Differentiable Ray Tracing Simulator

Maxime Vandegar, Michael Kagan

January 2021

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

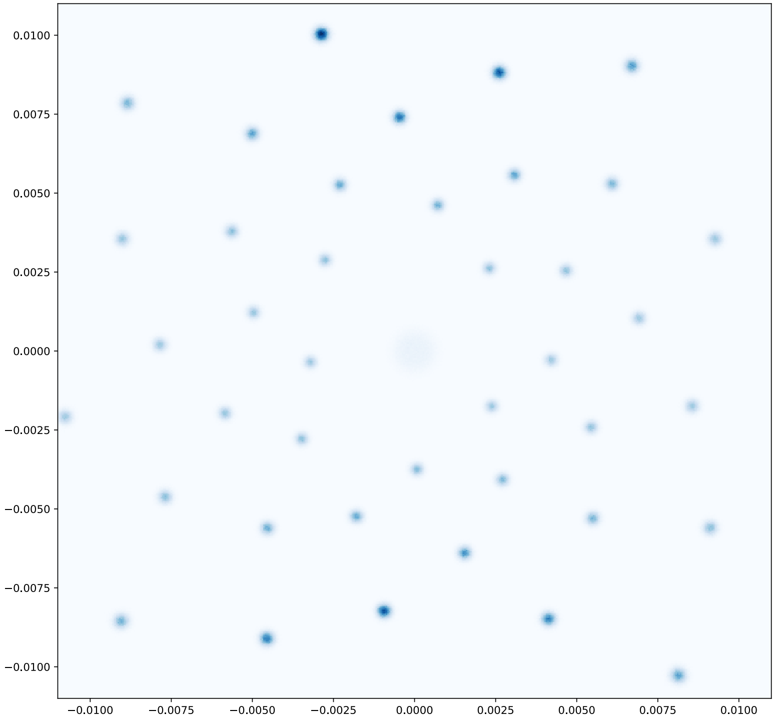
JAX

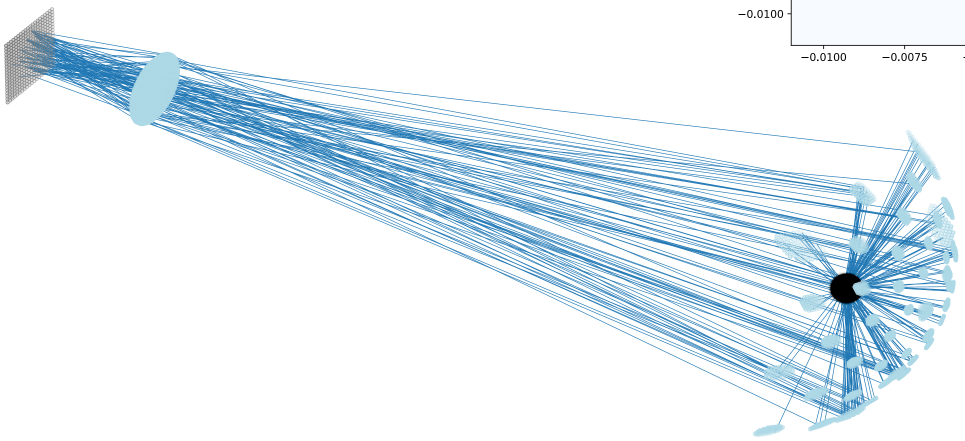
- Written in JAX (Autograd & XLA).
  - Autograd:
    - Automatically differentiates native Python and Numpy code.
    - Main purpose: gradient-based optimization.
  - XLA:
    - Compiles and runs programs on CPUs, GPUs and TPUs (fused operations).
- Functional programming paradigm.
- Numpy (Python library) like syntax.
- Automatic parallelization & vectorization.

- Differentiability could also be used for design optimization.

Preliminary results

- Imaging a cloud with a Gaussian density.
  - Tracing one ray at a time takes a few hours (1e6 rays).
  - It takes a few seconds after automatic vectorization.





Next steps

- Short-term goals:
  - Replace the gaussian density with a reallistic interference pattern.

$$N \times f_\gamma \times \left[1 + Cos\left(\frac{2\pi}{\lambda} x + \phi\right)\right] \times \frac{1}{\sigma}\, e^{-\frac{1}{2}\left[\frac{x-\mu}{\sigma}\right]^2}$$

  - Prepare the simulator for inference & 3D reconstruction.
    - Make sure that the gradients flow correctly.
    - Optimize the system so that images can be produced in a few seconds.