

EDIPIC-2D output data description

Dmytro Sydorenko

The output data produced by the code include:

- Text output with general messages reporting simulation progress.
- Binary snapshot files saved at certain moments of time:
 - 2d profiles of electric field parameters (potential, fields) and main moments of particle velocity distribution function (densities, flow velocities, currents, average energies, temperatures),
 - 1d and 2d velocity distribution functions.
- Text files with time dependencies of potential, electric fields, and densities in probes.
- Text files with numbers of collision events at each ion time step.
- Text files with numbers of particles collided with/emitted by all boundary objects at each electron step.

General text output (1)

```
.....
### Maximal number of electron time steps      2142 ###
### Internal balancing of cluster load occurs every    110 electron time steps ###
### Global load balancing occurs every    1100 electron time steps ###
### Checkpoints will be saves every  110000 electron time steps ###
### Fresh start, checkpoint NOT USED
.....
electron thermal velocity for the scale temperature v_Te_ms =  0.1186189E+07 m/s
circular electron plasma frequency for the scale density w_plasma_s1 =  0.1783985E+11 s^-1
electron Debye length for the scale density/temperature L_debye_m =  0.6649094E-04 m
mesh size delta_x_m =  0.3324547E-04 m
electron time step delta_t_s =  0.4671189E-11 s
scale value of electric field E_scale_Vm =  0.8662834E+07 V/m
scale value of magnetic field B_scale_T =  0.1217180E+01 T
scale value of velocity V_scale_ms =  0.7117132E+07 m/s
.....
```

The complete text output contains numerous messages from multiple procedures. Description of all these messages is out of the scope of the present document. Here is the part of output in the very beginning of a simulation which reports important parameters such as number of time steps, mesh size, time step, etc.

General text output (2)

Here is an example of the text output during one cycle of simulation which included ion advance.

Time step
number

PETSc convergence report

```
10
Total : number of electron particles =
Total : number of ion 1 particles =
  0 KSP Residual norm 3.890716031630e+03
  1 KSP Residual norm 1.490089920013e+02
  2 KSP Residual norm 7.282020560719e+00
  3 KSP Residual norm 3.380126867067e-01
  4 KSP Residual norm 1.816512628821e-02
  5 KSP Residual norm 1.112436972605e-03
  6 KSP Residual norm 6.161132178382e-05
  7 KSP Residual norm 2.408043284189e-06
----- doing ions at step 10 -----
electrons hit boundaries ::      138      31      179      25      182      13      74      18
ions ( 1 ) hit boundaries ::       9        0        5        1        3        0        1        1
Total collisions with neutral species 1 (Argon0) :: 2362      0      0      0      0      98      11.9      0
electrons emitted by boundaries ::      98      3.4     62.9      0.6      0.4      0.1      13.9      0.1
  10      0.718      0.1      3.8      0.1      0.1      3.4      62.9      0.6
.....
```

Time the computer spent to complete this cycle [seconds].

Here numbers from left to right correspond to elastic/inelastic/ionization collisions

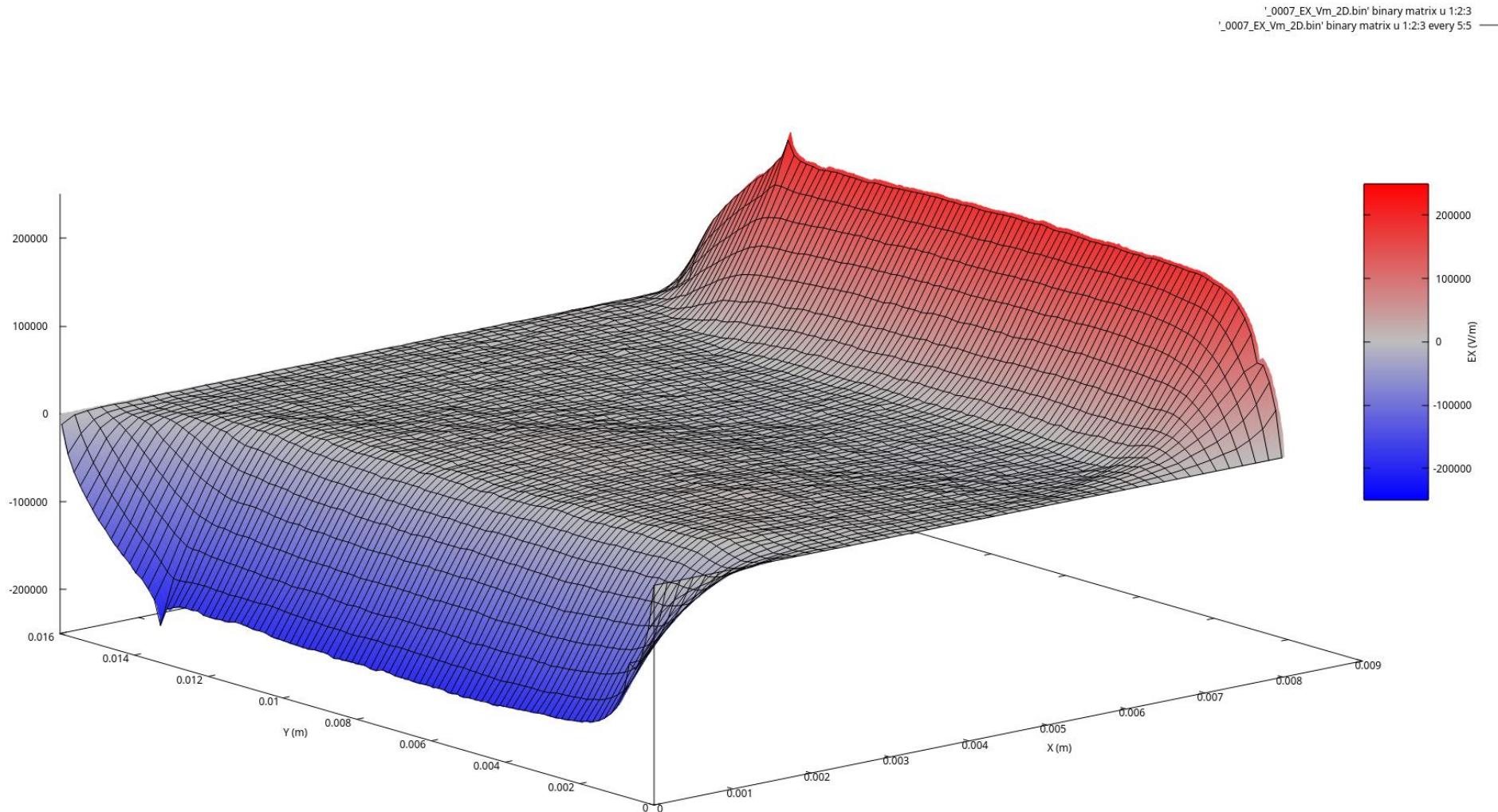
In these lines, numbers from left to right correspond to boundary objects 1 to 8.

Snapshots, 2d profiles of general field and plasma parameters

- Slides 6 to 34 show examples of 3d plots (variable vs coordinates x and y) of the data saved in snapshot files.
- The data are plotted with the gnuplot. The gnuplot commands are given in the green text boxes.
- The data are saved as matrices in binary format recognizable by gnuplot.
- Disclaimer: I didn't try to make the figures pretty, there is also no scientific value in these plots. They are obtained after only 10 ns of system evolution and are given here just to show which data are saved in which file and how one can plot/use it.

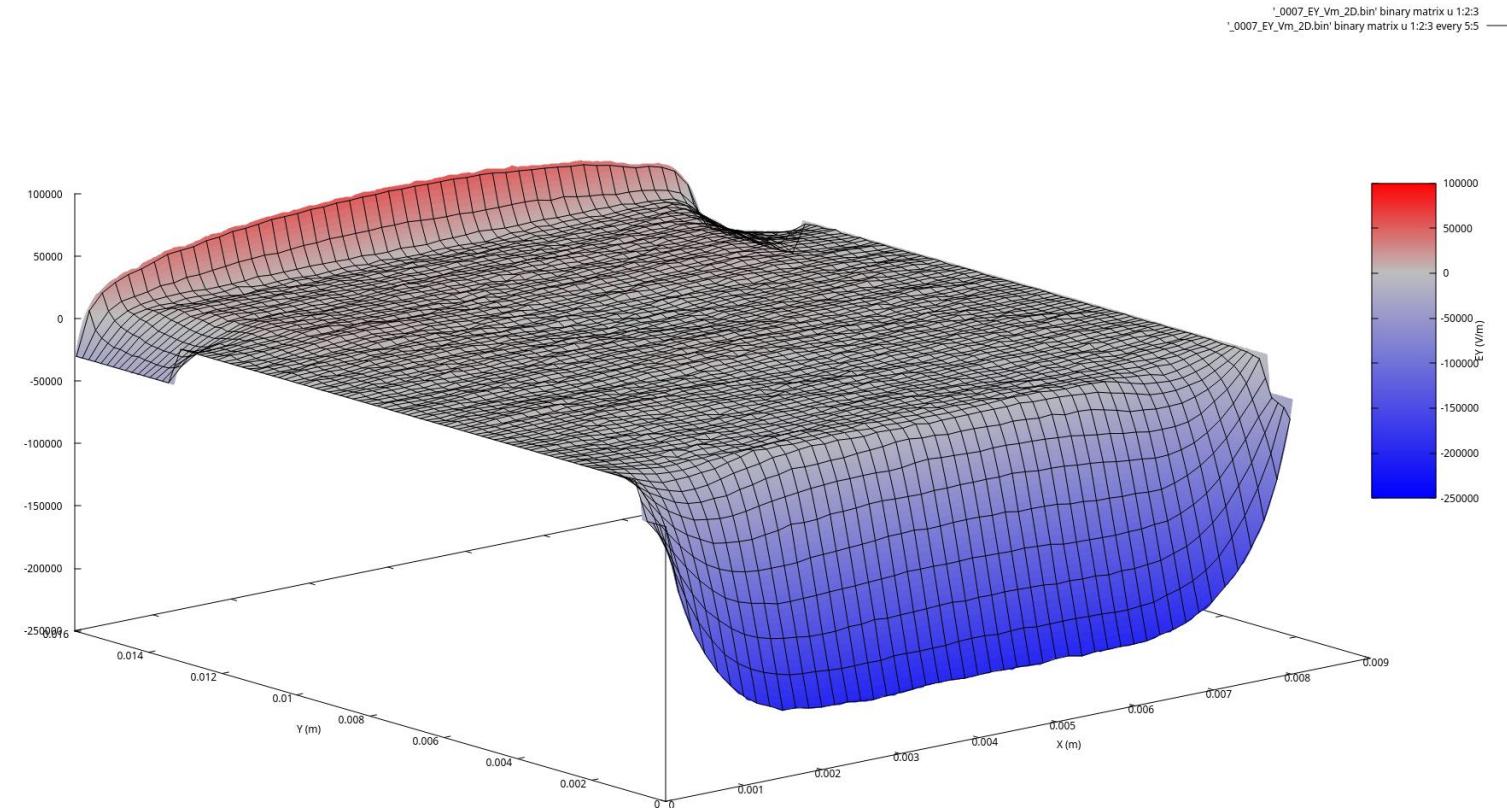
_NNNN_EX_Vm_2D.bin :: electric field component EX [V/m]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'EX (V/m)'; set ticslevel 0  
gnuplot> a=250000;set zran [-a:a];set cbran [-a:a]; set palette defined (-a 'blue', 0 'gray', a 'red')  
gnuplot> splot '_0007_EX_Vm_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_EX_Vm_2D.bin' binary matrix u 1:2:3 every 5:5 w l 1 lt 8
```



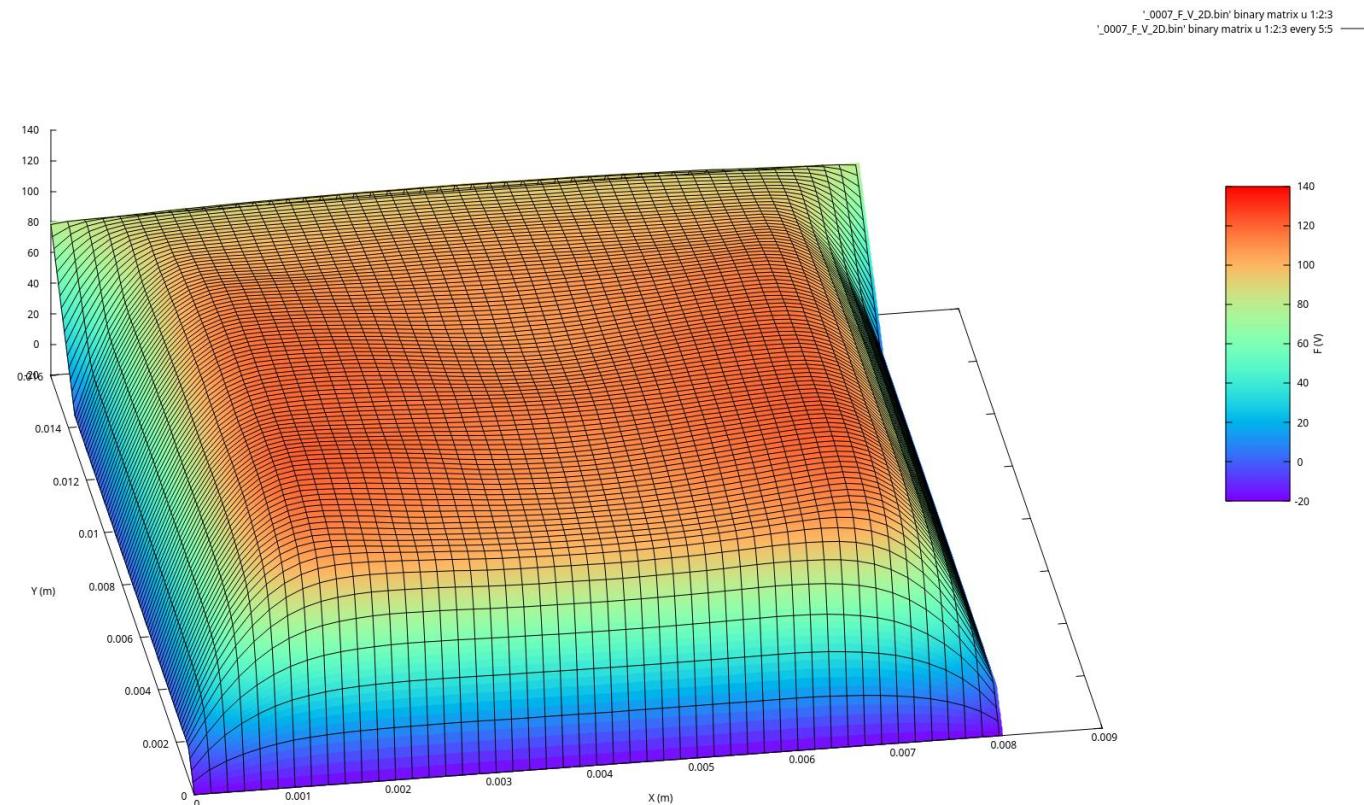
_NNNN_EY_Vm_2D.bin :: electric field component EX [V/m]

```
gnuplot> a=-250000;b=100000;set zran [a:b];set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'EY (V/m)';
gnuplot> splot '_0007_EY_Vm_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_EY_Vm_2D.bin' binary matrix u 1:2:3 every 5:5 w l lt 8
```



_NNNN_F_V_2D.bin :: electrostatic potential Φ [V]

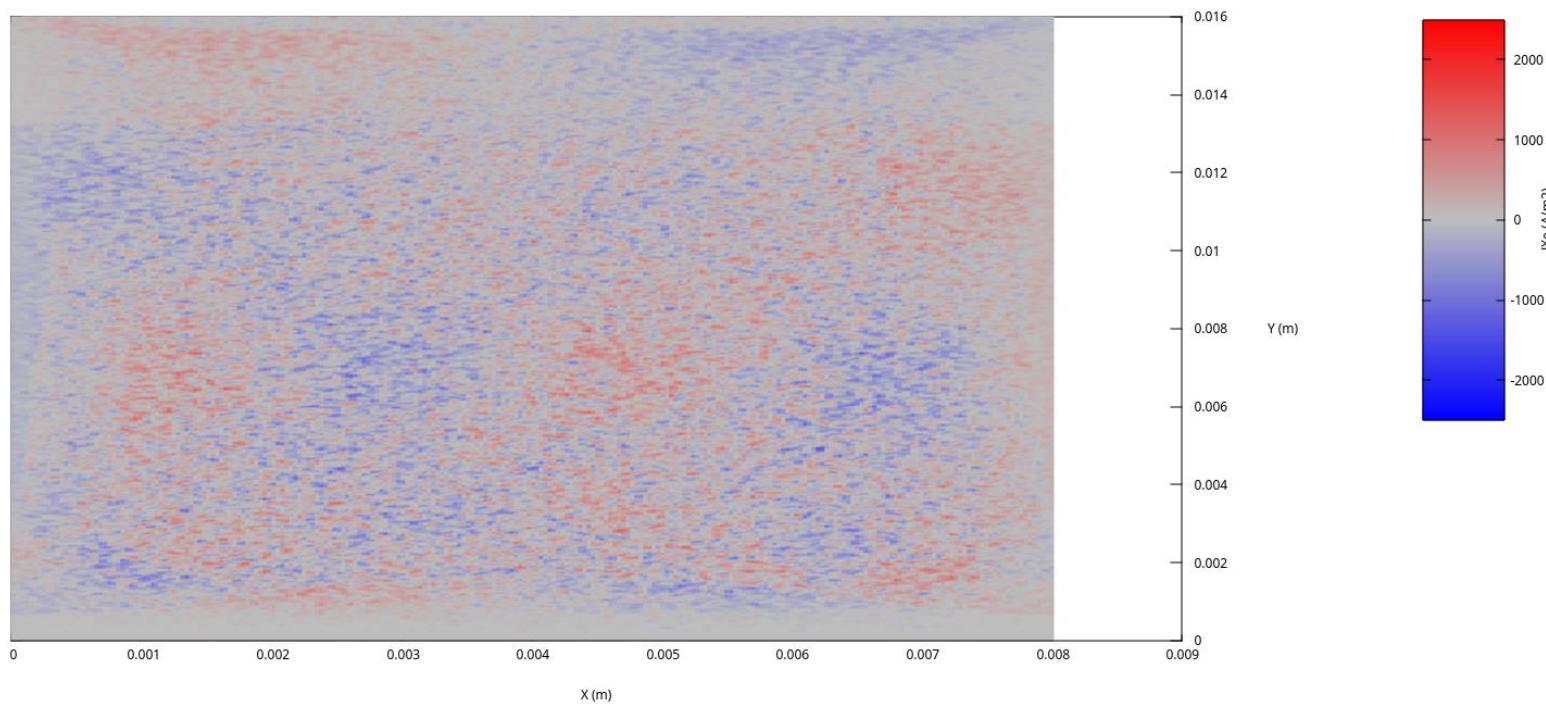
```
gnuplot> a=-20;b=140;set zran [a:b];set cbran [a:b];set palette rgbformulae 33,13,10;  
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'F (V)';  
gnuplot> splot '_0007_F_V_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_F_V_2D.bin' binary matrix u 1:2:3 every 5:5 w l lt 8
```



_NNNN_JXe_Am2_2D.bin :: electric current density component JX due to electrons [A/m²]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JXe (A/m2)';
gnuplot> a=-2500;b=2500;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set view 0, 0;
gnuplot> splot '_0007_JXe_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

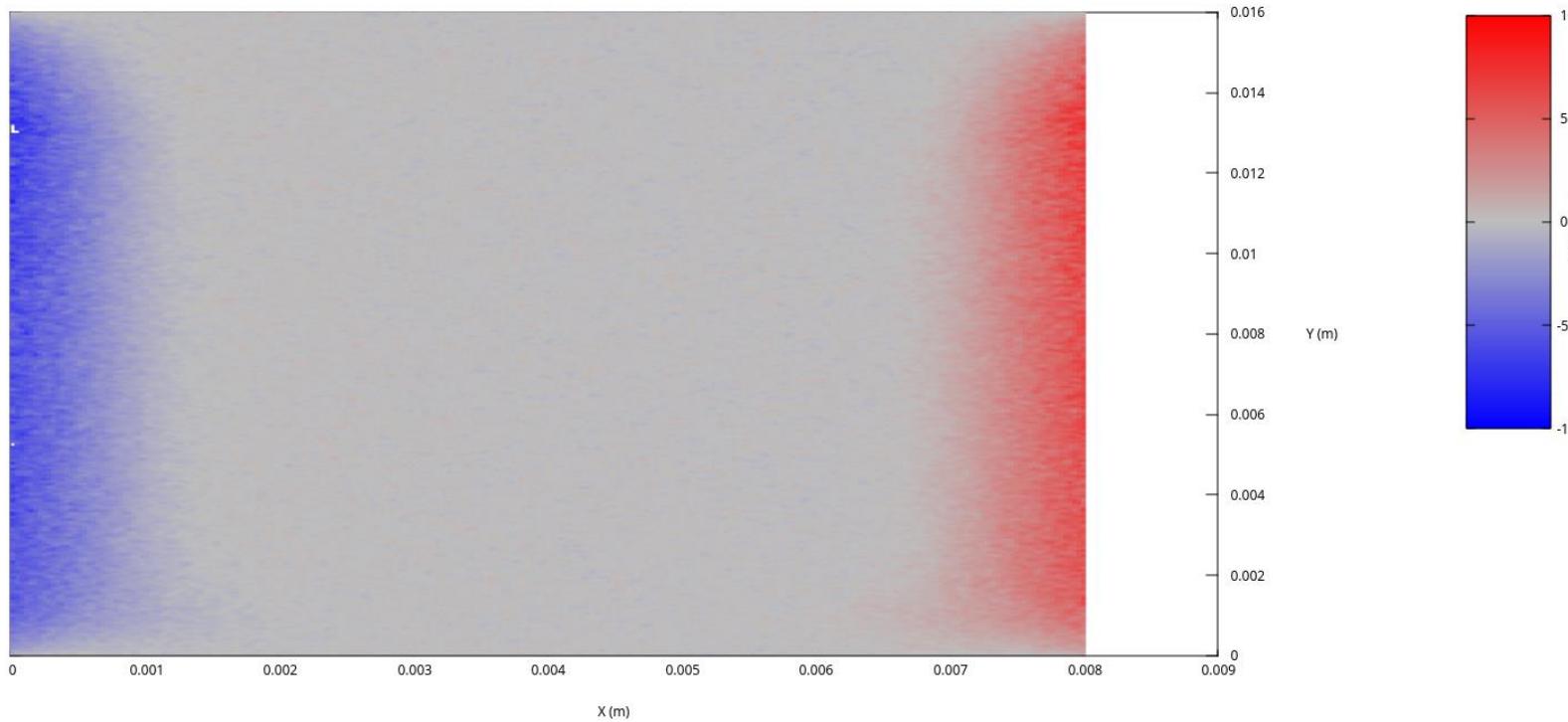
'_0007_JXe_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_JXi_1_Am2_2D.bin :: electric current density component JX due to ion species 1 [A/m²]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JXi_1 (A/m2)';
gnuplot> a=-10;b=10;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> splot '_0007_JXi_1_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

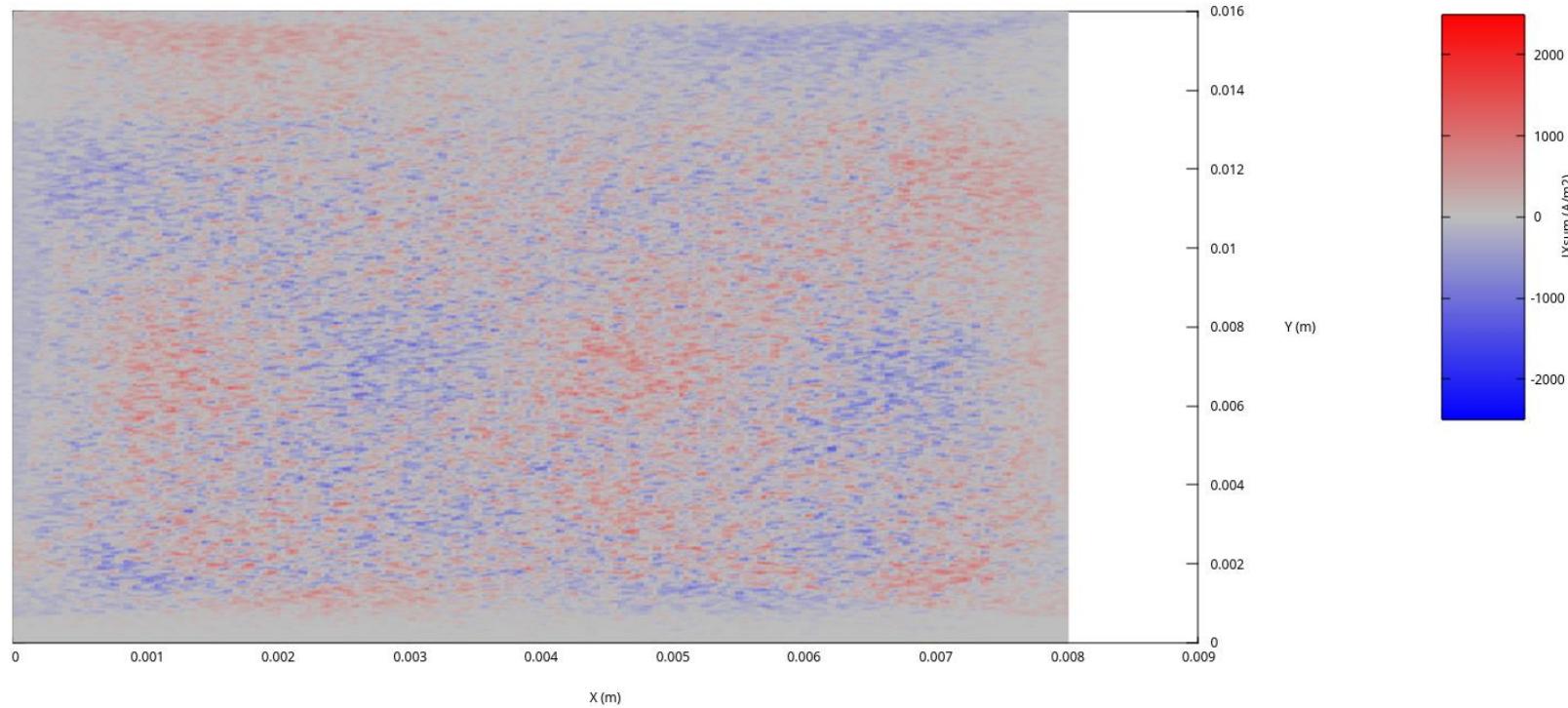
'_0007_JXi_1_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_JXsum_Am2_2D.bin :: total electric current density component JX due to all charged species [A/m²]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JXsum (A/m2)';
gnuplot> a=-2500;b=2500;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> splot '_0007_JXsum_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

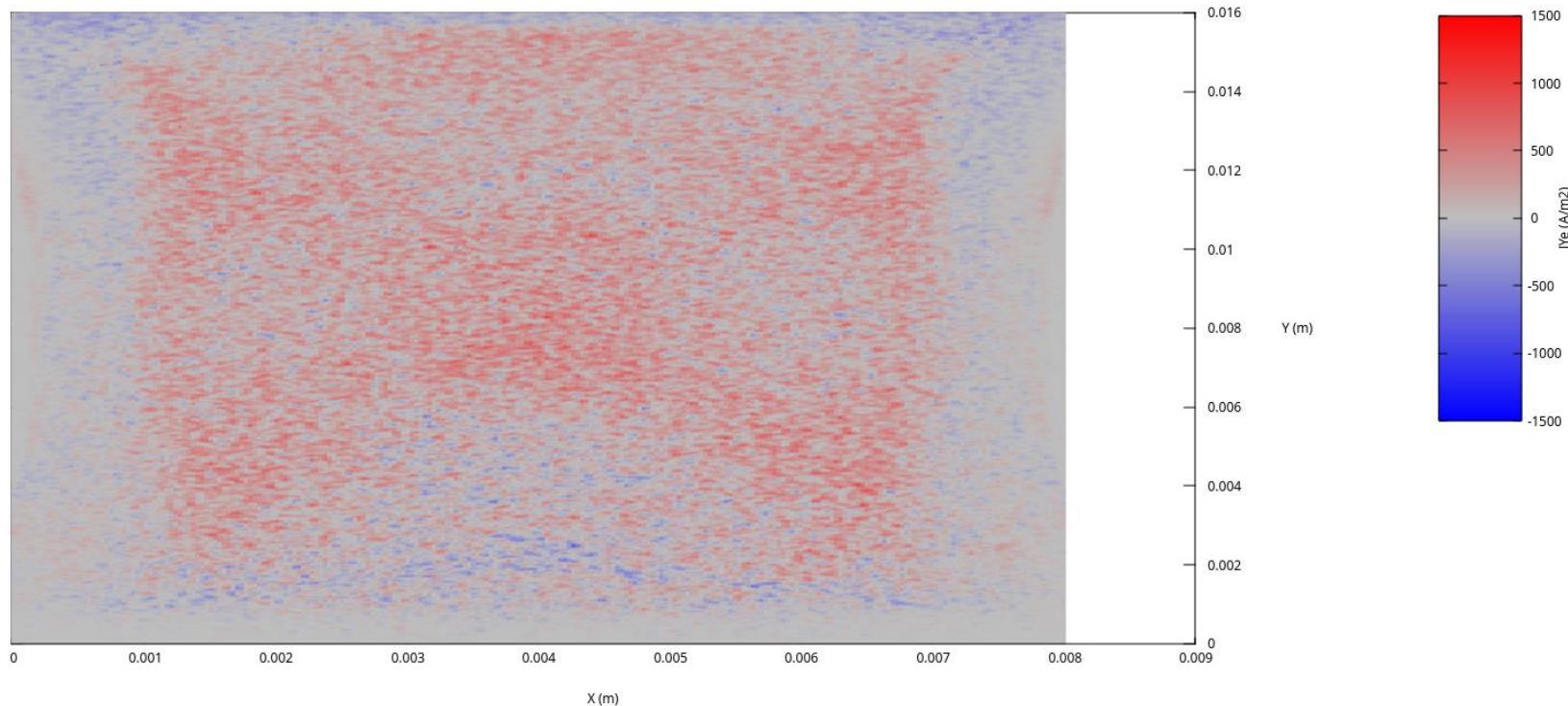
'_0007_JXsum_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_JYe_Am2_2D.bin :: electric current density component JY due to electrons [A/m²]

```
gnuplot> a=-1500;b=1500;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JYe (A/m2)';
gnuplot> splot '_0007_JYe_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

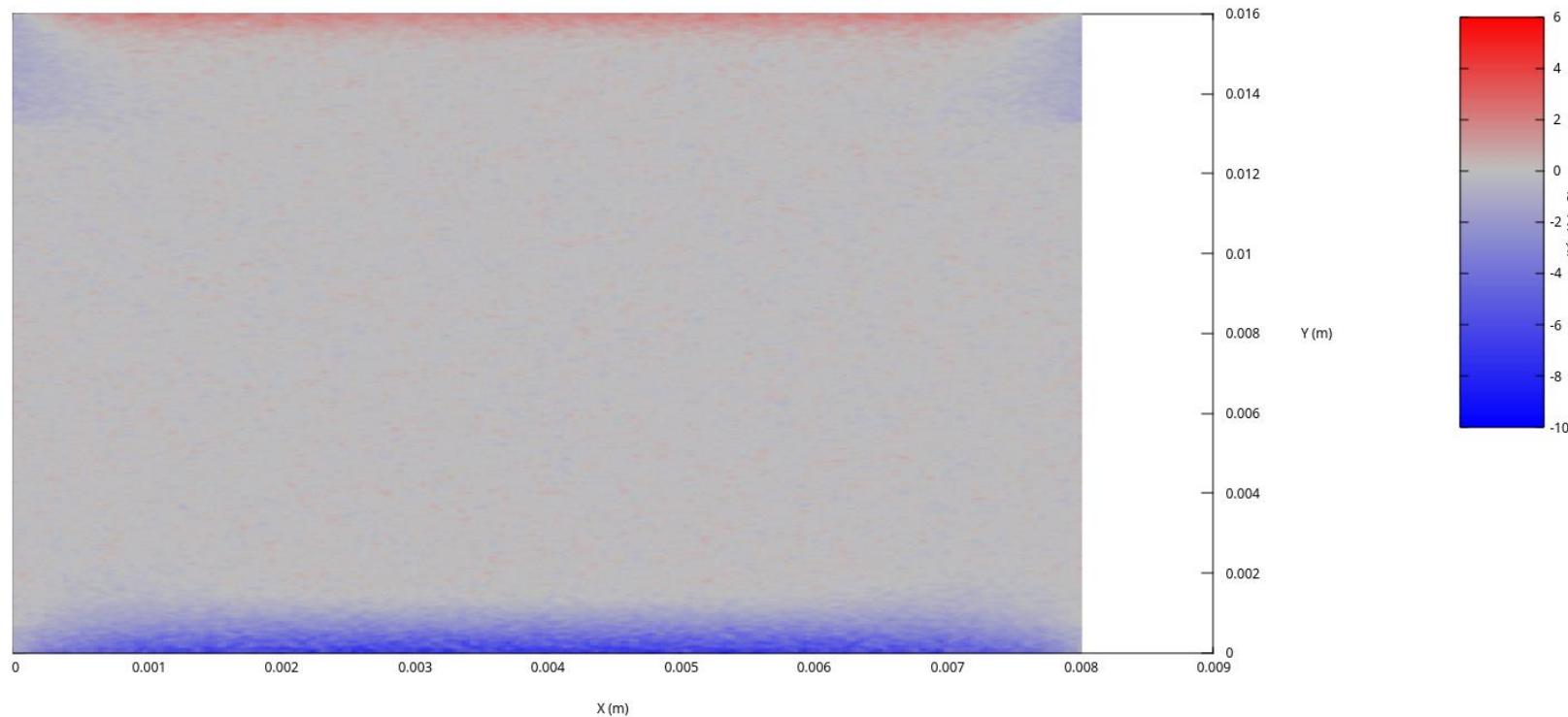
'_0007_JYe_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_JYi_1_Am2_2D.bin :: electric current density component JY due to ion species 1 [A/m²]

```
gnuplot> a=-10;b=6;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JYi_1 (A/m2)';
gnuplot> splot '_0007_JYi_1_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

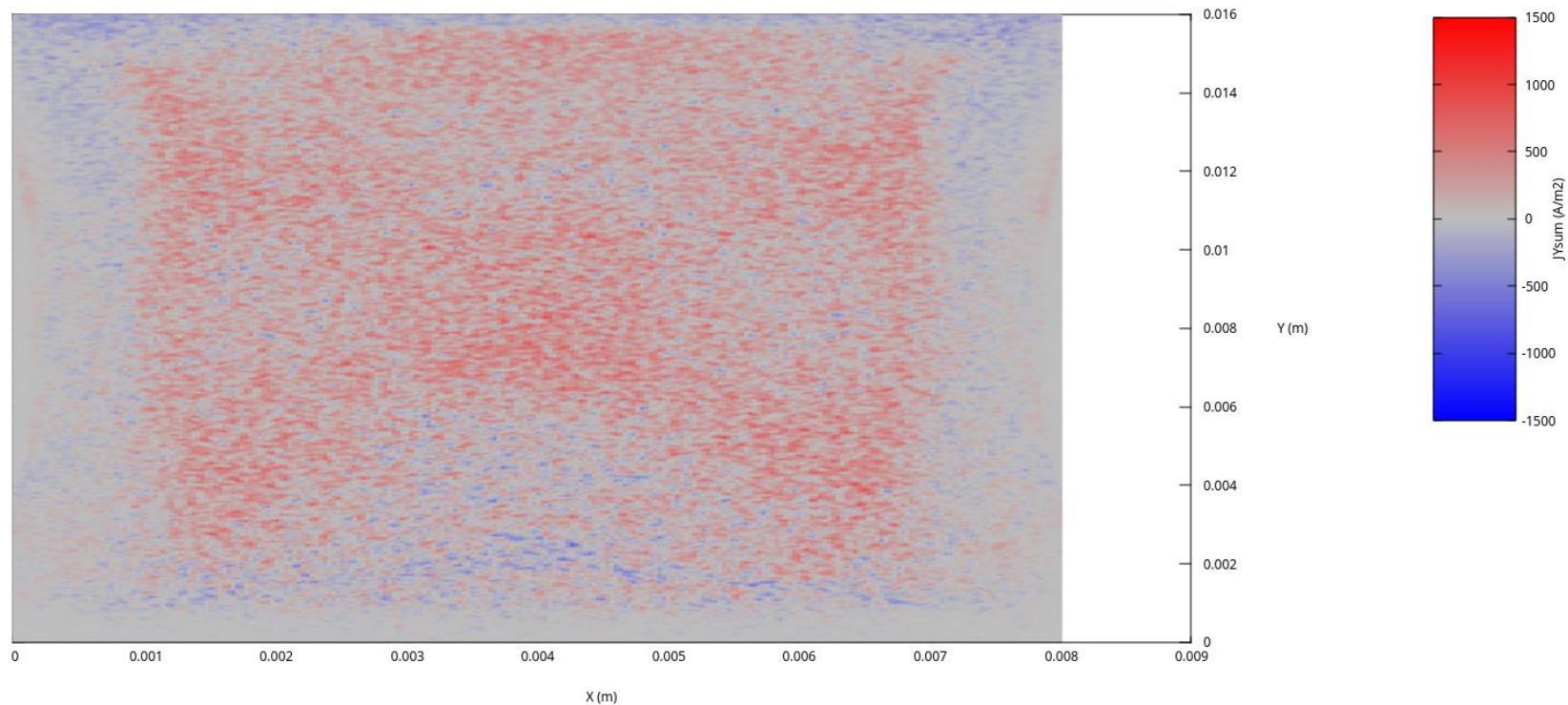
'_0007_JYi_1_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_JYsum_Am2_2D.bin :: total electric current density component JY due to all charged species [A/m²]

```
gnuplot> a=-1500;b=1500;set zran [a:b];set cbran [a:b];set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'JYsum (A/m2)';
gnuplot> splot '_0007_JYsum_Am2_2D.bin' binary matrix u 1:2:3 w pm3d
```

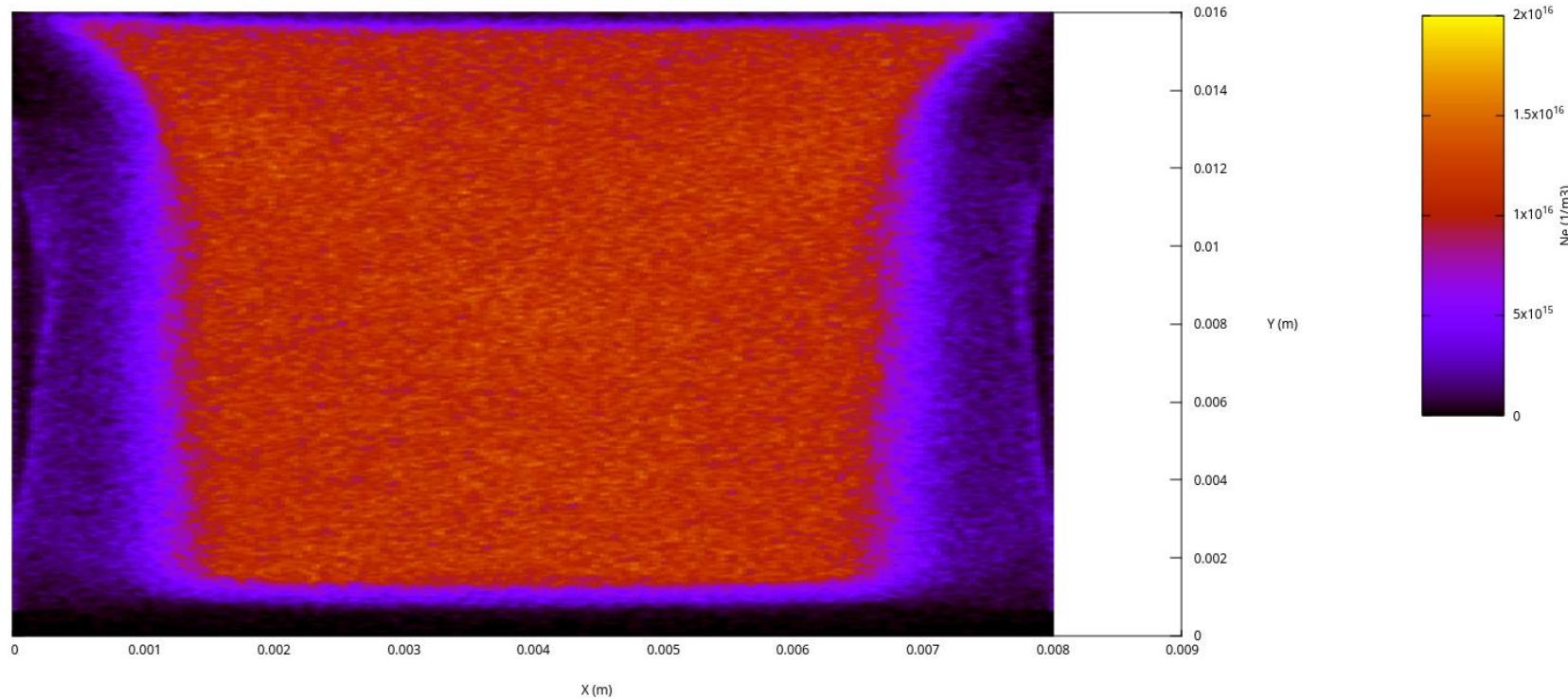
'_0007_JYsum_Am2_2D.bin' binary matrix u 1:2:3



_NNNN_Ne_m3_2D.bin :: electron number density [m⁻³]

```
gnuplot> a=0;b=2e16;set zran [a:b];set cbran [a:b]; set palette rgbformulae 7,5,15  
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'Ne (1/m3)';  
gnuplot> splot '_0007_Ne_m3_2D.bin' binary matrix u 1:2:3 w pm3d
```

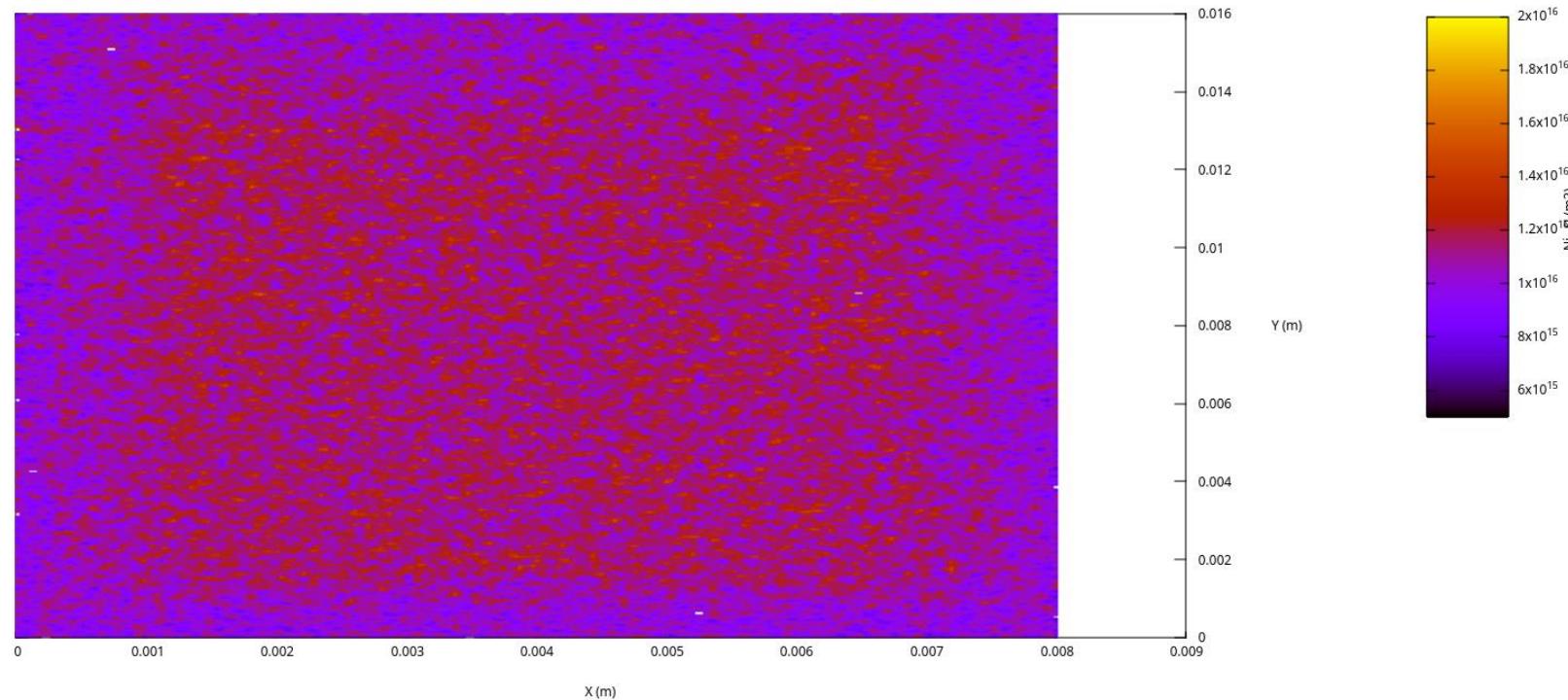
'_0007_Ne_m3_2D.bin' binary matrix u 1:2:3



_NNNN_Ni_1_m3_2D.bin :: number density of ion species 1 [m⁻³]

```
gnuplot> a=5e15;b=2e16;set zran [a:b];set cbran [a:b]; set palette rgbformulae 7,5,15  
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'Ni_1 (1/m3)';  
gnuplot> splot '_0007_Ni_1_m3_2D.bin' binary matrix u 1:2:3 w pm3d
```

'_0007_Ni_1_m3_2D.bin' binary matrix u 1:2:3

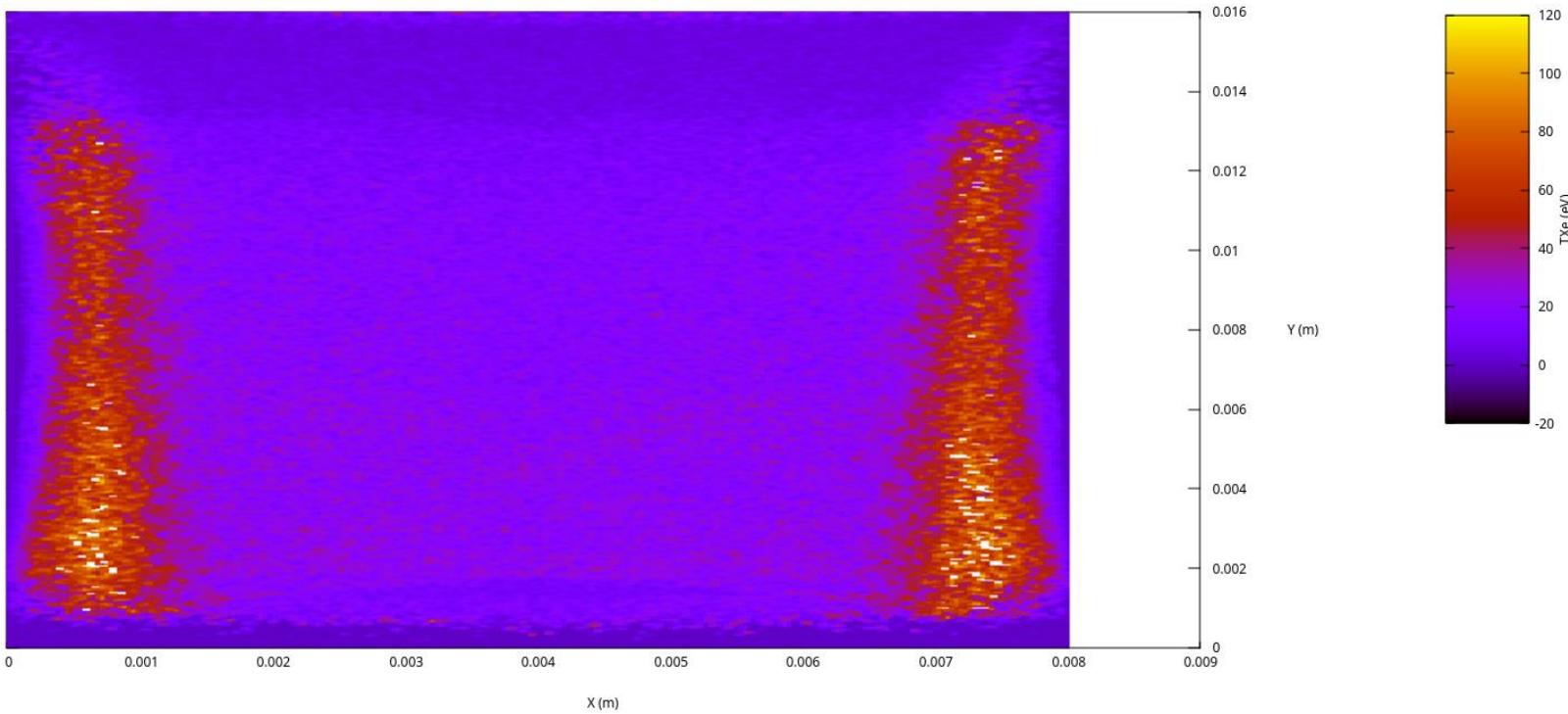


_NNNN_TXe_eV_2D.bin :: electron temperature along the X direction [eV]

```
gnuplot> set zran [-0.0001:120];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TXe (eV)';
gnuplot> splot '_0007_TXe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as
 $T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right)$.

'_0007_TXe_eV_2D.bin' binary matrix u 1:2:3



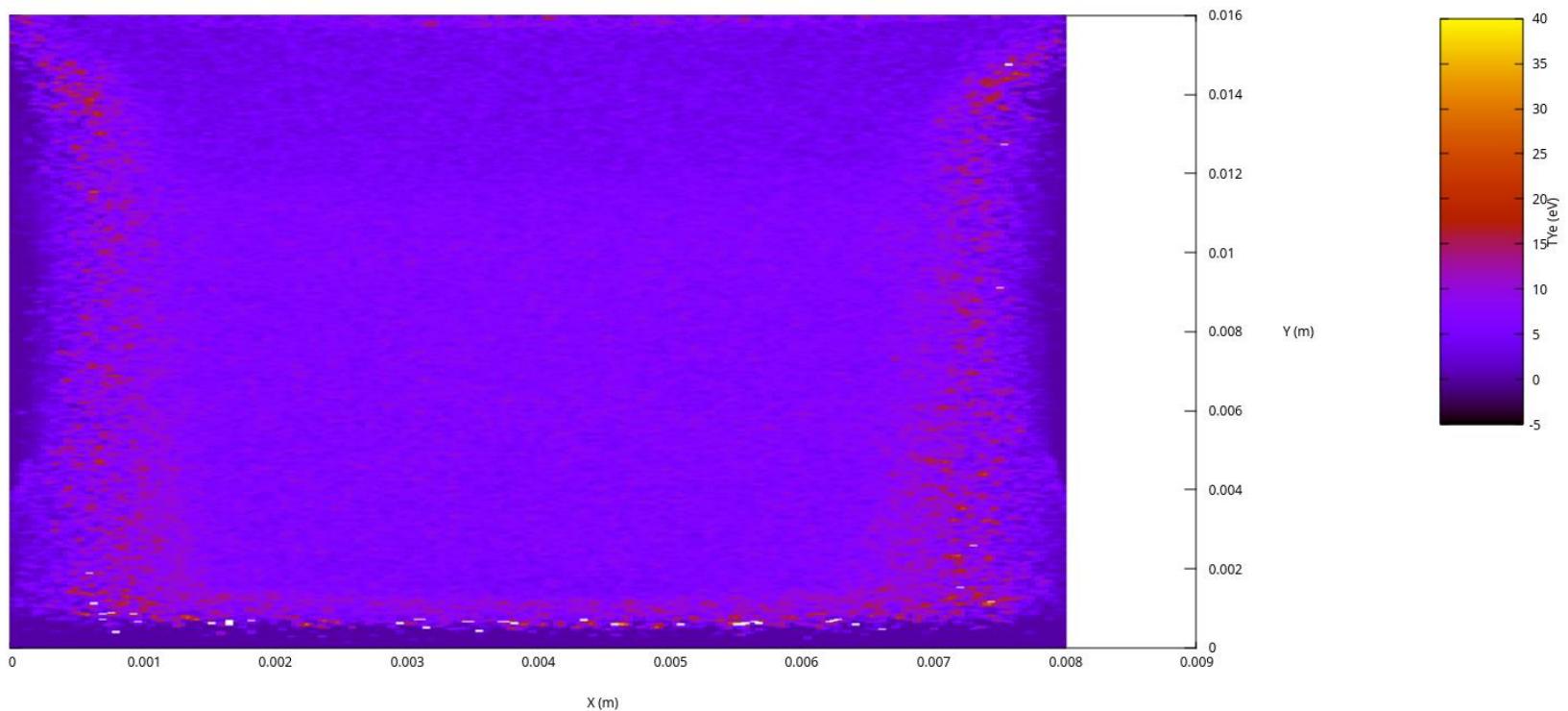
_NNNN_TYe_eV_2D.bin :: electron temperature along the Y direction [eV]

```
gnuplot> set zran [-0.0001:40];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TYe (eV)';
gnuplot> splot '_0007_TYe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as

$$T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right).$$

'_0007_TYe_eV_2D.bin' binary matrix u 1:2:3



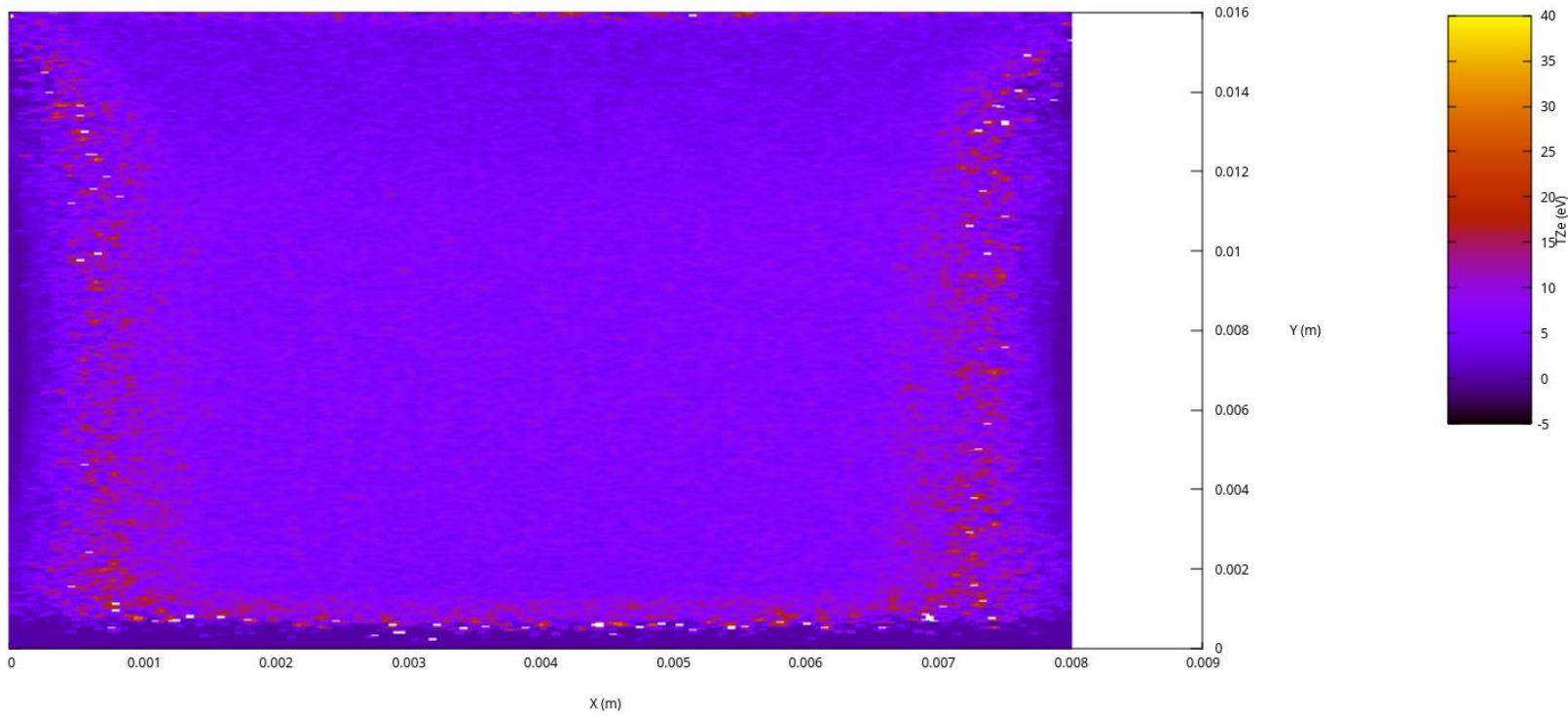
_NNNN_TZe_eV_2D.bin :: electron temperature along the Z direction [eV]

```
gnuplot> set zran [-0.0001:40];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TZe (eV)';
gnuplot> splot '_0007_TZe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as

$$T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right).$$

'_0007_TZe_eV_2D.bin' binary matrix u 1:2:3

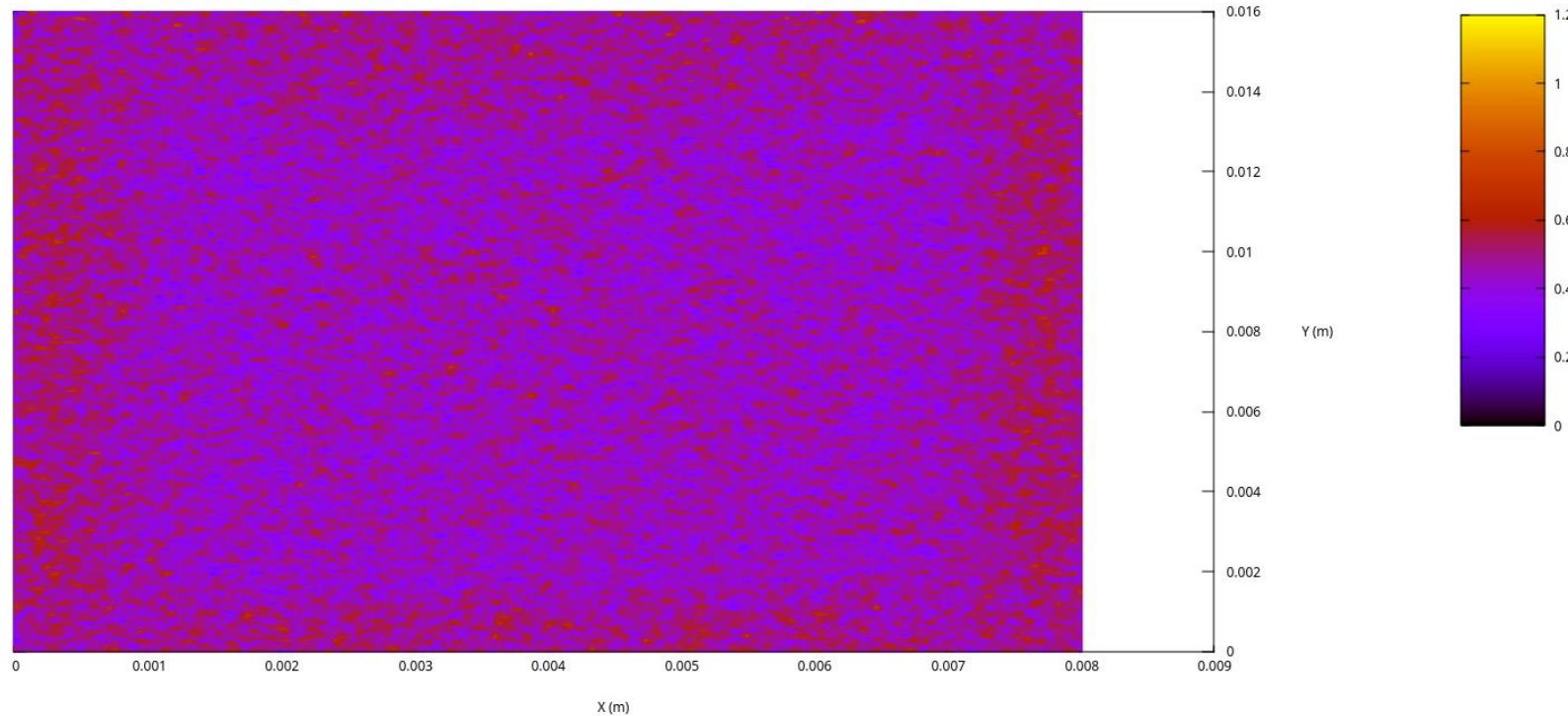


_NNNN_TXi_1_eV_2D.bin :: temperature along the X direction of ion species 1 [eV]

```
gnuplot> set zran [*:*]; set cbran [*:*]
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TXi_1 (eV)';
gnuplot> splot '_0007_TXi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as
 $T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right)$.

'_0007_TXi_1_eV_2D.bin' binary matrix u 1:2:3

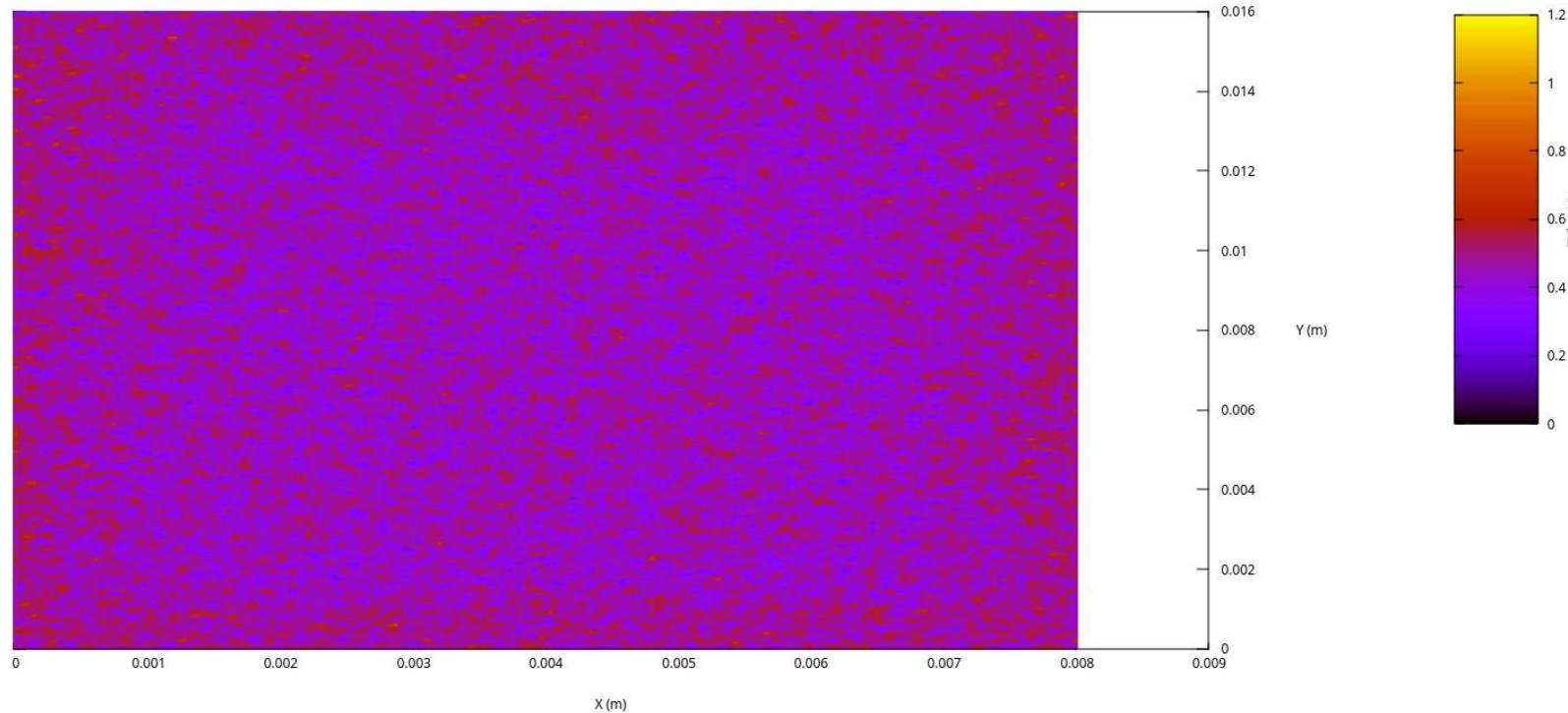


_NNNN_TYi_1_eV_2D.bin :: temperature along the Y direction of ion species 1 [eV]

```
gnuplot> set zran [*:*]; set cbran [*:*]
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TYi_1 (eV)';
gnuplot> splot '_0007_TYi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as
 $T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right)$.

'_0007_TYi_1_eV_2D.bin' binary matrix u 1:2:3

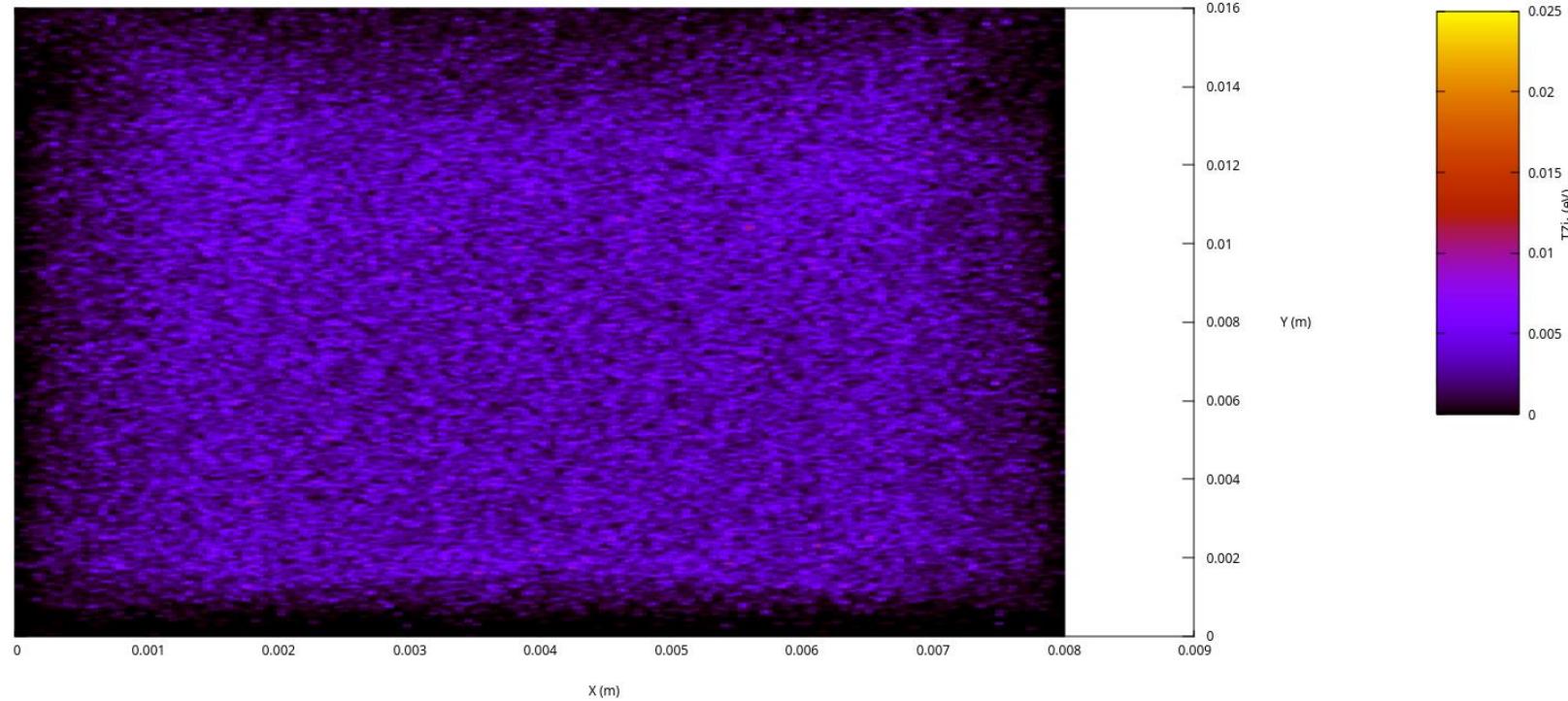


_NNNN_TZi_1_eV_2D.bin :: temperature along the Z direction of ion species 1 [eV]

```
gnuplot> set zran [*:*]; set cbran [*:*]
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'TZi_1 (eV)';
gnuplot> splot '_0007_TZi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

The temperatures are defined as
 $T_{x,y,z} = m \left(\langle v_{x,y,z}^2 \rangle - \langle v_{x,y,z} \rangle^2 \right)$.

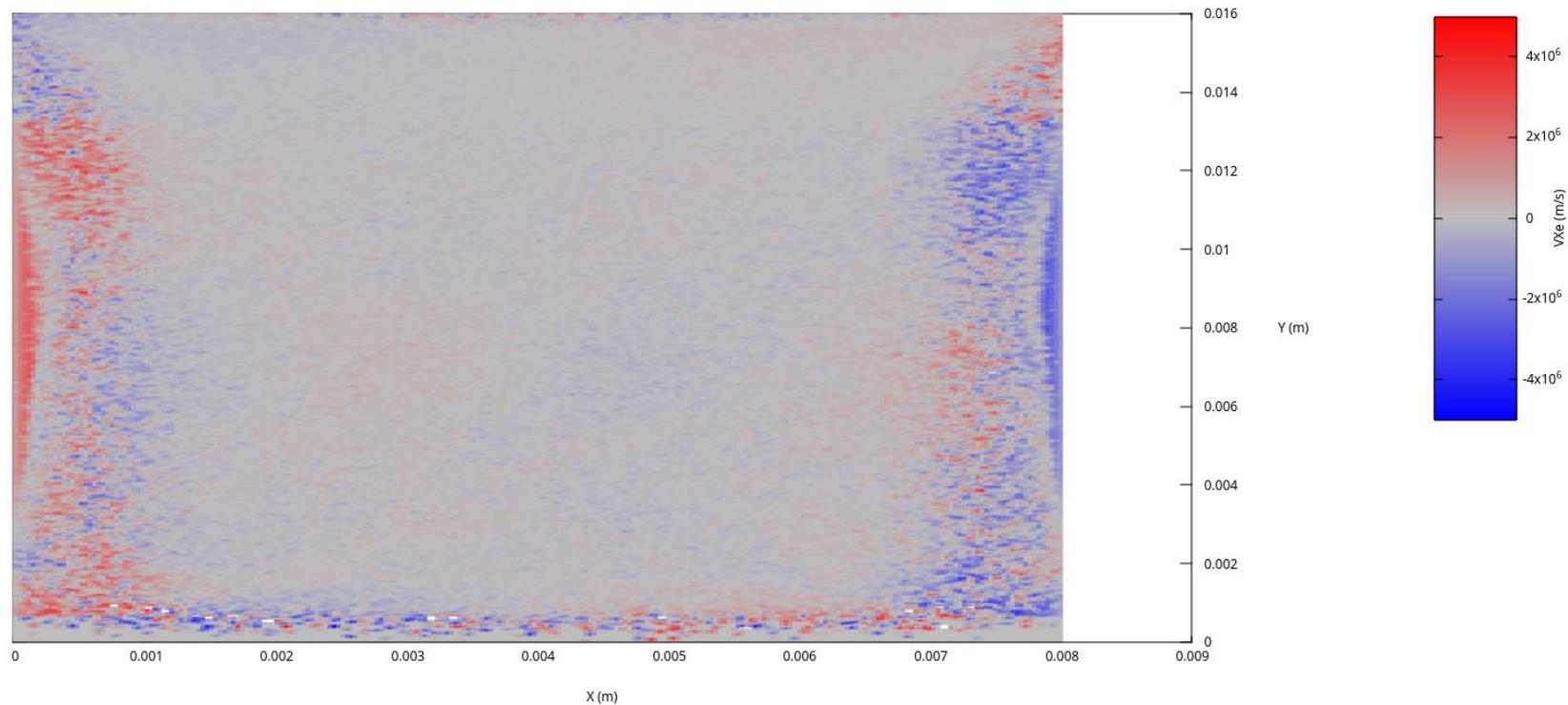
'_0007_TZi_1_eV_2D.bin' binary matrix u 1:2:3



_NNNN_VXe_ms_2D.bin :: average electron velocity along the X direction [m/s]

```
gnuplot> a=-5e6;b=5e6;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VXe (m/s)';
gnuplot> splot '_0007_VXe_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

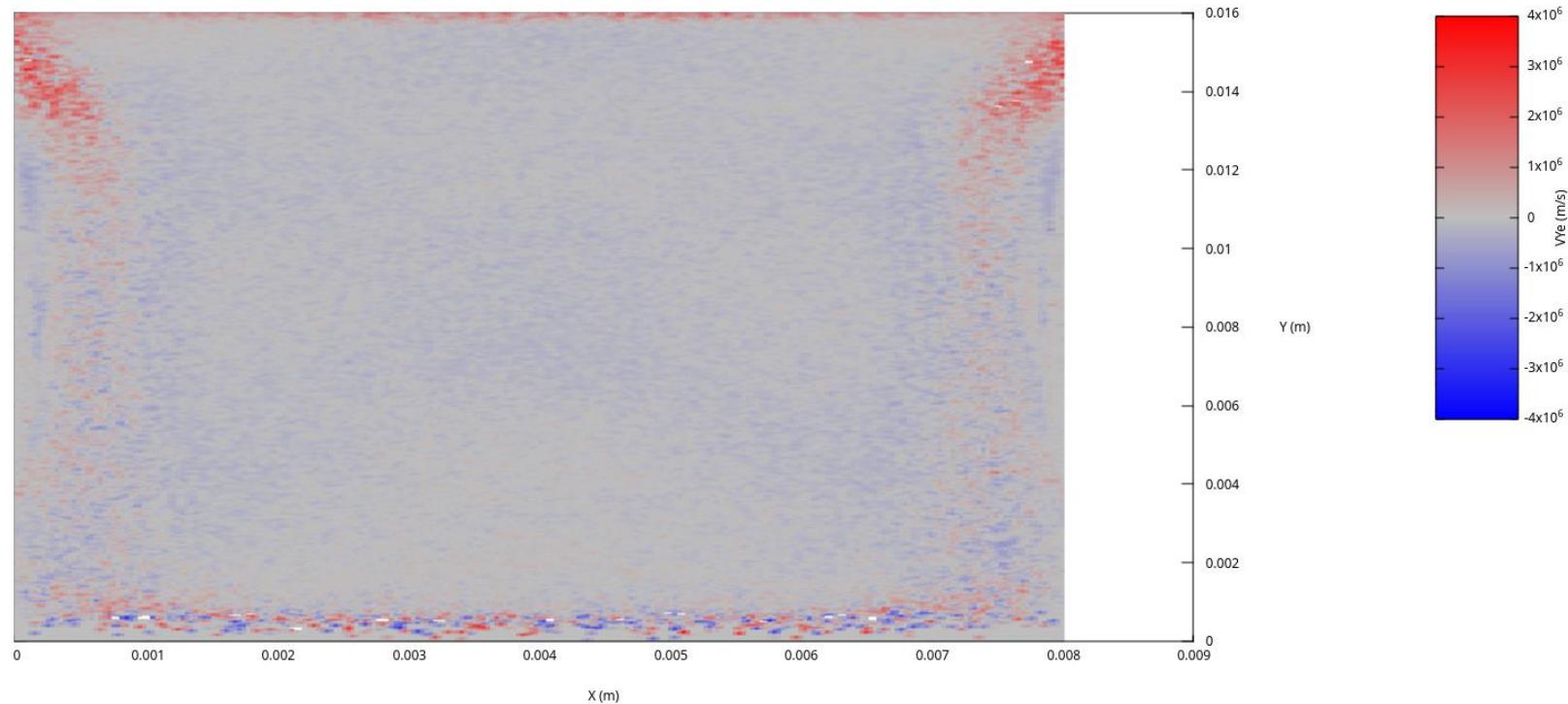
'_0007_VXe_ms_2D.bin' binary matrix u 1:2:3



_NNNN_VYe_ms_2D.bin :: average electron velocity along the Y direction [m/s]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VYe (m/s)';  
gnuplot> a=-4e6;b=4e6;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')  
gnuplot> splot '_0007_VYe_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

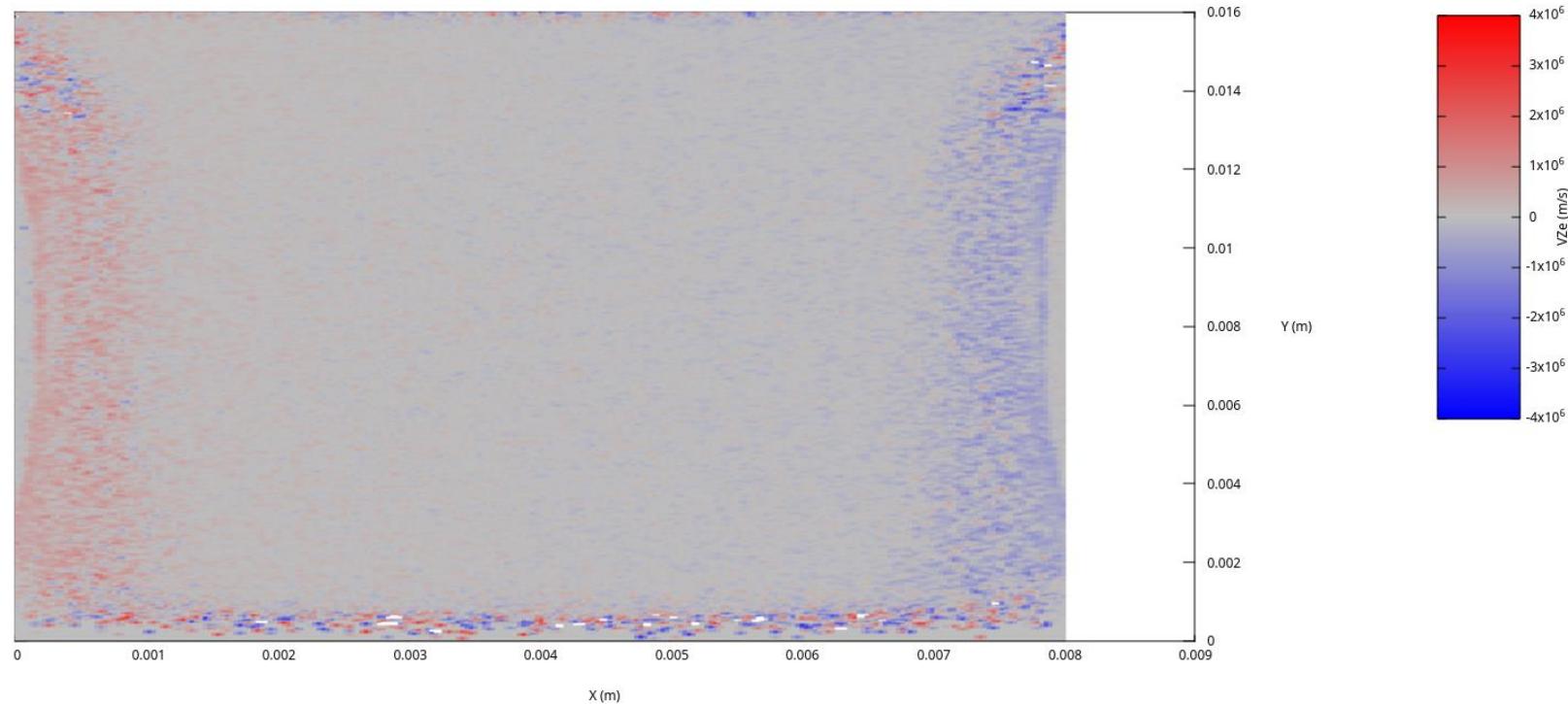
'_0007_VYe_ms_2D.bin' binary matrix u 1:2:3



_NNNN_VZe_ms_2D.bin :: average electron velocity along the Z direction [m/s]

```
gnuplot> a=-4e6;b=4e6;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VZe (m/s)';
gnuplot> splot '_0007_VZe_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

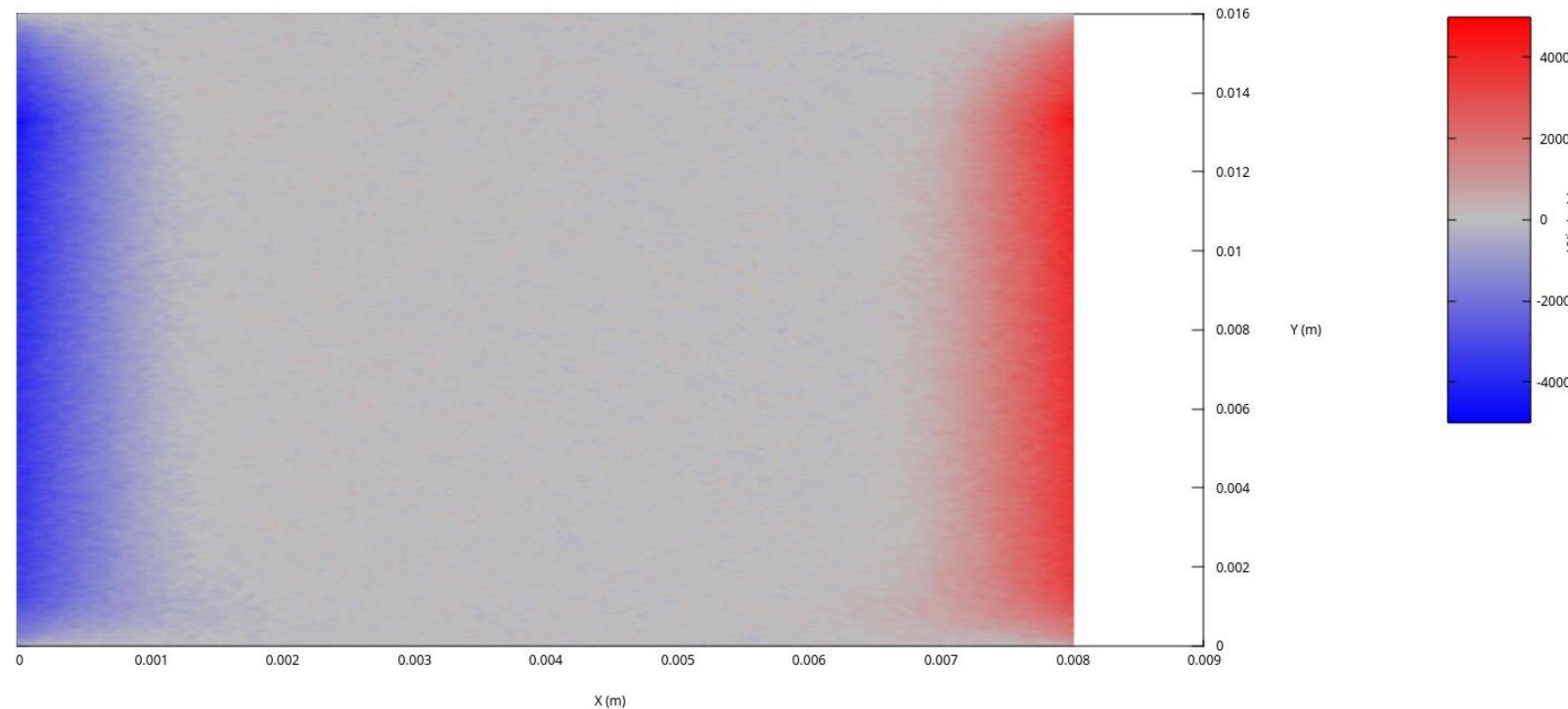
'_0007_VZe_ms_2D.bin' binary matrix u 1:2:3



NNNN_VXi_1_ms_2D.bin :: average velocity of ion species 1 along the X direction [m/s]

```
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VXi_1 (m/s)';  
gnuplot> a=-5e3;b=5e3;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')  
gnuplot> splot '_0007_VXi_1_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

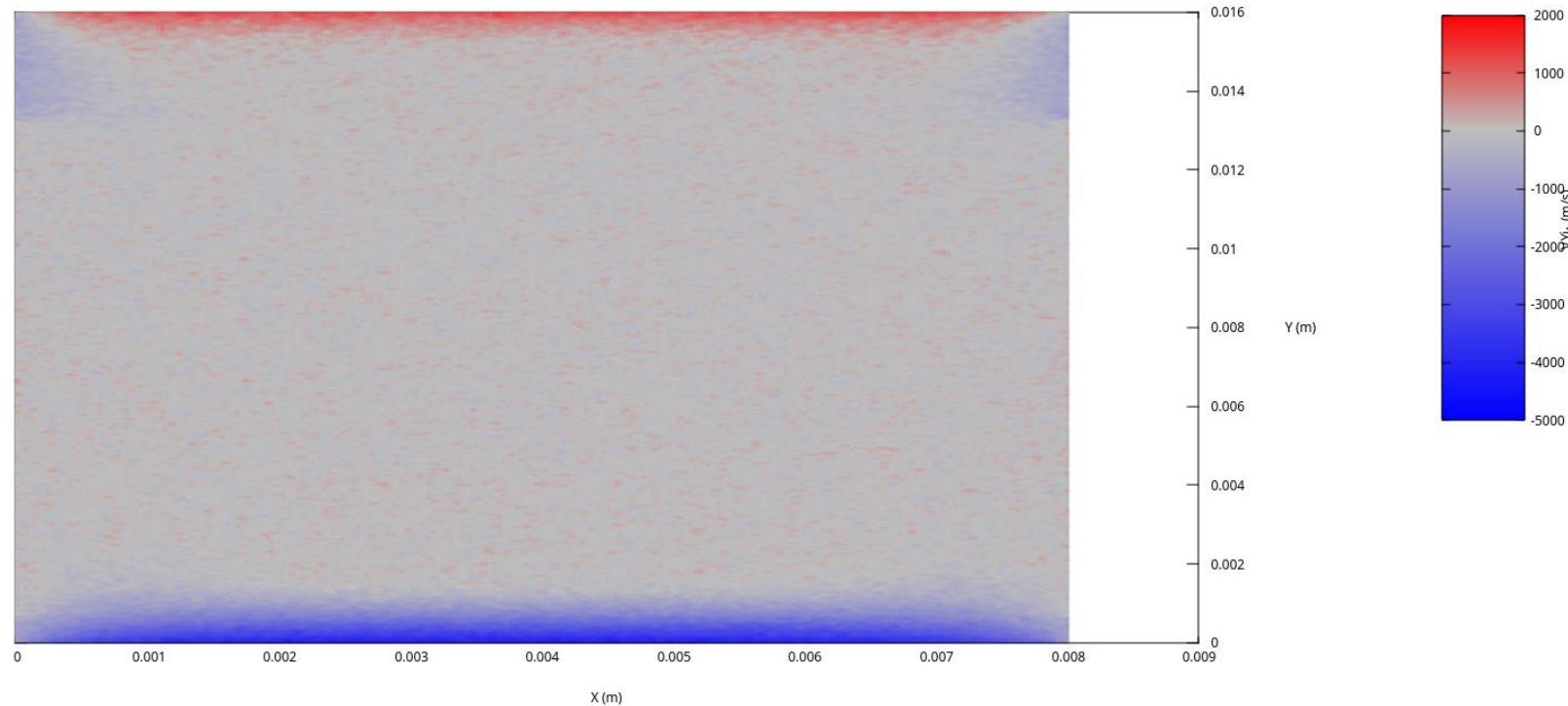
'_0007_VXi_1_ms_2D.bin' binary matrix u 1:2:3



_NNNN_VYi_1_ms_2D.bin :: average velocity of ion species 1 along the Y direction [m/s]

```
gnuplot> a=-5e3;b=2e3;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VYi_1 (m/s)';
gnuplot> splot '_0007_VYi_1_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

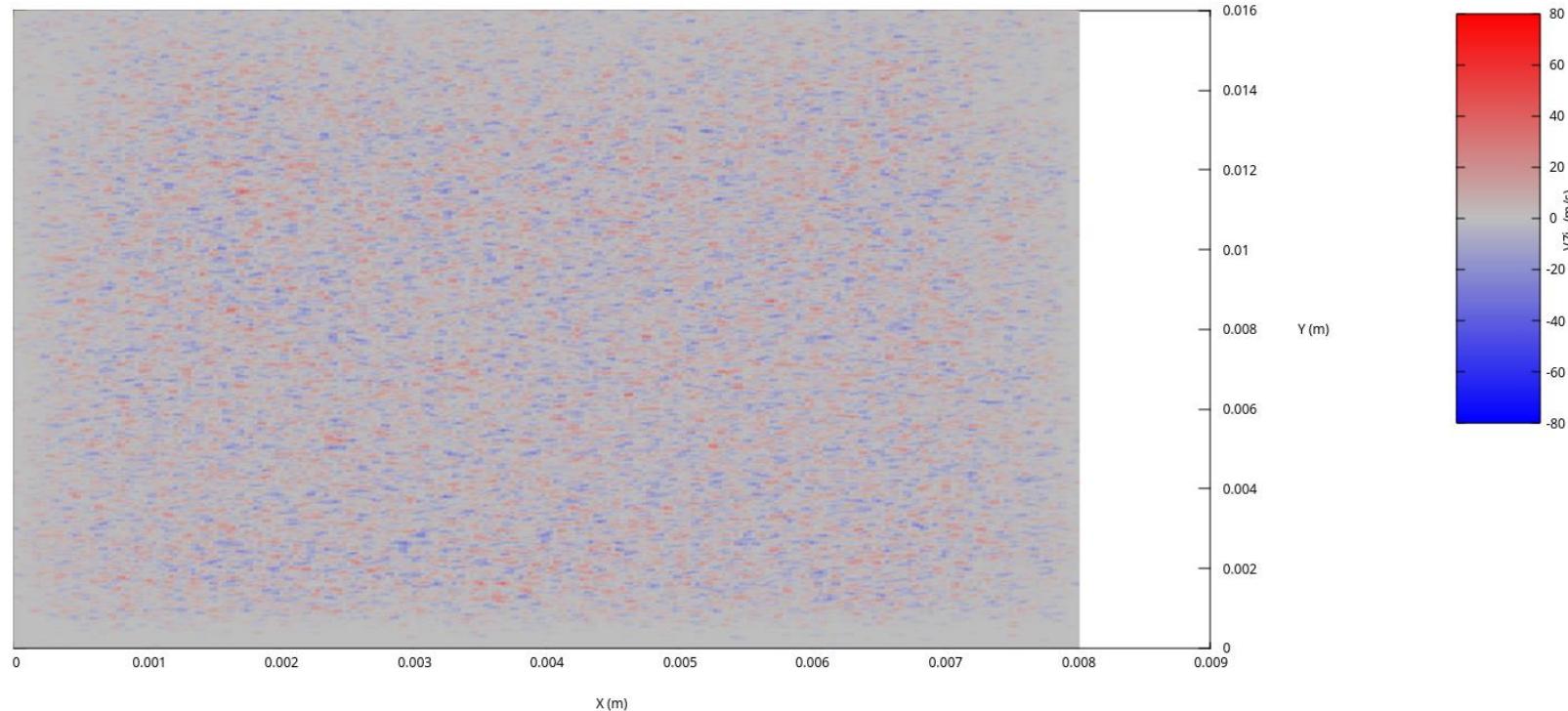
'_0007_VYi_1_ms_2D.bin' binary matrix u 1:2:3



_NNNN_VZi_1_ms_2D.bin :: average velocity of ion species 1 along the Z direction [m/s]

```
gnuplot> a=-80;b=80;set zran [a:b]; set cbran [a:b]; set palette defined (a 'blue', 0 'gray', b 'red')
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'VZi_1 (m/s)';
gnuplot> splot '_0007_VZi_1_ms_2D.bin' binary matrix u 1:2:3 w pm3d
```

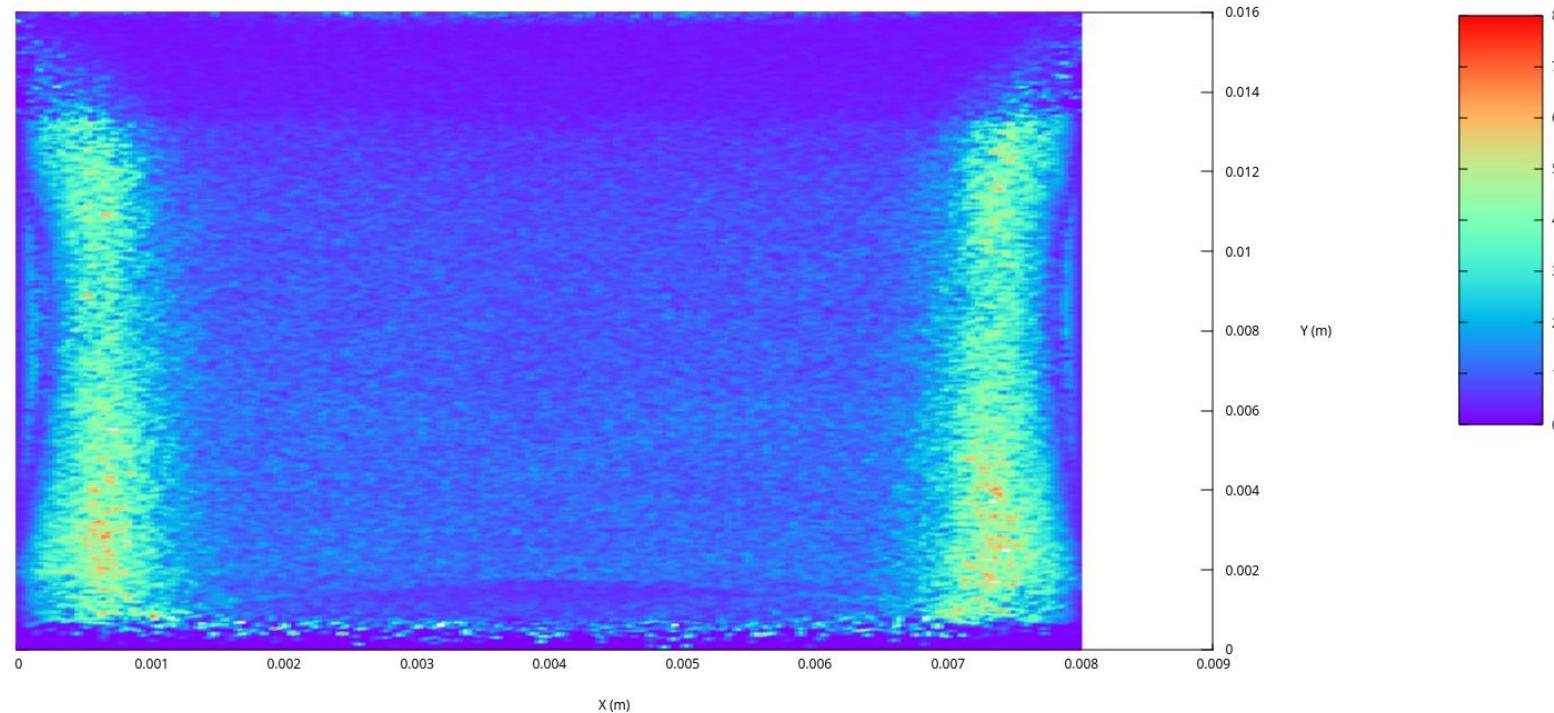
'_0007_VZi_1_ms_2D.bin' binary matrix u 1:2:3



_NNNN_WXe_eV_2D.bin :: average energy of electron motion along the X direction [eV]

```
gnuplot> set palette rgbformulae 33,13,10  
gnuplot> a=0;b=80;set zran [a:b]; set cbran [a:b];  
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WXe (eV)';  
gnuplot> splot '_0007_WXe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

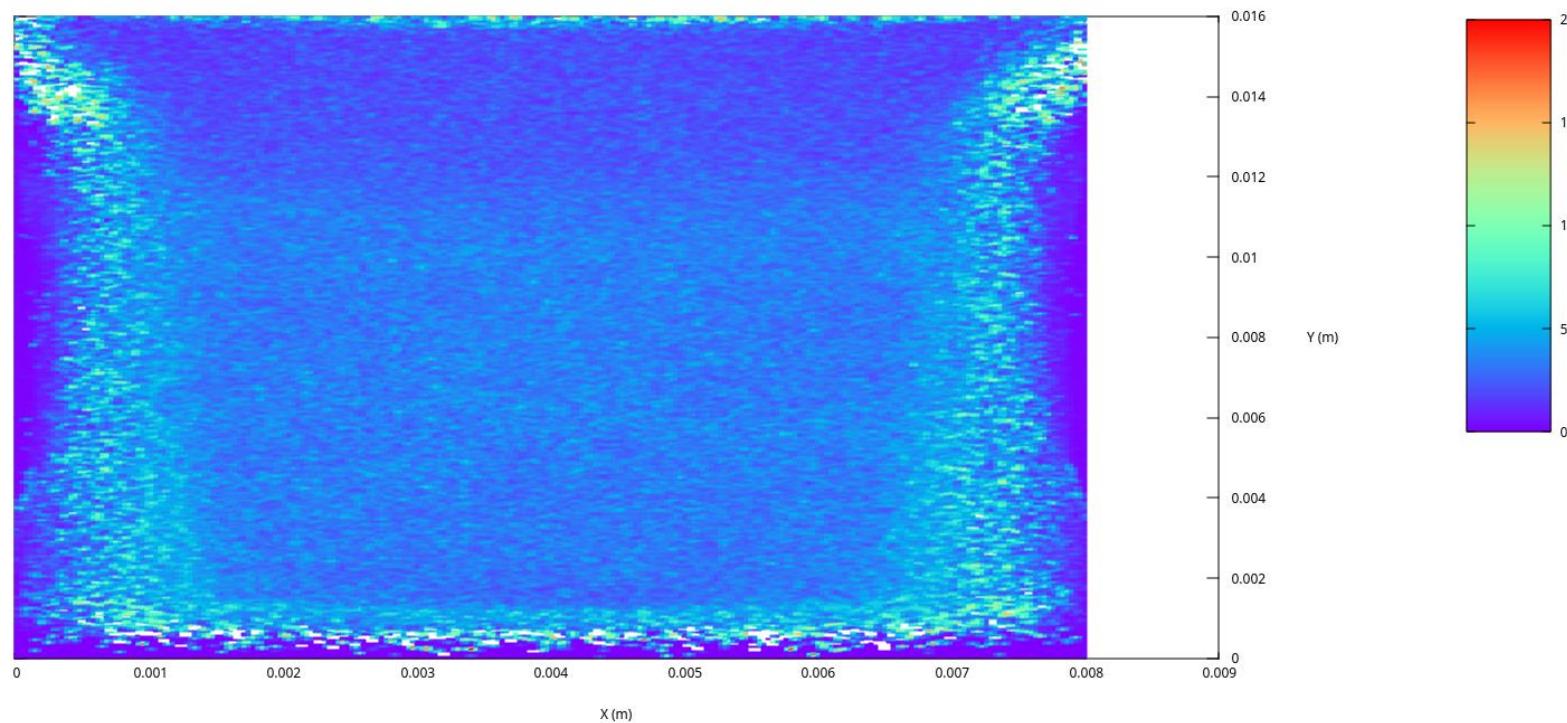
'_0007_WXe_eV_2D.bin' binary matrix u 1:2:3



_NNNN_WYe_eV_2D.bin :: average energy of electron motion along the Y direction [eV]

```
gnuplot> a=0;b=20;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WYe (eV)';
gnuplot> splot '_0007_WYe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

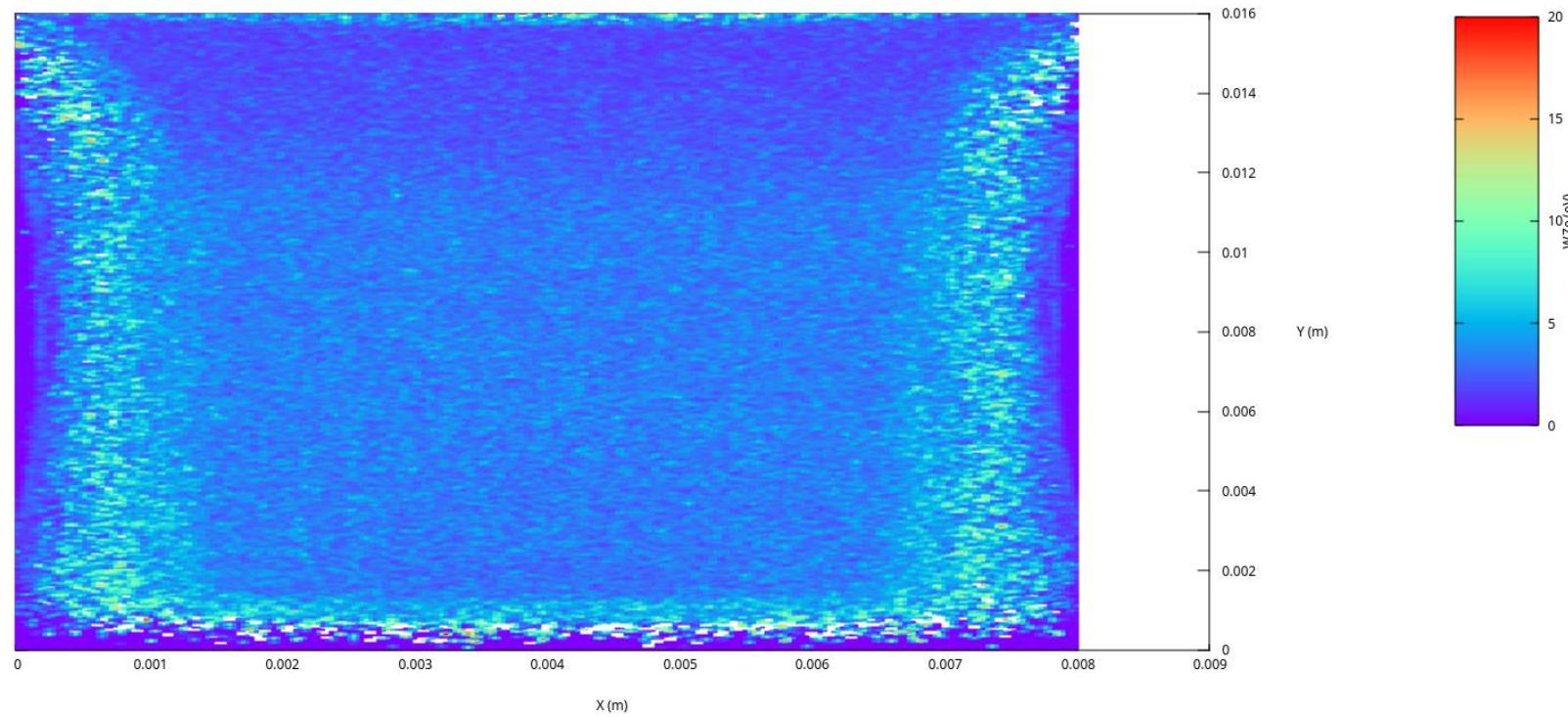
'_0007_WYe_eV_2D.bin' binary matrix u 1:2:3



_NNNN_WZe_eV_2D.bin :: average energy of electron motion along the Z direction [eV]

```
gnuplot> a=0;b=20;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WZe (eV)';
gnuplot> splot '_0007_WZe_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

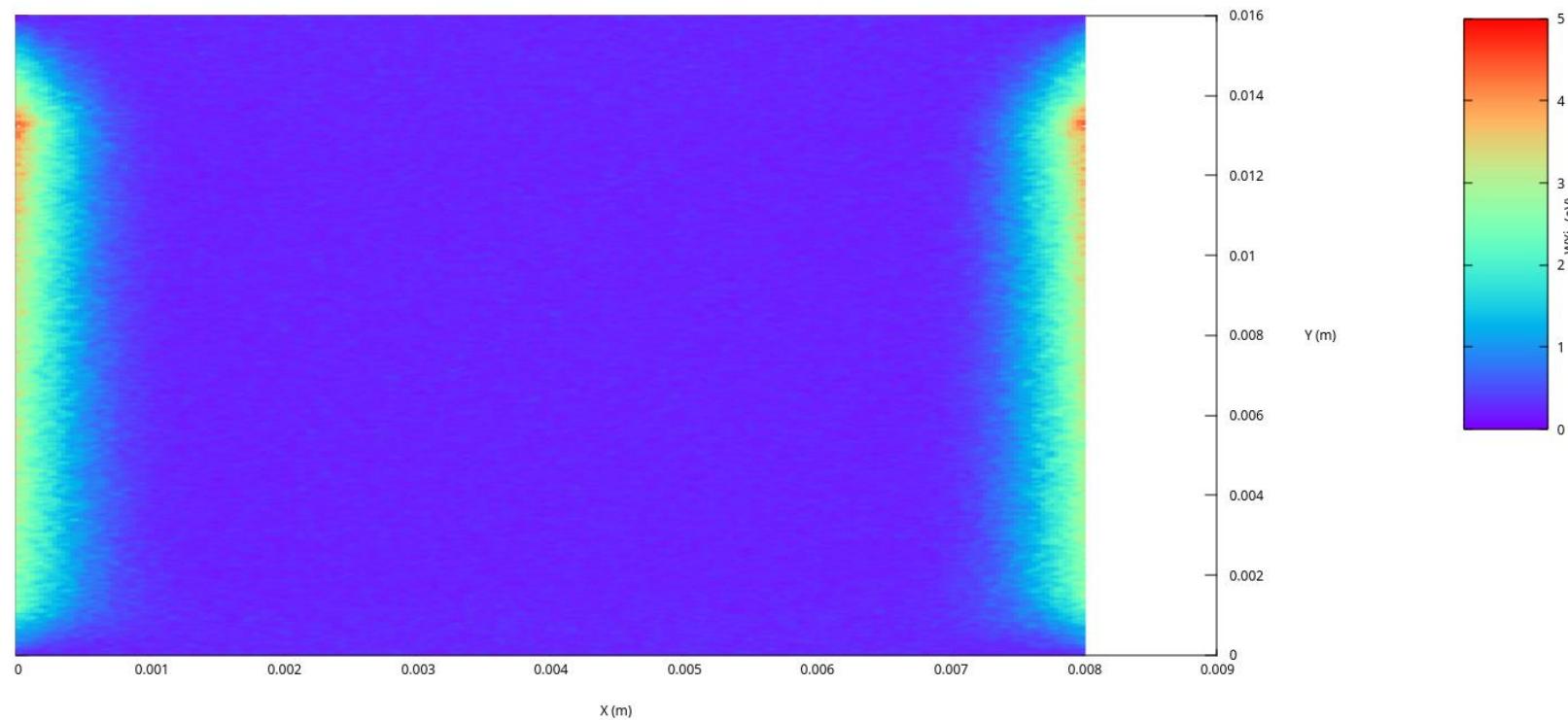
'_0007_WZe_eV_2D.bin' binary matrix u 1:2:3



_NNNN_WXi_1_eV_2D.bin :: average energy of motion of ion species 1 along the X direction [eV]

```
gnuplot> a=0;b=5;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WXi_1 (eV)';
gnuplot> splot '_0007_WXi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

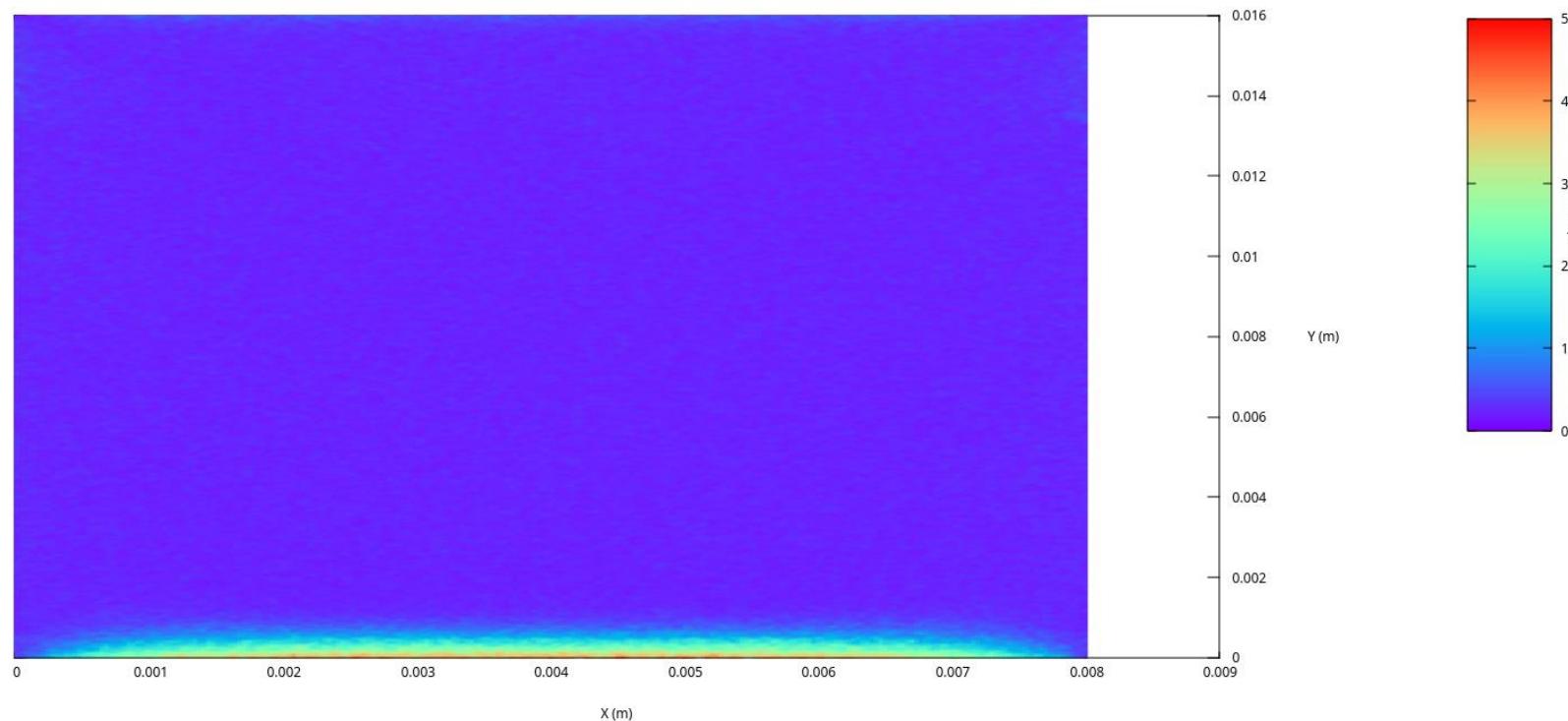
'_0007_WXi_1_eV_2D.bin' binary matrix u 1:2:3



_NNNN_WYi_1_eV_2D.bin :: average energy of motion of ion species 1 along the Y direction [eV]

```
gnuplot> a=0;b=5;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WYi_1 (eV)';
gnuplot> splot '_0007_WYi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

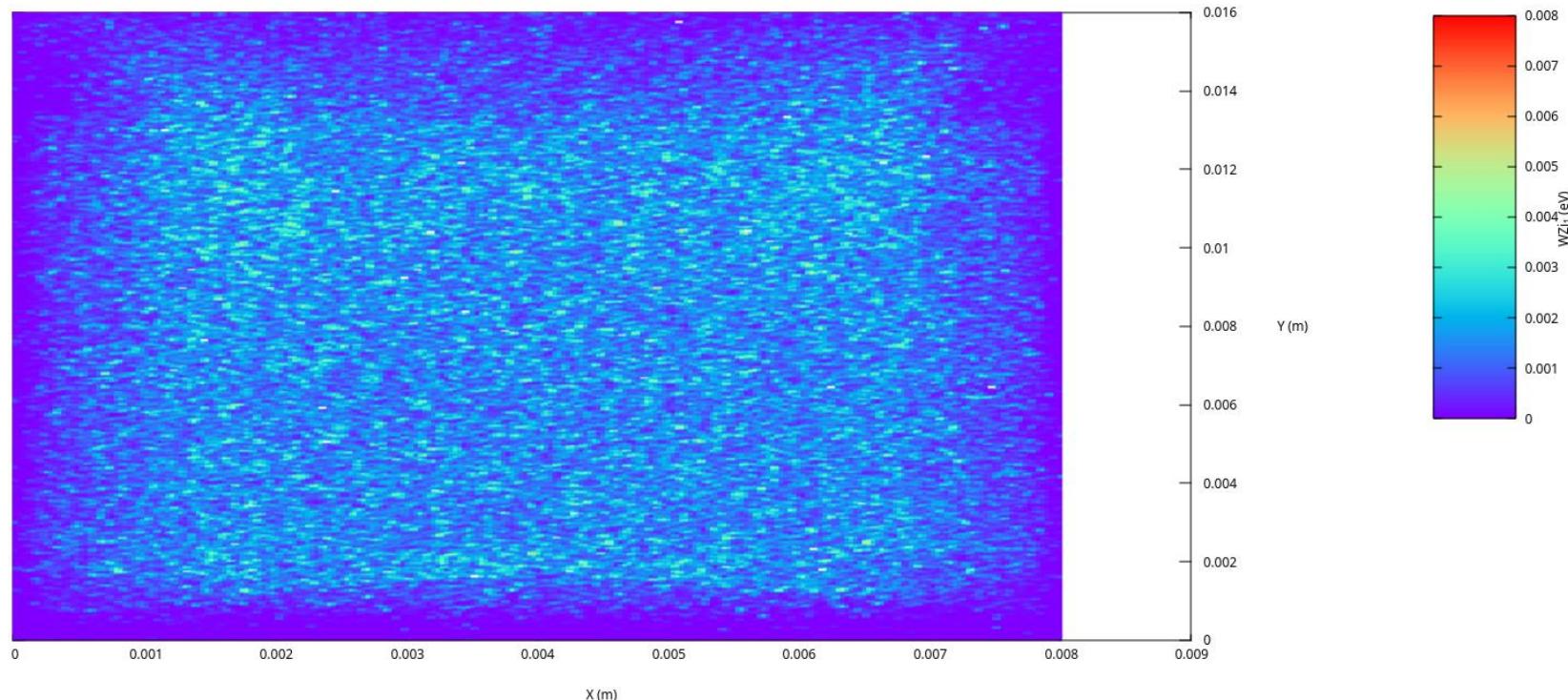
'_0007_WYi_1_eV_2D.bin' binary matrix u 1:2:3



_NNNN_WZi_1_eV_2D.bin :: average energy of motion of ion species 1 along the Z direction [eV]

```
gnuplot> a=0;b=0.008;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WZi_1 (eV)';
gnuplot> splot '_0007_WZi_1_eV_2D.bin' binary matrix u 1:2:3 w pm3d
```

'_0007_WZi_1_eV_2D.bin' binary matrix u 1:2:3



Note: initial ion velocities along Z were all zeros and in this setup the ions do not sense the magnetic field nor there is an electric field along Z, the **low but non-zero energy values** seen here are from ions produced in ionization collisions with a 3d (Vx, Vy, Vz) vector velocity corresponding to a Maxwellian distribution with the neutral temperature of 300 K.

Velocity distribution functions

- If requested, the code saves 1d [i.e. over one velocity component] velocity distribution functions (VDFs) for electrons and ions (snapshot file name `_NNNN_vdf1d.bin`) and/or 2d velocity distribution functions over VX and VY for electrons only (snapshot file name `_NNNN_vdf2d.bin`), here NNNN is the snapshot number.
- The original VDF files are binary. In order to get the VDF data in a format friendly for plotting software (gnuplot) one has to run a converting program:
 - `dataproc_convert_vdf1d_binary_to_asci.f90` for the 1d VDFs , and
 - `dataproc_convert_vdf2d_binary_to_asci.f90` for the 2d electron VDFs.
 - These programs can be found in github.com/PrincetonUniversity/EDIPIC-2D/tree/main/Postprocessing.

Example of output of the program processing 1d VDF binary files

```
./convertvdf1dbin2ascii.out
enter the number of the first snapshot
1
enter the number of the last snapshot
7
reading binary file _0001_vdf1d.bin complete
created file _0001_vdf_boxes.asc
created file _0001_evxdf.asc
created file _0001_eyydf.asc
created file _0001_evydf.asc
created file _0001_evzdf.asc
created file _0001_i_1_vxdf.asc
created file _0001_i_1_vydf.asc
created file _0001_i_1_vzdf.asc
file _0002_vdf1d.bin not found, skip
file _0003_vdf1d.bin not found, skip
file _0004_vdf1d.bin not found, skip
file _0005_vdf1d.bin not found, skip
file _0006_vdf1d.bin not found, skip
reading binary file _0007_vdf1d.bin complete
created file _0007_vdf_boxes.asc
created file _0007_evxdf.asc
created file _0007_eyydf.asc
created file _0007_evydf.asc
created file _0007_evzdf.asc
created file _0007_i_1_vxdf.asc
created file _0007_i_1_vydf.asc
created file _0007_i_1_vzdf.asc
```

Text files produced by data processing of 1d VDF binary files

- In this example, if the 1d VDF converter is applied to `_0007_vdf1d.bin` it would produce the following files:

```
-rw-rw-r-- 1 dima dima 91484 Feb 6 18:03 _0007_evxdf.asc  
-rw-rw-r-- 1 dima dima 91484 Feb 6 18:03 _0007_veydf.asc  
-rw-rw-r-- 1 dima dima 91484 Feb 6 18:03 _0007_evzdf.asc  
-rw-rw-r-- 1 dima dima 585186 Feb 6 18:03 _0007_i_1_vxdf.asc  
-rw-rw-r-- 1 dima dima 585186 Feb 6 18:03 _0007_i_1_vydf.asc  
-rw-rw-r-- 1 dima dima 585186 Feb 6 18:03 _0007_i_1_vzdf.asc  
-rw-rw-r-- 1 dima dima 8037 Feb 6 18:03 _0007_vdf_boxes.asc
```

1d electron VDFs over VX, VY, and VZ, respectively

1d VDFs for ion species 1 over VX, VY, and VZ, respectively

Spatial boxes used to calculate the VDFs

Each converted VDF file contains VDFs from all the spatial boxes.

Here, for example, is the header of file `_0007_evxdf.asc`:

```
# col 1 is the X-velocity [middle of the bin, units of v_th_e for Te = 4.000 eV which is 1186188.8 m/s]  
# the columns listed below contain [dim-less] number of macroparticles in the velocity bin calculated in the following box :  
# col 2 is for box 1 x-index 1 y-index 1 | xmin 0.000000 [m] xmax 0.002011 [m] | ymin 0.000000 ymax 0.002011 [m]  
.....  
# col 33 is for box 32 x-index 4 y-index 8 | xmin 0.006001 [m] xmax 0.008012 [m] | ymin 0.013980 ymax 0.015991 [m]
```

And here is the header of file `_0007_i_1_vxdf.asc`:

```
# col 1 is the X-velocity [middle of the bin, units of v_th_i for Ti = 4.000 eV and Ms = 40.00 (amu) which is 4392.8 m/s]  
# the columns listed below contain [dim-less] number of macroparticles in the velocity bin calculated in the following box :  
# col 2 is for box 1 x-index 1 y-index 1 | xmin 0.000000 [m] xmax 0.002011 [m] | ymin 0.000000 ymax 0.002011 [m]  
.....  
# col 33 is for box 32 x-index 4 y-index 8 | xmin 0.006001 [m] xmax 0.008012 [m] | ymin 0.013980 ymax 0.015991 [m]
```

Example of output of the program processing 2d VDF binary files

```
./convertvdf2dbin2ascii.out
enter the number of the first snapshot
1
enter the number of the last snapshot
7
reading binary file _0001_vdf2d.bin complete
created file _0001_vdf_boxes.asc
created file _0001_0001_evxvydf.asc
created file _0001_0002_evxvydf.asc
.....
created file _0001_0031_evxvydf.asc
created file _0001_0032_evxvydf.asc
file _0002_vdf2d.bin not found, skip
file _0003_vdf2d.bin not found, skip
file _0004_vdf2d.bin not found, skip
file _0005_vdf2d.bin not found, skip
file _0006_vdf2d.bin not found, skip
reading binary file _0007_vdf2d.bin complete
created file _0007_vdf_boxes.asc
created file _0007_0001_evxvydf.asc
created file _0007_0002_evxvydf.asc
.....
created file _0007_0031_evxvydf.asc
created file _0007_0032_evxvydf.asc
```

Text files produced by data processing of 2d VDF binary files

- In this example, if the 2d VDF converter is applied to `_0007_vdf2d.bin` it would produce the following files:

```
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0001_evxvydf.asc
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0002_evxvydf.asc
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0003_evxvydf.asc
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0004_evxvydf.asc
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0005_evxvydf.asc
.....
.....
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0031_evxvydf.asc
-rw-rw-r-- 1 dima dima 1844030 Feb 6 18:39 _0007_0032_evxvydf.asc
-rw-rw-r-- 1 dima dima 8037 Feb 6 18:39 _0007_vdf_boxes.asc
```

2d electron VDFs over VX and VY in 32 locations (numbered from 0001 to 0032)

Spatial boxes used to calculate the VDFs. Note that most likely the boxes used for the 1d and the 2d VDFs are the same, but it is not necessary always so.

A 2d VDF for each location (spatial box) is saved into a separate file with a name `_NNNN_LLLL_evxvydf.asc`, where NNNN is the snapshot number, LLLL is the number of the VDF spatial box. The data are saved in three columns: VX, VY, VDF function value.

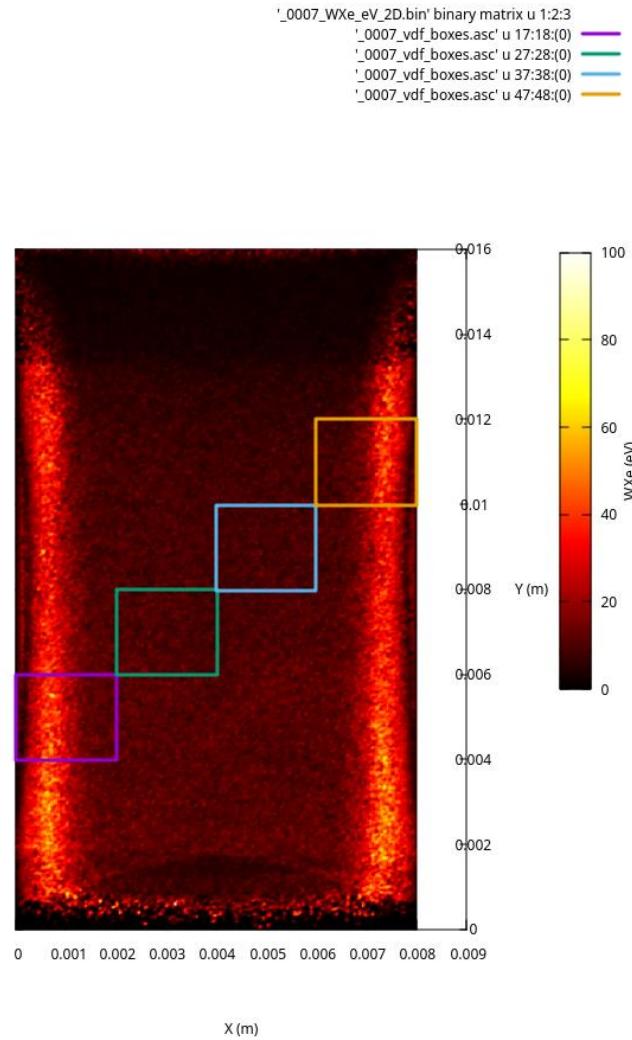
_0007_vdf_boxes.asc :: spatial boxes used to calculate the VDFs

This file has a header (shown below) explaining which columns should be used to plot a desired spatial box. Examples are given in the following slides.

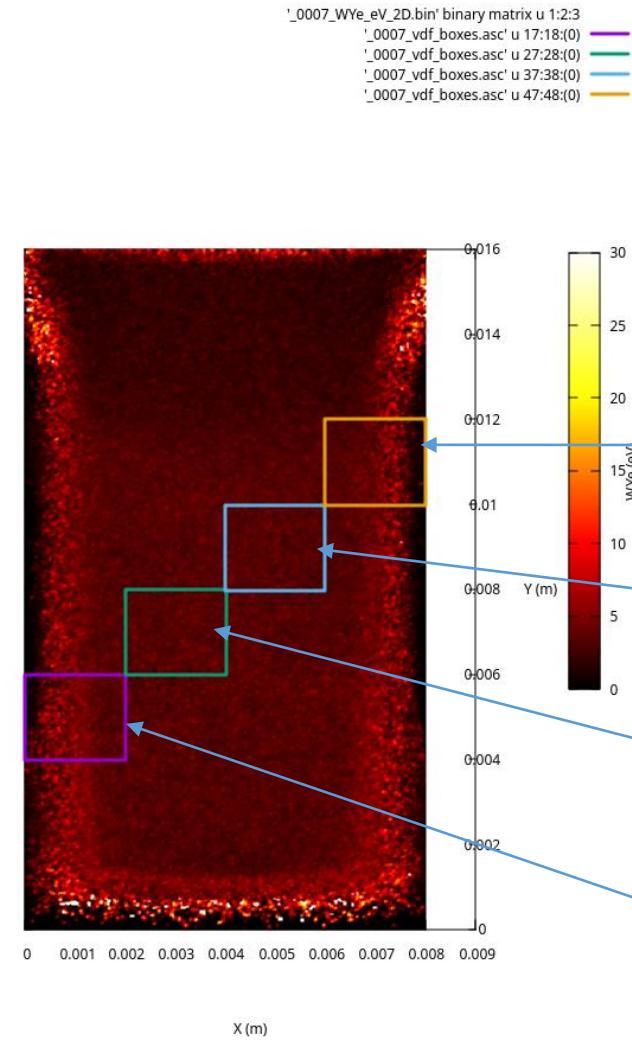
```
# columns    1      2 are box    1 x-index    1 y-index    1 | xmin  0.000000 [m] xmax  0.002011 [m] | ymin  0.000000 ymax  0.002011 [m]
# columns    3      4 are box    2 x-index    2 y-index    1 | xmin  0.002011 [m] xmax  0.004023 [m] | ymin  0.000000 ymax  0.002011 [m]
.....
# columns   17     18 are box    9 x-index    1 y-index    3 | xmin  0.000000 [m] xmax  0.002011 [m] | ymin  0.003989 ymax  0.006001 [m]
.....
# columns   27     28 are box   14 x-index    2 y-index    4 | xmin  0.002011 [m] xmax  0.004023 [m] | ymin  0.006001 ymax  0.008012 [m]
.....
# columns   37     38 are box   19 x-index    3 y-index    5 | xmin  0.003989 [m] xmax  0.006001 [m] | ymin  0.007979 ymax  0.009990 [m]
.....
# columns   47     48 are box   24 x-index    4 y-index    6 | xmin  0.006001 [m] xmax  0.008012 [m] | ymin  0.009990 ymax  0.012002 [m]
.....
# columns   61     62 are box   31 x-index    3 y-index    8 | xmin  0.003989 [m] xmax  0.006001 [m] | ymin  0.013980 ymax  0.015991 [m]
# columns   63     64 are box   32 x-index    4 y-index    8 | xmin  0.006001 [m] xmax  0.008012 [m] | ymin  0.013980 ymax  0.015991 [m]
```

These boxes are plotted in the next slide. Note that box 9 requires columns 17,18, box 14 requires columns 27,28, etc.

Average energy of electron motion along the X direction



Average energy of electron motion along the Y direction



Gnuplot commands which plot these figures are given in the next slide.

To plot the average energy of electron motion along the X direction with VDF spatial boxes #9, 14, 19, 24 (see the previous slide, left panel):

```
gnuplot> set palette rgbformulae 21,22,23
gnuplot> set view 0, 0
gnuplot> a=0;b=100;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WXe (eV)';
gnuplot> splot '_0007_WXe_ev_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_vdf_boxes.asc' u 17:18:(0) w 1 lt 1 lw 3,
'_0007_vdf_boxes.asc' u 27:28:(0) w 1 lt 2 lw 3, '_0007_vdf_boxes.asc' u 37:38:(0) w 1 lt 3 lw 3, '_0007_vdf_boxes.asc' u
47:48:(0) w 1 lt 4 lw 3
```

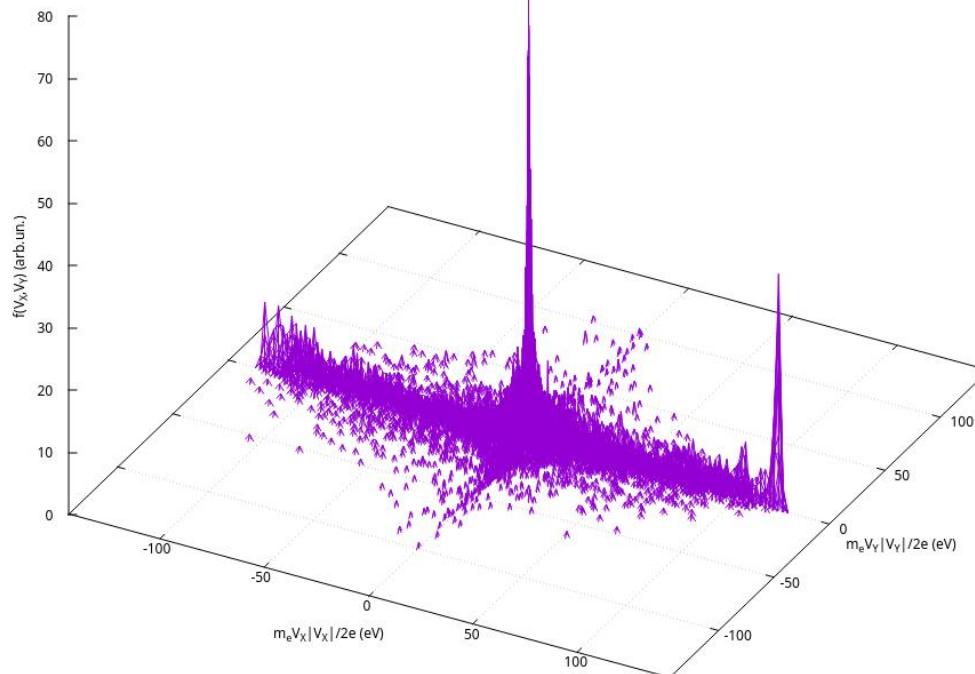
To plot the average energy of electron motion along the Y direction with VDF spatial boxes #9, 14, 19, 24 (see the previous slide, right panel):

```
gnuplot> set palette rgbformulae 21,22,23
gnuplot> set view 0, 0
gnuplot> a=0;b=100;set zran [a:b]; set cbran [a:b];
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WXe (eV)';
gnuplot> splot '_0007_WXe_ev_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_vdf_boxes.asc' u 17:18:(0) w 1 lt 1 lw 3,
'_0007_vdf_boxes.asc' u 27:28:(0) w 1 lt 2 lw 3, '_0007_vdf_boxes.asc' u 37:38:(0) w 1 lt 3 lw 3, '_0007_vdf_boxes.asc' u
47:48:(0) w 1 lt 4 lw 3
```

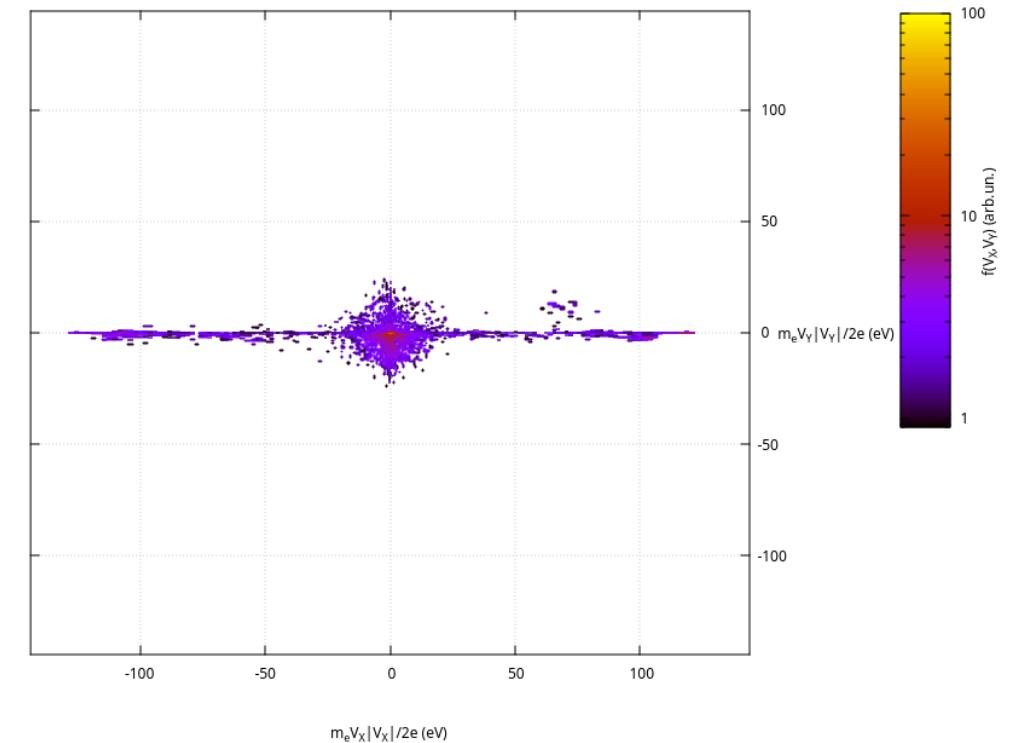
_0007_0009_evxvydf.asc :: 2d electron velocity distribution function for box 9, snapshot #7

The electron VDF over VX and VY plotted as a function of energy of motion along the X and Y direction.

```
'_0007_0009_evxvydf.asc' u ($1*abs($1)*Te_scale_eV);($2*abs($2)*Te_scale_eV);3
```



```
'_0007_0009_evxvydf.asc' u ($1*abs($1)*Te_scale_eV);($2*abs($2)*Te_scale_eV);3
```



Gnuplot commands which plot these figures are in the next slide.

To plot the electron VDF over VX and VY from box 9 as function of energy of motion along the X and Y directions representing the surface with lines (see the previous slide, left panel):

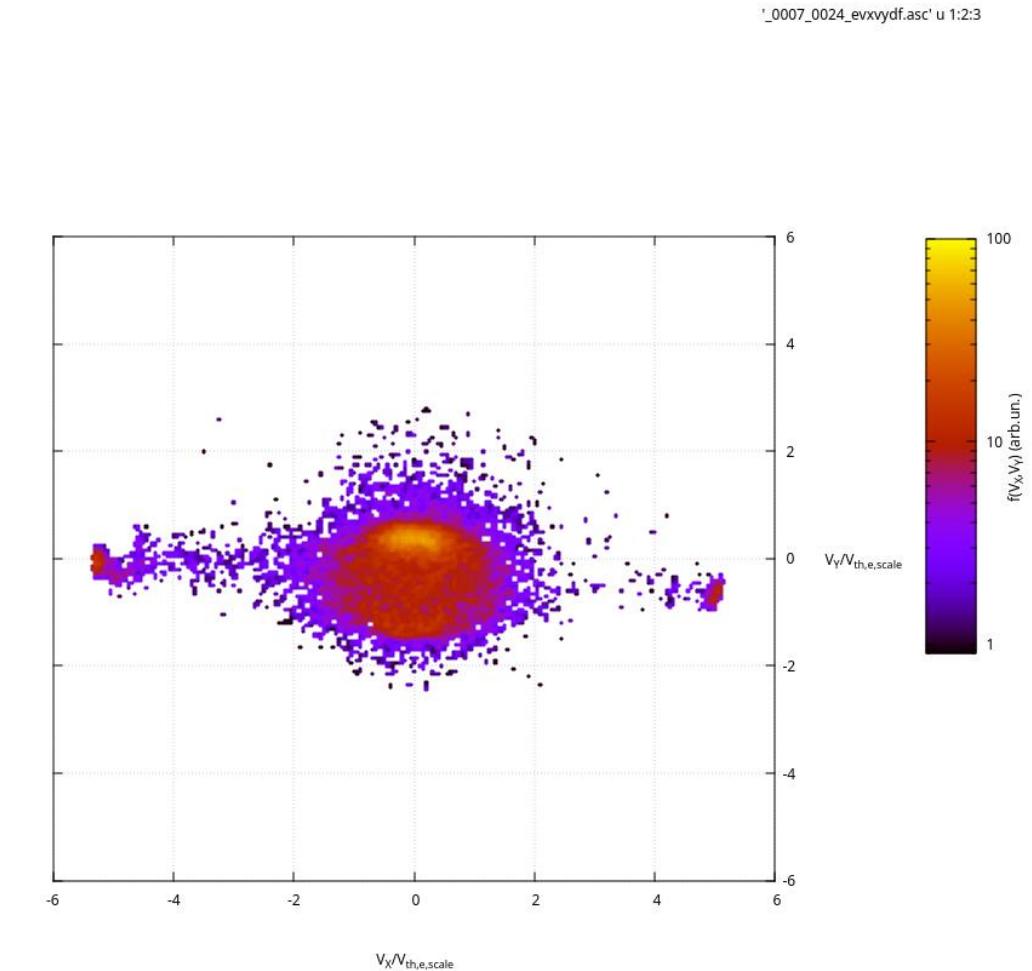
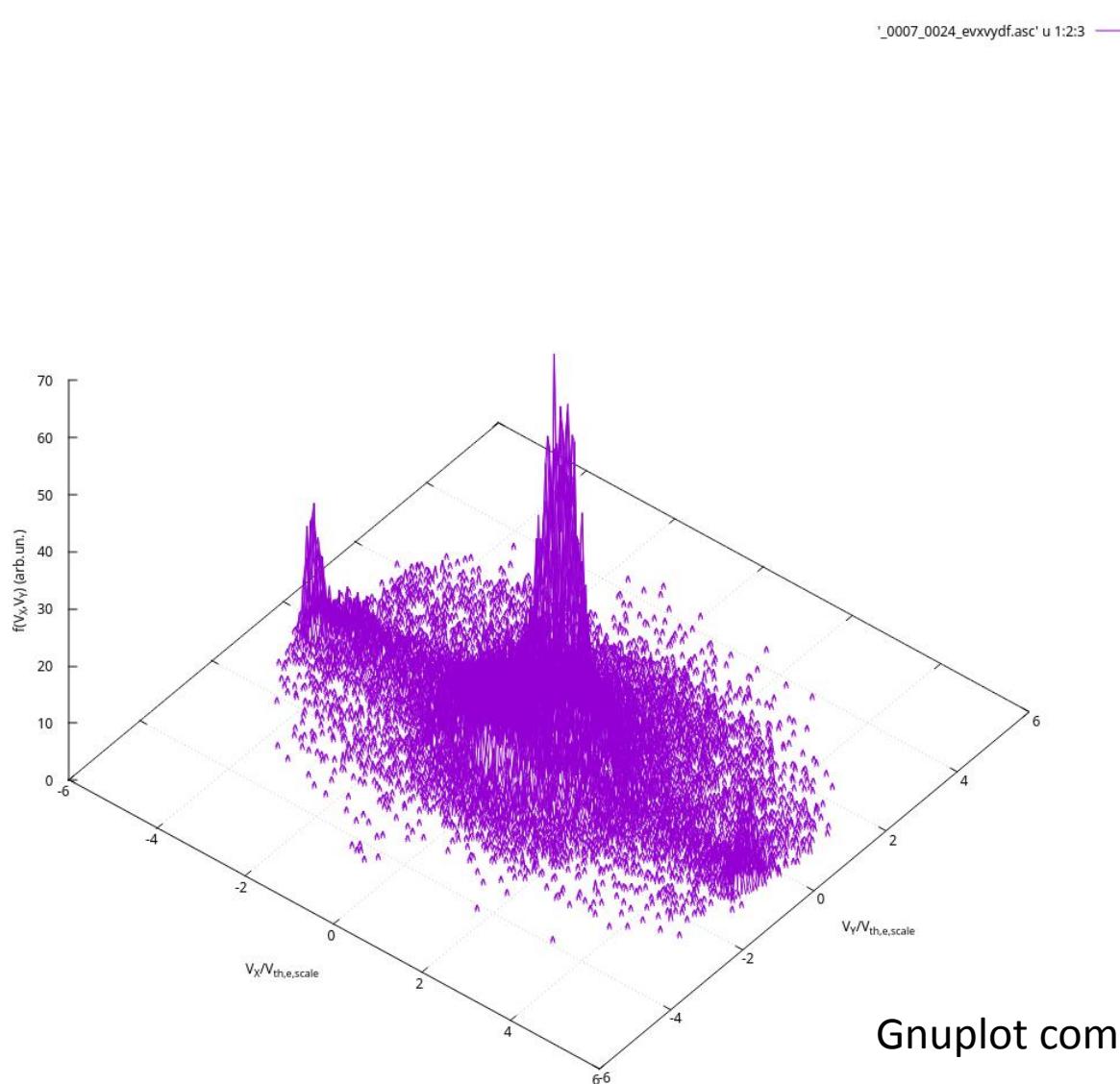
```
gnuplot> Te_scale_eV=4.0
gnuplot> set xran [-6**2*Te_scale_eV:6**2*Te_scale_eV]; set yran [-6**2*Te_scale_eV:6**2*Te_scale_eV]
gnuplot> set xlabel 'm_eV_X|V_X|/2e (eV)'; set ylabel 'm_eV_Y|V_Y|/2e (eV)'; set zlabel 'f(V_X,V_Y) (arb.un.)' rotate by 90
gnuplot> splot '_0007_0009_evxvydf.asc' u ($1*abs($1)*Te_scale_eV):($2*abs($2)*Te_scale_eV):3 w l
```

To plot the electron VDF over VX and VY from box 9 as function of energy of motion along the X and Y directions representing the surface with colormap (see the previous slide, right panel):

```
gnuplot> unset zlabel
gnuplot> set view 0, 0;
gnuplot> set zran [0.9:100]; set cbran [0.9:100]
gnuplot> set log z; set log cb
gnuplot> set palette rgbformulae 7,5,15
gnuplot> set xlabel 'm_eV_X|V_X|/2e (eV)'; set ylabel 'm_eV_Y|V_Y|/2e (eV)'; set cblabel 'f(V_X,V_Y) (arb.un.)'
gnuplot> splot '_0007_0009_evxvydf.asc' u ($1*abs($1)*Te_scale_eV):($2*abs($2)*Te_scale_eV):3 w pm3d
```

_0007_0024_evxvydf.asc :: 2d electron velocity distribution function for box 24, snapshot #7

The electron VDF over VX and VY plotted as a function of the velocity components in units of the thermal electron velocity for the scale temperature.



Gnuplot commands which plot these figures are in the next slide.

To plot the electron VDF over VX and VY from box 24 as function of the velocity components in units of the thermal electron velocity for the scale temperature, representing the surface with lines (see the previous slide, left panel):

```
gnuplot> unset log z; unset log cb  
gnuplot> set zran [0.1:*]; rep  
gnuplot> set xran [-6:6]; set yran [-6:6];  
gnuplot> set xlabel 'V_X/V_{th,e,scale}' ; set ylabel 'V_Y/V_{th,e,scale}' ; set zlabel 'f(V_X,V_Y) (arb.un.)' rotate by 90  
gnuplot> splot '_0007_0024_evxvydf.asc' u 1:2:3 w l
```

To plot the electron VDF over VX and VY from box 24 as function of the velocity components in units of the thermal electron velocity for the scale temperature, representing the surface with colormap (see the previous slide, right panel):

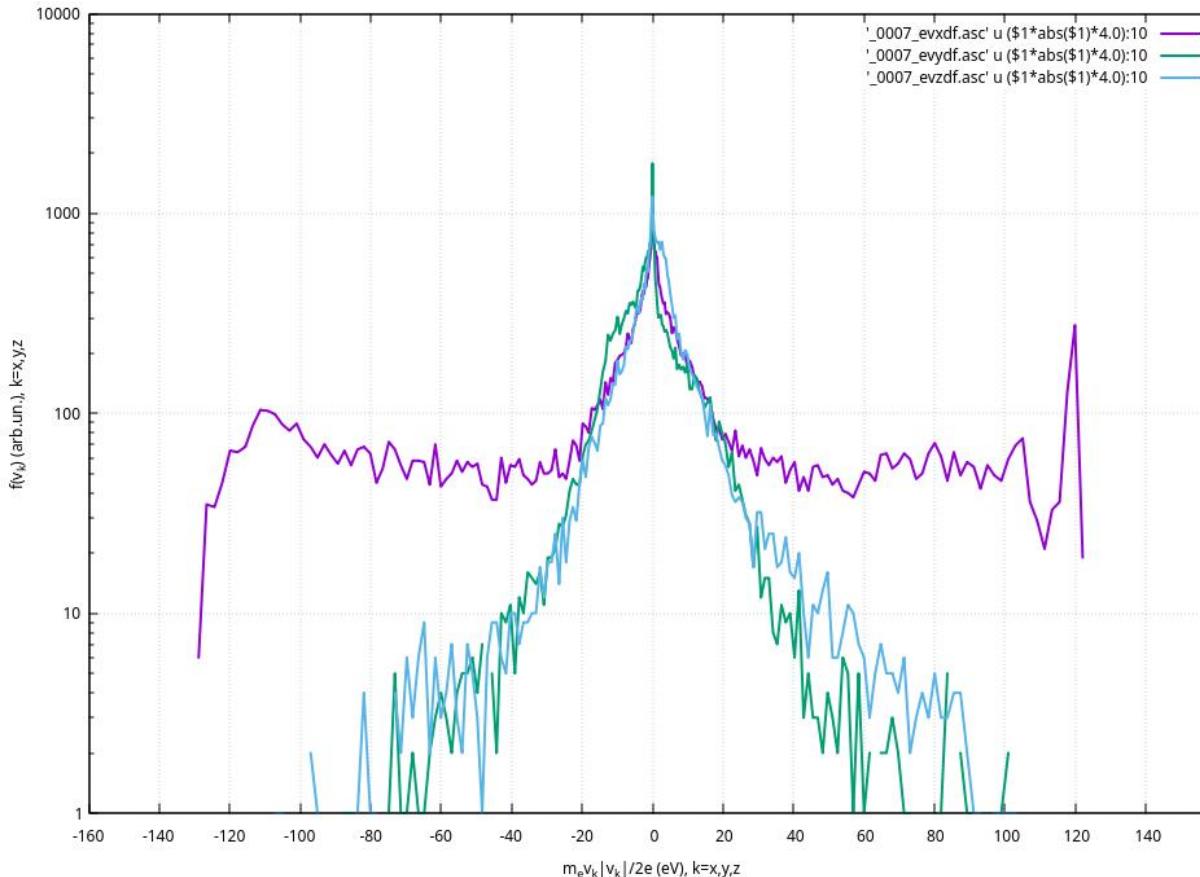
```
gnuplot> unset zlabel  
gnuplot> set view 0, 0  
gnuplot> set zran [0.9:100]; set cbran [0.9:100]  
gnuplot> set log z; set log cb  
gnuplot> set xran [-6:6]; set yran [-6:6]  
gnuplot> set xlabel 'V_X/V_{th,e,scale}' ; set ylabel 'V_Y/V_{th,e,scale}' ; set cblabel 'f(V_X,V_Y) (arb.un.)'  
gnuplot> splot '_0007_0024_evxvydf.asc' u 1:2:3 w pm3d
```

_0007_evxdf.asc :: 1d electron velocity distribution function over VX, snapshot #7
_0007_eyydf.asc :: 1d electron velocity distribution function over VY, snapshot #7
_0007_ezzdf.asc :: 1d electron velocity distribution function over VZ, snapshot #7

```

gnuplot> Te_scale_eV=4.0
gnuplot> set xran [-6**2*Te_scale_eV:6**2*Te_scale_eV]; set yran [1:1e4]
gnuplot> set xtics 20
gnuplot> set xlabel 'm_ev_k|v_k|/2e (eV), k=x,y,z' ; set ylabel 'f(v_k) (arb.un.), k=x,y,z'
gnuplot> plot '_0007_evxdf.asc' u ($1*abs($1)*4.0):10 w l lw 2, '_0007_eyydf.asc' u ($1*abs($1)*4.0):10 w l lw 2,
'_0007_ezzdf.asc' u ($1*abs($1)*4.0):10 w l lw 2

```



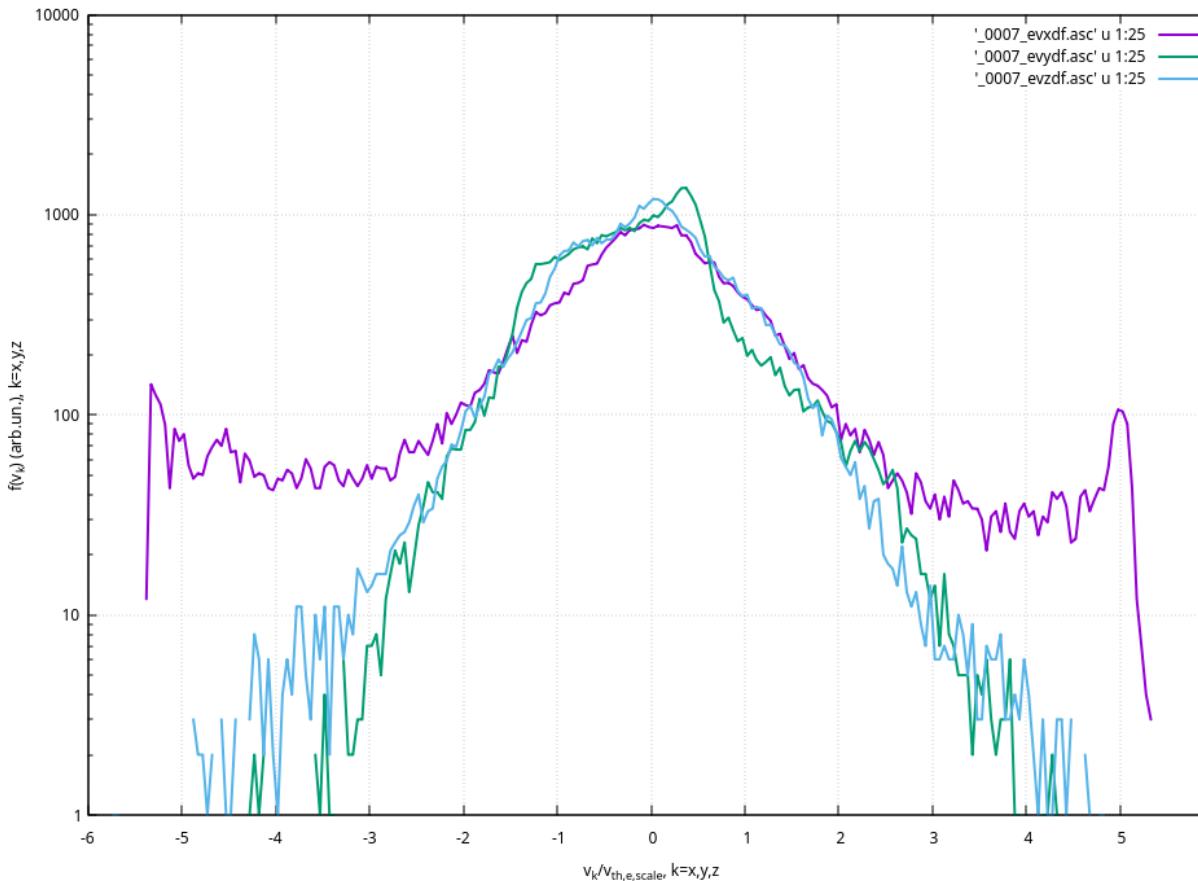
Here the 1d electron VDFs from box 10 are plotted as functions of the energy of electron motion along the corresponding (X,Y, or Z) direction. The negative energy values correspond to the motion in the negative direction of the coordinate axis.

_0007_evxdf.asc :: 1d electron velocity distribution function over VX, snapshot #7
_0007_eyydf.asc :: 1d electron velocity distribution function over VY, snapshot #7
_0007_ezzdf.asc :: 1d electron velocity distribution function over VZ, snapshot #7

```

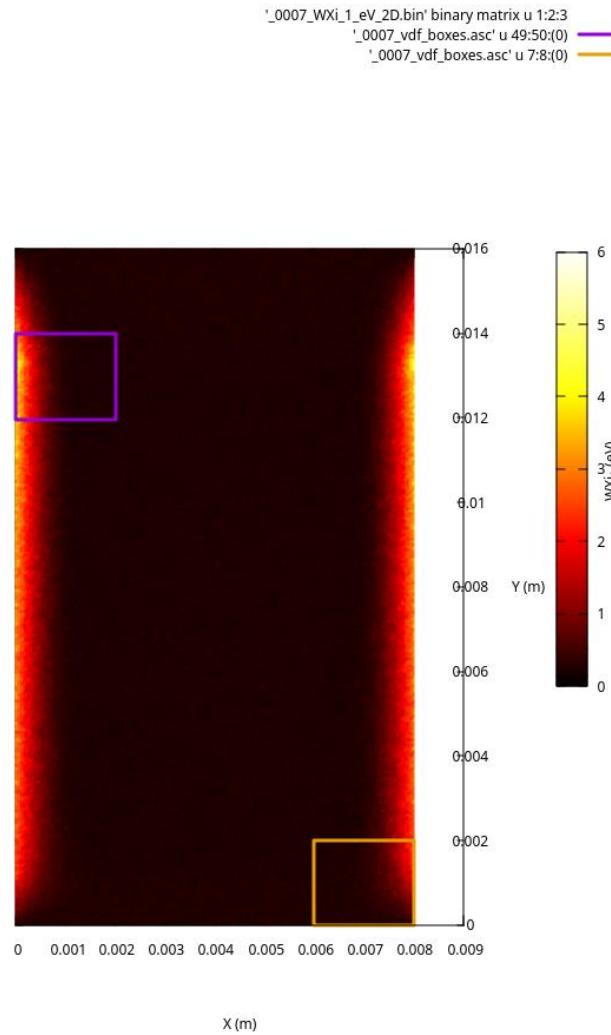
gnuplot> set xran [-6:6]
gnuplot> set xtics 1
gnuplot> set xran [-6:6]; set yran [1:1e4]
gnuplot> set xlabel 'v_k/v_{th,e,scale}, k=x,y,z'; set ylabel 'f(v_k) (arb.un.), k=x,y,z'
gnuplot> plot '_0007_evxdf.asc' u 1:25 w l lw 2, '_0007_eyydf.asc' u 1:25 w l lw 2, '_0007_ezzdf.asc' u 1:25 w l lw 2

```

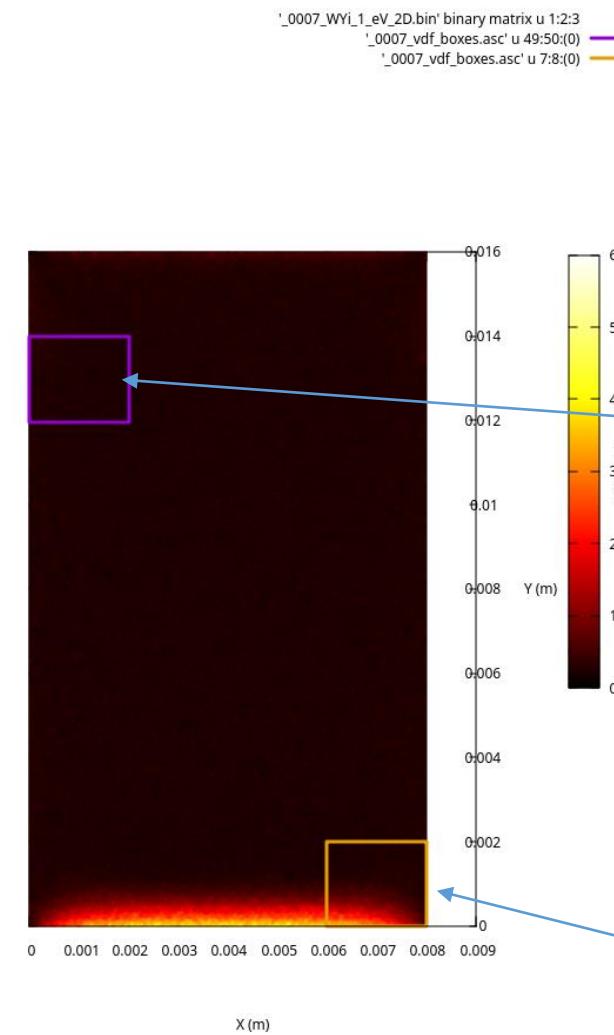


Here the 1d electron VDFs from box 24 are plotted as functions of the corresponding velocity components (VX,VY, or VZ) in units of the electron thermal velocity for the scale electron temperature.

Average energy of ion motion of species 1 along the X direction



Average energy of ion motion of species 1 along the Y direction



Gnuplot commands which plot these figures are given in the next slide.

To plot the average energy of motion of ion species 1 along the X direction with VDF spatial boxes #4 and 25 (see the previous slide, left panel):

```
gnuplot> set view 0, 0
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WXi_1 (eV)'
gnuplot> set palette rgbformulae 21,22,23
gnuplot> a=0;b=6;set zran [a:b]; set cbran [a:b]
gnuplot> splot '_0007_WXi_1_ev_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_vdf_boxes.asc' u 49:50:(0) w l lt 1 lw 3,
'_0007_vdf_boxes.asc' u 7:8:(0) w l lt 4 lw 3
```

To plot the average energy of motion of ion species 1 along the X direction with VDF spatial boxes #4 and 25 (see the previous slide, right panel):

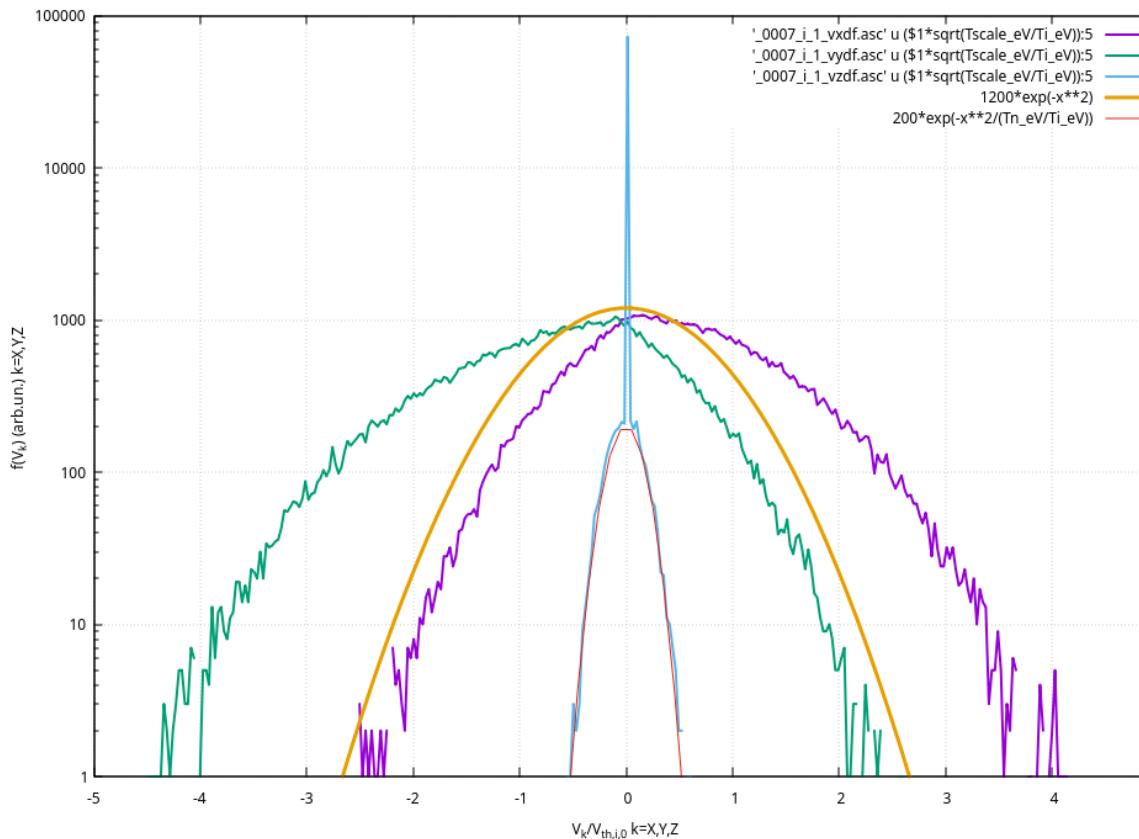
```
gnuplot> set view 0, 0
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'WYi_1 (eV)'
gnuplot> set palette rgbformulae 21,22,23
gnuplot> a=0;b=6;set zran [a:b]; set cbran [a:b]
gnuplot> splot '_0007_WYi_1_ev_2D.bin' binary matrix u 1:2:3 w pm3d, '_0007_vdf_boxes.asc' u 49:50:(0) w l lt 1 lw 3,
'_0007_vdf_boxes.asc' u 7:8:(0) w l lt 4 lw 3
```

`_0007_i_1_vxdf.asc :: 1d velocity distribution function of ion species 1 over VX, snapshot #7`
`_0007_i_1_vydf.asc :: 1d velocity distribution function of ion species 1 over VY, snapshot #7`
`_0007_i_1_vzdf.asc :: 1d velocity distribution function of ion species 1 over VZ, snapshot #7`

```

gnuplot> set xran [-5:5]
gnuplot> Ti_eV=0.5
gnuplot> Tscale_eV=4.0
gnuplot> Tn_eV=1.38e-23*300/1.6e-19
gnuplot> set yran [1:*)
gnuplot> set xlabel 'V_k/V_{th,i,0} k=X,Y,Z'; set ylabel 'f(V_k) (arb.un.) k=X,Y,Z'
gnuplot> plot '_0007_i_1_vxdf.asc' u ($1*sqrt(Tscale_eV/Ti_eV)):5 w 1 lw 2, '_0007_i_1_vydf.asc' u ($1*sqrt(Tscale_eV/Ti_eV)):5 w 1 lw 2, '_0007_i_1_vzdf.asc' u ($1*sqrt(Tscale_eV/Ti_eV)):5 w 1 lw 2, 1200*exp(-x**2) lw 3, 200*exp(-x**2/(Tn_eV/Ti_eV)) lw 1 lt 7

```



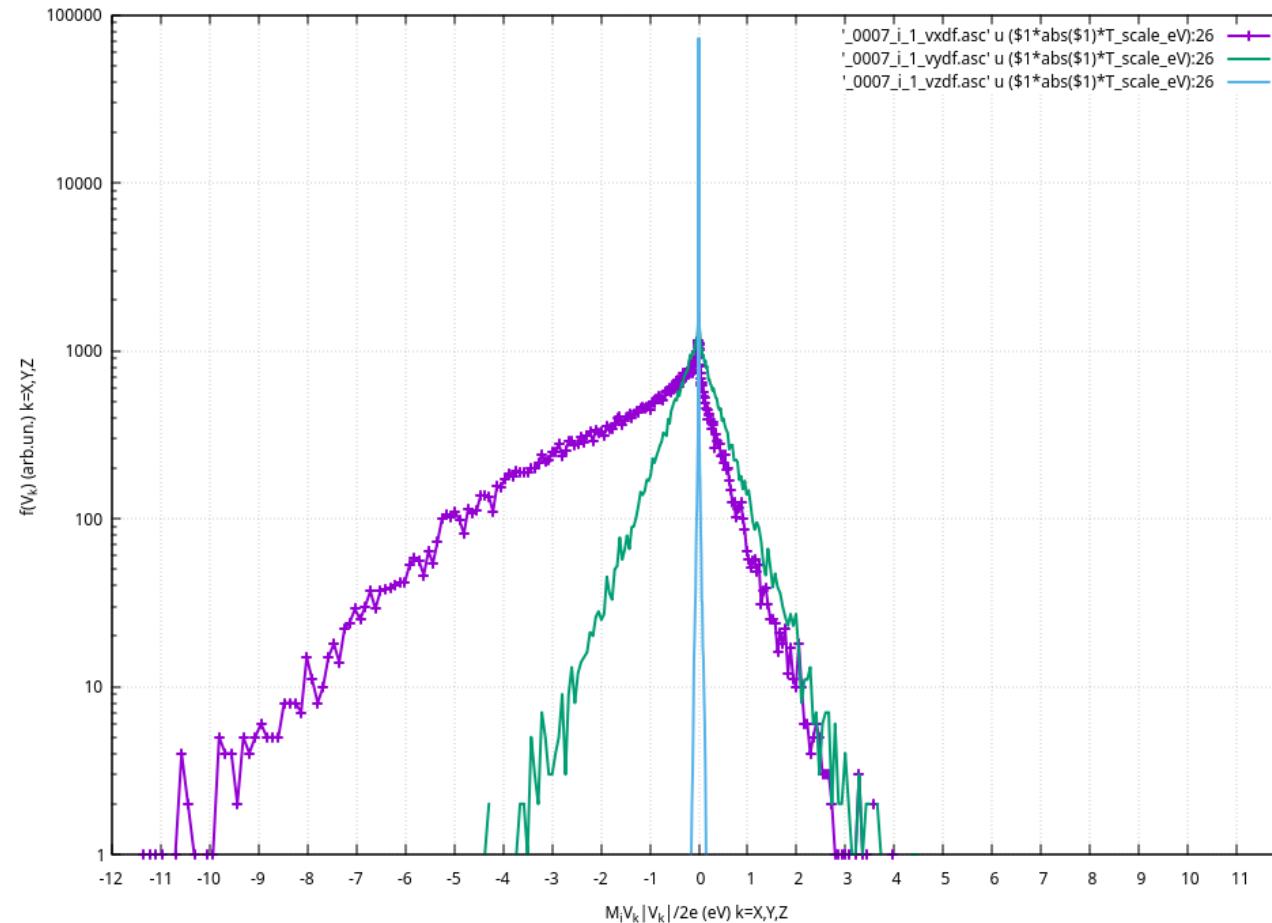
Here the 1d VDFs of ion species 1 from box 4 are plotted as functions of the corresponding velocity components (VX,VY, or VZ) in units of the ion thermal velocity for the temperature equal to the initial ion temperature (0.5 eV). Note that the original velocity bins are in units of the ion thermal velocity for the temperature equal to the scale electron temperature (4 eV). The dark yellow curve shows a Maxwellian VDF with the initial ion temperature, and the thin red curve represent a Maxwellian VDF with the temperature of neutrals (300 K).

_0007_i_1_vxdf.asc :: 1d velocity distribution function of ion species 1 over VX, snapshot #7
_0007_i_1_vydf.asc :: 1d velocity distribution function of ion species 1 over VY, snapshot #7
_0007_i_1_vzdf.asc :: 1d velocity distribution function of ion species 1 over VZ, snapshot #7

```

gnuplot> set xran [-12:12]
gnuplot> set xtics 1
gnuplot> T_scale_eV=4.0
gnuplot> set xlabel 'M_i V_k |V_k|/2e (eV) k=X,Y,Z'; set ylabel 'f(V_k) (arb.un.) k=X,Y,Z'
gnuplot> plot '_0007_i_1_vxdf.asc' u ($1*abs($1)*T_scale_eV):26 w linesp lw 2, '_0007_i_1_vydf.asc' u ($1*abs($1)*T_scale_eV):26 w 1 lw 2, '_0007_i_1_vzdf.asc' u ($1*abs($1)*T_scale_eV):26 w 1 lw 2

```



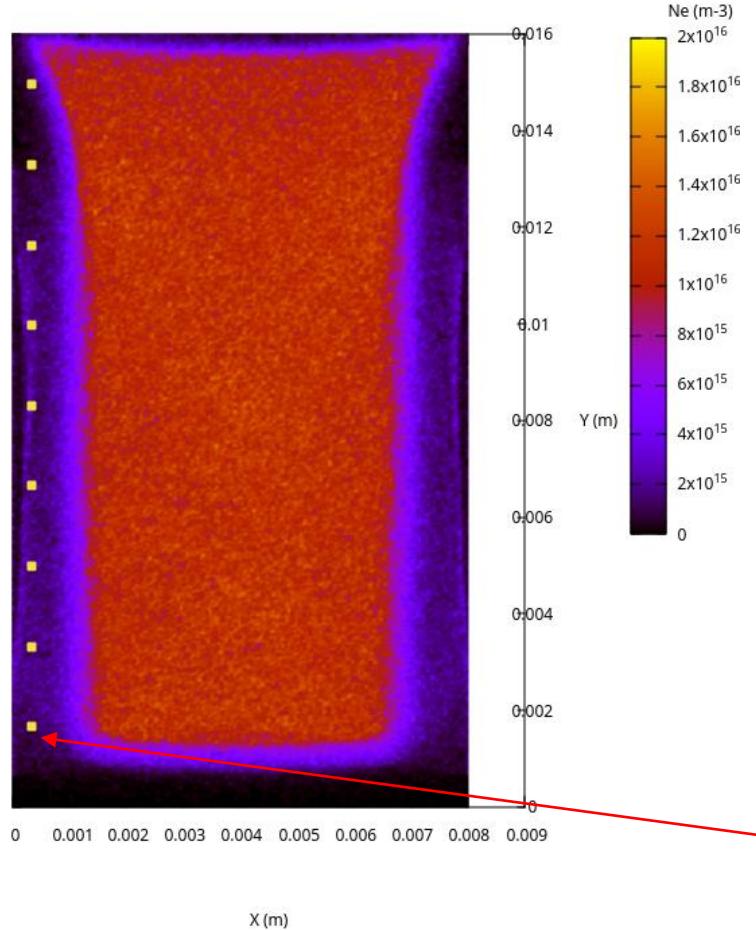
Here the 1d VDFs of ion species 1 from box 25 are plotted as functions of the energy of ion motion along the X, Y, and Z directions, respectively. The negative energy value correspond to the motion in the negative direction.

The code saves evolution in time of the following:

- Values in certain locations (probes):
 - Electrostatic potential
 - Electric field components EX and EY
 - Densities of electrons and all ion species
- Collision frequencies averaged over the whole system
- Numbers of particles collided with walls (boundary objects) and emitted by the walls

```
'_0007_Ne_m3_2D.bin' binary matrix u 1:2:3  
'_probelocs.dat' u ($4*0.001):($5*0.001):(1)
```

Locations of all probes are provided in file `_probelocs.dat`.



There are also files `_probelocs_cluster_RRRR_CCCC.dat` where RRRR is the global MPI rank of the cluster master, and CCCC is the MPI rank of the cluster master in COMM_CLUSTER (think of it as of the number of the cluster beginning from zero) – those files have locations of probes inside each cluster (may be handy sometimes). All `_probelocs*` files have headers describing what's inside.

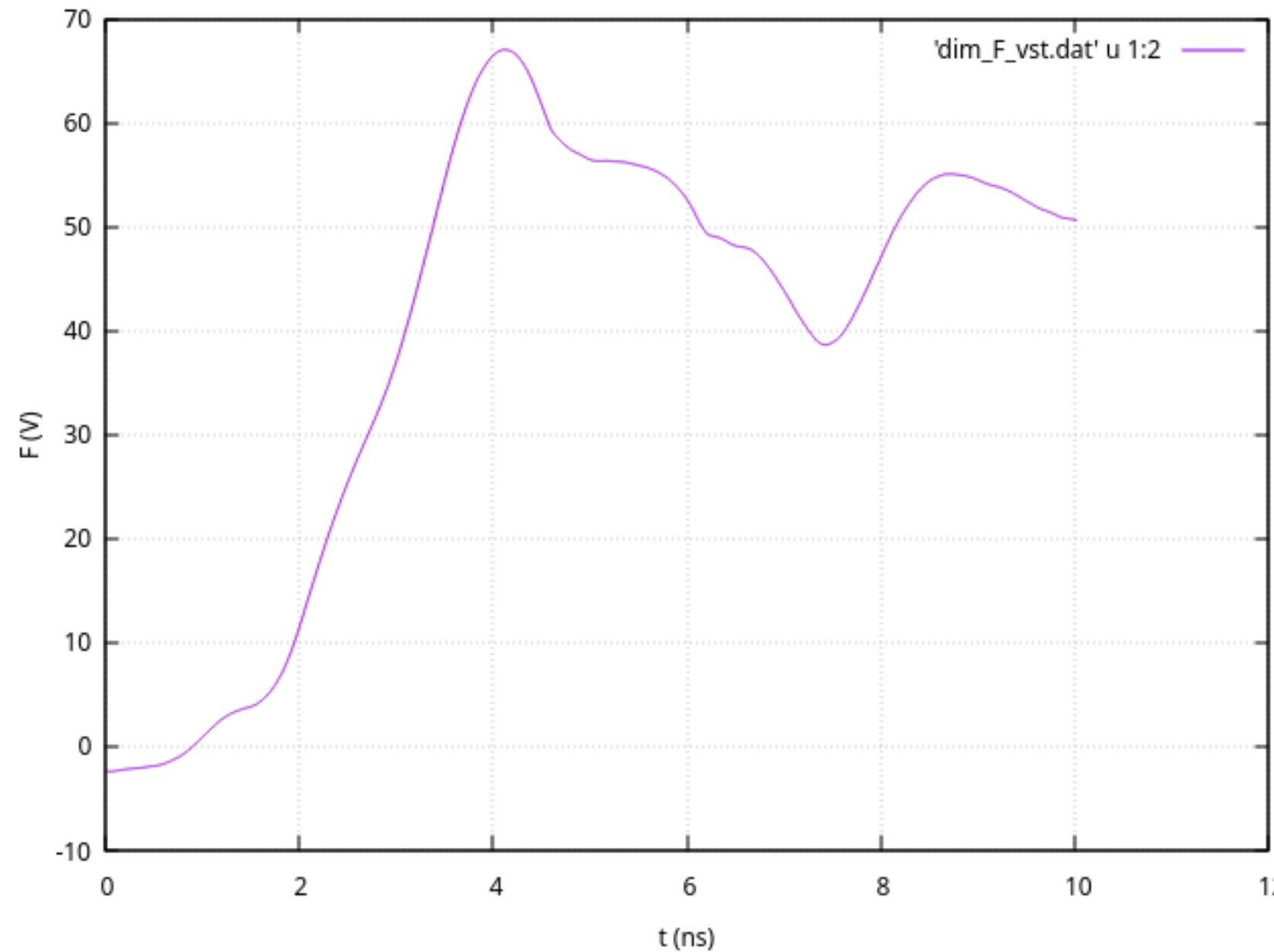
When looking at the probe data, it may be useful to have a picture where probe locations are plotted inside the simulation domain. An example of such a picture is on the left. In the following slides we show data from probe 1.

Location of probe 1

```
gnuplot> set zran [*:*]  
gnuplot> set xlabel 'X (m)'; set ylabel 'Y (m)'; set cblabel 'Ne (m-3)' offset -5,12 rotate by 0  
gnuplot> set palette rgbformulae 7,5,15  
gnuplot> splot '_0007_Ne_m3_2D.bin' binary matrix u 1:2:3 w pm3d, '_probelocs.dat' u ($4*0.001):($5*0.001):(1) lt 5
```

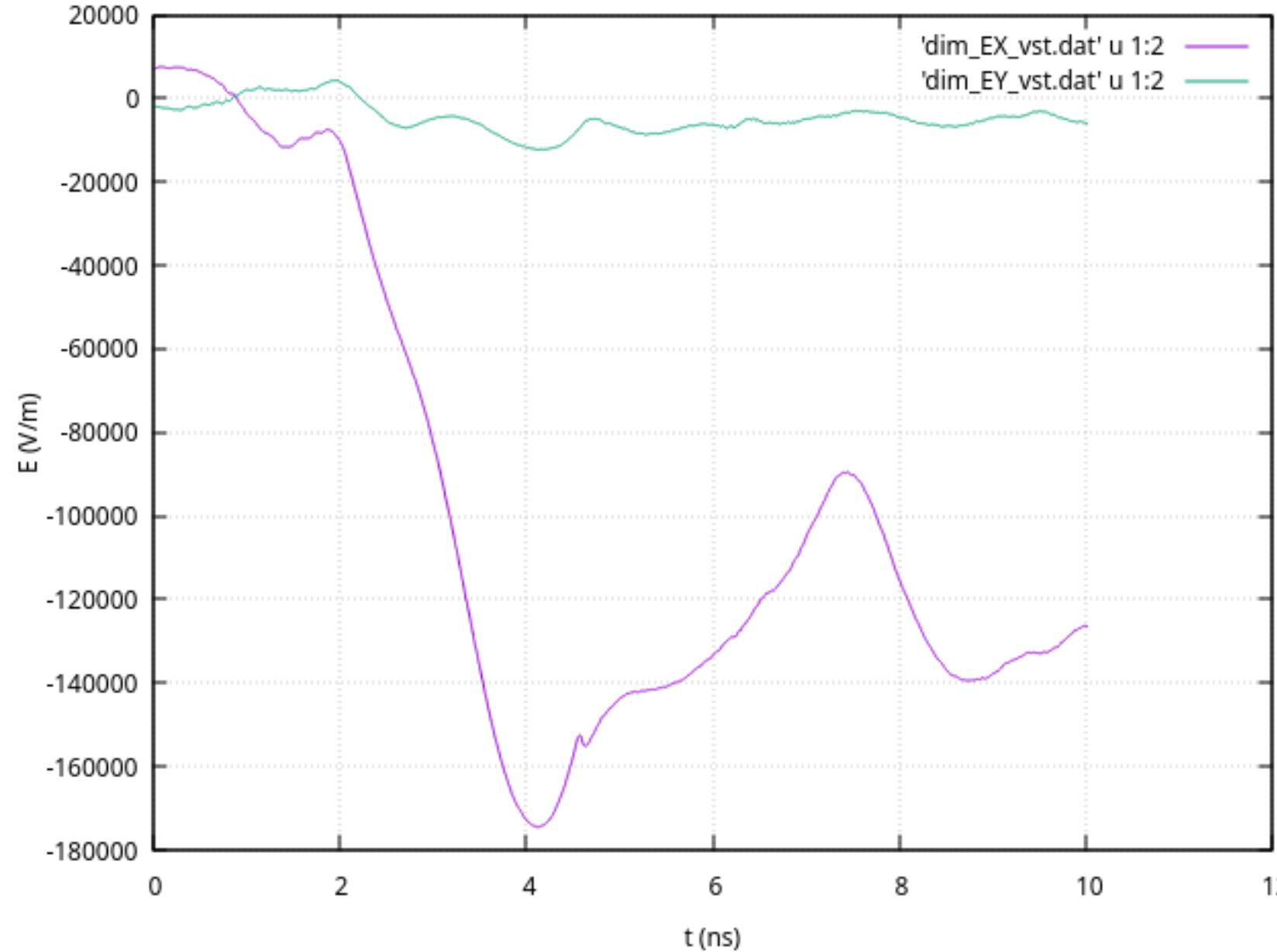
dim_F_vst.dat :: electrostatic potential in units of V

```
gnuplot> set xlabel 't (ns)'; set ylabel 'F (V)'  
gnuplot> plot 'dim_F_vst.dat' u 1:2 w l
```



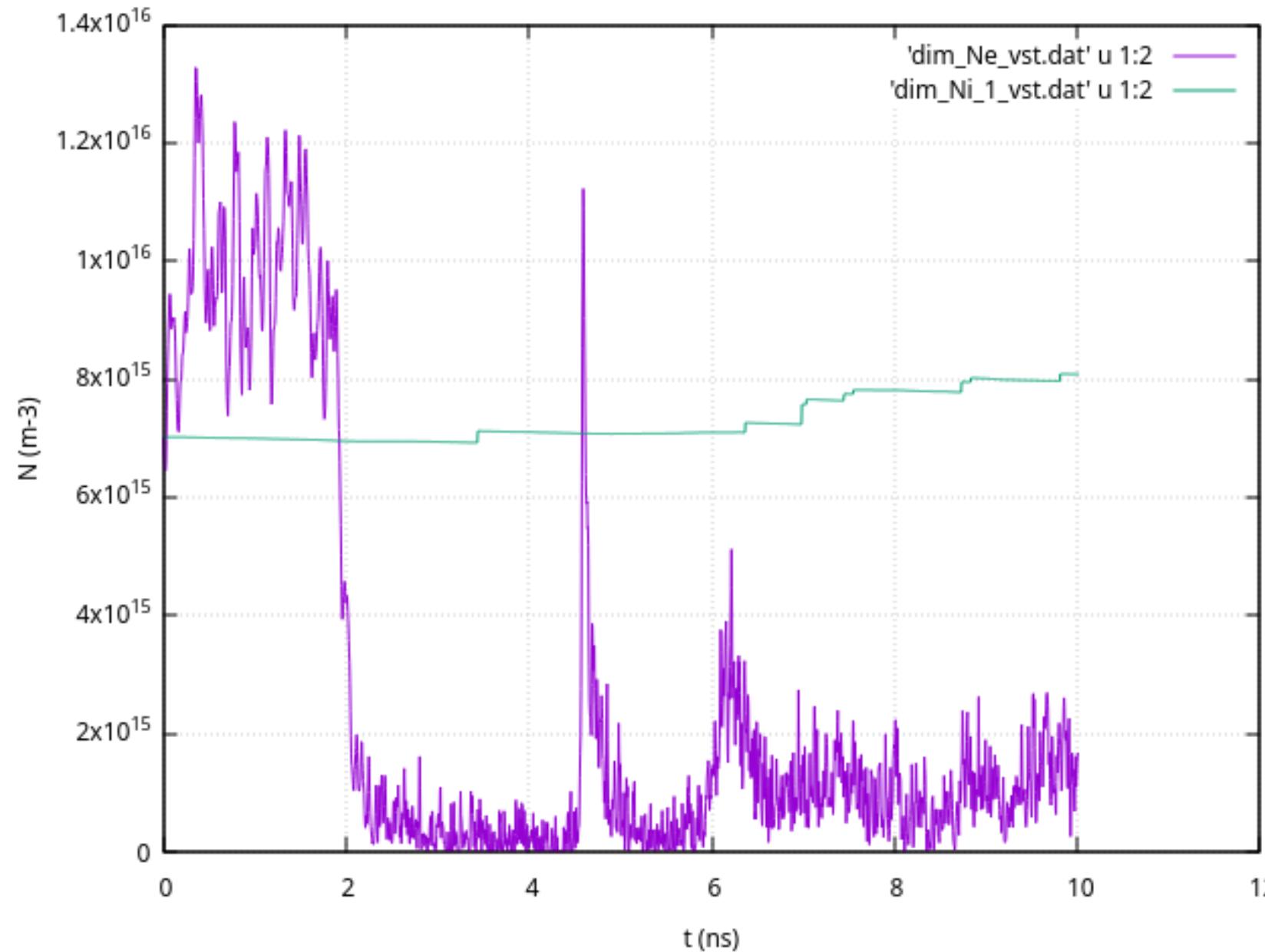
dim_EX_vst.dat and dim_EY_vst.dat :: electric field components EX and EY in units of V/m

```
gnuplot> set xlabel 't (ns)'; set ylabel 'E (V/m)'  
gnuplot> plot 'dim_EX_vst.dat' u 1:2 w l, 'dim_EY_vst.dat' u 1:2 w l
```



dim_Ne_vst.dat and dim_Ni_1_vst.dat :: number densities of electrons and ion species 1 in units of m^{-3}

```
gnuplot> set xlabel 't (ns)'; set ylabel 'N (m-3)'  
gnuplot> plot 'dim_Ne_vst.dat' u 1:2 w l, 'dim_Ni_1_vst.dat' u 1:2 w l
```



Time evolution of collision frequencies

The code saves numbers of electron-neutral collision events of each kind for each neutral species in files `history_coll_e_n_AAAAAA.dat`, where AAAAAA is the name of the neutral (introduced in `init_neutrals.dat`). The header of such a file for Argon (AAAAAA=Argon0) is given below. Note that the header provides values of the electron and ion time steps, and the data saved include the total number of electron macroparticles – this is sufficient to convert the number of collision events into a frequency of collisions, as shown in the next slide.

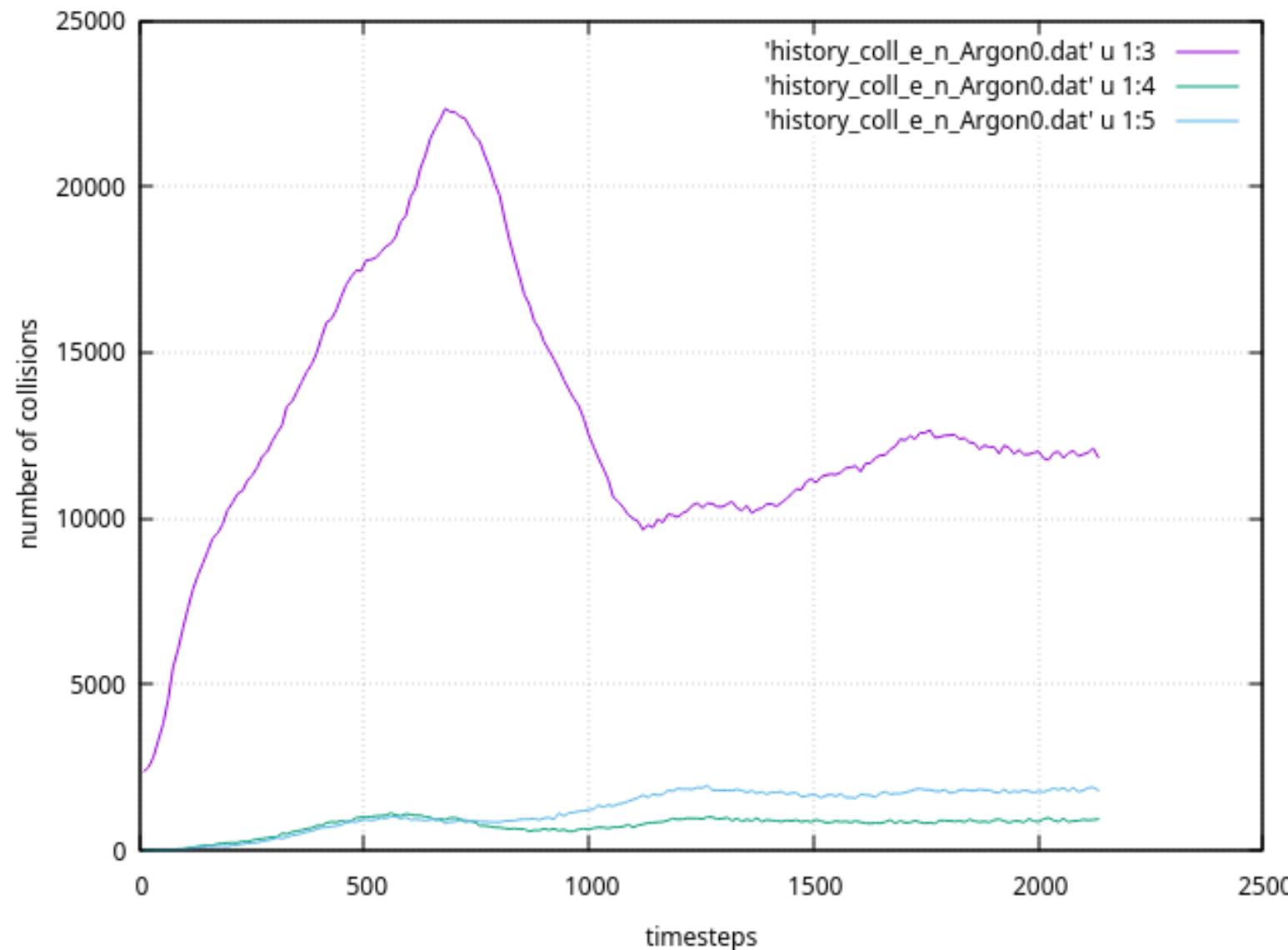
```
header of history_coll_e_n_Argon0.dat ::

# electron time step is 0.4671189E-11 s
#     ion time step is 0.5138308E-10 s
# column 1 is the electron step counter
# column 2 is the total number of electron macroparticles in the whole system
# column 3 is the number of collision events during past ion time step for collision process with id 1 type 10
# column 4 is the number of collision events during past ion time step for collision process with id 2 type 20
# column 5 is the number of collision events during past ion time step for collision process with id 3 type 30
```

In the header above, columns 3,4,5 are for elastic, inelastic, and ionization collisions, respectively.

history_coll_e_n_Argon0.dat :: number of electron neutral collision events at each ion time step (part 1)

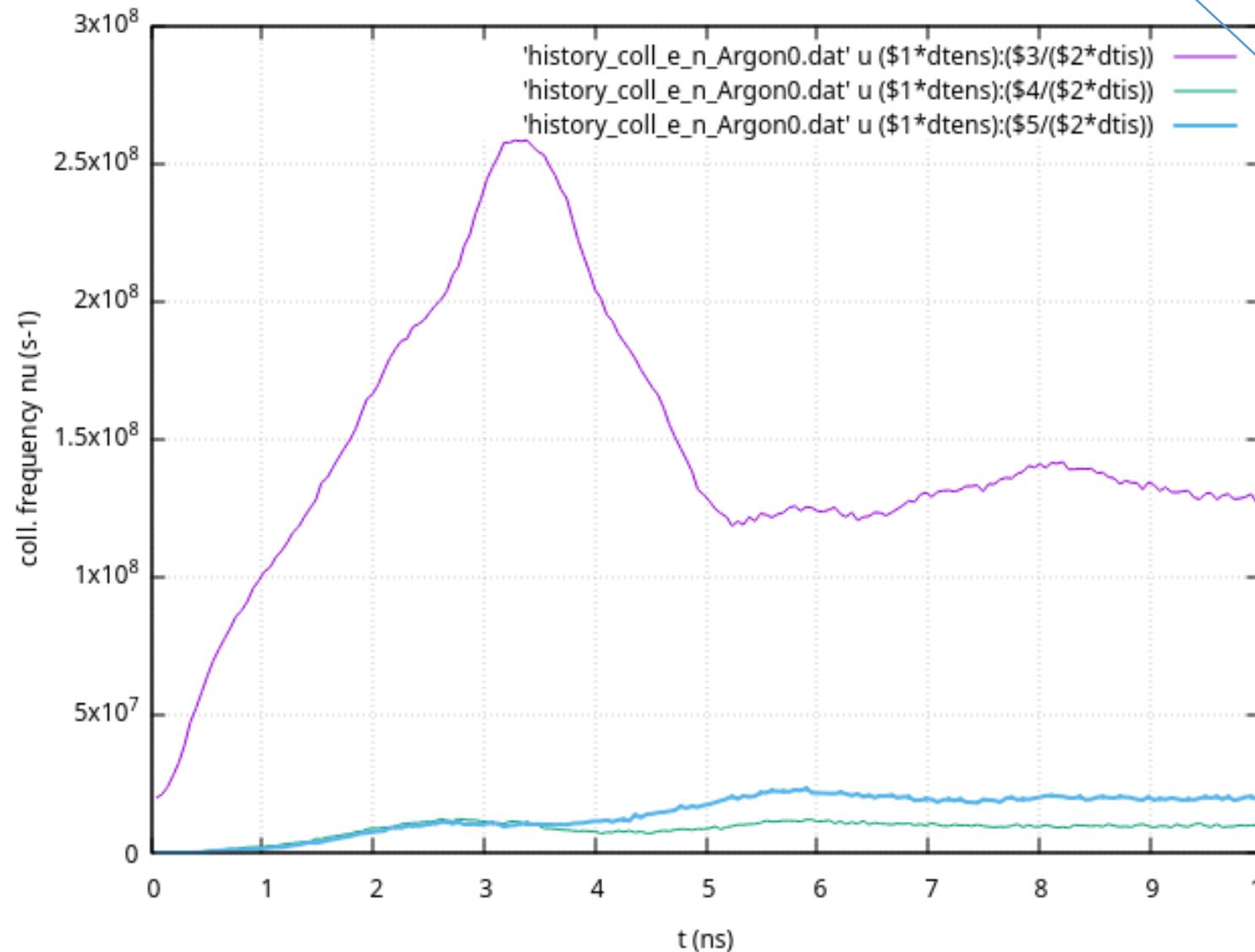
```
gnuplot> set xlabel 'timesteps' ; set ylabel 'number of collisions'  
gnuplot> plot 'history_coll_e_n_Argon0.dat' u 1:3 w l, 'history_coll_e_n_Argon0.dat' u 1:4 w l, 'history_coll_e_n_Argon0.dat' u 1:5 w l
```



Here the file data (numbers of collisions) are plotted directly, without converting them into collision frequencies.

history_coll_e_n_Argon0.dat :: number of electron neutral collision events at each ion time step (part 2)

```
gnuplot> dtens=1e9*0.4671189E-11  
gnuplot> dtis=0.5138308E-10  
gnuplot> set xlabel 't (ns)' ; set ylabel 'coll. frequency nu (s-1)'  
gnuplot> plot 'history_coll_e_n_Argon0.dat' u ($1*dtens):($3/($2*dtis)) w l, 'history_coll_e_n_Argon0.dat' u ($1*dtens):($4/($2*dtis)) w l, 'history_coll_e_n_Argon0.dat' u ($1*dtens):($5/($2*dtis)) w l lw 2
```



Convert number of collisions into frequency [ns], note the use of the number of macroparticles (data column 2)

Convert timestep into time [ns]

Here the data are converted into the collision frequencies.
In the gnuplot commands above,
dtens is the electron time step [ns],
dtis is the ion time step [s].

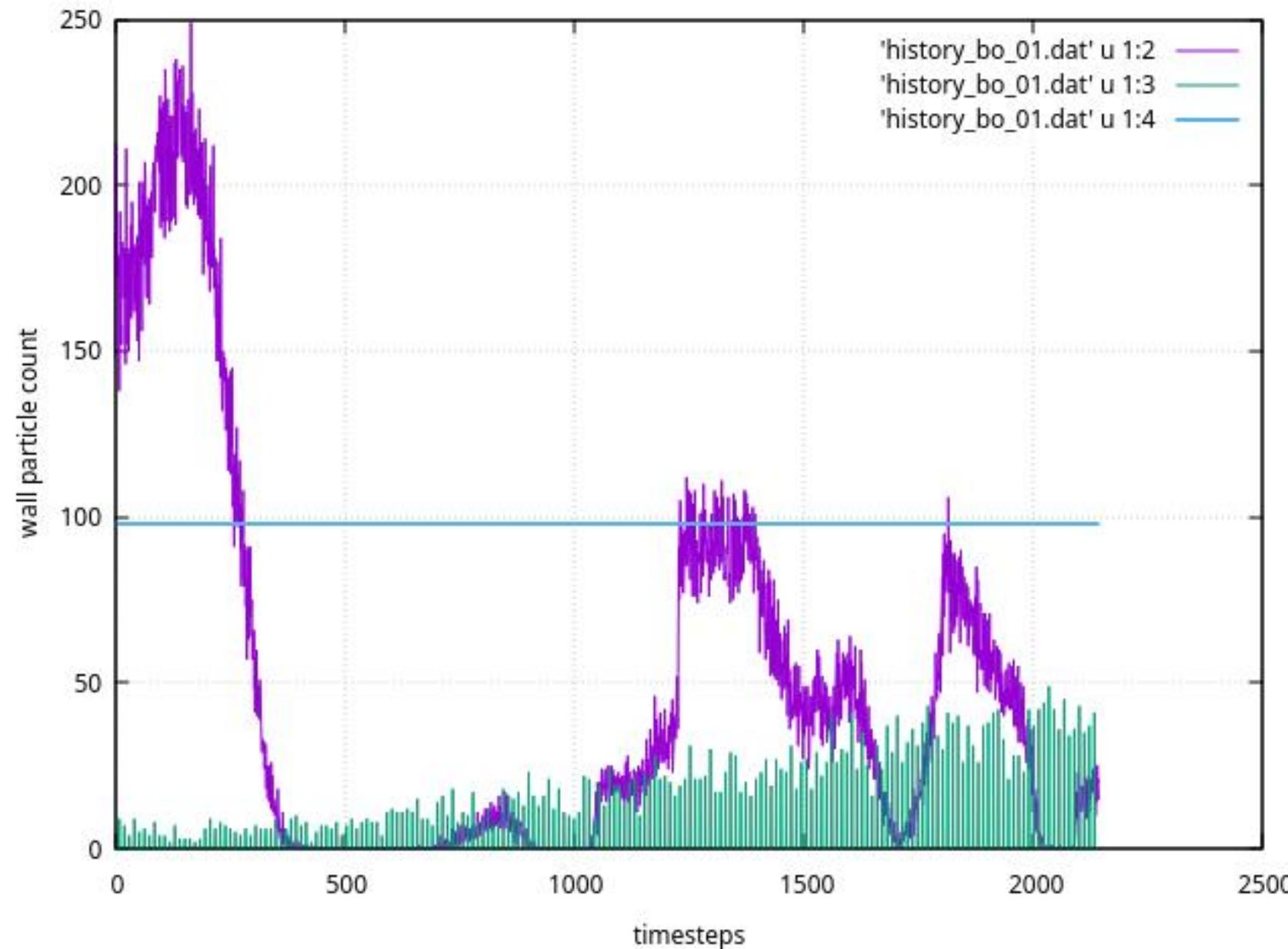
Time evolution of wall fluxes

- The code saves numbers of macroparticles (electron and all ion species) which collide with each boundary object, as well as the number of electron macroparticles emitted by each boundary object (if there are any) at each time step.
- The data are saved into files with names `history_bo_NN.dat`, where NN is the number of the boundary object.
- The data in these files are arranged as follows:
 - column 1 is the time step,
 - column 2 is the number of electron macroparticles collided with the boundary object,
 - columns 3 to 3+N_spec-1 are the numbers of macroparticles of ions species 1:N_spec collided with the boundary object,
 - column 3+N_spec is the number of electron macroparticles emitted by the boundary object.

In the examples below we consider only boundary object 1, which for this specific simulation is a metal electrode on the left edge of the simulation domain, capable of emitting a constant electron flux.

history_bo_01.dat :: number of electron and ion particles collided with / emitted by boundary object 1 (part 1)

```
gnuplot> set xlabel 'timesteps'; set ylabel 'wall particle count'  
gnuplot> plot 'history_bo_01.dat' u 1:2 w l, 'history_bo_01.dat' u 1:3 w l, 'history_bo_01.dat' u 1:4 w l lw 2
```



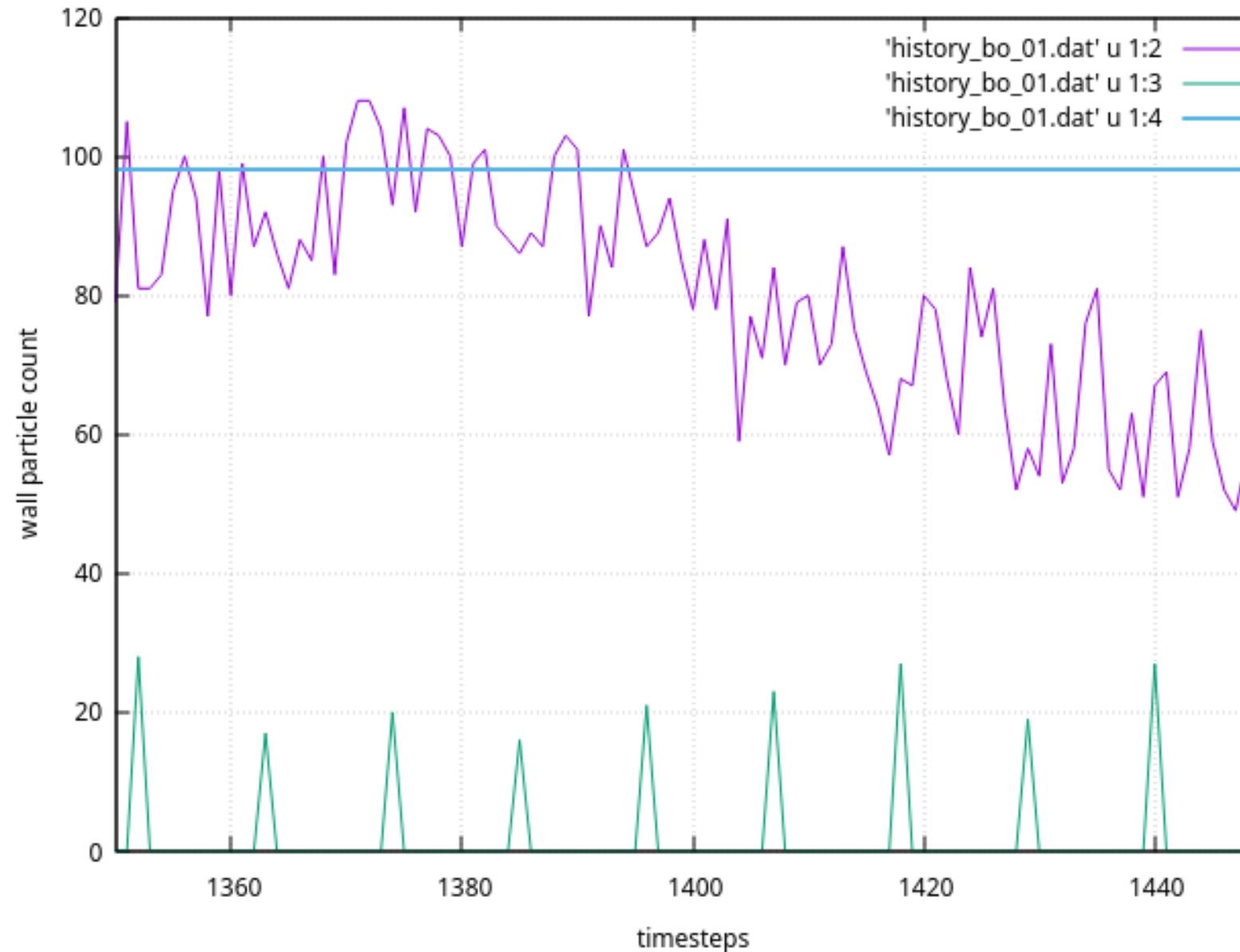
Here the file data (numbers particles collided with and emitted by the wall) are plotted directly, without converting them into electric current, for example.

Note that columns 2,3,4 are for:

- electrons collided with the wall,
- Ions collided with the wall,
- and electrons emitted by the wall, respectively.

history_bo_01.dat :: number of electron and ion particles collided with / emitted by boundary object 1 (part 2)

```
gnuplot> set xran [1350:1450];
gnuplot> set xlabel 'timesteps'; set ylabel 'wall particle count'
gnuplot> plot 'history_bo_01.dat' u 1:2 w l, 'history_bo_01.dat' u 1:3 w l, 'history_bo_01.dat' u 1:4 w l lw 2
```



This figure shows the same values as the one in the previous slide. Note that the ions are advanced every ion timestep which is 11 times the electron one. This is why for ions the graph is a set of spikes, with zeros in between, see the green curve.

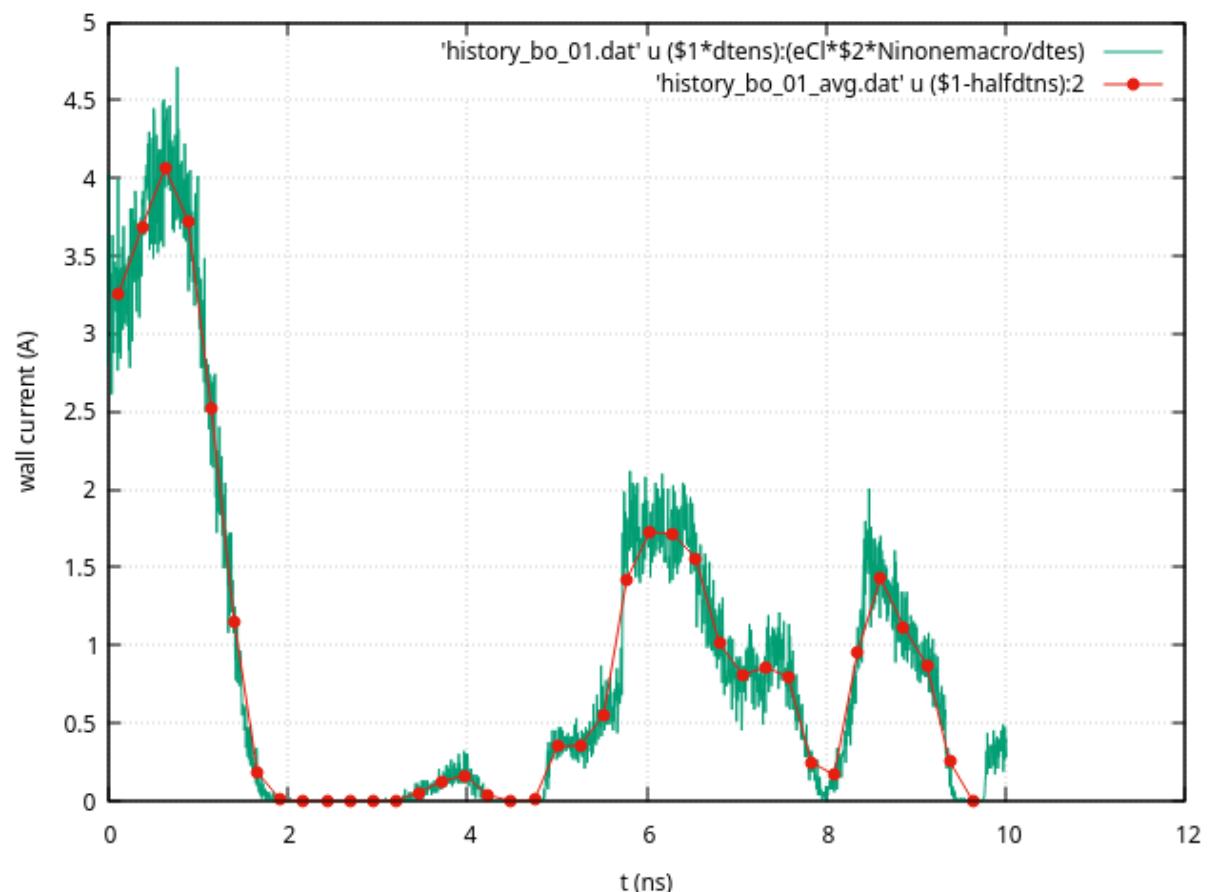
- The original particle-wall collision / emission data files may be converted into a more useful format by data processing program `dataproc_extract_bo_particle_fluxes_from_history.f90`.
- This program should run in the directory with the simulation input and output. It requires the following original input files of the simulation:
 - `init_configuration.dat`
 - `init_simcontrol.dat`
 - `init_particles.dat` .
- For each available file `history_bo_NN.dat` this data processing program produces file `history_bo_NN_avg.dat` where:
 - Column 1 is the time [ns]
 - Column 2 is the flux of electrons colliding with the wall in units of electric current [A]
 - Columns 3 to 3+N_spec-1 are the fluxes of ions of species 1 to N_spec colliding with the wall in units of electric current [A]
 - Column 3+N_spec is the flux of electrons emitted by the wall in units of electric current [A]
 - All fluxes are averaged over certain number of ion timesteps specified by the user.
 - All fluxes are non-negative, if the fluxes are treated as electric currents, their sign is assigned by the user.
 - In the top of the file the length of the corresponding boundary object [m] is given, so that one can calculate the current density at the boundary object.

Example of output of the program converting numbers of particles collided/emitted by walls into electric currents averaged over time

```
./getavgwallfluxes.out
init_configuration.dat is found. Reading the data file...
  Process
N_subcycles =  11
  Process
N_spec =      1
  average over how many ion records?
5
File history_bo_01.dat found, reading the data file ...
### The total number of records in the data file is      2143 (first record has number 1)
Reading from file history_bo_01.dat done
Created file history_bo_01_avg.dat with time and currents in dimensional units...
.....
.....
File history_bo_08.dat found, reading the data file ...
### The total number of records in the data file is      2143 (first record has number 1)
Reading from file history_bo_08.dat done
Created file history_bo_08_avg.dat with time and currents in dimensional units...
```

history_bo_01_avg.dat :: fluxes of electron and ion particles collided with / emitted by boundary object 1 in units of A (part 1)

```
gnuplot> eCl=1.6e-19
gnuplot> Nscalem3=1e17
gnuplot> dxm=0.3324547e-4
gnuplot> Nppc=200
gnuplot> Ninonemacro=Nscalem3*dxm*dxm/Nppc
gnuplot> halfdtns=0.252246/2
gnuplot> set xlabel 't (ns)'; set ylabel 'wall current (A)'
gnuplot> dtes=0.4671189E-11
gnuplot> dtens=1e9*dtes
gnuplot> plot 'history_bo_01.dat' u ($1*dtens):(eCl*$2*Ninonemacro/dtes) w l lt 2,
'history_bo_01_avg.dat' u ($1-halfdtns):2 w linesp lt 7
```

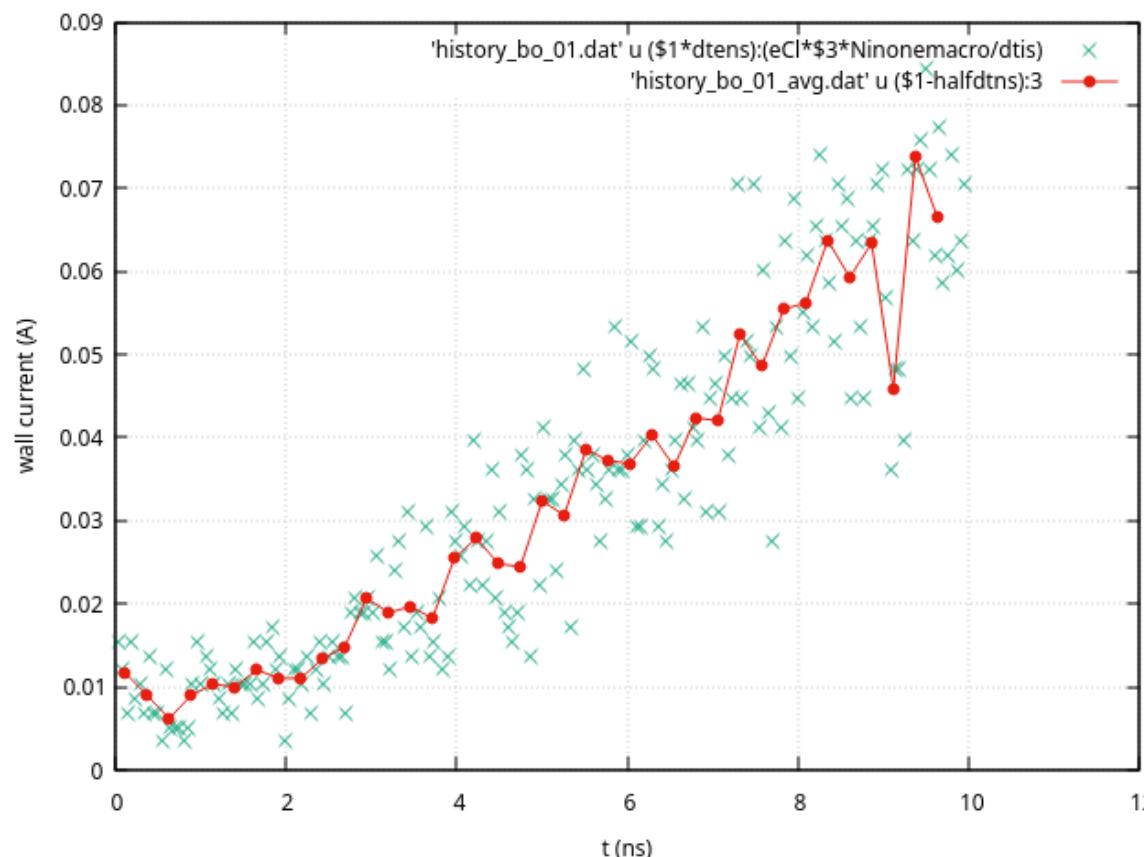


- eCl is the elementary charge [C]
- $Nscalem3$ is the scale density [m^{-3}] (see file `init_configuration.dat`)
- dxm is the grid cell size [m] (printed out during simulation or can be derived from probe locations in `_probelocs.dat`)
- $Nppc$ is the number of particles per cell for the scale density (see file `init_configuration.dat`)
- $Ninonemacro$ is the number of electrons in one macroparticle
- $halfdtns$ is half of the time step between averaged wall fluxes [ns] (see `history_bo_01.avg.dat`)
- $dtes$ is the electron time step [s] (printed during simulation or can be found in `history_coll_e_n_Argon0.dat`)
- $dtens$ is the electron time step [ns]

Here the average (over 5 ion time steps) flux of electrons colliding with the wall (red curve with dots) is plotted together with the original data of electron-wall collisions converted into the electric current (the green curve).

history_bo_01_avg.dat :: fluxes of electron and ion particles collided with / emitted by boundary object 1 in units of A (part 2)

```
gnuplot> set yran [1e-10:*];
gnuplot> eCl=1.602189e-19
gnuplot> Nscalem3=1e17
gnuplot> dxm=0.3324547e-4
gnuplot> Nppc=200
gnuplot> Ninonemacro=Nscalem3*dx*dx/Nppc
gnuplot> halfdtns=0.252246/2
gnuplot> set xlabel 't (ns)'; set ylabel 'wall current (A)'
gnuplot> dtens=1e9*0.4671189E-11
gnuplot> dtis=0.5138308E-10
gnuplot> plot 'history_bo_01.dat' u ($1*dtens):(eCl*$3*Ninonemacro/dtis) lt 2,
'history_bo_01_avg.dat' u ($1-halfdtns):3 w linesp lt 7
```

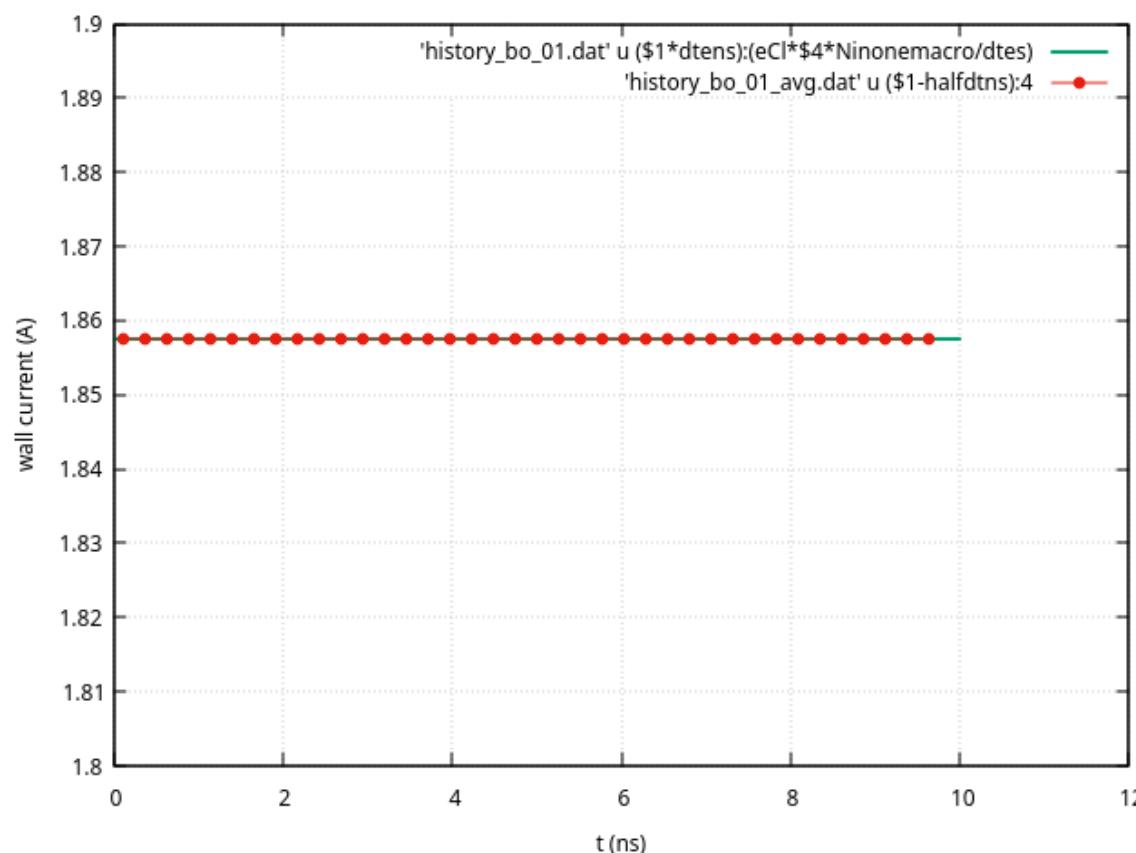


Here dtis is the ion time step which is N_subcycles=11 times bigger than the electron time step (see init_simcontrol.dat). It can be found in history_coll_e_n_Argon0.dat)

Here the average (over 5 ion time steps) flux of ions colliding with the wall (red curve with dots) is plotted together with the original data of ion-wall collisions converted into the electric current (the green diagonal crosses).

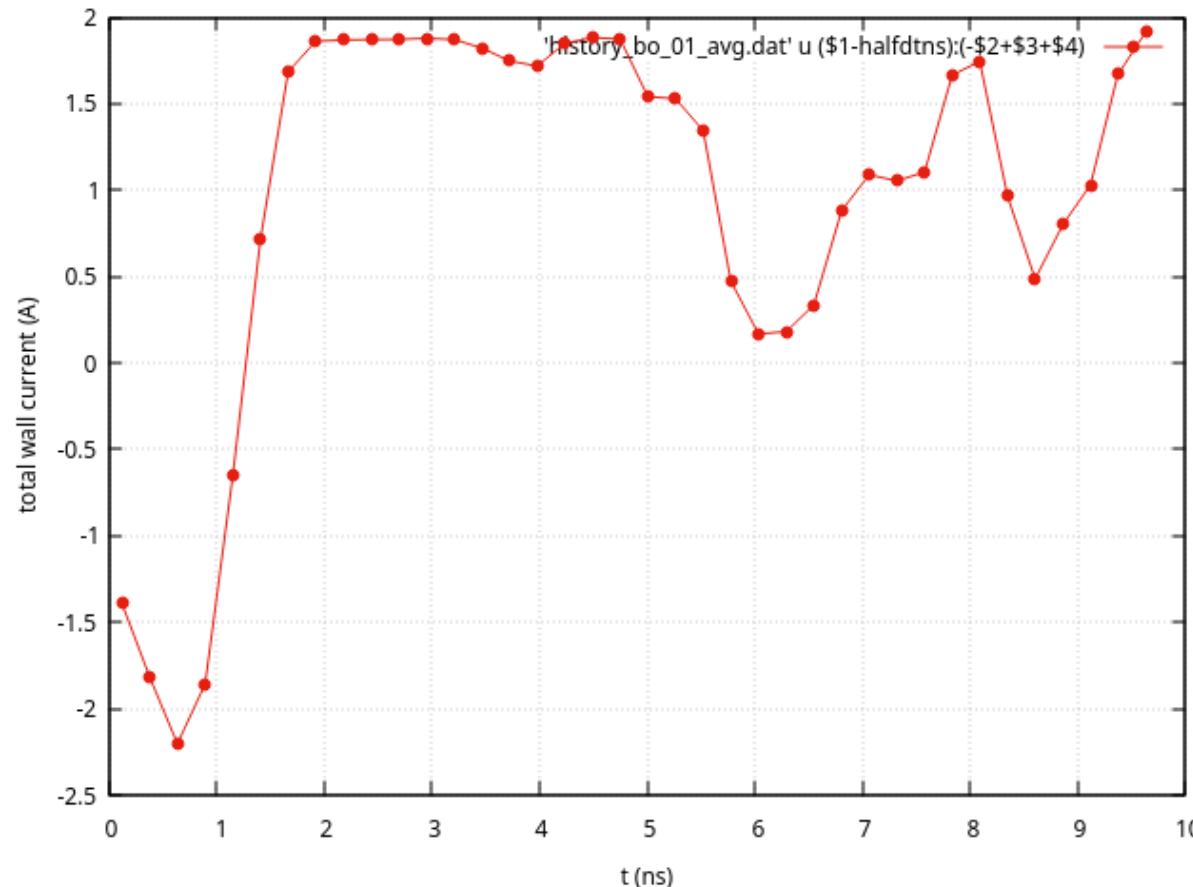
history_bo_01_avg.dat :: fluxes of electron and ion particles collided with / emitted by boundary object 1 in units of A (part 3)

```
gnuplot> set yran [1.8:1.9];
gnuplot> eCl=1.602189e-19
gnuplot> Nscalem3=1e17
gnuplot> dxm=0.3324547e-4
gnuplot> Nppc=200
gnuplot> Ninonemacro=Nscalem3*dx*dx/Nppc
gnuplot> halfdtns=0.252246/2
gnuplot> set xlabel 't (ns)'; set ylabel 'wall current (A)'
gnuplot> dtes=0.4671189E-11
gnuplot> dtens=1e9*dtes
gnuplot> plot 'history_bo_01.dat' u ($1*dtens):(eCl*$4*Ninonemacro/dtes) w 1 lt 2 lw 2,
'history_bo_01_avg.dat' u ($1-halfdtns):4 w linesp lt 7
```



Here the average (over 5 ion time steps) flux of electrons emitted by the wall (red curve with dots) is plotted together with the original data of emitted electron macroparticles converted into the electric current (the green curve).

```
gnuplot> set yran [*:*]
gnuplot> halfdtns=0.252246/2
gnuplot> set xlabel 't (ns)'; set ylabel 'total wall current (A)'
gnuplot> plot 'history_bo_01_avg.dat' u ($1-halfdtns):(-$2+$3+$4) w linesp lt 7
```



The average particle fluxes are not negative and carry no information about the direction of the corresponding electric current.

In this case, if it is assumed that a flux of positive charges into the wall creates a positive current, it is necessary to use the current of electrons colliding with the wall with a “minus” sign, and the current of emitted electrons with a “plus” sign.

The resulting total electric current to boundary object 1 is shown in this figure.