

# Primera simulación de la nube de electrones empleando PyECLOUD

Profesores:

Dra. Karla Beatriz Cantún Avila  
Dr. Geoffrey Humberto Israel Maury Cuna

Febrero, 2022

*En el presente documento encontrarás las instrucciones necesarias para preparar la carpeta de trabajo para realizar la primera simulación de la nube de electrones a un acelerador circular empleando el código PyECLOUD. Este está dividido en tres secciones que se refieren a las siguientes acciones:*

- Crear la carpeta de trabajo
- Iniciar la simulación con PyECLOUD
- Revisar archivos arrojados por la simulación

*Es necesario que cuentes con los siguientes recursos:*

- Computadora de escritorio o portátil personal con PyECLOUD instalado.
- Conexión estable a internet

## *Carpeta para la simulación*

Para crear una carpeta de trabajo es necesario clonar una de las carpetas que incluye como ejemplo el código PyECLOUD:

1. Crear una carpeta de trabajo en la ubicación deseada, fuera de la carpeta PyECLOUD\_Friends  

```
mkdir /home/usuario/ecloud_buildup_simulation
```
2. Cambiarse a la carpeta creada  

```
cd /home/usuario/ecloud_buildup_simulation
```
3. Copiar la carpeta ejemplo de PyECLOUD en la que se encuentran los archivos que permiten configurar la simulación.  

```
cp ../PyECLOUD_FRIENDS/PyECLOUD/testing/tests_buildup/  
LHC_ArcDipReal_450GeV_sey1.70_2.5e11ppb_b1_1.00ns/* .
```

## Ejecutar una simulación

Para realizar la simulación es necesario contar con el archivo “script” en el que se establecen las secuencias de los comandos del depurador para ejecutar el programa.

1. Clonar el archivo script a la carpeta de trabajo

```
cp ../PyECLOUD_Friends/PyECLOUD/testing/  
tests_buildup/000_run_simulation.py .
```

2. En el script debe modificarse, tanto la trayectoria en la que se encuentran los archivos de PyECLOUD como los archivos de la configuración de la simulación. Estos se establecen en las variables:

```
BIN y sim_folder
```

Actualiza cada una según la trayectoria en la que se ubique cada uno de estos elementos en tu cuenta.

3. Correr la simulación

```
python 000_run_simulation.py
```

4. Se generará como resultado de la ejecución un archivo de extensión .mat que contiene los arreglos de valores de las variables consideradas en el programa para la simulación

## Resultados de la simulación

Para trabajar con los resultados de la simulación se requiere de una aplicación que permita visualizar el contenido del archivo generado. Por ser una aplicación de código abierto, y versatilidad se recomienda el uso de la plataforma interactiva Jupyter Notebook

1. Se recomienda crear una carpeta donde desee guardar los archivos de código referentes a las gráficas.

```
mkdir /home/usuario/simulations
```

2. Cambiarse dentro de la carpeta.

```
cd /home/usuario/simulations
```

3. Copiar el archivo.mat dentro de la carpeta.

```
mv ../PyECLOUD_FRIENDS/PyECLOUD/testing/  
tests_buildup/000_run_simulation.py/archivo.mat .
```

- Instalar jupyter notebook con PIP escribiendo lo siguiente en la terminal:

```
pip install jupyterlab
```

4. Abrir en la carpeta que se desee trabajar jupyter notebook, ejecutando lo siguiente en la terminal:

```
jupyter notebook
```

A continuación, se abrirá la ventana principal de jupyter notebook que es como un visor de carpetas. Localizar la carpeta que creó en el paso 1 y abrirla.

Para realizar los análisis debemos explorar el contenido del archivo *archivo.mat* y de ahí seleccionar las variables con las que se debe trabajar.

### Importar librerías

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import loadmat
```

### Cargar el archivo (verificar la trayectoria en la que se encuentra)

```
mat=loadmat('archivo.mat')
```

### Crear una variable que almacene el nombre de las variables del archivo

```
matkeys = list(mat.keys())
```

### Desplegar la lista de variables

```
matkeys
```

### Remover los primeros tres valores de la lista que corresponden a los encabezados de la tabla

```
matkeys.remove('__header__')
matkeys.remove('__version__')
matkeys.remove('__globals__')
```

### Desplegar la tabla de variables enlistando el tamaño y tipo de datos almacenados en cada una

```
for key in matkeys:
    print('Parametro: ', key, ' Tamaño: ', mat[key].shape,
          ' Tipo: ', mat[key].dtype)
```

### Generar gráfica asociando las variables del archivo a los identificadores x y y

```
x= mat['t'].flatten()
y= mat['Nel_timep'].flatten()
```

### Establecer entorno de graficación, nombre de ejes y graficar

```
fig = plt.figure(figsize=(4,3))
ax=fig.add_subplot(111)
ax.set_xlabel('Tiempo [s]')
ax.set_ylabel('Num. de electrones')
ax.plot(x,y)
plt.plot()
```

# *Datos Importantes*

¿En cuál archivo de entrada se modifican los parámetros relacionados con el modelo de emisión secundaria?

¿En cuál archivo de entrada se establece el patrón de llenado?

¿Cuál es la variable que debe modificarse en dicho archivo para establecer un patrón de llenado determinado?

¿Cómo establecería en dicha variable un patrón de llenado de 4 trenes de 72 paquetes de partículas c/u con un espaciamiento entre trenes de 8 paquetes vacíos?

¿Cuál es la variable o parámetro donde se establece la intensidad el haz (es decir, el número de protones por paquete) y en qué archivo de entrada se encuentra?

Configure la siguiente sección magnética un cuadrupolo con un gradiente de  $100 \text{ T/m}^2$ , indicar el parámetro y archivo de entrada que debe modificarse y cuál sería la sintaxis

Configure una sección recta libre de campo magnético. Indique el archivo de entrada y la(s) variable(s) que deben modificarse

Configure una simulación que considere únicamente la ionización del gas residual como mecanismo de producción de la nube de electrones y una presión de vacío de 32 nTorr. Indique el archivo de entrada y las variables que deben de modificarse o activarse

¿Cuál es el parámetro y el archivo de entrada donde se establecería un haz con energía de 3 TeV?