High energy physics numerical simulations with GPUs and ML

- Augustas Giedraitis
- Francisco Ponte
- Tiago Vistas

Supervisor:
Marco Leitão

Summary

- Why numerical methods?
- Why GPU?
- Modeling of the medium
- Modeling of the jet
- Structure and workflow
- Preliminary results
- Next steps

Why numerical methods?

- Simulate conditions that are hard to recreate in laboratory environment
- Approximate solutions to analytically hard (or impossible) problems
- Efficiently solve systems that are analytically tedious

Why GPU?

- ➔ Each grid point can be updated independently at each timestep;
- ➔ Enables simulation run faster:
  - ◆ Scaling to higher resolutions;
  - ◆ Scaling from 2D to higher dimensions;
  - ◆ Larger parameter scans.

Modelling of the medium

The medium in which the jet traverses is modeled by an energy density given by the following equation:

$$E(x) = \int d^D x \frac{1}{2}|\nabla A(x)|^2 + \frac{m_{th}}{2}A(x)^2 + \frac{g^2}{4!}A(x)^4$$

Where A(x) is the gluon field, g is the coupling constant, m_{th} the “thermal mass” and D the number of dimensions of the system.

The Algorithm

Hamiltonian Monte Carlo

1. Initialize the gluon field and the conjugate momentum with a random (gaussian) distribution
2. Calculate the Hamiltonian for the given sample
3. Update the state with the Hamilton equations with a time reversible integrator (Leapfrog)
4. Calculate the new Hamiltonian and accept the metropolis step with a probability of:

   $min\{1; e^{-\Delta H}\}$ and refresh the momentum sample
5. At the end of the Monte Carlo steps, calculate the Energy Density field ($E(x)$)

Modelling of the jet

The jet is modeled by the continuity equation that depends on the energy densities of both the jet and the medium:

$$\varepsilon, \epsilon : (t, \vec{x}) \to \mathbb{R}$$

$$\frac{\partial \varepsilon}{\partial t} + \nabla \cdot \vec{J} = S_{ext}$$

$$\vec{J} = \vec{v}\varepsilon$$

$$S_{ext} = -\gamma\varepsilon$$

$$\gamma = g\epsilon$$

Modelling of the jet

With some arithmetics we get the following equation for a infinitezimal step of the jet's energy density with a constant velocity:

$$d\mathcal{E} = -(g\epsilon\mathcal{E} + \vec{v}\nabla\varepsilon)dt$$

Structure

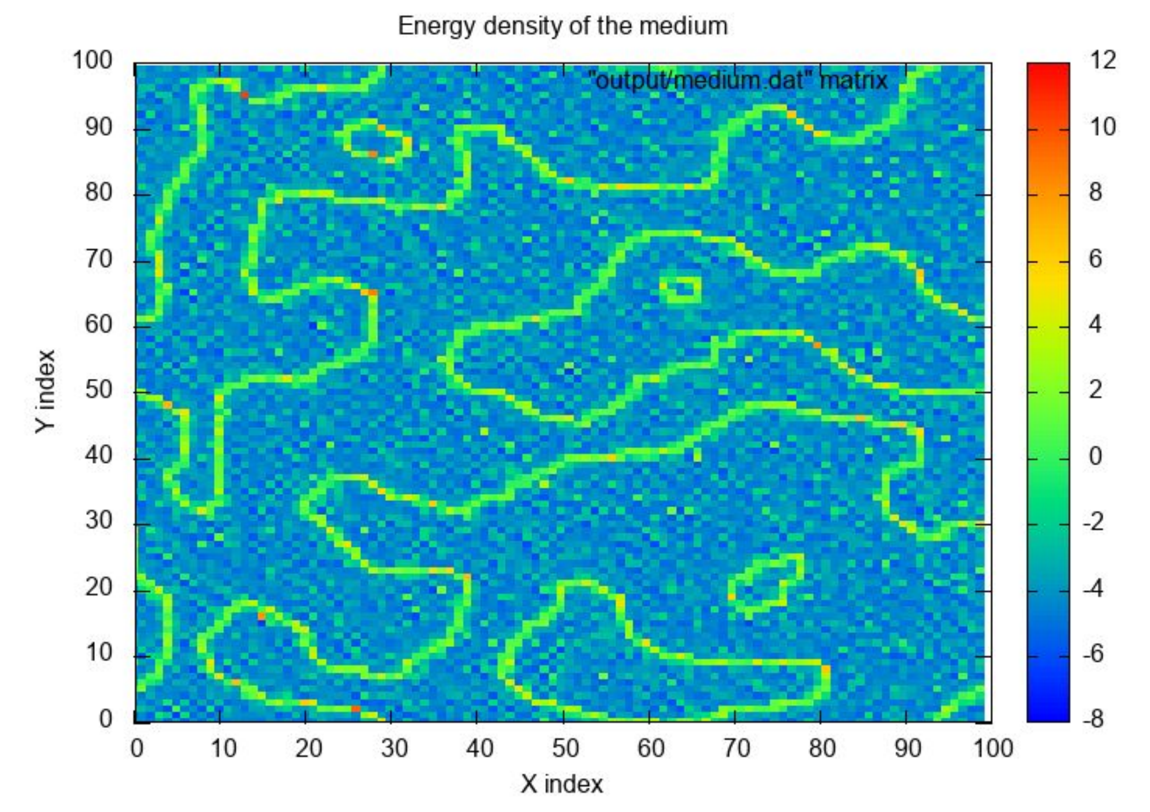
1. Done in C++ for its performance
2. Plasma model
    a. Class
    b. Encodes the properties of the plasma
3. Jet model
    a. Data structure
    b. Function
    c. Evolves in time through the medium
4. Modular design by separating medium and jet makes testing and extension easier

Workflow

1. Gnuplot
   a. Used to visualize the medium and the jet
   b. Generated snapshots at different time steps
   c. Built into an animation (GIF) to show time evolution
2. Git
   a. Tracked changes
   b. Coordinated contributions
   c. Ensured reproducibility.

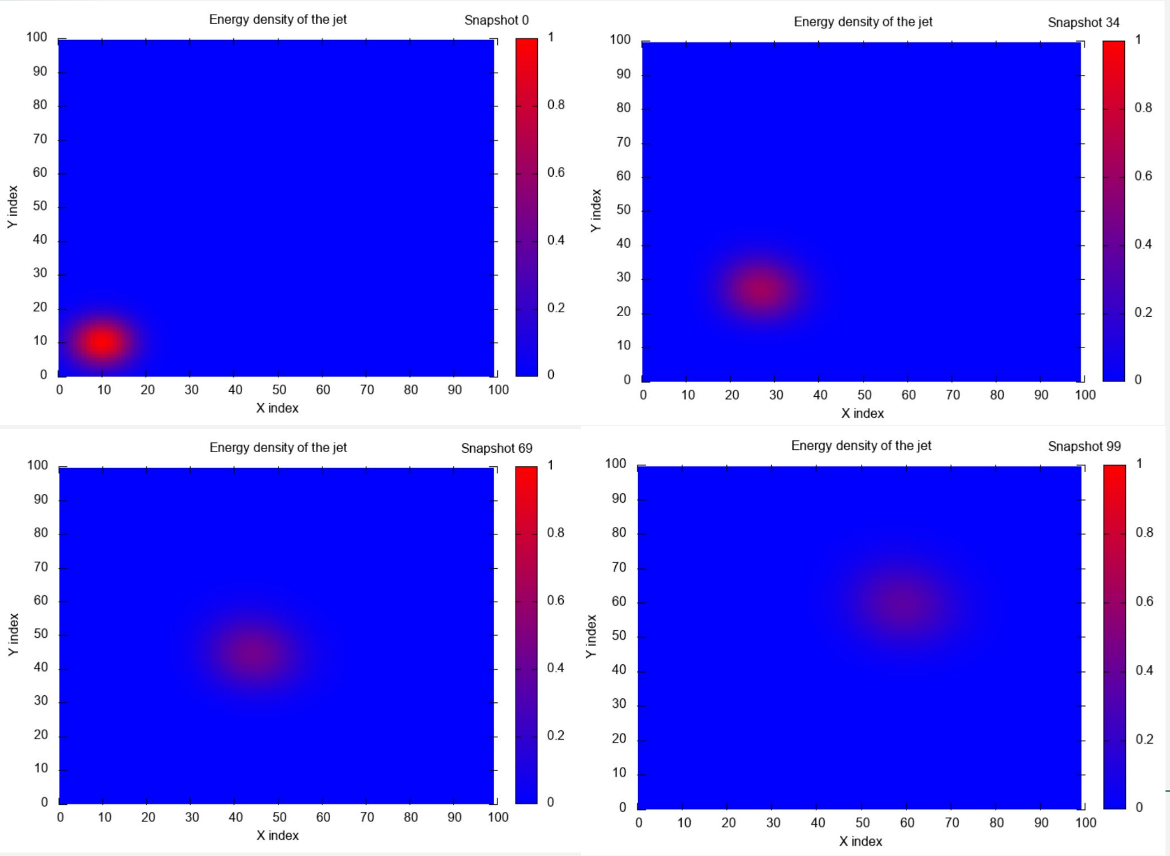
Preliminary results(1)

Medium



Preliminary results(2)

Jet



Preliminary results(3)

Performance

CPU

```
Medium creation time: 124370 ms
Jet initialization time: 0.2584 ms
Jet evolution time: 3142.09 ms
Snapshot creation time: 668.183 ms
```

Without parallelization:
2.07 min

GPU

```
Medium creation time: 14907.5 ms
Jet initialization time: 0.2216 ms
Jet evolution time: 3360.53 ms
Snapshot creation time: 1185.44 ms
```

With parallelization:
15 sec

Next steps

- ➔ Reduce memory and time for snapshots;
- ➔ Add ML surrogate model to approximate jet energy loss;
- ➔ Test the stability of our approach.