



Lorentz4 (beta version)



Nicola Rossi

and Alfredo C., Federico V., Francesca P., Marcello M

PTOLEMY General Meeting

Nimega - July 1-2, 2025

Outline

Motion of charged particles in static electromagnetic fields

Made to understand (analytical engine)

Meant for averaged squared physicists (ROOT/C++)

- Actual simulation tools
- Education and teaching

New features:

- Numerical (+ analytical) input E and B fields (CST, Comsol, ...)
- E and B from simple geometry
(B from wires, Laplace)
(E, from equipotential surfaces, BEM)

Repository

Prerequisite:

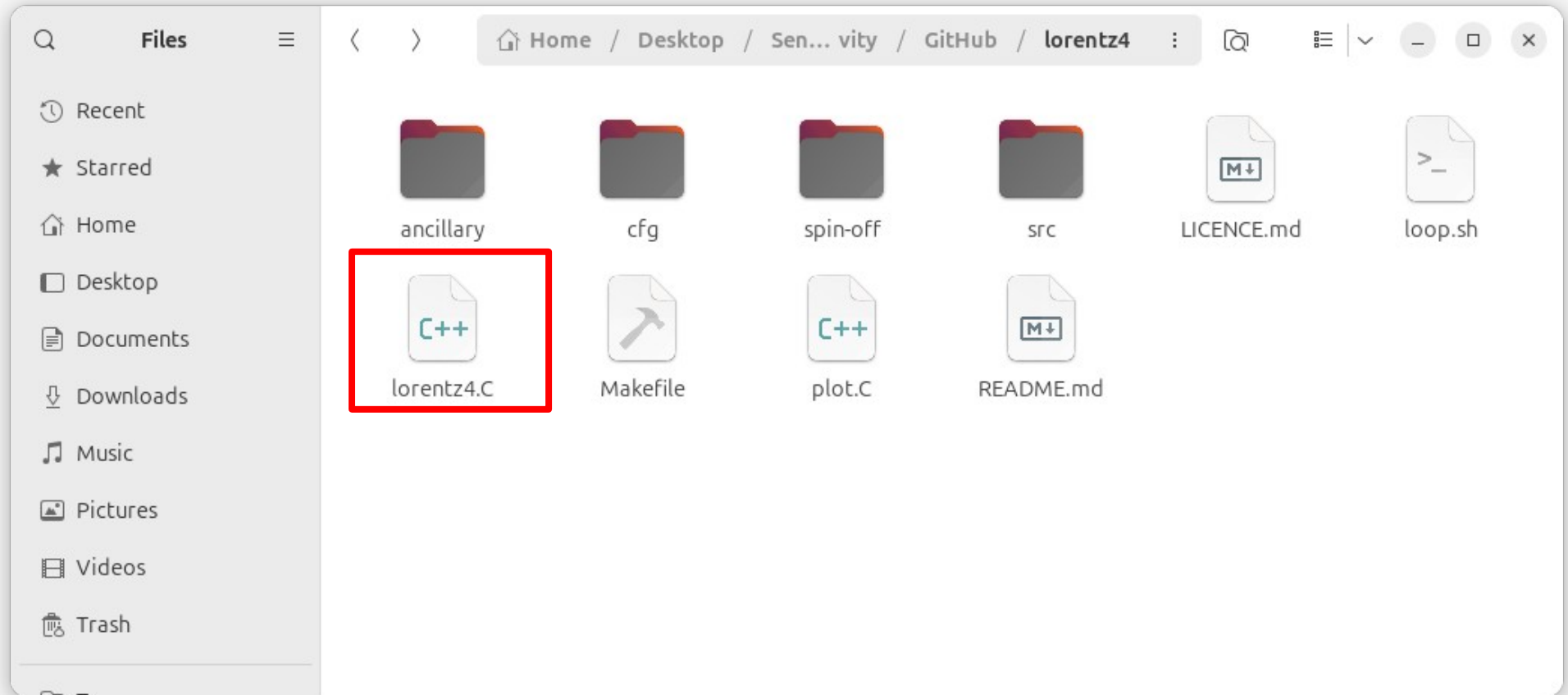
- ROOT/C++ (v6) installed
- GitHub account

mailto:

`nicola.rossi@lngs.infn.it`

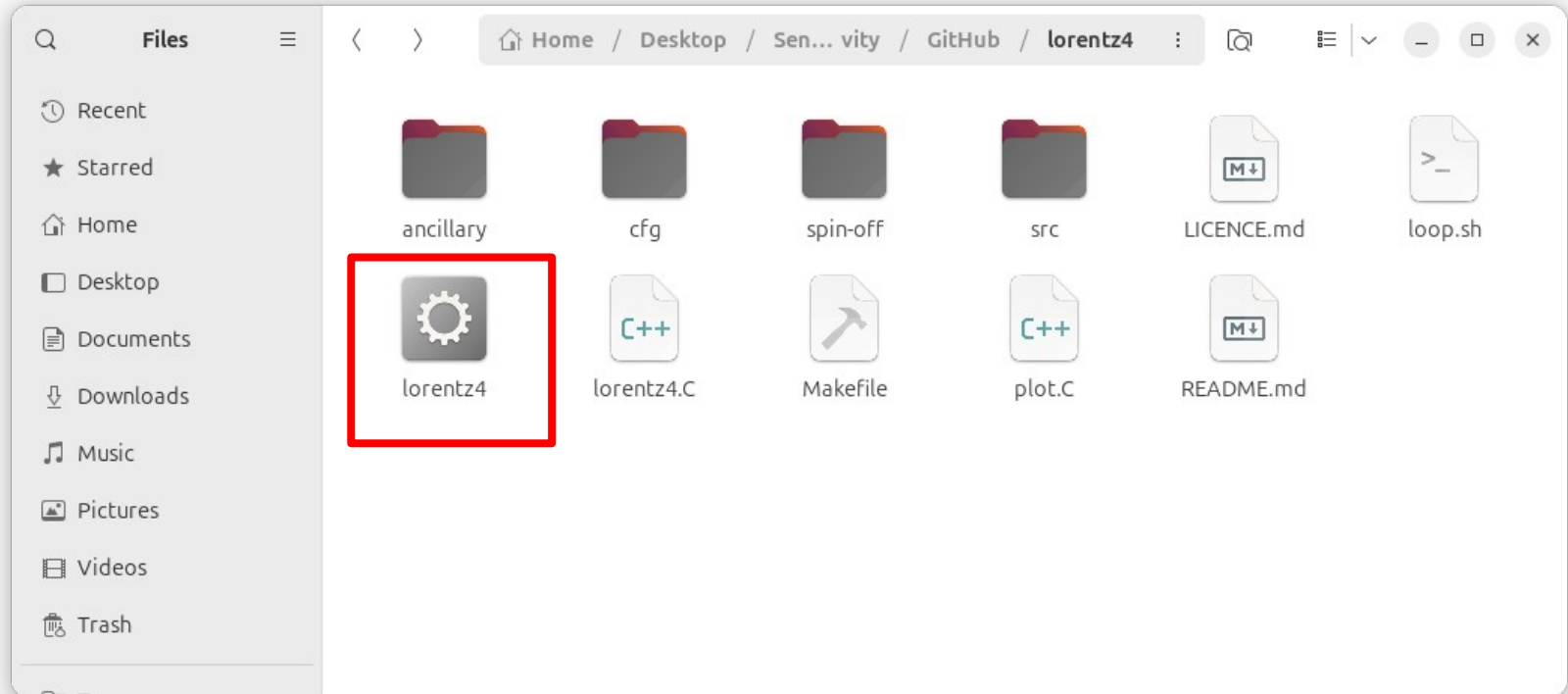
```
git clone git@github.com:gkrossi/lorentz4.git
```

Lorentz4 folder



How to compile

make



How to run

Optional

```
./lorentz4 0.0001 10 cfg/user.cfg user.root
```

exe

Step [ns]

Duration [ns]

Config file

Output

Relativistic formulas

$$\vec{p}(t) = \gamma m_0 \vec{v}(t)$$

$$K = \sqrt{(pc)^2 + (mc^2)^2} - mc^2$$

Kinetic energy

$$\dot{\vec{r}}(t) = \frac{\vec{p}(t)}{m_0 \gamma}$$

$$U = \int \vec{E} \cdot d\vec{r}$$

Potential energy

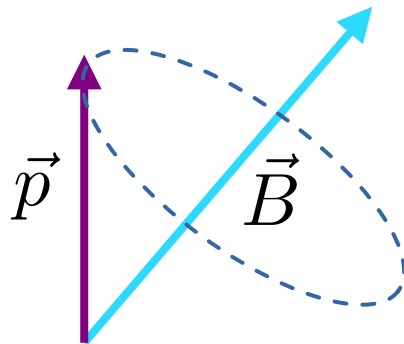
$$\dot{\vec{p}}(t) = q \left(\vec{E} + \frac{\vec{p}(t)}{m_0 \gamma} \times \vec{B} \right)$$

Integration: Runge-Kutta 4th (RK4)

RF emission

$$\frac{dp_{\perp}}{dt} \propto B^2 p_{\perp} \gamma$$
$$\frac{dp_{\parallel}}{dt} \simeq 0$$

Adiabatic
Condition



$$\frac{d\vec{p}}{dt} = \vec{F}_{\text{Lorentz}} + \vec{F}_{\text{Rad}}$$

Non conservative
(friction)

$$\vec{F}_{\text{Rad}} = -\frac{\mu_0 e^4}{6\pi c m_e^3} B \gamma \left(\vec{p} - \frac{\vec{p} \cdot \vec{B}}{B^2} \vec{B} \right)$$

Adaptive bin step

$$\begin{array}{c} \hline 2 \, dt \\ \hline dt \quad dt \\ \hline \end{array}$$

$$x(t + 2dt) \approx x_1 + (2dt)^5 \phi + \mathcal{O}(dt^6)$$

$$x(t + 2dt) \approx x_2 + 2dt^5 \phi + \mathcal{O}(dt^6)$$

$$x(t + 2dt) = x_2 + \frac{\Delta}{15} + \mathcal{O}(dt^6)$$

$$\Delta \equiv x_2 - x_1$$

Source (src) folder

Configuration
handling



L4config.h



L4fields.h



L4integrator.h



L4lib.h



L4track.h

Fields
(analytical,
numerical)

Integrator
(RK4)

Math
libraries

Trajectory
& observables

```
whirlpool.cfg
~/Desktop/Sensitivity/GitHub/lorentz4/cfg
Open Save
1 # Initial dynamics: energy (eV) and polar angles [deg]
2 E0      18000
3 theta0  30
4 phi0    90
5
6 # Initial position [m]
7 x0      0.3
8 y0      0
9 z0      0
10
11 # Exteral fields (true if filename is present, 0 otherwise )
12 ExtE    0
13 ExtB    0
14 WhichF  EB
15
16 # Eletric Field [V/m]
17 Ex      0
18 Ey      0
19 Ez      "0.1 *1e5"
20
21 # Magnetic Field [T]
22 Bx      "0.01*sin(atan2(y,x))/ sqrt(x*x+y*y)"
23 By      "-0.01*cos(atan2(y,x)) /sqrt(x*x+y*y)"
24 Bz      0
25
26 # Radiation energy loss: 0 = disabled; 1 = enabled
27 ifRad    0
28
29 # Stop conditions on trajectory (x, y, z)
30 StopIf   "x>1e9 && y > 1e9"
31
32 # Down Sampling: max number of points in the track (Np > 8)
33 Np       10000000
34
35 # Adaptive step: 0 = disabled; 1e-8 (or better) precision otherwise
36 Prcs     1e-8
37
38 # Verbose and plots: 0 = disabled; 1 = enabled
39 Verb     1
40
41 # Save options (kinematics, frenet, fields): 0 = disabled; 1 = enabled
42 Skin     1
43 Sfren    1
44 Sfiel    1
Plain Text Tab Width: 8 Ln 1, Col 47 INS
```

New
config file

Demos (cfg)



`init.cfg`

`Bconst.cfg`

`hemispheres.cfg`

`nicola.cfg`

`tully.cfg`

`EparB.cfg`

`hemispheres_ext.cfg`

`ptolemy.cfg`

`v_selector.cfg`

`ext_fields.cfg`

`RFcalib.cfg`

`whirlpool.cfg`

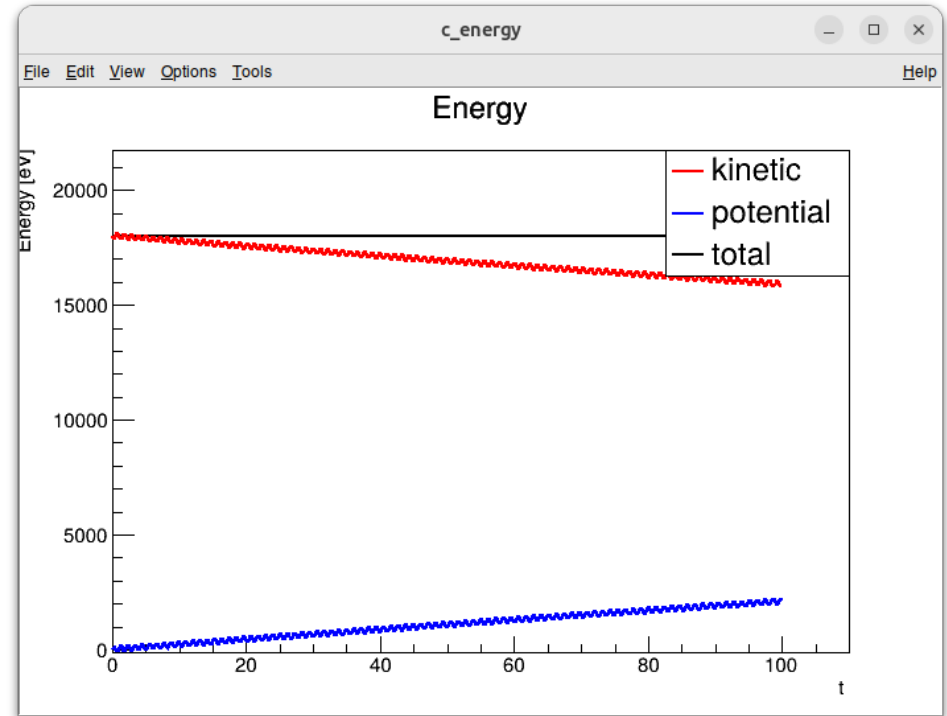
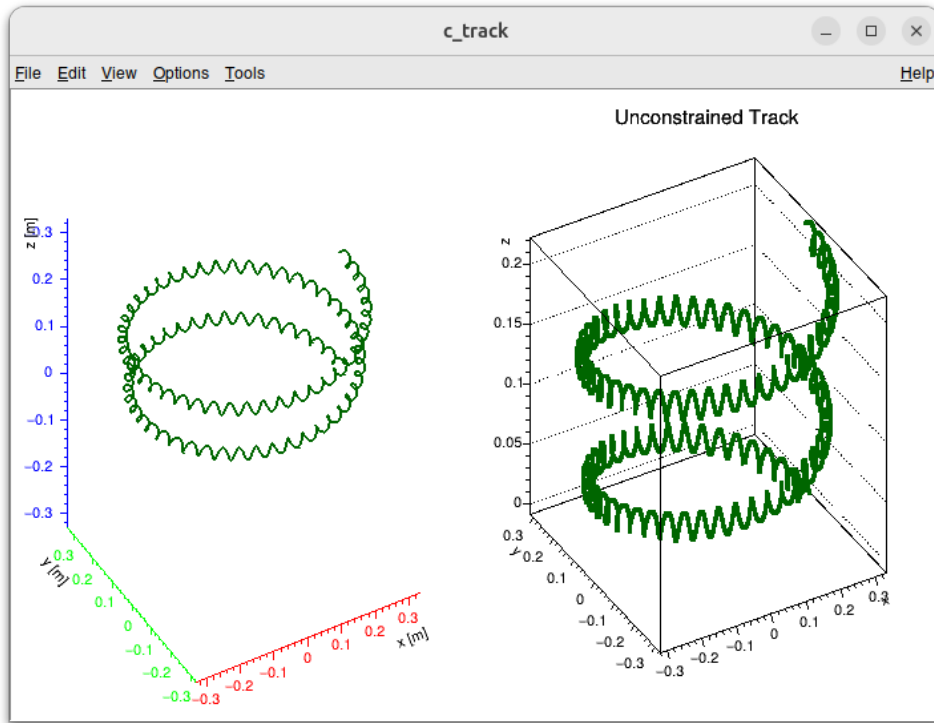
`gauss.cfg`

`katrin.cfg`

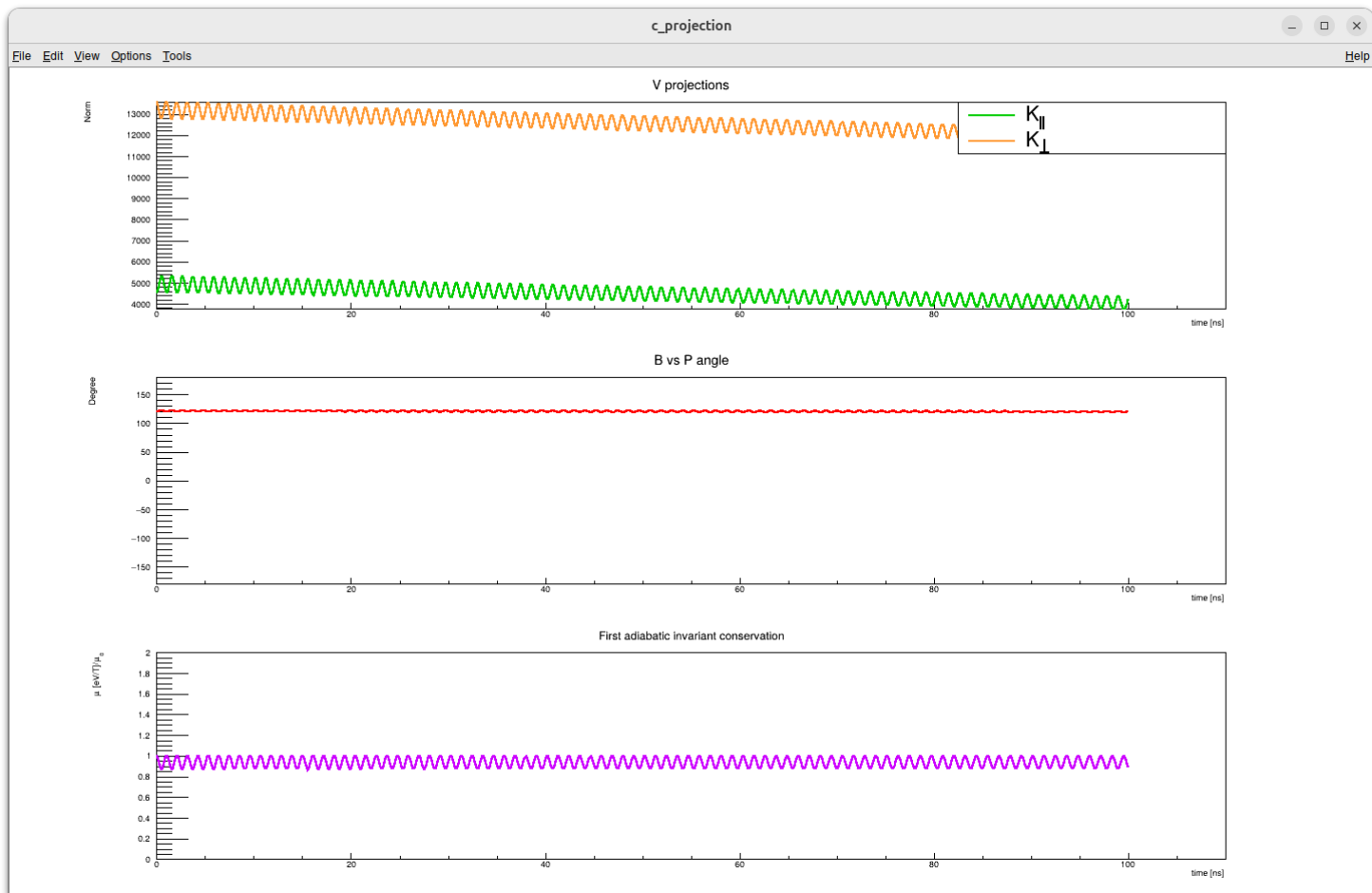
`SimEB.cfg`

`wonyong.cfg`

Trajectory



Energy

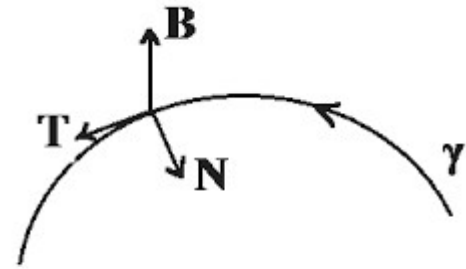
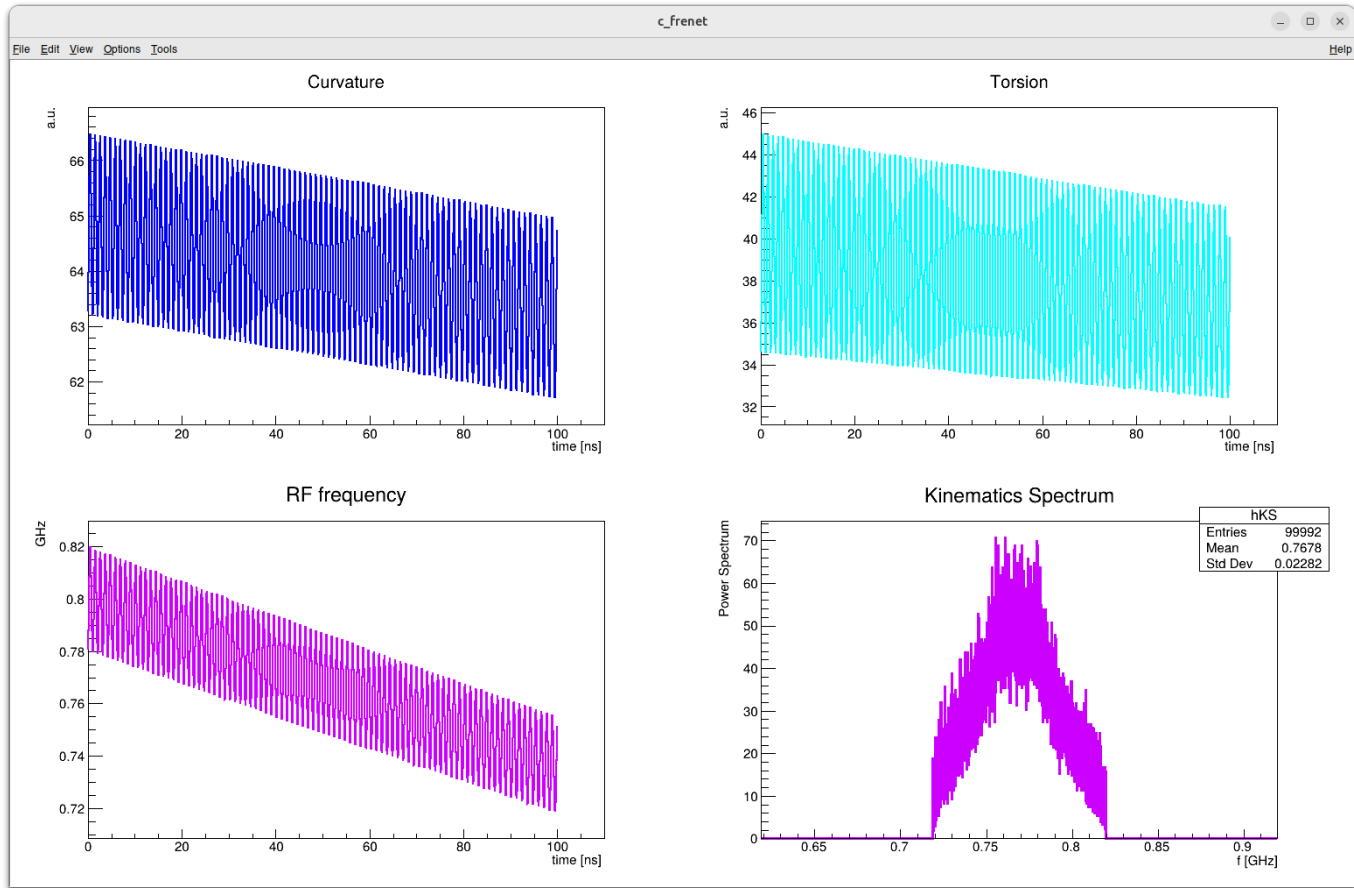


K_{par}

K_{ort}

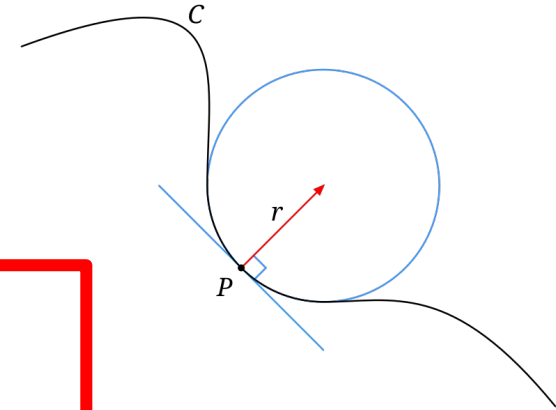
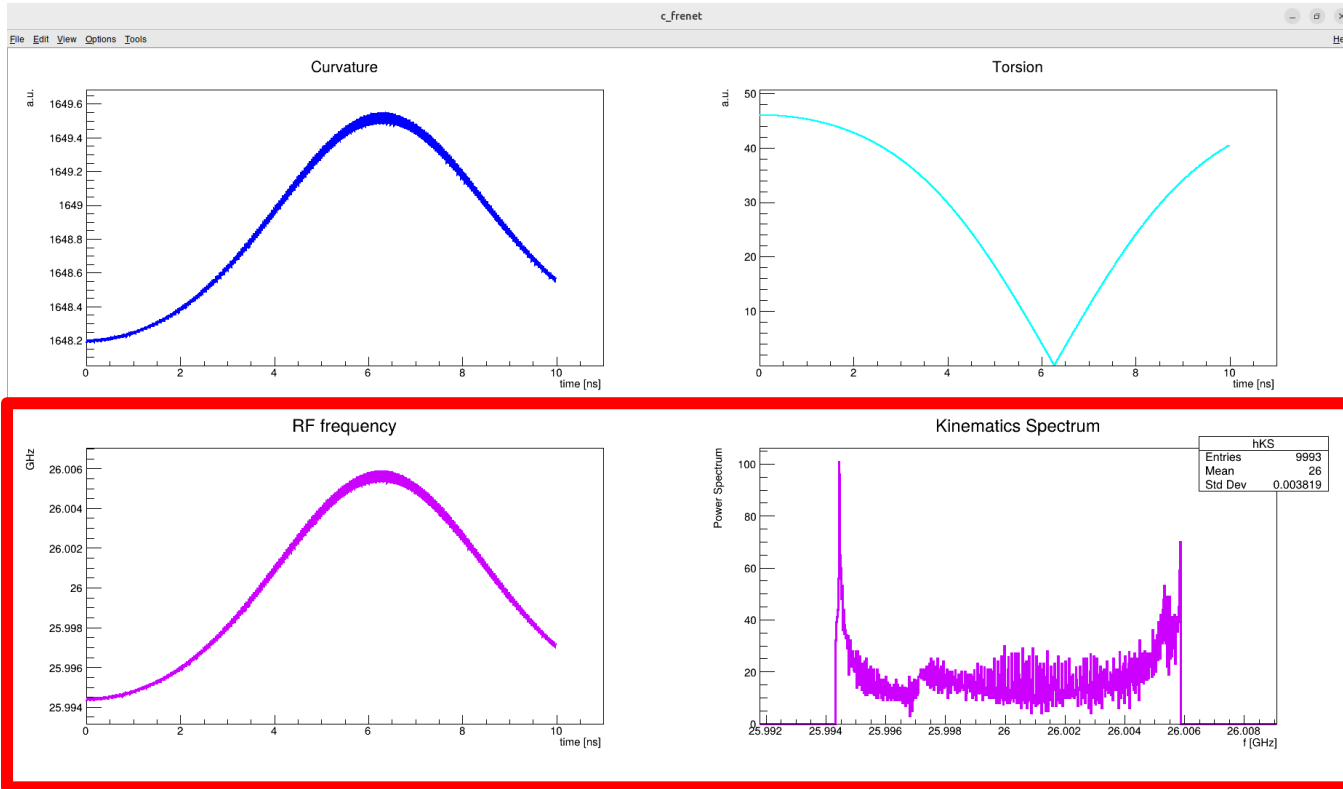
ϑ

μ



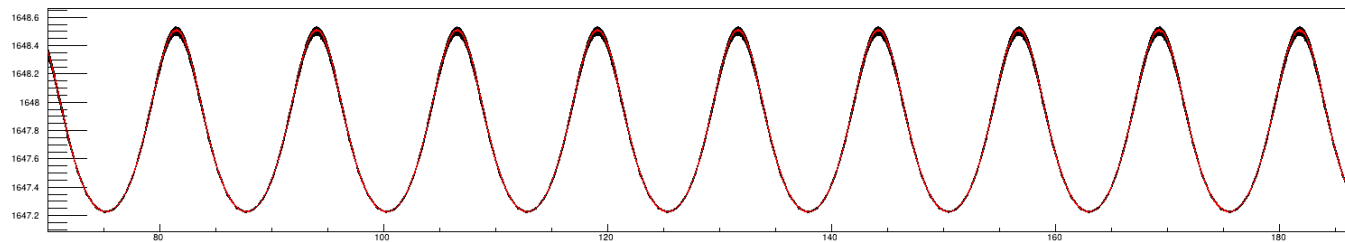
Frenet Params.

Kinematic spectrum

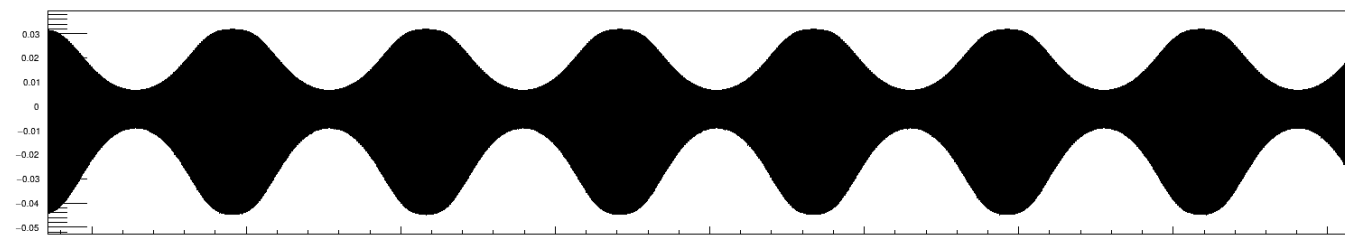


$$\omega_c = \frac{1}{\gamma} \frac{|\vec{r}' \times \vec{r}''|}{|\vec{r}'|^2}$$

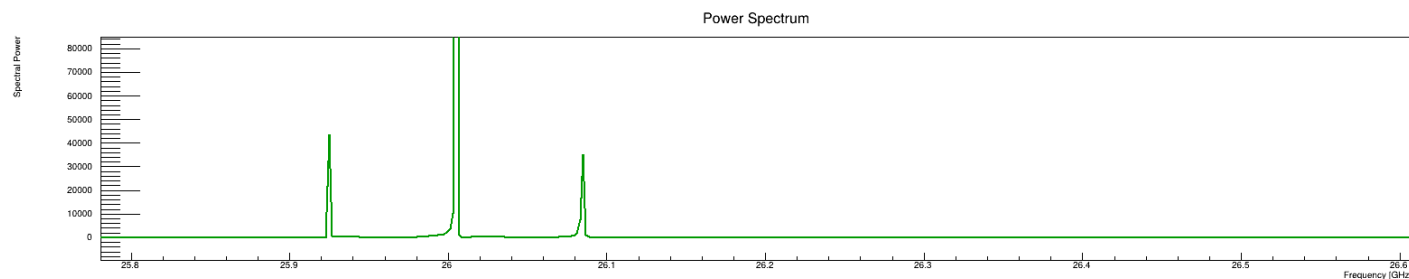
Curvature FFT



Curvature

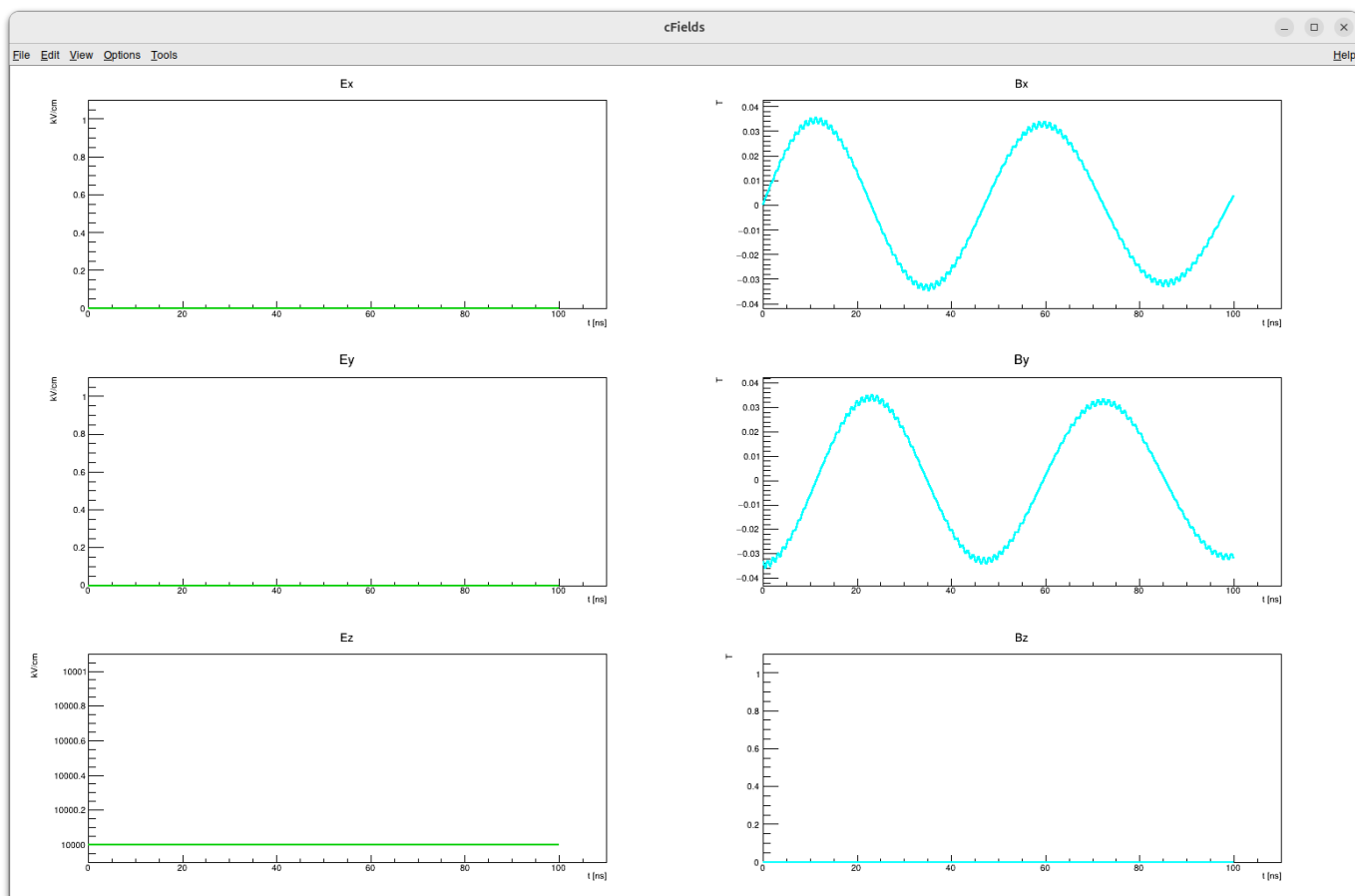


Residuals
After LOESS
Detrend



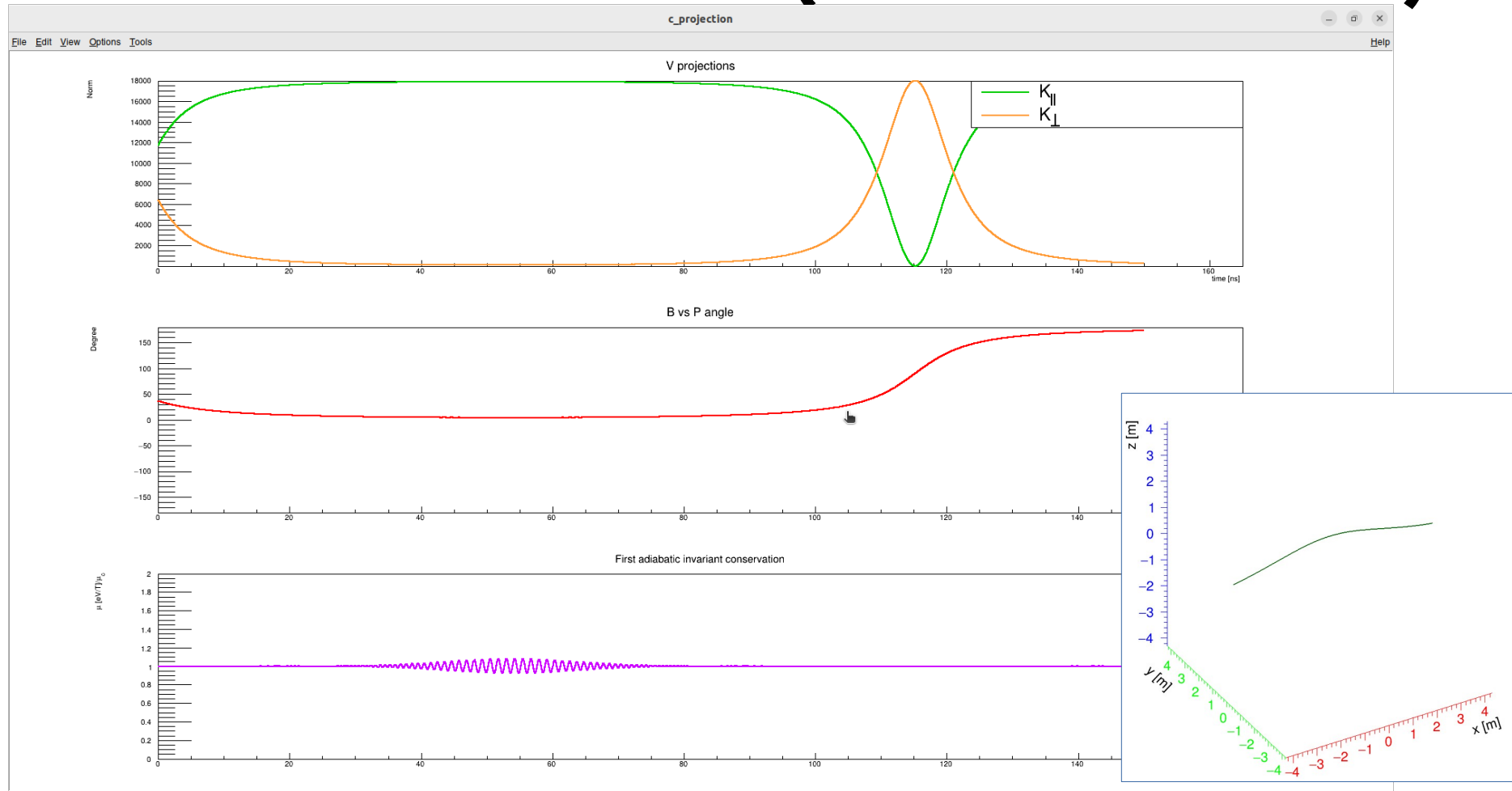
FFT
(Sidebands)

To be implemented

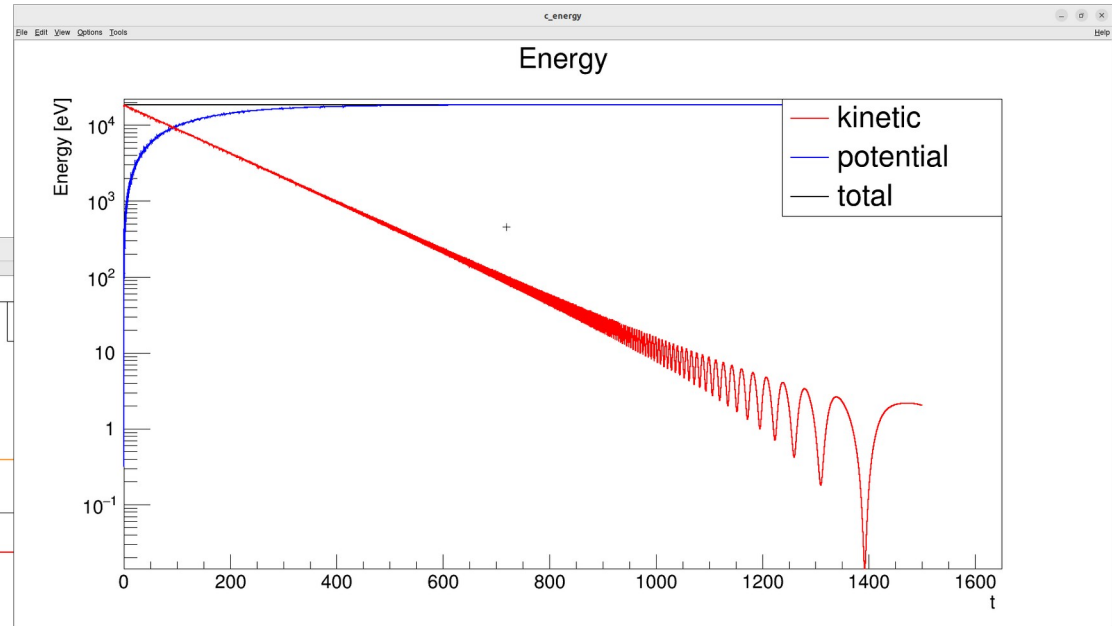
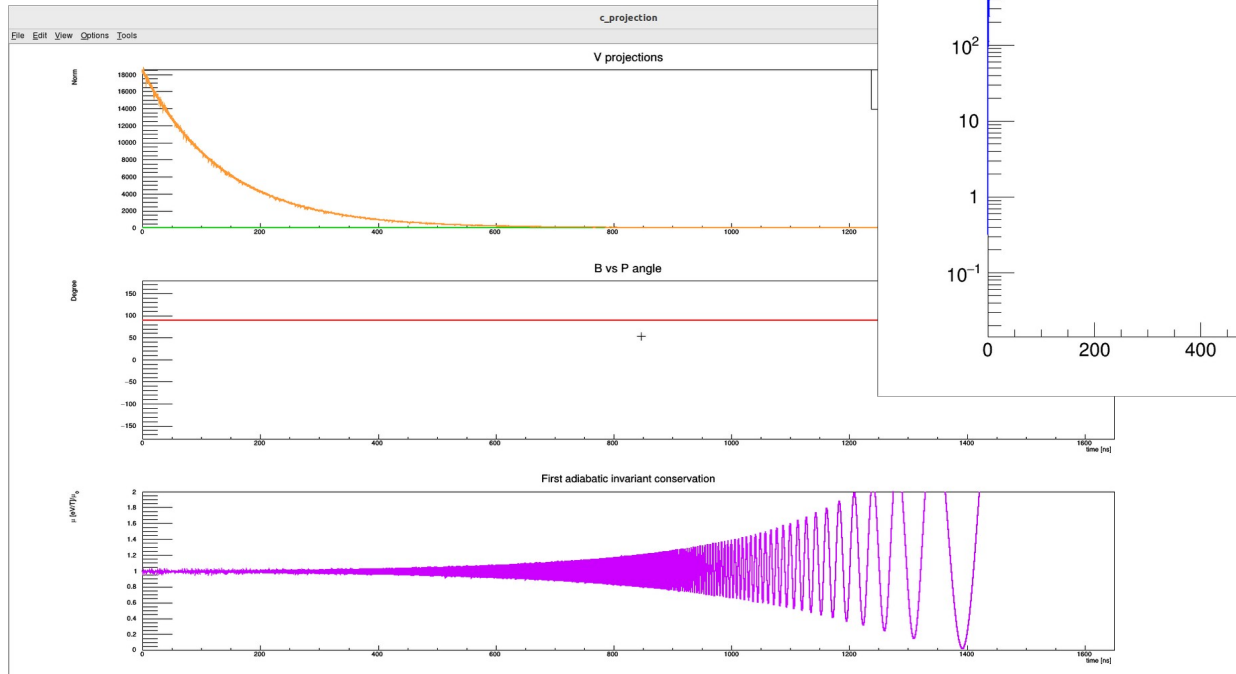


**E & B
fields
on track**

MAC-E filter (katrin.cfg)



Dyn.E.M. Filter (ptolemy.cfg)



Looping on many events

```
Open ▾  *loop.sh  Save  -  □  ×
~/Desktop/GIT/lorentz4

1 #!/bin/bash
2
3 n=0;
4 for energy in 10000 18500
5 do
6 echo "# Initial dynamics: energy (eV) and polar angles [deg]"
7 E0 $energy
8 theta0 45
9 phi0 0
10
11 # Initial position [cm]
12 x0 0.
13 y0 0
14 z0 0
15
16 # Electric Field [kV/cm]
17 Ex 3.*x
18 Ey 1.
19 Ez 0
20
21 # Magnetic Field [T]
22 Bx -1
23 By 0
24 Bz 0
25
26 # Down Sampling: max number of points in the track
27 Np 100000
28
29 # Step doubling precision error estimation: Enable = 1, Disable = 0
30 Prcs 0" > cfg/loop_${n}.cfg
31
32 ./lorentz4 0.0001 1 cfg/loop_${n}.cfg out_${n}.root
33 ((n=n+1))
34 done
35 root "plot.C($n)"

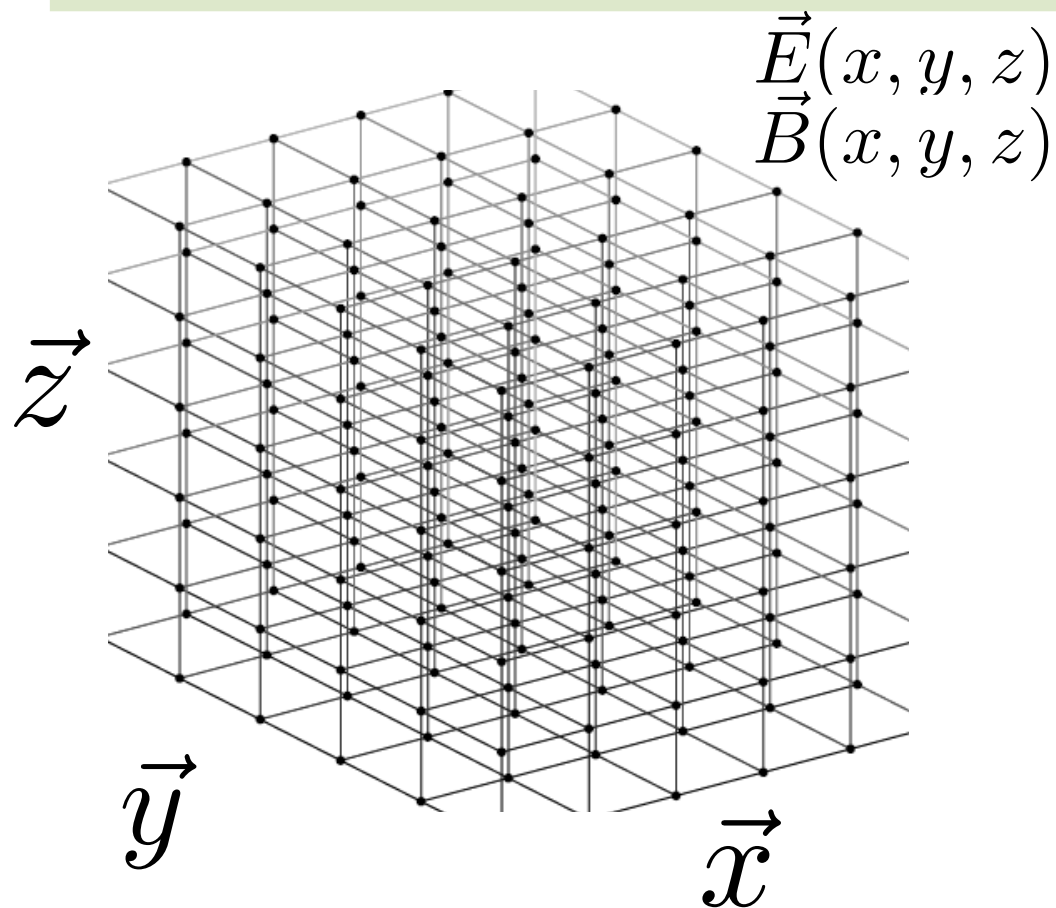
sh ▾ Tab Width: 8 ▾ Ln 37, Col 1 ▾ INS
```

`./loop.sh`
(parallel)

`root plot.C`
(TChain)

Add `qsub` for parallel
computing

External simulated fields



Instructions in
ancillary/rootify

Rootfile

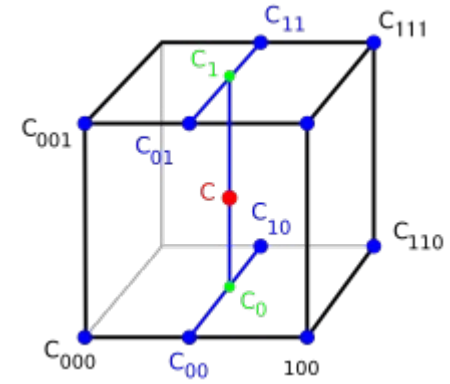
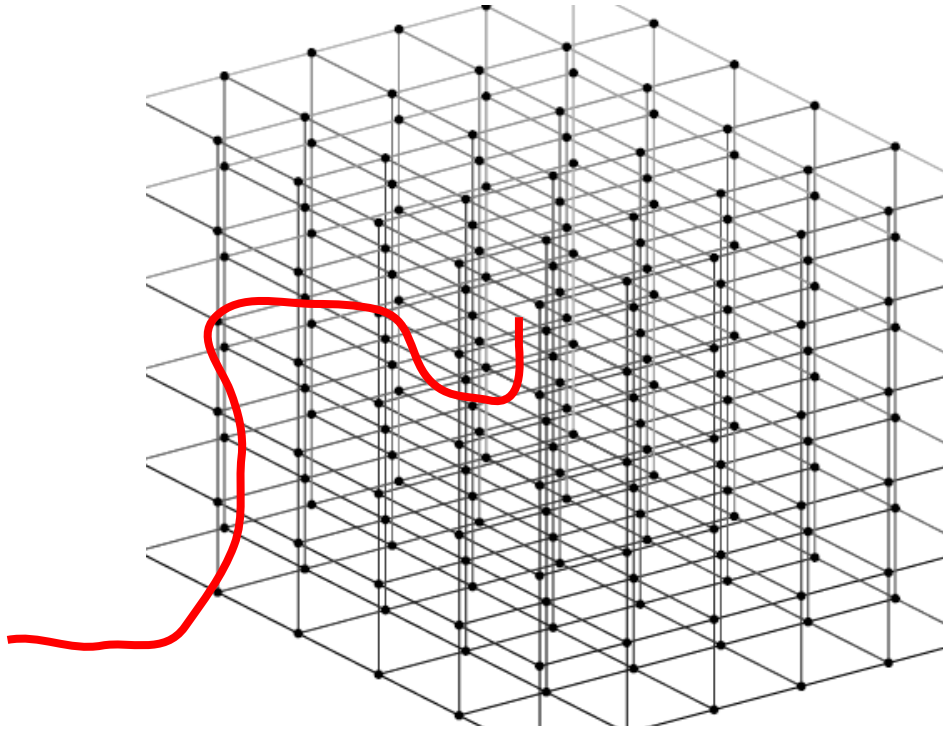
Tree rE, rB, E, B

Example

```
rE[0] = vector<double> x  
rE[1] = vector<double> y  
rE[2] = vector<double> z
```

```
E = Ex:Ey:Ez  
vector<vector<vector<double>>>
```

Interpolator



Fast trilinear
interpolation algorithm

$$\vec{r}_{j+1} = \vec{r}_j + \vec{F}_{(j-1) \rightarrow j} \Delta t$$

Config file

```
# Exteral fields (true if filename is present, 0 otherwise ), WhichF E/B: select
E or B or EB
ExtE      e_trap_light.root
ExtB      e_trap_light.root
WhichF    EB

# Eletric Field [V/m]
Ex        0
Ey        0
Ez        0

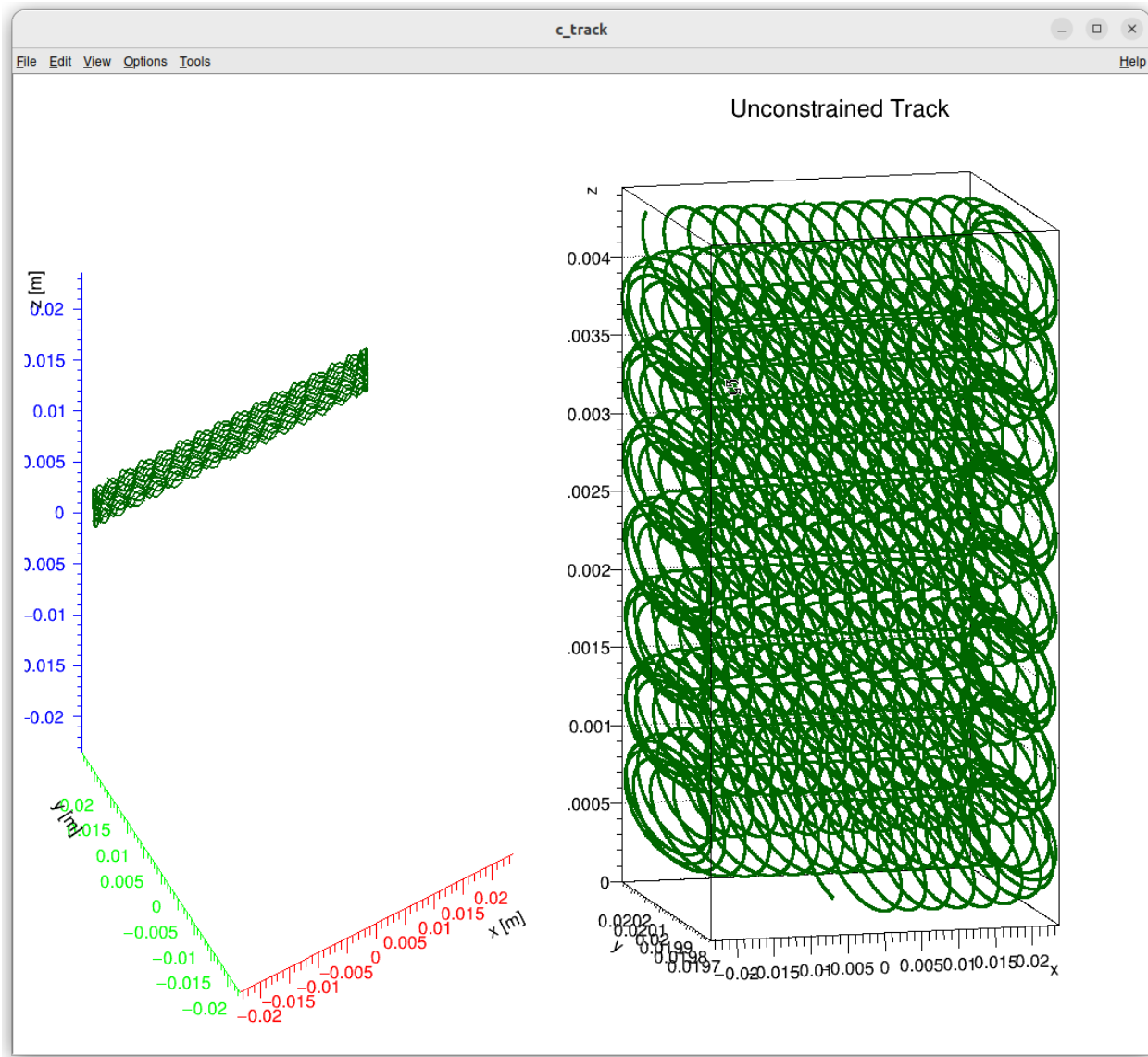
# Magnetic Field [T]
Bx        0
By        0
Bz        0
```

Options

Analytical only

Analytical + Simulated

B and E analytical and/or
simulated in all
combination



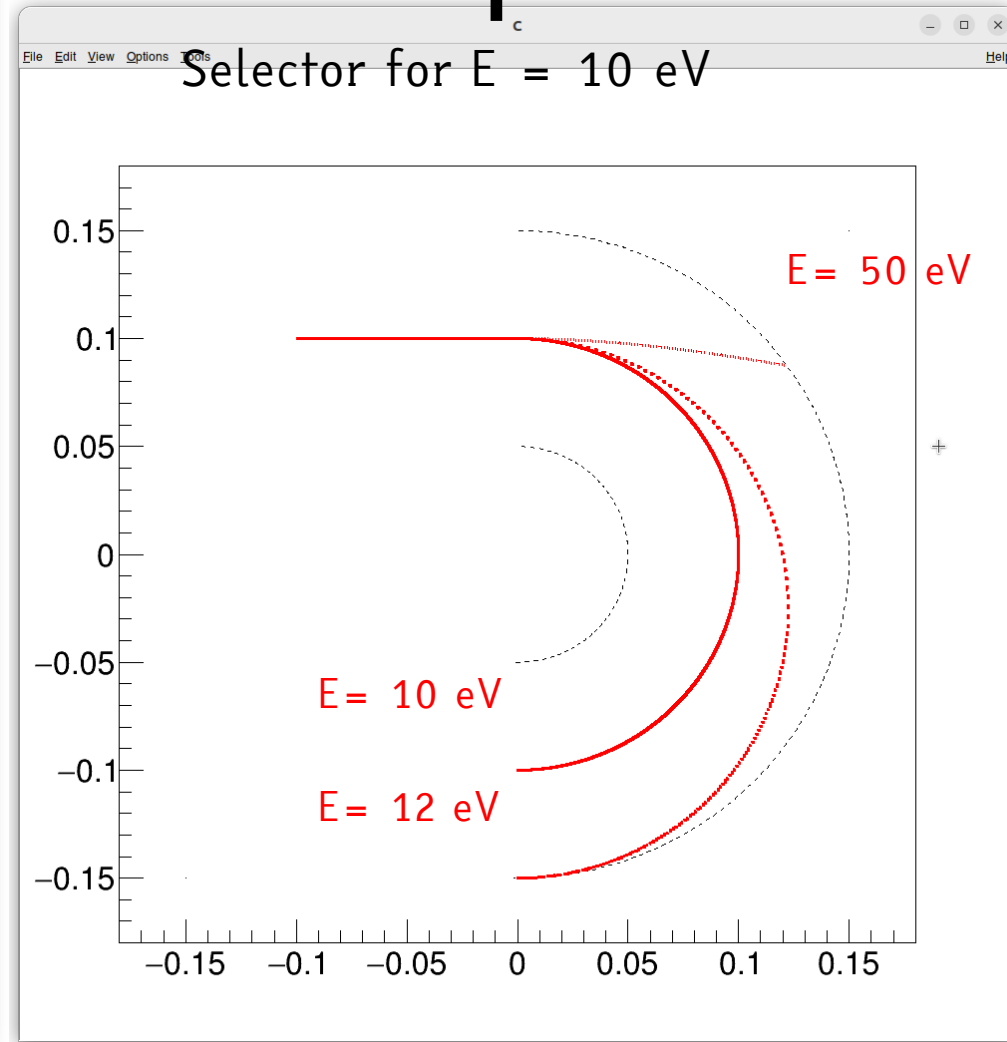
Simulation
of
the RF trap
(ext_fields.cfg)

Hemispheres

```
hemispheres.cfg
~/Desktop/Sensitivity/GitHub/lorentz4/cfg
Open Save

1 # Initial dynamics: energy (eV) and polar angles [deg]
2 E0      12
3 theta0  90
4 phi0    0
5
6 # Initial position [m]
7 x0      -0.1
8 y0      0
9 z0      0.1
10
11 # External fields (true if filename is present, 0 otherwise)
12 ExtF    0
13
14 # Electric Field [V/m]
15 Ex      26.67*0.05*0.15/(0.15-0.05)*x/pow(x*x+y*y+z*z,1.5)*(x>=0)+0*(x<0)
16 Ey      26.67*0.05*0.15/(0.15-0.05)*y/pow(x*x+y*y+z*z,1.5)*(x>=0)+0*(x<0)
17 Ez      26.67*0.05*0.15/(0.15-0.05)*z/pow(x*x+y*y+z*z,1.5)*(x>=0)+0*(x<0)
18
19 # Magnetic Field [T]
20 Bx      0
21 By      0
22 Bz      0
23
24 # Down Sampling: max number of points in the track
25 Np      100000
26
27 # Stop conditions on trajectory (x, y, z)
28 StopIf (sqrt(x*x+y*y+z*z)<0.05)+(sqrt(x*x+y*y+z*z)>0.15)+(z<0)*(x<0)
29
30 # Adaptive step: 0 = disabled; 1e-8 (or better) precision otherwise
31 Prcs    0
32
33 # Verbose and plots: 0 = disabled; 1 = enabled
34 Verb    1
```

Plain Text Tab Width: 8 Ln 34, Col 10 INS



lorentz4



ancillary



rootify

simeb

SimEB (beta)

E and B fields
generator

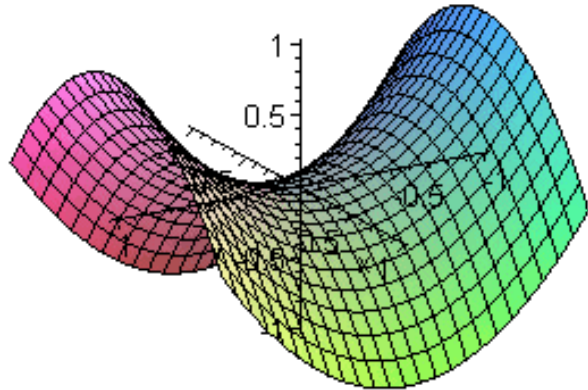
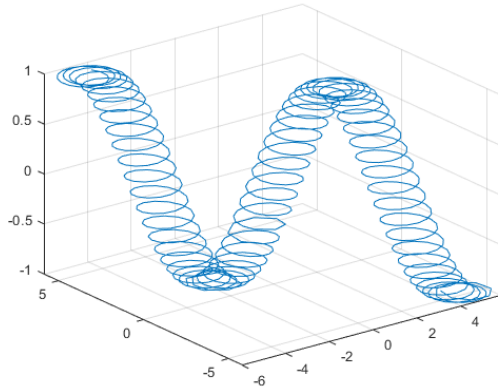
Descriptive geometry



B fields (Laplace)

Wires as parametric curves

$$\gamma(t) = (x(t), y(t), z(t))$$

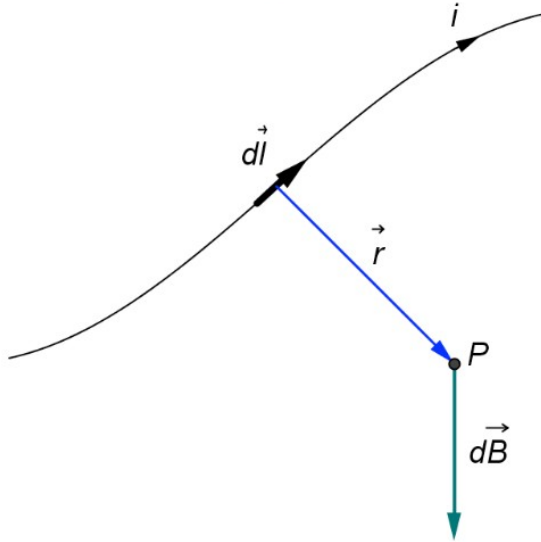


E fields (BEM)

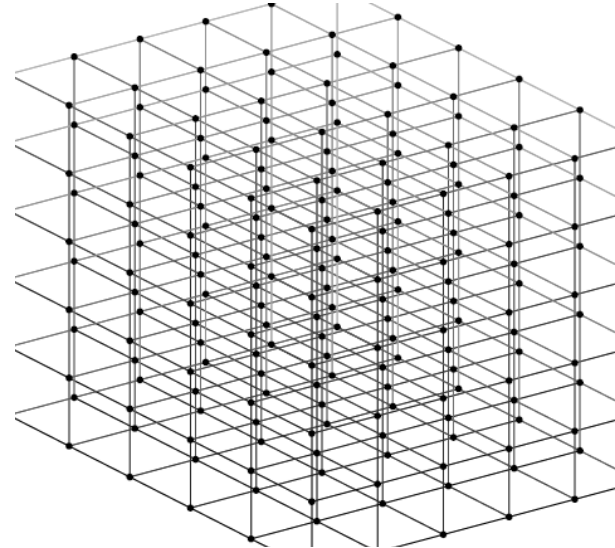
Equipotential conductive surfaces

$$\phi(u, v) = (x(u, v), y(u, v), z(u, v))$$

Laplace formula



$$\vec{B} = \frac{\mu_0}{4\pi} i \int_l \frac{d\vec{l} \times \vec{r}}{r^3}$$



$$\gamma(t) = (x(t), y(t), z(t))$$

Boundary Element Method

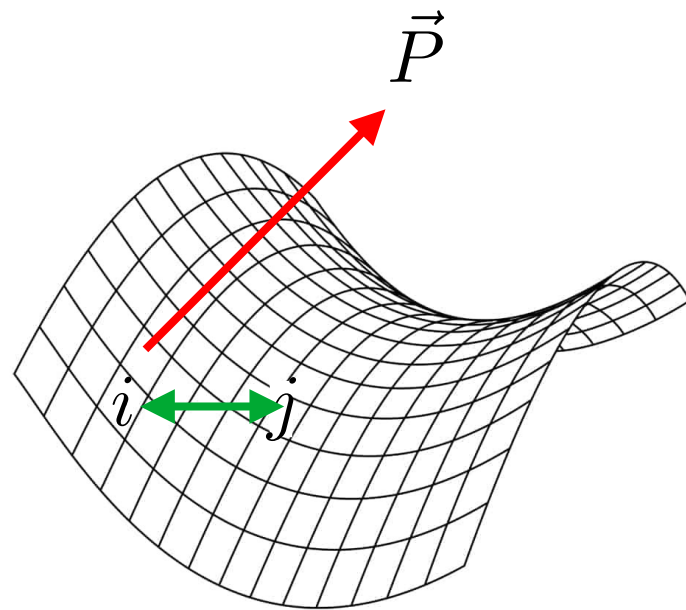
$$\nabla^2 V = \sigma$$

$$V_{\Sigma} = V_0$$

P.D.E. with
Dirichlet's
conditions

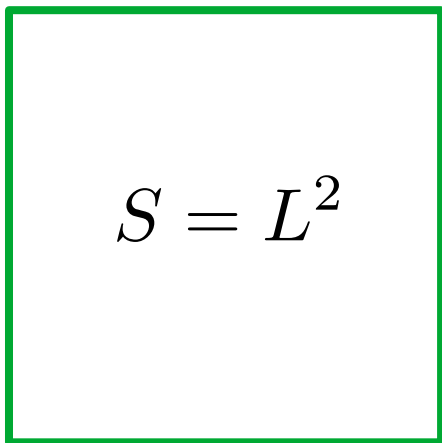
$$V_i \simeq \frac{1}{4\pi_0} \sum_j \frac{\sigma_j}{|\vec{r}_i - \vec{r}_j|} \Delta S_j = G_{ij} \sigma_j$$

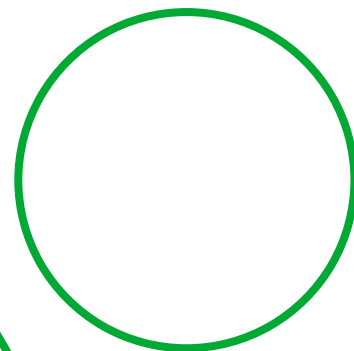
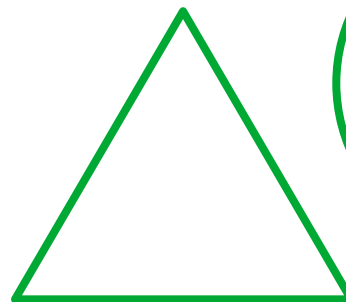
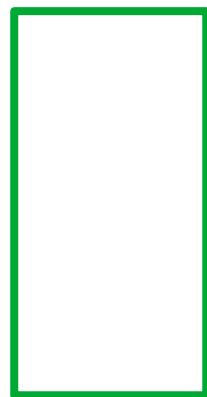
$$V(\vec{P}) = G_{pj} \sigma_j^*$$



Self interaction

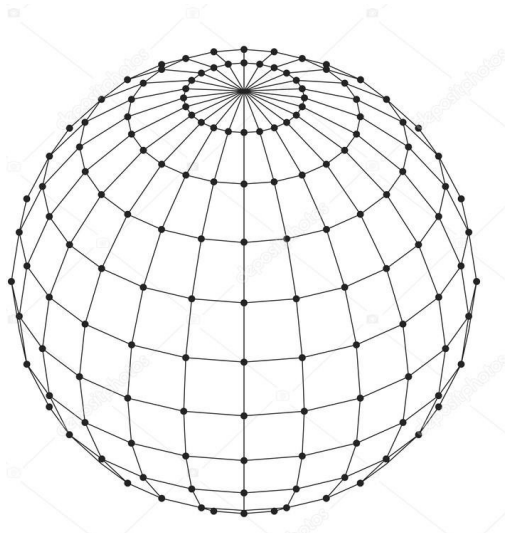
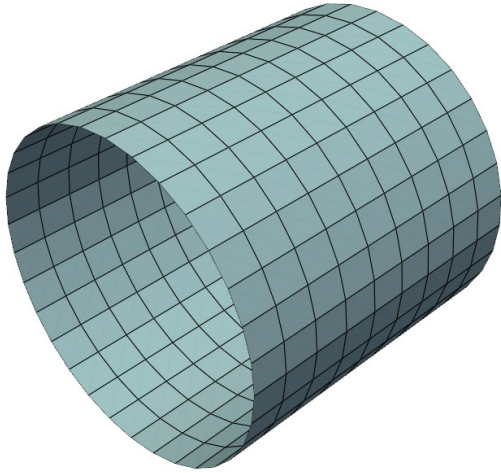
$$\int_L dx \int_L dy \frac{1}{\sqrt{x^2 + y^2}} \simeq \frac{3.52}{L}$$


$$S = L^2$$



$$\iint dS \frac{1}{\sqrt{x^2 + y^2}} \simeq \frac{3.52}{\sqrt{S}}$$

Meshing

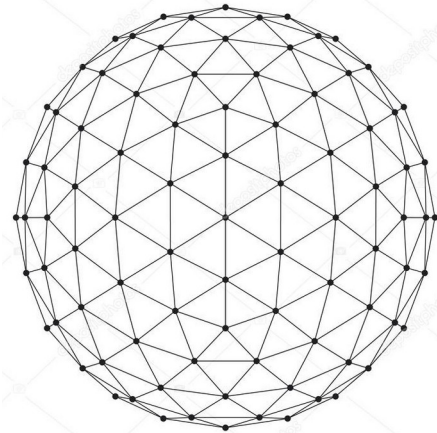


$$(u, v) \rightarrow \vec{\phi}(u, v)$$

SimEB meshing

Matrix inversion

- Gauss-Seidel (NO spheres)
- Lower-upper (LU) decomposition



```
template.C
~/Desktop/Sensitivity/GitHub/lorentz4/ancillary/simeb

Open  Save  -  □  ×

25 string msolver = "gauss"; // or LU
26
27 vector<double> x = Volume(-1, 1, 30); // define range here
28 vector<double> y = Volume(-1, 1, 30);
29 vector<double> z = Volume(-1, 1, 30);
30
31 TCanvas *c0 = new TCanvas("c0", "c0", 800, 800);
32 TGraph2D* g0 = new TGraph2D();
33 g0->SetTitle("Source shapes; x [m]; y [m]; z [m]");
34
35 g0->SetPoint(0, x.front(), y.front(), z.front()); // define plot range
36 g0->SetPoint(1, x.back(), y.back(), z.back());
37 g0->Draw("P");
38
39 vector<Surface> S; // surface vector
40 vector<double> Vi; // Potential vector
41 vector<vector<Point3D>> L; // line element
42 vector<double> Bi; // unit magnetic field [T x m]
43 TGraph2D* gE = new TGraph2D(); // show sources
44 TGraph2D* gB = new TGraph2D(); // show sources
45
46 // Add new E source adding the potential and the parametric surface phi(u, y)
47 AddEsource(S, Vi, 100, "x[0]", "x[1]", "0.25", -0.5, 0.5, 50, -0.5, 0.5, 50, gE);
48 AddEsource(S, Vi, -100, "x[0]", "x[1]", "-0.25", -0.5, 0.5, 50, -0.5, 0.5, 50, gE);
49
50
51 // Add new B source adding the magnetic field unit and the parametric curve gamma(t)
52 AddBsource(L, Bi, 1, "0.5*cos(x[0])", "0.5*sin(x[0])", "0.20", 0, 2*TMath::Pi(), 200, gB);
53 AddBsource(L, Bi, 1, "0.5*cos(x[0])", "0.5*sin(x[0])", "-0.20", 0, 2*TMath::Pi(), 200, gB);
54
55 cout << "All E and B sourced added" << endl;
56
```

template.C

EXAMPLES:

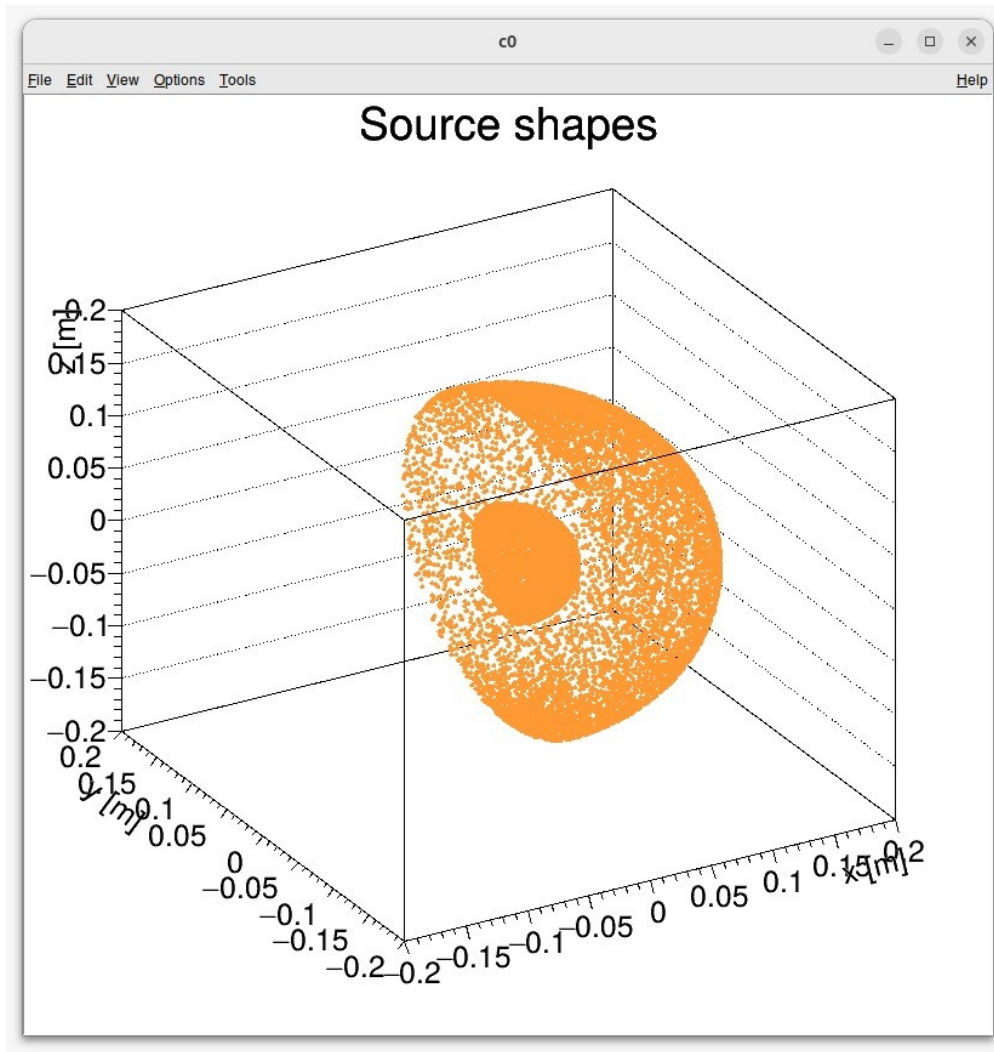
ex0_square.C
ex1_einzel.C
ex2_helmoltz.C
ex3_tokamak.C
ex4_EparB.C
ex5_hemispheres.C
ex6_tpc.C
ex7_trap.C

Compile & run

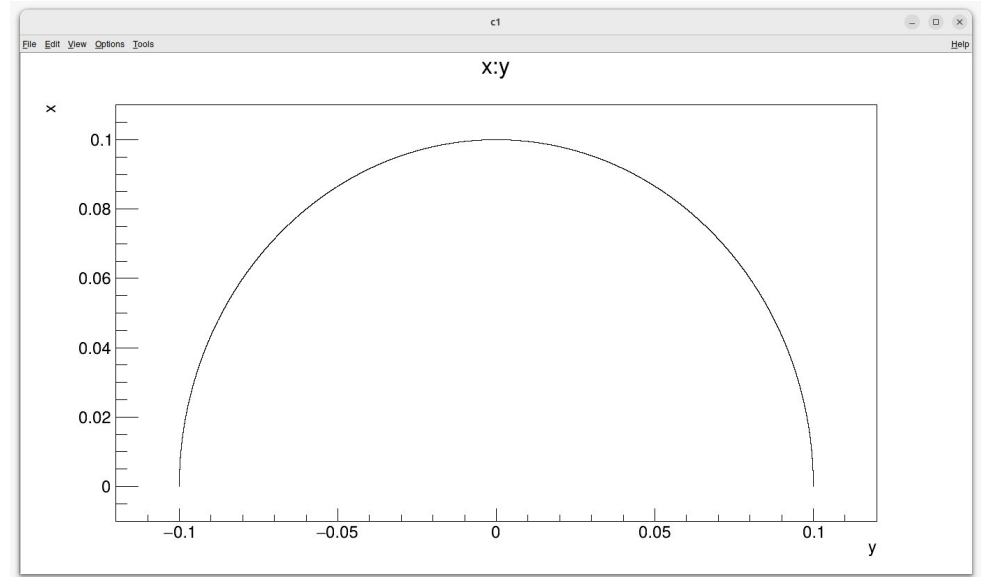
```
cp template.C mytemplate.C (edit)  
./compile.sh template.C  
./template myout.root
```

Generate myout.root in lorentz4 input
format (out.root otherwise)

```
./compile.sh drawfields.C (Display fields)  
./drawfields myout.root
```

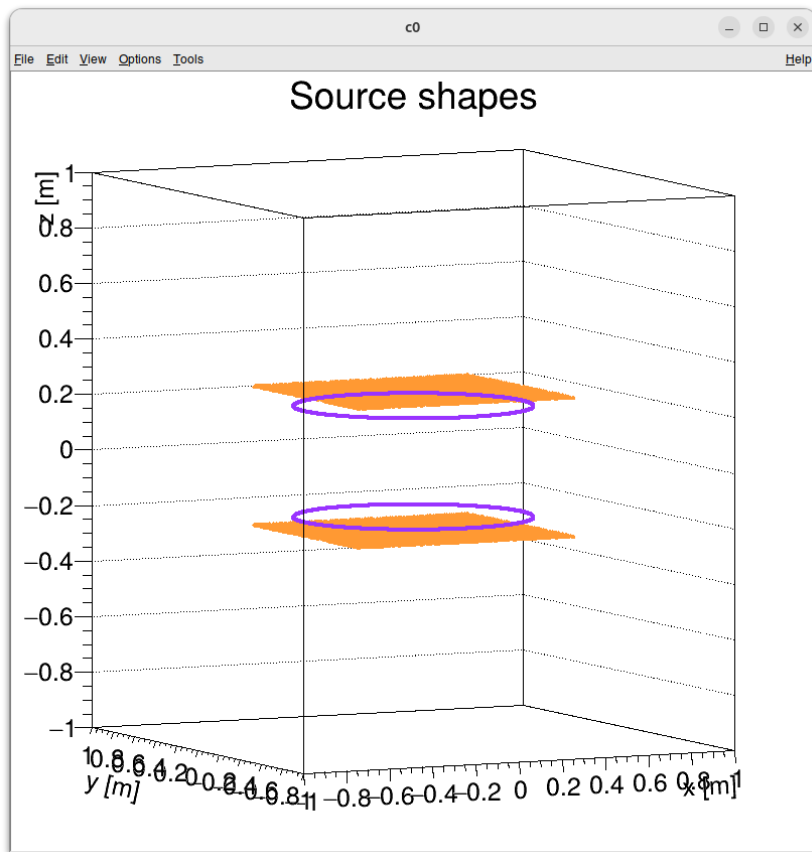


hemispheres

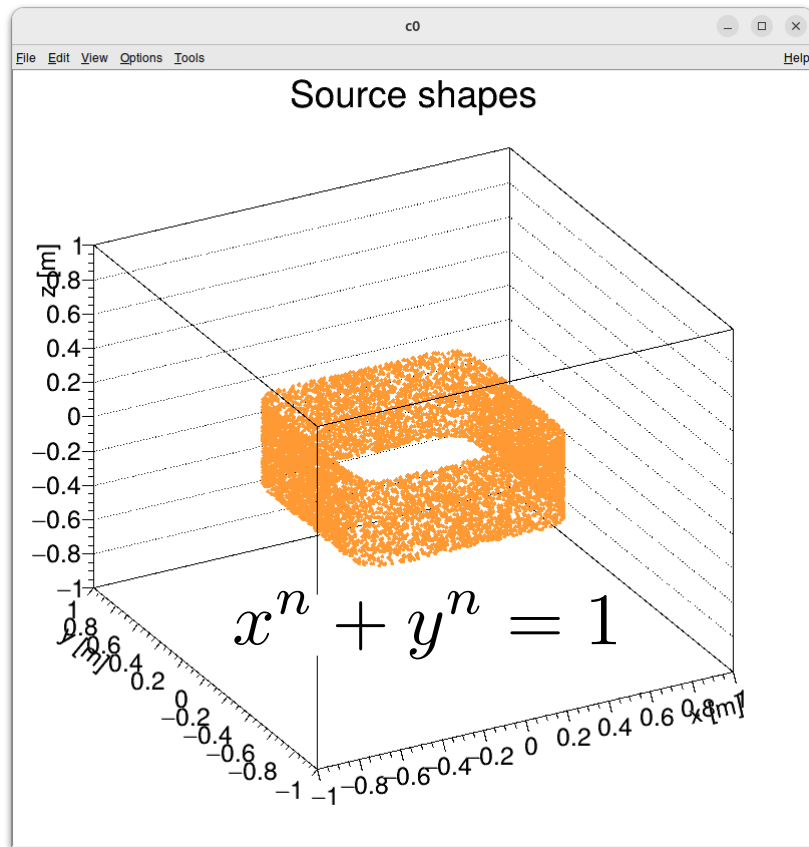


Track simulation
with **lorentz4**

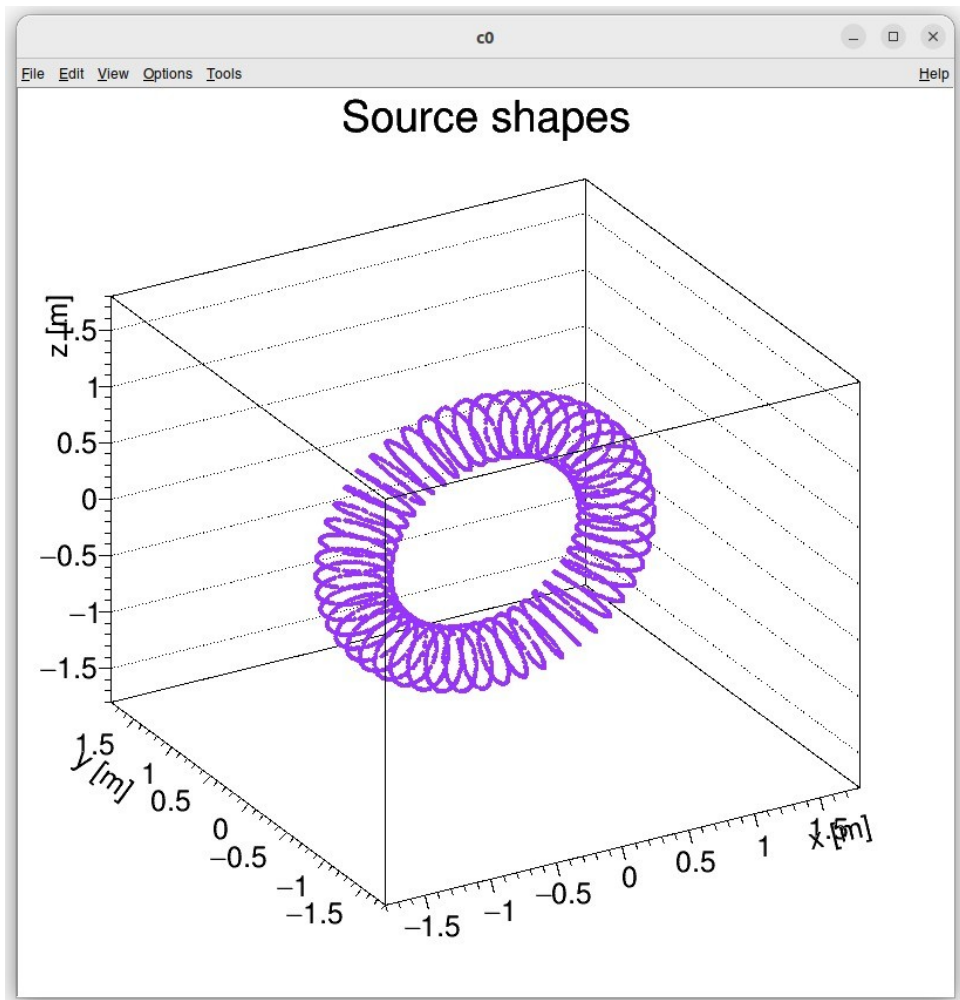
Examples



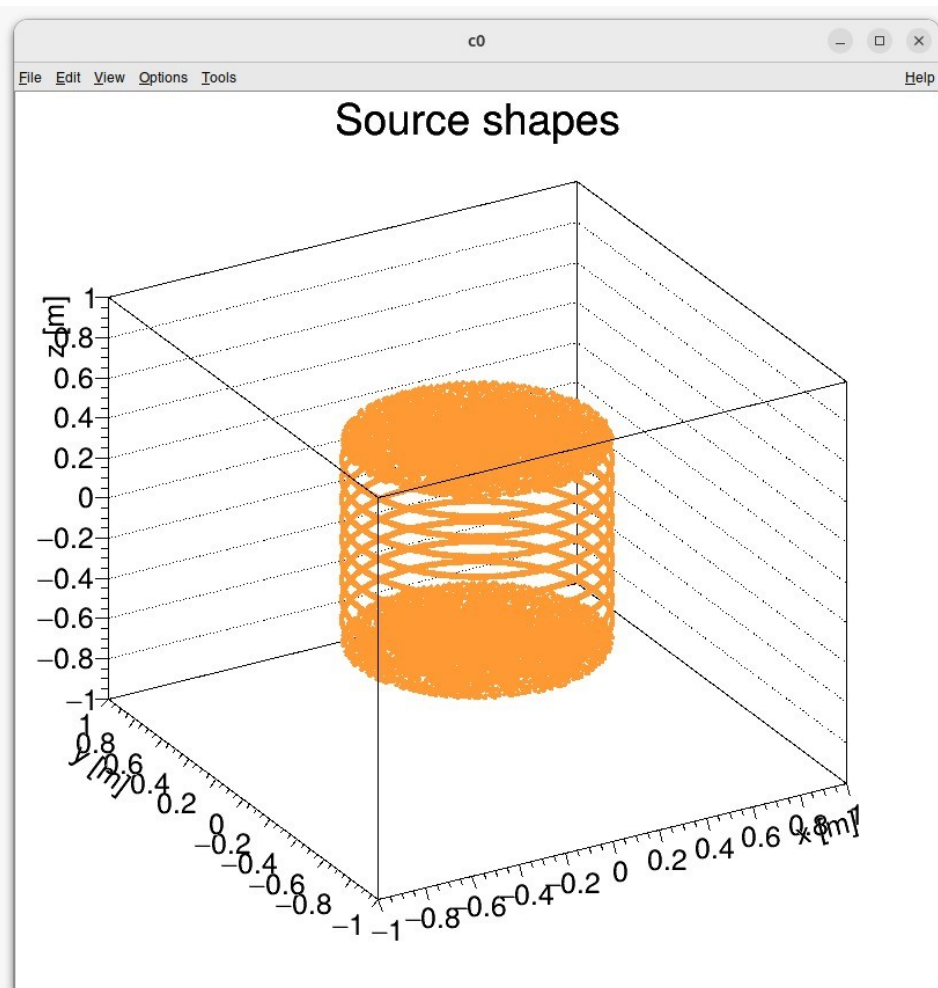
template



square (DarkSide)

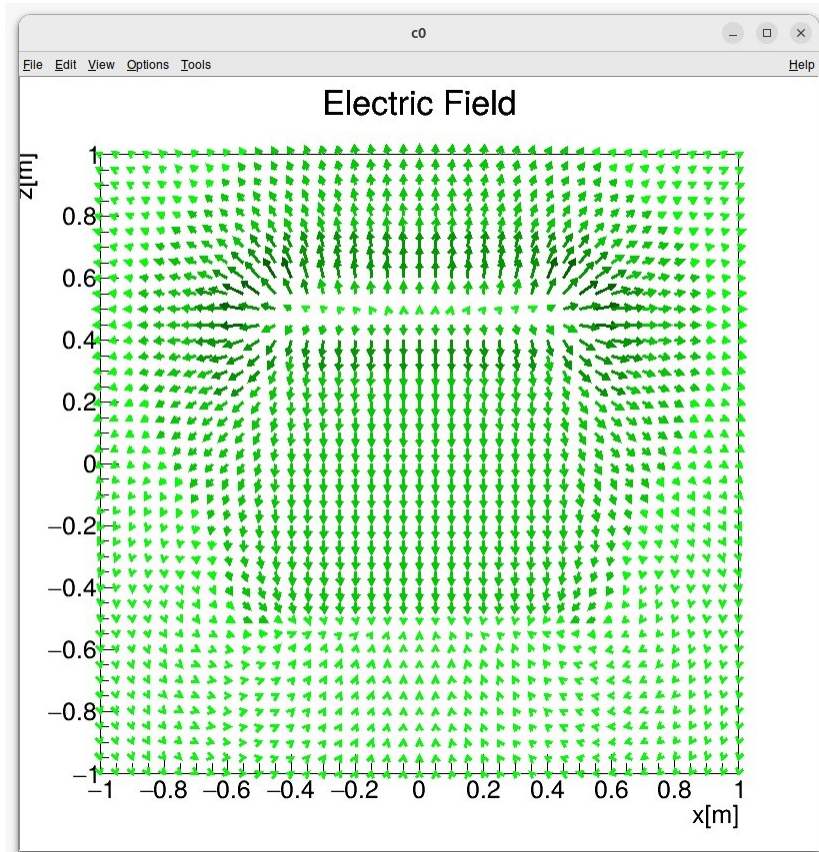


tokamak

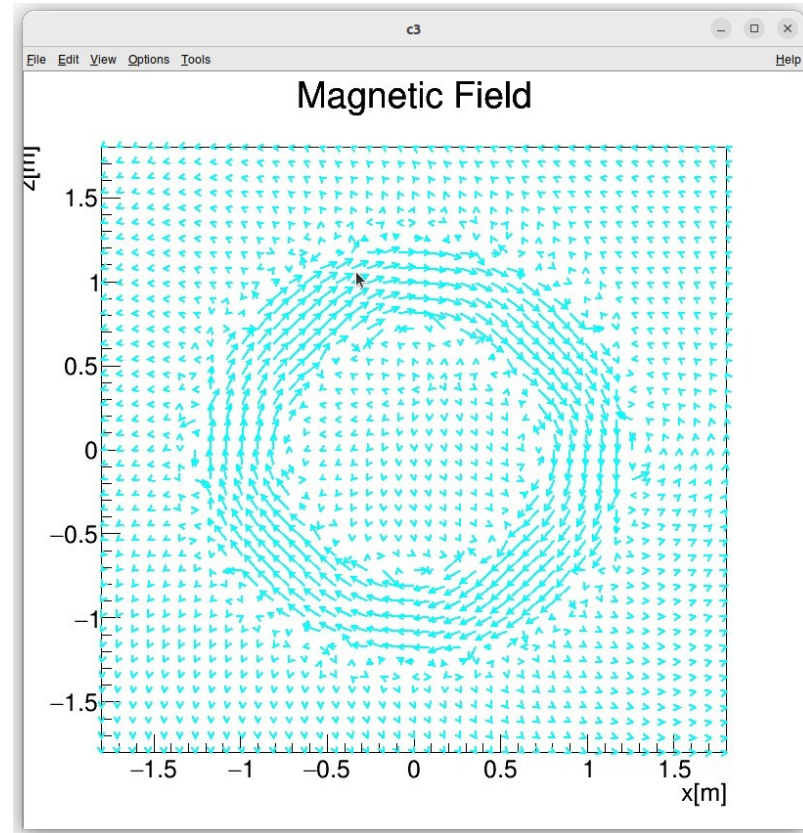


Tpc (XENONnT)

drawfields.C



tpc



tokamak

Outlooks

- Cleaning (C++ standard)
- Improvement of algorithms
- Documentation (maybe **arXiv**)

Any other suggestion is really welcome!



Thank you
for your attention