

# Simulation and Visualisation of e-column in IOTA ring using ParaView

Diletta Milana

*under the supervision of*

**Charles Thangaraj (FAST-IOTA)**

*and*

**Chong Shik Park (IOTA), Giulio Stancari (IOTA)**



# Paraview

An open-source, multi-platform and **extremely flexible**  
data analysis tool for extremely **large datasets**

Great, but **why** do we need this?

# The pipeline (1)

**simulation**



```
electrons_004000.txt
0 -0.001803827728 -0.002565578783 0.187499165300 0
1 -0.007118024162 -0.000416739833 0.155742518928 0
2 -0.001625481857 -0.001040461636 0.160654636022 1
3 -0.002626024022 -0.006366324857 0.142098638337 0
4 -0.006809051652 -0.004957525872 0.100555373559 0
5 -0.005476233917 -0.004252995283 0.092449075150 0
6 -0.004278192725 -0.006563516158 0.091188208178 0
7 -0.003549761553 -0.004711722512 0.055201566955 0
8 -0.006256324653 -0.000139722876 0.111396055439 0
9 -0.008024367168 -0.001947361665 0.156278185925 0
10 -0.001793474195 -0.004937261567 0.100803539897 1
11 -0.000972019423 -0.001493496252 0.000000000000 1
12 -0.000177878550 -0.000134484287 0.000000000000 1
13 -0.002268511951 -0.000790097831 0.000000000000 1
14 -0.004911240976 -0.000903702145 0.000000000000 1
15 -0.000436599051 -0.000380647677 0.100000000000 1
16 -0.000735911682 -0.000313844502 0.000000000000 1
17 -0.003104668961 -0.003484500805 0.024806331381 1
18 -0.001313524477 -0.00066515821 0.09805136958 1
19 -0.002431350434 -0.000718416488 0.014781482764 1
20 -0.001214937130 -0.000064631726 0.146384599802 1
21 -0.001176347013 -0.000380361078 0.095392049445 1
22 -0.000709065754 -0.002023749499 0.102808438057 1
23 -0.000444428162 -0.001799446002 0.185497167735 1
24 -0.000550194375 -0.003034492196 0.060961899238 1
25 -0.001074244115 -0.002070412586 0.197607482142 1
26 -0.005052651027 -0.001992778524 0.053267072130 1
27 -0.000287491286 -0.000162827658 0.177445488314 1
28 -0.000454042529 -0.001893567299 0.117614356875 1
29 -0.001406467443 -0.000513551747 0.174143220618 1
```

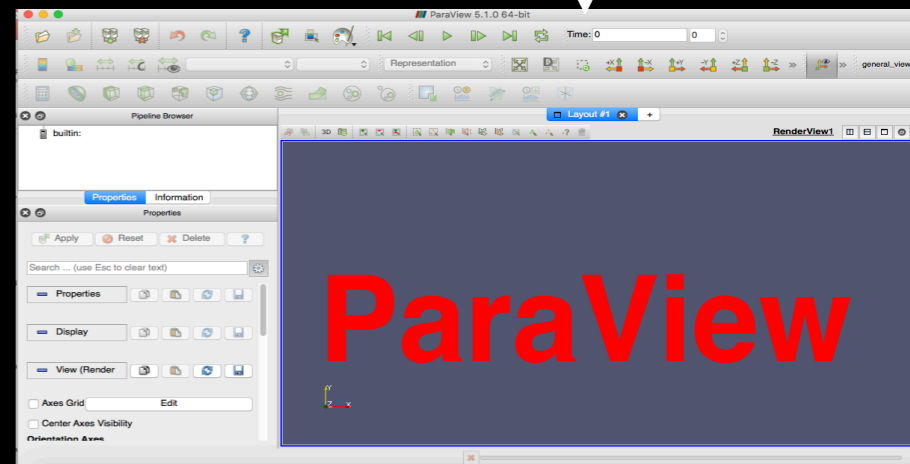
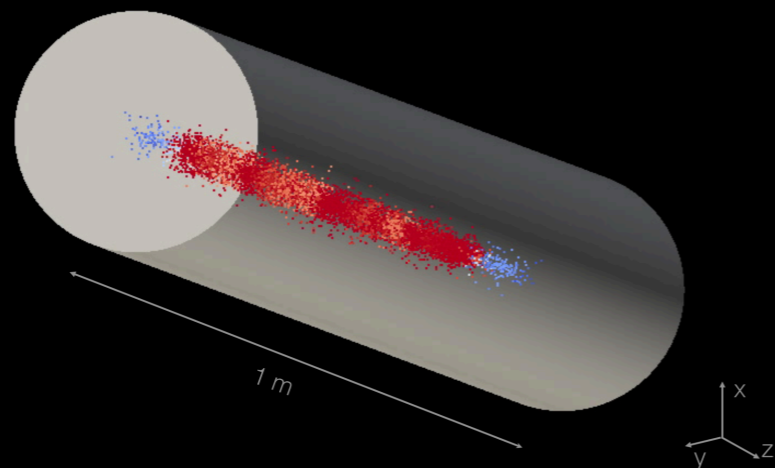
**file**



```
electrons_003500.txt
0,-0.002818353819,-0.001225297198,0.028058384296,1
1,-0.002264426841,-0.001489587945,0.009352382714,1
2,-0.005955534822,0.006715599229,0.006084227214,0
3,-0.001761828671,0.000628601768,0.014034560853,1
4,0.000102867179,0.002258199701,0.033710922673,0
5,-0.000485350741,-0.001638509203,0.166490387816,0
6,-0.001952805705,-0.002930052861,0.120021844337,1
7,-0.001420468301,-0.006986056905,0.164035460119,0
8,-0.004675000000,-0.002504099716,0.191490619737,0
9,-0.000950000000,-0.001269181069,0.182374964505,0
10,-0.005800000000,0.000000000000,0.000000000000,0
11,-0.003100000000,0.000000000000,0.000000000000,0
12,-0.004700000000,0.000000000000,0.000000000000,0
13,-0.003400000000,0.000000000000,0.000000000000,0
14,-0.003257886669,-0.006730292803,0.180644134101,0
15,-0.002618481423,-0.002725282692,0.186040697175,1
16,-0.001392638551,-0.001085078008,0.182528157613,1
17,-0.001400917185,-0.003219514744,0.188639644548,1
18,-0.006101011257,-0.001602898201,0.000000000000,0
19,-0.001148210841,-0.002259098269,0.000000000000,0
20,-0.002524498229,-0.002350970834,0.149049000000,1
21,-0.000130497124,-0.000213257618,0.141086000000,1
22,-0.002170621159,-0.003052001721,0.162033000000,1
23,-0.000825562254,-0.002868100934,0.13769110364,1
24,-0.001161214590,-0.001246031520,0.129849342420,1
25,-0.00127346409,-0.004603555746,0.141605914421,1
26,-0.000045413108,-0.000033121383,0.149987247871,1
27,-0.002036491829,-0.004791427394,0.171568446729,1
28,-0.000744451244,-0.000438467095,0.190049606639,1
29,-0.000332702211,-0.001770358609,0.163821577995,1
```

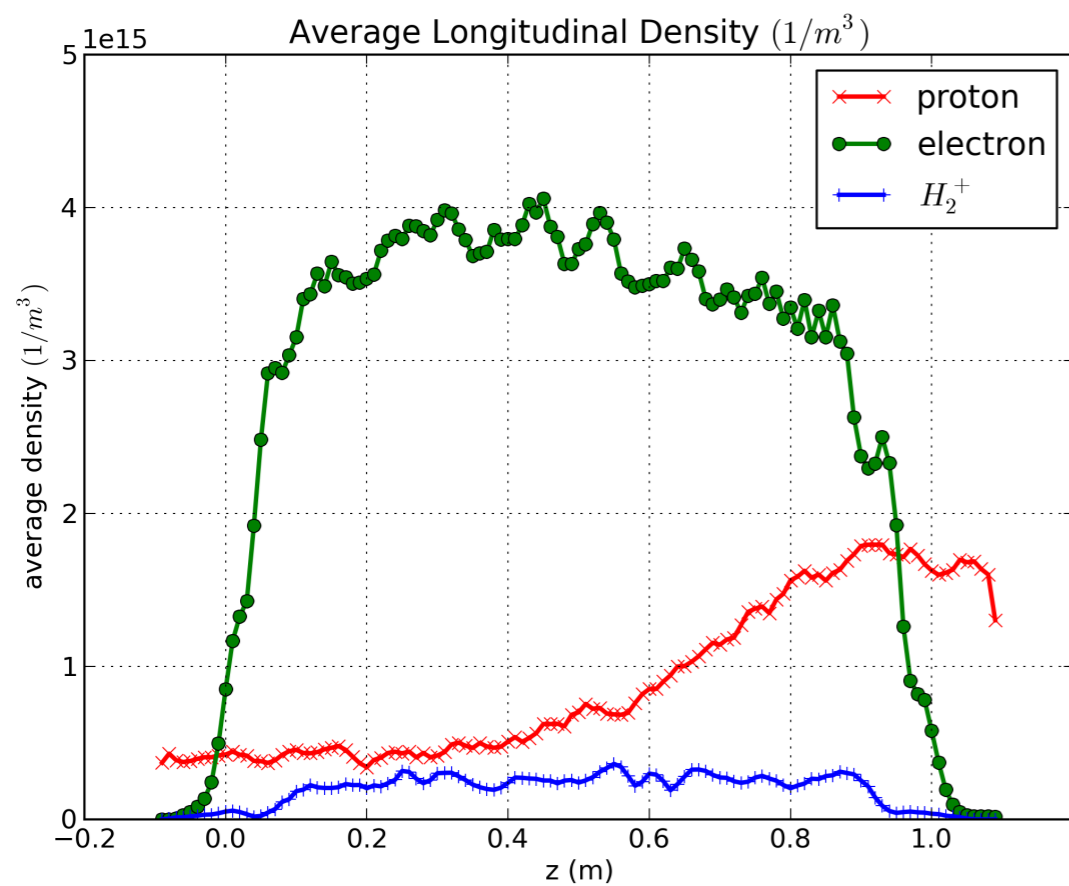
**Formatted file**

**Python script**

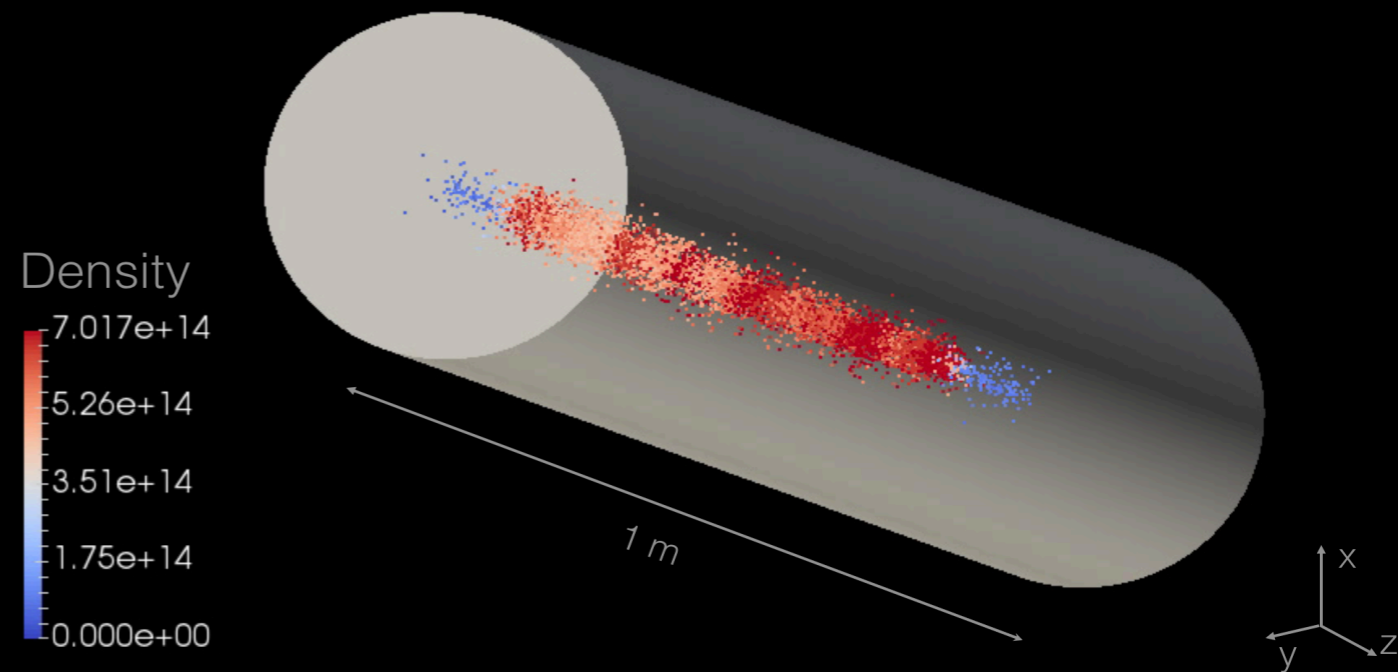


# Example: density profile

Traditional 2D view



ParaView's 3D view



A concrete application:  
the **electron column**  
simulation

# Why electron columns?

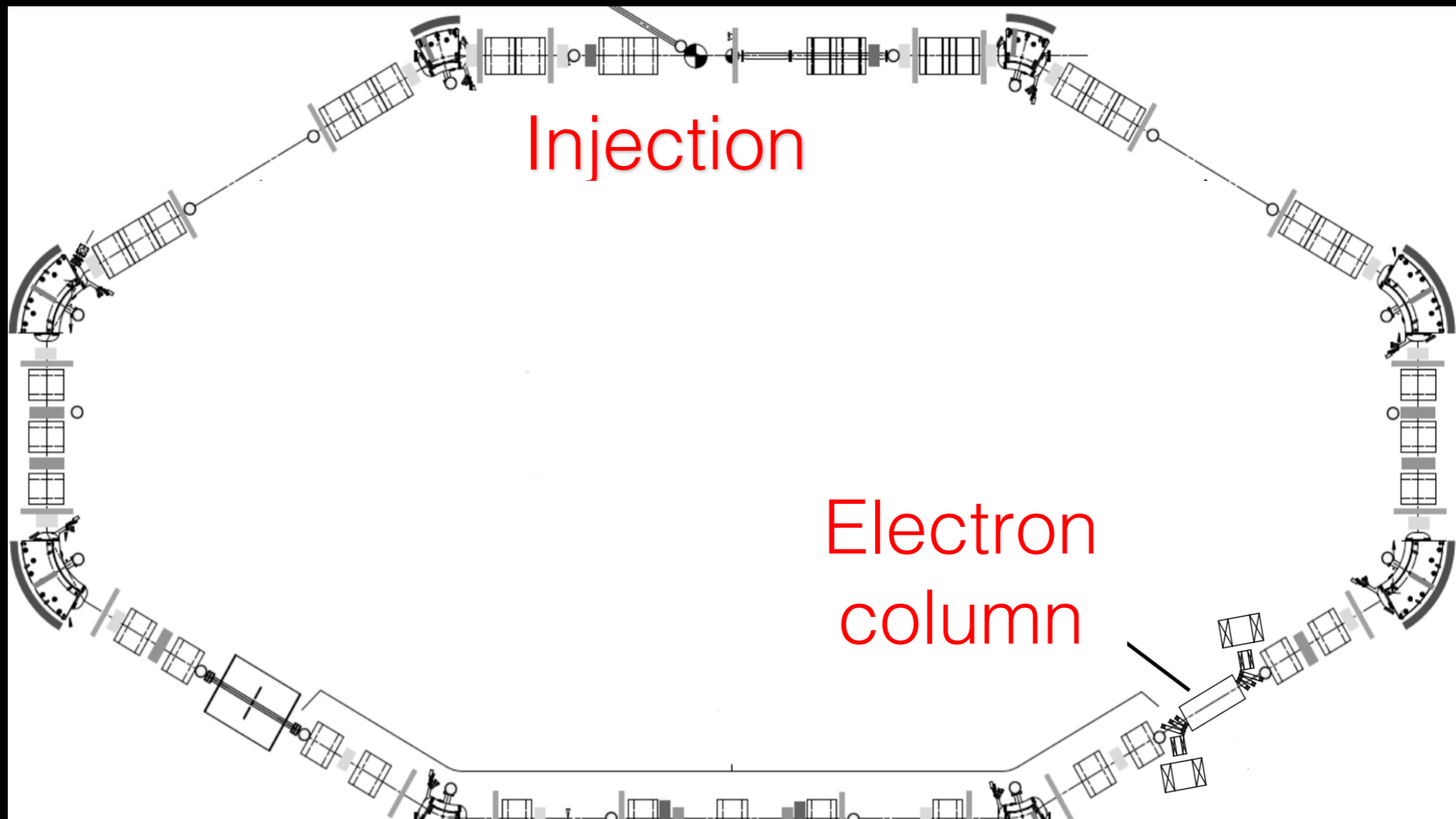
- The performance of high-power accelerators, synchrotrons and storage rings is strongly limited by losses and instabilities, such as the **space charge** effect.
- Space-charge forces in a beam result from mutual **Coulomb repulsion**, creating an internal electric field.
- This could lead to **beam losses** that reduce and limit the current.

# Space Charge Compensation

- To compensate we need **opposite charge**. Two ways to do it: **electron lens** or **electron column**. Here we focus on the **e-column** method.
- The e-column has potential to **improve the performance** of circular accelerators.

# Schematics of IOTA ring

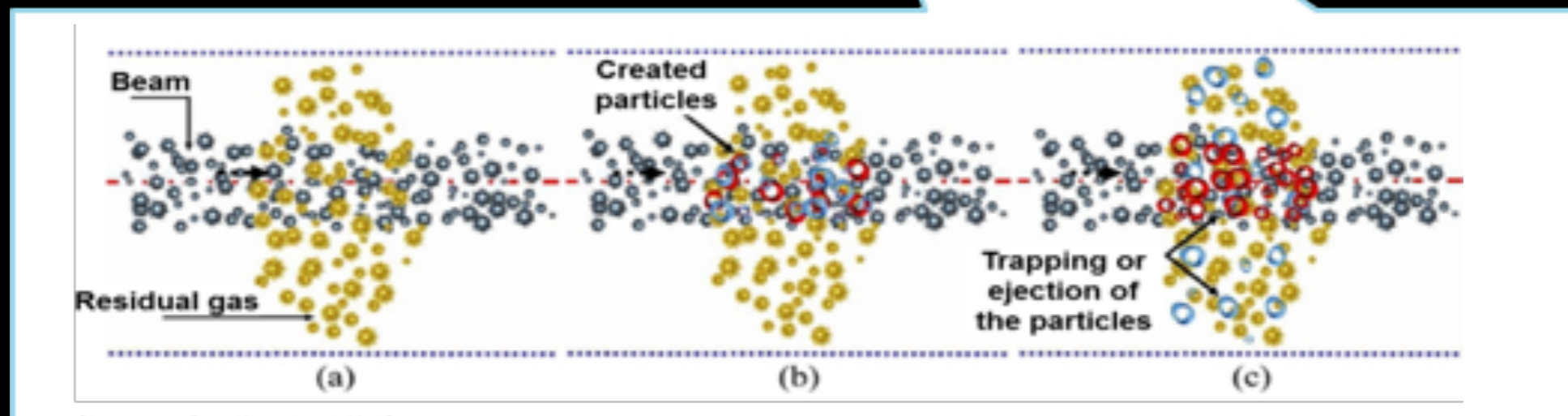
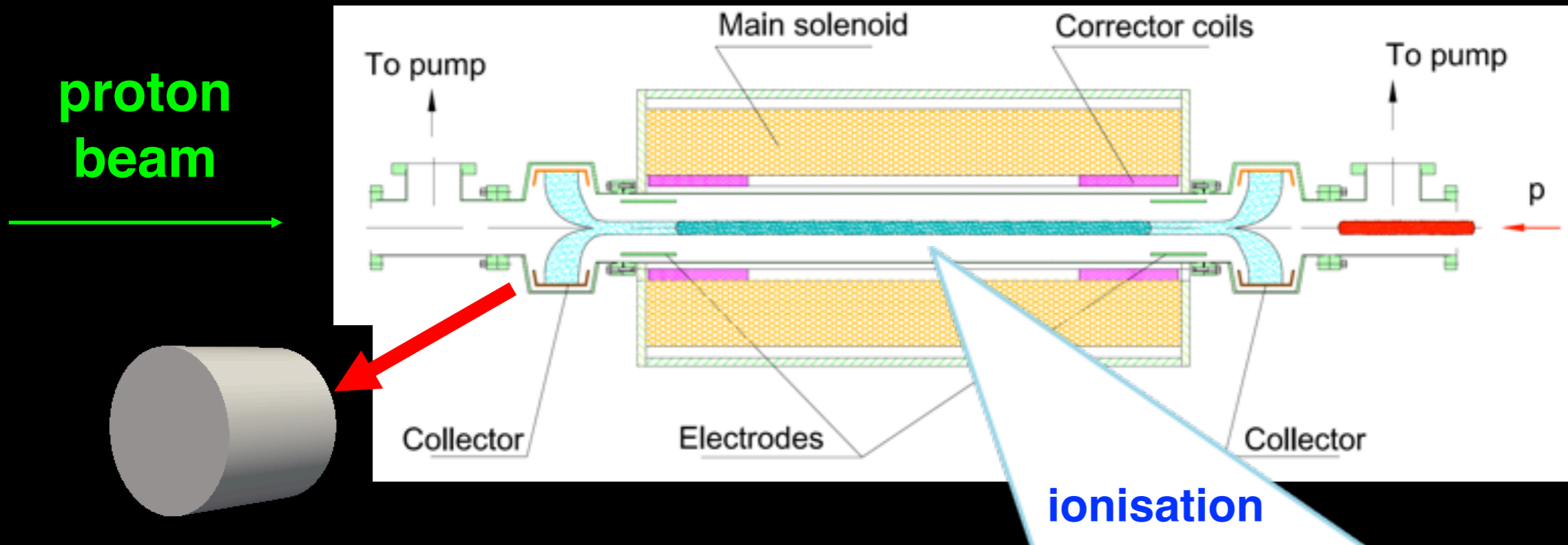
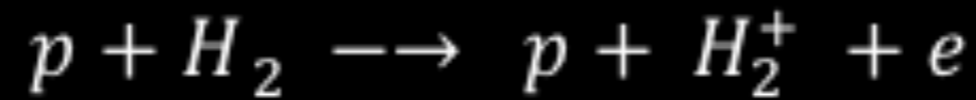
## Integrable Optics Test Accelerator





# The electron column

Vladimir Shiltsev's experimental setup



# The pipeline (2)

**WARP  
simulation**



```
electrons_004000.txt
0 -0.001803827728 -0.002565578783 0.187499165300 0
1 -0.007118024162 -0.000416739833 0.155742518928 0
2 -0.001625481857 -0.001040461636 0.160654636022 1
3 -0.002626024022 -0.006366324857 0.142098638337 0
4 -0.006809051652 -0.004957525872 0.100555373559 0
5 -0.005476233917 -0.004252995283 0.092449075150 0
6 -0.004278192725 -0.006563516158 0.091188208178 0
7 -0.003549761553 -0.004711722512 0.055201566955 0
8 -0.006256324653 -0.000139722876 0.111396055439 0
9 -0.008024367168 -0.001947361665 0.156278185925 0
10 -0.001793474195 -0.004937261567 0.100803539897 1
11 -0.000972019423 -0.00149376252 0.04951001725
12 -0.000177878550 -0.00013712267 0.0661064565
13 -0.002268511951 -0.00079077831 0.08903972868
14 -0.004911240976 -0.0009032145 0.03210796776
15 -0.000436599051 -0.000380767 0.15281256312
16 -0.000735911682 -0.00031345 0.0050197501
17 -0.003104668961 -0.003484500805 0.024806331381 1
18 -0.001313524477 -0.000656515821 0.098055136958 1
19 -0.002431350434 -0.000718416488 0.014781482764 1
20 -0.001214937130 -0.000064631726 0.146384599802 1
21 -0.001176347013 -0.000380361078 0.095392049445 1
22 -0.000709065754 -0.002023749499 0.102808438057 1
23 -0.000444428162 -0.001799446002 0.185497167735 1
24 -0.000550194375 -0.003034492196 0.060961899238 1
25 -0.001074244115 -0.002070412586 0.197607482142 1
26 -0.005052651027 -0.001992778524 0.053267072130 1
27 -0.000287491286 -0.000162827658 0.177445488314 1
28 -0.000454042529 -0.001893567299 0.117614356875 1
29 -0.001406467443 -0.000513551747 0.174143220618 1
```

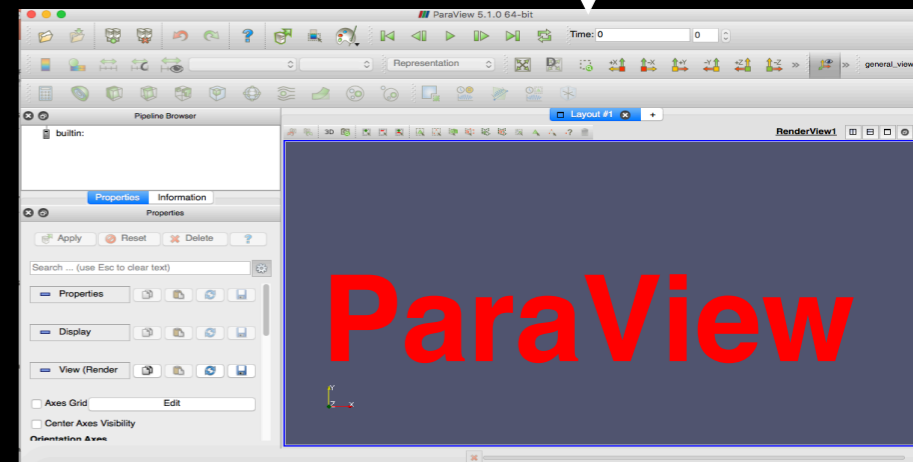
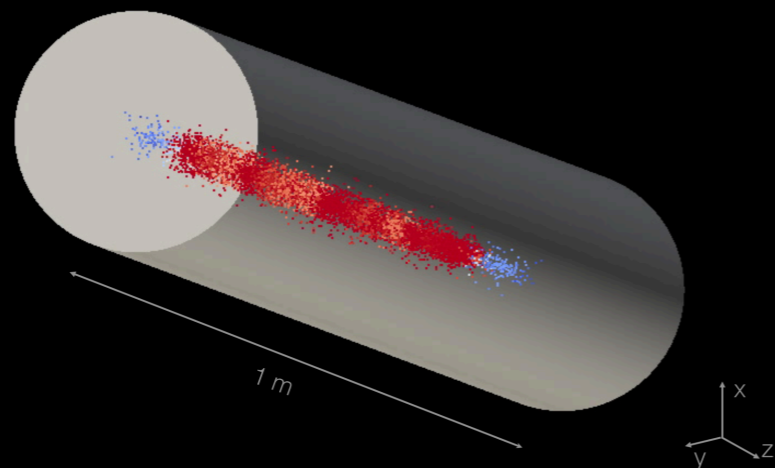
**txt file**



```
electrons_003500.txt
0,-0.002818353819,-0.001225297198,0.028058384296,1
1,-0.002264426841,-0.001489587945,0.009352382714,1
2,-0.005955534822,0.006715599229,0.006084227214,0
3,-0.001761828671,0.000628601768,0.014034560853,1
4,0.000102867179,0.002258199701,0.033710922673,0
5,-0.000485350741,-0.001638509203,0.166490387816,0
6,-0.001952805705,-0.002930052861,0.120021844337,1
7,-0.001420468301,-0.006986056905,0.164035460119,0
8,-0.004675100054,-0.002504099716,0.191490619737,0
9,-0.000957000054,-0.001269181069,0.182374964505,0
10,-0.00589217914,0.00093151036,0.108125738
11,-0.0031100055,0.0051036,0.108125738
12,-0.00471869146,0.00044503,0.111119
13,-0.00341677096,0.000777418,0.117037
14,-0.00325788669,-0.006730292803,0.18064413410,0
15,-0.002618481423,-0.002725282692,0.186040697175,1
16,-0.001392638551,-0.001085078008,0.182528157613,1
17,-0.001400917185,-0.003219514744,0.188639644548,1
18,-0.006101011257,-0.00102898201,0.1055269
19,-0.001148210841,-0.00090026,0.02896
20,-0.002524498229,-0.001509726,0.09284916
21,-0.000130497124,-0.000132578,0.1418609
22,-0.002170621159,-0.000520093,0.0663307
23,-0.00025562254,-0.00200810093,0.13791161504,1
24,-0.001161214590,-0.001246031520,0.129849342420,1
25,-0.00127346409,-0.004603555746,0.141605914421,1
26,-0.000045413108,-0.00003121383,0.149987247871,1
27,-0.002036491829,-0.004791427394,0.171568446729,1
28,-0.000744451244,-0.000438467095,0.190049606639,1
29,-0.000332702211,-0.001770358609,0.163821577995,1
```

**Formatted  
txt file**

**Python  
script**



# Python for Paraview

- Readers, filters, render views.
- Each visualisation can be generated by a **python script**, that takes as input the results of the simulation.
- Starting from this code, **anyone** can run the visualisation scripts and setting the appropriate parameters

...let's get started!

# General settings (1)

reset and read

```
from paraview.simple import *
import sys
import os
import glob

#indicates where the output files (of the simulation) can be found
output_path='/Users/diletta/Desktop/format_output/'

#indicates where all imported scripts can be found
PATH_TO_SCRIPTS="/Users/diletta/Desktop/Fermiworking/fermascripts/paraview/"

sys.path.append(os.path.abspath(PATH_TO_SCRIPTS))

#reset current session
from reset import *
reset_all()

#reading input data
electrons_ = CSVReader(FileName=glob.glob(output_path+"electrons_*"))
```

# General settings (2)

## *text annotations*

```
protonsText = Text()
protonsTextDisplay = Show(protonsText, renderView1)
protonsText.Text = 'Protons'
protonsTextDisplay.FontSize = 8
protonsTextDisplay.WindowLocation = 'AnyLocation'
RenameSource('ProtonsText', protonsText)

protonsTextDisplay = GetDisplayProperties(protonsText,
view=renderView1)
protonsTextDisplay.Color = [0.12259098191805905, 1.0,
0.16134889753566797]
protonsTextDisplay.Position= [0.90,0.80]
```

# General settings (3)

## *creating the pipe*

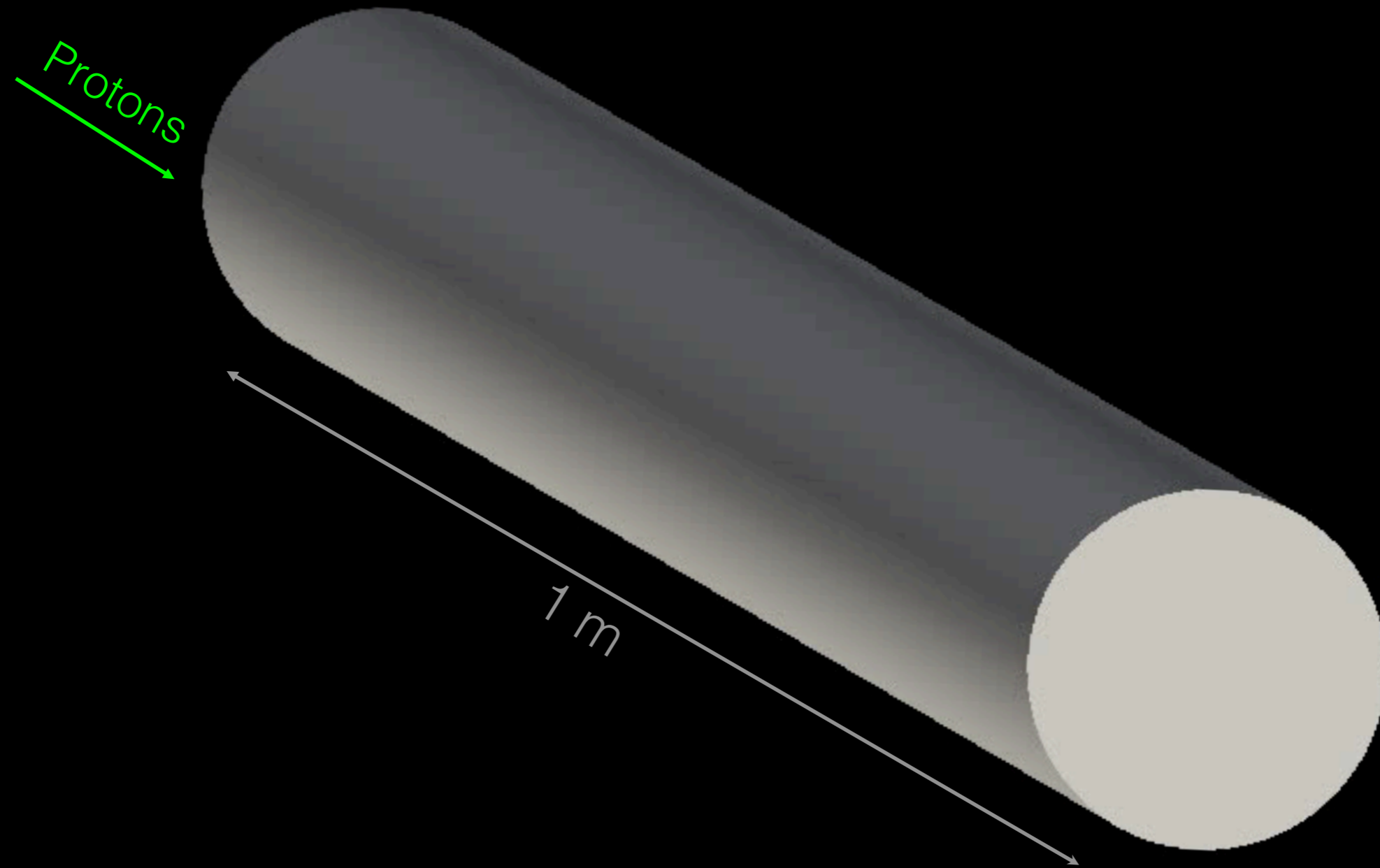
```
# create a new cylinder (the pipe)
cylinder1 = Cylinder()
cylinder1.Resolution = 350
cylinder1.Height = 1.1
cylinder1.Radius = 0.03
cylinder1.Center = [0.0, 0.0, 0.0]

# Transforming the pipe
transform1 = Transform(Input=cylinder1)
transform1.Transform = 'Transform'
transform1.Transform.Translate = [0.0, 0.0, 0.55]
transform1.Transform.Rotate = [90.0, 0.0, 90.0]
transform1.Transform.Scale = [1.0, 1.0, 1.0]

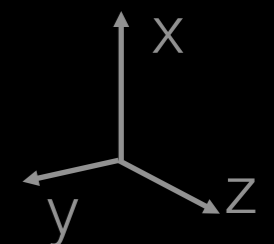
# show data in view
transform1Display = Show(transform1, renderView1)
```

# Evolution of the electron column

$B=0.1\text{ T}, V=-10\text{ V}$



Protons  
Electrons  
H2+



# Evolution of the electron column

*python script*

```
#creating a new reader
electrons_ =
CSVReader(FileName=glob.glob(output_path+"electrons_*"))

# filtering
tableToPoints1 = TableToPoints(Input=electrons_)
tableToPoints1.XColumn = 'Field 1'
tableToPoints1.YColumn = 'Field 2'
tableToPoints1.ZColumn = 'Field 3'

# filtering
clip1 = Clip(Input=tableToPoints1)
clip1.ClipType = 'Scalar'
clip1.Scalars = ['POINTS', 'Field 3']
clip1.Value = 0.0
```



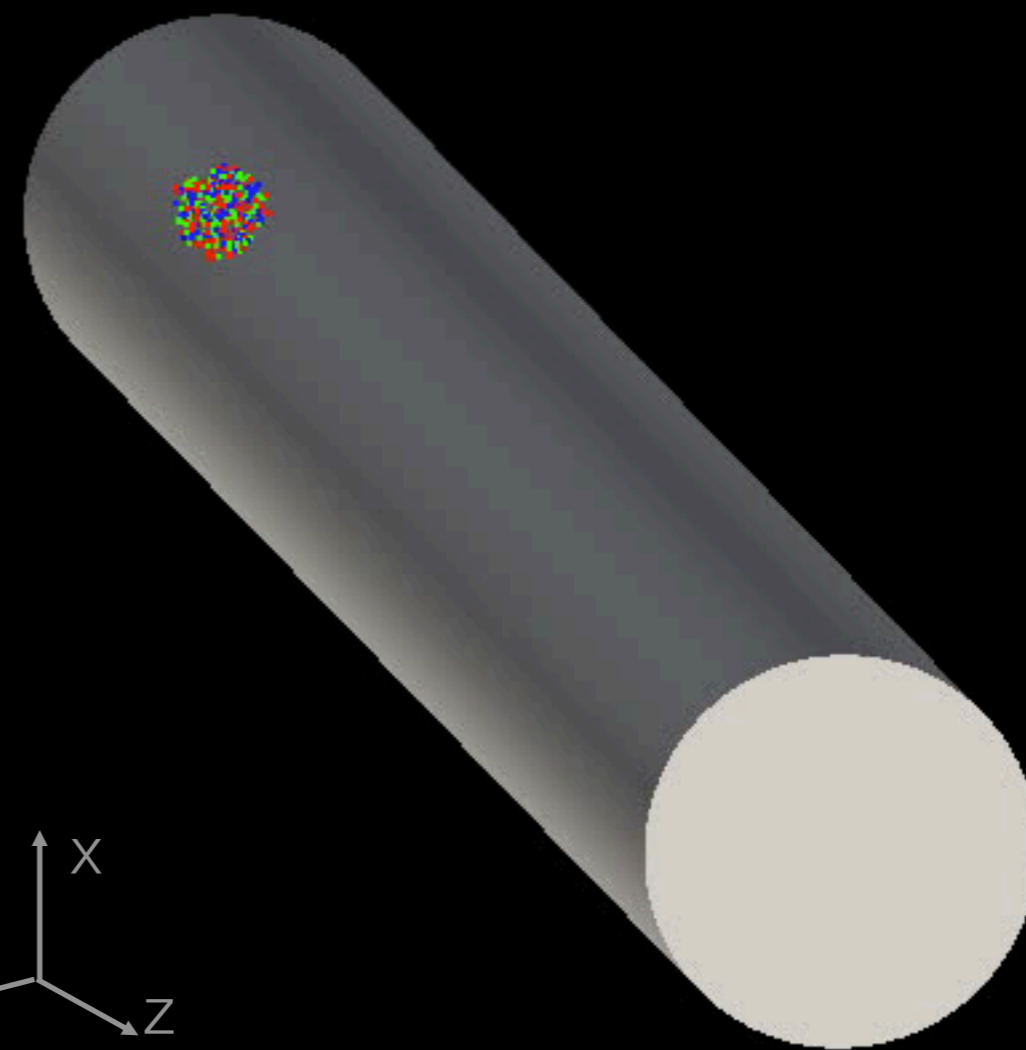
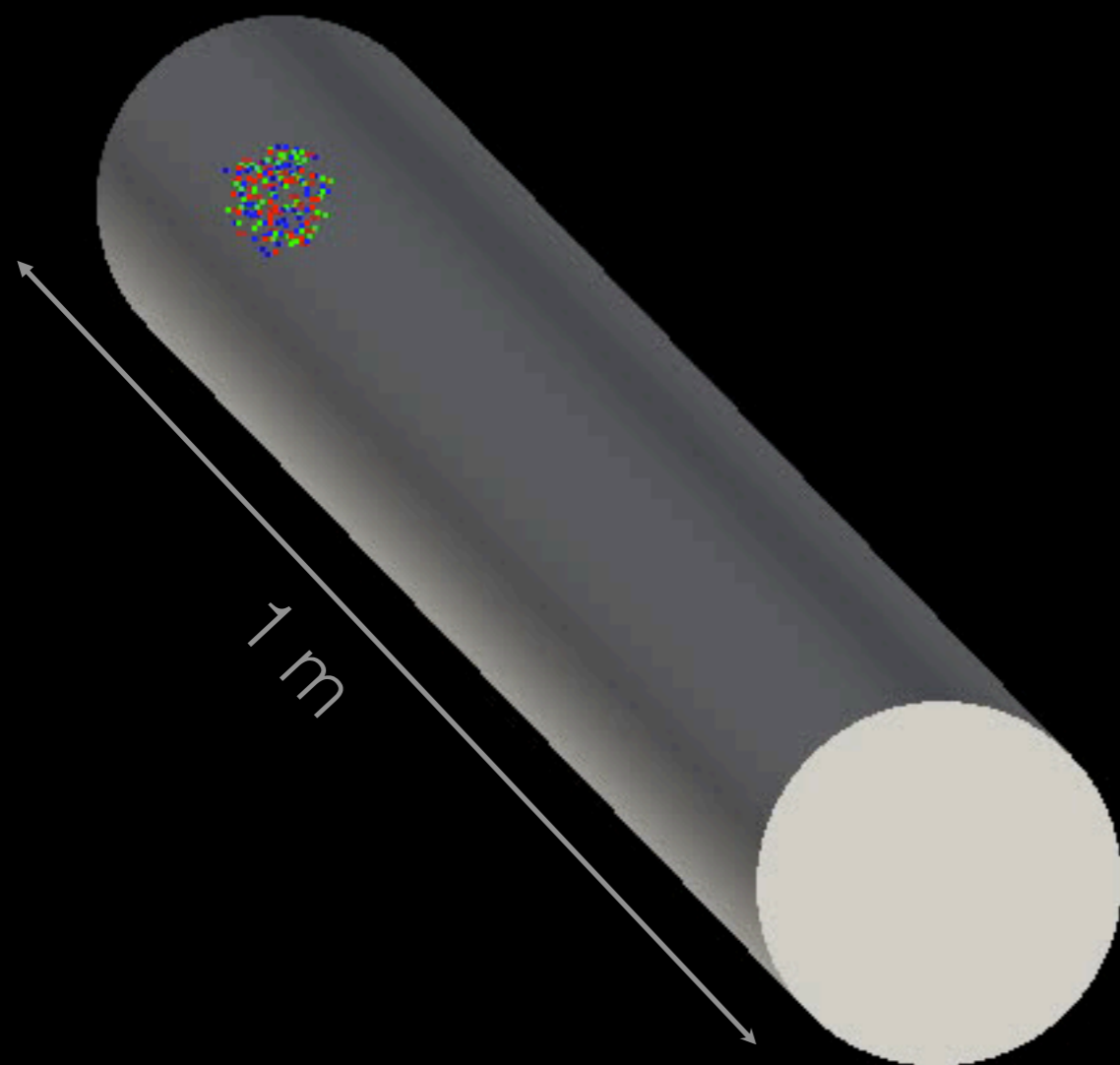
# Evolution of the electron column

comparing different runs

$B=0.1\text{ T}$ ,  $V=-10\text{ V}$

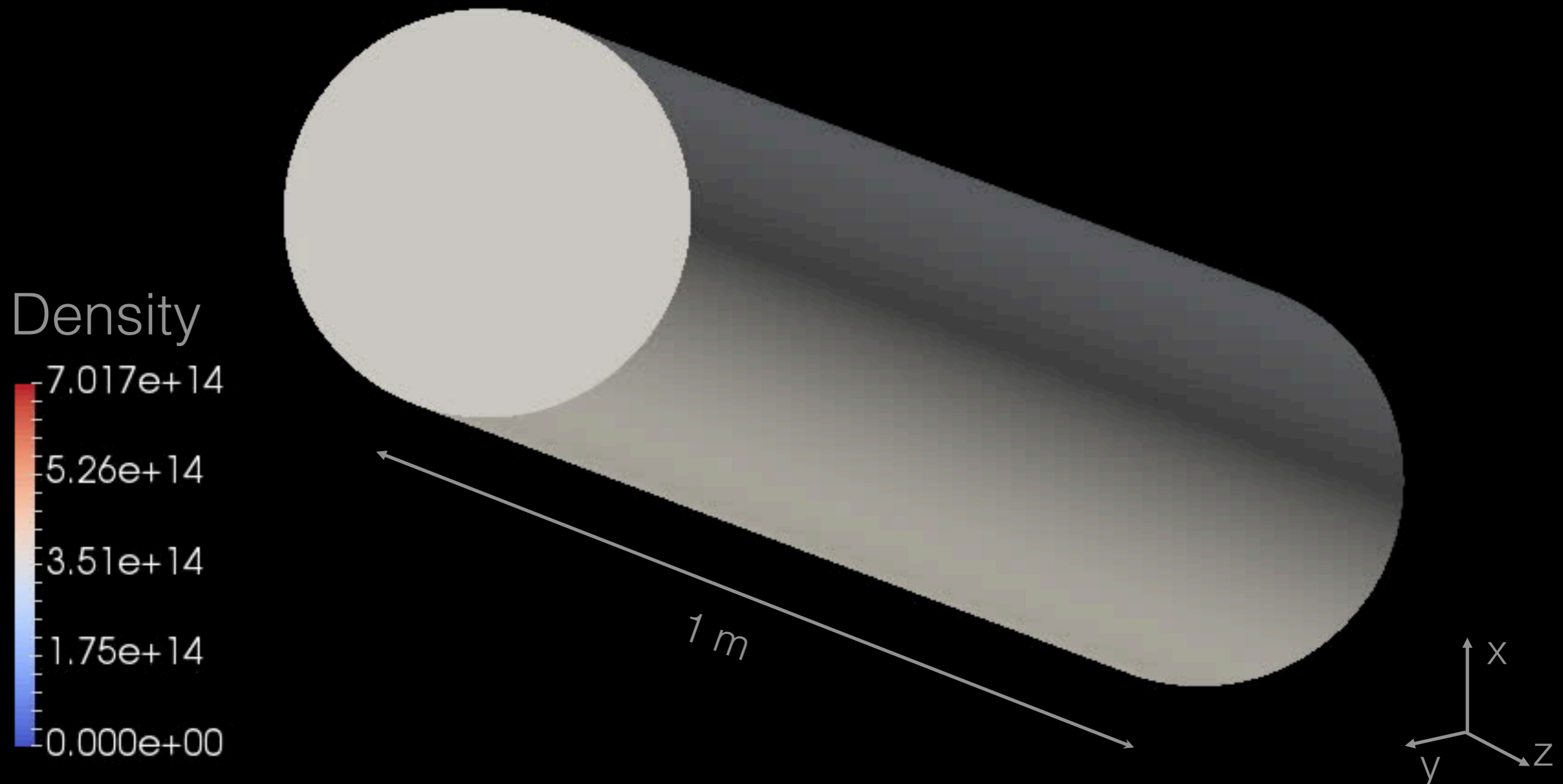
$B=0.1\text{ T}$ ,  $V=-5\text{ V}$

Protons  
Electrons  
 $\text{H}_2^+$



# Mapping the electron density

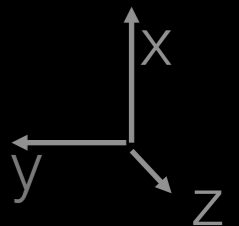
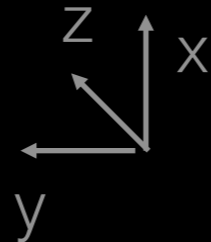
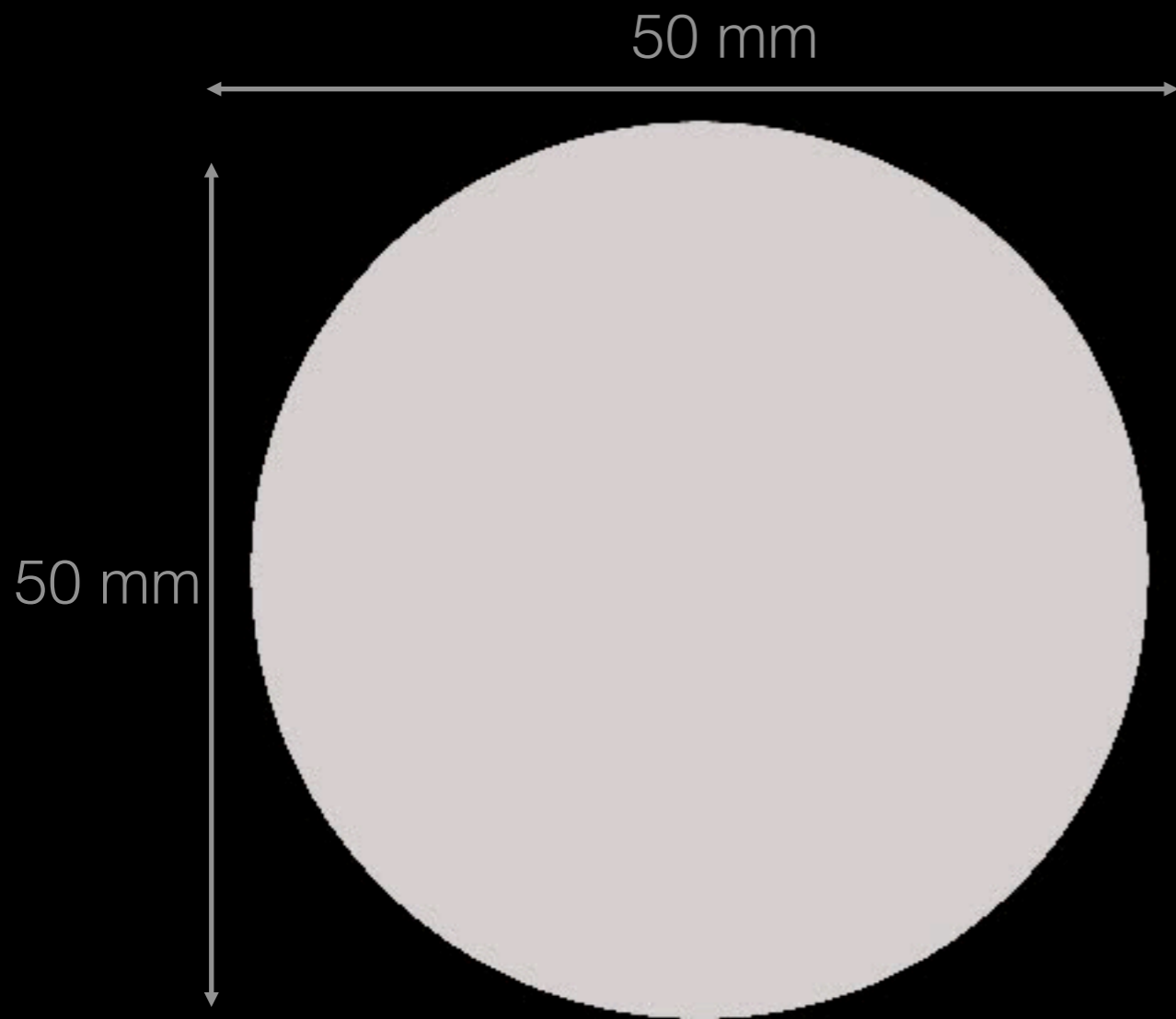
$B=0.1\text{ T}$ ,  $V=-10\text{ V}$



# Detail: end views

$B=0.1\text{ T}$ ,  $V=-10\text{ V}$

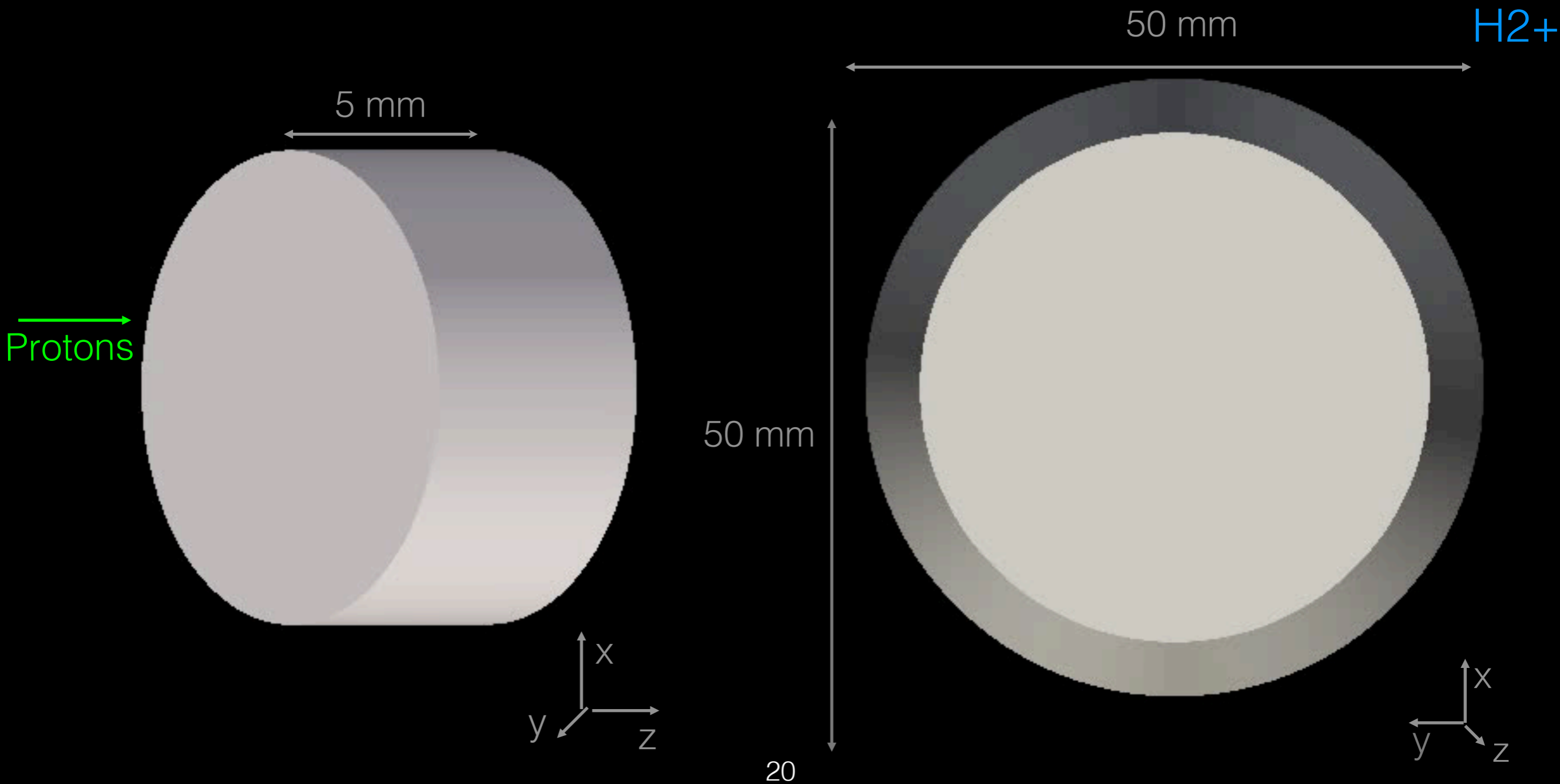
Protons  
Electrons  
 $\text{H}_2^+$



# detail: slice views

$B=0.1\text{ T}$ ,  $V=-10\text{ V}$

Protons  
Electrons  
 $\text{H}_2^+$



# detail: slice views

python script

```
Threshold_min= 0.54 #where the slice begins when z=0  
Threshold_max=0.58 #where the slice ends when z=1
```

```
# create a new slice
```

```
isoVolume1 = IsoVolume(Input=tableToPoints1)  
isoVolume1.InputScalars = ['POINTS', 'Field 3']  
isoVolume1.ThresholdRange = [Threshold_min,  
Threshold_max]
```

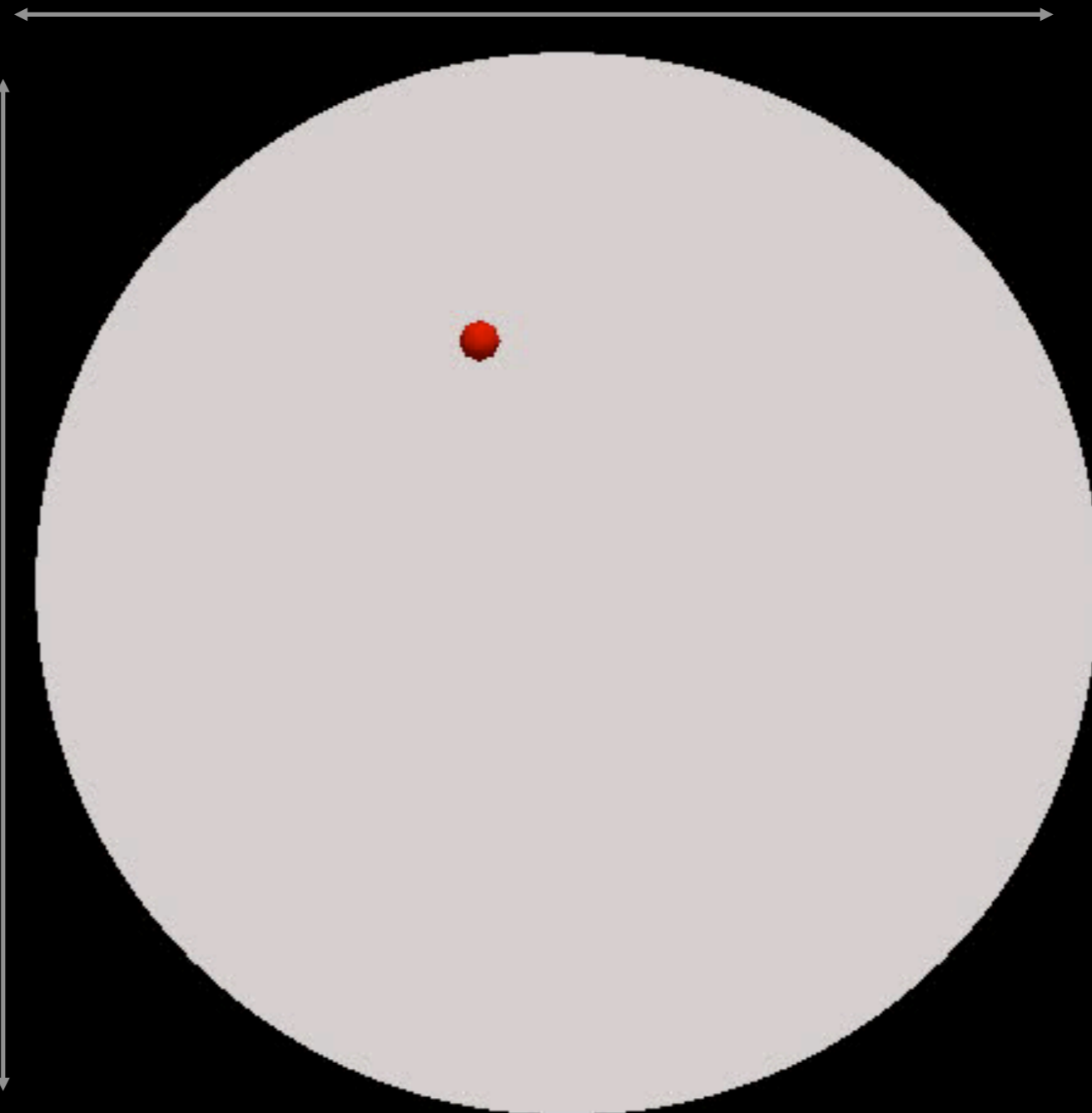
```
# show data in view
```

```
isoVolume1Display = Show(isoVolume1, renderView1)
```

# Detail: tracking an electron\*

$B=0.1\text{ T}$ ,  $V=-10\text{ V}$

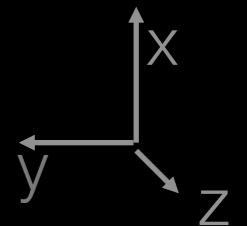
50 mm



x coord



50 mm



# Detail: tracking an electron\*

python script

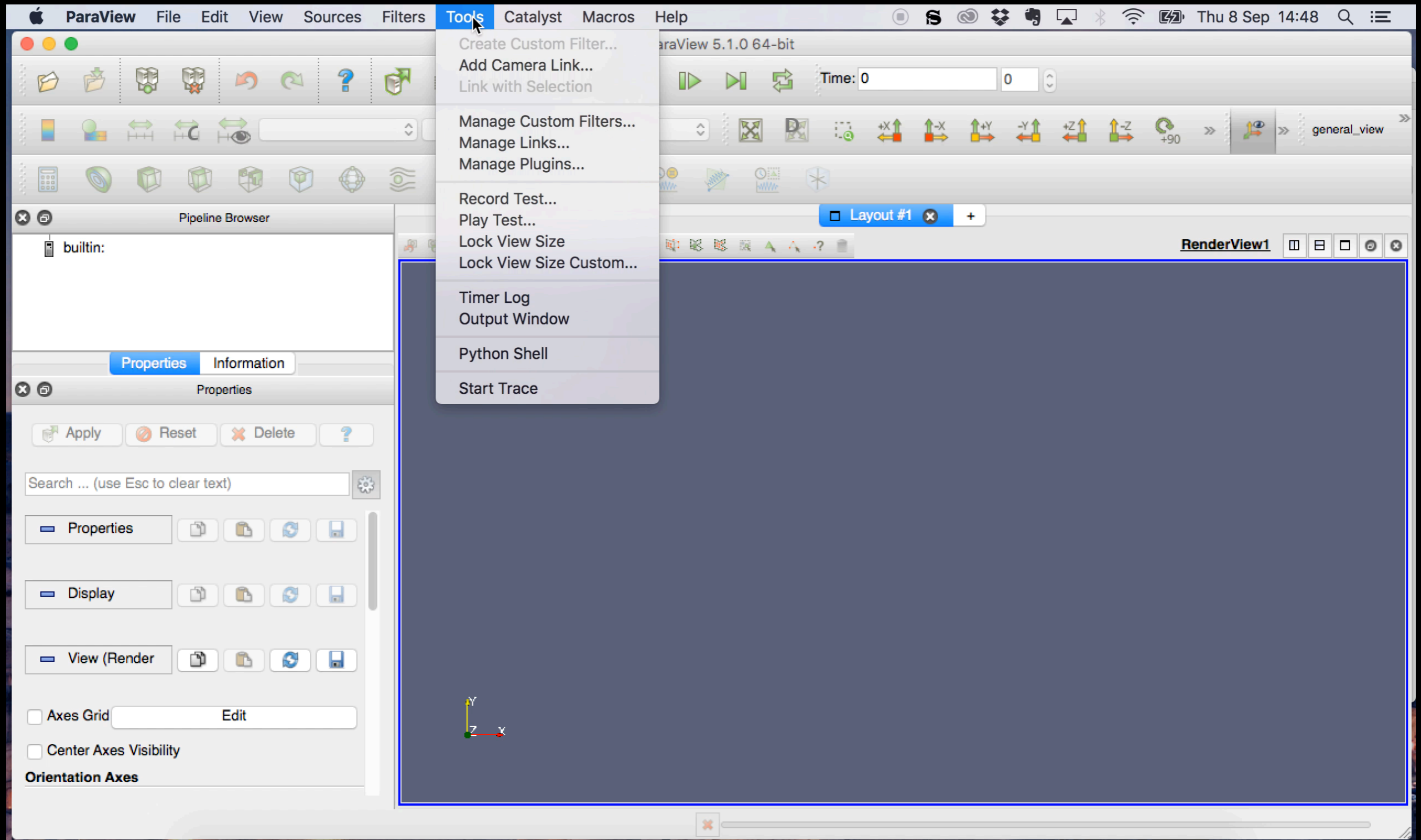
```
MASK_POINTS=10000
PATH_LENGTH= 1000
PARTICLE_RADIUS= 0.002

# create a new 'Temporal Particles To Pathlines'
temporalParticlesToPathlines1 =
TemporalParticlesToPathlines(Input=tableToPoints1,
    Selection=None)
temporalParticlesToPathlines1.MaskPoints = MASK_POINTS
temporalParticlesToPathlines1.MaxTrackLength = PATH_LENGTH
temporalParticlesToPathlines1.MaxStepDistance = [1.0, 1.0, 1.0]
temporalParticlesToPathlines1.IdChannelArray = 'Global or Local IDs'

temporalParticlesToPathlines1Display =
Show(temporalParticlesToPathlines1, renderView1)
```

# A tutorial

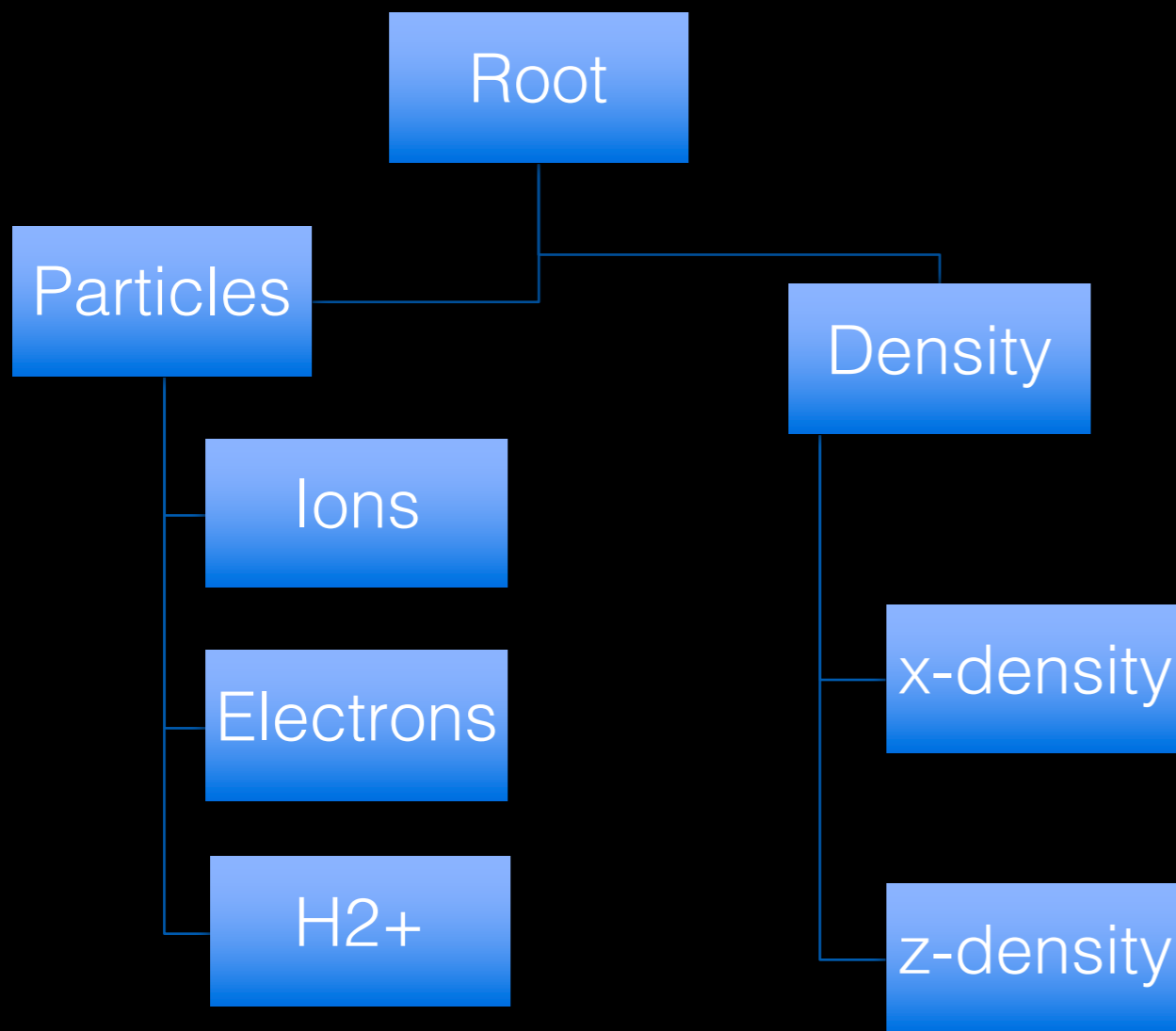
## integrating python and GUI





# breaking bottlenecks...

## HDF5



- Open source technology suite for **high volume and/or complex data**
- Guarantees **flexible, efficient** storage and I/O
- **Xml**-style hierarchical structure

# The pipeline

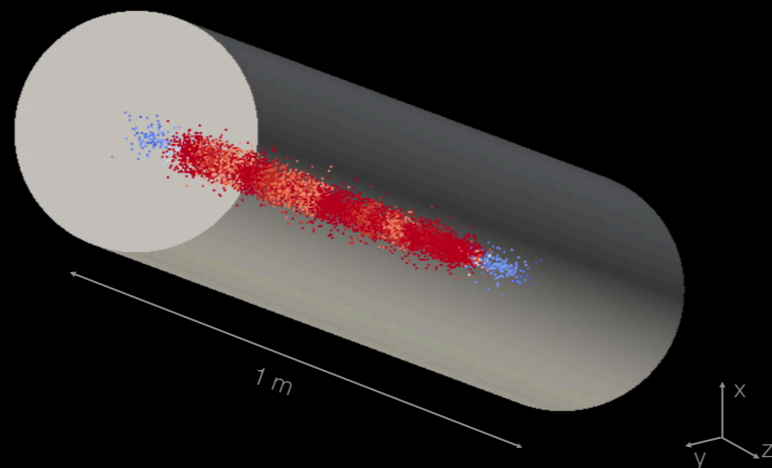
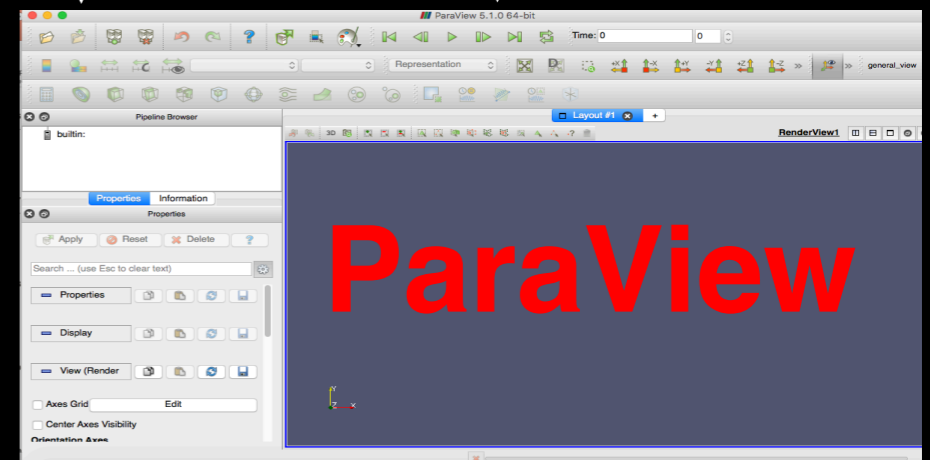
**WARP  
simulation**



```
electrons_003500.txt
0,-0.002818353819,-0.001225297198,0.028058384296,1
1,-0.002264426841,-0.001489587945,0.009352382714,1
2,-0.005955534822,0.006715599229,0.006084227214,0
3,-0.001761828671,0.000628601768,0.014034560853,1
4,0.000102867179,0.002258199701,0.033710922673,0
5,-0.000485350741,-0.001638509203,0.166490387816,0
6,-0.001952805705,-0.002930052861,0.120021844337,1
7,-0.001420468301,-0.006986056905,0.164035460119,0
8,-0.004676210441,-0.002504099716,0.191490619737,0
9,-0.000958406668,-0.001269181069,0.182374964505,0
10,-0.005897717914,-0.001037163619,0.156858445986,0
11,-0.003145680636,-0.005251530360,0.179088725738,0
12,-0.004729869146,-0.002604455031,0.132115619966,1
13,-0.003476671136,-0.000100000000,0.000000000000,0
14,-0.003257880059,-0.000000000000,0.000000000000,0
15,-0.002618480000,0.000000000000,0.000000000000,0
16,-0.001392630000,0.000000000000,0.000000000000,0
17,-0.001400910000,0.000000000000,0.000000000000,0
18,-0.006101010000,0.000000000000,0.000000000000,0
19,-0.001148210041,-0.00259098269,0.183028962777,1
20,-0.002524498229,-0.002350978834,0.149284916489,1
21,-0.000130497124,-0.000213257618,0.141418609452,1
22,-0.002170621159,-0.003052001721,0.162663307911,1
23,-0.000825562254,-0.002868100934,0.137891161364,1
24,-0.001161214590,-0.001246031520,0.129849342420,1
25,-0.001273746409,-0.004603555746,0.141605914421,1
26,-0.000045413108,-0.000033121383,0.149987247871,1
27,-0.002036491829,-0.004791427394,0.171568446729,1
28,-0.000744451244,-0.000438467095,0.190049606639,1
29,-0.000332700000,0.000000000000,0.000000000000,1
```

**HDFS file**

**Python script**



# A simple implementation

```
#create an hdf5 file for each timestep
h5file= PWpyt.PW("timestep_%06d.h5" % iter)

#create an array of particles
ions_r = []

#for every ion, add its info to the previous array:
    ions_r.append([i,xions[i],yions[i],zions[i]])

#convert the array into a numpy array
ions_np= np.array(ions_r)

#add the array of ions as an attribute of that file
h5file.protons= ions_np

h5file.close()
```

# What comes next

- Practical issues such as ring dynamics of the primary proton beam and multiple passes thru the e-column
- **2016** North American Particle Accelerator Conference (NAPAC16)

# What I learnt

- Physics is **really** complex
- One of the best ways to understand what is going on is **to see it**
- Understanding is one thing, making it **clear to others** is a completely different business
- It takes a lot of work to make the **complex look simple!**

# What I hope you'll take home

- Paraview is an extremely **flexible** and **versatile**, and it can be used for almost every type of data.
- The power of **visualisation** goes well beyond fancy graphics
- The strength of a simulation lies not only in the power and efficiency of the code, but also in the **usability** of its output
- The strength of any script lies in its **re-usability**

To Charles Thangaraj, Chong Shik Park and Giulio Stancari

To the **Fermilab** Accelerator Science and Technology (**FAST**) – Integrable Optics Test Accelerator  
(**IOTA**) department

To Giorgio Bellettini, Emanuela Barzi, Simone Donati and the CAIF associates

To my colleagues and roommates – great physics tutors

To Fermilab – a truly incredible opportunity

*Thank you!*