**Variables “guiding center” for electrons**

Instead the “old” set of canonic pairs of variables  for electrons let us the “new” set of guiding center pairs :



Function :

‘**fromGuidingCenter**’

and vice versa:



Function :

‘**toGuidingCenter**’

In these expressions, the Larmor radius  is used (function : ‘**rLarmor**’).

Total Hamiltonian in “old” variables is as follows:







and in an “new” variables:





Part  of total Hamiltonians describes the collision electron and ion. Due to this collision, the momenta of both particles are changed. If  is a time of this interaction, then in the “old” variables (function : ‘**MapZ\_C**’ from function ‘**apply\_MapZ**’ )



and in an “new” variables



**Variables action-angle for electrons (I don’t know how reproduce that!)**

Let replace the canonic pair of the “fast” variables with pair of the “slow” variables . Phase  does not affect the dynamics of the collision and action  is (formulas (8) from [1]; function : ‘**actionJ**’)



Hamiltonian in this variable equals (function : ‘**H\_gc**’)



 (10a) from [1]; function: ‘**H\_gc\_0**’ 

Then the changing of momenta due to collision are as follows: if



then



**Code ‘mcool2.py’ (description)**

Indices:



Vectors of coordinates.

Ion: 

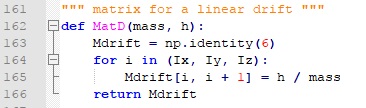
Electron: 

Electron (guiding center): 

Electron (phase-action): 

Functions:



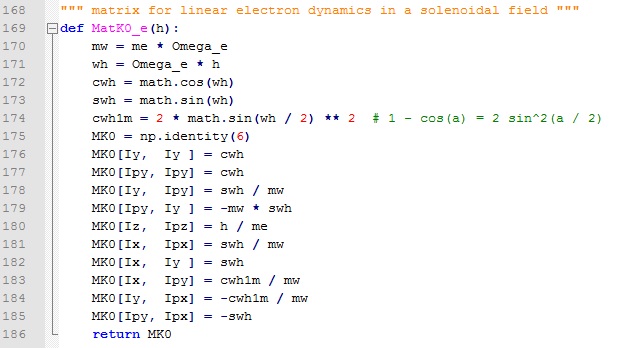




This matrix describes the changing of coordinates through drift gap:





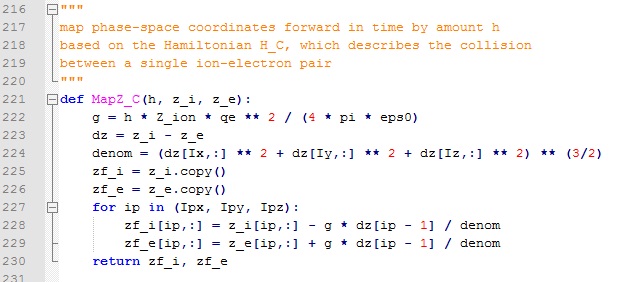




This matrix describes the motion of the electron in the longitudinal magnetic field **(It seems me that these are very strange transformation!)**:

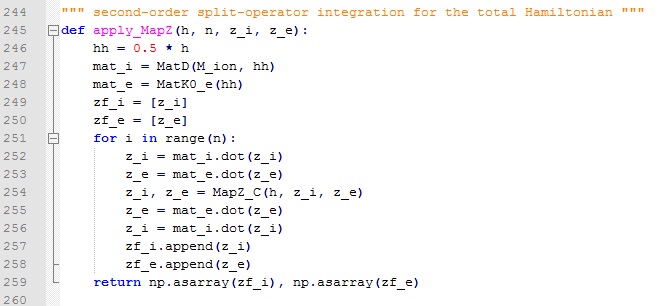












This function realizes the following transformation for  time steps with duration each:



Here  describes drift transformation for ion with matrix  and practically the same for coordinates of guiding center of electron:

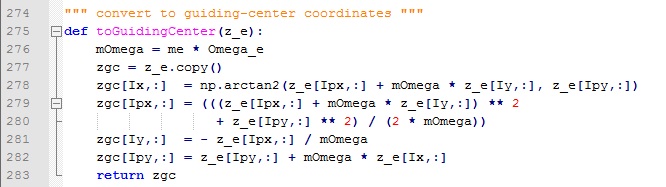


matrix  describes collision of the ion with an electron. Then for each time step the transformation is



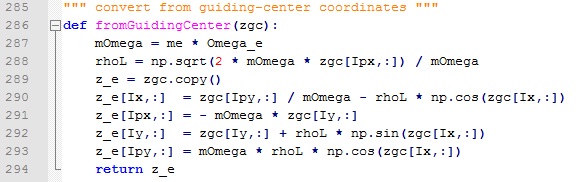
Functions returns arrays  for each of the time steps.





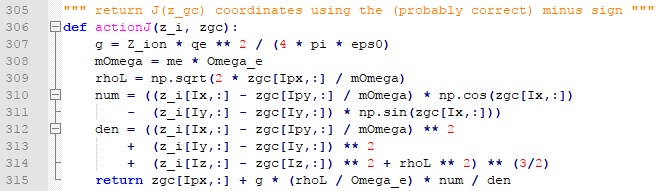






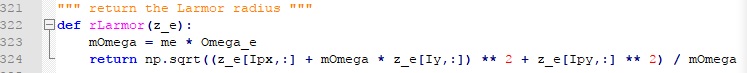






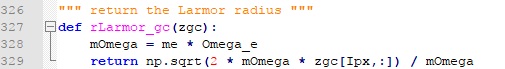






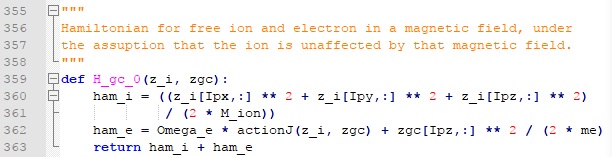






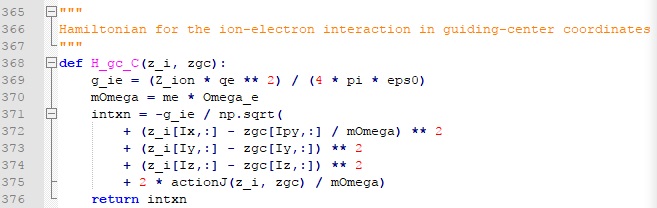






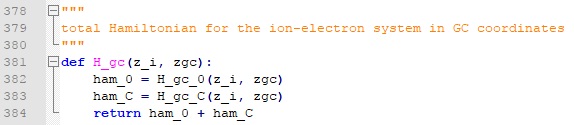






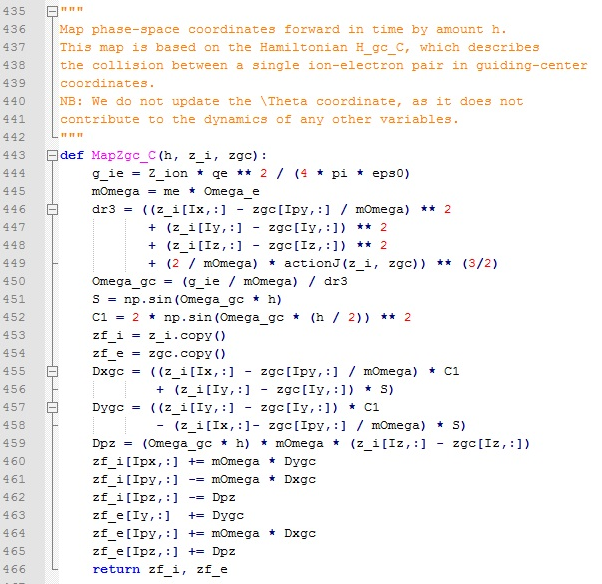








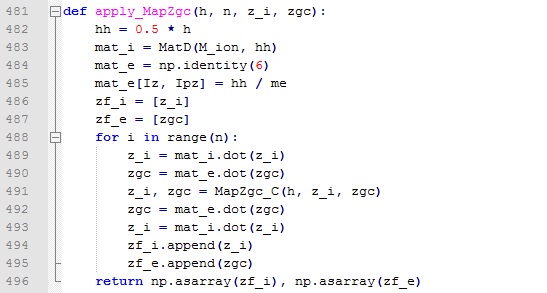




**It seems me that this is very strange transformation (It contradicts the formulas (11a-11c) from [1]; I cannot reproduce it!):**







This function realizes the following transformation for  time steps with duration each:



Here  describes drift transformation for ion with matrix  and practically the same for coordinates of guiding center of electron:

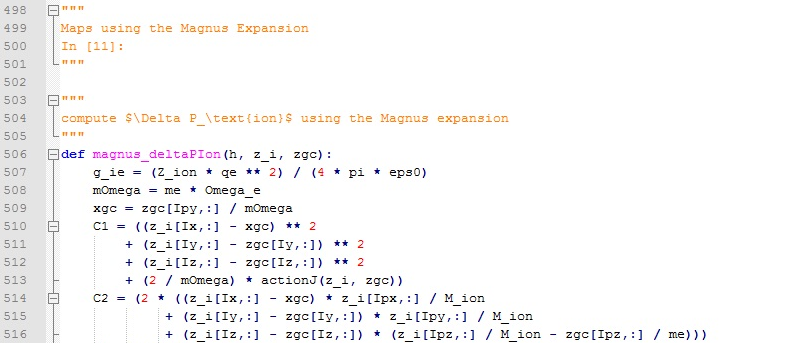


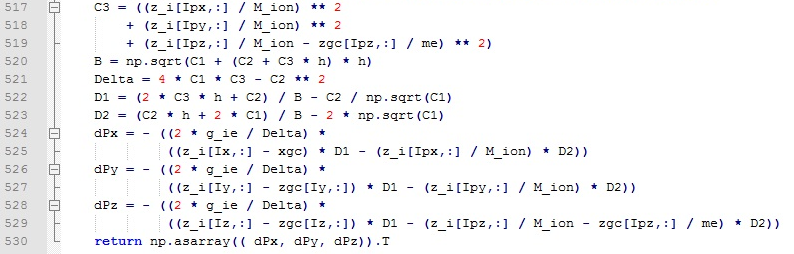
matrix  describes collision of the ion with an electron. Then for each time step the transformation is



Functions returns arrays  for each of the time steps.





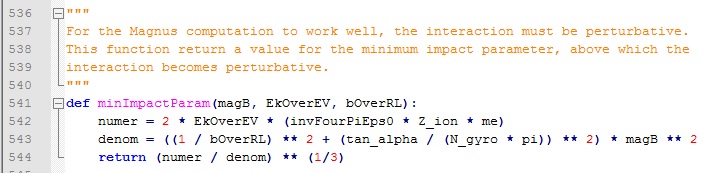


**I cannot reproduce that:**



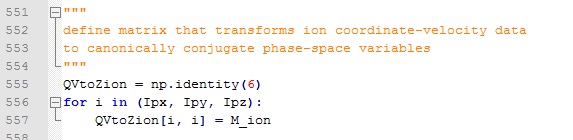
Function returns vector .

Input: longitudinal magnetic field, ,maximal perturbation ratio, enforce adequate averaging, an adequate opening angle; output minimal impact parameter.



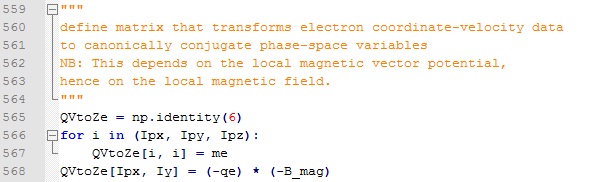








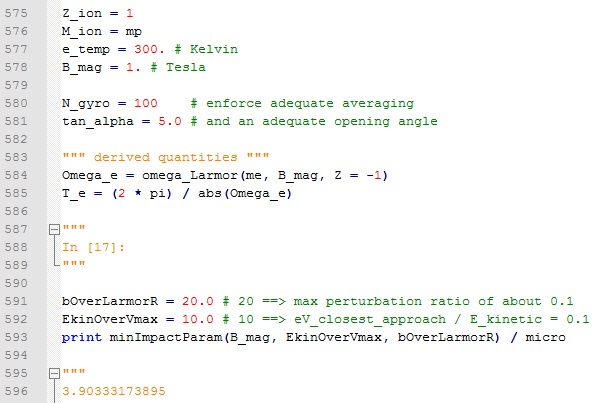


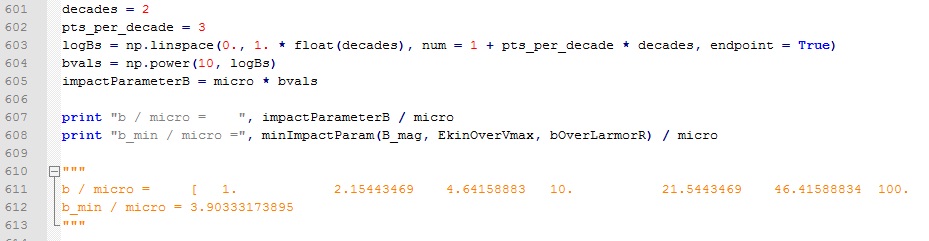


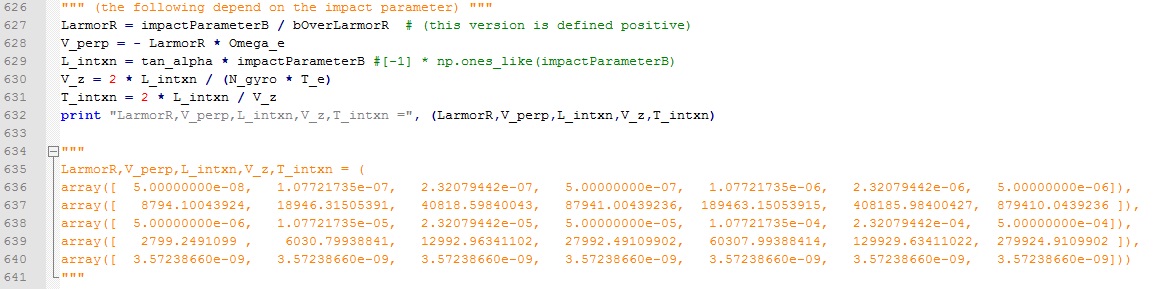


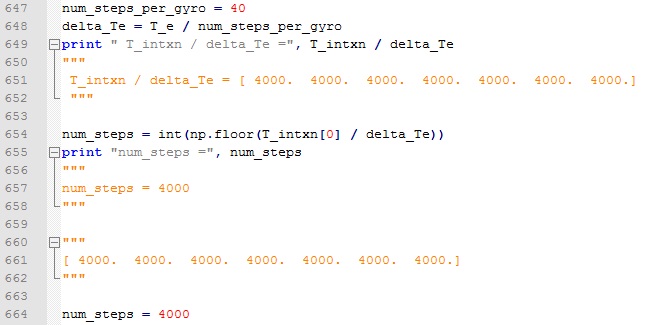
**Code ‘mcool2.py’ (simulation)**

Constants:

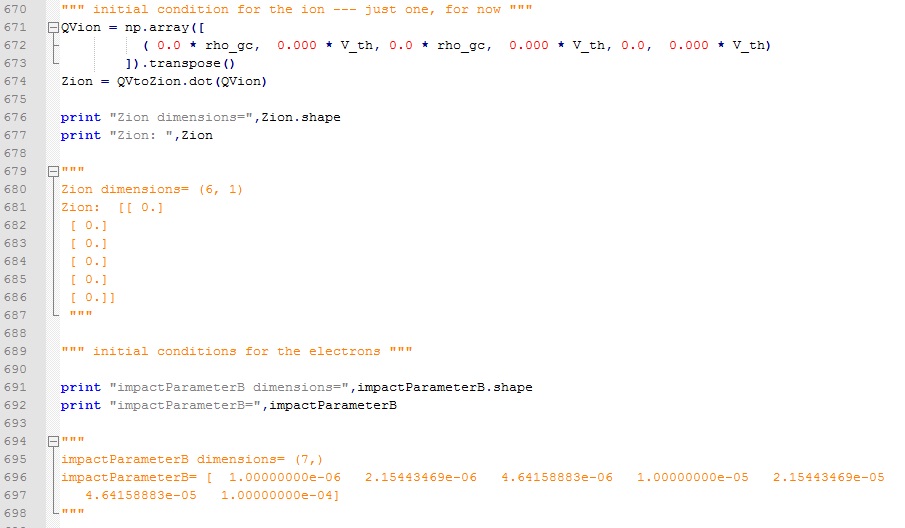


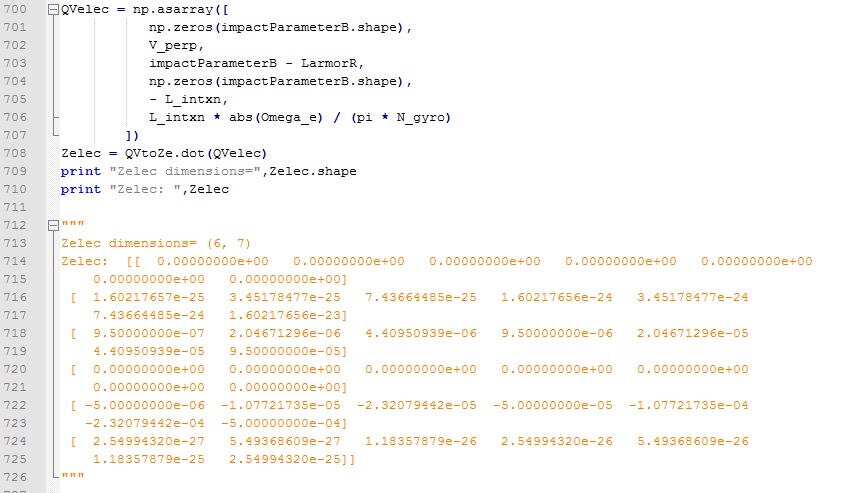












References

1. D.L. Bruhwiler, S.D. Webb. *New Algorithm for Dynamical Friction of Ions in a Magnetized Electron Beam.* AIP Conf. Proc. **1812**, 050006 (2017). <http://aip.scitation.org/doi/abs/10.1063/1.4975867>.
2. David Bruhwiler, Stephen Webb, Dan T. Abell. *A New Approach to Calculating Dynamical Friction for Magnetized Electron Cooling.* Presented at HSC Section Meeting, CERN (Hadron Synchrotron Collective effects), 24 April 2017, Geneva.