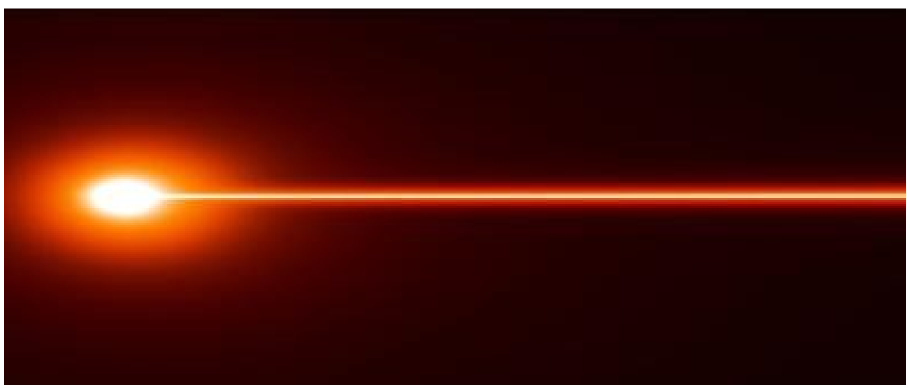
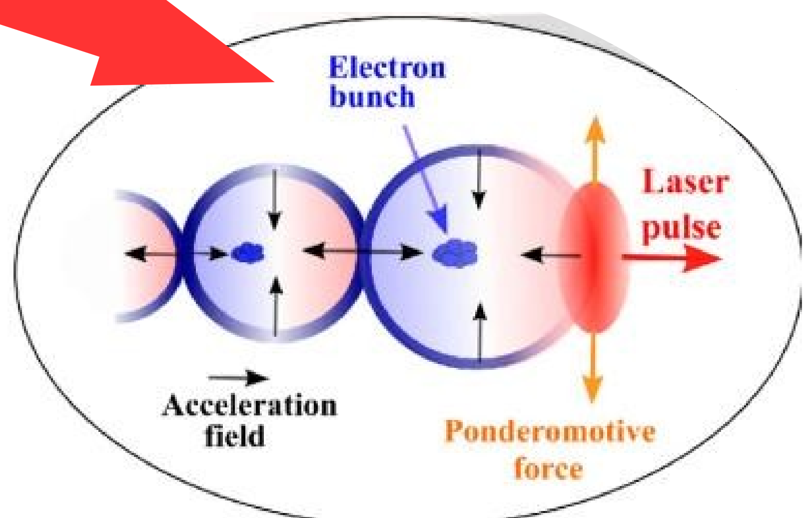
Geant4 ML-based laser-plasma sources term implementation in the context of the PALLAS beamline design

K. Cassou. A. Huber, V. Kubytskyi,
M. Lenivenko, A. Sytov

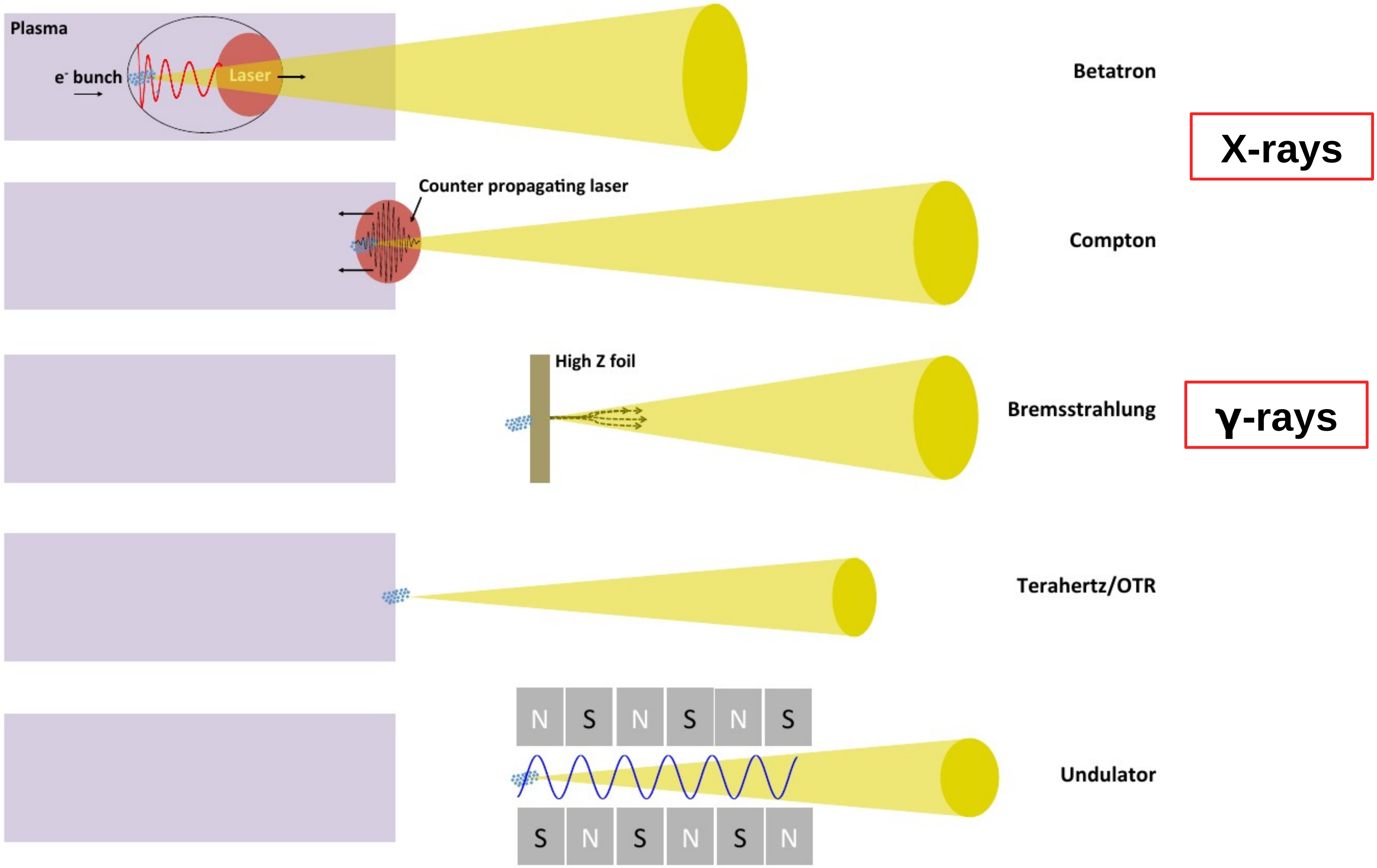
Laser plasma wakefield acceleration (LWFA)

Electron beam energy:
MeV — tens of **GeV**





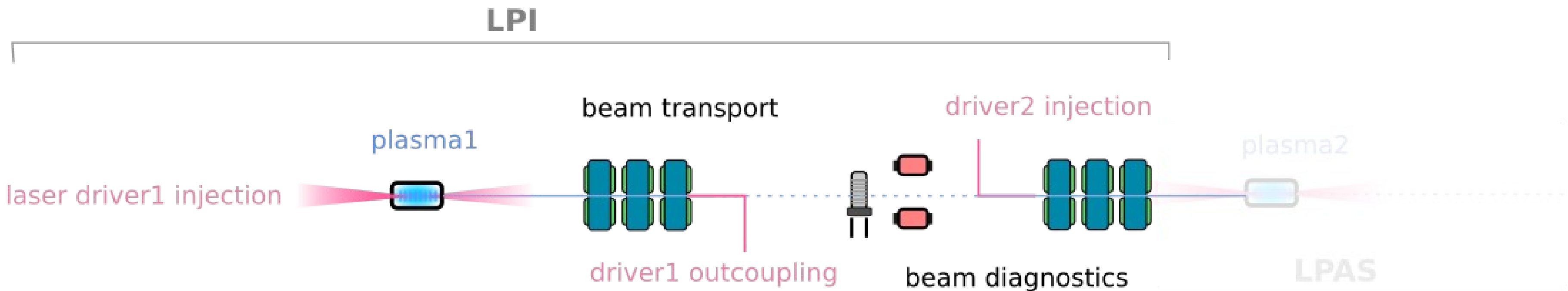
Radiation source based on plasma acceleration of electrons



PALLAS Project

Test facility for laser-plasma injector optimisation towards RF control reliability

In the context of advanced accelerator high quality beam laser plasma injector (LPI) for **EuPRAXIA** [1] preparatory technical design phase and future high gradient accelerator R&D at IJClab [2]: 10 Hz 250MeV LPI test facility to improve **quality and stability of e- beam generated by laser-plasma accelerator**.



Plasma target development [3]
- testing various gas cell type
- Continuous flow
- In-lin integration

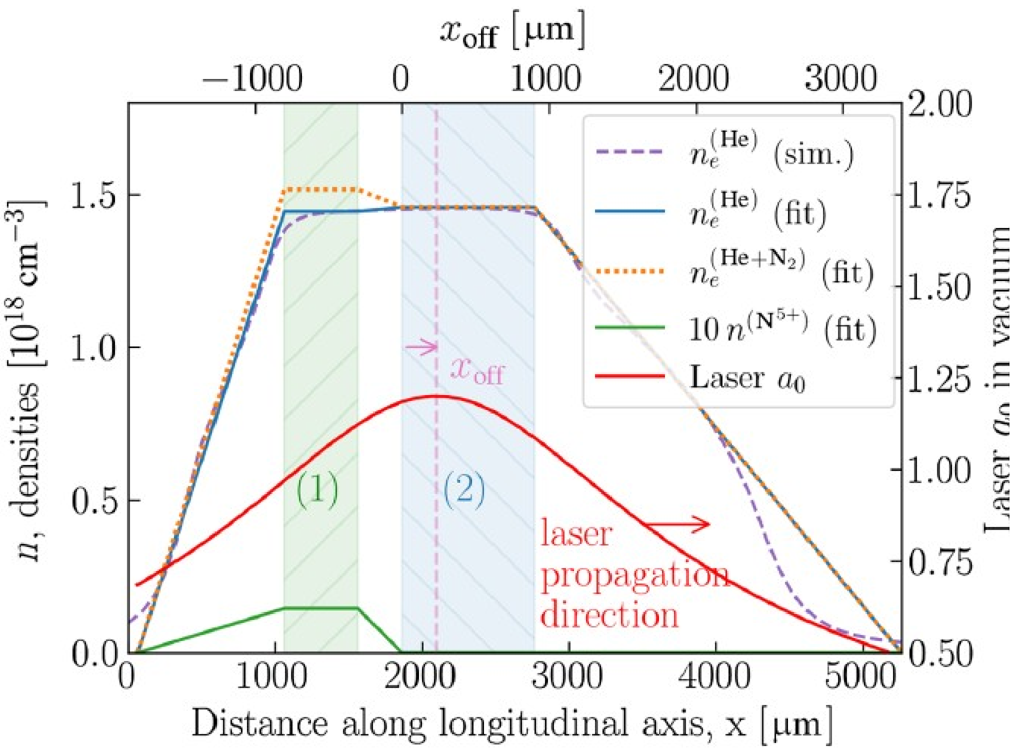
Beam Transport [4]
- compact flexible capture
- active plasma lens capture test
- collimator system for energy selection

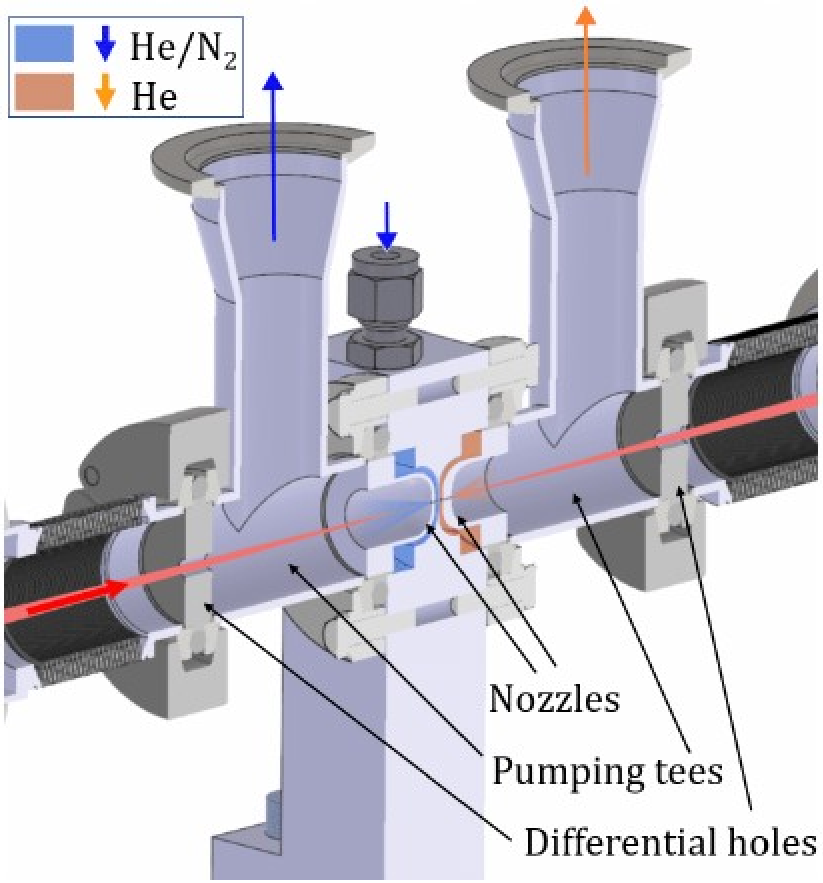
e- beam compact characterization
- down to 500 keV energy
- resolution emittance
- longitudinal phase space

[1] Assmann, R. W. et al. EuPRAXIA Conceptual Design Report. Eur. Phys. J. Spec. Top. 229, 3675–4284 (2020).
[2] pallas.ijclab.in2p3.fr
[3] Drobniak, P. et al. Random scan optimization of a laser-plasma electron injector based on fast particle-in-cell simulations. Phys. Rev. Accel. Beams 26, 091302 (2023), Drobniak, P. et al. Two-chamber gas target for laser-plasma accelerator electron source. Preprint at http://arxiv.org/abs/2309.11921 (2023).
[4] C. Guyot et al. Benchmarking CODAL beam dynamics code: a laser plasma case study, IPAC 2023 WEPL084

Gas flow simulations

Gas density profile generated using **OpenFOAM** code





First chamber (with N2 gas dopant) is used for injection, while the second one (pure He) for acceleration

PIC simulations

PHYSICAL REVIEW ACCELERATORS AND BEAMS **26,** 091302 (2023)

Random scan optimization of a laser-plasma electron injector based on fast particle-in-cell simulations

P. Drobniak,* E. Baynard, C. Bruni, K. Cassou, C. Guyot, G. Kane, S. Kazamias, V. Kubytskyi, N. Lericheux, B. Lucas, and M. Pittman
Laboratoire de Physique des 2 Infinis Irène Joliot-Curie—IJCLab—UMR 9012 CNRS Université Paris Saclay, 91405 Orsay cedex, France

F. Massimo
Laboratoire de Physique des Gaz et des Plasmas—LPGP—UMR 8578, CNRS, Université Paris-Saclay, 91405 Orsay, France

A. Beck and A. Specka
Laboratoire Leprince-Ringuet—LLR—UMR, 7638 CNRS Ecole polytechnique, 91128 Palaiseau cedex, France

P. Nghiem and D. Minenna
CEA-Irfu, Centre de Saclay, Université Paris-Saclay, 91191 Gif-sur-Yvette, France

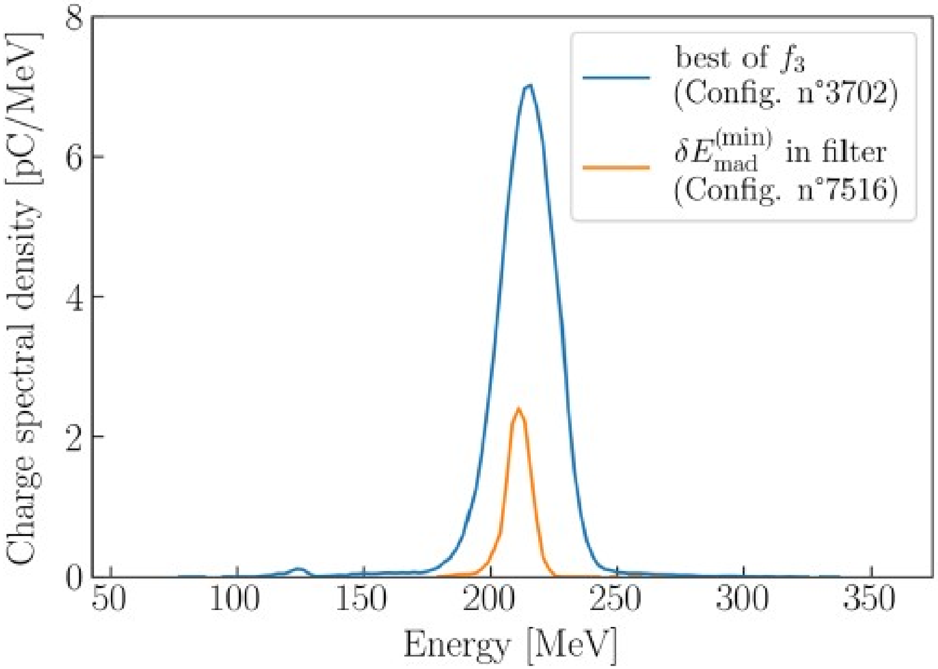


TABLE II. Input and beam parameters of LPI configurations 3702 and 7516.

	Best of f_3	$\delta E_{\mathrm{mad}}^{(\mathrm{min})}$ in filter
N°	3702	7516
Origin	RS2	RS4
p (mbar)	58.6	47.8
$a_{0,\mathrm{vac,max}}$	1.43	1.23
x_{off} (μm)	558	1680
c_{N_2} (%)	1.88	6.17
$a_{0,\mathrm{eff,max}}$	3.73	2.58
Q (pC)	198	30
E_{med} (MeV)	215	212
δE_{mad} (%)	3.53	1.55
$\epsilon_{y,n}$ (μm)	5.03	1.74

PIC simulations with **SMILEI**

PALLAS Project: ML

A modeling of the **full beamline** using **Geant4** is **required** for collimator design

Dataset and **first ML** model generated **by PALLAS**

Key idea:
ML model into Geant4 to create a **Geant4 electron beam source** based on plasma acceleration

```
#ifndef B1PrimaryGeneratorAction_h
#define B1PrimaryGeneratorAction_h 1

#include "G4VUserPrimaryGeneratorAction.hh"
#include "globals.hh"
#include "G4GeneralParticleSource.hh"
#include "G4ParticleGun.hh"
#include <memory>
#include <onnxruntime_c_api.h>
#include <onnxruntime_cxx_api.h>
class G4ParticleGun;
class G4Event;
```

```
PrimaryGeneratorAction::PrimaryGeneratorAction(): G4VUserPrimaryGeneratorAction(),
    fParticleGun(0),
    fEnvelopeBox(0)
{
    G4int n_particle = 1;
    fParticleGun  = new G4ParticleGun(n_particle);

    // default particle kinematic
    fParticleGun->SetParticleDefinition(
        G4ParticleTable::GetParticleTable()->FindParticle("e-"));

    //Neural network: create onnx session
    Ort::Env env(ORT_LOGGING_LEVEL_WARNING, "plasma");
    Ort::SessionOptions session_options;
    session_options.SetIntraOpNumThreads(1);
    auto sessionLocal =
        std::make_unique<Ort::Session>(env, "model2.onnx", session_options);
    fSession = std::move(sessionLocal);

    // Get input node information
    fMemory_info = Ort::MemoryInfo::CreateCpu(
        OrtAllocatorType::OrtArenaAllocator, OrtMemTypeDefault);
```

ML: the model

model input

p (mbar) → **pressure**

$a_{0,\text{vac,max}}$ → **normalized peak vector potential**

x_{off} (µm) → **laser focal position offset**

c_{N_2} (%) → **dopant concentration**

model output

Q (pC) → **beam charge**

E_{med} (MeV) → **beam energy**

δE_{mad} (%) → **beam energy spread**

$\epsilon_{y,n}$ (µm) → **beam emittance**

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 50)	250
p_re_lu (PReLU)	(None, 50)	50
dropout (Dropout)	(None, 50)	0
dense_1 (Dense)	(None, 50)	2,550
p_re_lu_1 (PReLU)	(None, 50)	50
dropout_1 (Dropout)	(None, 50)	0
dense_2 (Dense)	(None, 50)	2,550
p_re_lu_2 (PReLU)	(None, 50)	50
dropout_2 (Dropout)	(None, 50)	0
dense_3 (Dense)	(None, 4)	204

```
Model: "sequential"

Total params: 9,092 (35.52 KB)

Trainable params: 9,092 (35.52 KB)

Non-trainable params: 0 (0.00 B)
```

ML

```python
#################### Calling the dataframe and preparing the data ##########################
root_df = './'
root_models = './'

df01=pd.read_pickle("D:/B001/dataframe_surrogateopt_randscan.pickle")

# Deleting no injection configurations
df01['dE_mad'] = df01['dE_mad'].replace([0], np.nan)
df01 = df01[df01['dE_mad'].notna()]

# Choose needed columns (if take all columns scaler will have an error as some columns not numeric)
df01 = df01[["x_of", "a_0", "c_N2","p_1","E_med","dE_mad","q_end","n_emit_y"]]
df = df01.copy()

print(len(df))

min_values = df.min()
print("Minimum values:\n", min_values)

max_values = df.max()
print("Maximum values:\n", max_values)

print(repr(df))
```

Model from large & fast PIC simulations

```
Minimum values:
 x_of        -3.998247e+02
a_0          1.100516e+00
c_N2         2.064164e-03
p_1          1.009451e+01
E_med        5.849625e+01
dE_mad       8.269990e-04
q_end        6.498496e-17
n_emit_y     2.504754e-09
dtype: float64
Maximum values:
 x_of         1.798325e+03
a_0          1.849792e+00
c_N2         1.199827e-01
p_1          9.995741e+01
E_med        7.252073e+02
dE_mad       7.512749e-01
q_end        9.688866e-10
n_emit_y     8.082236e-05
dtype: float64
```

Rescaling

```python
# Data resclaing
scaler = MinMaxScaler() #MinMax works very good if borders of inputs in different data sets at least close to each other
df_transformed = scaler.fit_transform(df)
df = pd.DataFrame(df_transformed,
                  columns=df.columns.values.tolist())

min_values = df.min()
print("Minimum values:\n", min_values)

max_values = df.max()
print("Maximum values:\n", max_values)

print(repr(df))
```

```
Minimum values:
 x_of       0.0
a_0        0.0
c_N2       0.0
p_1        0.0
E_med      0.0
dE_mad     0.0
q_end      0.0
n_emit_y   0.0
dtype: float64
Maximum values:
 x_of       1.0
a_0        1.0
c_N2       1.0
p_1        1.0
E_med      1.0
dE_mad     1.0
q_end      1.0
n_emit_y   1.0
dtype: float64
```

Entrée [51]:
```python
# our test set
x_test.to_numpy()
```

Out[51]:
```
array([[0.04414279, 0.03696395, 0.14718525, 0.92301997],
       [0.98181373, 0.37218034, 0.6189666 , 0.19693942],
       [0.2311267 , 0.1554619 , 0.53956506, 0.61822893],
       ...,
       [0.90003744, 0.62109482, 0.49487012, 0.0943088 ],
       [0.70989143, 0.06717811, 0.68066123, 0.97240594],
       [0.73405388, 0.32554836, 0.07308455, 0.57993409]])
```

Model on G4 with ONNX

```
Output tensor values:
0.13585 0.618697 0.214069 0.246862
0.412162 0.0140921 0.00862622 7.28369e-05
0.157164 0.486243 0.312213 0.23243
```

Model on python with ONNX

Entrée [53]:
```python
#predict the same with onnx model
#you need to rerun entire notebook to get the same results as for keras
input_data=x_test.to_numpy()
session.run(None, {input_name: input_data})
```

Out[53]:
```
[array([[1.3585013e-01, 6.1869740e-01, 2.1406868e-01, 2.4686235e-01],
        [4.1216230e-01, 1.4092088e-02, 8.6262226e-03, 7.2836876e-05],
        [1.5716442e-01, 4.8624346e-01, 3.1221294e-01, 2.3242965e-01],
        ...,
        [3.6191791e-01, 1.4332026e-02, 8.8895857e-03, 3.8057566e-05],
        [9.0673089e-02, 6.6863406e-01, 2.7304298e-01, 5.6681705e-01],
        [6.4218014e-01, 5.3200334e-02, 9.9563301e-02, 7.5561881e-02]],
       dtype=float32)]
```

Correspondence OK

ML

```
G4double fXof_min = -399.8247;
G4double fXof_max = 1798.325;
G4double fXof_rescaled = fXof / (fXof_max - fXof_min) - fXof_min/(fXof_max - fXof_min);

G4double fA0_min = 1.100516;
G4double fA0_max = 1.849792;
G4double fA0_rescaled = fA0 / (fA0_max - fA0_min) - fA0_min/(fA0_max - fA0_min);

G4double fCN2_min = 0.002064164;
G4double fCN2_max = 0.1199827;
G4double fCN2_rescaled = fCN2 / (fCN2_max - fCN2_min) - fCN2_min/(fCN2_max - fCN2_min);

G4double fP1_min = 10.09451;
G4double fP1_max = 99.95741;
G4double fP1_rescaled = fP1 / (fP1_max - fP1_min) - fP1_min/(fP1_max - fP1_min);

G4cout << "Xof rescaled = " << fXof_rescaled << G4endl;
G4cout << "A0 rescaled = " << fA0_rescaled << G4endl;
G4cout << "CN2 rescaled = " << fCN2_rescaled << G4endl;
G4cout << "P1 rescaled = " << fP1_rescaled << G4endl;

std::vector<G4double> input_tensor_values =
    {fXof_rescaled, fA0_rescaled, fCN2_rescaled, fP1_rescaled};
```

```
/gun/SetStatusONNX true
/laser/SetOffsetLaserFocus 1680 ## um
/laser/SetNormVecPotential 1.23
/laser/SetFracDopTargetChamber 0.0617 # %(/100)
/laser/SetPressure 47.8 #mbar
```

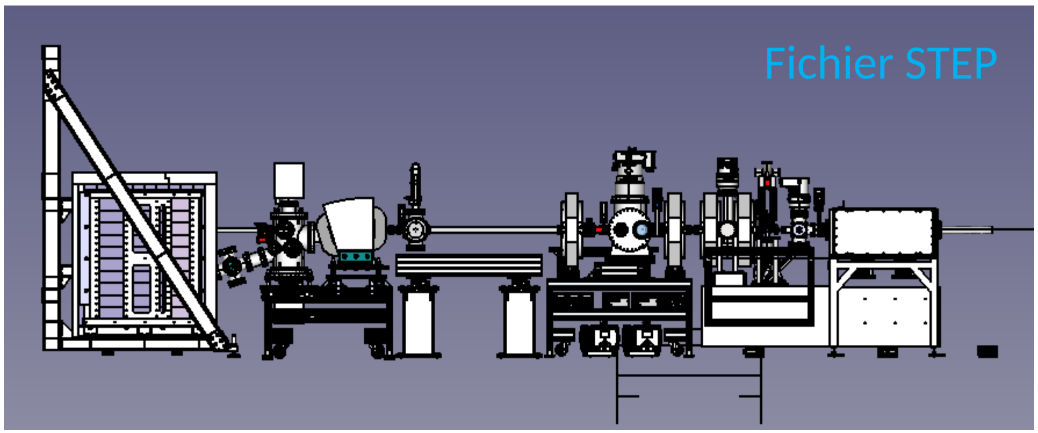
From real values, each parameter is rescaled according to min/max transformation

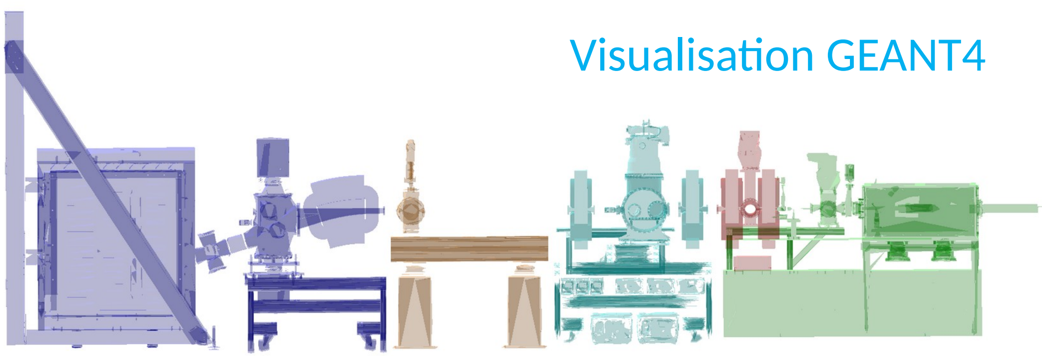
```
// kinetic energy
G4double Ekin_min = 58.49625*MeV;
G4double Ekin_max = 725.2073*MeV;
G4double Ekin = out_vals[0] * (Ekin_max - Ekin_min) + Ekin_min;
G4cout << "Ekin = " << Ekin << G4endl;

// spread of kinetic energy
G4double dEkin_min = 0.000826999;
G4double dEkin_max = 0.7512749;
G4double dEkin = out_vals[1] * (dEkin_max - dEkin_min) + dEkin_min;
G4cout << "dEkin = " << dEkin << G4endl;

G4double Q_min = 6.50e-17;
G4double Q_max = 9.69e-10;
G4double Q = out_vals[2] * (Q_max - Q_min) + Q_min;
G4cout << "Q = " << Q << G4endl;

// beam emittance (by now it's the same for x and y)
G4double epsb_min = 2.50e-9 ;
G4double epsb_max = 8.08e-5 ;
G4double epsb = out_vals[3] * (epsb_max - epsb_min) + epsb_min;
G4cout << "beam emittance = " << epsb << G4endl;

// gaussian with Ekin mean and dEking standard deviation
Ekin = G4RandGauss::shoot(Ekin, dEkin*Ekin);
G4cout << "Ekin from random shoot = " << Ekin << G4endl;
```

PALLAS in GEANT4

Modelization on GEANT4 :

- Modelization of PALLAS [via STEP files]



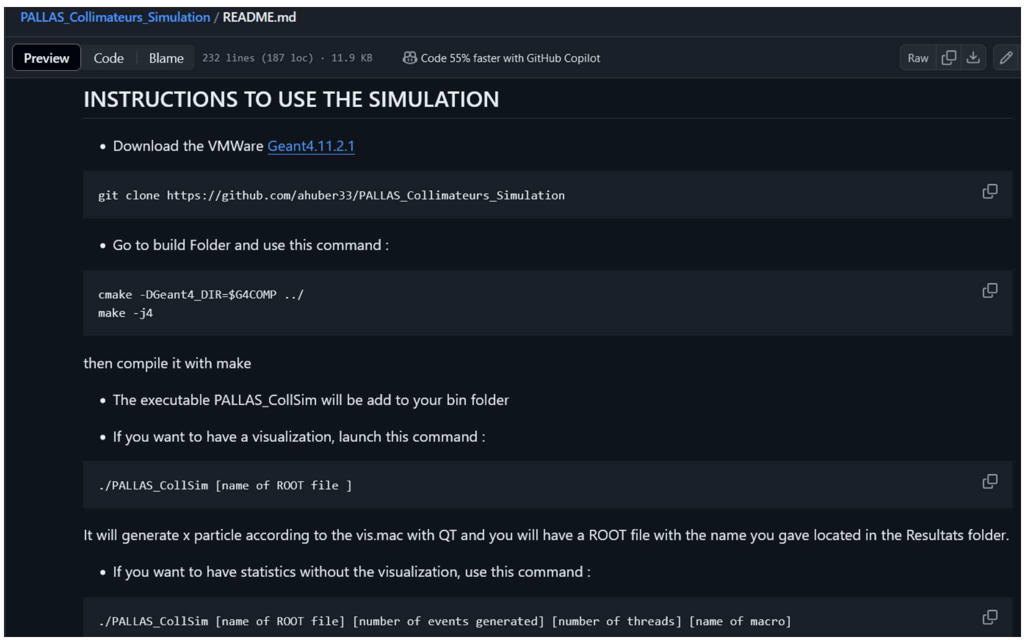


Convertion from STEP files into GDML using **FreeCAD** software

Simulation

<u>Modelization on Geant4 :</u>

- Modelization of PALLAS [via STEP files]
- Implementation MultiThreading
- Implementation collimators for studies
- Implementation documentation


PALLAS_Collimateurs_Simulation / README.md

Preview Code Blame 232 lines (187 loc) · 11.9 KB Code 55% faster with GitHub Copilot Raw

INSTRUCTIONS TO USE THE SIMULATION

- Download the VMWare Geant4.11.2.1

```
git clone https://github.com/ahuber33/PALLAS_Collimateurs_Simulation
```

- Go to build Folder and use this command :

```
cmake -DGeant4_DIR=$G4COMP ../
make -j4
```

then compile it with make

- The executable PALLAS_CollSim will be add to your bin folder
- If you want to have a visualization, launch this command :

```
./PALLAS_CollSim [name of ROOT file ]
```

It will generate x particle according to the vis.mac with QT and you will have a ROOT file with the name you gave located in the Resultats folder.

- If you want to have statistics without the visualization, use this command :

```
./PALLAS_CollSim [name of ROOT file] [number of events generated] [number of threads] [name of macro]
```

huber@cenbg.in2p3.fr

Simulation available on personal GitHub :
https://github.com/ahuber33/PALLAS_Collimateurs_Simulation/tree/main

Simulation also available on official GitLab PALLAS :
https://gitlab.in2p3.fr/pallas/design/-/tree/main?ref_type=heads

Modélisation

Modelization on Geant4 :

- Modelization of PALLAS [via STEP files]
- Implementation MultiThreading
- Implementation collimators for studies
- Implementation documentation

```
/control/alias VerticalCollimatorThickness 20
/control/alias HorizontalCollimatorThickness 60
/control/alias CollimatorSpectrometerDistance 50
/control/alias CollimatorVHDistance 10
/control/alias YParticleGenerationOffset 140 #Sum of precedent values

######################################################
########## PART FOR MESSENGER INFORMATIONS ##########
######################################################
/geometry/SetStatusRoundCollimator false
/geometry/SetCollimatorThickness 150 mm
/geometry/SetCollimatorInternalRadius 10 mm
/geometry/SetCollimatorExternalRadius 100 mm
/geometry/SetCollimatorDistanceBetweenPlates 0 mm

/geometry/SetVerticalCollimatorMaterial G4_W
/geometry/SetHorizontalCollimatorMaterial G4_Pb

/geometry/SetVerticalCollimatorThickness {VerticalCollimatorThickness} mm
/geometry/SetHorizontalCollimatorThickness {HorizontalCollimatorThickness} mm
/geometry/SetCollimatorSpectrometerDistance {CollimatorSpectrometerDistance} mm
/geometry/SetCollimatorVHDistance {CollimatorVHDistance} mm
/gun/SetYParticleGenerationOffset {YParticleGenerationOffset} mm
/geometry/SetOpenVerticalCollimator 0 mm
/geometry/SetOpenHorizontalCollimator 0 mm
/geometry/SetCollimatorLength 100 mm


/display/SetStatusDisplayCelluleGeometry false
/display/SetStatusDisplayLIFGeometry false
/display/SetStatusDisplaySection1Geometry false
/display/SetStatusDisplaySection2Geometry false
/display/SetStatusDisplaySection3Geometry false
/display/SetStatusDisplaySection4Geometry true
/display/SetStatusDisplaySection4DumpGeometry false
/run/reinitializeGeometry


/field/SetStatusMapBField false
/field/SetConstantBField 0.0 tesla #0.4 tesla

/step/SetTrackingStatus false
```

- */geometry/* corresponds to definition of specific volume & distance (here the Collimator) :
  - SetStatusRoundCollimator for the status of if we use Round collimator or not (USED FOR TESTS)
  - SetCollimatorThickness for the thickness of the collimator
  - SetCollimatorInternalRadius for the part of the collimator without matter
  - SetCollimatorExternalRadius for the external radius of the collimator
  - SetVerticalCollimatorMaterial to define the material of the vertical collimator
  - SetHorizontalCollimatorMaterial to define the material of the horizontal collimator
  - SetVerticalCollimatorThickness to define the thickness of the vertical collimator from the alias defined before
  - SetHorizontalCollimatorThickness to define the thickness of the horizontal collimator from the alias defined before
  - SetCollimatorSpectrometerDistance to define the distance between the end of the collimator and the begin of spectrometer from the alias defined before
  - SetCollimatorVHDistance define the distance between the end of the vertical collimator and the begin of the horizontal collimator from the alias defined before
  - SetYParticleGenerationOffset define the Offset parameter for the aprticle generation according to the different values of the 2 collimators. => DEFINED WITH GUN MESSENGER
  - SetOpenVerticalCollimator define the aperture of the vertical collimator
  - SetOpenHorizontalCollimator define the aperture of the horizontal collimator
  - SetCollimatorLength define the other dimensions of the collimator
  - **IMPORTANT** You can find a **Geometry.cc** file where all the possible LogicalVolume are created and a **PALLAS_CollSimGeometryConstruction.cc** where these functions are call to construct the geometry.
- */display/* manages if some part of the geometry are taken into account or no :
  - SetStatusDisplayCelluleGeometry for the "2 cells part"
  - SetStatusDisplayLIFGeometry for the LIF part
  - SetStatusDisplaySection1Geometry for the 2 first Quadrupole with ISOChamber
  - SetStatusDisplaySection2Geometry for Q3, Q4, ASMRemovalChamber, BreadboardRemovalChamber, ChassisRemovalChamber and ISOTubes
  - SetStatusDisplaySection3Geometry for ASMPoutre & YAGStation
  - SetStatusDisplaySection4Geometry for DipoleChamber, Dipole & 2 YAGs
  - SetStatusDisplaySection4DumpGeometry for shieldings, chassis & DiagsChamber
  - /run/reinitializeGeometry is mandatory to take into account the messegner informations
- */field/* manages the magnetic field inside the spectrometer :

PALLAS in GEANT4

<u>Modelization on GEANT4 :</u>

- Modelization of PALLAS [via STEP files]
- Implementation MultiThreading
- Implementation collimators for studies
- Implementation documentation

- *ML model for particles generation [collaboration with IJCLab & INFN]*
- **ML model by IJCLab**
- **ML implementation in Geant4 using ONNX by INFN**

- **Implementation of tracking in optics (quad)**
- **Submission of the example into Geant4**
- **Development of more complex neural network models (GP-based surrogate model, ...)**


Thank you for attention!