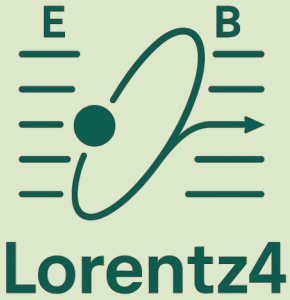


Lorentz4
(beta version)

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 geomtry
  (B from wires, Laplace)
  (E, from equipotential surfaces, BEM)

Repository

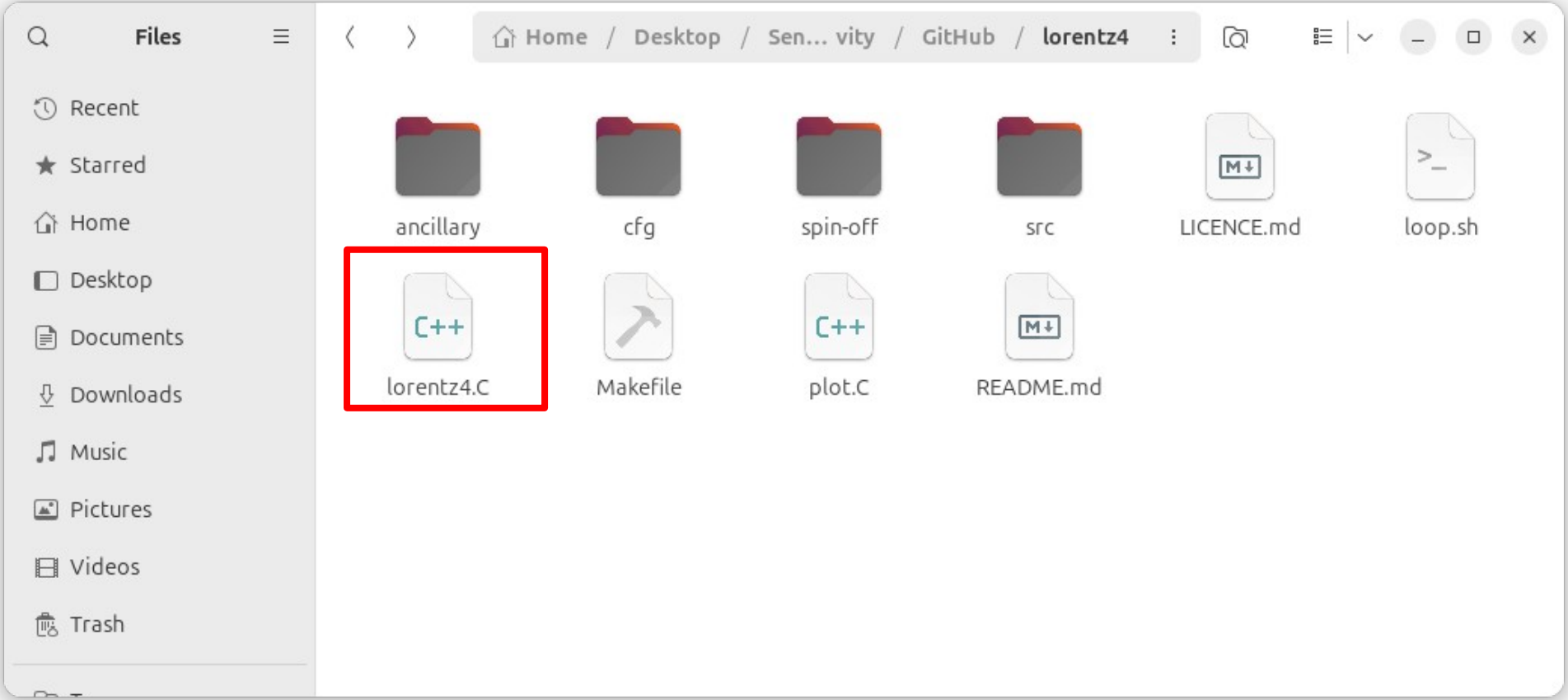
Prerequisite:

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

mailto:
nicola.rossi@lngs.infn.it

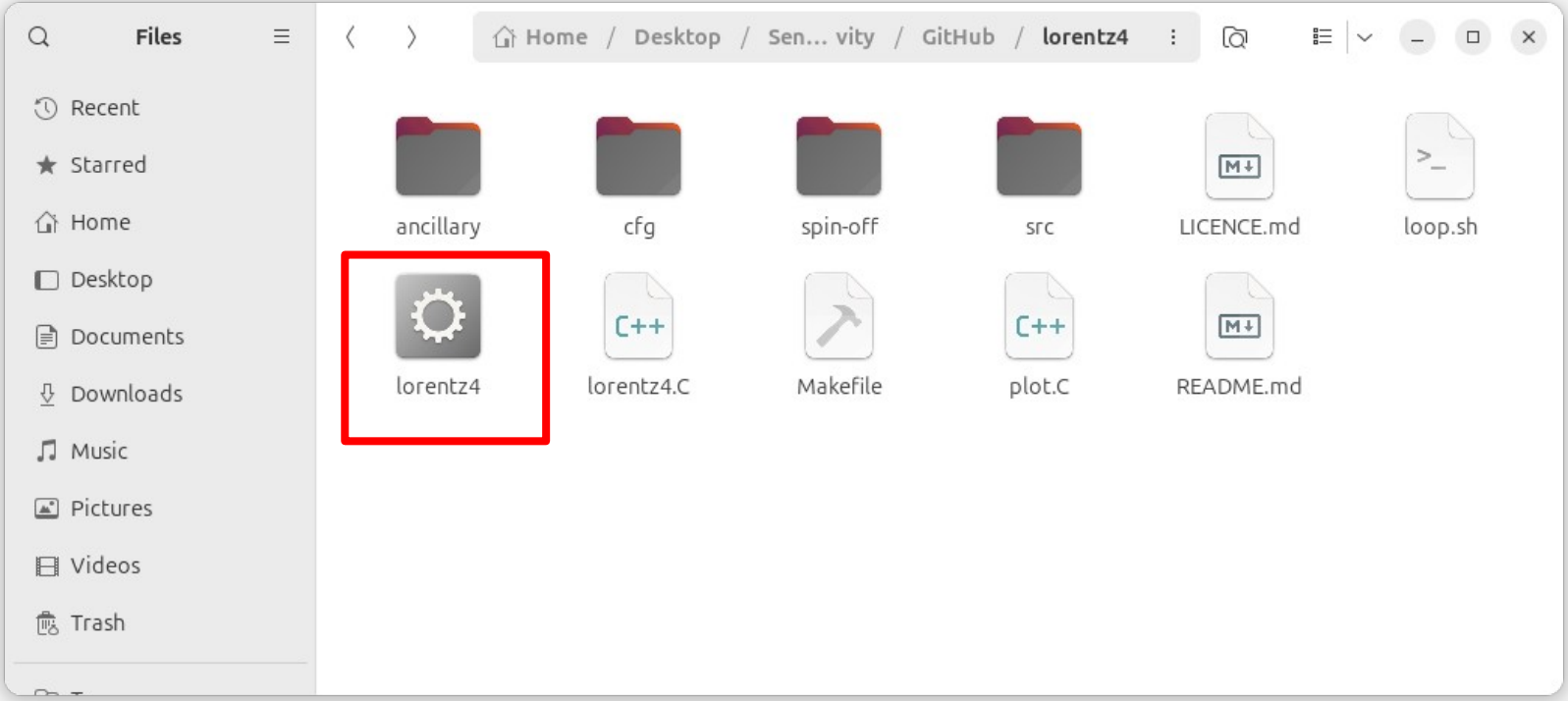
```
git clone git@github.com:gkrossi/lorentz4.git
```

Lorentz4 folder



How to compile

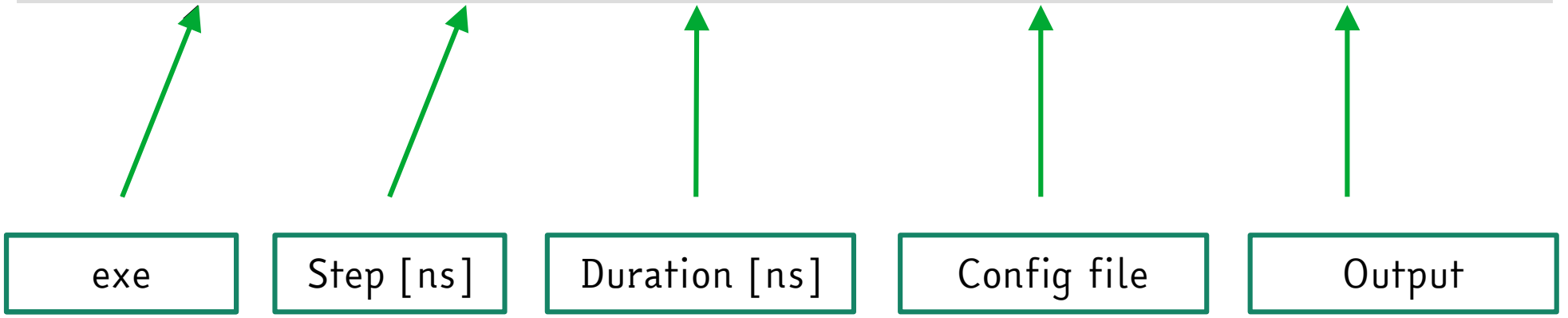
```
make
```



How to run

Optional

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



Relativistic formulas

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

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

Kinetic energy

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

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

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

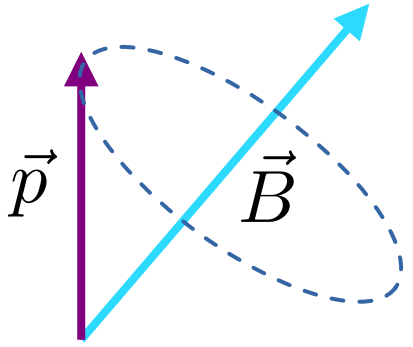
Potential energy

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

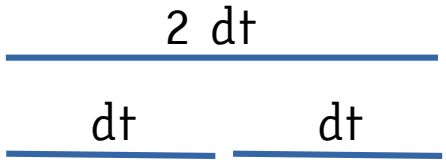


$$\frac{d\vec{p}}{dt} = \vec{F}_{\text{Lorentz}} + \vec{F}_{\text{Rad}}$$

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

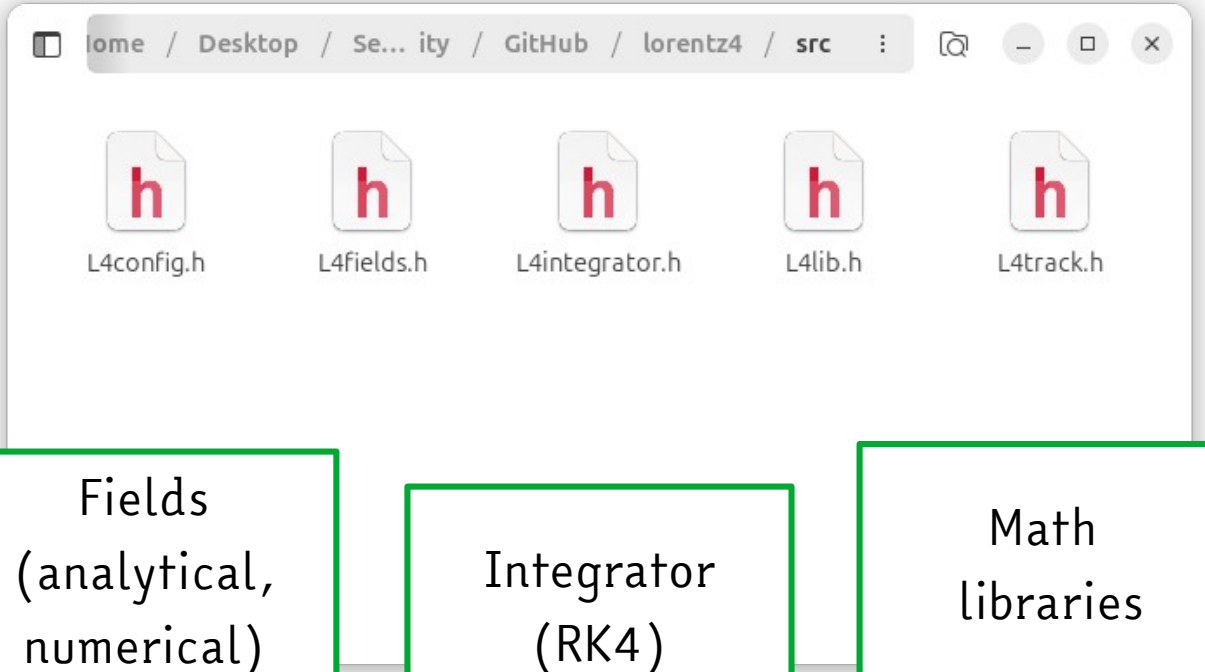


$$x(t+2dt) \approx x_1 + (2dt)^5\phi + \mathcal{O}(dt^6)$$
$$x(t+2dt) \approx x_2 + 2dt^5\phi + \mathcal{O}(dt^6)$$

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

$$\Delta \equiv x_2 - x_1$$

Source (`src`) folder



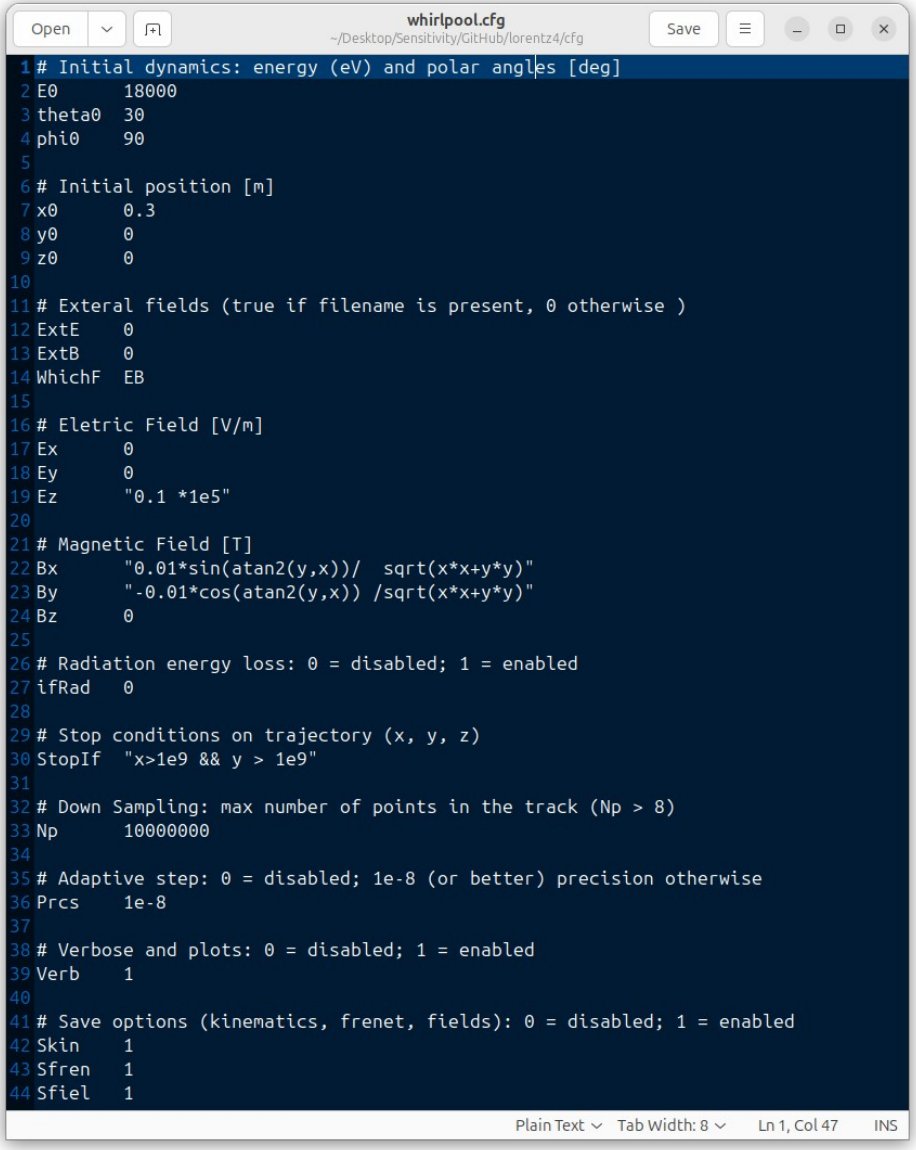
Configuration handling

Trajectory & observables

Fields (analytical, numerical)

Integrator (RK4)

Math libraries



New config file

Demos (cfg)

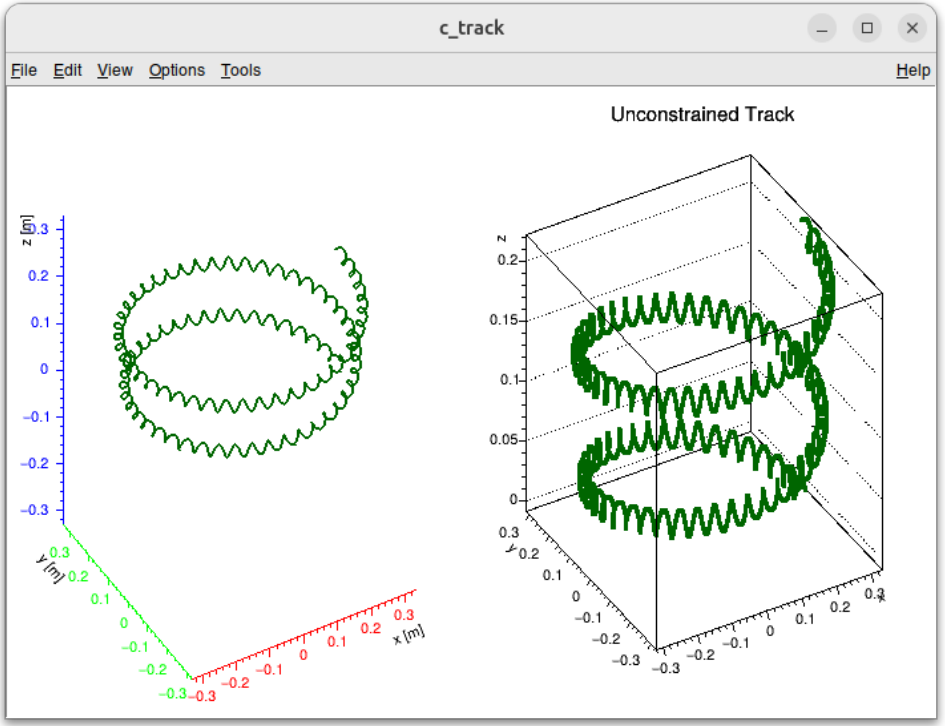
init.cfg

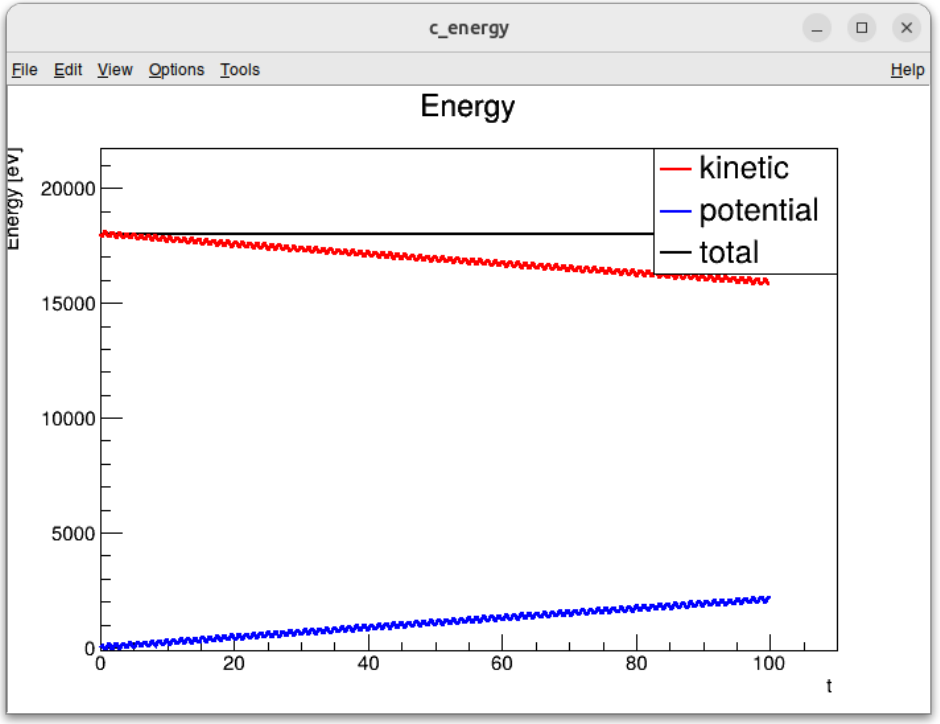
Bconst.cfg
hemispheres.cfg
nicola.cfg
tully.cfg
EparB.cfg
hemispheres_ext.cfg

ptolemy.cfg
v_selector.cfg
ext_fields.cfg

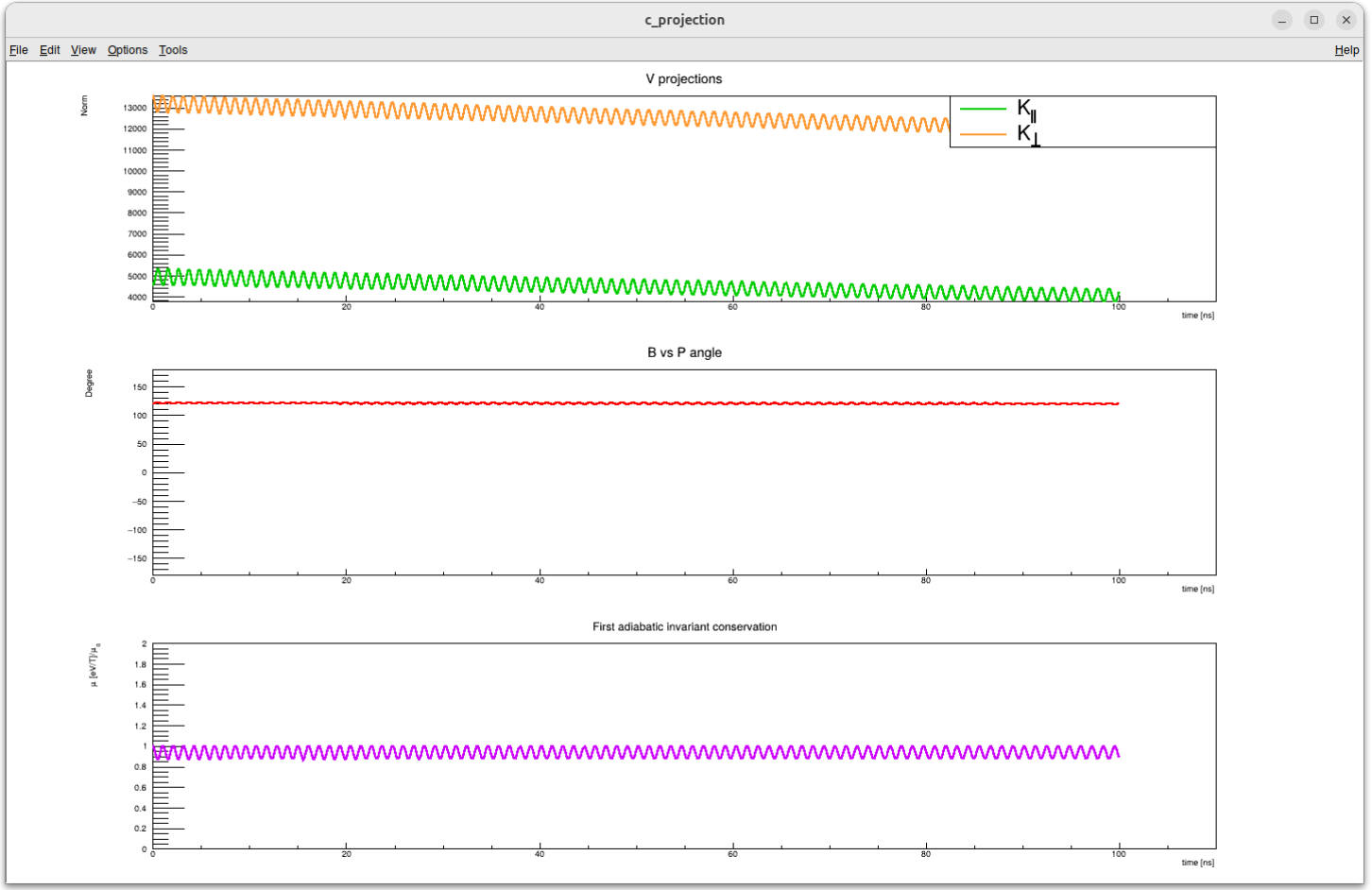
RFcalib.cfg
whirlpool.cfg
gauss.cfg
katrin.cfg
SimEB.cfg
wonyong.cfg

Trajectory





Energy

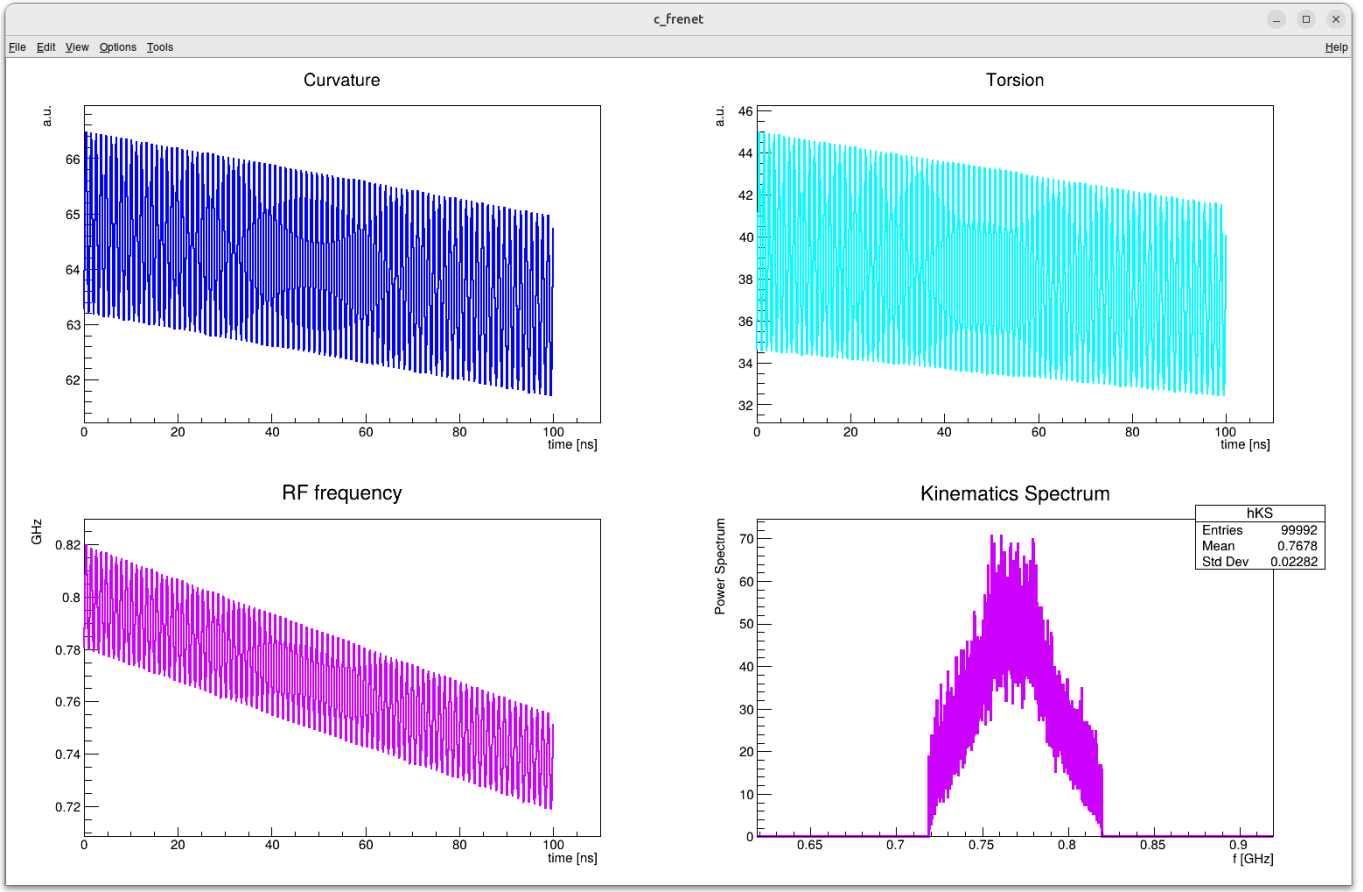


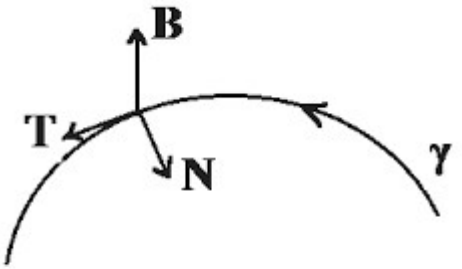
K_{par}

K_{ort}

ϑ

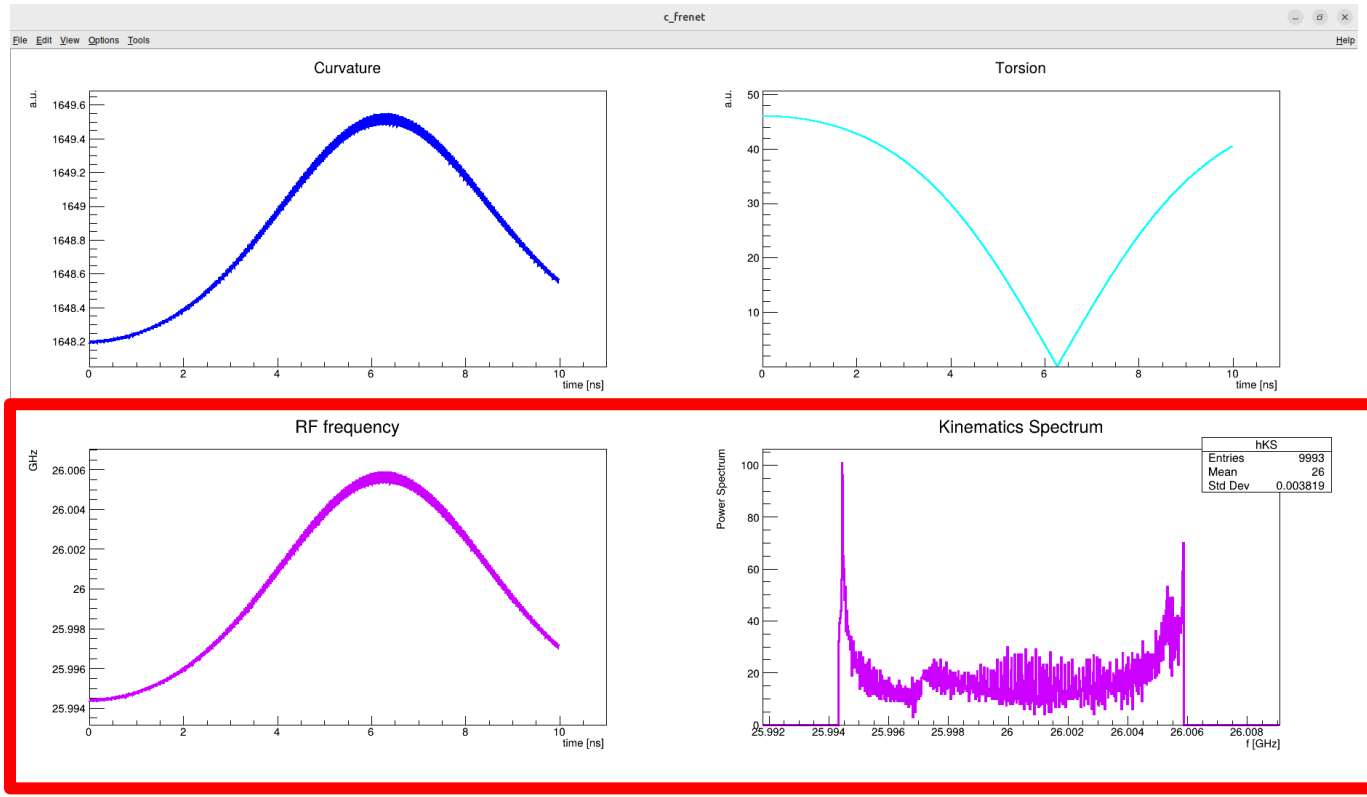
μ

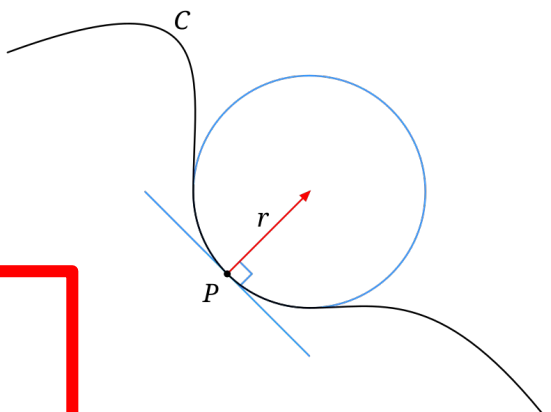




Frenet Params.

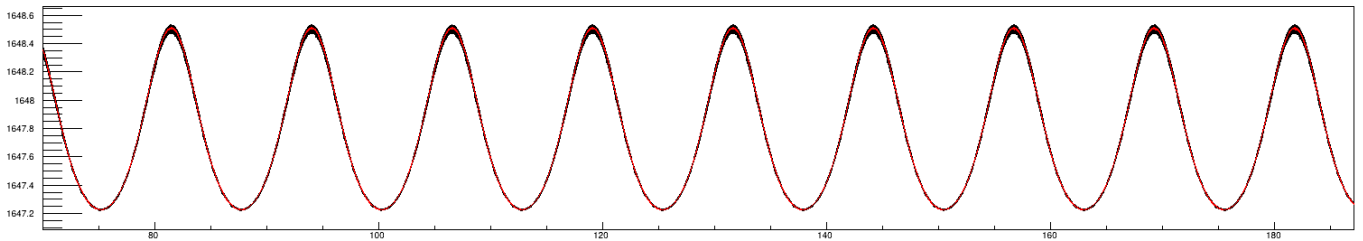
Kinematic spectrum



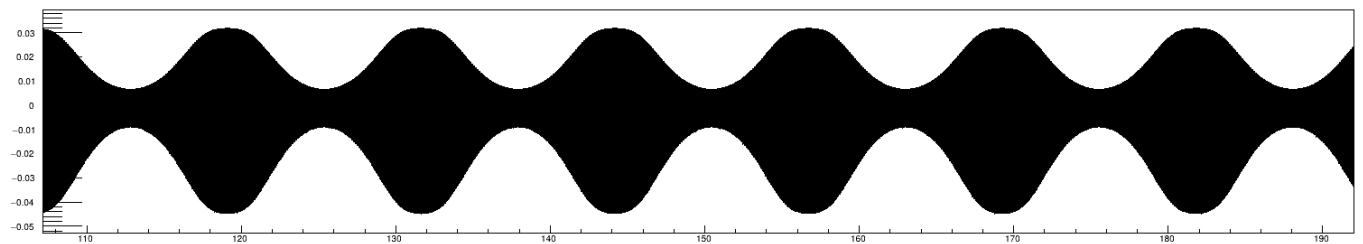


$$\omega_c = \frac{1}{\gamma} \frac{|\vec{r'} \times \vec{r''}|}{|\vec{r'}|^2}$$

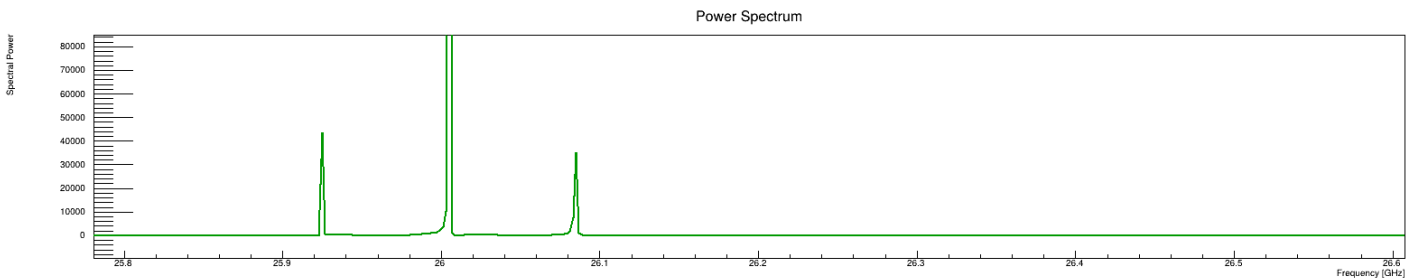
Curvature FFT



Curvature

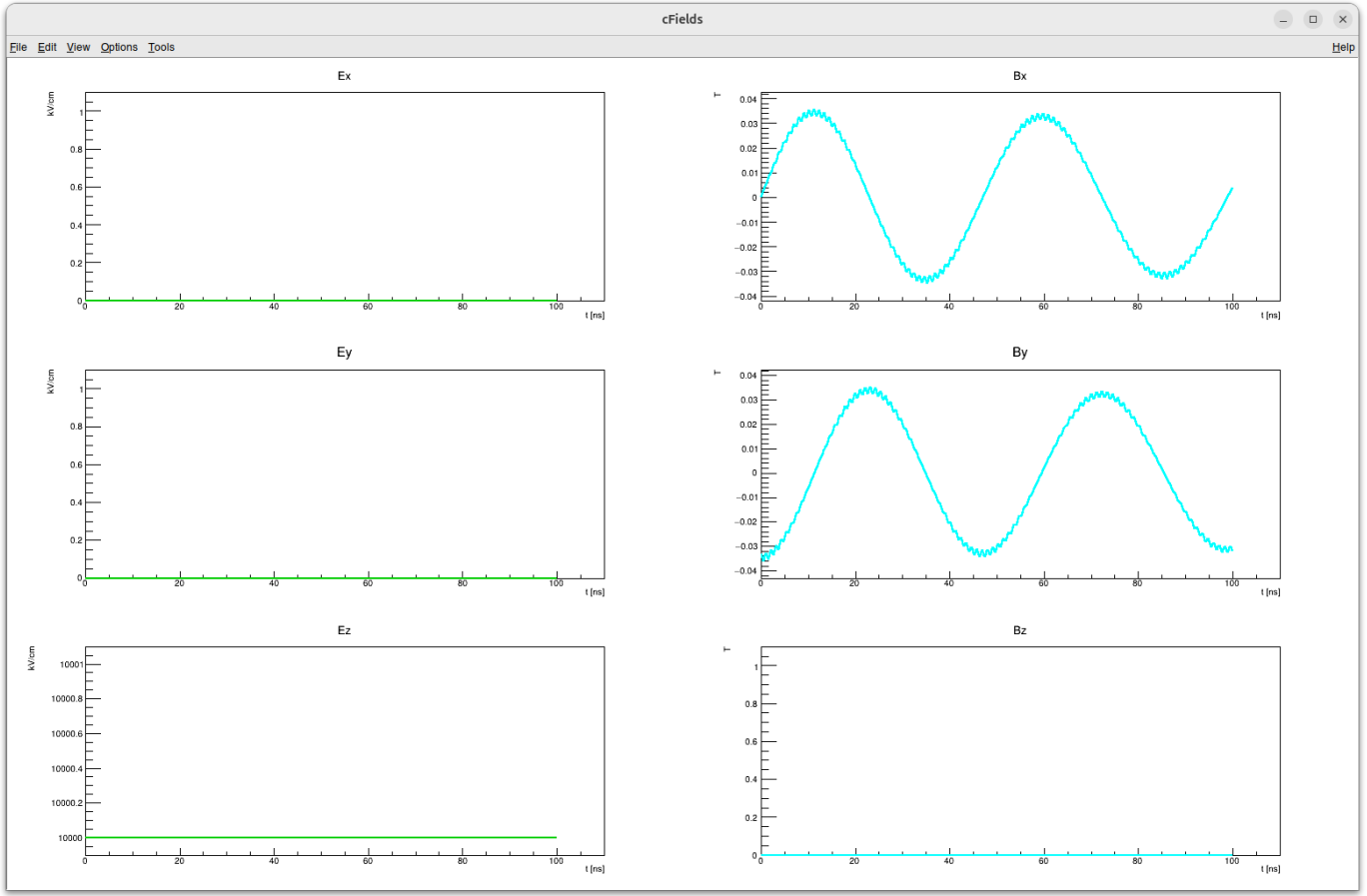


Residuals
After LOESS
Detrend



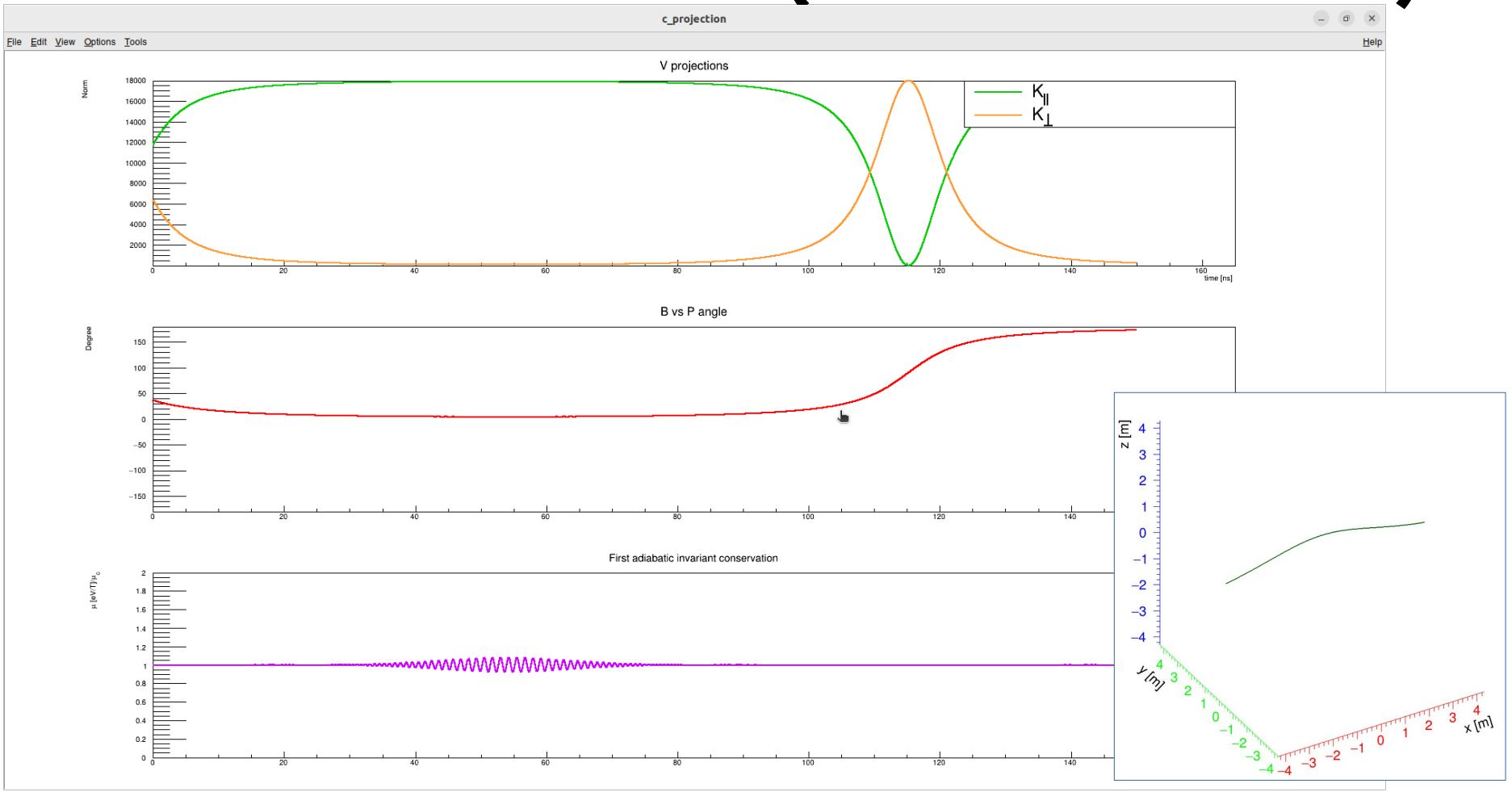
FFT
(Sidebands)

To be implemented

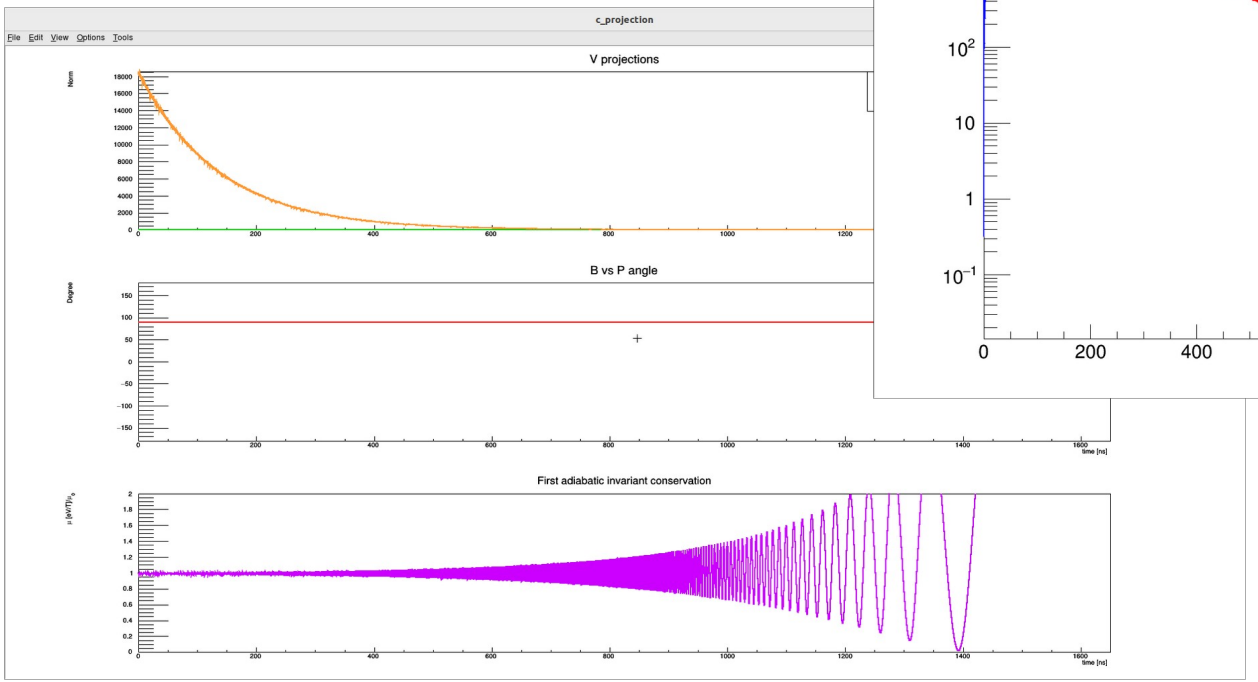


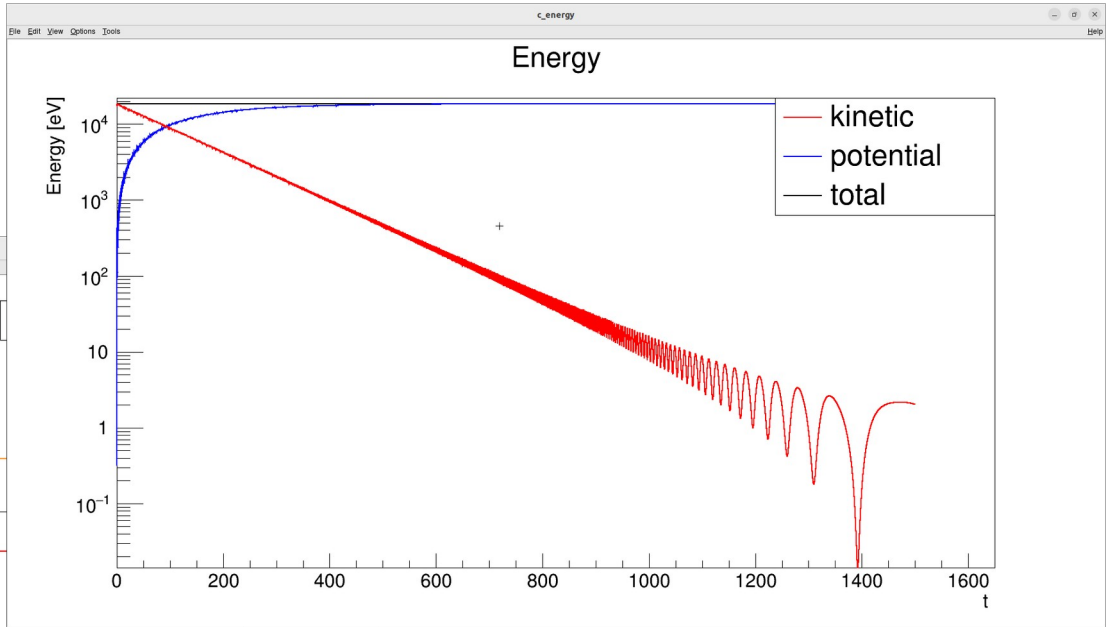
E & B fields on track

MAC-E filter (`katrin.cfg`)



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





Looping on many events

*loop.sh
~/Desktop/GIT/lorentz4

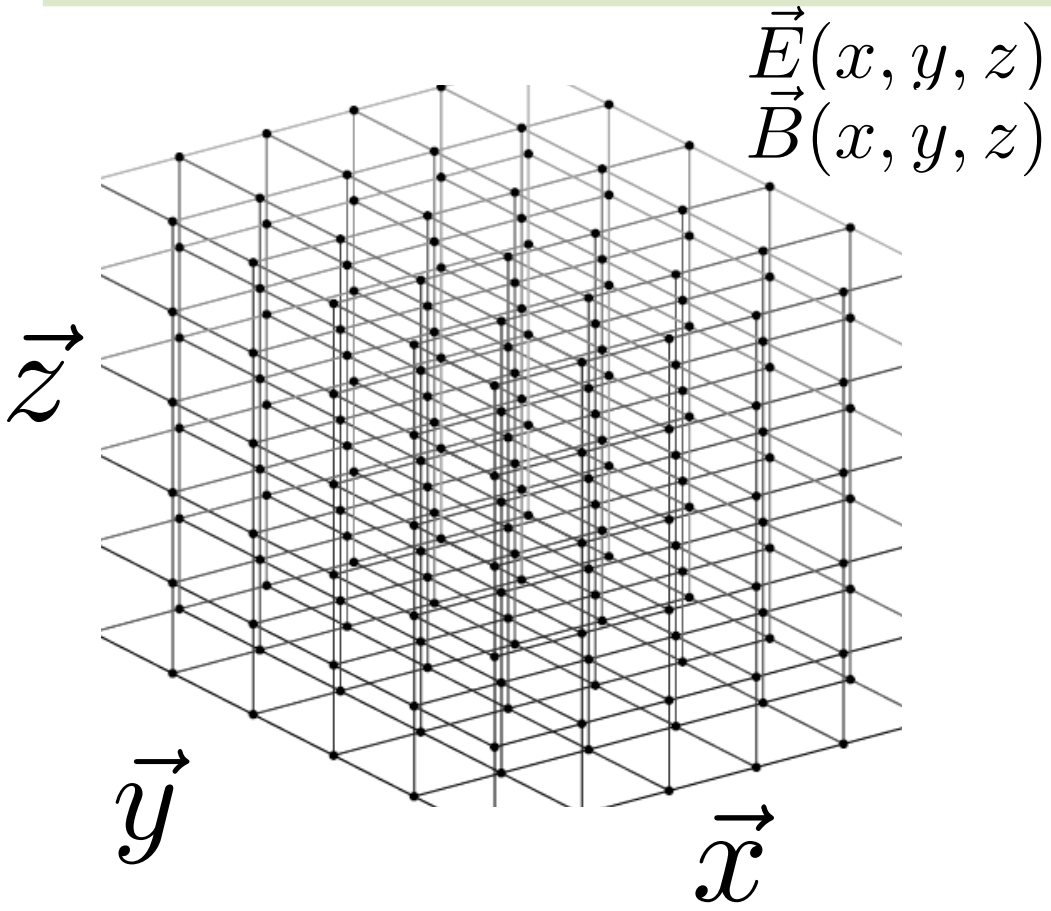
```
#!/bin/bash

n=0;
for energy in 10000 18500
   do
   echo "# Initial dynamics: energy (eV) and polar angles [deg]
   E0      $energy
   theta0  45
   phi0    0

   # Initial position [cm]
   x0      0.
   y0      0
   z0      0

   # Eletric Field [kV/cm]
   Ex      3.*x
   Ey      1.
   Ez      0

   # Magnetic Field [T]
   Bx      -1
   By      0
   Bz      0

   # Down Sampling: max number of points in the track
   Np      100000

   # Step doubling precision error estimation: Enable = 1, Disable = 0
   Prcs 0" > cfg/loop_$n.cfg

  ./lorentz4 0.0001 1 cfg/loop_$n.cfg out_$n.root
  ((n=n+1))
done
root "plot.C($n)"
```

`./loop.sh`
(parallel)

`root plot.C`
(TChain)

Add **qsub** for parallel computing

External simulated fields



Instructions in
ancillary/rootify

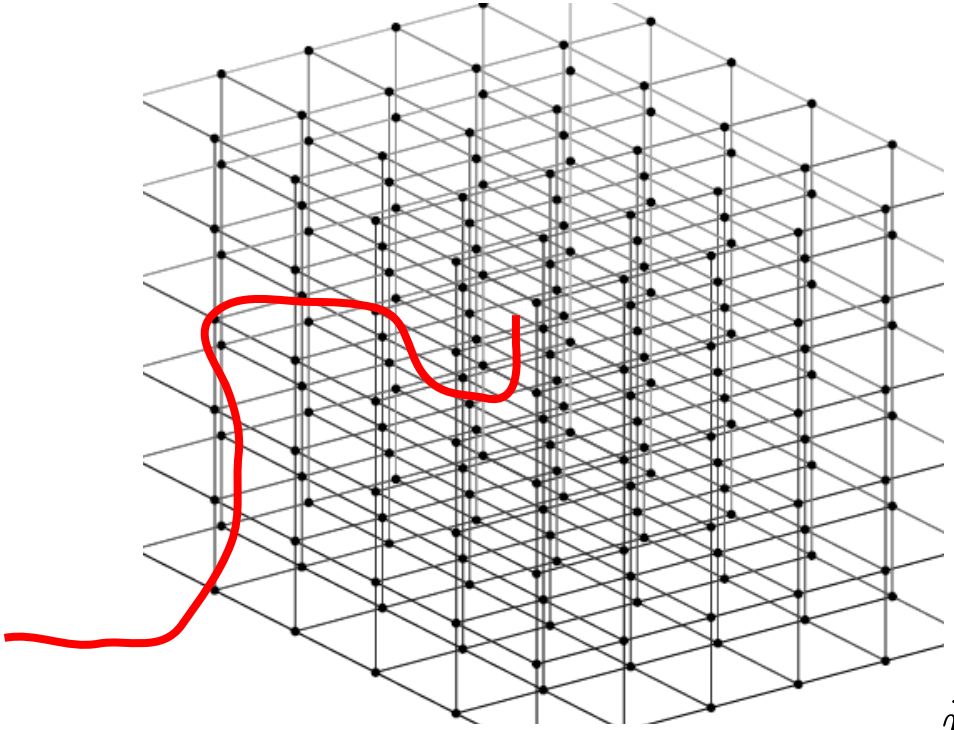
Rootfile

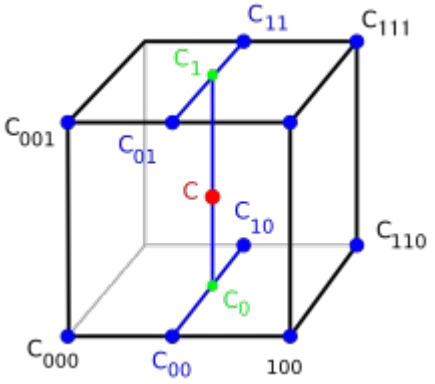
```
Tree rE, rB, E, B
```

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

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

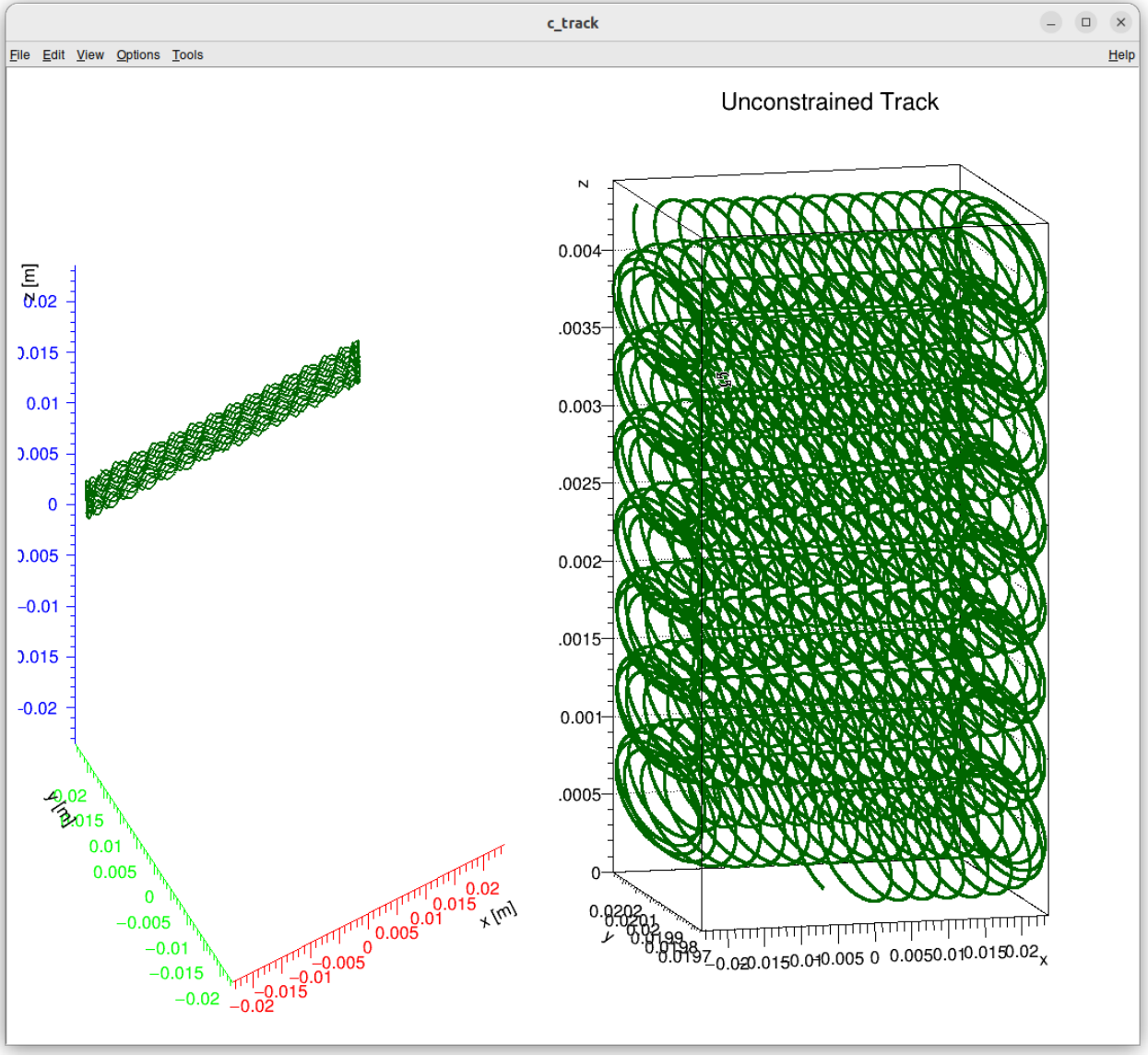
Config file

```
# 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

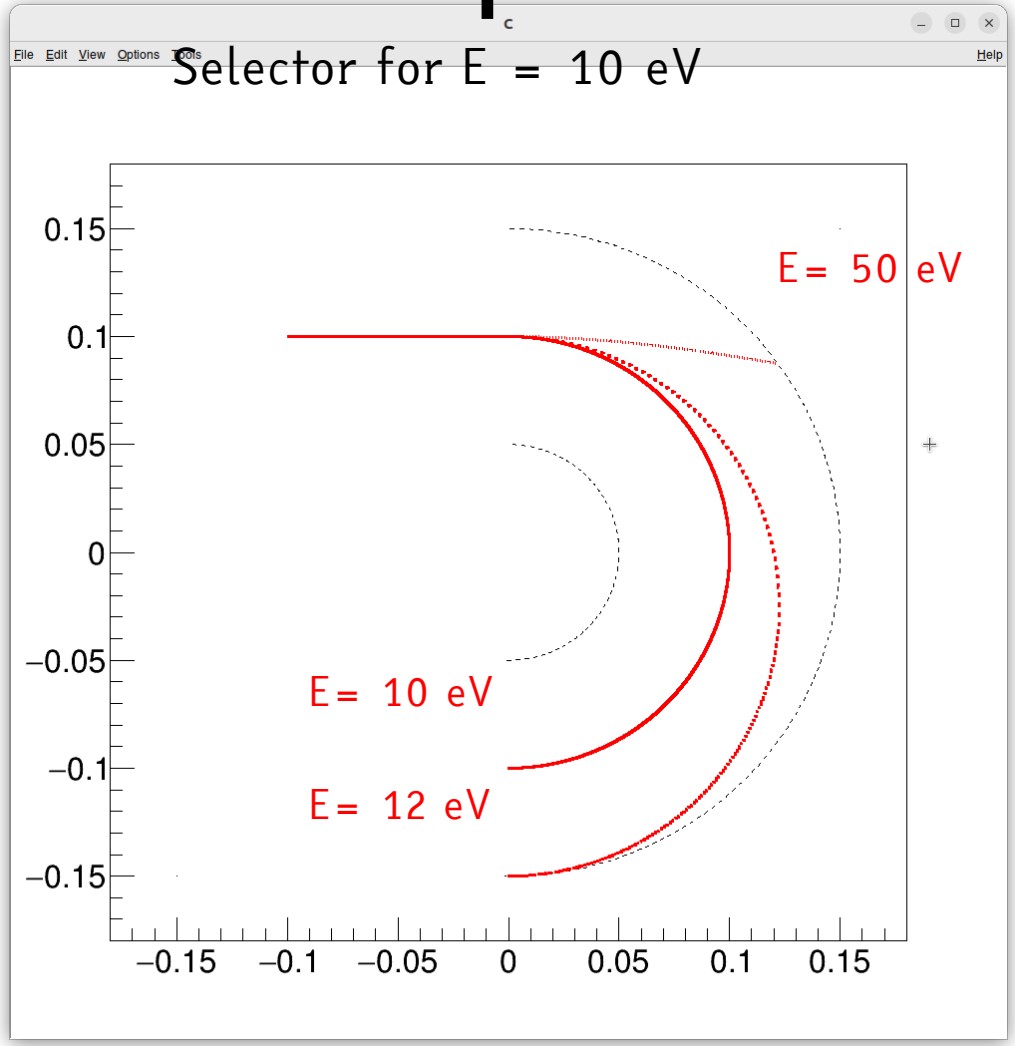
(ext_fields.cfg)

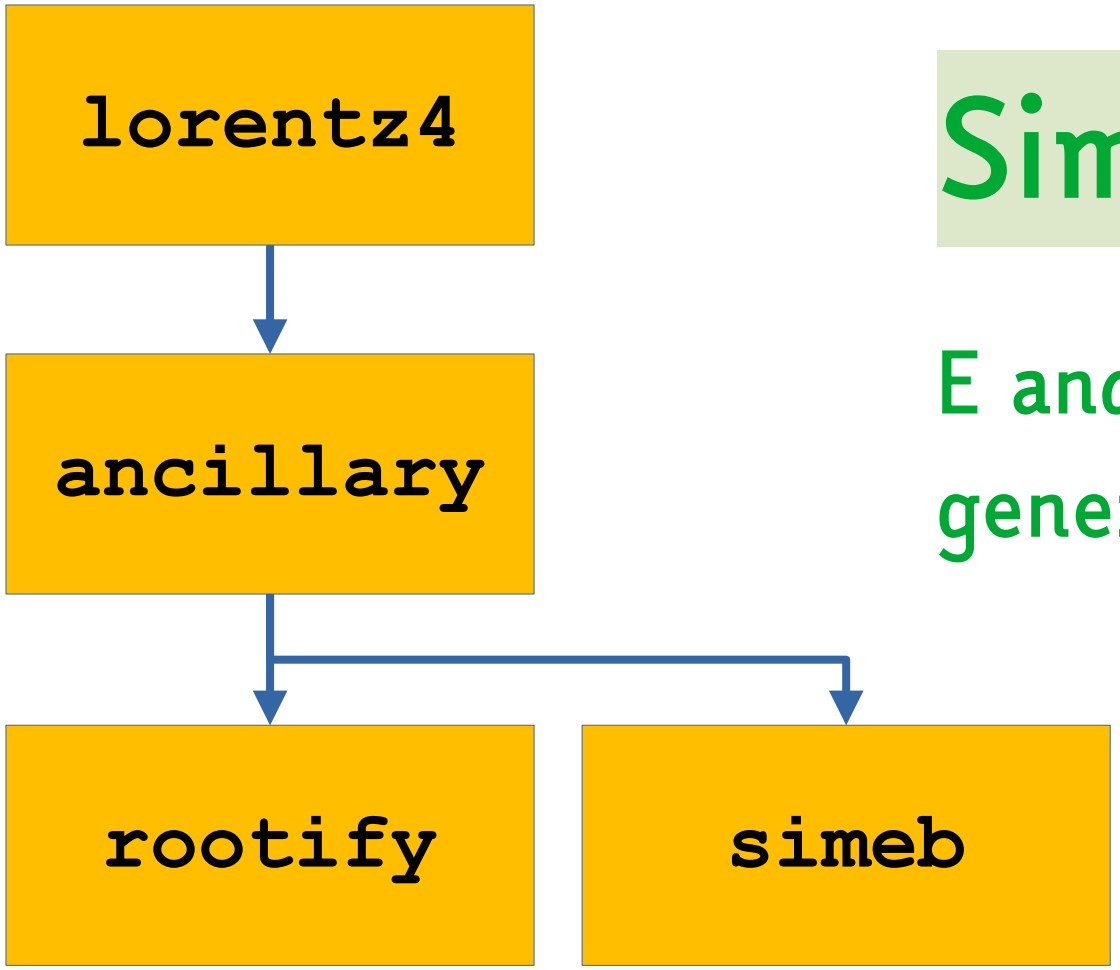
Hemispheres

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

```
# Initial dynamics: energy (eV) and polar angles [deg]
E0      12
theta0  90
phi0    0

# Initial position [m]
x0      -0.1
y0      0
z0      0.1

# Exteral fields (true if filename is present, 0 otherwise )
ExtF    0

# Eletric Field [V/m]
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)
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)
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)

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

# Down Sampling: max number of points in the track
Np      100000

# Stop conditions on trajectory (x, y, z)
StopIf  (sqrt(x*x+y*y+z*z)<0.05)+(sqrt(x*x+y*y+z*z)>0.15)+(z<0)*(x<0)

# Adaptive step: 0 = disabled; 1e-8 (or better) precision otherwise
Prcs    0

# Verbose and plots: 0 = disabled; 1 = enabled
Verb    1
```

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

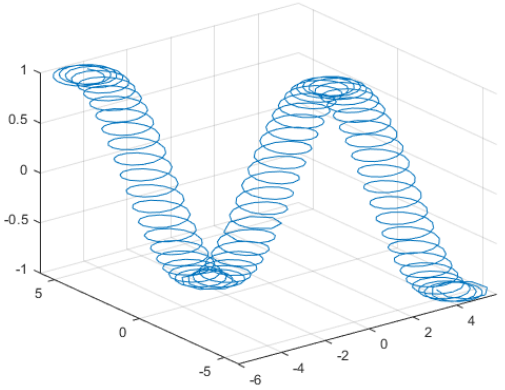




SimEB (beta)

E and B fields generator

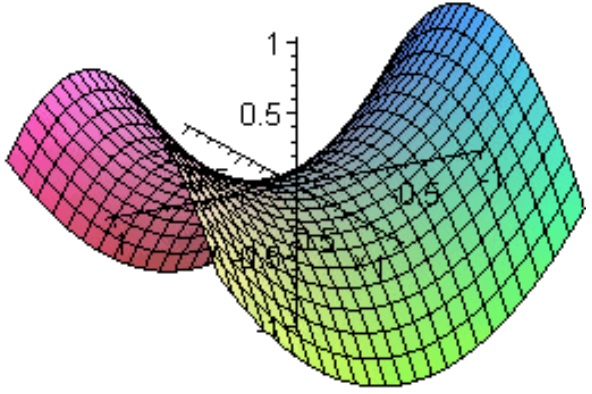
Descriptive geometry



B fields (Laplace)

Wires as parametric curves

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

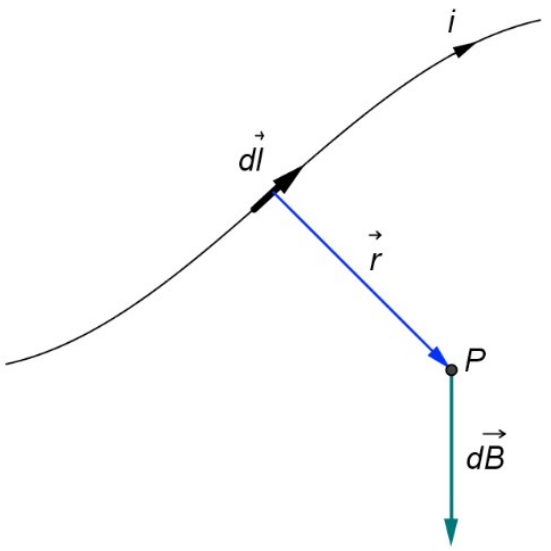


E fields (BEM)

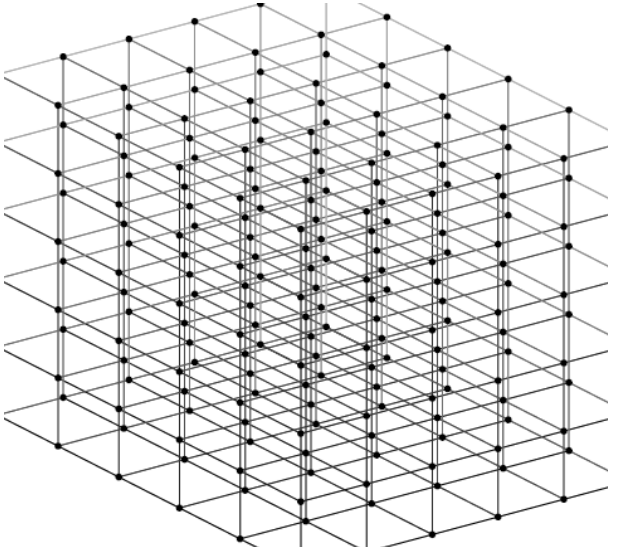
Equipotential conductive surfaces

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

Laplace formula



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



γ(t) = (x(t), y(t), z(t))

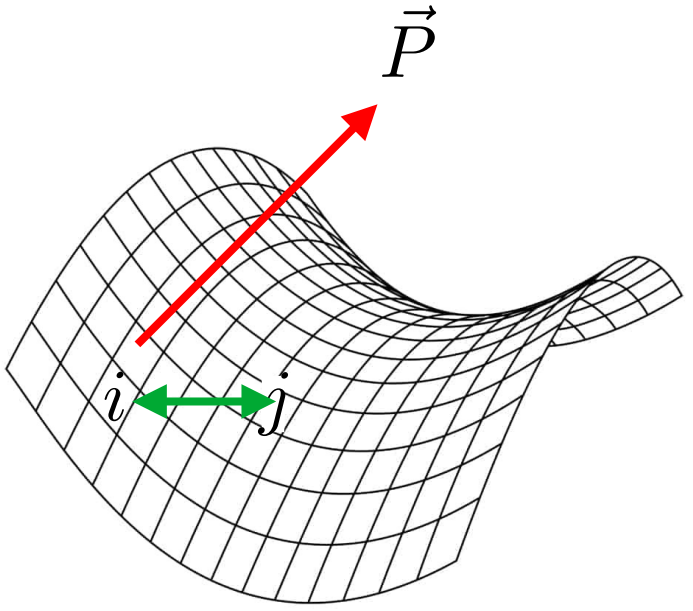
Boundary Element Method

$$\nabla^2 V = \sigma$$
$$V_\Sigma = V_0$$

P.D.E. with Dirichlet's conditions

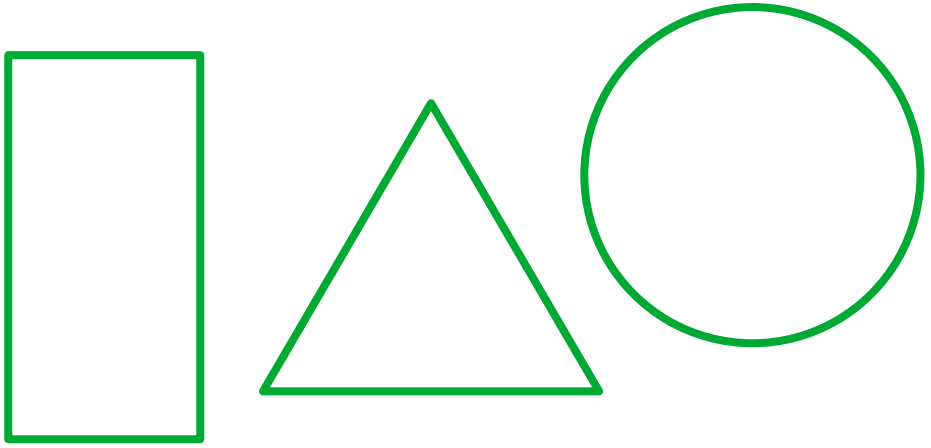
$$V_i \simeq \frac{1}{4\pi_0} \sum_j \frac{\sigma_j}{|\vec{r_i} - \vec{r_j}|} \Delta S_j = G_{ij}\sigma_j$$

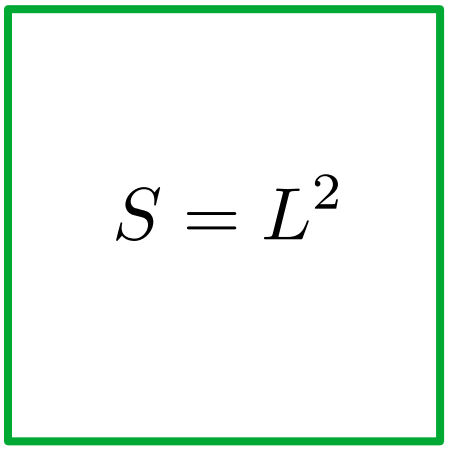
$$V(\vec{P}) = G_{pj}\sigma_j^*$$



Self interaction

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

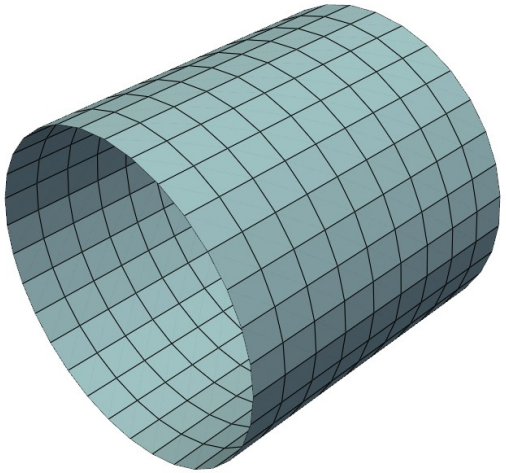


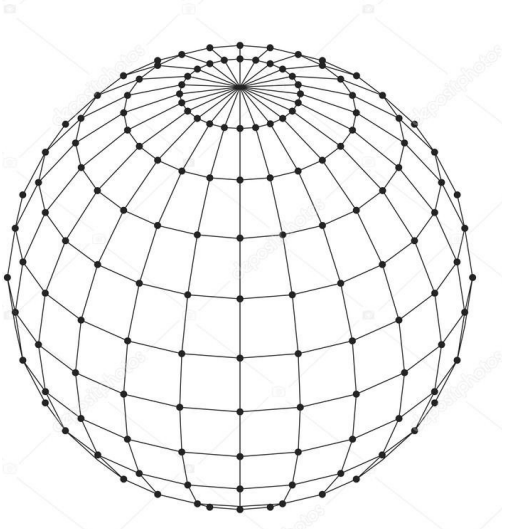


$$S = L^2$$

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

Meshing



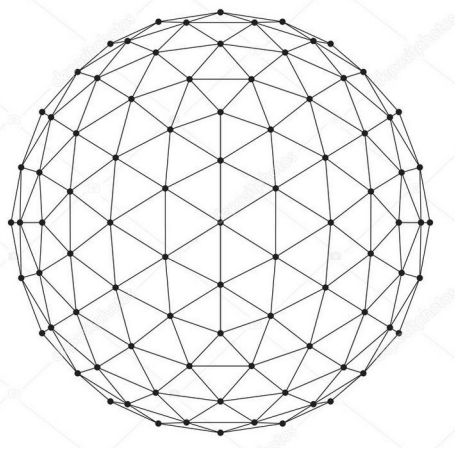


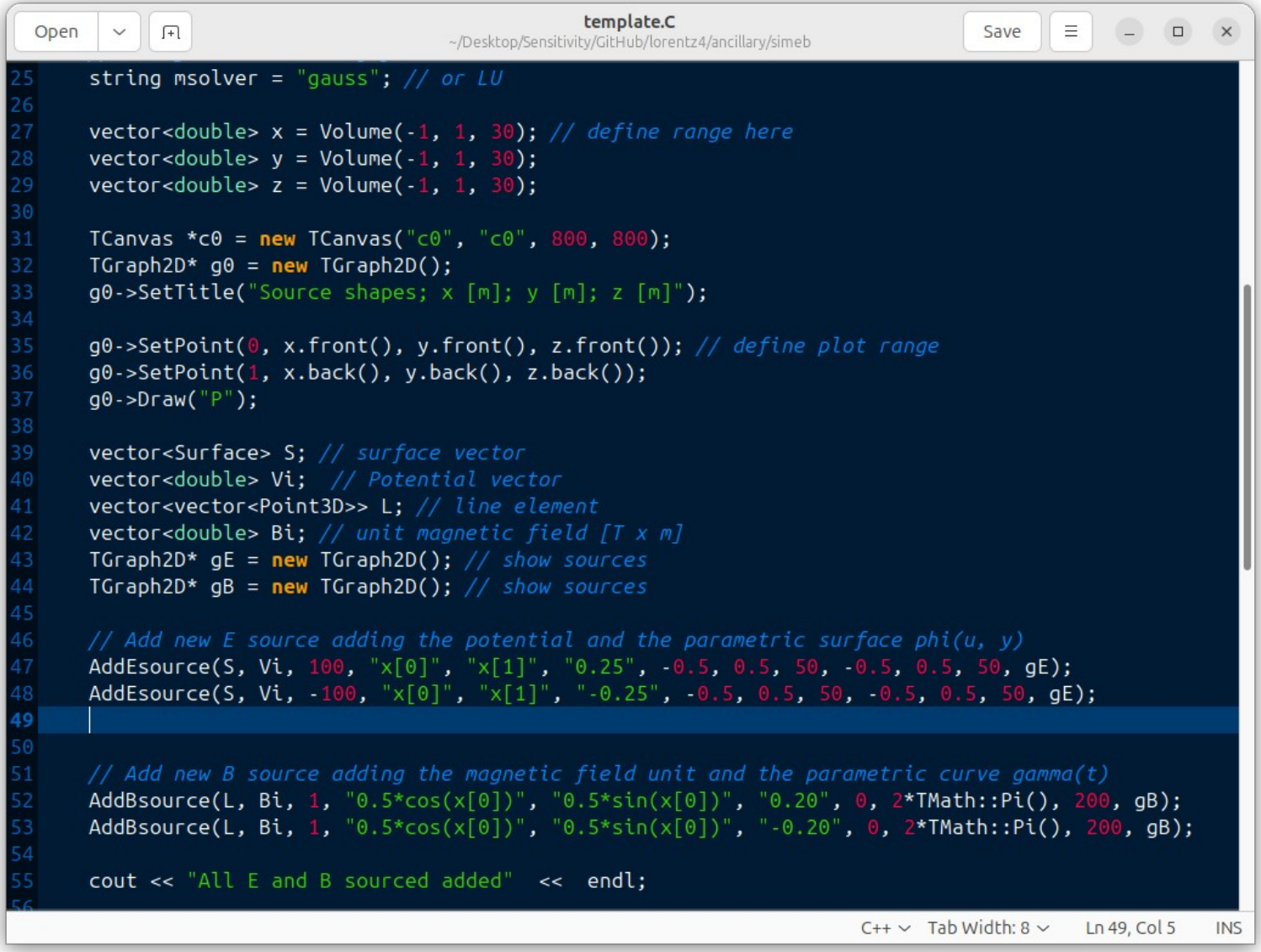
$$(u, v) \rightarrow \vec{\phi}(u, v)$$

SimEB meshing

Matrix inversion

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





template.C

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

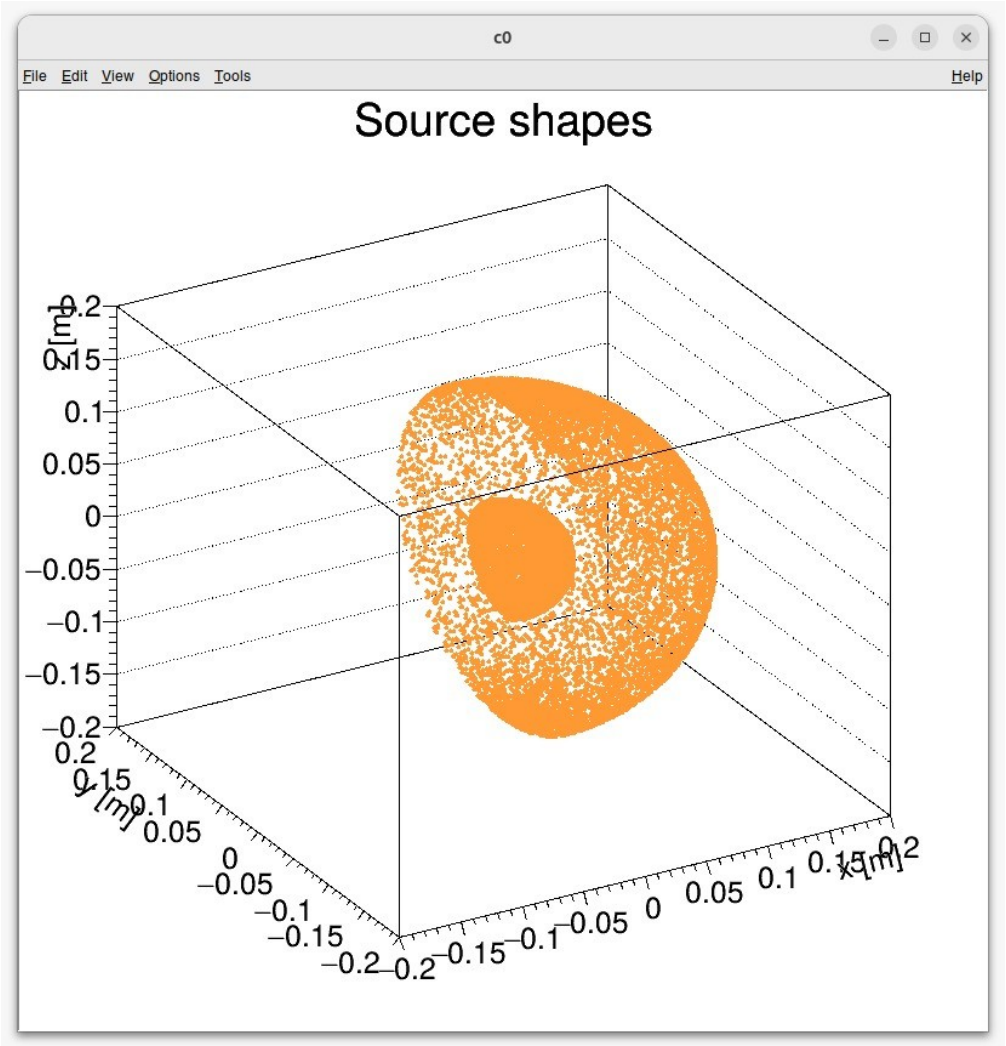
```
cp template.C mytemplate.C
```
(edit)

```
./compile.sh template.C
./template myout.root
```

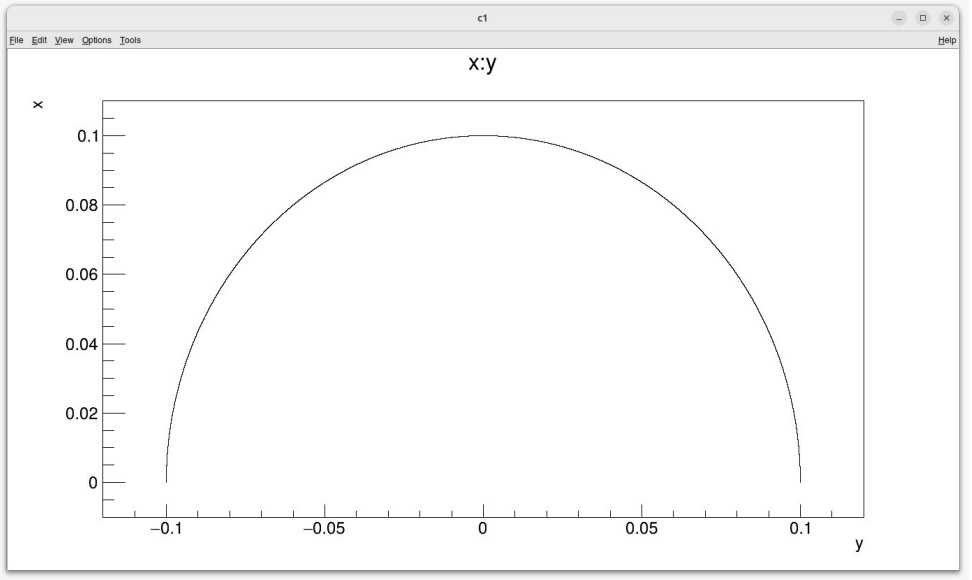
Generate `myout.root` in lorentz4 input format (`out.root` oterwise)

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

```
./drawfields myout.root
```

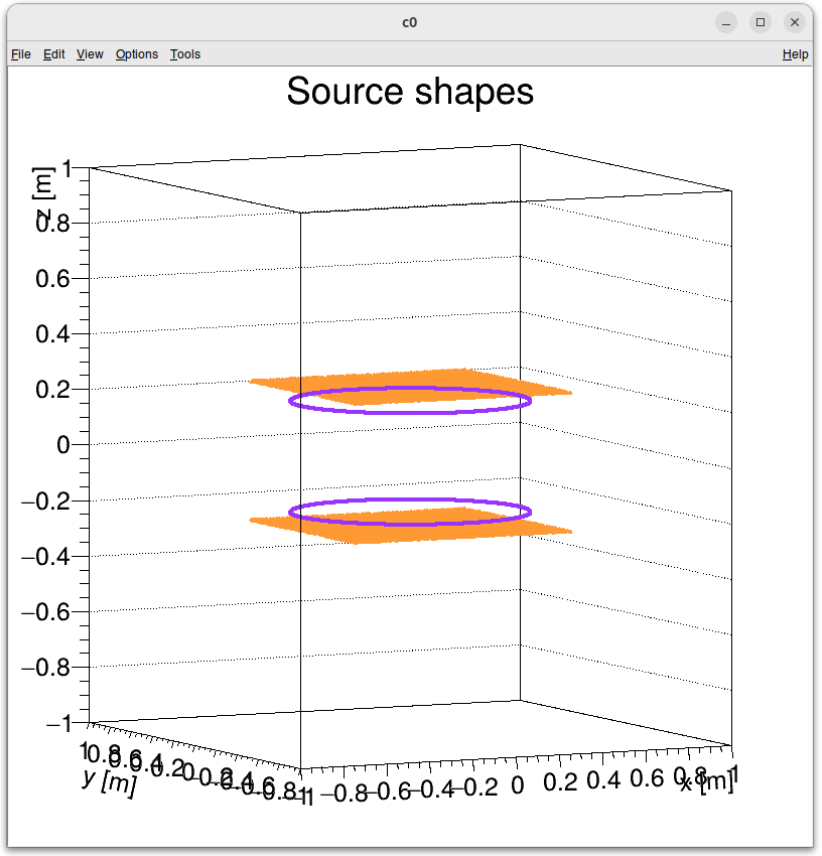


hemispheres

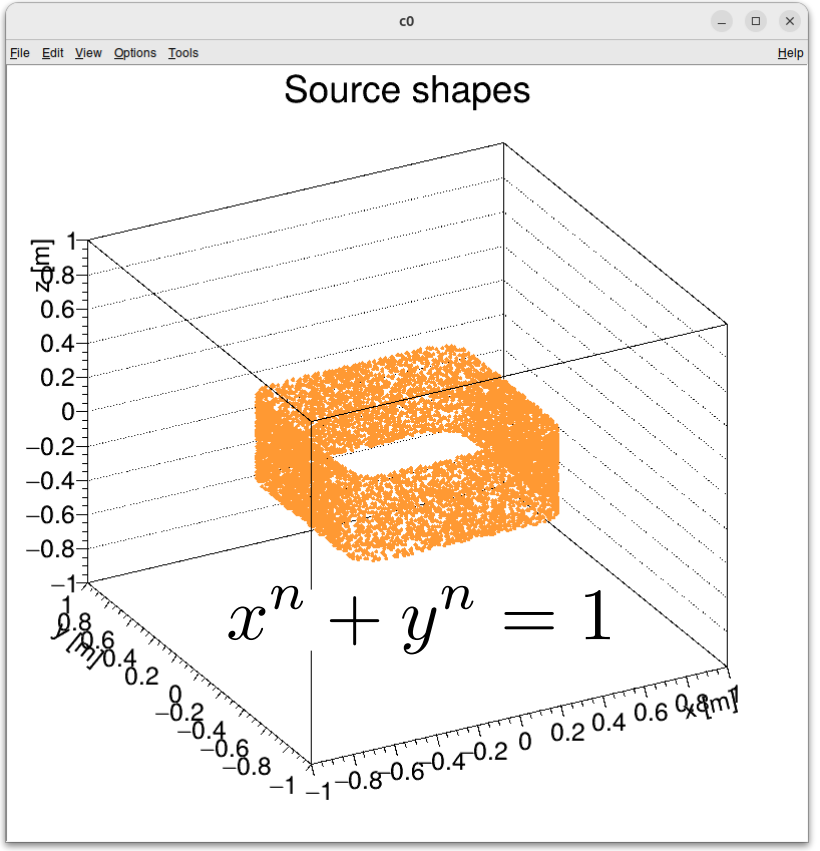


Track simulation
with **lorentz4**

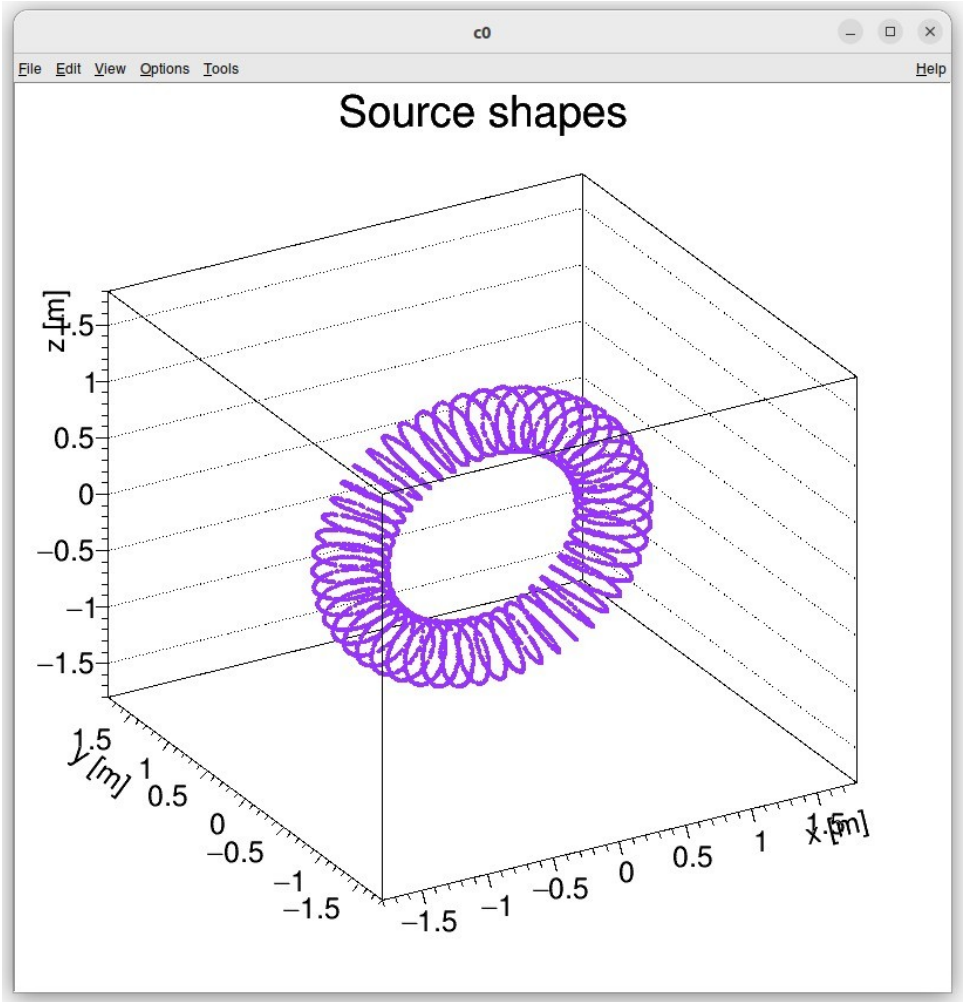
Examples



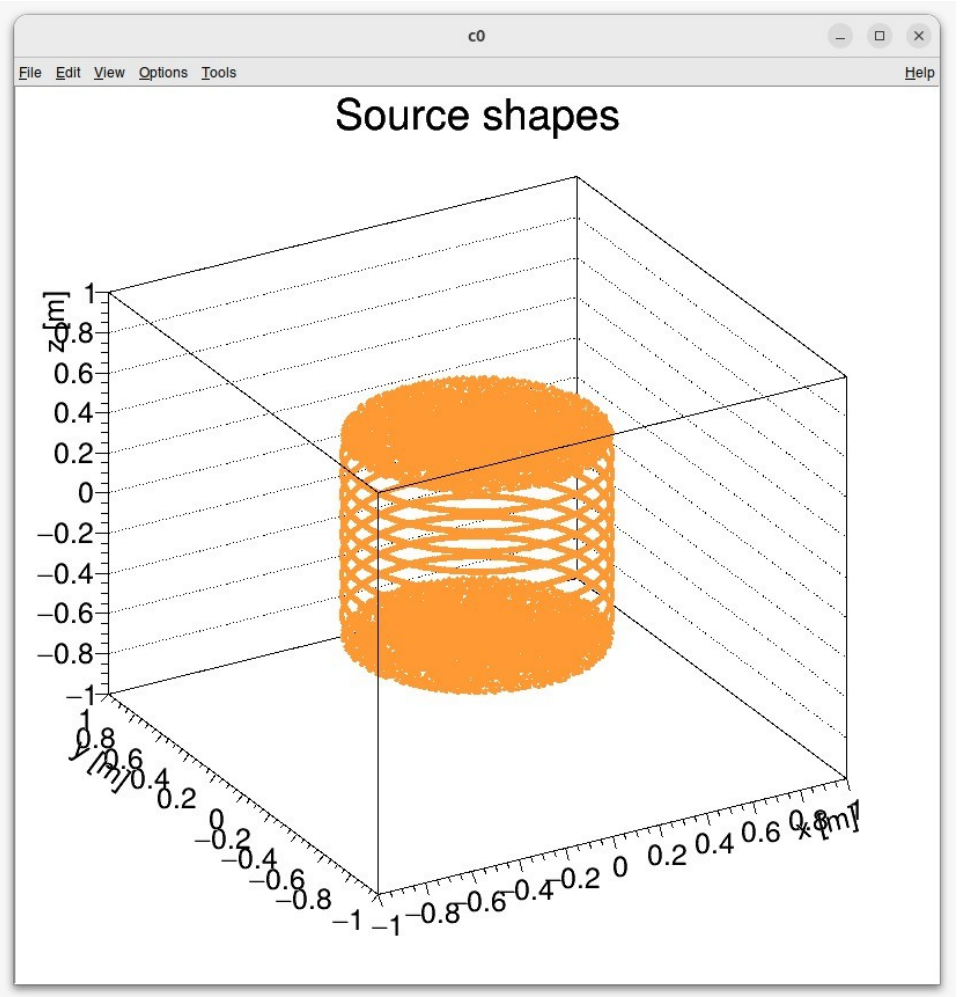
template



square (DarkSide)

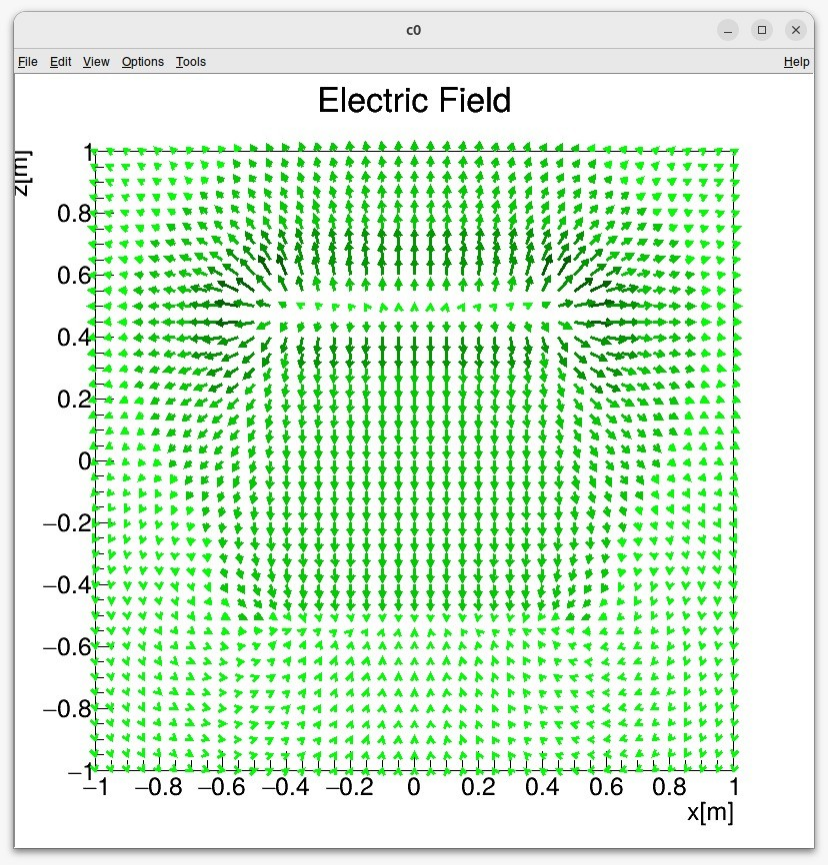


tokamak

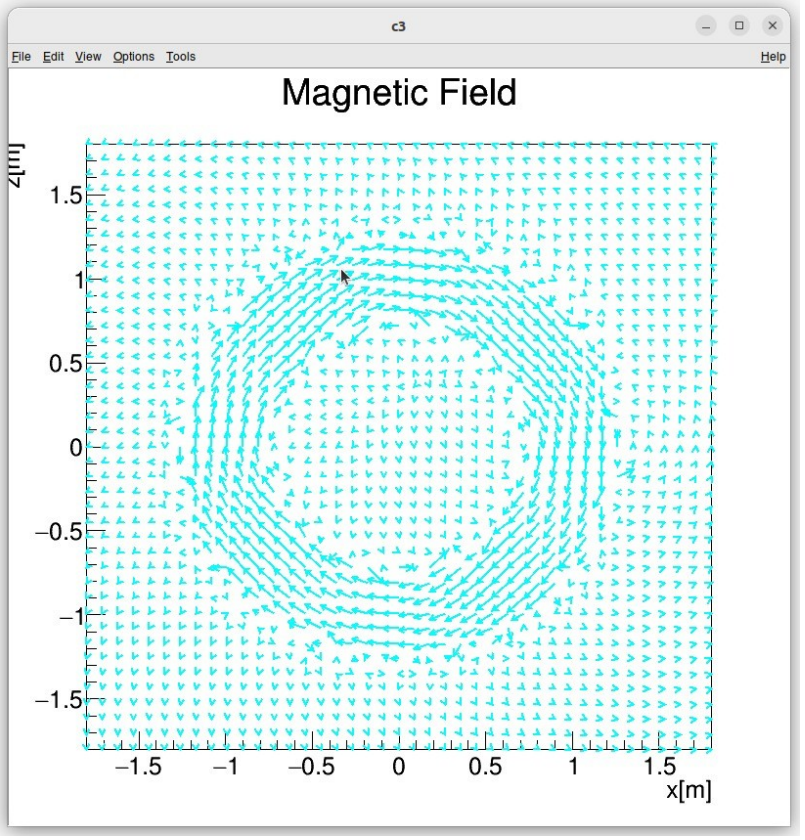


Tpc (XENONnT)

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