6.1.2

September 13, 2022

1 Imports

[ ]:
```
import sdf_helper as sh
import matplotlib.pyplot as plt
import os
from matplotlib import colors
import sdf
plt.rcParams["font.size"]=13
```

[ ]:
```
EXTENT = [-20,20, -25, 25]
```

2 Parameters

- Simulation Box: 40 $\mu m \times 40\ \mu m \times 50\ \mu m$
- Number of Cells: $400 \times 400 \times 1000$
- Plasma Density: $n_0 = 10^{26}\ m^{-3}$
- Laser Intensity: $I = 10^{23}\ Wm^{-2}$
- Laser Wavelength: $\lambda = 10^{-6}\ m$
- Particle Per Cell: 2
- FWHM of the Laser: $1\ \mu m$
- Width of the Laser: $5\ \mu m$
- The laser propagates in z direction and is s-polarized, that is, the electric field vector of the laser oscillates in the x axis.
- The laser starts at $(20\,\mu m \times 20\,\mu m \times -2\,\mu m)$ which is the center of the xy-plane $2\,\mu m$ in front of the simulation box.

3 Fields With Time

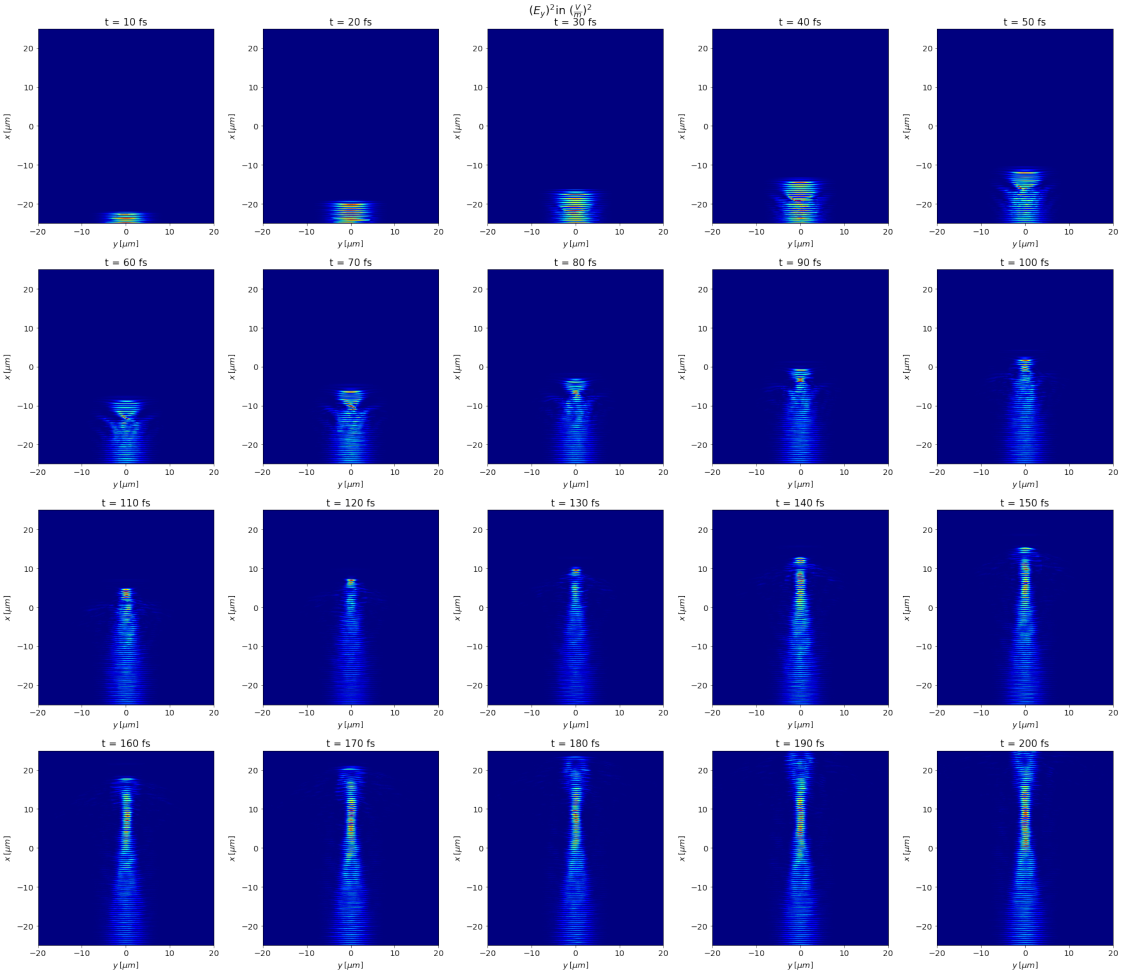
[ ]:
```
def plot_field(data_dir, ax, component="y"):
    raw_data = sdf.read(data_dir)
    comp = {
        "x":raw_data.Electric_Field_Ex,
        "y":raw_data.Electric_Field_Ey,
        "z":raw_data.Electric_Field_Ez,
    }
    field = comp[component].data
    t = raw_data.Header['time']*1e15
    field = field/(field.max()+1e-10)
    ax.imshow(field**2, cmap='jet', origin='lower',
    extent=EXTENT,
    aspect='auto',
    interpolation='nearest',
    # norm=colors.Normalize(vmin=-0.3, vmax=0.5),
    )
    ax.set_xlabel('$y \, [\mu m]$')
    ax.set_ylabel('$x \, [\mu m]$')
    ax.set_title(f't = {t:.0f} fs')
```

[ ]:

```
all_files = [f for f in os.listdir('.') if f.endswith('.sdf')]
```

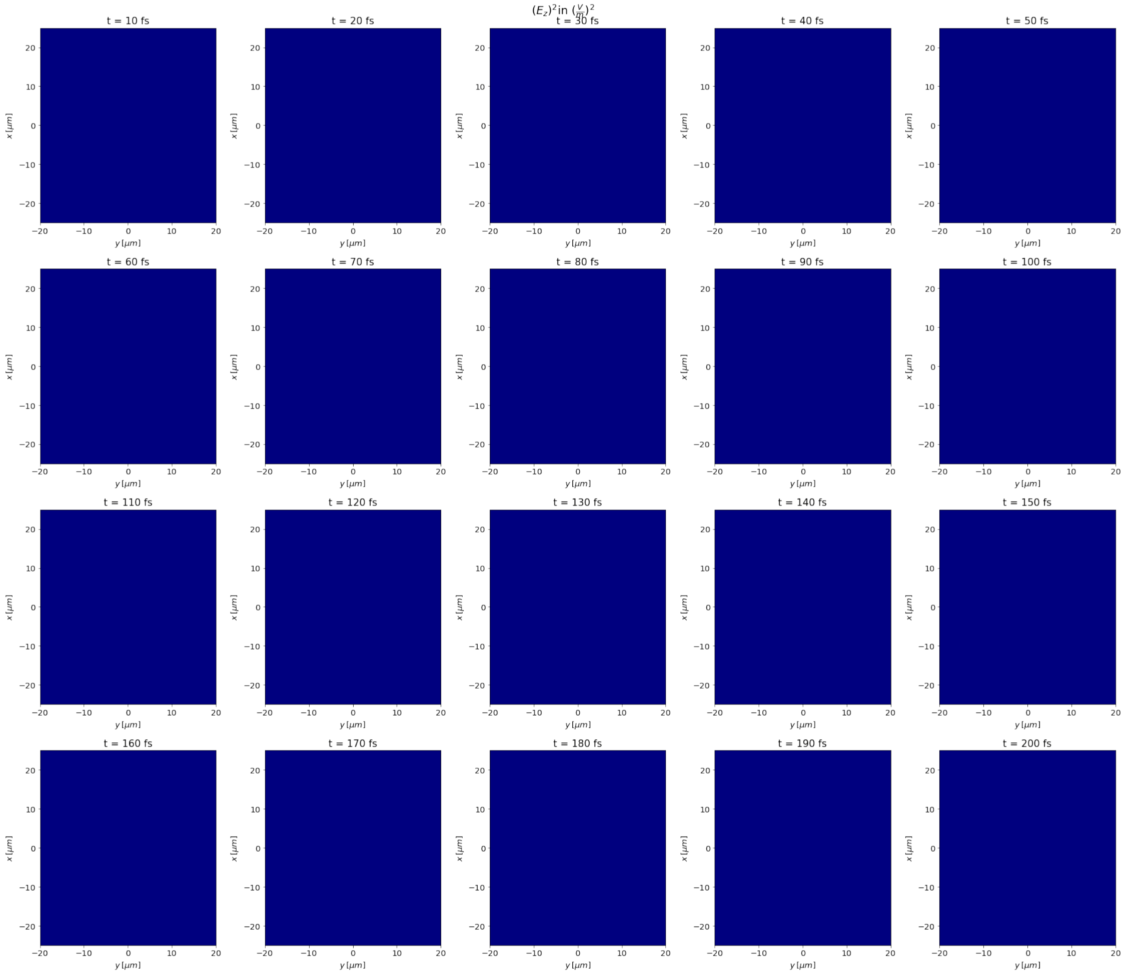
[ ]:

```
fig, ax = plt.subplots(4, 5, figsize=(25, 22))
i=0
component="y"
field = f"E_{component}"
fig.suptitle(rf"$({field})^2$" +r"in $({\frac{V}{m}})^2$", fontsize=18)
i=0
for data_dir in all_files[1:]:
    t = i*10
    plot_field(data_dir, ax[i // 5, i % 5],component=component)
    i += 1
fig.tight_layout()
```



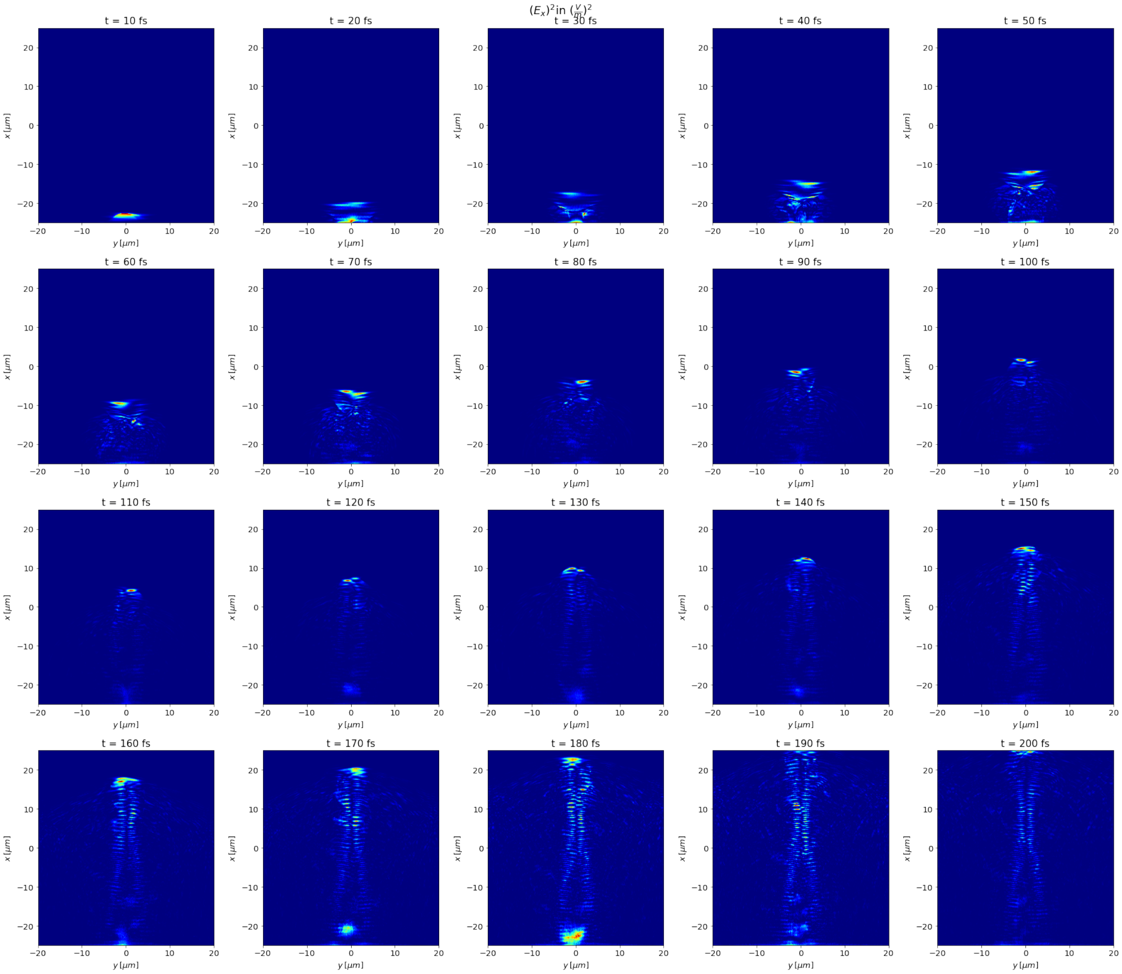
[ ]:

```
fig, ax = plt.subplots(4, 5, figsize=(25, 22))
i=0
component="z"
field = f"E_{component}"
fig.suptitle(rf"$({field})^2$" +r"in $({\frac{V}{m}})^2$", fontsize=18)
i=0
for data_dir in all_files[1:]:
    t = i*10
    plot_field(data_dir, ax[i // 5, i % 5],component=component)
    i += 1
fig.tight_layout()
```



[ ]:

```
fig, ax = plt.subplots(4, 5, figsize=(25, 22))
i=0
component="x"
field = f"E_{component}"
fig.suptitle(rf"$({field})^2$" +r"in $({\frac{V}{m}})^2$", fontsize=18)
i=0
for data_dir in all_files[1:]:
    t = i*10
    plot_field(data_dir, ax[i // 5, i % 5],component=component)
    i += 1
fig.tight_layout()
```



4 Field With Scale

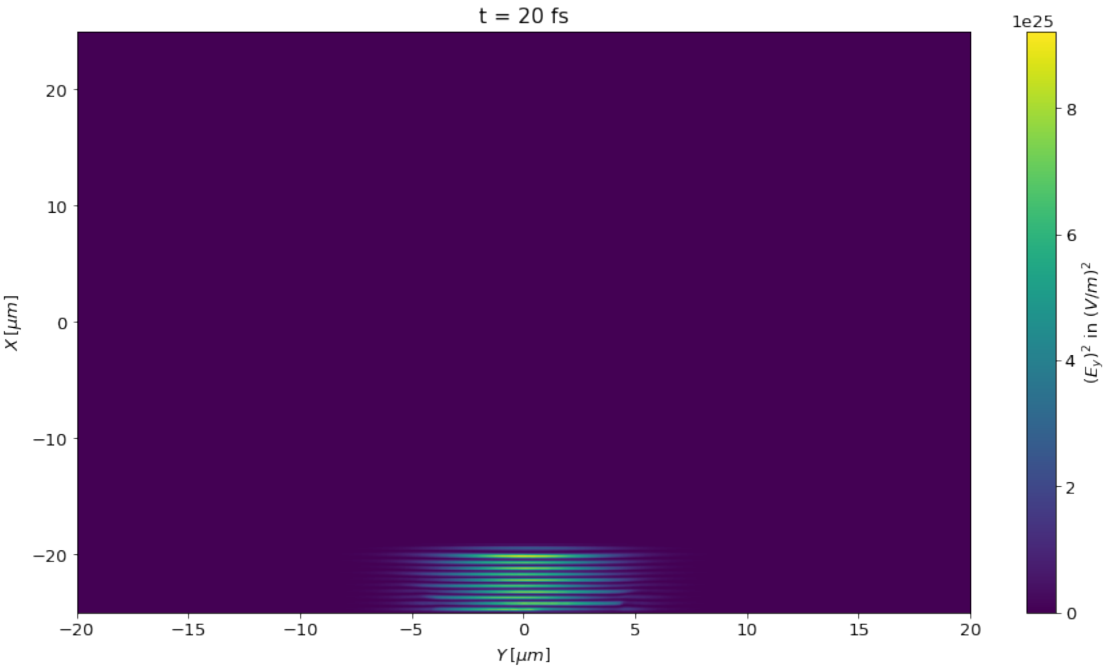
[ ]:
```
def plot_with_scale(data_dir, component="x"):
    raw_data = sdf.read(data_dir)
    comp = {
        "x":raw_data.Electric_Field_Ex,
        "y":raw_data.Electric_Field_Ey,
        "z":raw_data.Electric_Field_Ez,
    }
    field = comp[component]

    t = raw_data.Header['time']*1e15
    c_label = f"$(E_{component})^2$ in $({field.units})^2$"

    plt.figure(figsize=(15,8))
    plt.imshow(field.data**2, cmap='viridis', origin='lower',
    extent=EXTENT,
    aspect='auto',
    )
    plt.xlabel('$Y \, [\mu m]$')
    plt.ylabel('$X \, [\mu m]$')
    plt.title(f't = {t:.0f} fs')
    cbar = plt.colorbar()
    cbar.ax.set_ylabel(c_label)
```

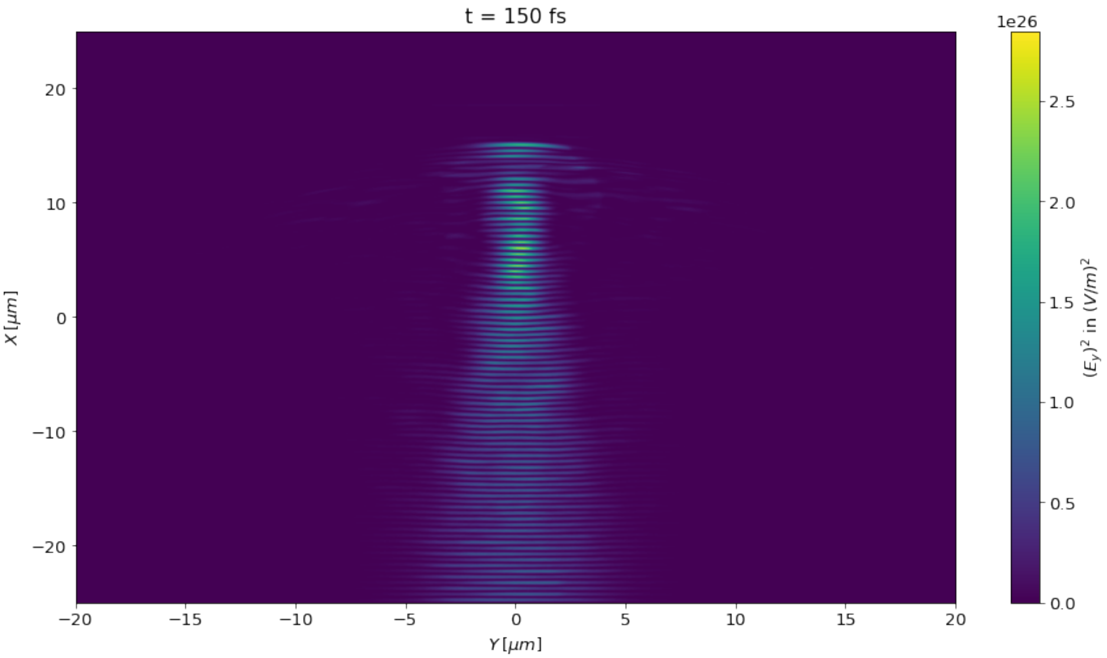
[ ]:

```
plot_with_scale("0002.sdf", component="y")
```

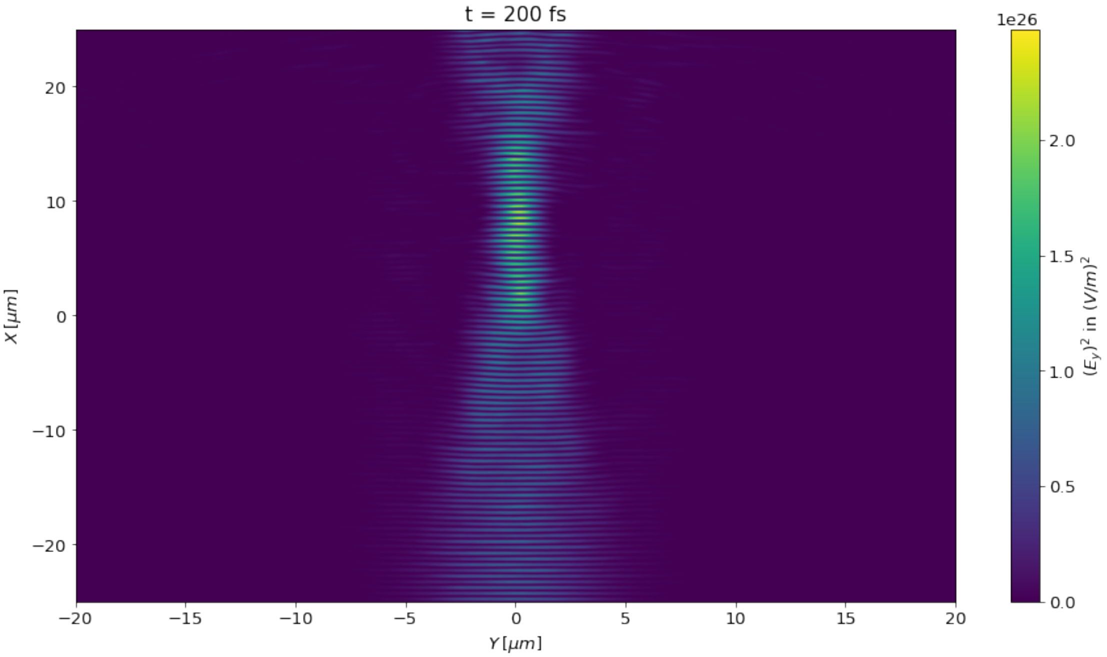


[ ]:

```
plot_with_scale("0015.sdf", component="y")
```



```
[ ]: plot_with_scale("0020.sdf", component="y")
```



5 Charge Density

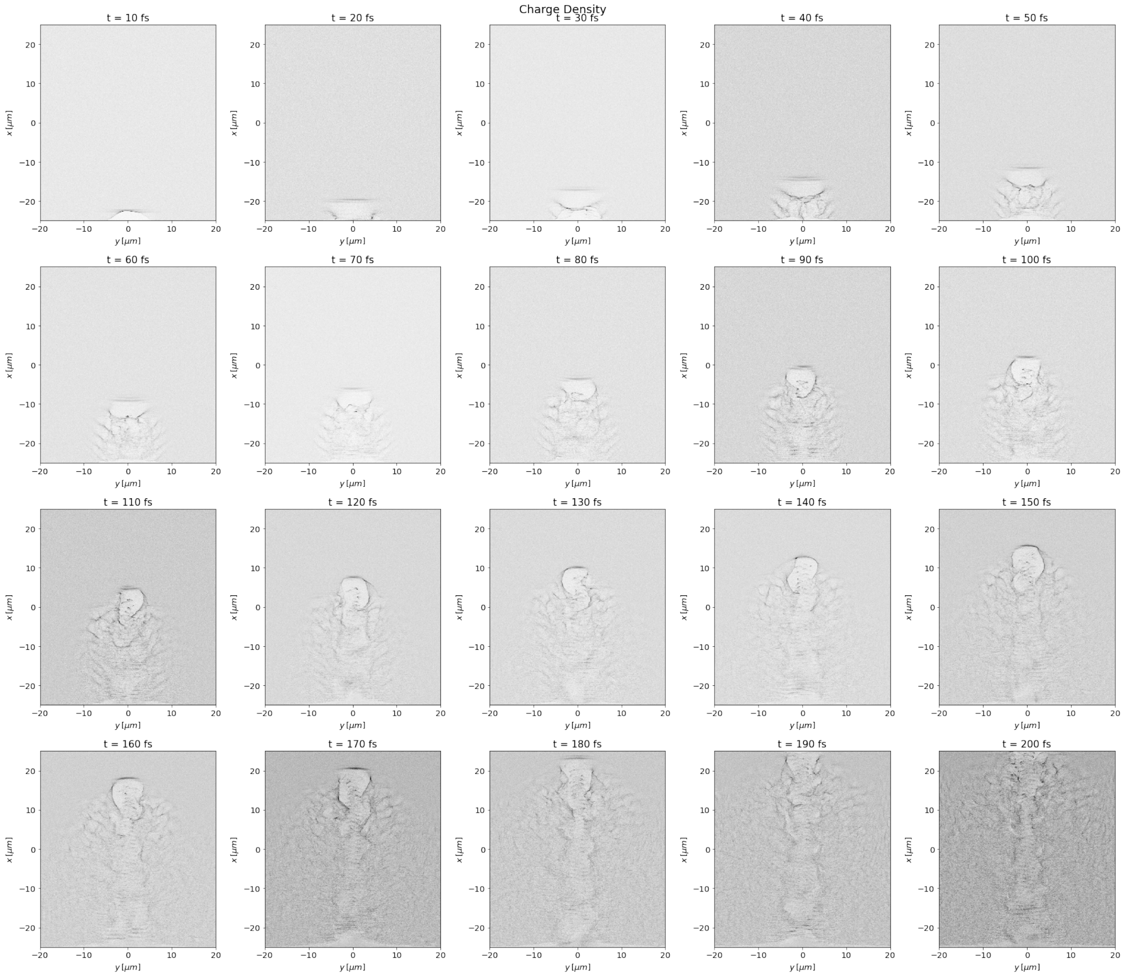
[ ]:

```
def plot_charge(data_dir, ax):
    raw_data = sdf.read(data_dir)

    variable = raw_data.Derived_Charge_Density.data
    t = raw_data.Header['time']*1e15
    ax.imshow(variable, cmap='gray', origin='lower',
    extent=EXTENT,
    aspect='auto',
    interpolation='nearest',
    # norm=colors.Normalize(vmin=-0.06, vmax=-0.002),
    )
    ax.set_xlabel('$y \, [\mu m]$')
    ax.set_ylabel('$x \, [\mu m]$')
    ax.set_title(f't = {t:.0f} fs')
```

[ ]:

```
fig, ax = plt.subplots(4, 5, figsize=(25, 22))
i=0
fig.suptitle("Charge Density", fontsize=18)
i=0
for data_dir in all_files[1:]:
    t = i*10
    plot_charge(data_dir, ax[i // 5, i % 5])
    i += 1
fig.tight_layout()
```


Charge Density
t = 10 fs
t = 20 fs
t = 30 fs
t = 40 fs
t = 50 fs
t = 60 fs
t = 70 fs
t = 80 fs
t = 90 fs
t = 100 fs
t = 110 fs
t = 120 fs
t = 130 fs
t = 140 fs
t = 150 fs
t = 160 fs
t = 170 fs
t = 180 fs
t = 190 fs
t = 200 fs
$x\,[\mu m]$
$y\,[\mu m]$
−20
−10
0
10
20