Description of the scripts

*‘threeApproachComparison\_v4.py’* and *‘threeApproachVisualization\_v0.py’*

Let us recall some problems which were solved before writing this code to compare different approaches to calculating the momentum transfer in the collision of electrons with ions (there are *notesToChoiceCodeParameters.docx*, *mapOfMaximalImpactParameter.docx* and *kinematicOfMagnetizedElectron.do*cx in github repository).

The first of these is how to form a set of the tracks of the electrons so that you have enough data to construct a 3D surface of the investigated parameters in a coordinate system of :



As shown in *notesToChoiceCodeParameters.docx*, a simple "scan" over the range of possible values of the impact parameter  and electron velocities  gave a set of separate tracks on the plane with large "voids" between them. In the same document, another approach is proposed. The "scanning" of the "coordinates"  in the required intervals with a sufficiently small step determines the initial value of the relative transverse velocity  of the electron, as the solution of the corresponding cubic equation:

 where 

The found value of the relative transverse velocity  is used for sequential calculation of the initial parameters  of the track. This approach provided a "dense" filling of the plane  with tracks of the electrons.

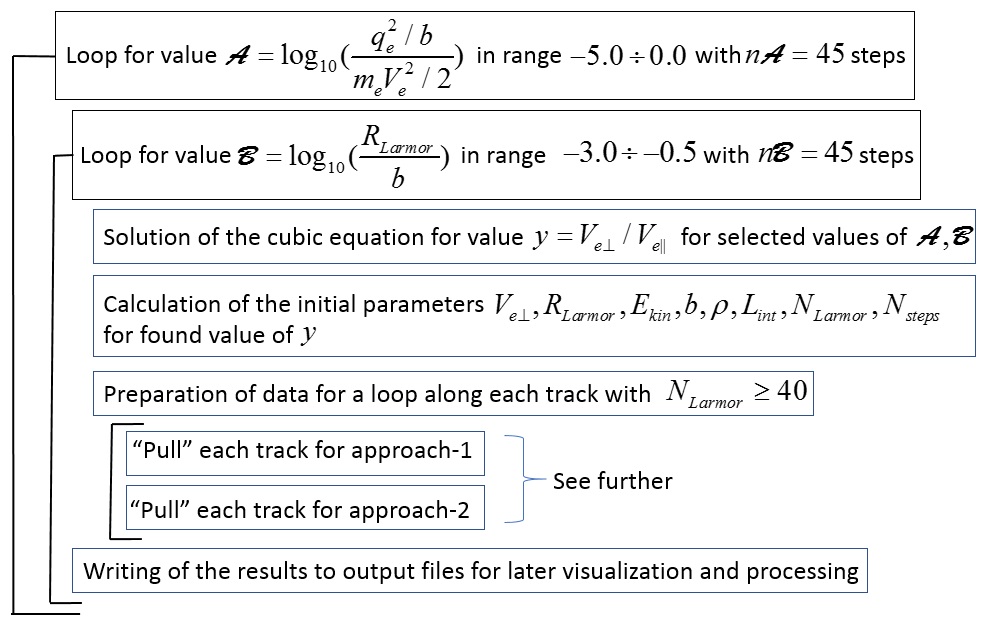
The next question is to choose reasonable values for the values of the initial impact parameter  and longitudinal part  of the trajectory of the electron. In the code, they are selected as follows:

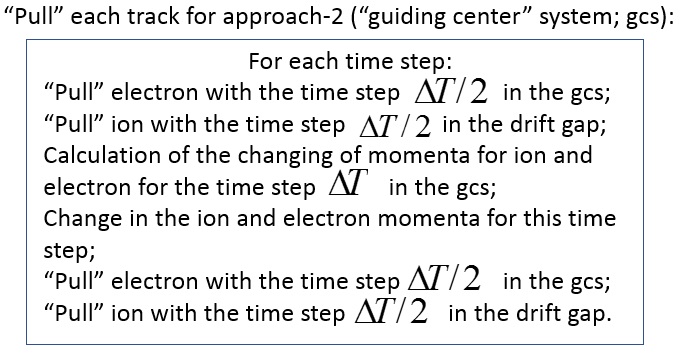
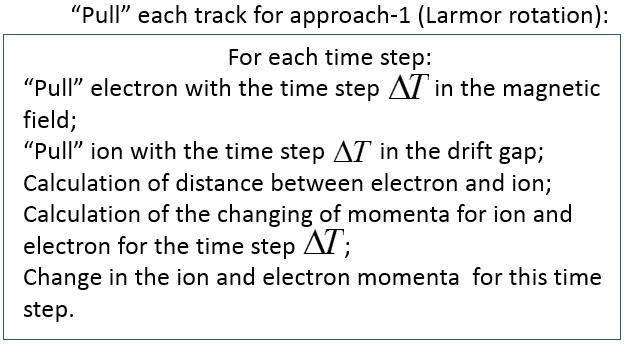


Here  is so named critical impact parameter for which the electrons are magnetized. Naturally, this selection must satisfy to demand that for these values of parameters the ion field acting on this passing electron is not yet screened by the other electrons of the beam.

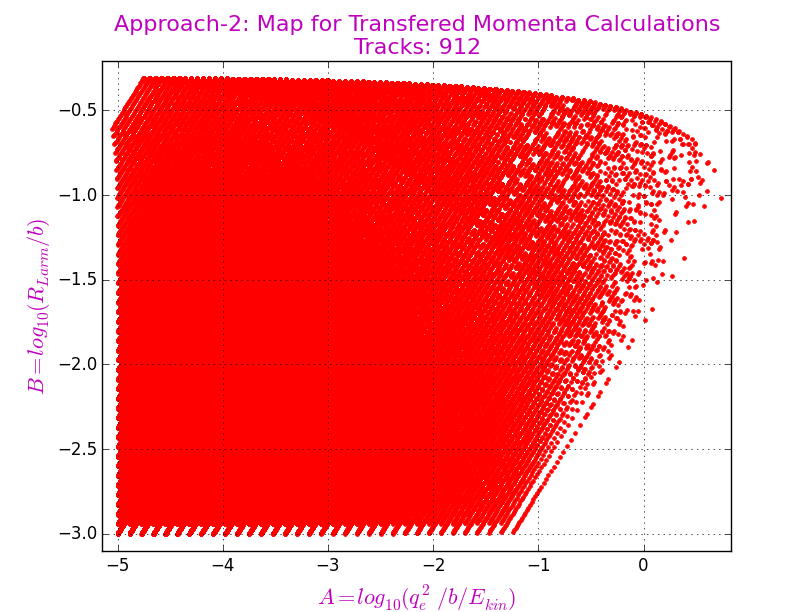
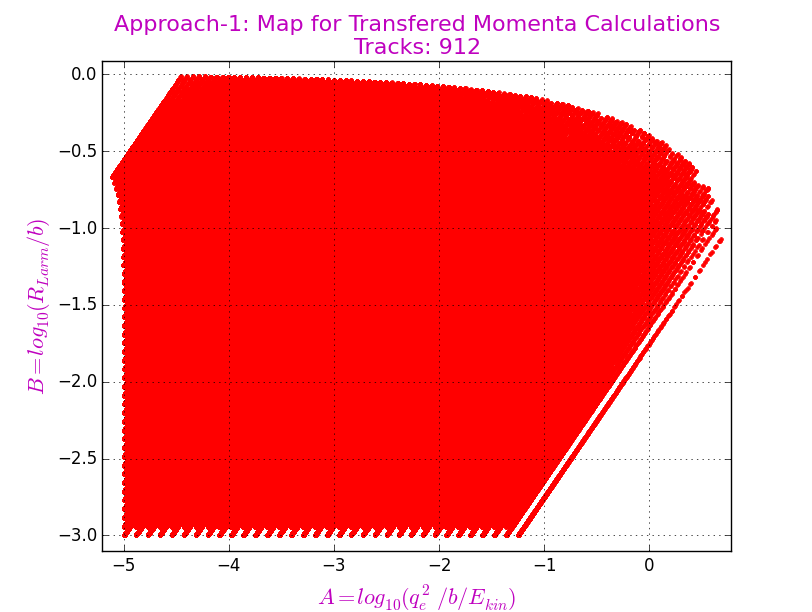
There is also a problem for visualizing the results. Since each track of electron usually contains many tens (and sometimes hundreds) of Larmor rotations, the result arrays are extremely large: 912 tracks used for modeling include more than 5 million points (for integration with Larmor circles) and more than 200,000 for integration with their averaging. The description of the procedure for "bypassing" of this problem will be given later.

Code flow diagram for *‘threeApproachComparison\_v4.py’:*

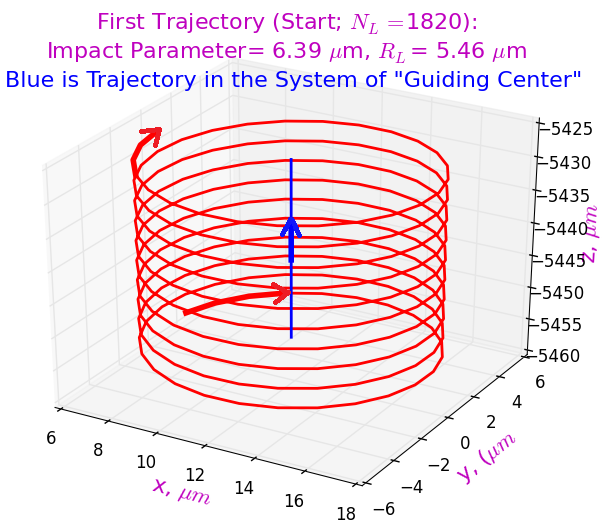
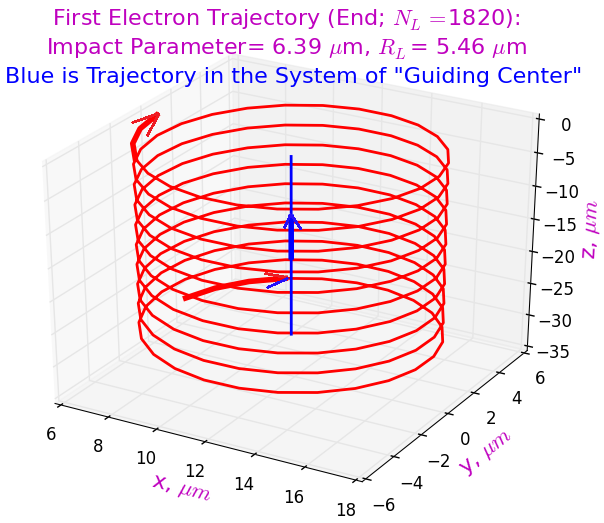




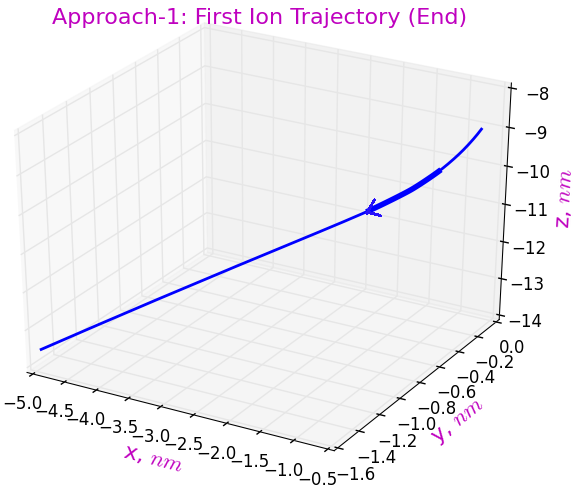
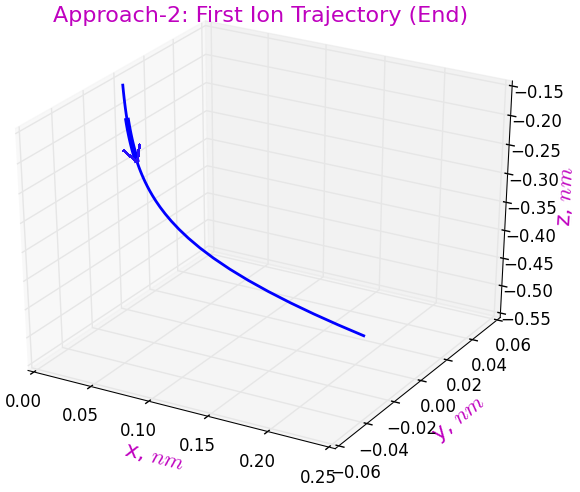
The comparison of the first track for each approach are shown in the next figures.



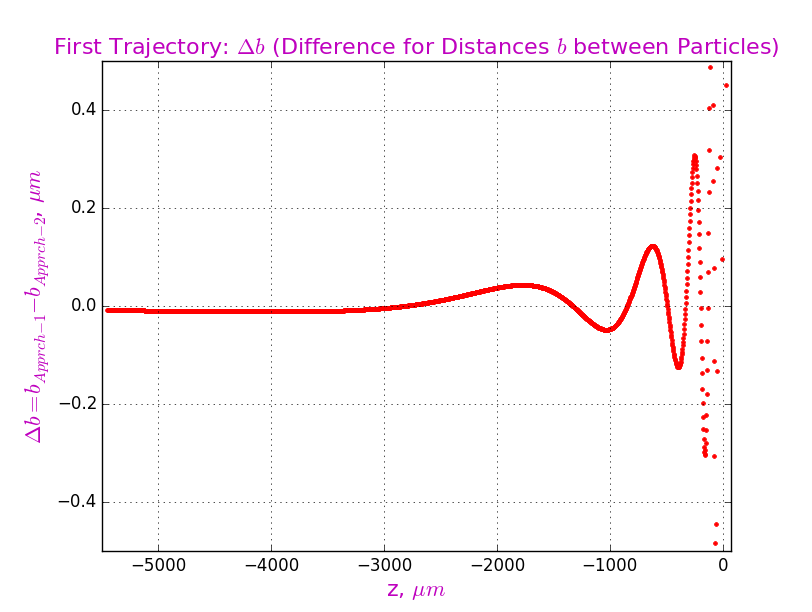
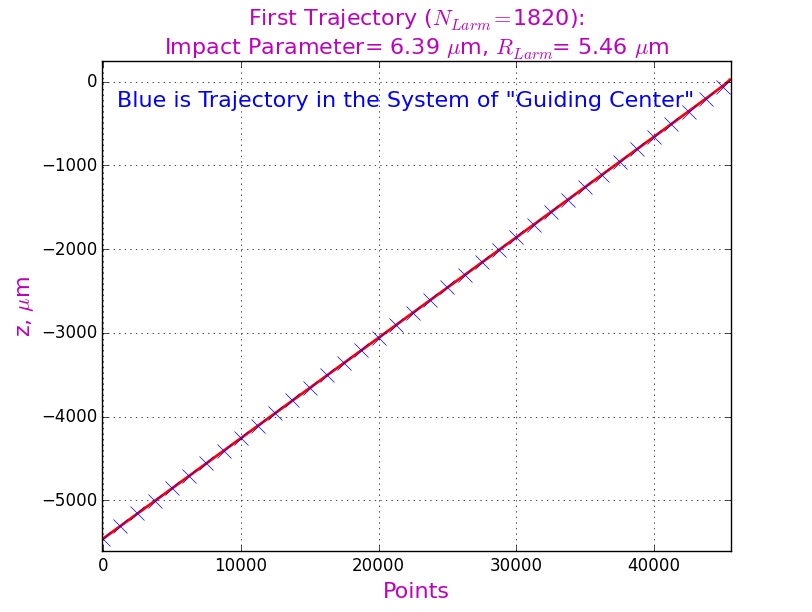
“Filling” of the plane .

Beginning and end of electron’s trajectory for both approaches.

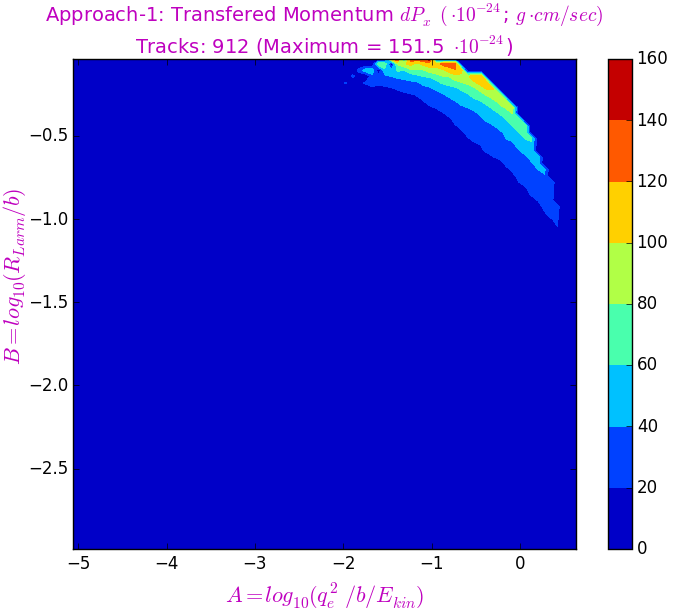
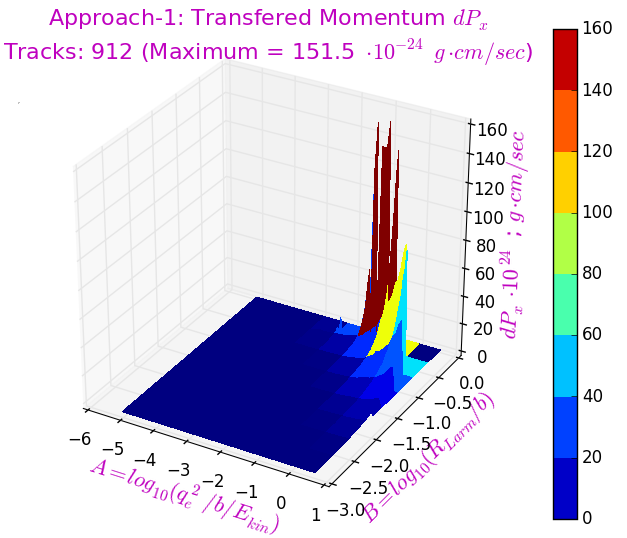
 

End of ion’s trajectory for both approaches.

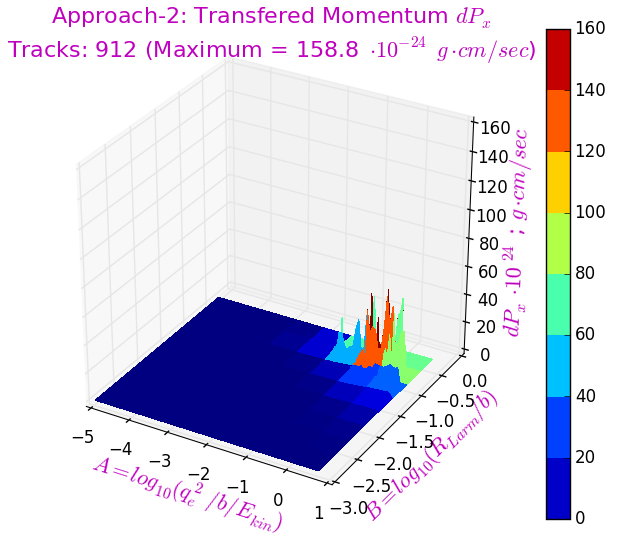
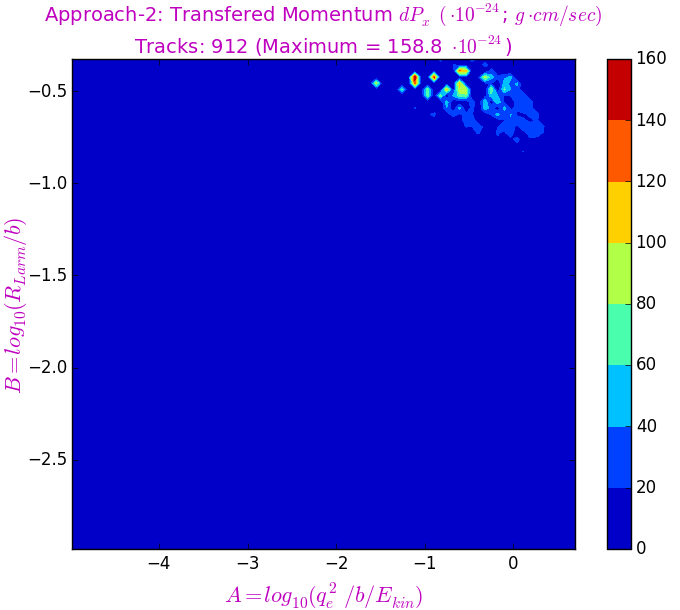


Trajectories of the electron for both approaches (left) and difference for distance between electron and ion for both approach (right).

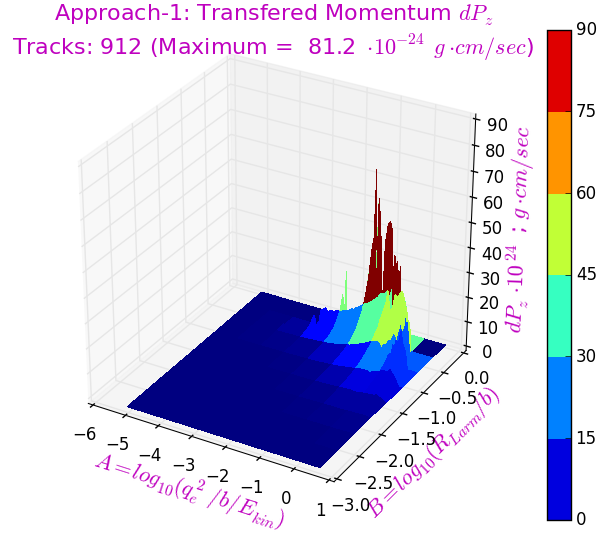
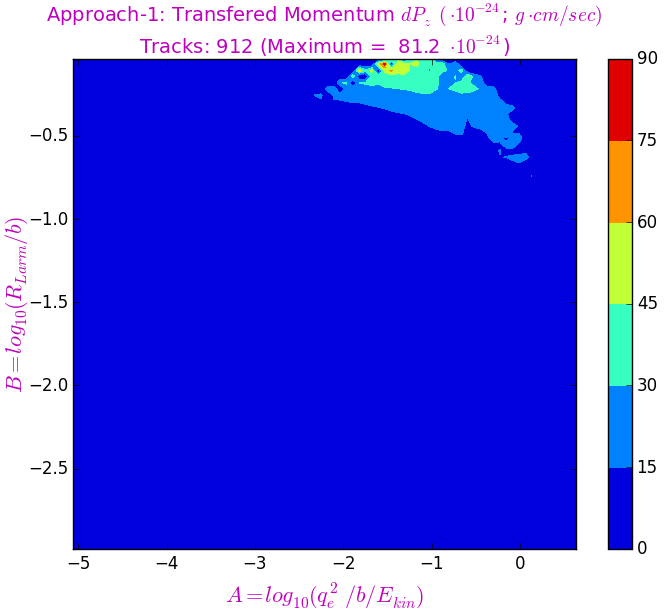
Last figures show a very small (less then part of micron) difference in the approaches in determining the distance between the electron and the ion during their mutual movement past each other. So, the comparison of transferred momenta for both approaches is entirely acceptable.



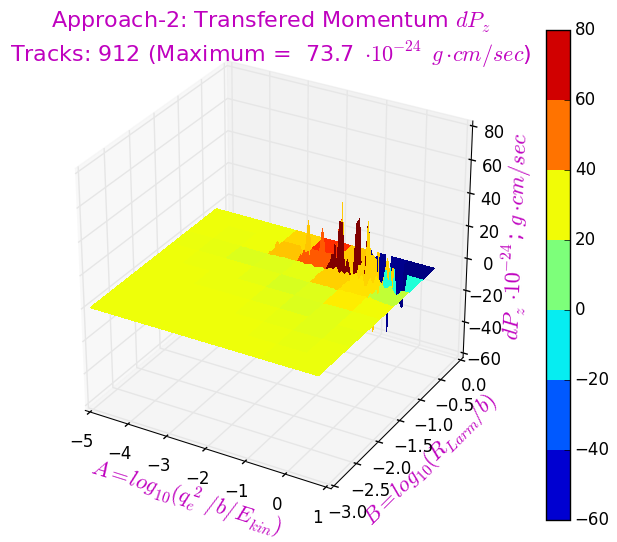
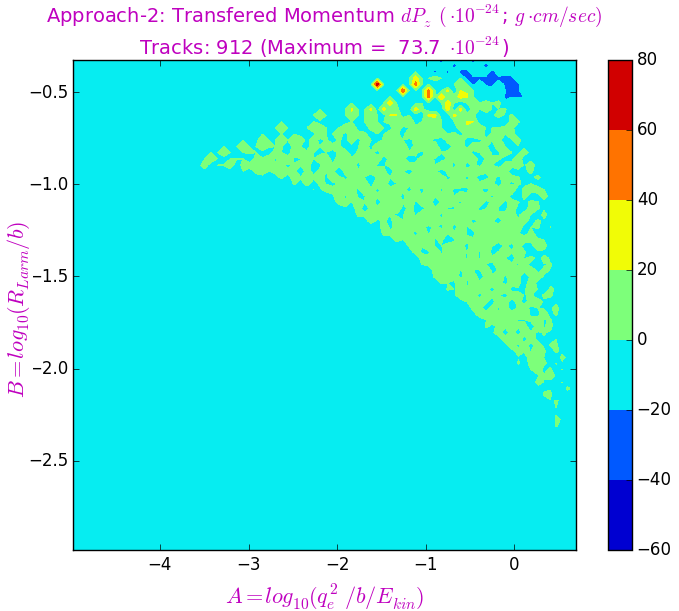
Transferred Momentum : Approach-1.

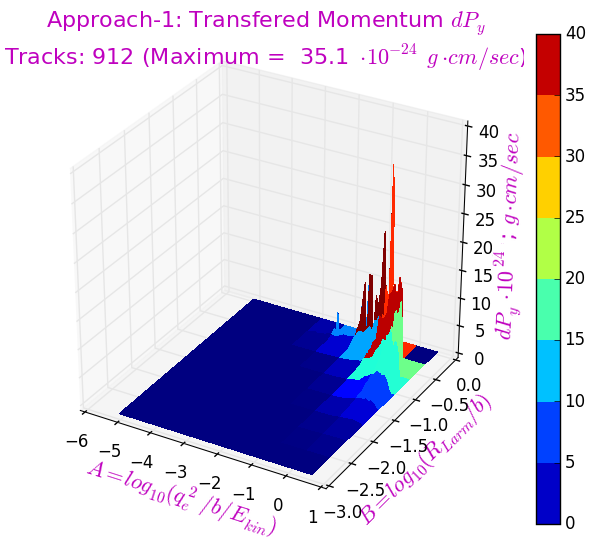
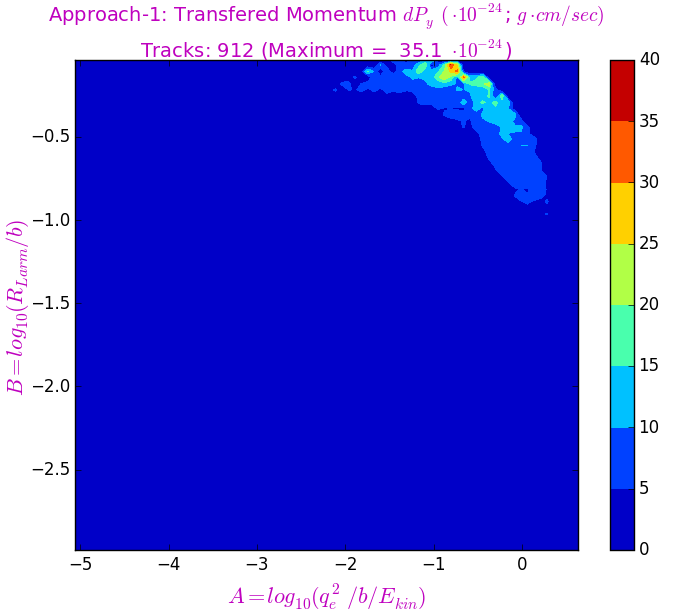
Transferred Momentum : Approach-2.

Transferred Momentum : Approach-1.

Transferred Momentum : Approach-2.

Transferred Momentum : Approach-1.

