B. Simulation of the GRACE beam line

The simulation program **GRACESimu** simulates the track of the particles inside of GRACE. It is built using the library IonBeam simulator (`IBSimu`) [46], which is a C++ library for for doing 3D simulation of electrostatic optics for charged particles.

The general setup of a simulation using `IBSimu` is the following: First the geometry of the experimental setup and the electric potential applied to each component is defined by the user. The user also defines the particles that should be tracked by giving their position, momentum, energy, charge, and mass at the starting point. Then `IBSimu` calculates the electric field in the defined experimental setup by numerically solving the Poisson equation using a finite difference method. At the end the particles are tracked trough the experimental setup using the Runge-Kutta method to solve the equation of motions. At any point along the track the state of the particle can be extracted.

The program **GRACESimu** implements this procedure for the geometry of GRACE. The input is the beam as it enters GRACE, meaning the input is the output from the **DegraderSimu** program. The output is the state of the particles as they hit any hard surface of GRACE. The geometry of the simulation is made such that the detector plane is at the end of the extraction line, and in this way information about the particles in the detector plane can be collected.

To download the IBSimu library go here: `http://ibsimu.sourceforge.net/download.html`. Instruction for installing the program with all its prerequisites is here: `http://ibsimu.sourceforge.net/installation.html`. For this example and the work of this thesis `ibsimu-1.0.6` was used.

B.1. Running the program

The **GRACESimu** program can be downloaded from Github in the repository named `GRACEbeamline`. Clone this repository and checkout tag `v1.1`, navigate to the `simulation` folder and check that all files are there.

```
git clone https://github.com/helgaholmestad/GRACEbeamline.git
git checkout v1.1
cd GRACEbeamline/simulations
ls
analysis.cpp                   inputfiles     loopVoltages.sh
findHitsInDetectorPlane.py     loopScan.sh    Makefile  simu.cpp
README.md
```

If another version of IBSimu is used, one needs to change the IBSimu version in the Makefile. The file `simu.cpp` contains the code that implements the geometry and the tracking of the particles. The simulation is compiled and run by the following command:

```
make
./simu D1 D2 E1 E2 inputfil outputfolder outputname
```

The explanation of the input variables is found in table B.1.

Name	Description
D1	Potential on D1 in V
D2	Potential on D2 in V
E1	Potential on E1 in V
E2	Potential on E2 in V
inputfile	The definition of the particles as they enter GRACE, this is the output from the **DegraderSimu** program
outputfolder	A folder where the output file will be written to. This folder has to exist before the program is run
outputname	An extension that is added to name of the outputfile.

Table B.1.: Explanation of the input variables to the **DegraderSimu** program.

A test inputfile consisting of 5000 antiprotons is found in the folder `inputfiles`. To run the simulation in the setting D_1=0, D_1=4 kV, E_1=4 kV, E_2=4 kV with the example inputfile do:

```
make
mkdir output
./simu 0 4000 4000 4000 inputfiles/testData.txt output test
```

This will result in the file `D1_0D2_4000E1_4000E2_4000_scanningtest.txt` in the folder `output`. As seen the file name contains information about the voltages applied to the electrodes. The output file has the following format:

```
-0.0646884  -0.00635098     0.216698  8.16357  216.203  initial 7.984790  3.000000  0.165374  0.160135  0.958666  -0.282619  -0.032947
-0.0471698    0.0497768 -7.82461e-07  11.1211  49.3167  initial 11.178900  3.000000  0.008536  -0.361915  0.167904  -0.653226  0.738312
 0.0648033  -0.00503579     0.243682  4.28857  385.444  initial 4.264250  3.000000  -0.303774  -0.123173  0.953798  0.299578  -0.022840
-0.0419013   -0.0496913    0.0105694  12.4184  42.9533  initial 12.496000  3.000000  -0.309322  -0.024272  0.334651  -0.581608  -0.741445
-0.0378919    0.0528027    0.0234933  1.99433  120.581  initial 2.148260  3.000000  -0.055215  0.208186  0.634056  -0.459914  0.621652
```

The three first columns is the position (x,y,z) in cm of the particles as they are stopped by any of the hard surfaces inside of GRACE. Column 4 is the kinetic energy in keV and column 5 is the time in ns that it takes for the particles to travel from the entrance of GRACE to the surface where they are stopped. The parameters after the column named `inital` is just a repetition of the line defining the particle in the inputfile.

Usually it is interesting to find the particles that actually end up in the detector plane. In the simulation the detector plane is at the end of the extraction line, so the particles are stopped here and written to file. A small python script was written to filter out the particles that ends up in the detector plane. This program is called `findHitsInDetectorPlane.py` and can be run as:

```
python findHitsInDetectorPlane.py inputfile outputfile
```

For the example simulation this program is run as:

```
python findHitsInDetectorPlane.py output/D1_0D2_4000E1_4000E2
_4000_scanningtest.txt particlesOnDetector.txt
```

In the file `particlesOnDetector.txt` now consist of only the three particles that makes it to the detector plane:

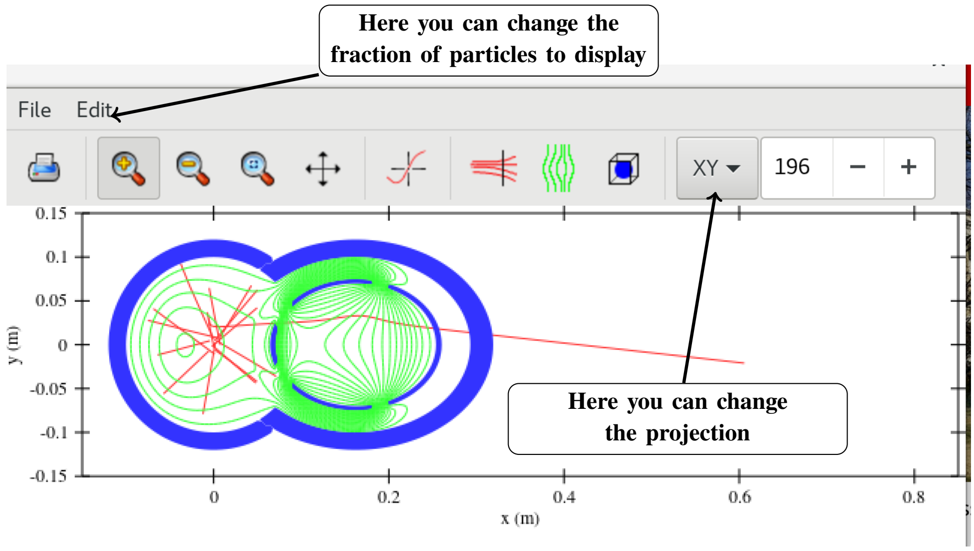
```
0.614458   0.0294305    1.18336  4.26177  1992.46  initial 4.264250  3.000000  -0.303774  -0.123173  0.953798  0.299578  -0.022840
0.605777  -0.0210164    1.19063  4.29489  1904.75  initial 4.302420  3.000000  0.040214  -0.308617  0.989834  -0.029629  0.139108
0.598242   0.0449757    1.19698  4.25528  1920.13  initial 4.259040  3.000000  -0.064945  0.161956  0.981463  0.074944  -0.176392
```

This file has exactly the same format as the output from the **GRACESimu** program, but contains only the lines corresponding to the particles that made it to the detector plane.

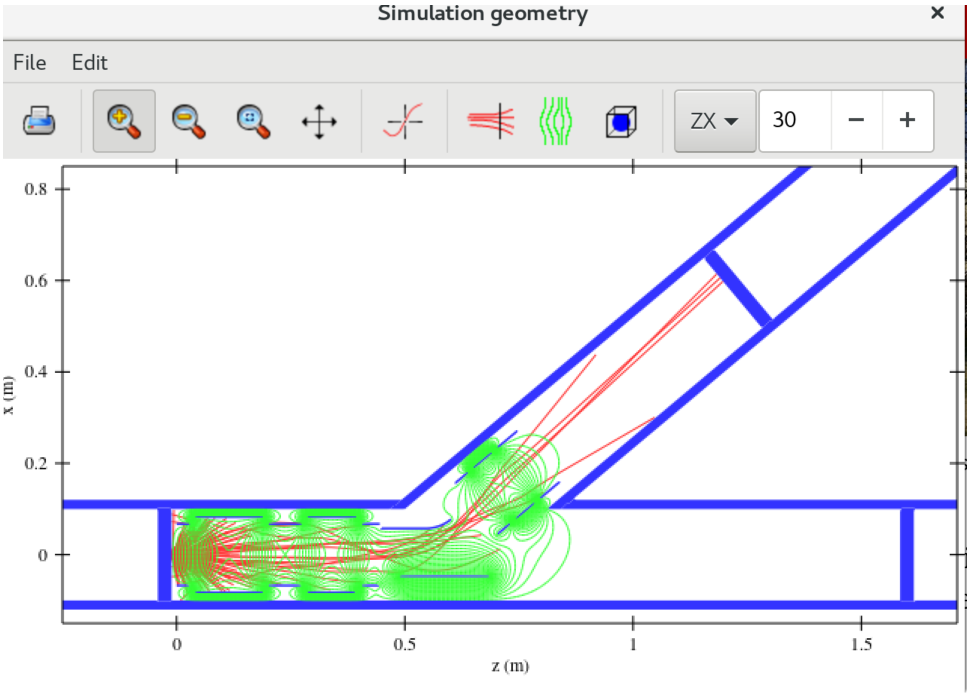
The program `analysis.cpp` provides a graphical display of the geometry and the tracks of the particles. The program is run by the command

```
  ./analysis geom.dat epot.dat pdb.dat
```

Figure B.1a shows the GUI in the default setting where the geometry is shown in the xy view, and only 1 out of every 11 tracks are drawn. The fraction of tracks to be shown can be changed by clicking Edit->Preference->Geometry->Trajectory division. As default the trajectory division is set to 11, and by setting this to 1 all tracks are drawn. As shown in figure B.1a the projection can also be changed. Figure B.1b shows the GUI in the zx view and with all tracks drawn.


Here you can change the fraction of particles to display
File
Edit
XY
196
−
+
Here you can change the projection
y (m)
x (m)
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
0
0.2
0.4
0.6
0.8

(a) The GUI in the default setting



(b) Better setting to understand what is going on

Figure B.1.: The graphical interface for **GRACESimu**.