to the magnetic configuration of the device. Part of this array is read from a file (ex. the magnetic field **B**) and the remaining is calculated (ex. its gradient ∇B) and stored in memory in order to save CPU time in the next steps. The interpolations in this array are linear, provided that the spatial grid is dense enough. The size of the magnetic array may be ∼400 MB, representing most of the memory that ISDEP uses.

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005

For the estimation of thermal ion transport, the trajectory is initialized according to the plasma density, locally Maxwellian in v^2 and uniform in λ. Alternatively, ISDEP can read the output of a neutral transport code, like EIRENE [8] or a plasma heating related code like FAFNER2 [9], to calculate the initial distribution.

4.2. Orbit integration

After the initialization routines, every node starts to integrate trajectories independently of each other. The statistical independence is guaranteed reading the random seed locally in the node (from `/dev/urandom` in Linux clusters). The pseudo-random numbers needed are obtained with the Parisi–Rapuano generator [10]. Then the orbits are integrated, according to Section 3, and the selected data written in a file. This is the most CPU time consuming stage.

There are two main output files in ISDEP: trajectory files (OUT.DAT) and histogram files (OUT.HIS). In the former the 5D position in phase space is stored for each trajectory at selected times. Since usually the plasma evolution is smooth in logarithmic scale, the measurement times are chosen to be approximately equidistant in logarithmic scale. The latter contains histograms of different particle quantities (energy, distribution function, rotation velocity, radial flux, ...) at the selected times. In order to increase statistics the following technique is applied in the histograms. Assuming that the evolution of the system is slow, one may take all the measurements at times $t \in (0.9\,t_0,\ 1.1\,t_0)$ belonging to t_0. In this way the statistical errors are significantly reduced.

The main particle loss mechanism in ISDEP is collisions with the vacuum vessel. The hit points of test particles are stored in a file, allowing the study of heat deposition and impurity sputtering on the vacuum vessel. Additionally, charge exchange processes can be taken into account, if the neutral density profile in the plasma is provided.

The trajectory length (i.e. the typical CPU time per particle) depends on to the fusion device studied and on the initial state of the test particles. Generally speaking, trajectories are longer for good confinement systems. For example, a supra-thermal ion from Fig. 1 takes ∼5 s (using RK4) whereas a thermal ion in ITER (see Ref. [11]) can take ∼15 min.

4.3. Background fields

We denote *background fields* the arrays containing all the physical quantities that are needed to calculate the forces in the equations of motion. There are two kinds of background fields in ISDEP: the magnetic configuration array and the 1D plasma profiles.

As we commented in Section 4.1, the magnetic configuration array is initialized in the early stages and the quantities $\rho(x), \mathbf{B}(x), \nabla B(x), \nabla \times \mathbf{b}(x), \ldots$ are interpolated when required. Even though we use linear interpolations for this task, this is the most time consuming step, representing ∼75% of the total execution time. It has been checked that the space discretization is dense enough and has no influence in the final result.

Plasma density, temperature and potential profiles depend only on the effective radius ρ. A simple linear interpolation is used, with $\rho(x)$ taken from the magnetic array.

4.4. Analysis

The output of every node is stored in one particular node and is analyzed locally. In a standard Monte Carlo calculation the average and it statistical error of a sample X_i of N events is:

$$\overline{X} = \frac{1}{N}\sum_{1}^{N} X_i, \qquad \Delta\overline{X} = \sqrt{\frac{\overline{X^2} - \overline{X}^2}{N-1}}. \tag{22}$$

For linear functions of the sample, this is the right way for error calculation. But for nonlinear functions $f(X)$, formula (22) might overestimate the statistical error. In order to compute reliably the statistical dispersions in $f(X)$, the jack-knife technique is applied [12]: let us consider a set of N independent, identically distributed vectorial random variables, $\mathbf{X}_i, i = 1, \ldots, N$, and a nonlinear function g of the expectation values $\langle\mathbf{X}\rangle$. By a vector random variable, we mean a set of M physical quantities that are measured on the very same experiment or numerical simulation: $\mathbf{X}_i = (X_i^{(1)}, X_i^{(2)}, \ldots, X_i^{(M)})$. It is possible that the components of $\mathbf{X}_i$ are statistically correlated. The problem that the jack-knife method solves is that of computing the statistical error for our estimator of $g(\langle\mathbf{X}\rangle)$.

The procedure takes care at once of two problems: (i) it treats correctly the statistical correlations among the components of $\mathbf{X}_i$ and (ii) it avoids the instabilities caused by the non-linear nature of the function g. The procedure is as follows:

We first compute the average of the random variables as

$$\overline{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{X}_i, \tag{23}$$

and construct the estimator for $g(\langle\mathbf{X}\rangle)$

$$\overline{g} = g\left(\overline{\mathbf{X}}\right). \tag{24}$$

A direct computation of the statistical error using the random variables $g_i = g(\mathbf{X}_i)$ would be unpractical (unless g is nearly a linear function). On the other hand, an error propagation computation requires to take into account the statistical correlations of the different components of $\mathbf{X}_i$. A simple alternative procedure consist of the following. First define the (not independent) random variables

$$\mathbf{X}_i^{\mathrm{JK}} = \frac{1}{N-1}\sum_{j=1;j\neq i}^{N} \mathbf{X}_j, \tag{25}$$

and

$$g_i^{\mathrm{JK}} = g(\mathbf{X}_i^{\mathrm{JK}}), \tag{26}$$

then compute the jack-knife estimate for the statistical error of $\overline{g}$

$$\Delta\overline{g} = \sqrt{(N-1)\left[\sum_{i=1}^{N}\frac{(g_i^{\mathrm{JK}})^2}{N} - \left(\sum_{i=1}^{N}\frac{g_i^{\mathrm{JK}}}{N}\right)^2\right]}. \tag{27}$$

Note that the error is proportional to the square root of the number of blocks. The number of blocks should be large enough, say 50, so this technique works properly. It is fairly easy to show that the jack-knife method provides the very same error estimate that Eq. (22), if the function g is linear on the X_i. The jack-knife method is then convenient to calculate the error of nonlinear functions of the sample, like the kurtosis of the velocity distribution. Since it is defined as $\kappa = \frac{\langle v^4\rangle}{\langle v^2\rangle^2}$, the jack-knife deals with the nonlinearity in the division and with the correlation between $\overline{v}^4$ and $\overline{v}^2$.

The reader will note that we evaluate nonlinear functions of the mean values, simply by substitution of the exact averages by the Monte Carlo mean value. Doing so we incur in a bias, which is typically of order $1/N$ (hence, negligible as compared with the statistical error $\sim 1/\sqrt{N}$).

4.5. Introduction of nonlinear terms

The physical description of the plasma implemented in ISDEP is a linear theory so, in principle, the background plasma is not modified. In particular, it means that the plasma background

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005

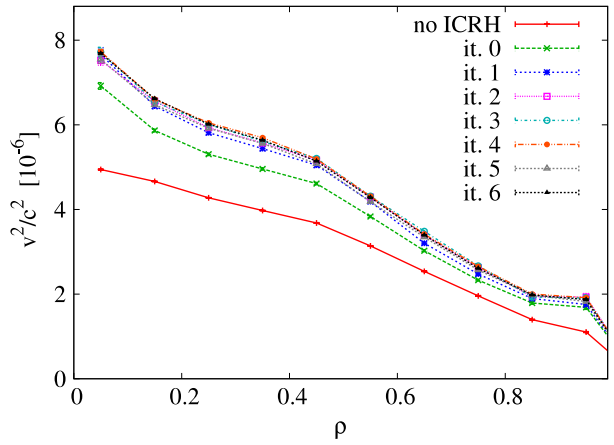

Fig. 2. Example of the iterative procedure. We modify the background temperature of a test tokamak with microwave heating. The profile converges after 6 iterations.

has infinite inertia and specific heat. In Ref. [20] we proposed a method to modify the background profiles, i.e., including some non-linearities. The procedure is to update the background with the results of the calculations. As of yet, only modifications in the background temperature have been considered, but all the background profiles could be modified iteratively. For the test particles ensemble, the temperature profile is taken to be the average kinetic energy in an interval of $\Delta\rho = 0.1$ centered in ρ at a time t: $v^2(\rho, t)$. Let q_i be the quotient of the average kinetic energy in the i-th iteration (v_i^2) and the original energy profile (v_0^2):

$$q_i(\rho, t) = \frac{v_i^2(\rho, t)}{v_0^2(\rho)}. \tag{28}$$

Then, in the iteration $i+1$ we take as temperature the initial profile, multiplied by q_i:

$$T_{i+1}(\rho, t) = T_0(\rho, t)\, q_i(\rho, t). \tag{29}$$

Since the coordinates (ρ, t) are discretized, a linear interpolation is done to obtain $T(\rho, t)$ at arbitrary position and time. We stop iterating when $T_{i+1}(\rho, t) = T_i(\rho, t)$ within errorbars, which is the final self-consistent profile. As an example, we studied the ion temperature rising in a simple tokamak model due to ion microwave heating. The effects of microwaves are introduced by including the wave particle interaction in the Langevin equations [13]. Fig. 2 shows the energy (or temperature) profile for each iteration. It can be seen that the temperature converges to a self-consistent value after 6 iterations.

4.6. Steady state calculations

The steady state of a system is a time-invariant state in which the particle and energy sources and sinks are in equilibrium with each other. In ISDEP we calculate the steady state of the test particle distribution function, using the Green function's formalism, following [14]. Let $f(x, t)$ be the distribution function of our system, t the time, x the coordinates in phase space, $\mathcal{L}$ a differential operator over f and $S(x, t)$ the source term. With this notation, the problem is expressed as:

$$\mathcal{L}\,(f(x, t)) = S(x, t). \tag{30}$$

In the case of interest $f(x, t)$ is the minority particle distribution function, $\mathcal{L}$ is the Fokker–Planck operator for the guiding center and Boozer–Kuo Petravic collision operator and the source is the continuous injection particles into the plasma. A Green function $G(x, t; x_0)$ is defined such that

$$\mathcal{L}(G(x, t; x_0)) = \delta(x - x_0)\,\delta(t), \tag{31}$$

with x_0 playing the role of initial position in the 5D phase space. Then:

$$f(x, t) = \int \mathrm{d}t_0\, \mathrm{d}x_0\, G(x, t - t_0; x_0)\, S(x_0, t_0), \tag{32}$$

because

$$\mathcal{L}\,(f(x, t)) = \int \mathrm{d}t_0\, \mathrm{d}x_0\, \mathcal{L}\,(G(x, t - t_0; x_0)) \times S(x_0, t_0) = S(x, t). \tag{33}$$

Note that the only contribution to this integral comes when $t = t_0$. In the systems studied here the source is assumed to be constant in time. This is in agreement with the linear description of the problem because the background plasma is kept constant. This technique should not be used in combination with the inclusion of nonlinear terms of Section 4.5. If we take $S(x, t) = S(x)$, then:

$$f(x, t) = \int \mathrm{d}t_0\, \mathrm{d}x_0\, G(x, t - t_0; x_0)\, S(x_0) \tag{34}$$

$$= \int \mathrm{d}t_0 \int \mathrm{d}x_0\, G(x, t - t_0; x_0)\, S(x_0). \tag{35}$$

Defining

$$H(x, t - t_0) = \int \mathrm{d}x_0\, G(x, t - t_0; x_0)\, S(x_0), \tag{36}$$

the distribution function becomes a time integral:

$$f(x, t) = \int_0^t \mathrm{d}t_0\, H(x, t - t_0) = \int_0^t \mathrm{d}t_0\, H(x, t_0). \tag{37}$$

Except for a multiplicative constant, the function $H(x, t)$ is calculated by ISDEP. Finally, with a 1D numerical integration, $f(x, t)$ can be easily found. In fact, for sufficient large times, it is expected that $f(x, t)$ is constant in time because of the equilibrium between continuous injection and particle losses (all test particles always escape from the device).

Due to its linear nature, ISDEP cannot provide absolute values of f, so the results are usually presented normalized. Nevertheless, real values can be calculated multiplying f times the incoming flux of particles.

4.7. Computing platforms

Since the required communication between nodes is none, ISDEP can run in several computing platforms: high performance computers (HPCs) and distributed platforms. The scaling with the number of nodes is, in all cases, close to linear. The HPC facilities used are the EULER cluster at CIEMAT and the BIFI cluster.

In distributed platforms there is no fast communication between the nodes. On the contrary, a huge number of nodes are available. These nodes are very inhomogeneous in performance and characteristics, so ISDEP should be as robust as possible to minimize the problems caused by this fact. Grid and volunteer computing are the main resources used (ISDEP has been ported to the Grid under EGEE II & III [15] and EGI-InSPIRE [16] projects). Fig. 3 shows the ISDEP work flow on the Grid. The volunteer computing projects Ibercivis [17] and its precursor Zivis have provided hundreds of thousands of CPU hours. Moreover, they had an important role in the divulgation of fusion science in Spain and Portugal. ISDEP was designed from the early steps to run on Grid architectures, but it had to be adapted to volunteer computing.

Ibercivis uses the BOINC libraries [18]. The BOINC version of ISDEP must fulfill some basic requirements of volunteer computing platforms. The main constriction is that the execution time of all jobs must be shorter than 30 min. This problem is overcome by

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005

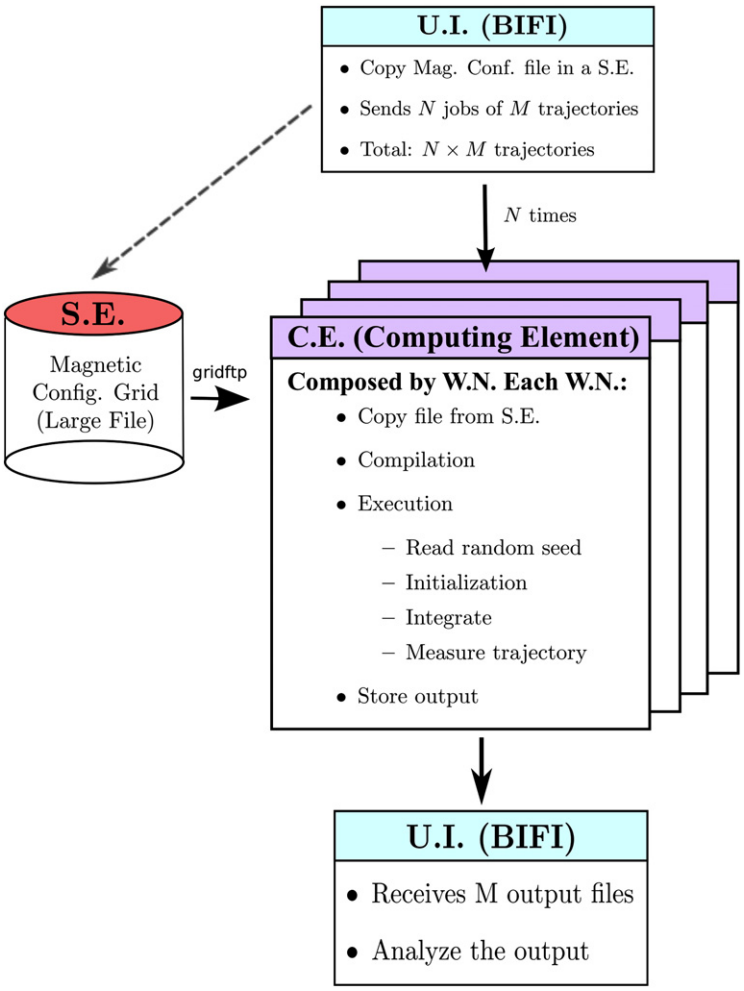

Fig. 3. ISDEP work flow on the Grid. First, the magnetic field file is copied to a Storage Element (SE) and the jobs submitted to the Computing Elements (CE), composed by Worker Nodes (WN). Each WN retrieves a copy of the magnetic field file from the SE, integrates a certain number of trajectories and compresses the result. When finished, all output files are copied back to the User Interface (UI) and then locally analyzed. The potential of the Grid relies on the huge number of WN available (around 10^4 in January 2012 in the Fusion Virtual Organization).

integrating a single trajectory per job. An important benefit of the single-trajectory-per-job strategy is that we avoid biasing our statistical sampling. Indeed, in this way we do not risk a systematic rejection of the jobs were the longest trajectories appear.

Additionally, the usage of memory and disk must be reduced as much as possible to minimize the load in the local computer. Finally, we have included some CRC (Cyclic Redundancy Check) check-sums as a basic security protocol. We assume that this check is enough to validate the results and we do not submit duplicate jobs to Ibercivis.

The choice of a computing platform depends strongly on the problem to be solved. Grid and volunteer computing are usually cheaper and more powerful than HPC for ISDEP, but they present longer latency times. As an informal rule we may say that if the simulation is longer than one day in HPC, we must use the Grid or volunteer computing. Moreover, many HPC operators do not allow to run this kind of embarrassing parallel problems.

5. Overview of the physical results

5.1. Thermal Ion transport in TJ-II

ISDEP was originally applied to the TJ-II stellarator [19]. A first work calculating the transport of thermal ions in ECRH plasmas was published in [5]. The thermal ion transport in absence of ion–electron collisions was calculated in that paper. The violation of the local ansatz, assumed in standard Neoclassical calculations, was explicitly quantified. Additionally, this work estimated the poloidal accumulation of particles, hence breaking the 1D character of the problem, and deviation of the test-particle distribution function from de Maxwellian because of the selective escape of particles. From this data, a first estimate of the ion contribution to the bootstrap current was provided.

5.2. CERC and ion confinement

A particular effect, known as the Core Electron Root Confinement (CERC) was simulated in [20]. Collisions with electrons and the inclusion of non-linear terms were implemented here with a complex workflow in the EGEE Fusion Virtual Organization. The conditions of the plasma before and after the transition to CERC were simulated. The variation of the radial electric field and the rising of the electron temperature were the ingredients required to reproduce the experimentally observed rise of the ion temperature. The typical number of iterations needed is 20 and a total of about 300 CPU-years were required.

5.3. Violation of Neoclassical ordering in TJ-II and non-diffusive transport

The Neoclassical (NC) ordering is violated in TJ-II in the long mean free path regime. In this case, the typical radial width

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005

of the orbits is larger than the characteristic plasma length, so the NC ordering is not satisfied. The standard NC tools cannot be applied with precision and, as a consequence, non-diffusive features of the radial transport in TJ-II may be found. In [21], the radial transport was estimated for ions at different radial locations and different plasma regimes. A rough estimate of the Hurst exponent (which quantifies the diffusive character of the transport) was extracted from the simulations. The local ansatz was shown to be approximately fulfilled for plasmas heated by Neutral Beam Injection (NBI), but not when heated by Electron Cyclotron Resonant Heating (ECRH). Non-diffusive features appear in the long mean free path regime.

5.4. Flux expansion divertor studies

The last paper in TJ-II included a detailed study of escape particles and their asymmetric fluxes on the vacuum chamber [22]. These asymmetries could be exploited to develop a flux expansion divertor [23]. The fluxes on the plasma wall were calculated in several divertor configurations. The toroidal and poloidal resolution available in ISDEP was a key factor in the proposal of a new divertor based on the flux expansion concept at TJ-II.

5.5. 3D transport in ITER

Most of the transport codes applied to the future tokamak ITER [24] consider axisymmetry. Since the number of toroidal coils is limited (18 in this case), all the tokamaks present a small toroidal asymmetry produced by the ripple of the magnetic field. The low collisionality of these plasmas makes that the typical size of the orbits is large enough to cast doubts to ton the standard Neoclassical calculations. Several simulations of the ion kinetic transport were done, scanning different ripple intensities [11]. The differences in the transport properties of the device were found using a simple model for the perturbation where the ripple presents a sinusoidal variation in the toroidal direction. In this work ISDEP was benchmarked with another orbit code [25].

5.6. Fast ions

Neutral Beam Injection (NBI) is a common method for plasma heating and fueling in fusion devices. A high energy beam of neutrals is launched to the plasma and is immediately ionized, giving place to a small population of fast ions in the plasma. ISDEP solved the transport for these ions, computing the steady state distribution inciting in the TJ-II and LHD [26] stellarators [27]. The fast ion source coming from NBI was calculated with the MC codes FAFNER [9] and HFREYA [28], respectively.

6. Summary and conclusions

ISDEP is a useful code that studies ion transport in fusion plasmas using a kinetic approach. It overcomes the limitations of keeping the NC ordering and it does not assume nested surfaces, toroidal and poloidal symmetries, or diffusive transport, but still keeping good computing performance. It calculates the distribution function of a minority ion population, with the possibility of estimating the steady state distribution function and including evolution of the whole plasma temperature.

The Monte Carlo method used is based on the equivalence between the Fokker–Planck and Langevin equations, transforming a partial derivative equation into a set of stochastic differential equations. This makes unnecessary the data communication between computing nodes, allowing the use of distributed architectures like grids and volunteer computing. ISDEP uses different numerical methods to solve the system of five SDEs which represent the ion motion in the confining device. The choice of the numerical method depends on the particular conditions of the problem, essentially the test particle typical speed and the collisional regime.

It has been mainly applied to the TJ-II stellarator, although it has produced results in an ITER equilibrium and in the LHD stellarator geometry, both in the time varying and the steady state regimes. ISDEP is becoming a practical tool to study fast ion dynamics in stellarators because in some experimental conditions the plasma background is almost stationary, fulfilling the requirements of the code and the Green function formalism. The code has been benchmarked and comparisons of the fast ion distribution function with experimental results in TJ-II [29] and LHD [30] are in process.

In addition, ISDEP can also deal with the ion microwave heating equations, provided the wave absorption profile. The solution of this problem in ITER is underway and will be presented in future works.

Acknowledgment

The authors acknowledge partial financial support from project ENE2008-06802/FTN, FIS2009-1248-C03.

References

[1] A.H. Boozer, Reviews of Modern Physics 76 (2005) 1071.
[2] P.E. Kloeden, E. Platen, Numerical Solution of Stochastic Differential Equations, Springer-Verlag, Berlin, New York, 1992.
[3] P. Helander, D.J. Sigmar, Collisional Transport in Magnetized Plasmas, Cambridge University Press, Cambridge, 2001.
[4] R.J. Goldston, P.H. Rutherford, Introduction to Plasma Physics, Taylor and Francis, London, 1995.
[5] F. Castejón, et al., Plasma Physics and Controlled Fusion 49 (2007) 753.
[6] A.H. Boozer, G. Kuo-Petravic, Physics of Fluids 24 (5) (1981) 851.
[7] T.S. Chen, A General form of the Coulomb Scattering Operators for Monte Carlo Simulations and a Note on the Guiding Center Equations in Different Magnetic Coordinate Conventions, vols. 0/50, Max Planck Institute für Plasmaphisik, Germany, 1988.
[8] www.eirene.de/html/relevant_reports.html.
[9] A. Teubel, J. Guasp, M. Liniers, Monte Carlo simulations of NBI into the TJ-II helical axis stellarator, Max-Plank Institut für Plasmaphysik, Garching, Germany, Tech. Rep. 4/268, 1994.
[10] G. Parisi, F. Rapuano, Physics Letters B 157 (1985) 301–302.
[11] A. Bustos, et al., Nuclear Fusion 50 (2010) 125007.
[12] D. Amit, V. Martin-Mayor, Field Theory, the Renormalization Group and Critical Phenomena, third ed., World Scientific Publishing, Singapore, 2005.
[13] F. Castejón, S. Eguilior, Plasma Physics and Controlled Fusion 45 (2003) 159.
[14] S. Murakami, et al., Nuclear Fusion 46 (2006) S425.
[15] F. Castejón, et al., Computing and Informatics 27 (2008) 261.
[16] http://www.egi.eu.
[17] D. Benito, et al., Proceedings for 2008 Ibergrid Meeting, Porto, Portugal, 2008, p. 273. www.ibercivis.net.
[18] http://boinc.berkeley.edu/.
[19] C. Alejaldre, et al., Fusion Technology 17 (1990) 131.
[20] J. Velasco, et al., Nuclear Fusion 48 (2008) 065008.
[21] J. Velasco, F. Castejón, A. Tarancón, Physics of Plasmas 16 (2009) 052303.
[22] F. Castejón, et al., Nuclear Fusion 49 (2009) 085019.
[23] R. Mangi, EPS—33rd Plasma Phys. Conference, Rome, Italy, 2005.
[24] M. Shimada, et al., Nuclear Fusion 47 (2007) S1.
[25] R. Seki, et al., Plasmas and Fusion Research 5 (2010) 014.
[26] A. Komori, et al., Fusion Science and Technology 58 (1) (2010) 1.
[27] A. Bustos, et al., Nuclear Fusion 51 (2011) 083040.
[28] S. Murakami, N. Nakajima, M. Okamoto, Fusion Technology 27 (1995) 256.
[29] R. Balbin, et al., EPS 2005 Meeting, Tarragona, Spain, D5.001, 2005.
[30] M. Osakabe, et al., Plasma and Fusion Research 5 (2010).

Please cite this article in press as: J.L. Velasco, et al., ISDEP: Integrator of stochastic differential equations for plasmas, Computer Physics Communications (2012), doi:10.1016/j.cpc.2012.04.005