Physics AUC, vol. 22, 68-76 (2012) **PHYSICS AUC**

Code details in calculation of diffusion coefficients using Decorrelation Trajectory Method

I. Petrisor and M. Negrea
Association Euratom-MEdC Romania,
Department of Physics, University of Craiova,
13 A.I. Cuza Street, 200585 Craiova, Romania

Abstract

We presented some codes details based on Runge-Kutta-Fehlberg 45 (RKF45) method for the calculation of different global Lagrangian quantities specific to Langevin equations that characterize the diffusion in tokamak plasma.

1 Introduction

A central issue for fusion is the description of turbulence phenomena in a plasma at high temperature [1]. The magnetic and the electrostatic turbulence appear as plausible candidates in order to explain some features of a hot magnetized plasma. The intensity of the magnetic and the electrostatic fluctuations is measured by specific dimensionless Kubo numbers. Trapping phenomena in fusion plasma models are studied also by the decorrelation trajectory method [2]-[5]. In this paper we present some information about the numerical codes used in the calculations related to the physical models (see e.g. [6]-[11]) based on decorrelation trajectory method. Because these numerical codes have considerable length we will only give information about their structure, the main idea and short information about the parallelisation scheme used for the determination of the diffusion coefficients.

2 Diffusion of electrons in an electromagnetic stochastic field

In the case of the study of the electron diffusion in a electromagnetic field (a model analyzed in the paper [7]) we numerically evaluated the Lagrangian correlation tensor, the running diffusion tensor and also the asymptotic diffusion tensor using an optimized numerical code also based on Runge-Kutta-Fehlberg 45 (RKF45) method like in [12]. With this code, a large enough number of decorrelation trajectories [i.e. solutions of the following system of equations given by (2) and (3) in different subensembles S] with the initial condition

$$\mathbf{x}^S(0) = \mathbf{0} \tag{1}$$

are calculated.

$$\frac{dx^S(\tau)}{d\tau} = Kv_x^S\left(\mathbf{x}^S(\tau)\right)B(\tau) \equiv w_x^S\left(\mathbf{x}^S(\tau),\tau\right) \tag{2}$$

$$\frac{dy^S(\tau)}{d\tau} = x^S(\tau)K_s + Kv_y^S\left(\mathbf{x}^S(\tau)\right)B(\tau) \equiv w_y^S\left(\mathbf{x}^S(\tau),\tau\right) \tag{3}$$

For each Lagrangian correlation tensor an enough number of decorrelation trajectories were considered - between $31^3 - 35^3$. For each trajectory, a non-uniformly range of time, with up to 200 points, was used. The final integration time was chosen to be in the range $[10,\ 15]$ in order to obtain the asymptotic regime. From the numerical point of view, this problem is practically equivalent to the calculation of the three-dimensional integral

$$I(\theta) = \iiint dx\ dy\ dz\ e^{-\frac{1}{2}(x^2+y^2+z^2)} f(x,y,z,\theta)$$

This integral is specific to our calculations because the global Lagrangian correlation tensor is:

$$L_{ij}(\theta) = (2\pi)^{-\frac{3}{2}} \int d\varphi^0 \int dv_x^0 \int dv_y^0 \times \tag{4}$$
$$\times \exp\left\{-\frac{1}{2}\left[\left(v_x^0\right)^2 + \left(v_y^0\right)^2 + \left(\varphi^0\right)^2\right]\right\} L_{ij}^S\left(\theta, v_x^0, v_y^0, \varphi^0\right)$$

while the running diffusion tensor is given by:

$$D_{ij}(\tau) = \int_0^\tau L_{ij}(\theta)\, d\theta \tag{5}$$

After the calculation of three-dimensional integral from equations (4), an additional integral over time must be done (5) in order to obtain the running diffusion tensor components. In the equations given in (4), the exponential function allows us to take relatively small integration limits [see below the equations (12-14)].

In order to do the numerical simulation (the complementary study of the DCT method) the parallel computer named ASTER from ULB-VUB, Bruxelles, Belgium was used. This computer can deal with large matrix (e.g. 4096×4096) for the discrete representation of the electrostatic stochastic field. The spectrum $S(k)$ was taken like an isotropic one and it was also defined using the results from experimental measurements of the drift turbulent regimes. The spectrum was constructed by taking into account the parameters specific to the DCT method. We used a spectrum, given by a combination between a Gaussian spectrum (used for small wave vectors) and a power function spectrum: k^{-3} (used for large wave vectors). Using this shape of the spectrum of the stochastic electrostatic potential we were able to make a direct comparison between the results obtained by the two methods, i.e. the numerical simulations and the DCT method. Usually, in the framework of numerical simulations some mean values are evaluated, and there are real difficulties to obtain an correct equivalence between the DCT results and those corresponding to the numerical simulations. We have "generated" a structure for the electrostatic stochastic field, using as much as possible large matrix for the ASTER computer and considering the behavior of the two-point correlation of the stochastic field specific to the DCT case, i.e. a decreasing one to zero, like a negative exponential function without oscillations.

We employed like integrator, a specific one to a Runge-Kutta 4 method (RK4), and from the statistical point of view a relatively large number of cells: 96 $\times$ 8, each one containing 64 trajectories. All these trajectories from all the cells have contributed to the construction (calculation) of the mean squared displacement in both radial and poloidal directions and consequently to the calculation of the diffusion tensor components. A lot of values for the Kubo numbers were used; for each pair (K^{sim} and K_s^{sim}) of each run we have used about 96×35 minutes for each monoprocessor, i.e. about 56 monoprocessor-hours (we have used a parallelisation scheme for $4-8$ monoprocessors). On the contrary, for the DCT method, the average time for any run was 10 time smaller than that corresponding to numerical simulation. The total calculation time on the ASTER computer was about 2300 monoprocessor-hours.

3 Diffusion coefficients in the case of zonal flow generation

In the framework of DCT applied in the zonal flow cases [[8], [11]], we provide here short information about the numerical aspects. We have implemented the Runge-Kutta-Fehlberg 45 method (RKF45) in order to evaluate each trajectory (it was numerically integrated the corresponding system of equations for each decorrelation trajectory). We developed a compatible Fortran 77 code, for a parallel computer in order to calculate $5-$dimensional numerical integrals (using the information/ numerical data from a large number of trajectories). From numerical point of view, the $5-$dimensional numerical integrals are equivalent to 4×4 Lagrangian correlation tensor of the directly fluctuating velocities (we have here a coupled effect in both physical and wave vector spaces) and each one-dimensional integral (from the $5-$dimensional numerical integral) was approximated with a sum of an odd number of terms ($N = 19 \div 27$, but sometimes up to 41!). After the realization of the optimization and fixing of an acceptable numerical error (like $3-6\%$) we have considered for each trajectory an unequal distributed points (400 points) for the time range. The final time was considered $\theta_{final} = 15$ in order to gain the asymptotic regime.

In the following, we give the libraries included in the parallel code (called "dctzf.f") in order to obtain an exe file (called "ddd") under UNIX OS:

$$\text{f90 -g -o ddd dctzf.f -lmpi -lpmpi -lelan -lmpija} \tag{6}$$

The method used for the parallelisation of the code (the transformation of a single processor code into a parallel code) was done by suppressing the loop corresponding of the 5th numerical integral. Thus, we allocated N processors for the calculation of the integral, practically each processor performing the calculation of a single term of the sum that contain N terms used in the approximation of the last numerical integral. Each processor has performed 4 algorithmic loops, the total number of trajectories was N^4. The total time for the calculation was reduced using a number of N monoprocessors. The quantity of the data performed by each monoprocessor was almost the same, and each monoprocessor needed almost the same time to finish the calculation. After this stage was used a single processor (using MPI-GATHER) to collect all the data to finish all the calculations and also to write the external final files with the needed results.

We display below the main part of the script file (this file was called e.g. "zf_ex.txt") used for the running (in "in batch" mode). This file contains a lot of information about

the total number of the monoprocessors used for the calculation, the maximum time for running, the type of the required errors, in what manner are wrote the external files, where we have stored the external files, etc.

.....

The content of the file "zf_ex.txt":

```
#BSUB -n 23
#BSUB -c 20:00
#BSUB -J testZFDCT27
#BSUB -e ERREUR.%J
#BSUB -o SORTIE.%J
..
cd $TMPDIR
cp /home/p570jmi/ddd .
prun -s -t -v -n 23 ./ddd
cp Ljj.txt /home/p570jmi/.
cp Ljk.txt /home/p570jmi/.
cp Ljx.txt /home/p570jmi/.
cp Ljy.txt /home/p570jmi/.
cp Lkj.txt /home/p570jmi/.
...
cp Dyk.txt /home/p570jmi/.
cp Dyx.txt /home/p570jmi/.
cp Dyy.txt /home/p570jmi/.
cd
pwd
ls -al
```

.....

An example of the script file used to run a compiled file (called "ddd") that include the directions for the parallel computer in order to use 27 monoprocessors (for the parallel computer called CHROME):

```
#QSUB -q prod
#QSUB -lT 180000
#QSUB -n 27
#QSUB -r test_NP_27
#QSUB -eo
cd $TMPDIR
cp /home/ip/ddd .
prun -t -v -n 27 ddd
...
cp Dyy.txt /home/ip/.
cd
pwd
ls -al
```

.....

For some simple verification in direct way we used some directive like the following:

$$\text{prun -I -t -v -n 5 ddd} \tag{7}$$

this means that only 5 monoprocessors were employed for the compiled file "ddd" in order to follow some errors or warnings.

After the calculation of the Lagrangian tensors, an extra integral was calculated: the time integral, thus the running diffusion tensor. The name of the code was $\mathcal{DCT}-\mathcal{ZF}$ ($\mathcal{D}e\mathcal{C}$orrelation $\mathcal{T}$rajectory method for $\mathcal{Z}$onal $\mathcal{F}$low structures). All the numerical calculation were done using the parallel computer CHROME (the project *P570*) and the initial tests, the optimization and the initial parallelisation was performed on the parallel computer named IXIA. All these efforts were done with the collaboration of the CEA Cadarache France, using the DENEB computer (DRFC).

We give bellow the first part of the code with the parallelisation scheme:

```
       Program DCTZONAL_2_7F

      implicit real*8 (a-h,k,j,l,o-z)
c-----7  MPI==>      presence of    mpif.h
          include 'mpif.h'
C-----7--------------------------------------------------------------====72
c-----7  MPI==>  see comments below for the roles of aMPIf1**  and GGL**

      dimension aMPIf1Lxx(2960),aMPIf1Lxk(2960),aMPIf1Lkx(2960),
     &aMPIf1Lkk(2960)
      dimension aMPIf1Lyy(2960),aMPIf1Lyj(2960),aMPIf1Ljy(2960),
     &aMPIf1Ljj(2960)
      dimension aMPIf1Lxy(2960),aMPIf1Lyx(2960),aMPIf1Lxj(2960),
     &aMPIf1Ljx(2960)
      dimension aMPIf1Lyk(2960),aMPIf1Lky(2960),aMPIf1Lkj(2960),
     &aMPIf1Ljk(2960)

      dimension GGLxx(180560),f2Lxx(2960,61),f3Lxx(2960,61)

.........................

.........................
      common/iu/ xv(2960),yv(2960),tc(2960),dtet,kxv(2960),kyv(2960)
      common/iu/ vxv(2960),vyv(2960),wxv(2960),wyv(2960)
      common/iu1/ Delta,D,L0,J0x,J0y,k0x,k0y,F0x,F0y
c-----7MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM72
C=====7===========         INITIALISATION MPI    ======================72
C=====7        number of PE  = nsize    must be ODD   and >=5     ====72
C=====7              0 <=    irank   <=  nsize-1                  ====72
C-----7--------------------------------------------------------------====72
          call MPI INIT(ierr1)
          call MPI COMM RANK(MPI COMM WORLD,irank,ierr2)
          call MPI COMM SIZE(MPI COMM_WORLD,nsize,ierr3)
                tt1=MPI WTIME()
c-----7MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM72
CC             write(*,2100) tt1 ,    ierr1,   ierr2,ierr3
           IF(MOD(nsize,2).eq.0)then
           write(*,2200)nsize,nsize-1
               nsize=nsize-1
               if(nsize.lt.5)GOTO 999
         ENDIF
 2100 format('after INIT MPI T=',F18.3,' 0=',I3,' 0=',I3,' 0=',I3)
 2200 format('YOU MUST CHOICE AN ODD NUMBER OF PE >5',I4,' reduced ',I4)
c-----7++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++72
c-----7++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++72
                npmax=(nsize-1)/2
c-----7++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++72
      arx=0.8
      ary=0.2

          k0x = dsqrt(arx)
          k0y = dsqrt(ary)
```

Initially, about 180 runs were done, for different combination of the Kubo numbers and different initial wave vectors. The total calculation time was 1200 monoprocessor hours for the parallel computer. An extra credit was used, of 1200 calculation hours. For this problem, we done more than 300 runs.

4 Diffusion of stochastic sheared magnetic field lines

In order to evaluate the Lagrangian correlation tensor for "The magnetic field line diffusion" problem [6] we used a numerical code based on a modified Euler method for the calculation of the solutions of the differential system of equations specific to the magnetic field line diffusion. Next, we implemented a Runge-Kutta-Fehlberg 45 (RKF45) method [12] in order to obtain the code necessary to calculate the Lagrangian correlation tensor. This code can be easily runned on a regular PC. An enough large number of decorrelation

trajectories ($120 \times 31 \times 31$) in order to evaluate (finally) the solutions for the Langevin equations (8)

$$\frac{dx(z)}{dz} = K_m b_x[x(z), y(z), z] = \\ = K_m \left.\frac{\partial \psi(x,y,z)}{\partial y}\right|_{\mathbf{x}=\mathbf{x}(z)}$$

$$\frac{dy(z)}{dz} = K_m b_y[x(z), y(z), z] + K_s x(z) = \\ = -K_m \left.\frac{\partial \psi(x,y,z)}{\partial x}\right|_{\mathbf{x}=\mathbf{x}(z)} + K_s x(z) \tag{8}$$

The main necessary quantities are given in the equations (9), (10) and (11) and were considered in our calculations.

The Lagrangian correlation of the directly fluctuating parts from Eqs.(8) is defined as usual as:

$$L_{ij}(z) = K_m^2 \left\langle b_i(\mathbf{x}(0);0) b_j\left[\mathbf{x}(z);z\right]\right\rangle \tag{9}$$

where $\langle ... \rangle$ denotes the ensemble average over the realizations of the fluctuating magnetic field components and $\mathbf{x} = (x, y)$. The running diffusion coefficient is calculated from (9) as:

$$D_{ij}(z) = \int_0^z d\zeta L_{ij}(\zeta) \tag{10}$$

provided that the stochastic field is "stationary"; the asymptotic diffusion coefficient is calculated as the following limit:

$$D_{ij}^{as} = \lim_{z\to\infty} D_{ij}(z) \tag{11}$$

From the mathematical point of view, the calculus of the Lagrangian correlation involves the numerical evaluation of a three-dimensional integral, such as:

$$I = \iiint dx\ dy\ dz\ f(x,y,z) = \int_{x_1}^{x_2} dx \int_{y_1(x)}^{y_2(x)} dy \int_{z_1(x,y)}^{z_2(x,y)} dz\ f(x,y,z) \tag{12}$$

In order to make as much as possible a quasi-correct calculation of a three-dimensional integral (12), we must fix the integration limits of each single-integral from the multi-dimensional integral. We will firstly fix the limits x_i, and then the functions $y_i(x)$, $z_i(x,y)$, where $i = 1, 2$. Also, a careful analysis of the order of magnitude of the integrand will allow us for an optimization of the numerical calculation. In the analysis it is also important to analyze the shape of the function $f(x,y,z)$. In the current problem, the function $f(x,y,z)$ is practically a product between a polynomial b_j^S $(j = x, y)$ and an exponential function $\exp\left[-\frac{\left(\psi^0\right)^2+\left(b^0\right)^2}{2}\right]$ [see below the Eqs. (15)].

The presence of the exponential like $e^{-\frac{1}{2}(x^2+y^2)}$will influence the choice of the limits of the integration. Usually, from the numerical point of view, the polynomial function

that multiplies the exponential function is of order of unity, thus we will find a precise estimation of the integral like $\iiint dx\ dy\ dz\ e^{-\frac{1}{2}(x^2+y^2+z^2)}$ or $\iint dx\ dy\ e^{-\frac{1}{2}(x^2+y^2)}$. It is well known that a Poisson integral with infinite integrations limits, has the value:

$$\int_{-\infty}^{\infty} dx\ e^{-\frac{1}{2}(x^2)} = \sqrt{2\pi} \simeq 2.507 \tag{13}$$

From the numerical point of view we evaluate an integral like a sum, and because the integrand from (13) is of the form $e^{-\frac{1}{2}(x^2)}$, we can consider the limits of integration to be ± 4 instead of $\pm\infty$; this is a consistent good simplification in our case! We will approximate the above integral with a sum that contains a number of maximum 31 terms. We divided the limits defined in the range $[-4, 4]$ into a number of pairs, non-uniform distributed (in order to catch the maximum of the Gaussian). In the same way, we will evaluate the next two-dimensional integral

$$\begin{aligned}
&\iint dx\ dy\ e^{-\frac{1}{2}(x^2+y^2)} \simeq \\
&\simeq \int_{-p}^{p} dx \int_{-\sqrt{p^2-x^2}}^{\sqrt{p^2-x^2}} dy\ e^{-\frac{1}{2}(x^2+y^2)}, \qquad p = 4 \\
&\int_{-p}^{p} dx \int_{-\sqrt{p^2-x^2}}^{\sqrt{p^2-x^2}} dy\ e^{-\frac{1}{2}(x^2+y^2)} \simeq \\
&\simeq \sum_{i\ =\ -p}^{p} \Delta x_i \sum_{j\ =\ -\sqrt{p^2-(\Delta x_i)^2}}^{\sqrt{p^2-(\Delta x_i)^2}} \Delta y_j\ e^{-\frac{1}{2}((\Delta x_i)^2+(\Delta y_i)^2)}
\end{aligned} \tag{14}$$

where Δx_i is a small interval (we stressed that we have non-uniform distribution between $[-p,\ +p]$, and also Δy_j - a non-uniform distribution). For further details see [12].

After the evaluation of two-dimensional integral, over ψ^0 and b^0, we calculate the numerical integral over the angle α. For the latter we have chosen a number of 120 points. In order to calculate the Lagrangian correlation tensor we used approximately $120 \times 31 \times 31$ decorrelation trajectories (at each run); for each trajectory we have used 200 points for the "time" variable z, and the final integration time (z-max) was chosen to be equal to 10; this is considered a safe value in order to reach the asymptotic regime. Integrating the equations (15) over z we have obtained the components of the running diffusion tensor components D_{ij} (remember that z plays the role of the time).

$$\begin{aligned}
L_{xx}(z) = (2\pi)^{-3/2} K_m^2 \int_0^{2\pi} d\alpha \int_{-\infty}^{\infty} d\psi^0 \int_0^{\infty} db^0 \{ \left(b^0\right)^2 \\
\cos\alpha \exp\left[-\frac{\left(\psi^0\right)^2 + \left(b^0\right)^2}{2} \right] b_x^S \left[\mathbf{x}^S(z); z \right] \}
\end{aligned}$$

$$L_{xy}(z) = (2\pi)^{-3/2} K_m^2 \int_0^{2\pi} d\alpha \int_{-\infty}^{\infty} d\psi^0 \int_0^{\infty} db^0 \{ \left(b^0\right)^2 \cos\alpha \exp\left[-\frac{\left(\psi^0\right)^2 + \left(b^0\right)^2}{2} \right] b_y^S \left[\mathbf{x}^S(z); z \right] \}$$

$$L_{yx}(z) = (2\pi)^{-3/2} K_m^2 \int_0^{2\pi} d\alpha \int_{-\infty}^{\infty} d\psi^0 \int_0^{\infty} db^0 \{ \left(b^0\right)^2 \sin\alpha \exp\left[-\frac{\left(\psi^0\right)^2 + \left(b^0\right)^2}{2} \right] b_x^S \left[\mathbf{x}^S(z); z \right] \}$$

$$L_{yy}(z) = (2\pi)^{-3/2} K_m^2 \int_0^{2\pi} d\alpha \int_{-\infty}^{\infty} d\psi^0 \int_0^{\infty} db^0 \{ \left(b^0\right)^2 \sin\alpha \exp\left[-\frac{\left(\psi^0\right)^2 + \left(b^0\right)^2}{2} \right] b_y^S \left[\mathbf{x}^S(z); z \right] \} \tag{15}$$

The extension of this calculation method for the anisotropic case is straightforward [[10]]. From numerical point of view, we replace the integration over the angle taking into account as variables for the integration, the components of the fluctuating magnetic field. Thus, we have evaluated the following three-dimensional integrals

$$I(z) \simeq \int_{-\infty}^{\infty} d\psi^0 \int_{-\infty}^{\infty} db_y^0 \int_{-\infty}^{\infty} db_x^0 \{ b_x^0 \exp\left(-\frac{\left(\psi^0\right)^2 + \left(\Lambda^{-1} b_x^0\right)^2 + \left(b_y^0\right)^2}{2} \right) b_x^S \left[\mathbf{x}^S(z); z \right] \} \tag{16}$$

Using a final integration of the Lagrangian tensor over z we will obtain the running diffusion tensor components D_{ij} and finally the asymptotic diffusion tensor.

5 Conclusion

In this paper we presented the main ideas about some numerical codes used in few already published papers. Some mathematical aspects were presented briefly in order to figured out the correct choices of the input in numerical scheme. Also, the optimization of any numerical code impose mainly to apply some proper mathematical algorithms only in relevant range of numerical parameters. Therefore, a simple mathematical analysis of integrands of Lagrangian tensors (or diffusion tensors) allows us an important cut of multidimensional numerical integrals and use the computational resources only in relevant domain. Some aspect of parallelisation scheme of this particular class of numerical problems we presented.

References

[1] Balescu R, **Aspects on Anomalous Transport in Plasmas**, 2005

[2] Vlad M, Spineanu F, Misguich J H and Balescu R 1998 *Phys. Rev. E* **58** 7359

[3] Vlad M, Spineanu F, Misguich J H and Balescu R 2000 *Phys. Rev. E* **61** 3023

[4] Vlad M, Spineanu F, Misguich J H and Balescu R 2001 *Phys. Rev. E 63* 066304

[5] Vlad M, Spineanu F, Misguich J H and Balescu R 2002 *Nucl. Fusion* **42** 157

[6] Negrea M, Petrisor I and Balescu R 2004 *Phys. Rev. E* **70** 046409

[7] Petrisor I, Negrea M and Weyssow B 2007 *Phys. Scripta* **75** 1

[8] Balescu R, Petrisor I and Negrea M 2005 *Plasma Phys. Control. Fusion* **47** 2145

[9] Balescu R 2003 *Phys. Rev. E* **68** 046409

[10] Negrea M, Petrisor I and Weyssow B 2007 *Plasma Phys. Control. Fusion* **49**, 1767

[11] Negrea M, Petrisor I and Weyssow B 2008 *Phys. Scripta* **77** 055502

[12] Forsythe G E, Malcolm M A and Moler C B 1977 **Computer Methods for Mathematical Computations** (Englewood: Prentice-Hall); Watt H A and Shampine L F 1977 **Subroutine RKF45 in Computer Methods for Mathematical Computations** (Englewood: Prentice-Hall)