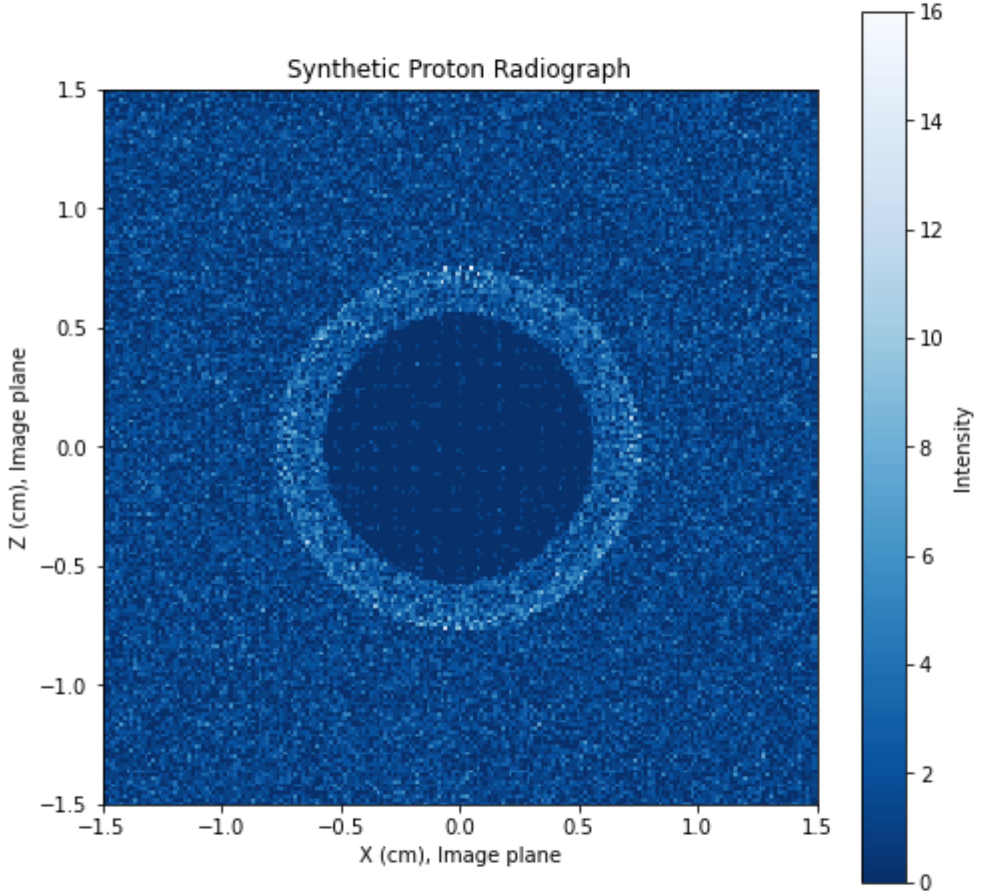
Charged Particle Radiography with PlasmaPy

plasmapy



University of Rochester
Laboratory for Laser Energetics

Peter Heuer
6/23/2021

My Background

University of Rochester
B.S. Physics, 2010-2014

- **Undergraduate research in quantum optics**

University of California Los Angeles
Ph.D. Physics, 2014-2020

- **Transitioned to plasma physics because I was (and am!) excited about fusion energy.**
- **Conducted lab experiments on the formation of astrophysical shocks.**

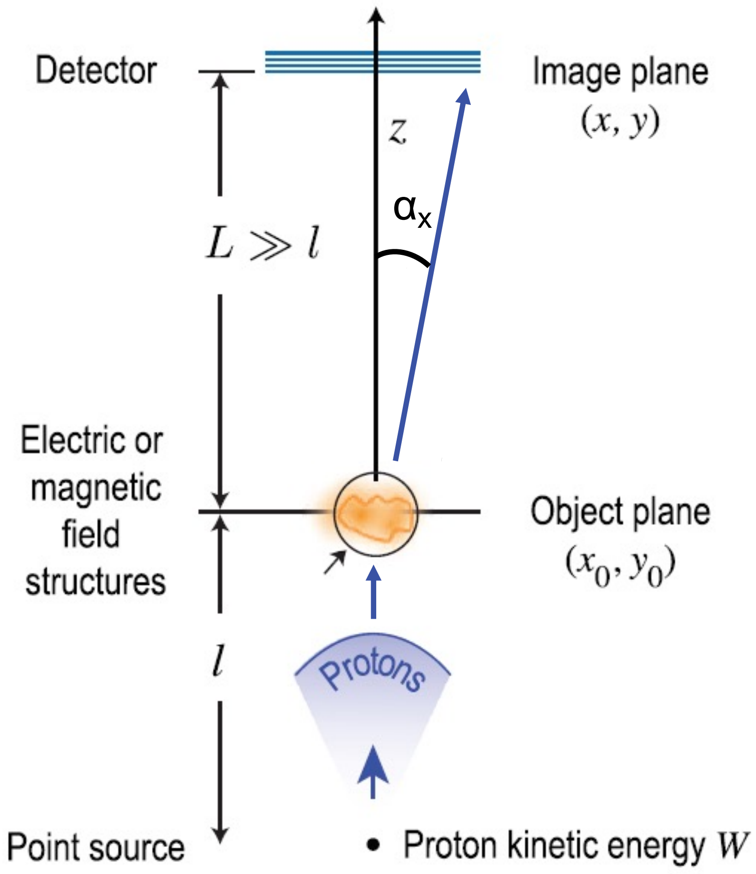
University of Rochester Laboratory for Laser Energetics
Assistant Scientist, 2020-present

- **Conducting experiments studying the influence of magnetic fields in high energy density plasmas for inertial fusion energy, lab astrophysics, and fundamental physics.**

Charged particle radiography and PlasmaPy

- **Charged particle radiography is an increasingly common diagnostic for E&M fields in high density plasmas.**
- **For experiments with non-linear deflections, synthetic radiography is an important analysis tool.**
- **The PlasmaPy charged particle radiography module provides an open-source toolkit for producing synthetic radiographs.**

What is charged particle radiography?



(Kugland et al. 2012 RSI)

Charged Particle Radiography is a technique for measuring electric and magnetic fields in high energy density plasmas

Electric and magnetic fields exert a force on the charged particles

$$\vec{F} = q\vec{E} + q\,\vec{v} \times \vec{B}$$

Leading to deflections by angles

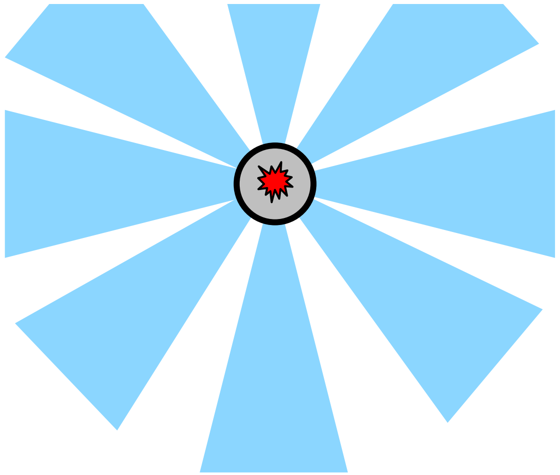
$$\alpha_x = \frac{e}{2W}\int E_x \cdot dy + \frac{e}{\sqrt{2m_p W}}\int B_z \cdot dy$$

$$\alpha_z = \frac{e}{2W}\int E_z \cdot dy - \frac{e}{\sqrt{2m_p W}}\int B_x \cdot dy$$

where m_p and W are the particle mass and kinetic energy.

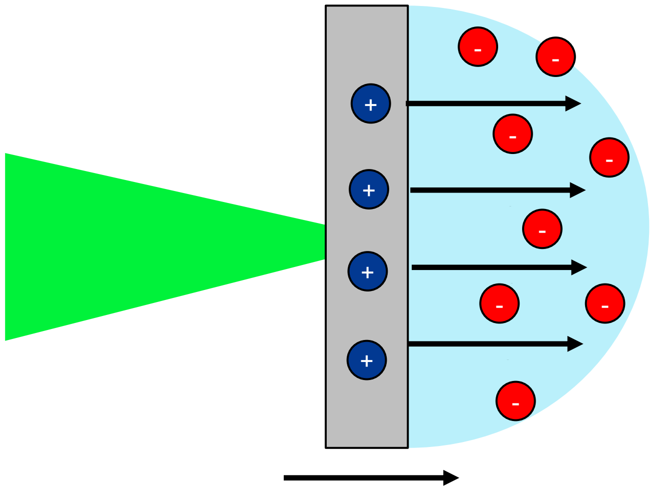
Proton sources for charged particle radiography

Imploding Capsules



An inertial confinement fusion implosion produces D-^{3}He and D-D fusion, resulting in two nearly mono-energetic populations of protons.

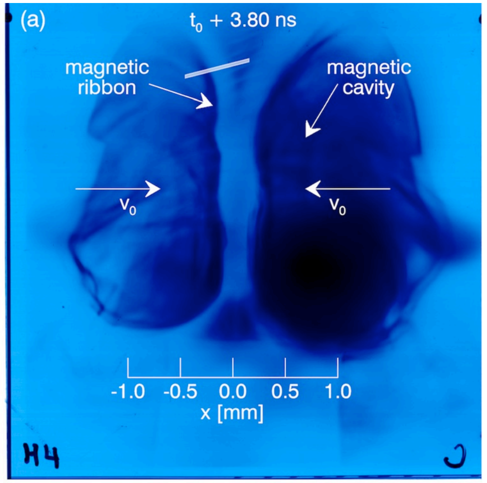
Target Normal Sheath Acceleration



A high-intensity laser produces a strong sheath field at a laser target, producing a broad spectrum of proton energies.

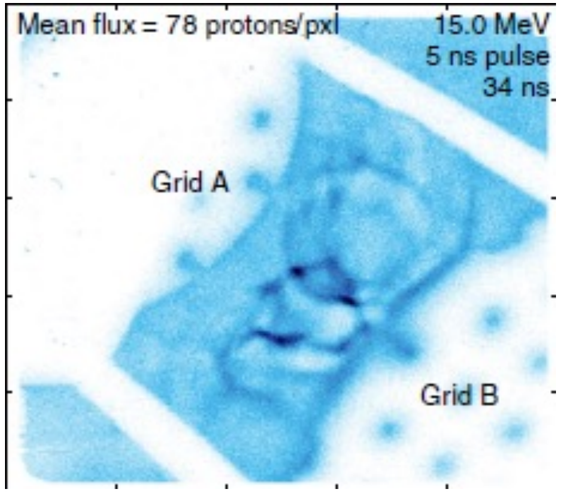
Examples of experimental proton radiographs

Schaeffer et al. 2017



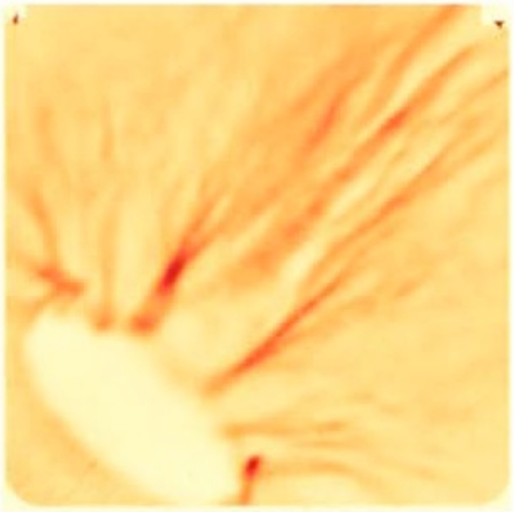
Collisionless Shocks

Tzeferacos et al. 2018



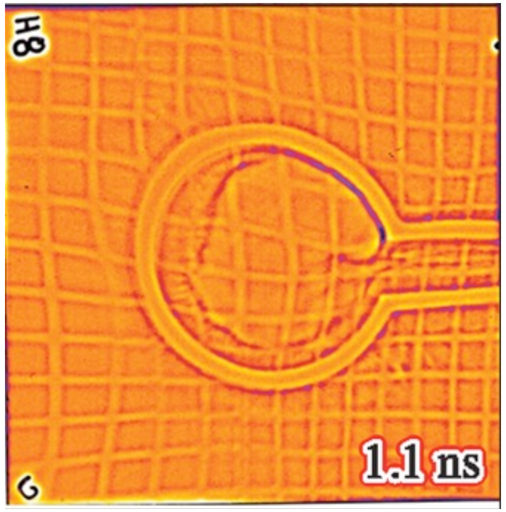
Turbulent Dynamo

Gao et al. 2019



Plasma Jets

Peebles et al. 2020



Laser Driven Coils

In most cases of interest, the particle trajectories are highly nonlinear!

Approaches to analysis

UR LLE

Direct Inversion*

For very small deflections, the line-integrated field can be directly calculated from the radiograph (at least, in theory).

Algorithmic Inversion**,†

For moderate deflections, algorithms can be used to find a line-integrated magnetic field that is consistent with the radiograph.

Synthetic Radiography

Simulations that track protons through simulated or assumed fields can create synthetic radiographs. This technique works even for nonlinear deflections!

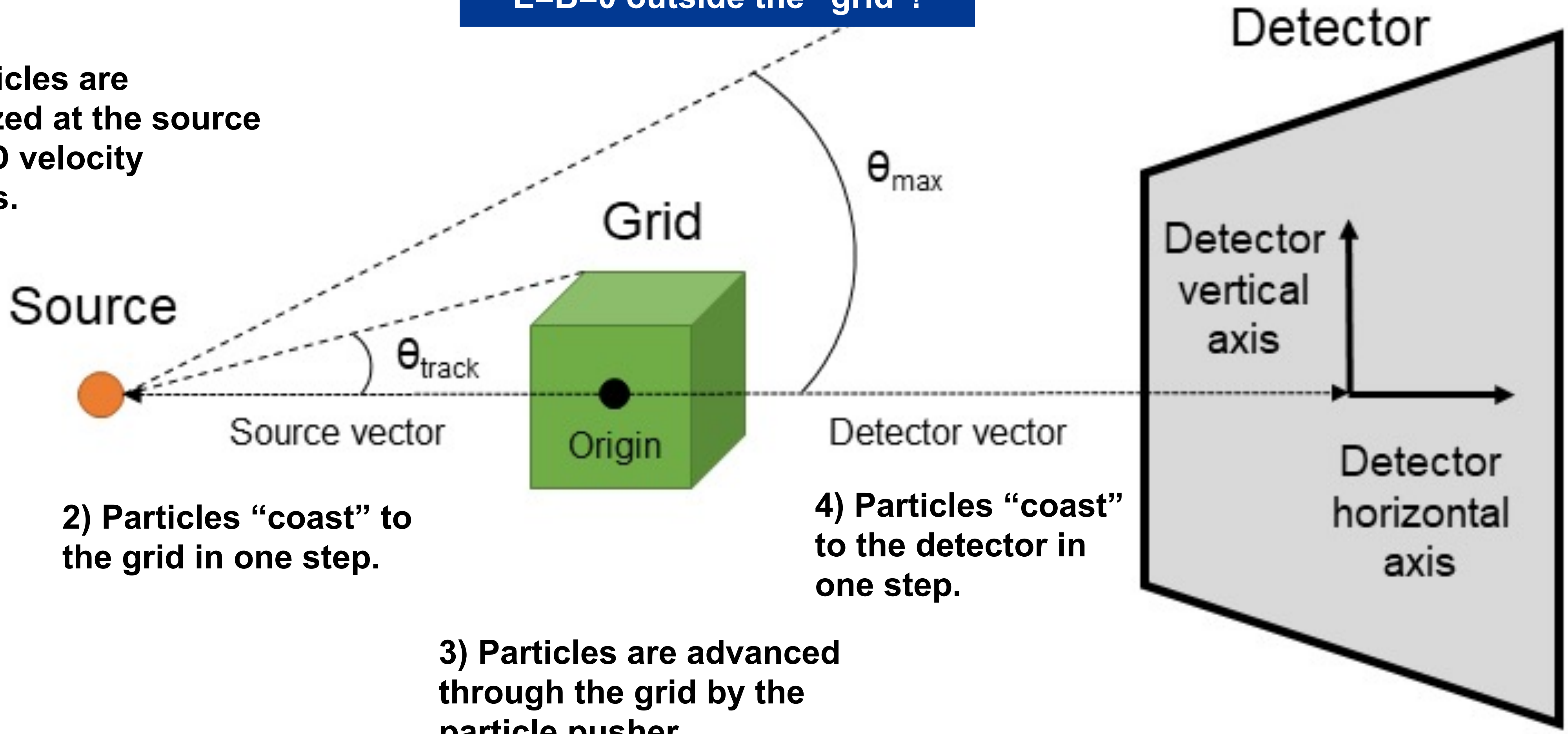
Synthetic radiography is the only technique that works on highly nonlinear radiographs!

* Kugland et al. 2012 RSI
** Bott et al. 2017 JPP
† Kasim et al. 2019 PRE

Creating synthetic radiographs from simulated or analytic fields

E=B=0 outside the "grid"!

1) Particles are initialized at the source with 3D velocity vectors.

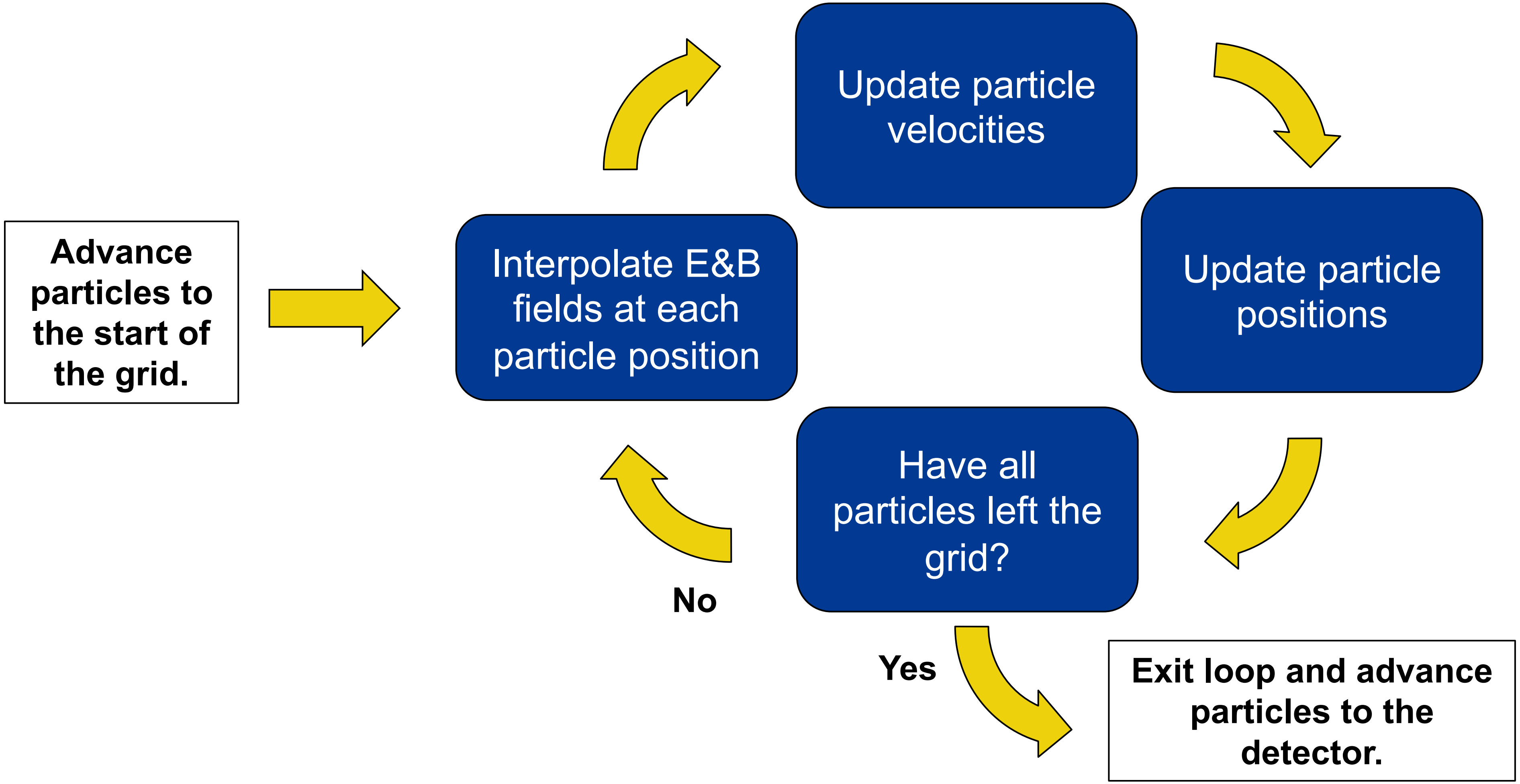


2) Particles "coast" to the grid in one step.

4) Particles "coast" to the detector in one step.

3) Particles are advanced through the grid by the particle pusher

The particle tracking loop



Advancing particles using the Boris push algorithm

The simplest particle advance algorithm is

$$\boldsymbol{v}^{+} = \boldsymbol{v}^{-} + \Delta t\,(q\boldsymbol{E} + \boldsymbol{v}^{-}\times\boldsymbol{B}\,)$$

but unfortunately this algorithm does not conserve energy!

The Boris push algorithm fixes this problem by centering the velocity calculation in between the timesteps (e.g. at –dt/2 and dt/2). It looks like this:

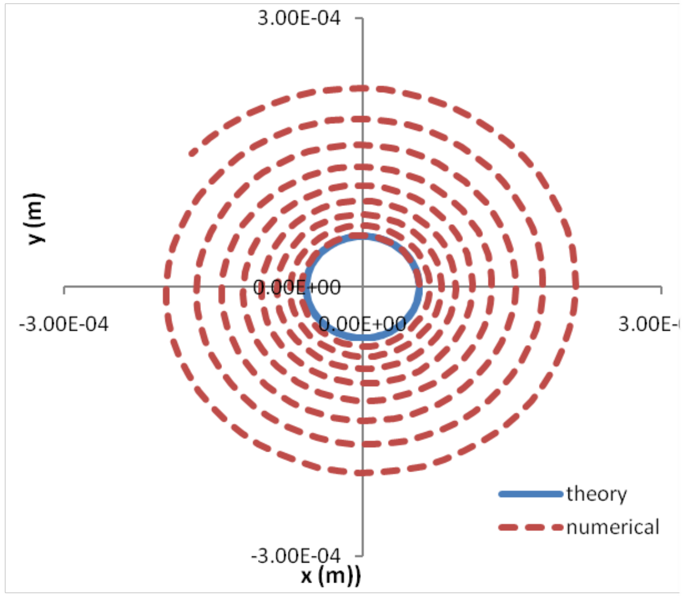
$$\mathbf{v}^{+} = \mathbf{v}^{-} + \mathbf{v}' \times \mathbf{s}$$

Where $\mathbf{v}' = \mathbf{v}^{-} + \mathbf{v}^{-} \times \mathbf{t}$

$$\mathbf{s} = \frac{2\mathbf{t}}{1+t^2} \qquad t = \frac{q\vec{B}}{m}\frac{\Delta t}{2}$$

The Boris algorithm is already implemented in PlasmaPy!

Simple Push



Boris Push

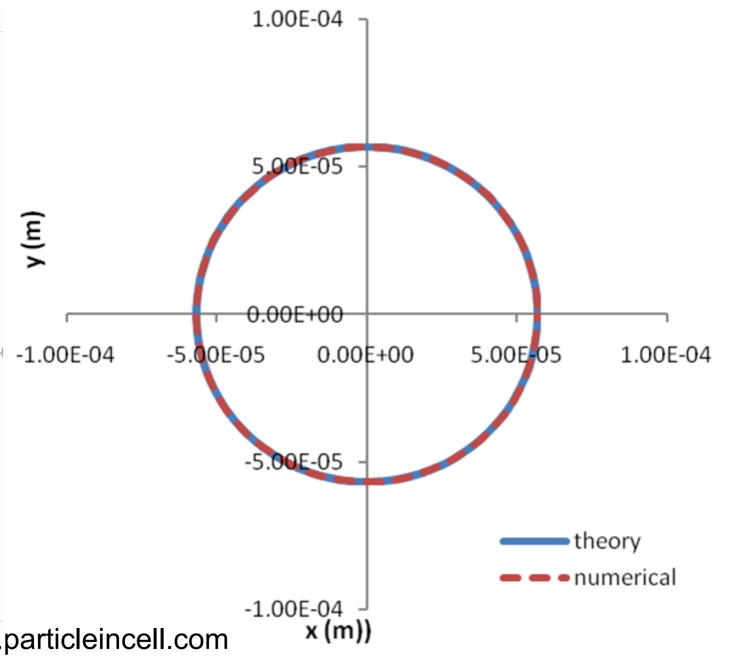
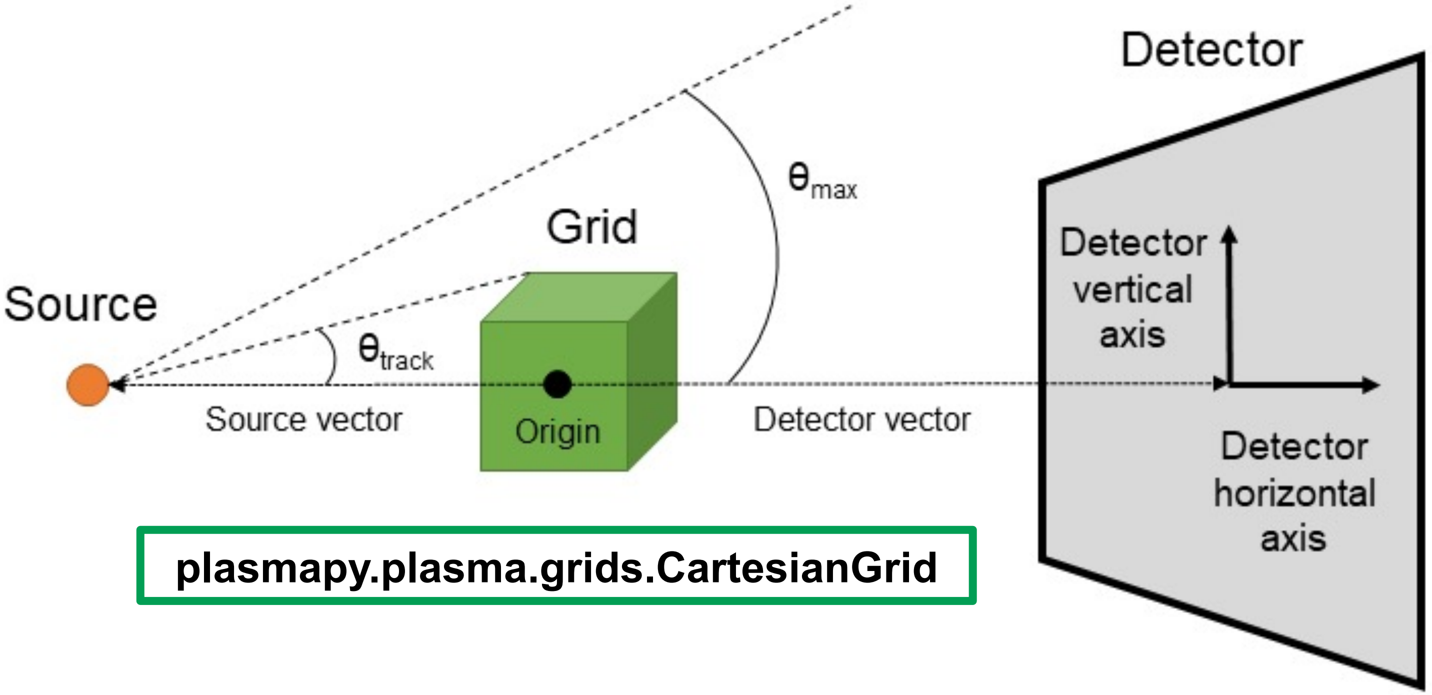


Image credit: www.particleincell.com

The Boris push algorithm conserves energy

* Birdsall & Langdon *Plasma Physics via Computer Simulation* 2004
** https://www.particleincell.com/2011/vxb-rotation/

Creating a Synthetic Radiograph with PlasmaPy

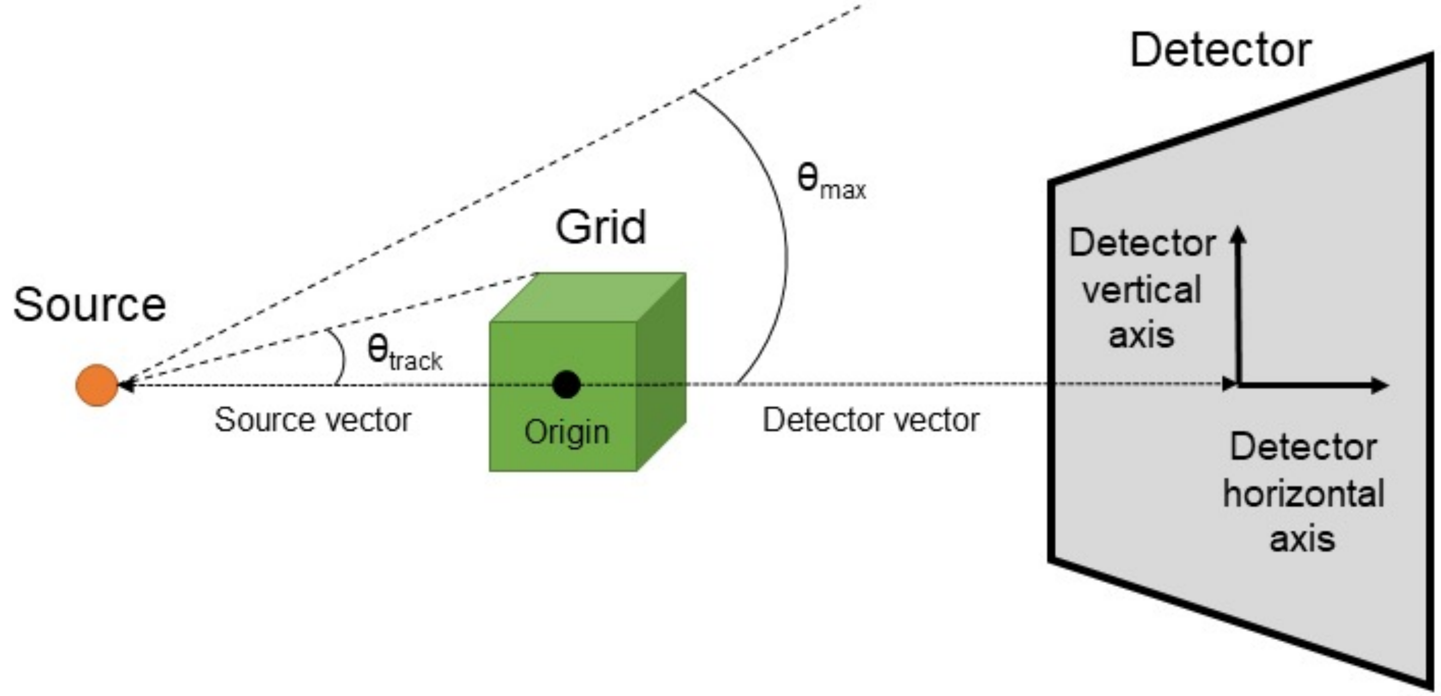


plasmapy.plasma.grids.CartesianGrid

```
source = (0 * u.mm, -10 * u.mm, 0 * u.mm)
detector = (0 * u.mm, 100 * u.mm, 0 * u.mm)

sim = prad.SyntheticProtonRadiograph(grid, source, detector, verbose=True)
```

Creating a Synthetic Radiograph with PlasmaPy




```
sim.create_particles(1e5, 3 * u.MeV, max_theta=np.pi / 15 * u.rad, particle="p")
```

Creating a Synthetic Radiograph with PlasmaPy

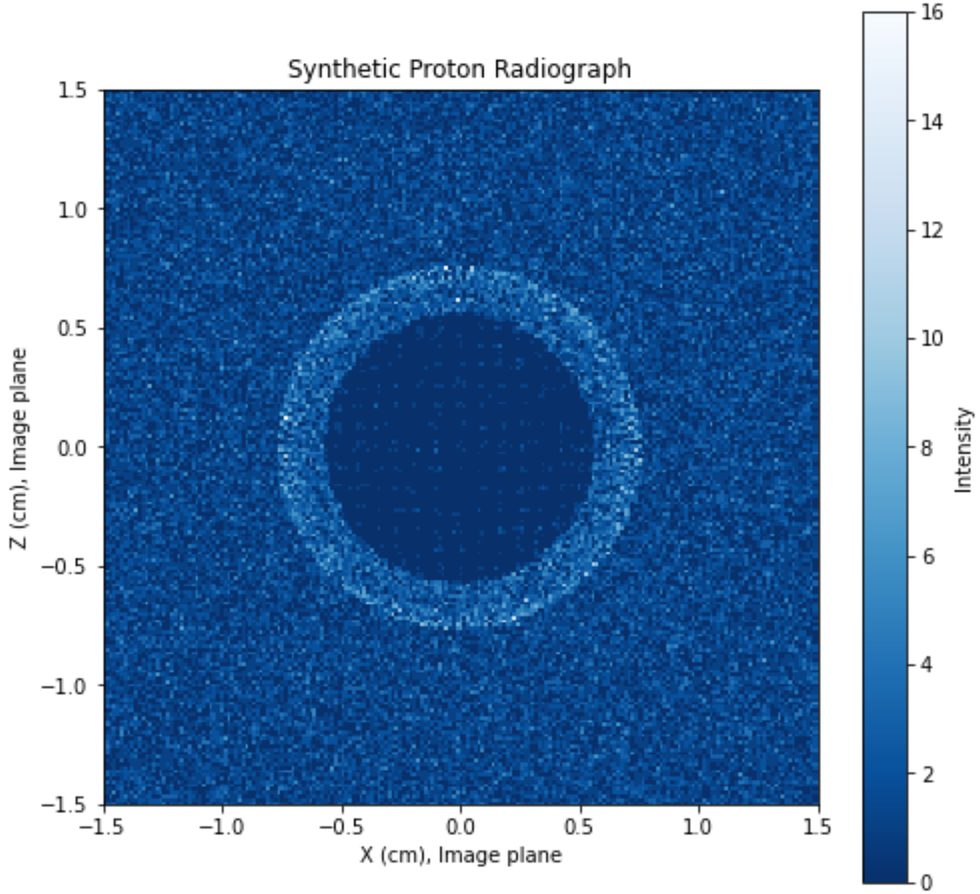
```
sim.run()
```

```
Particles on grid:   0%|          1.8e+02/5.6e+04 particles
Run completed
Fraction of particles tracked: 55.5%
Fraction of tracked particles that entered the grid: 64.4%
Fraction of tracked particles deflected away from the detector plane: 0.0%
```

Creating a Synthetic Radiograph with PlasmaPy

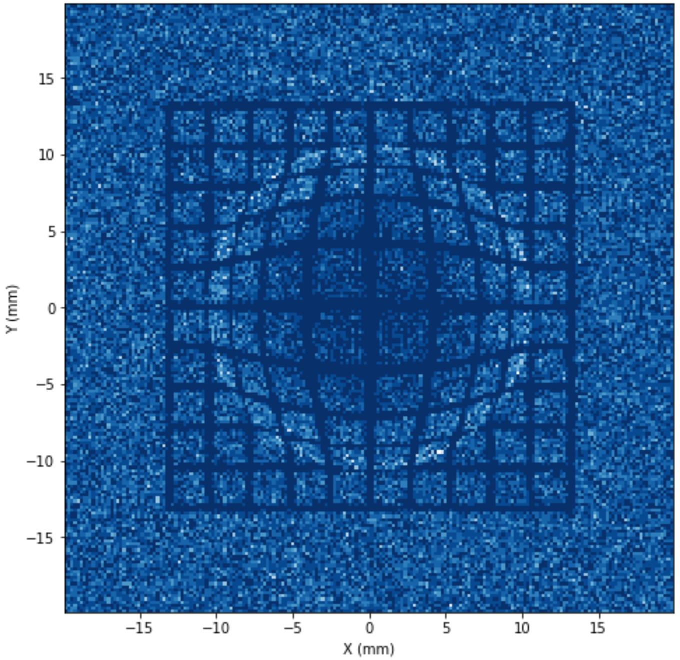
```
# A function to reduce repetative plotting
def plot_radiograph(hax, vax, intensity):
    fig, ax = plt.subplots(figsize=(8, 8))
    plot = ax.pcolormesh(
        hax.to(u.cm).value,
        vax.to(u.cm).value,
        intensity.T,
        cmap="Blues_r",
        shading="auto",
    )
    cb = fig.colorbar(plot)
    cb.ax.set_ylabel("Intensity")
    ax.set_aspect("equal")
    ax.set_xlabel("X (cm), Image plane")
    ax.set_ylabel("Z (cm), Image plane")
    ax.set_title("Synthetic Proton Radiograph")


size = np.array([[-1, 1], [-1, 1]]) * 1.5 * u.cm
bins = [200, 200]
hax, vax, intensity = sim.synthetic_radiograph(size=size, bins=bins)
plot_radiograph(hax, vax, intensity)
```



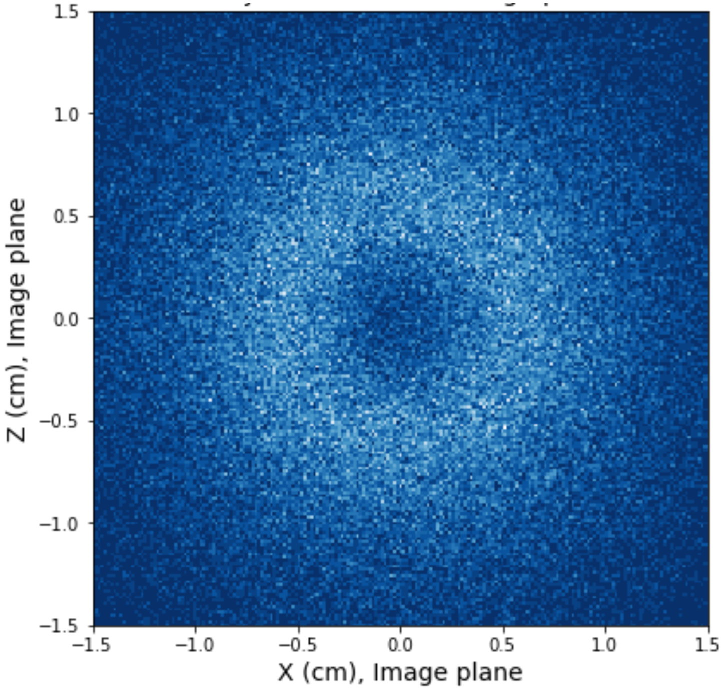
Advanced applications

Including a Fiducial Mesh




```
location = np.array([0, -2, 0]) * u.mm
extent = (1 * u.mm, 1 * u.mm)
nwires = (9, 12)
wire_diameter = 20 * u.um
sim.add_wire_mesh(location, extent, nwires, wire_diameter)
```

Custom Proton Source Profiles




```
sim.load_particles(pos, vel, particle=particle)
```

Charged particle radiography and PlasmaPy

UR LLE

- **Charged particle radiography is an increasingly common diagnostic for E&M fields in high density plasmas.**
- **For experiments with non-linear deflections, synthetic radiography is an important analysis tool.**
- **The PlasmaPy charged particle radiography module provides an open-source toolkit for producing synthetic radiographs.**

Please reach out if you have questions!

Peter Heuer
pheu@lle.rochester.edu