

HELMHOLTZ

# LASY (LAser manipulationS made easY)

Enabling realistic laser pulses in start-to-end simulations

K. Pöder<sup>1</sup>, R. J. Shalloo<sup>1</sup>, I. Andriyash<sup>2</sup>, L. Fedeli<sup>3</sup>, A. Ferran Pousa<sup>1</sup>, A. Huebl<sup>4</sup>, S. Jalas<sup>1</sup>, M. Kirchen<sup>1</sup>, R. Lehe<sup>4</sup>, A. Sinn<sup>1</sup>, J.-L. Vay<sup>4</sup> & M. Thévenet<sup>1</sup>

<sup>1</sup> Deutsches Elektronen Synchrotron (DESY)

<sup>2</sup> Laboratoire d'Optique Appliquée (LOA)

<sup>3</sup> Commissariat à l'énergie atomique et aux énergies alternatives (CEA),

<sup>4</sup> Lawrence Berkeley National Laboratory (LBNL)



# A modern simulation study combines multiple codes

Simulation studies gain from being:

## Multi-physics

- Hydrodynamics (HOFI, discharge, etc.)
- Beam dynamics, application

## Energy efficient

- Efficient codes for reduced model
- Combine the most appropriate tools

## Realistic

- Experimental laser profiles
- Experimental plasma/beam profiles

## Comprehensive

- Ensembles of simulations
- (Bayesian) Optimization

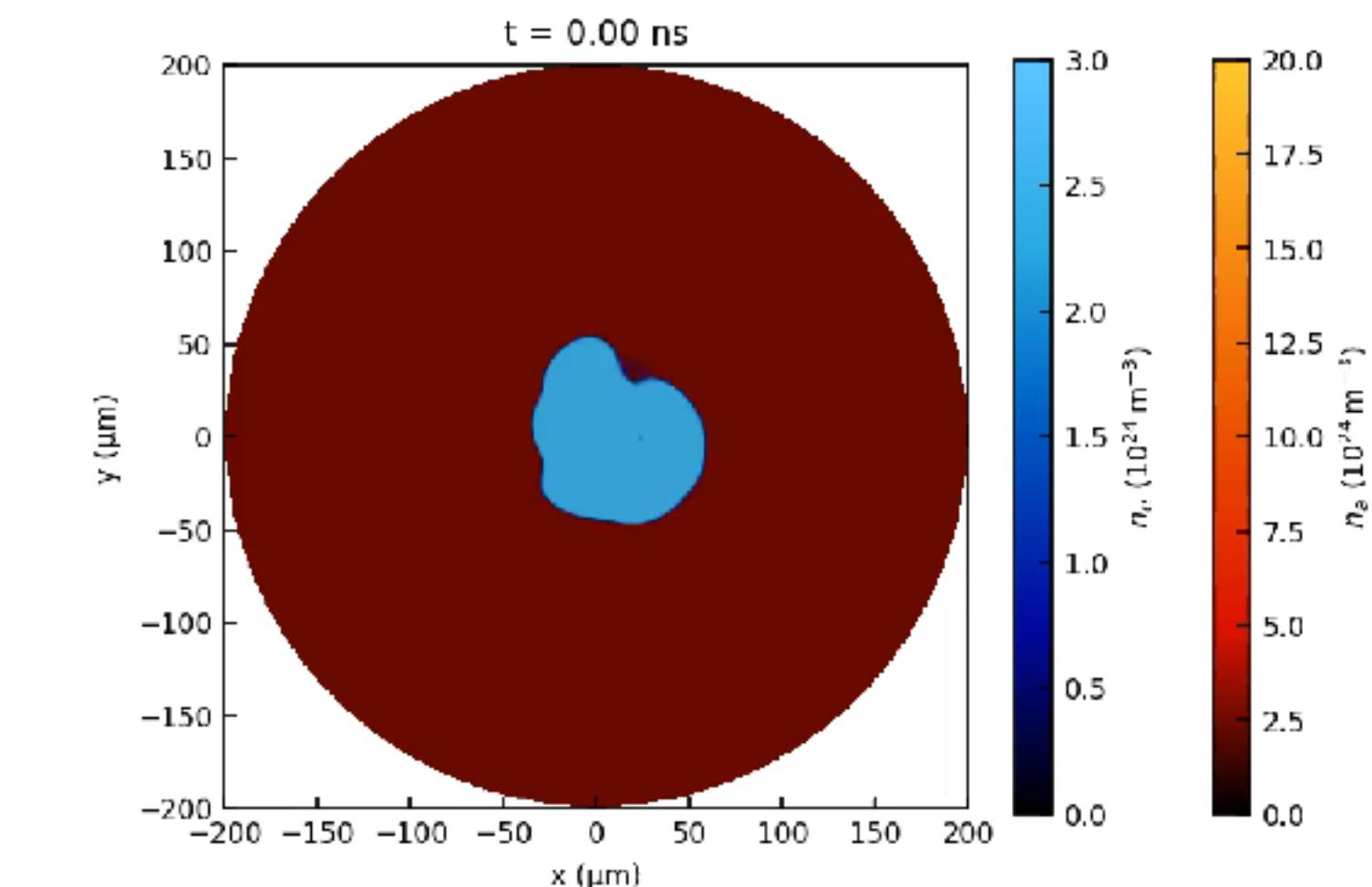
We contribute to this ecosystem

COMSOL-plasma

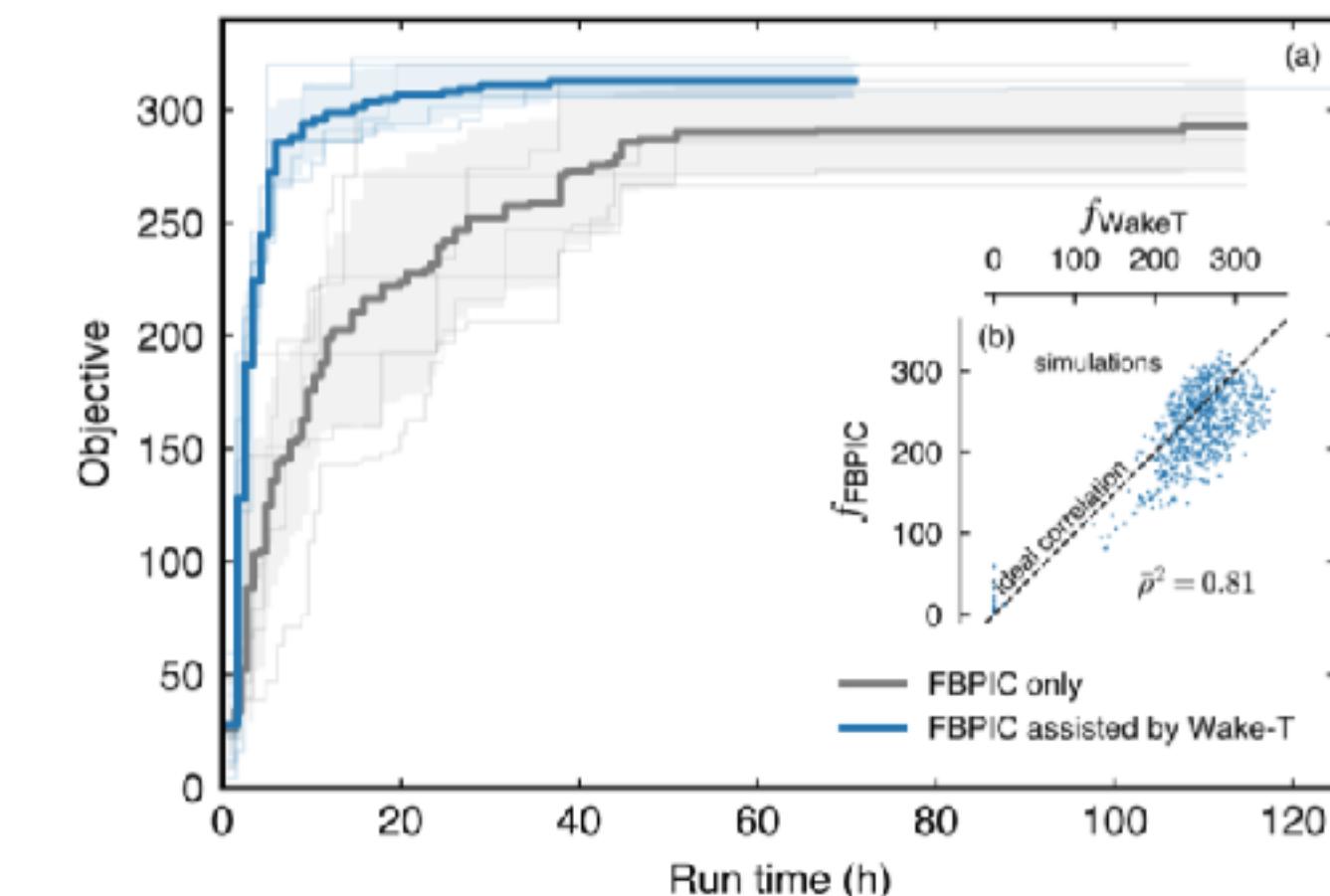
HiPACE++, Wake-T  
(see next slide)



Scalable optimization on experiments and simulations



Collaboration with UNIVERSITY OF OXFORD



Collaboration with BERKELEY LAB Argonne NATIONAL LABORATORY

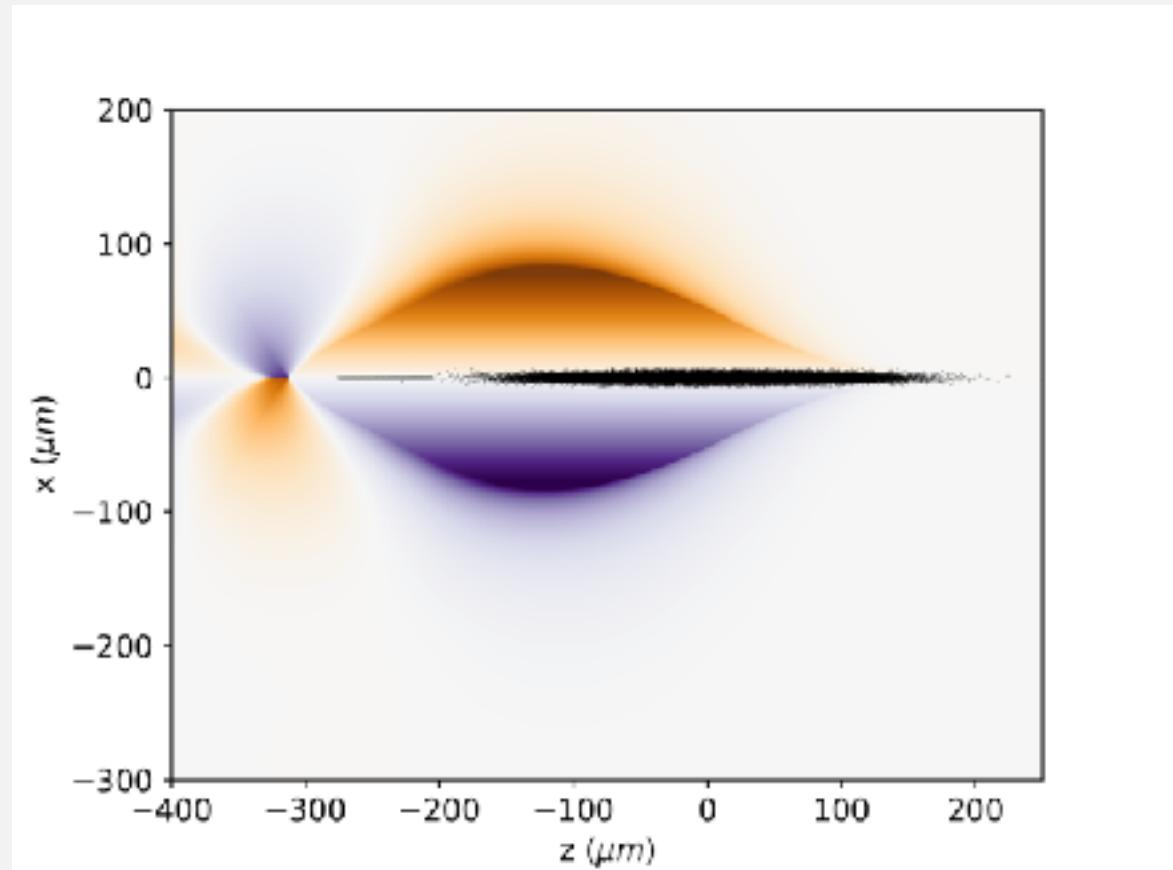
# Fully converged simulations with reduced models at modest cost

## HiPACE++

- Open-source, GPU-capable multi-physics 3D quasistatic particle-in-cell code (laser-driven or beam-driven), C++
- Collaboration



BERKELEY LAB



Fully converged (nm-scale resolution with mesh refinement) 3D simulations of a 20 GeV stage starting at 175 GeV for a 135 nm emittance beam w/ ion motion takes 30 min on 16 GPU-equipped nodes (Frontier has 9400) → small allocation allows thousands

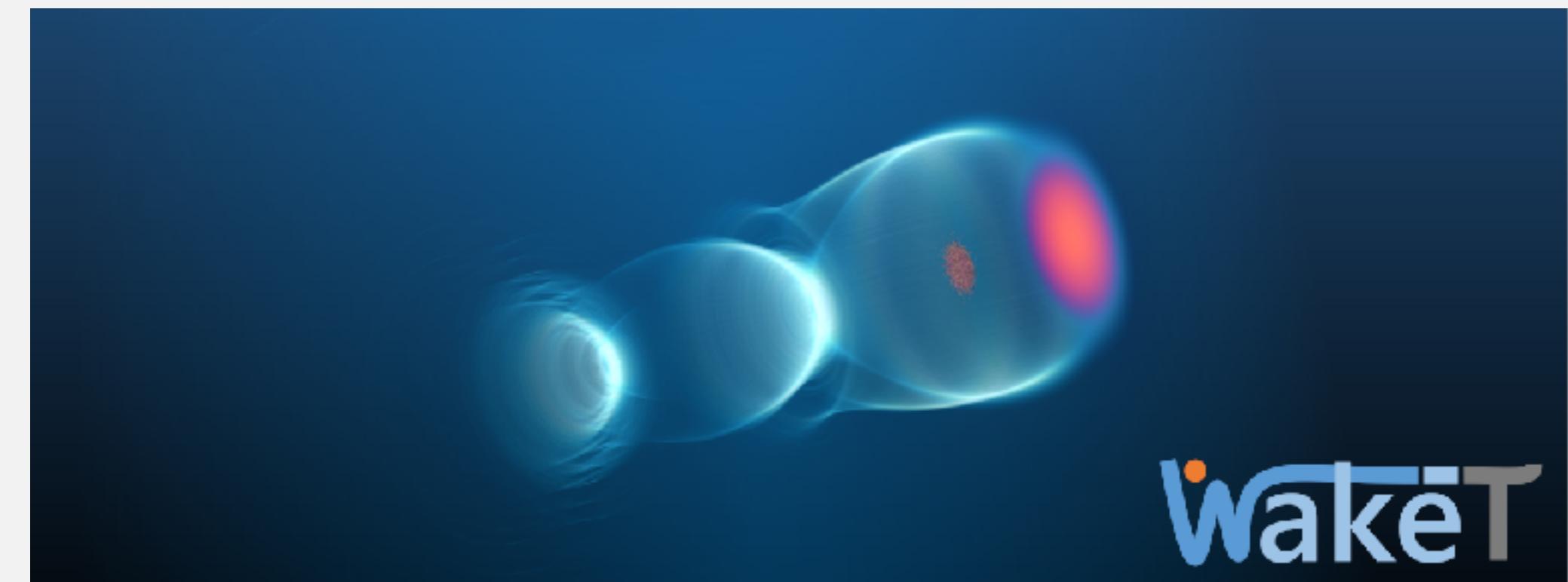
S. Diederichs et al., Comput. Phys. Comm. 278: 108421 (2022)

<https://agenda.infn.it/event/35577/contributions/208606>

<https://github.com/Hi-PACE/hipace>



- Open-source, 2D (axisymmetric) quasistatic code for (laser/beam-driven), incl. ion motion, Python



1 plasma stage takes seconds-to-minutes on a laptop, suitable for design studies. 20 stages + optimization to 150 GeV in < 1 hour

A. Ferran Pousa et al., J. Phys.: Conf. Ser. (2019)

A Ferran Pousa, et al. Proc. IPAC23, 1533

<https://github.com/AngelFP/Wake-T>



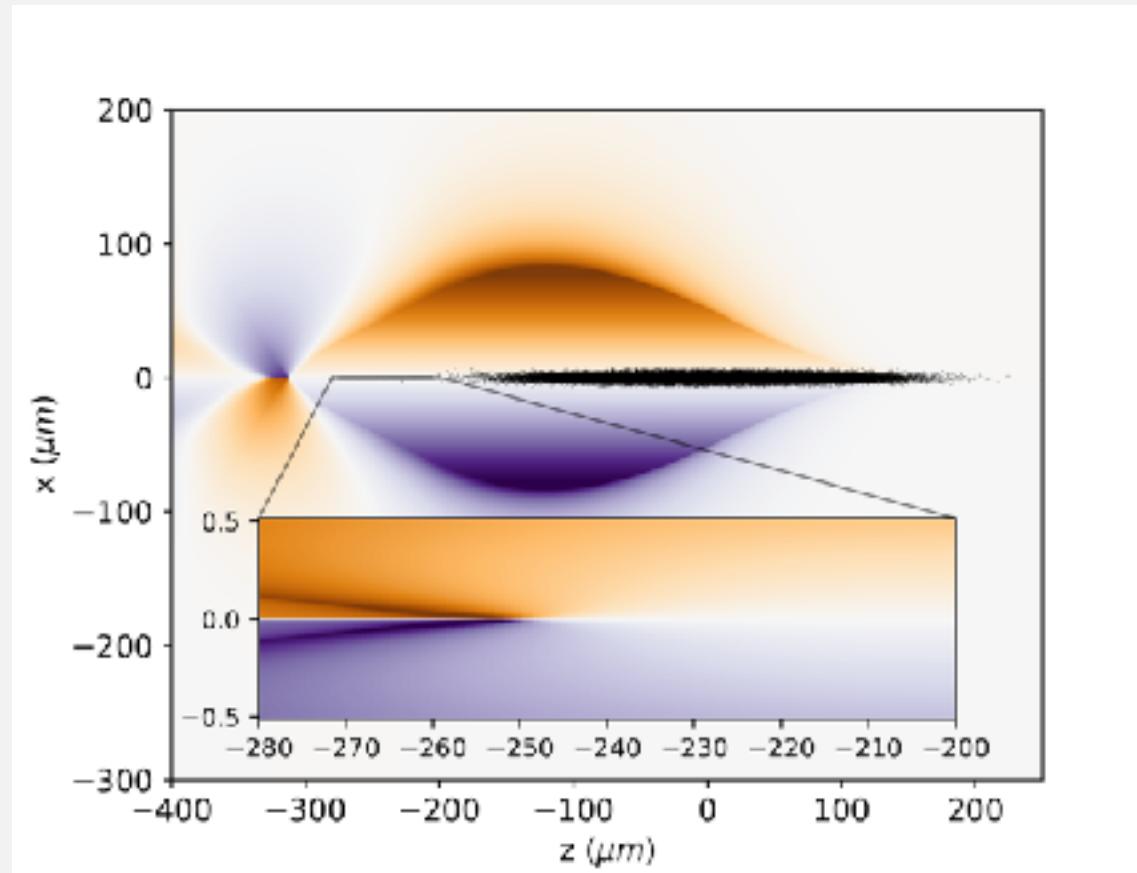
# Fully converged simulations with reduced models at modest cost

## HiPACE++

- Open-source, GPU-capable multi-physics 3D quasistatic particle-in-cell code (laser-driven or beam-driven), C++
- Collaboration



BERKELEY LAB



Fully converged (nm-scale resolution with mesh refinement) 3D simulations of a 20 GeV stage starting at 175 GeV for a 135 nm emittance beam w/ ion motion takes 30 min on 16 GPU-equipped nodes (Frontier has 9400) → small allocation allows thousands

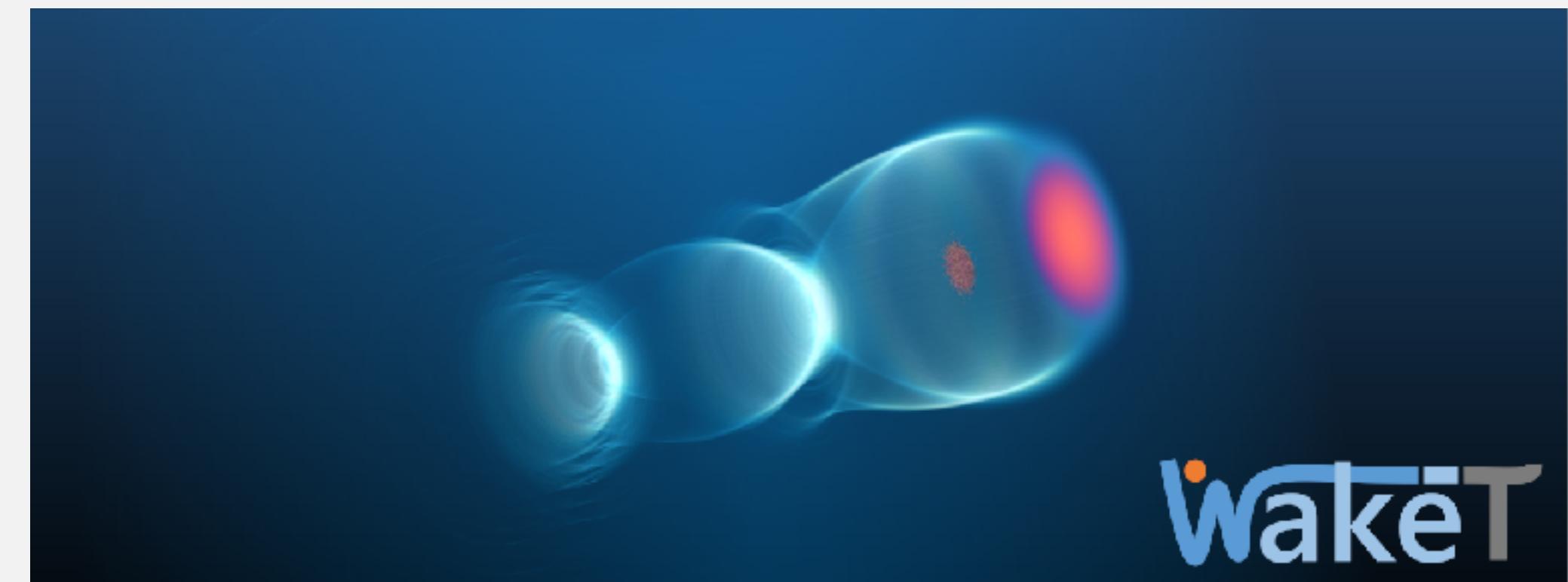
S. Diederichs et al., Comput. Phys. Comm. 278: 108421 (2022)

<https://agenda.infn.it/event/35577/contributions/208606>

<https://github.com/Hi-PACE/hipace>



- Open-source, 2D (axisymmetric) quasistatic code for (laser/beam-driven), incl. ion motion, Python



1 plasma stage takes seconds-to-minutes on a laptop, suitable for design studies. 20 stages + optimization to 150 GeV in < 1 hour

A. Ferran Pousa et al., J. Phys.: Conf. Ser. (2019)

A Ferran Pousa, et al. Proc. IPAC23, 1533

<https://github.com/AngelFP/Wake-T>



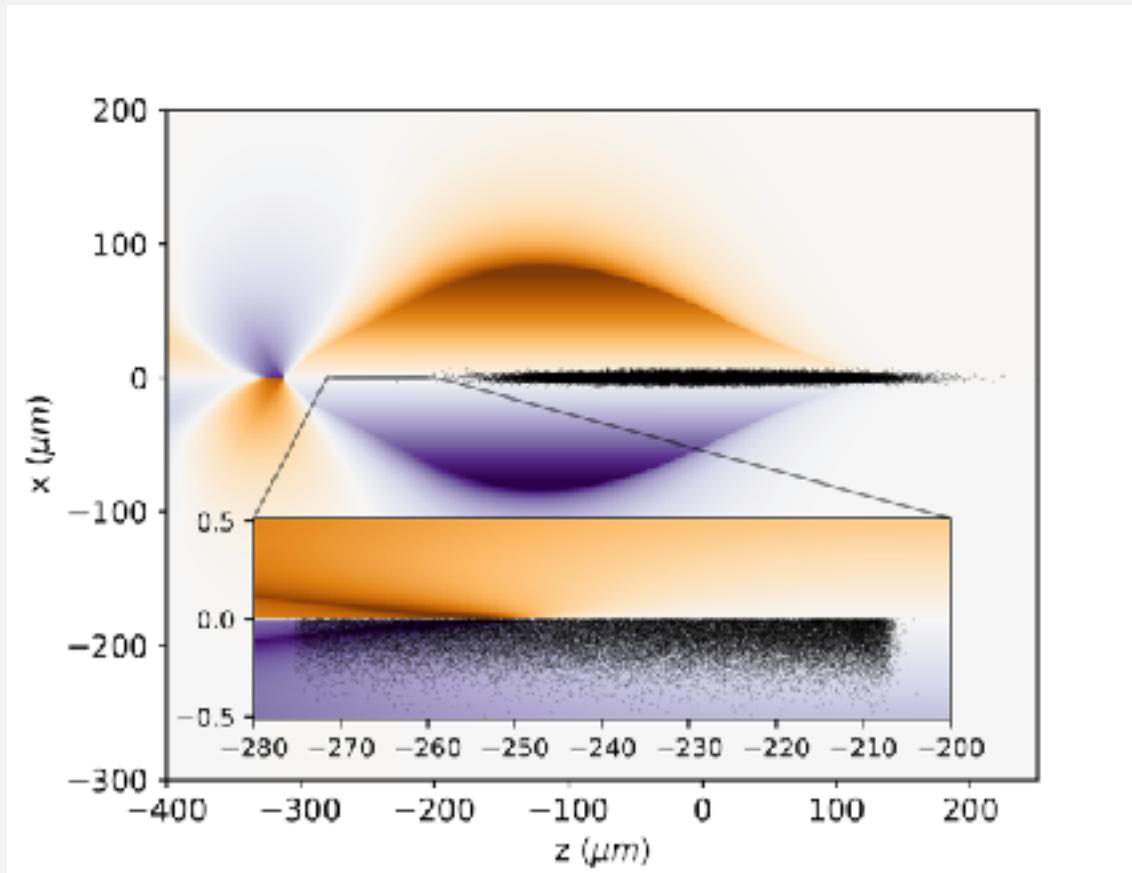
# Fully converged simulations with reduced models at modest cost

## HiPACE++

- Open-source, GPU-capable multi-physics 3D quasistatic particle-in-cell code (laser-driven or beam-driven), C++
- Collaboration



BERKELEY LAB



Fully converged (nm-scale resolution with mesh refinement) 3D simulations of a 20 GeV stage starting at 175 GeV for a 135 nm emittance beam w/ ion motion takes 30 min on 16 GPU-equipped nodes (Frontier has 9400) → small allocation allows thousands

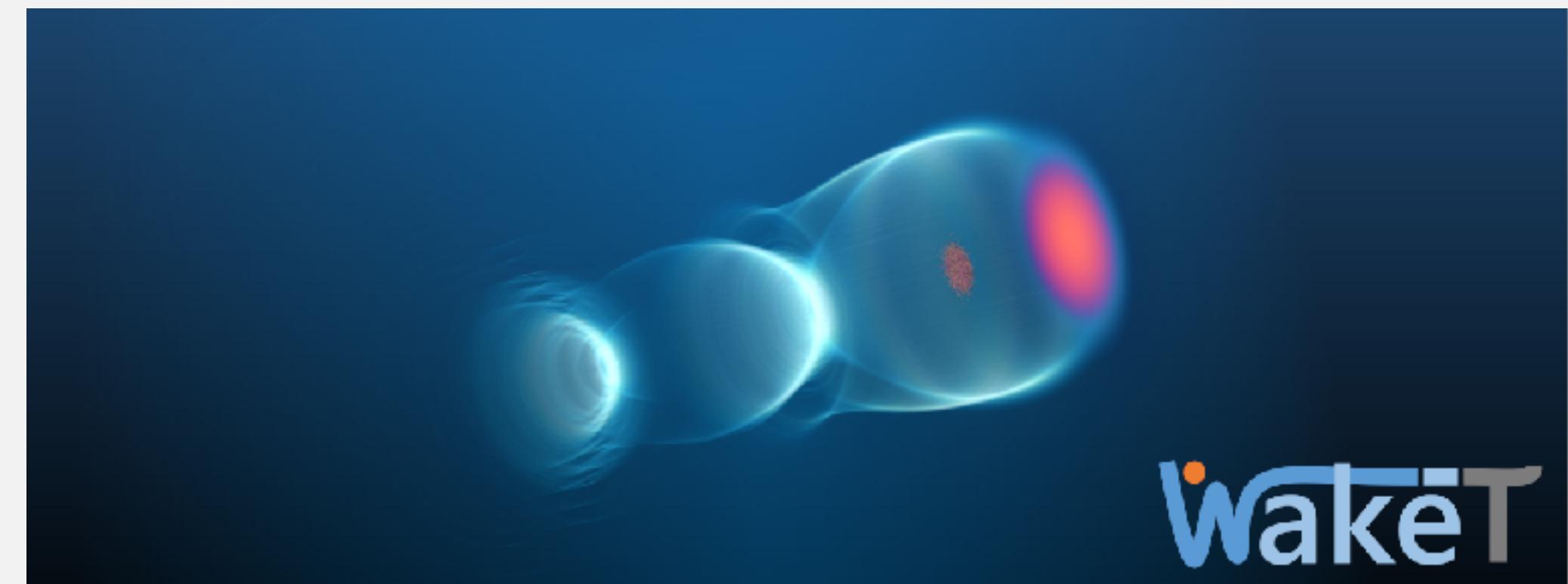
S. Diederichs et al., Comput. Phys. Comm. 278: 108421 (2022)

<https://agenda.infn.it/event/35577/contributions/208606>

<https://github.com/Hi-PACE/hipace>



- Open-source, 2D (axisymmetric) quasistatic code for (laser/beam-driven), incl. ion motion, Python



1 plasma stage takes seconds-to-minutes on a laptop, suitable for design studies. 20 stages + optimization to 150 GeV in < 1 hour

A. Ferran Pousa et al., J. Phys.: Conf. Ser. (2019)

A Ferran Pousa, et al. Proc. IPAC23, 1533

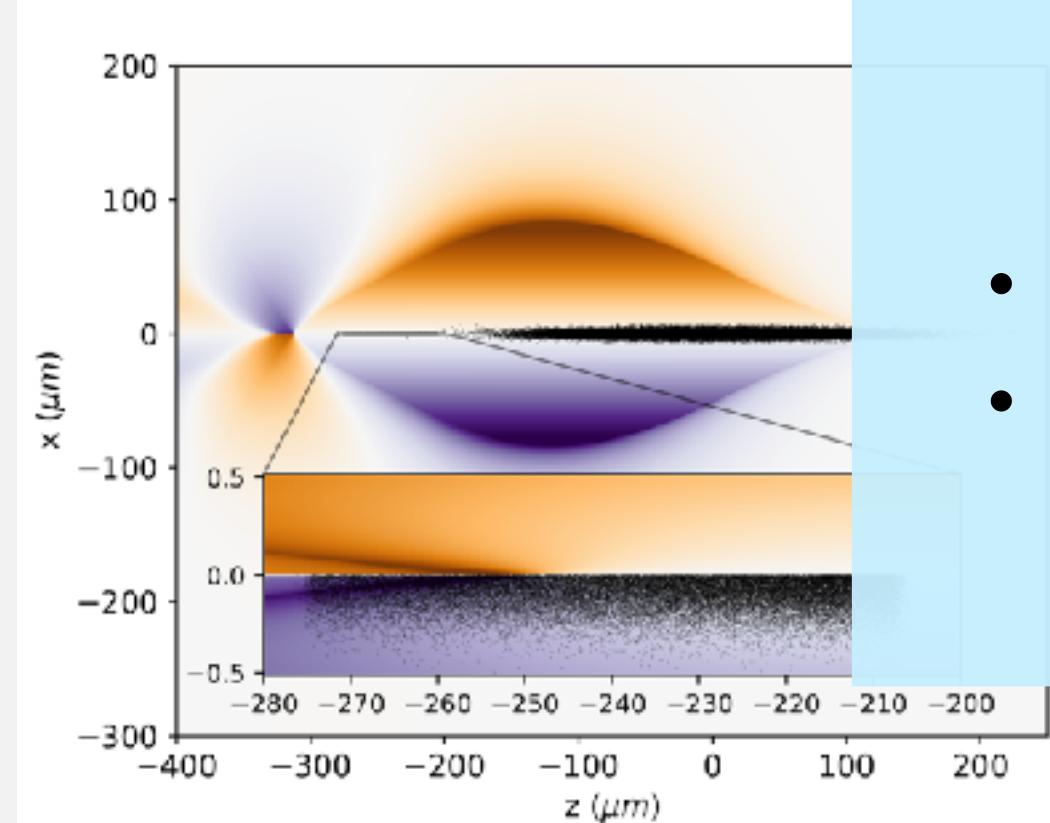
<https://github.com/AngelFP/Wake-T>



# Fully converged simulations with reduced models at modest cost

## HiPACE++

- Open-source, GPU-capable multi-physics 3D quasistatic particle-in-cell code (laser-driven or beam-driven), C++
- Collaboration



### Significant speedup by combining codes

- FBPIC (injection) + Wake-T (acceleration)
- WarpX + HiPACE++  
→ enabled by LASY

Fully converged (nm-scale resolution with mesh refinement) 3D simulations of a 20 GeV stage starting at 175 GeV for a 135 nm emittance beam w/ ion motion takes 30 min on 16 GPU-equipped nodes (Frontier has 9400) → small allocation allows thousands

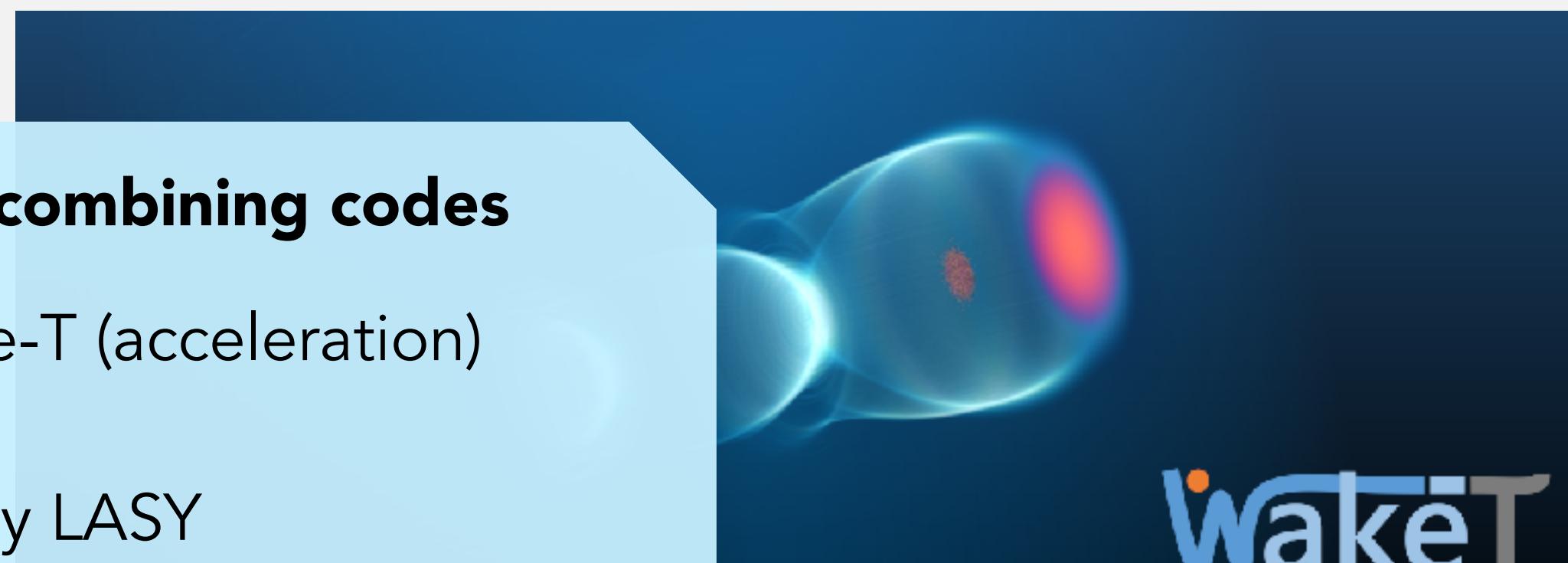
S. Diederichs et al., Comput. Phys. Comm. 278: 108421 (2022)

<https://agenda.infn.it/event/35577/contributions/208606>

<https://github.com/Hi-PACE/hipace>



- Open-source, 2D (axisymmetric) quasistatic code for (laser/beam-driven), incl. ion motion, Python



1 plasma stage takes seconds-to-minutes on a laptop, suitable for design studies. 20 stages + optimization to 150 GeV in < 1 hour

A. Ferran Pousa et al., J. Phys.: Conf. Ser. (2019)

A Ferran Pousa, et al. Proc. IPAC23, 1533

<https://github.com/AngelFP/Wake-T>



# LASY is a cornerstone of our S2E simulation ecosystem

Simulation studies gain from being:

## Multi-physics

- Hydrodynamics (HOFI, discharge, etc.)
- Beam dynamics, application

## Energy efficient

- Efficient codes for reduced model
- Combine the most appropriate tools

## Realistic

- Experimental laser profiles
- Experimental plasma/beam profiles

## Comprehensive

- Ensembles of simulations
- (Bayesian) Optimization



A Python library to simplify laser profile manipulations

- Use measured laser profiles in simulations
- Combine codes together (e.g. electromagnetic & quasistatic PIC)

LASY is a community effort



BERKELEY LAB

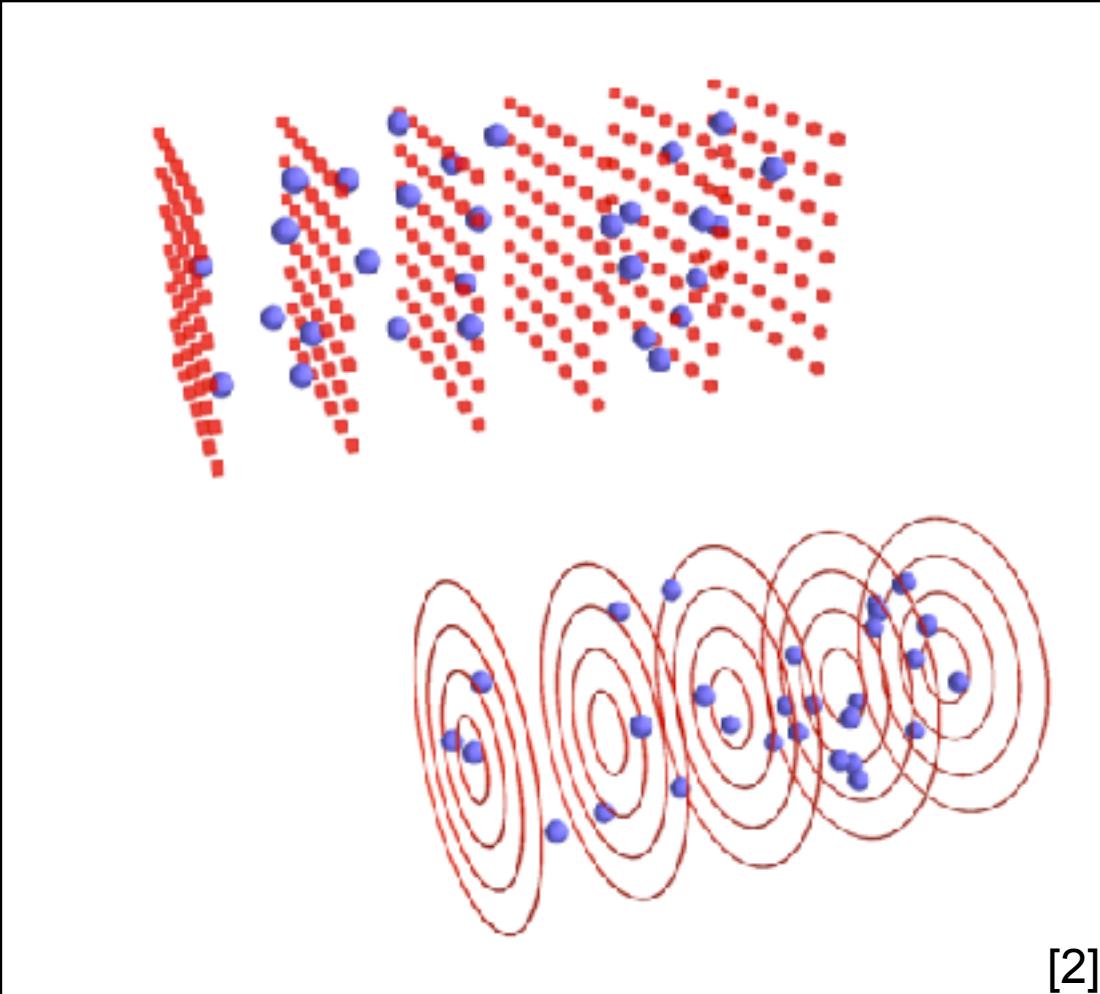
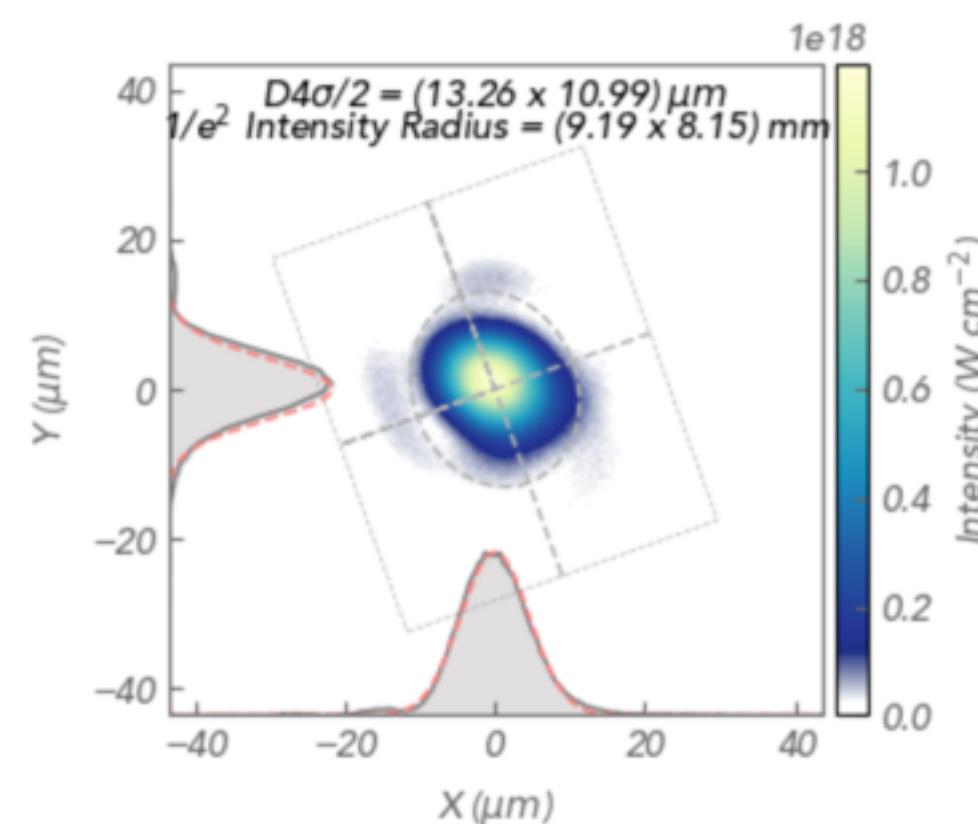


# Introducing LASY

# Laser initialisation in Simulations is becoming more complex

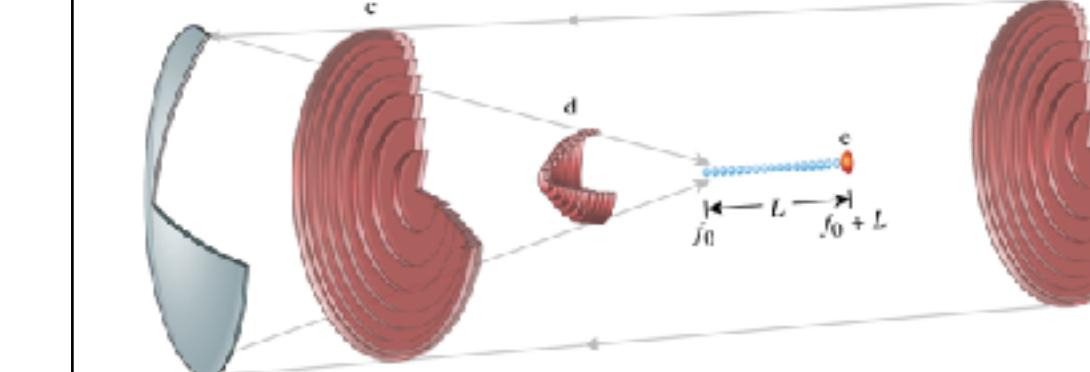
## This brings exciting new possibilities but can be challenging

**Realistic laser profiles** are key for realistic simulations of laser-plasma interactions [1]

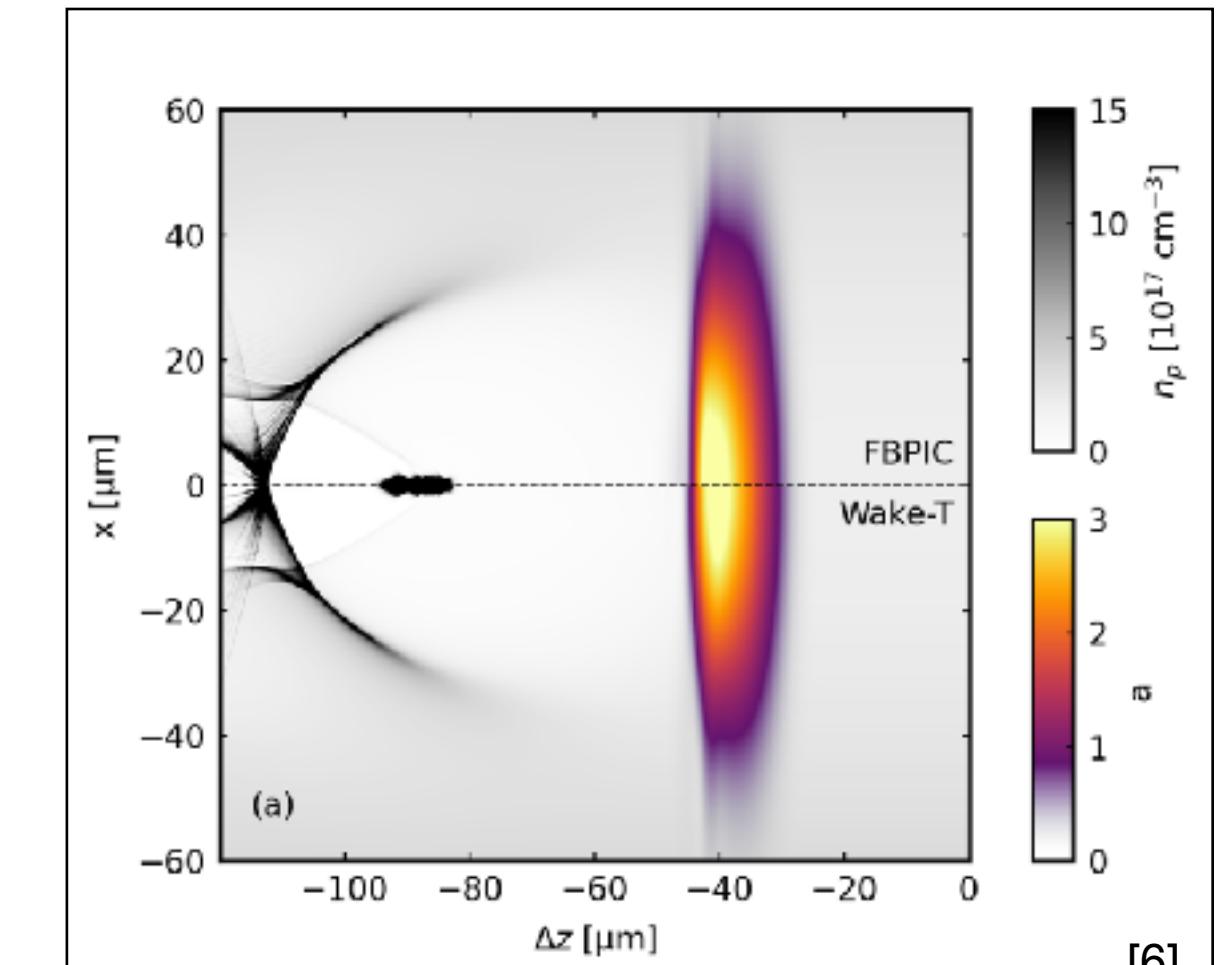


Start-to-end workflows require interfacing simulation tools with **very different conventions and representations**

Implementing **laser profiles with sophisticated structure** (e.g. flying-focus, OAM, STC etc.) is challenging) [3-5]



[3]



**No standard initialisation** conventions can make comparing results complex

**LASY simplifies these workflows with modern programming methods (Open-source, Python, CI/CD, data standards)**

[1] B. Beaurepaire et al., *Phys. Rev. X* **5**, 03102 (2015)

[2] <https://fbpic.github.io/>

[3] J. P. Palastro et al., *Phys. Rev. Lett.* **124**, 134802 (2020)

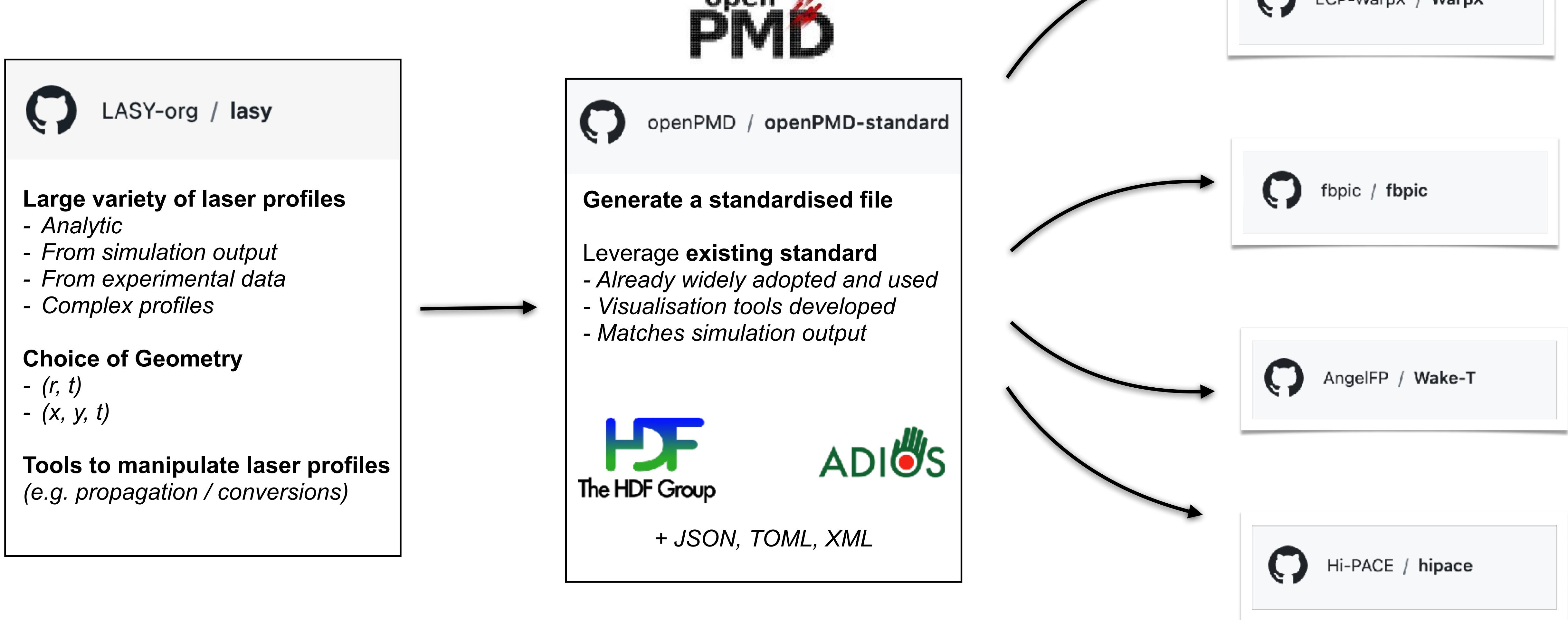
[4] C. Caizergues et al., *Nat. Photon.* **14**, 475 (2020)

[5] A. Debus et al., *Phys. Rev. X* **9**, 031044 (2019)

[6] A. Ferran Pousa et al., *Phys. Rev. Accel. Beams* **26**, 084601 (2023)

# Introducing LASY

## An Open-Source Community Solution



M. Thévenet *et al*, submitted to EAAC 2023 proceedings (arXiv:2403.12191)

# Introducing LASY

## An Open-Source Community Solution

[LASY-org / lasy](#) Public

### Open-Source Development

Code Issues 37 Full requests 14 Actions Projects 6 Security Insights

development Go to file Code About

hightower8083 and pre-commit-ci[bot] f... 2d45110 · last week 216 Commits

File	Description	Last Commit
.github/workflows	Publish to PyPI on release (#159)	7 months ago
docs	Add ccs- longitudinal profile (#207)	2 months ago
examples	Implement Gerchberg Saxton Algorith...	8 months ago
lasy	fix a bug with show method for N>1 (#...	last week
tests	Closes #215 , add overloading for addi...	last month
.gitignore	Make energy calibration the default (#...	last year
.pre-commit-config.yaml	[pre-commit.ci] pre-commit autoupda...	2 weeks ago
.readthedocs.yaml	Fix issues in the documentation (#146)	10 months ago
.zenodo.json	Zenodo File (#172)	7 months ago
README.md	Update instalation instructions (#192)	5 months ago
conda.yml	Merge hackathon branch into developm...	last year
legal.txt	[pre-commit.ci] auto fixes from pre-co...	last year
license.txt	[pre-commit.ci] auto fixes from pre-co...	last year
pyproject.toml	Pre-Commit: pydocstyle (#120)	last year
requirements.txt	remove the version specifier for axipro...	3 months ago
setup.py	Add version number for lasy (#131)	last year

README License

**lasy**

Overview

## LASY 0.4.0 Documentation

### Online Documentation

**lasy** (LAser manipulations made eaSY) is a Python library that facilitates the initialization of complex laser pulses, in simulations of laser-plasma interactions.

More specifically, **lasy** offers many ways to define complex laser pulses (e.g. from commonly-known analytical formulas, from experimental measurements, etc.) and offers pre-processing functionalities (e.g. propagation, re-normalization, geometry conversion). The laser field is then exported in a standardized file, that can be read by external simulation codes.

The code is open-source and hosted on [github](#). Contributions are welcome!

**Getting Started**  
New to **lasy**? Check this out for installation instructions and a first example.

**Overview of the Code**  
An overview of the key concepts and functionality of the code.

**More Information**

**API Reference**  
Get into the nuts and bolts of the **lasy** API with the documentation here.

**Tutorials**  
Some step-by-step guides to using the code and some common examples which you might find useful.

**CI/CD**

All checks have passed  
6 successful checks

- CodeQL / Analyze (python) (pull\_request) Successful in 4m Details
- Unix / unittest (ubuntu-latest) (pull\_request) Successful in 54s Required Details
- Unix / pyflakes (pull\_request) Successful in 4s Required Details
- Code scanning results / CodeQL Successful in 3s — No new alerts in code changed by this pull re... Details
- docs/readthedocs.org:lasydoc — Read the Docs build succeeded! Details
- pre-commit.ci - pr — checks completed successfully Details

# Introducing LASY

## An Open-Source Community Solution



Ángel Ferran Pousa  
Sören Jalas  
Manuel Kirchen  
Rob Shalloo  
Alexander Sinn  
Maxence Thévenet

Axel Huebl  
Rémi Lehe  
Jean-Luc Vay

Igor Andriyash

Luca Fedeli  
Thomas Clark

Spencer Jolly

New Contributors Highly Encouraged!

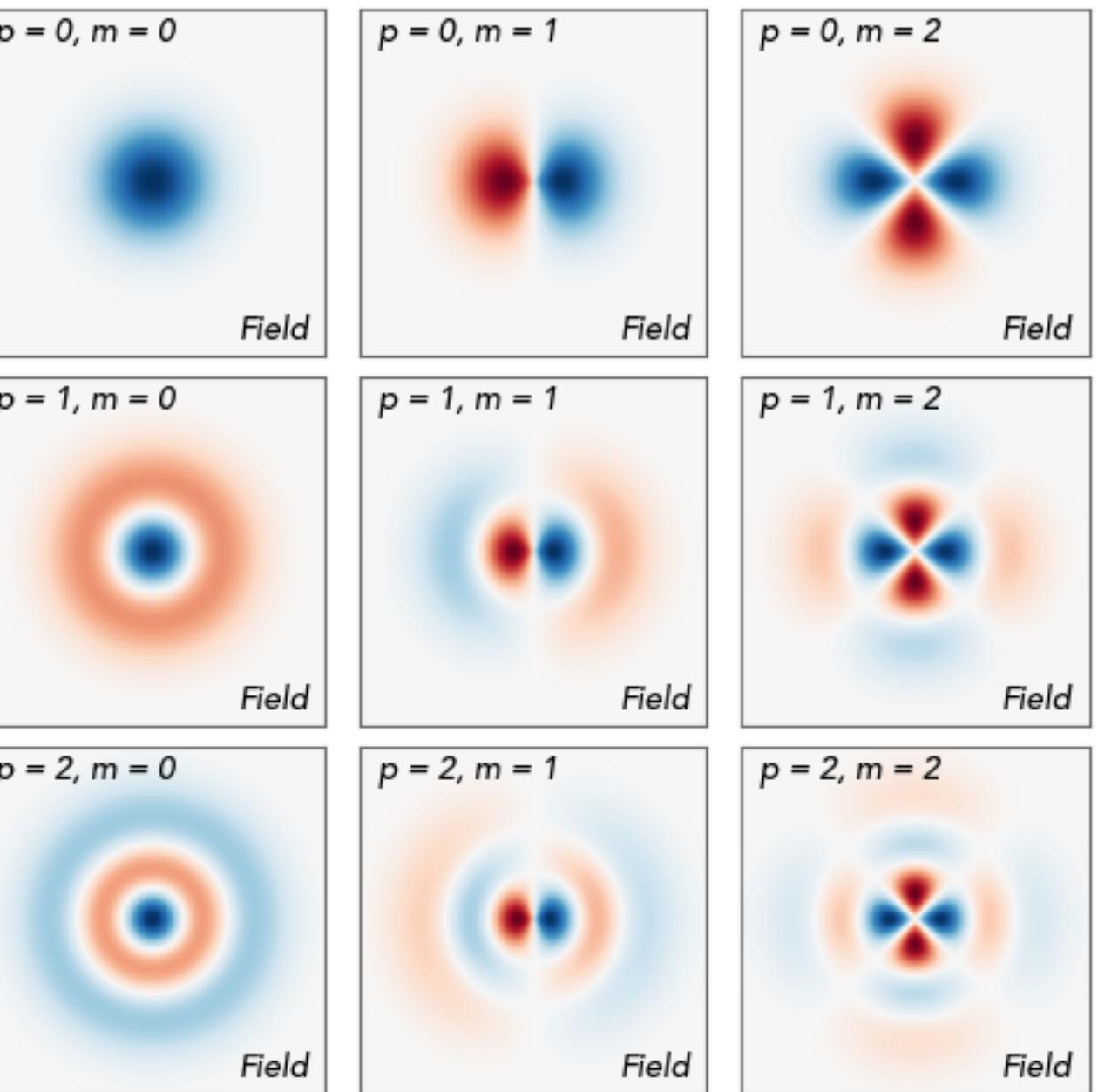
# Defining A Laser Pulse

# Laser Profile Initialisation

## Analytic Profiles

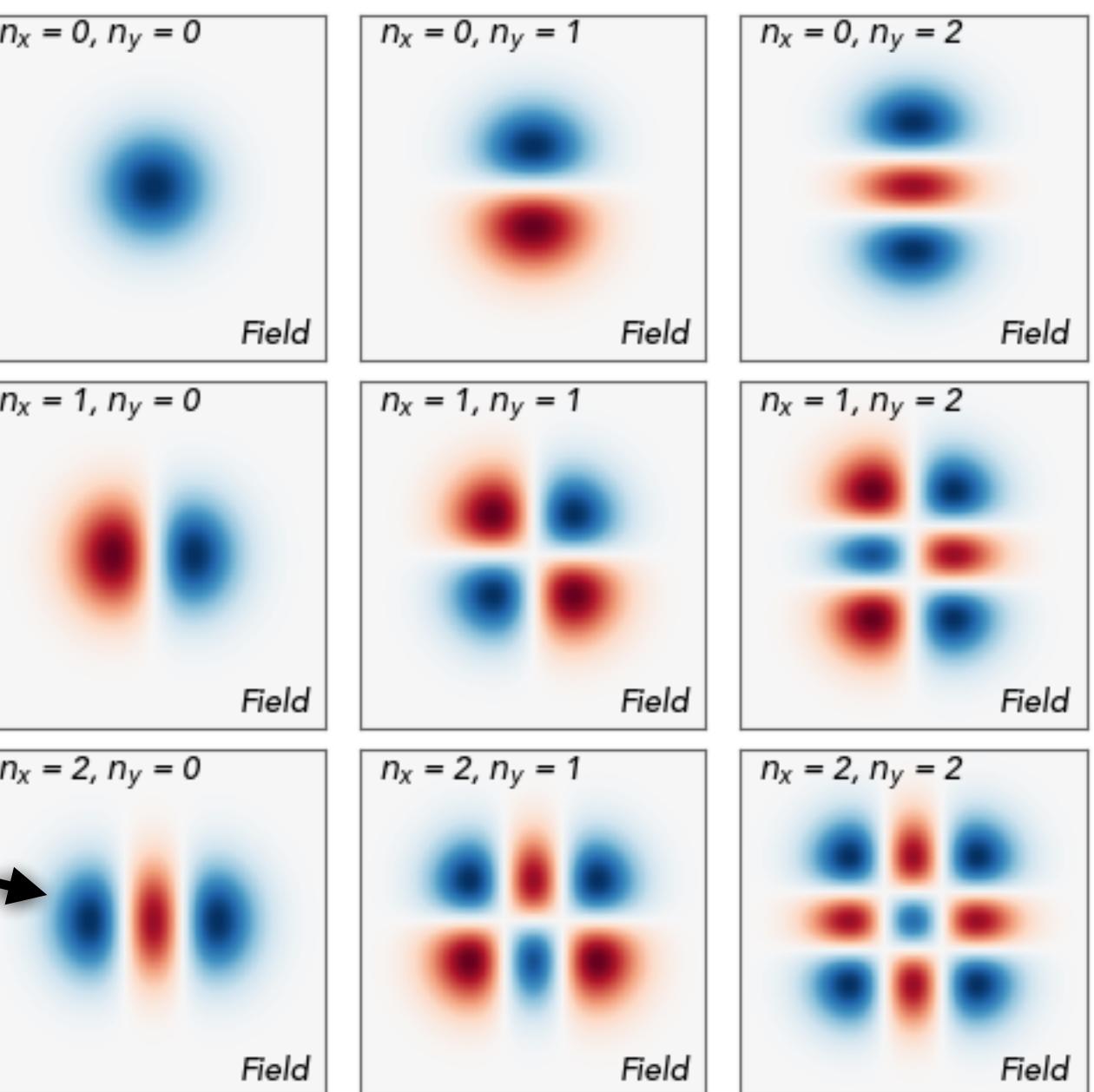
Large variety of transverse analytic profiles already included

- Gaussian
- Supergaussian
- Jinc
- Laguerre Gaussian
- Hermite Gaussian



Longitudinal Profiles can be defined separately

Profiles can then be Combined



```
● ● ●  
from lasy.profiles.transverse.hermite_gaussian_profile import HermiteGaussianTransverseProfile  
w0      = 10e-6  
n_x    = 2  
n_y    = 0  
  
transverse_profile = HermiteGaussianTransverseProfile(w0,n_x,n_y)
```

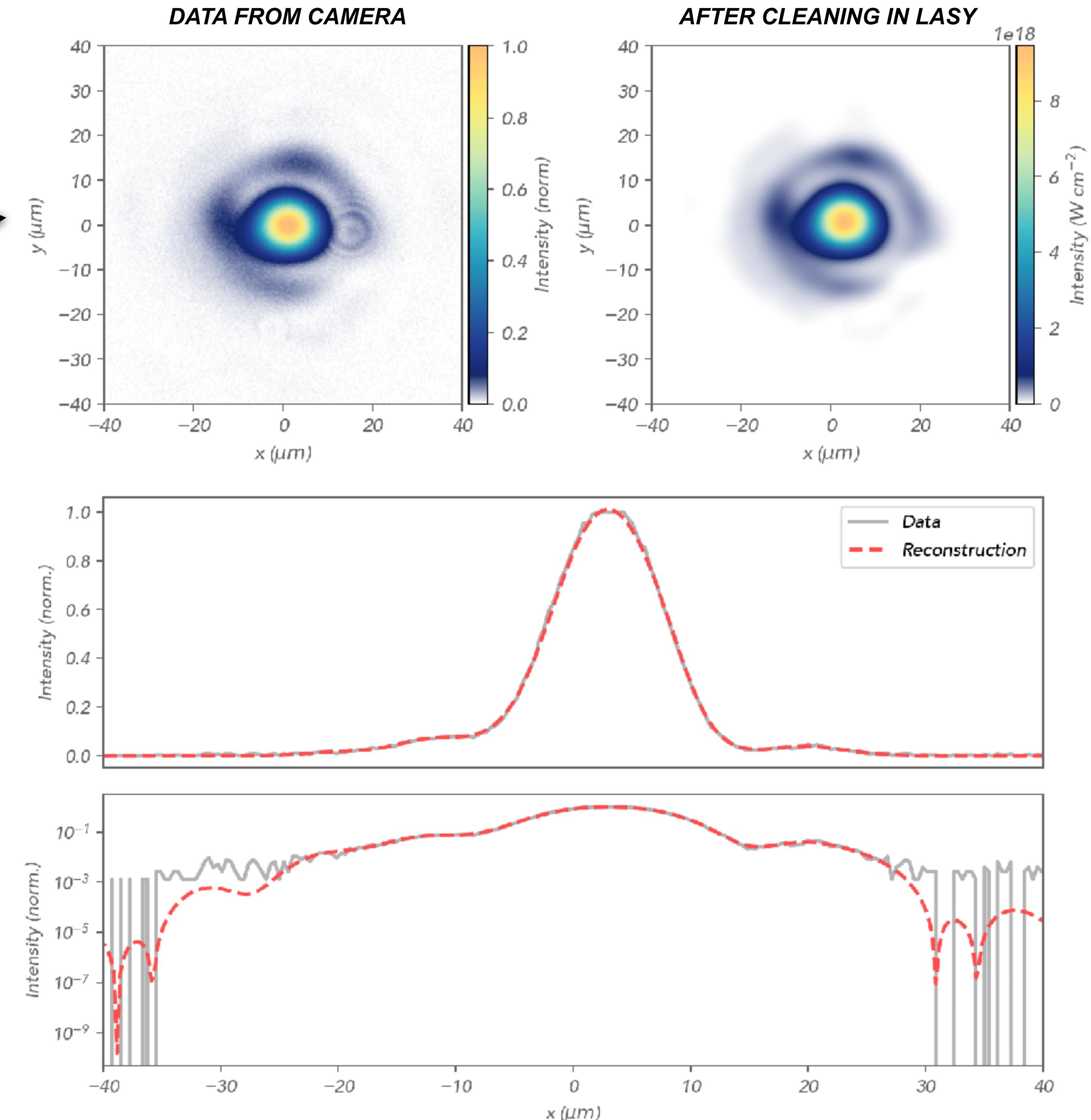
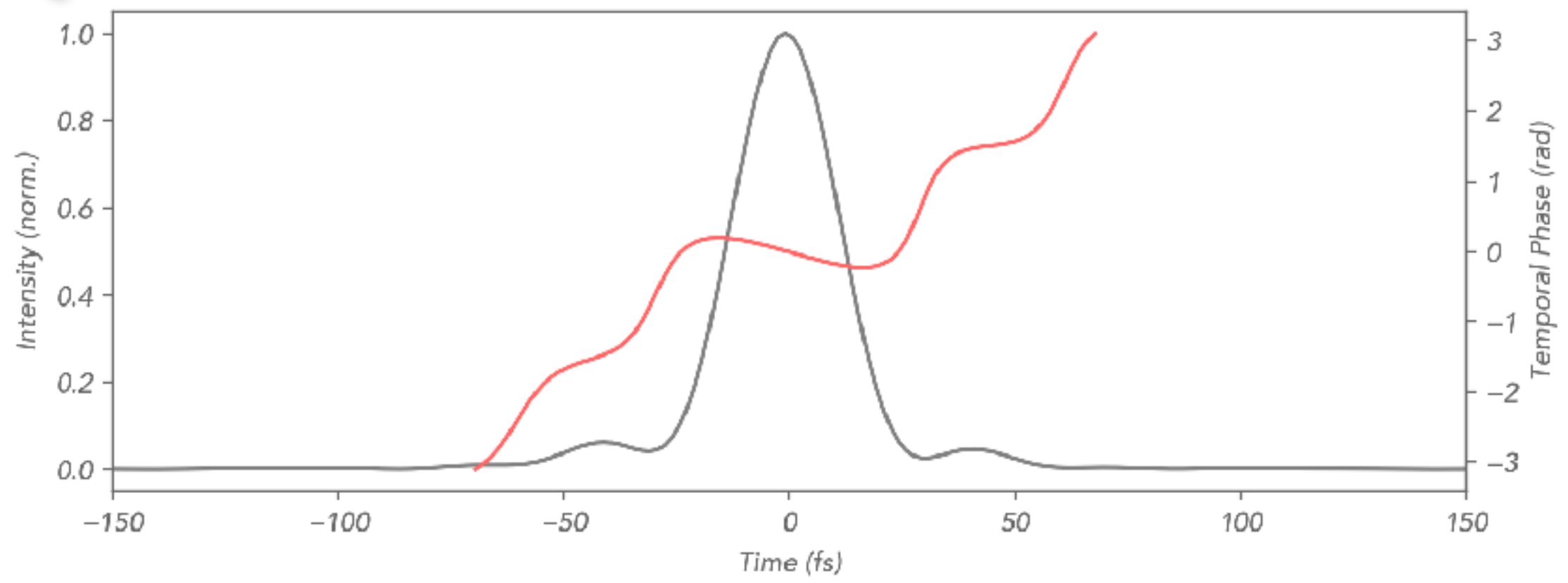
# Laser Profile Initialisation From Experimental Measurements

Incorporate Laser Fluence Measurements (e.g. Camera)

Some post-processing available in LASY

Incorporate Pulse Duration Measurements (e.g. Wizzler)

Incorporate advanced diagnostics (e.g. INSIGHT)



# Laser Profile Initialisation

## From Simulation to Simulation

### FBPIC [1]: Electromagnetic PIC code capturing injection

*Laser pulse: self-consistent electric and magnetic fields*

### Wake-T [2]: Quasi-static code for fast & accurate simulations on a laptop

*Laser pulse: envelope of the vector potential*

#### FBPIC

- Run simulation
- Output data

#### LASY

- Directly read openPMD data from FBPIC
  - Convert full-field to envelope representation
  - Convert electric field to vector potential
- Save to standard format



#### Wake-T

- Initialise LASY data
- Continue Simulation

# Laser Profile Initialisation

## From Simulation to Simulation

**FBPIC** [1]: Electromagnetic PIC code capturing injection

Laser pulse: self-consistent electric and magnetic fields

**Wake-T** [2]: Quasi-static code for fast & accurate simulations on a laptop

Laser pulse: envelope of the vector potential

### FBPIC

- Run simulation
- Output data

### LASY

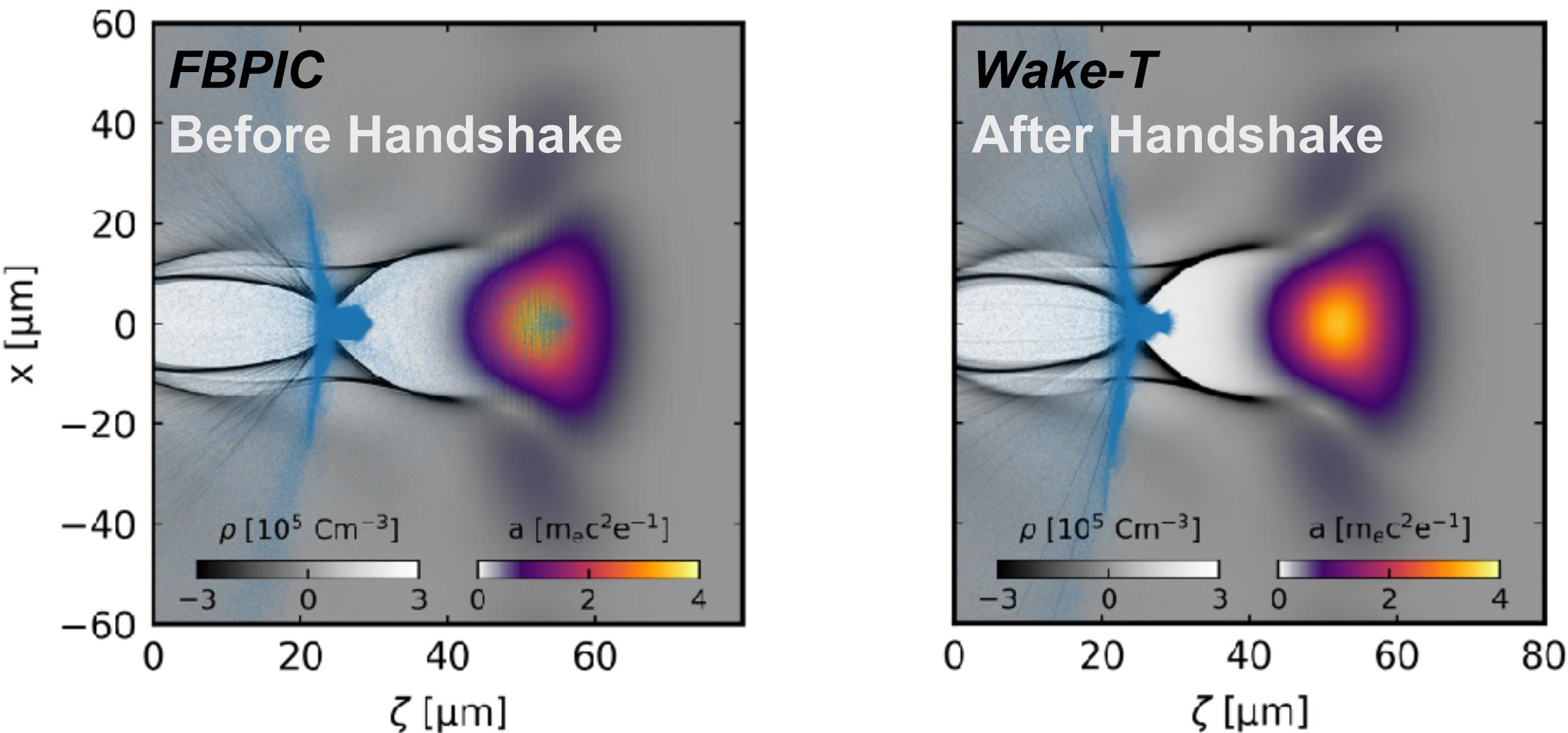
- Directly read openPMD data from FBPIC
  - Convert full-field to envelope representation
  - Convert electric field to vector potential
- Save to standard format



### Wake-T

- Initialise LASY data
- Continue Simulation

Define laser pulse directly from openPMD file



[1] R. Lehe et al., *Comput. Phys. Commun.* **203**, 66 (2016)

[2] A. Ferran Pousa et al., *Journ. Phys.* **1350.1** IOP Publishing (2019)

# Laser Profile Initialisation

## From Simulation to Simulation

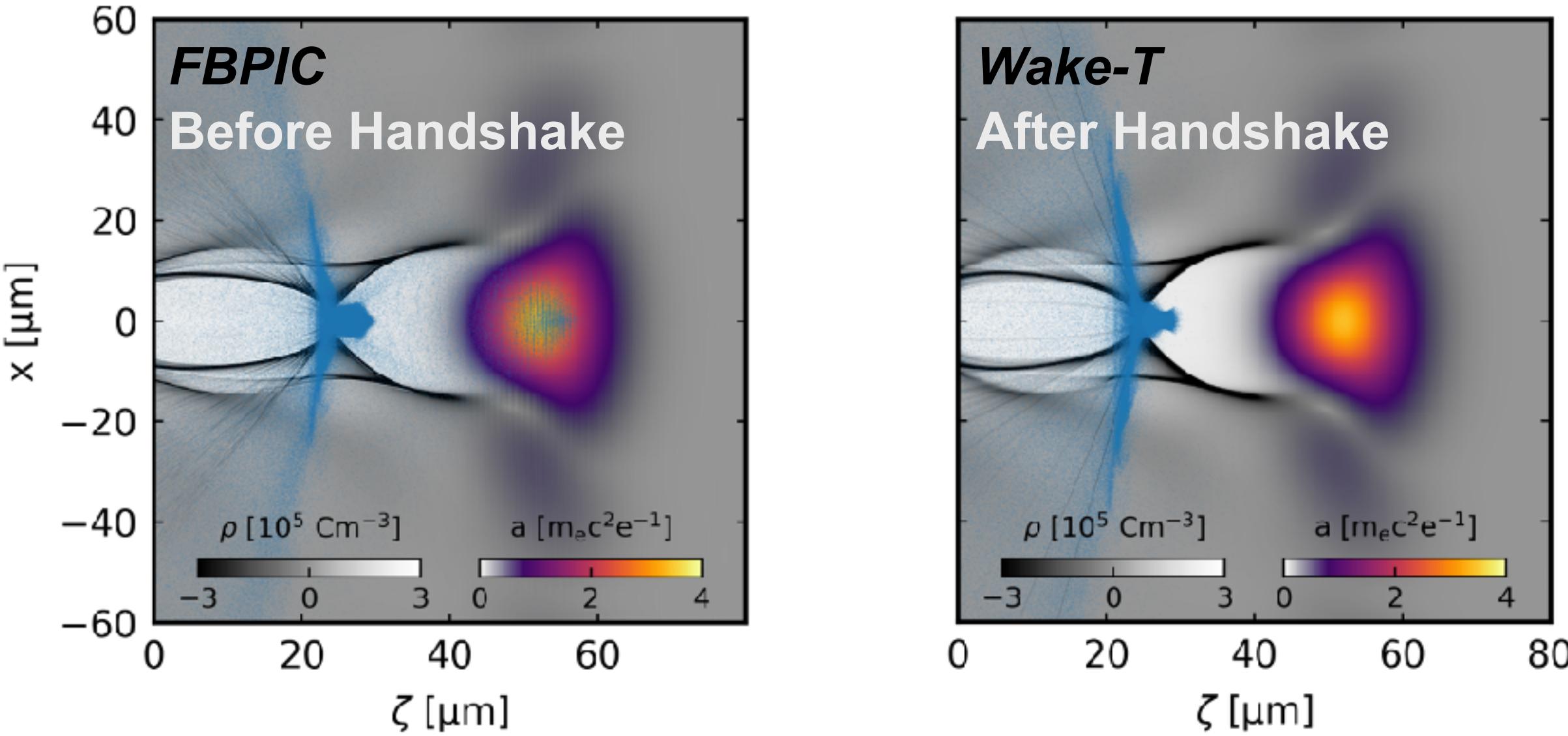
**FBPIC** [1]: Electromagnetic PIC code capturing injection

Laser pulse: self-consistent electric and magnetic fields

**Wake-T** [2]: Quasi-static code for fast & accurate simulations on a laptop

Laser pulse: envelope of the vector potential

Define laser pulse directly from openPMD file



### FBPIC

- Run simulation
- Output data

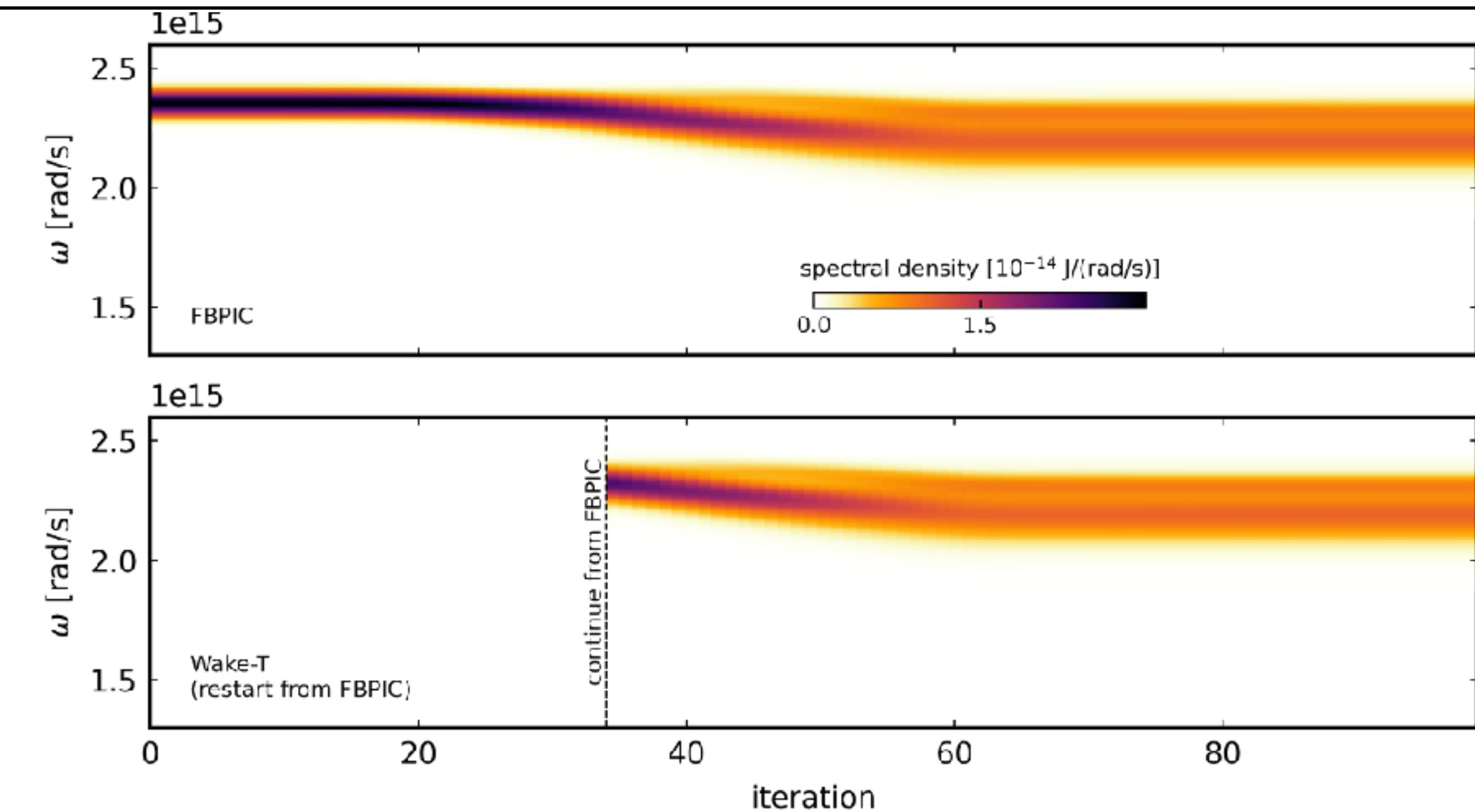
### LASY

- Directly read openPMD data from FBPIC
  - Convert full-field to envelope representation
  - Convert electric field to vector potential
- Save to standard format



### Wake-T

- Initialise LASY data
- Continue Simulation



[1] R. Lehe et al., *Comput. Phys. Commun.* **203**, 66 (2016)

[2] A. Ferran Pousa et al., *Journ. Phys.* **1350.1** IOP Publishing (2019)

# Laser Profile Initialisation

## From Simulation to Simulation

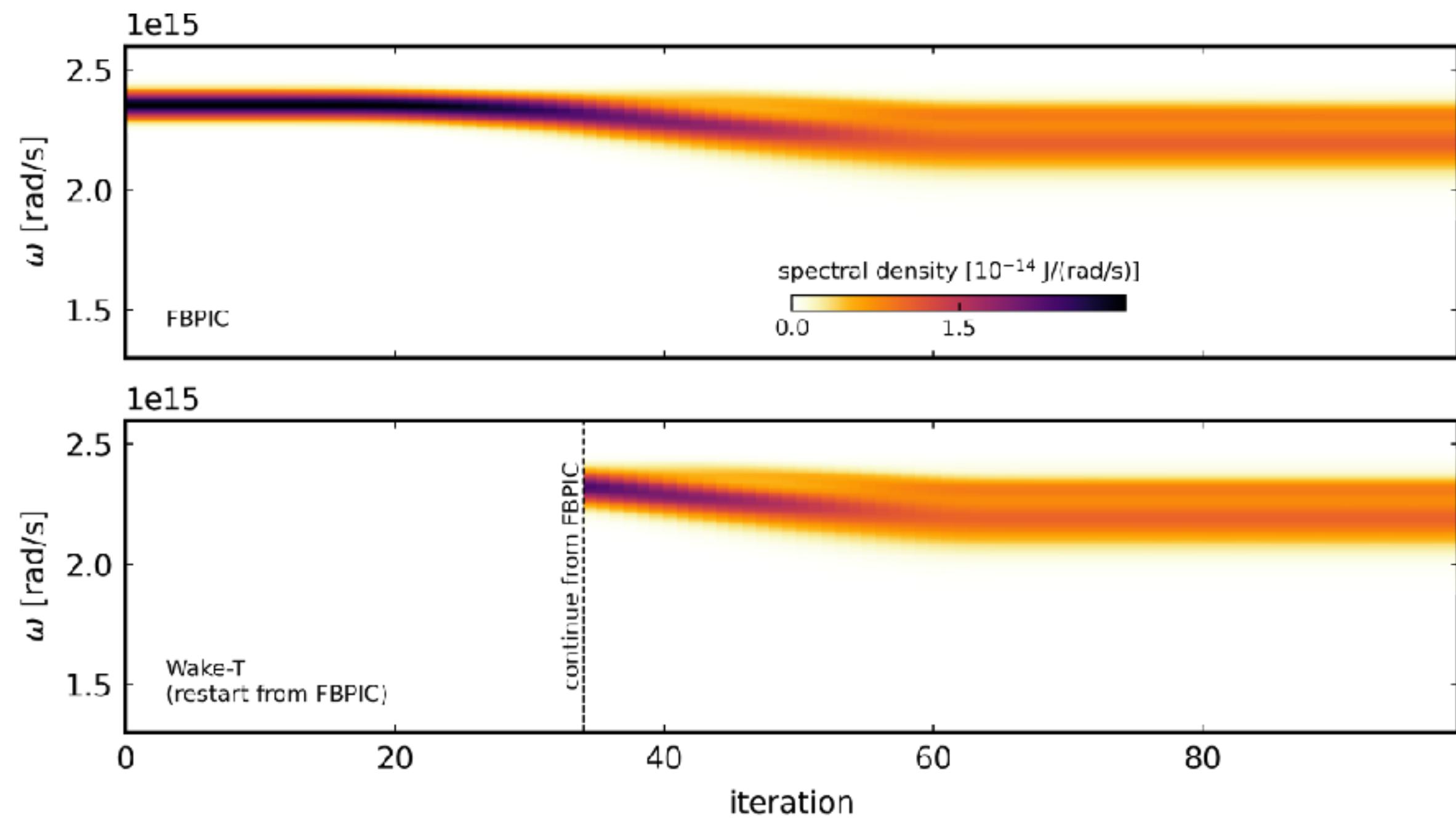
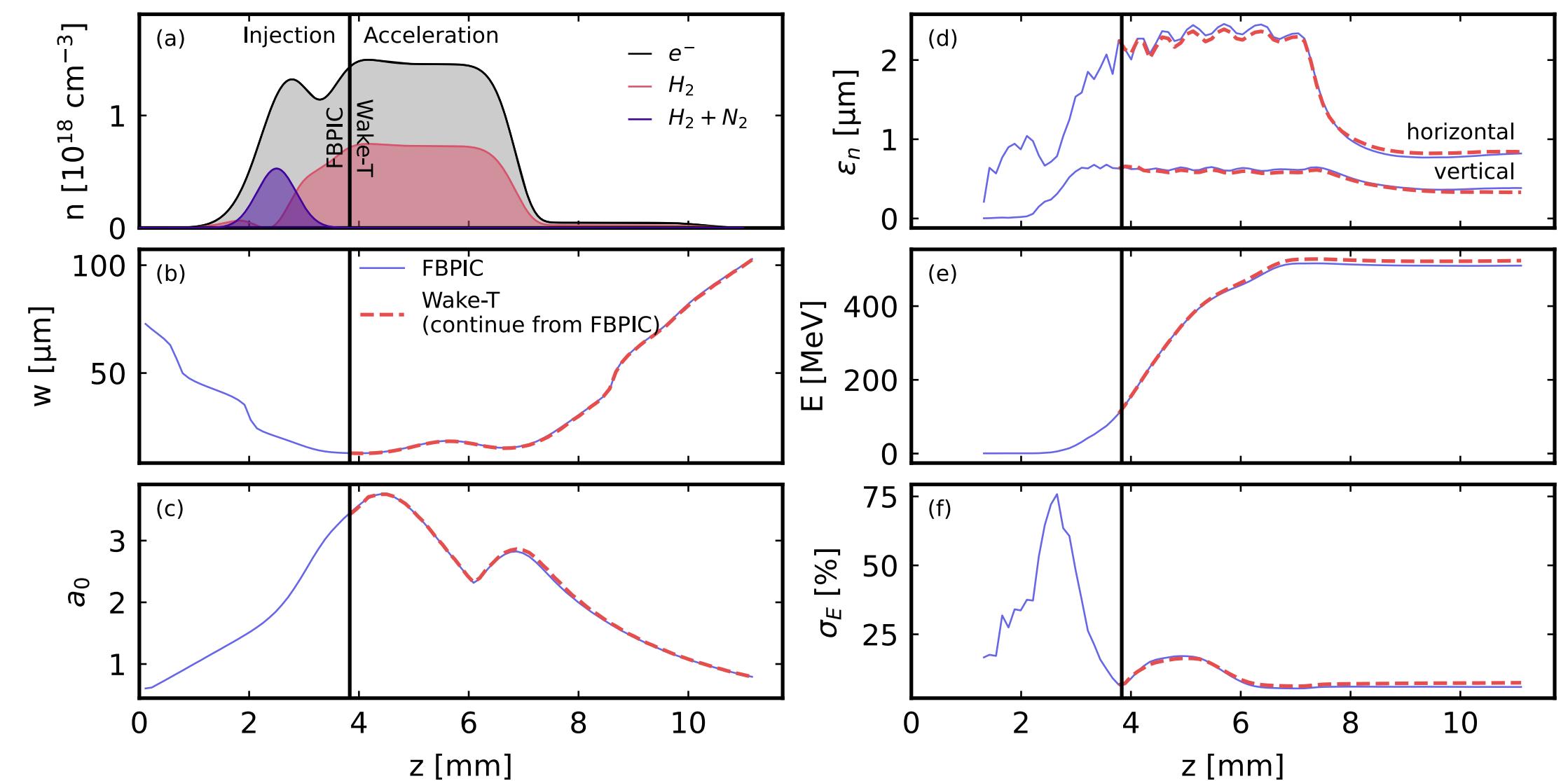
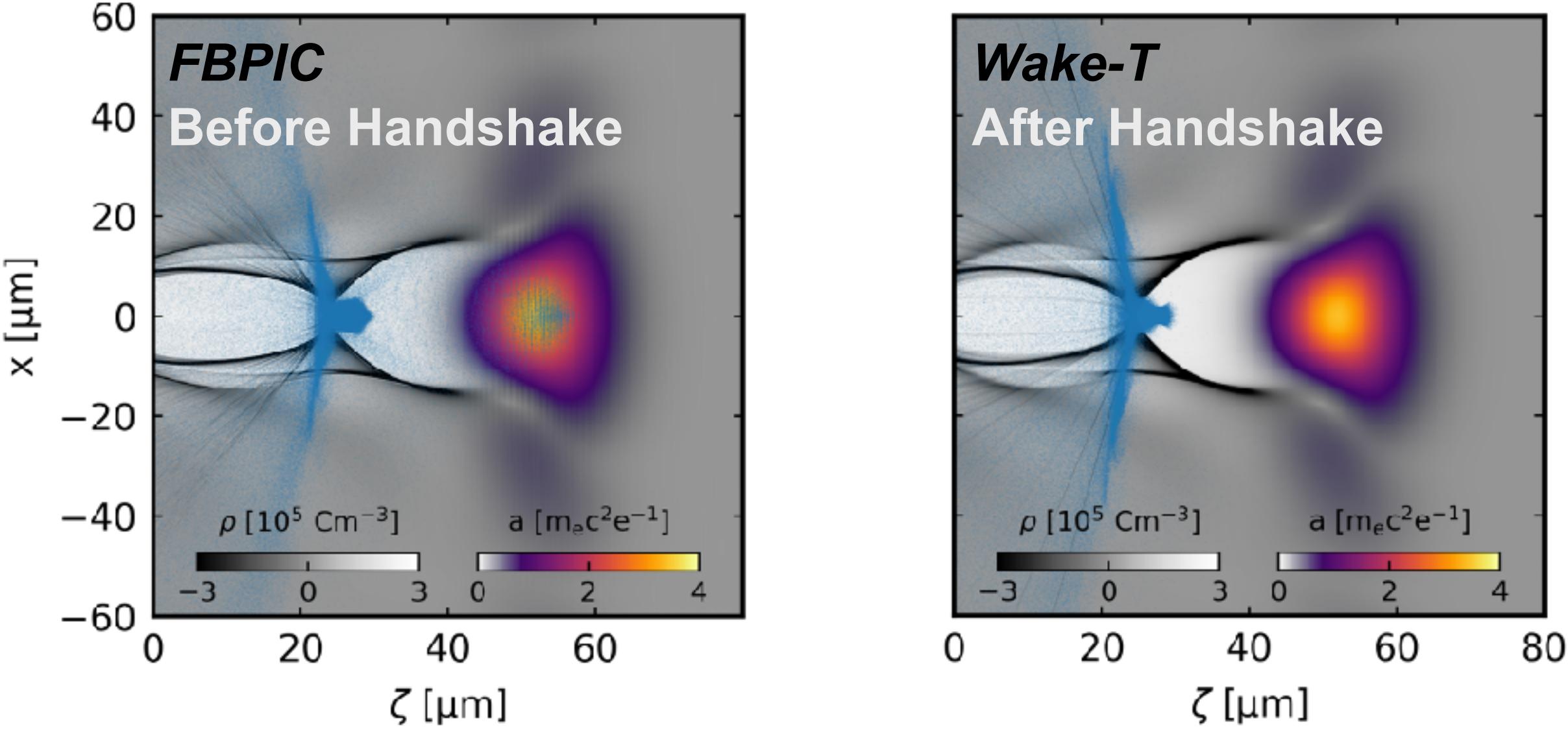
**FBPIC [1]: Electromagnetic PIC code capturing injection**

*Laser pulse: self-consistent electric and magnetic fields*

**Wake-T [2]: Quasi-static code for fast & accurate simulations on a laptop**

*Laser pulse: envelope of the vector potential*

Define laser pulse directly from openPMD file



[1] R. Lehe et al., *Comput. Phys. Commun.* **203**, 66 (2016)

[2] A. Ferran Pousa et al., *Journ. Phys.* **1350.1** IOP Publishing (2019)

# Laser Profile Initialisation

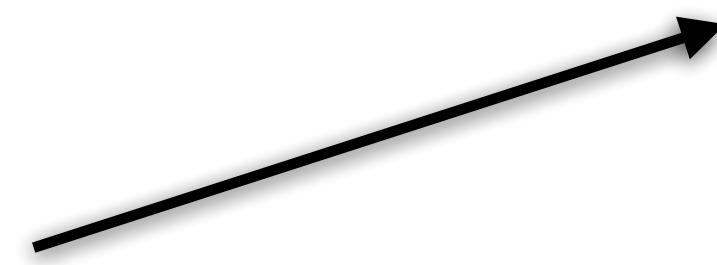
## Spatio-Temporal Couplings (STCs)

**Lasers with STCs are possible in LASY**

Currently under development

First implementation of spatial chirp ready for review

Possibility to incorporate flying focus and other pulses of interest



Space time profile #182

[Open](#) spencerjolly wants to merge 10 commits into [LASY-org:development](#) from [spencerjolly:space-time](#)

Conversation 0 Commits 10 Checks 4 Files changed 1

spencerjolly commented on Sep 13 · edited Member ...

This adds the ability to create a pulse with space-time couplings. Spatial chirp at the focus ( $z=0$ ) can be added with a real-valued parameter 'sc'. This uses the highest level 'Profile' class. An interesting test is if the propagator works properly to propagate such pulses to the beginning of a simulation box. Analytical equations could also be used to directly calculate the fields at a user-defined  $z$ -position.

Watch this space!

# Laser Pulse Manipulation

# Laser Pulse Manipulation

## Propagation in Vacuum

Powered by Axiprop

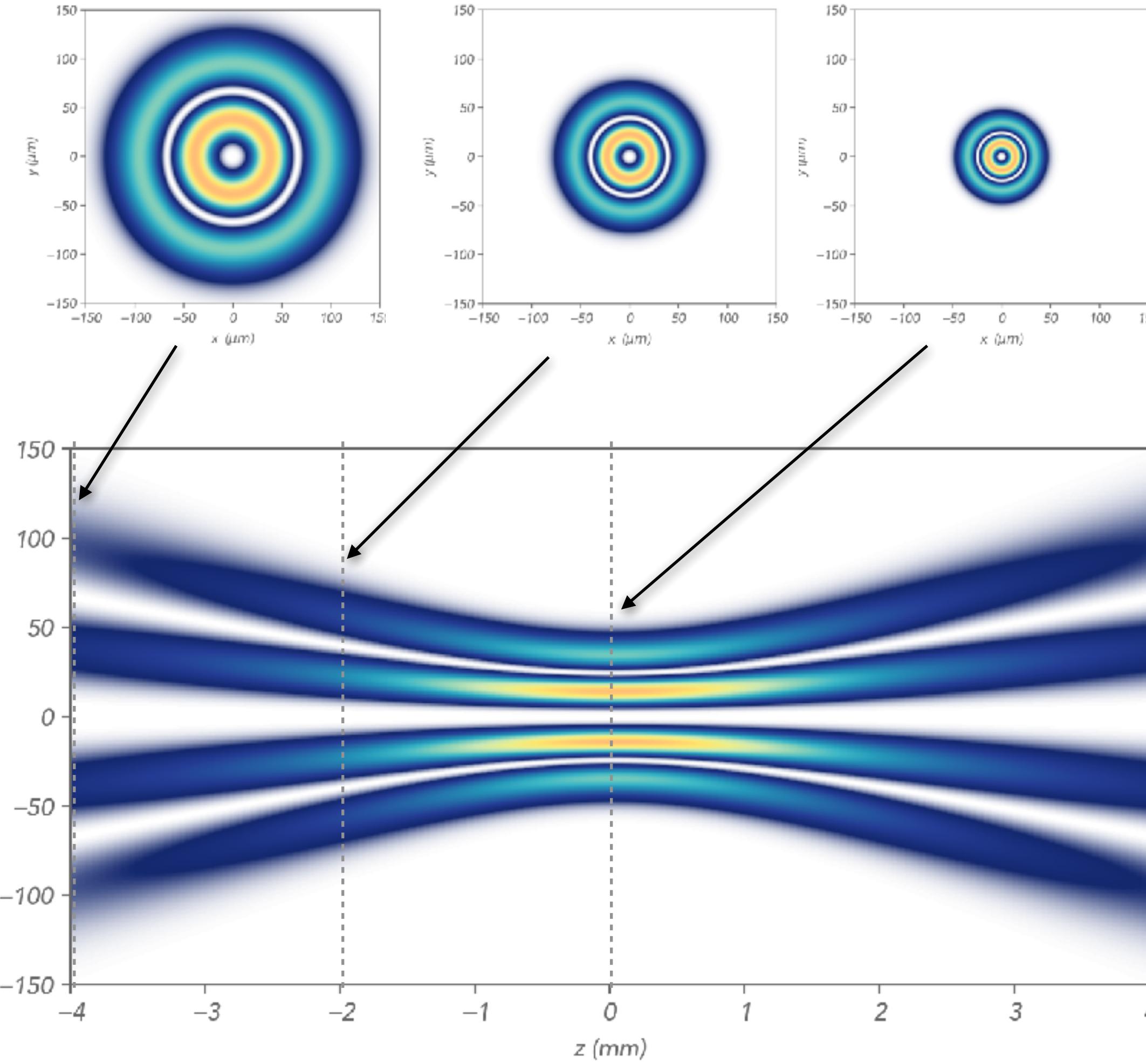
I. Andriyash



hightower8083 / axiprop

Code Issues Pull requests 2

axiprop Public



Laser can be defined in one plane and then numerically propagated to where it is needed

Propagation algorithms

- Cartesian or cylindrical geometry
- On CPU or GPU

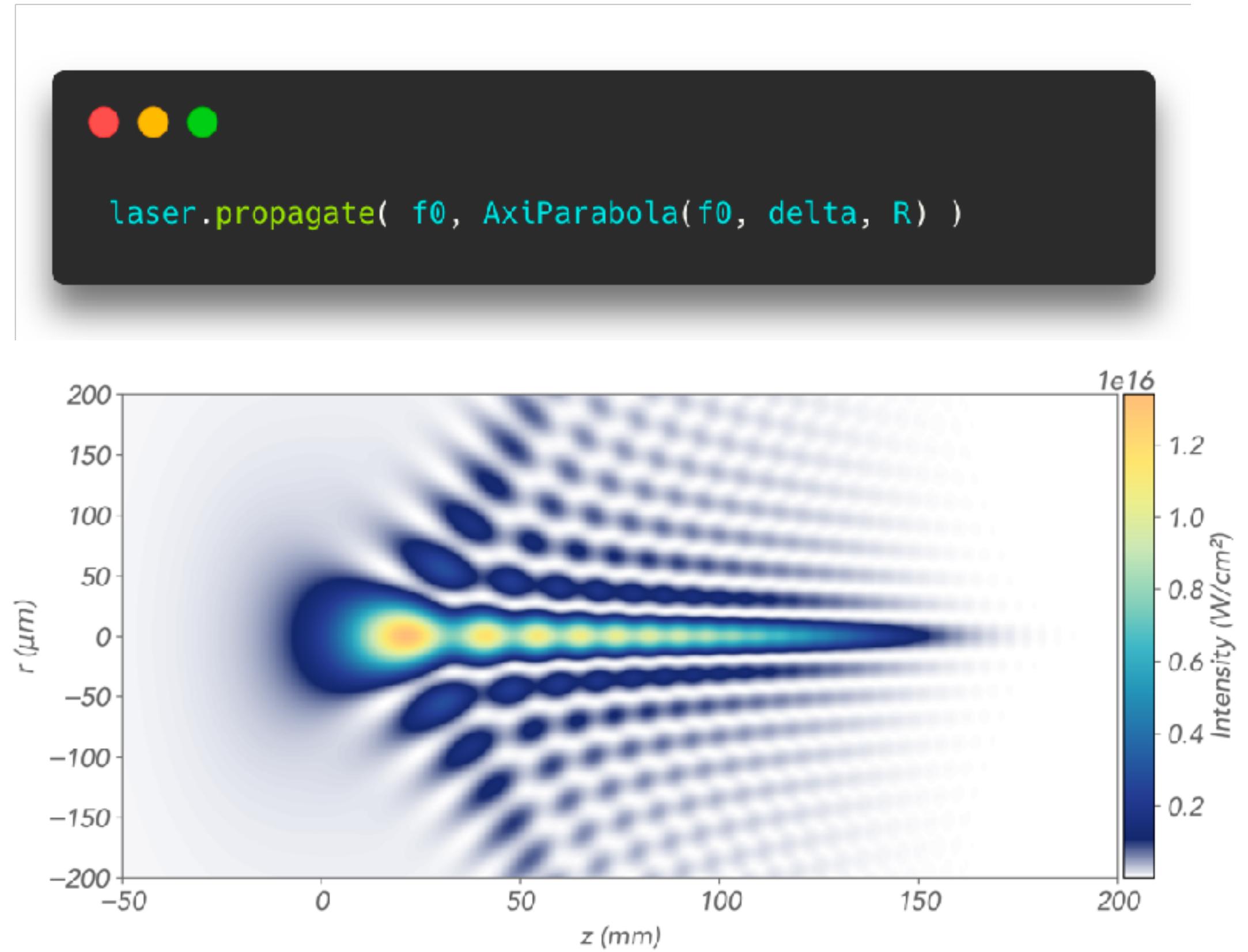


# Laser Pulse Manipulation

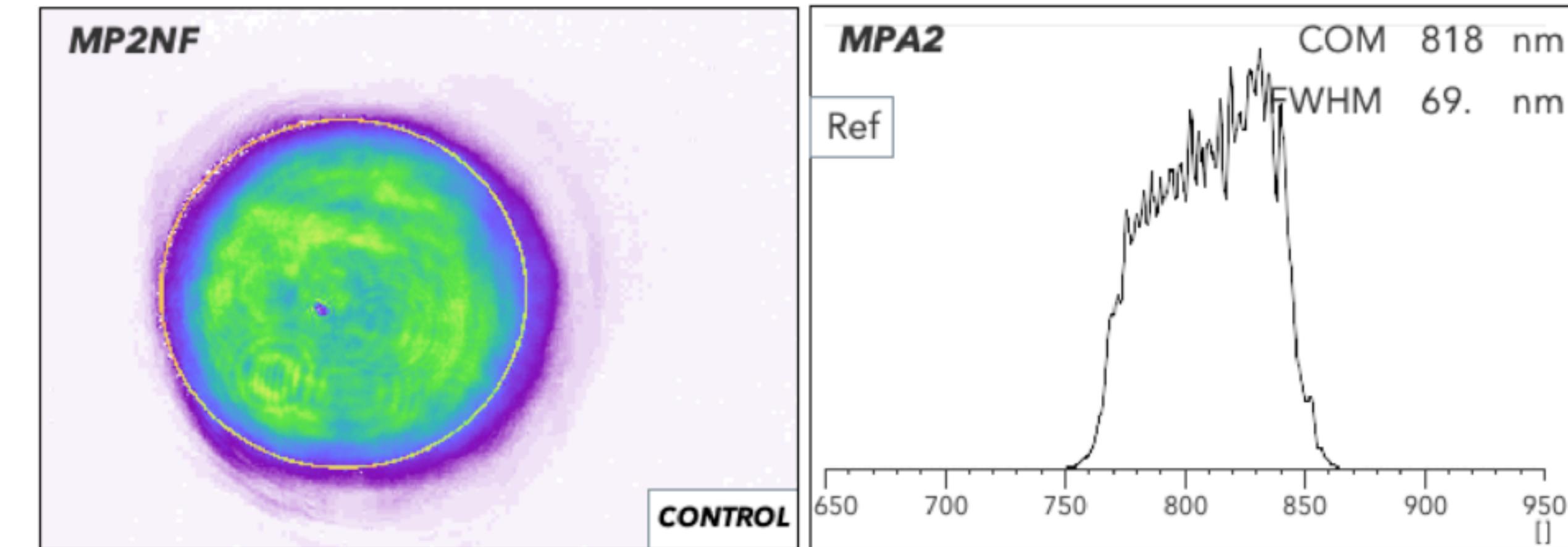
## Adding Optical Elements

Plans to add optical elements to LASY (PR #199)

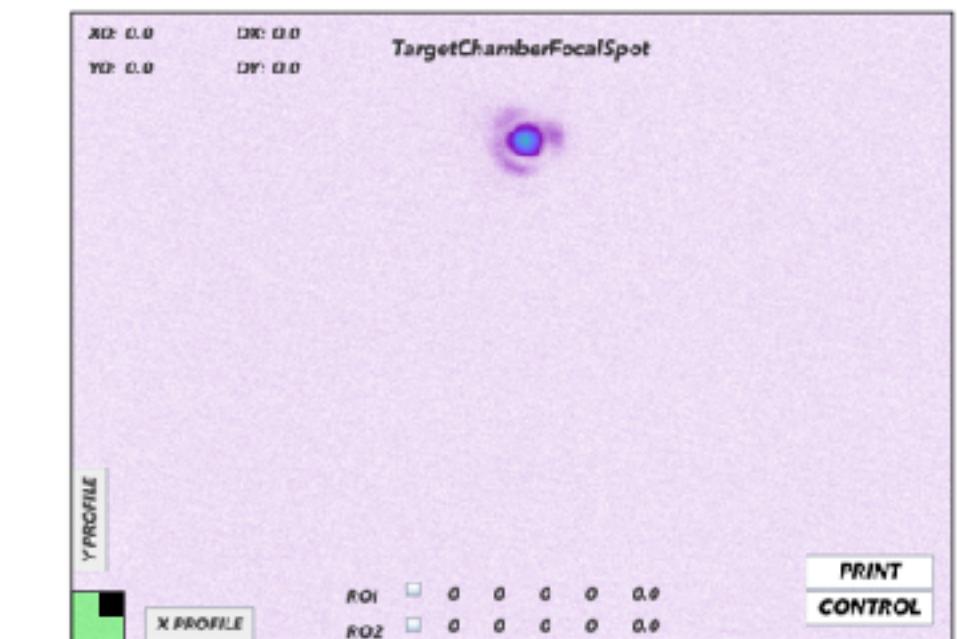
Example: Axiparabola



Example:  
Defining laser based on NF measurements from Online Diagnostics



Calculate focal profile and use  
for simulations



# Laser Pulse Manipulation

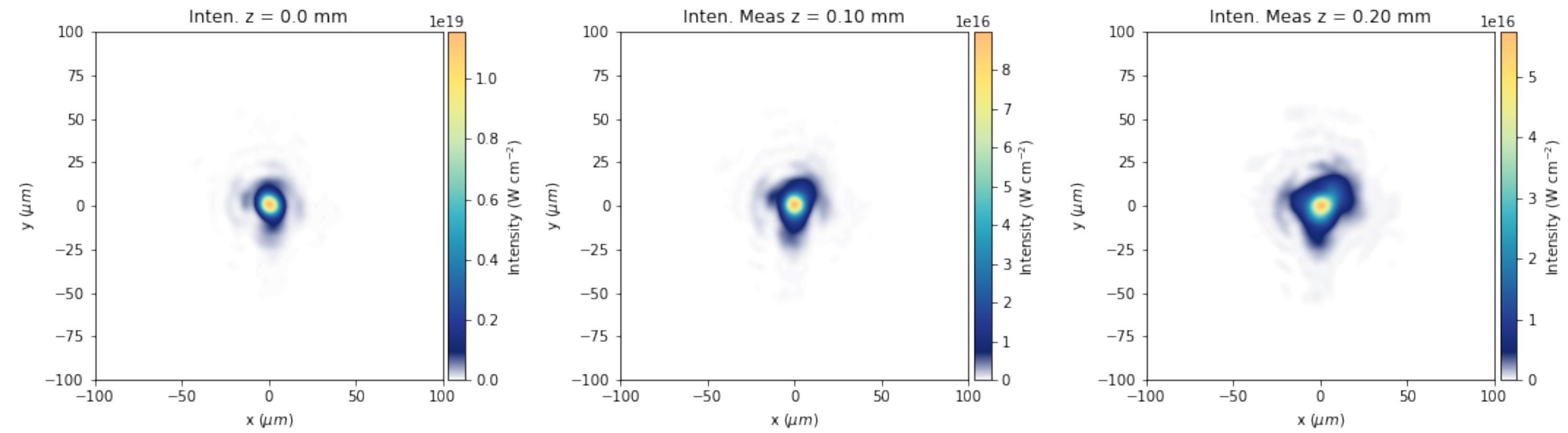
## Gerchberg Saxton Algorithm



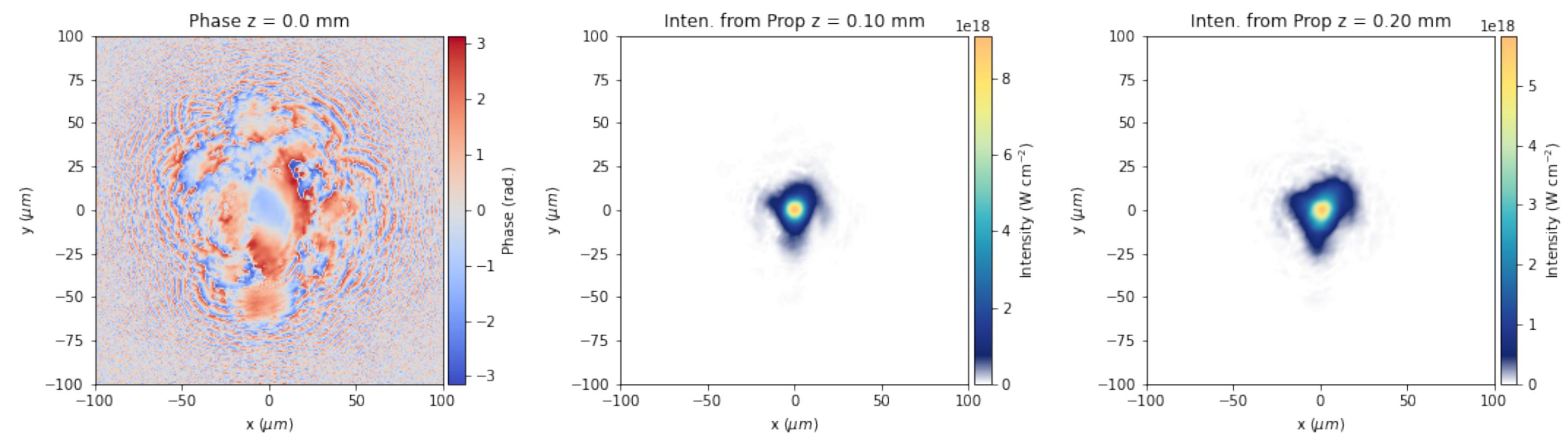
```
phaseBackward, phaseForward, amp_error = gerchberg_saxton_algo( laser1, laser2, propDist )
```

Iterative algorithm to calculate laser phase

Measured laser profiles



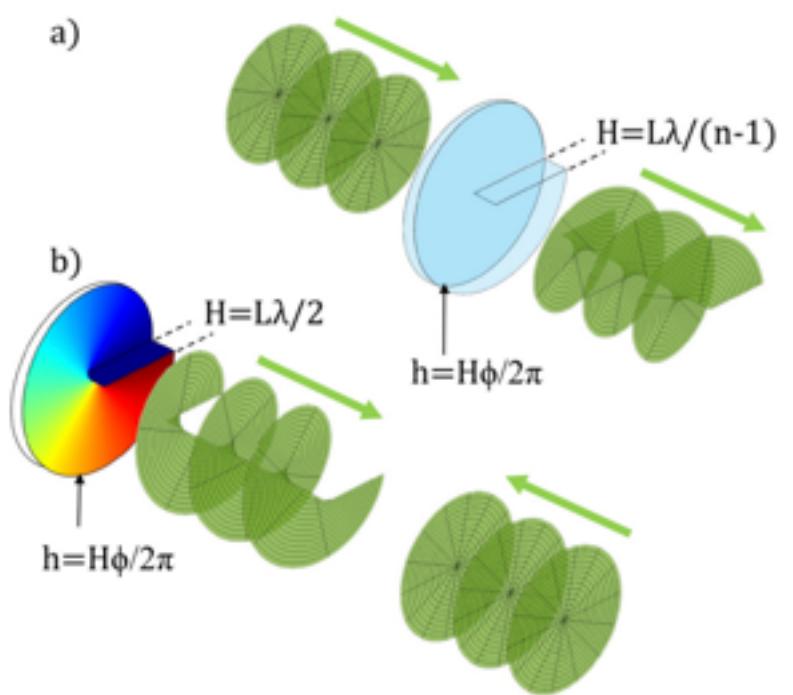
Retrieved phase  
+  
Calculated intensity downstream



# Modelling complex experimental pulses

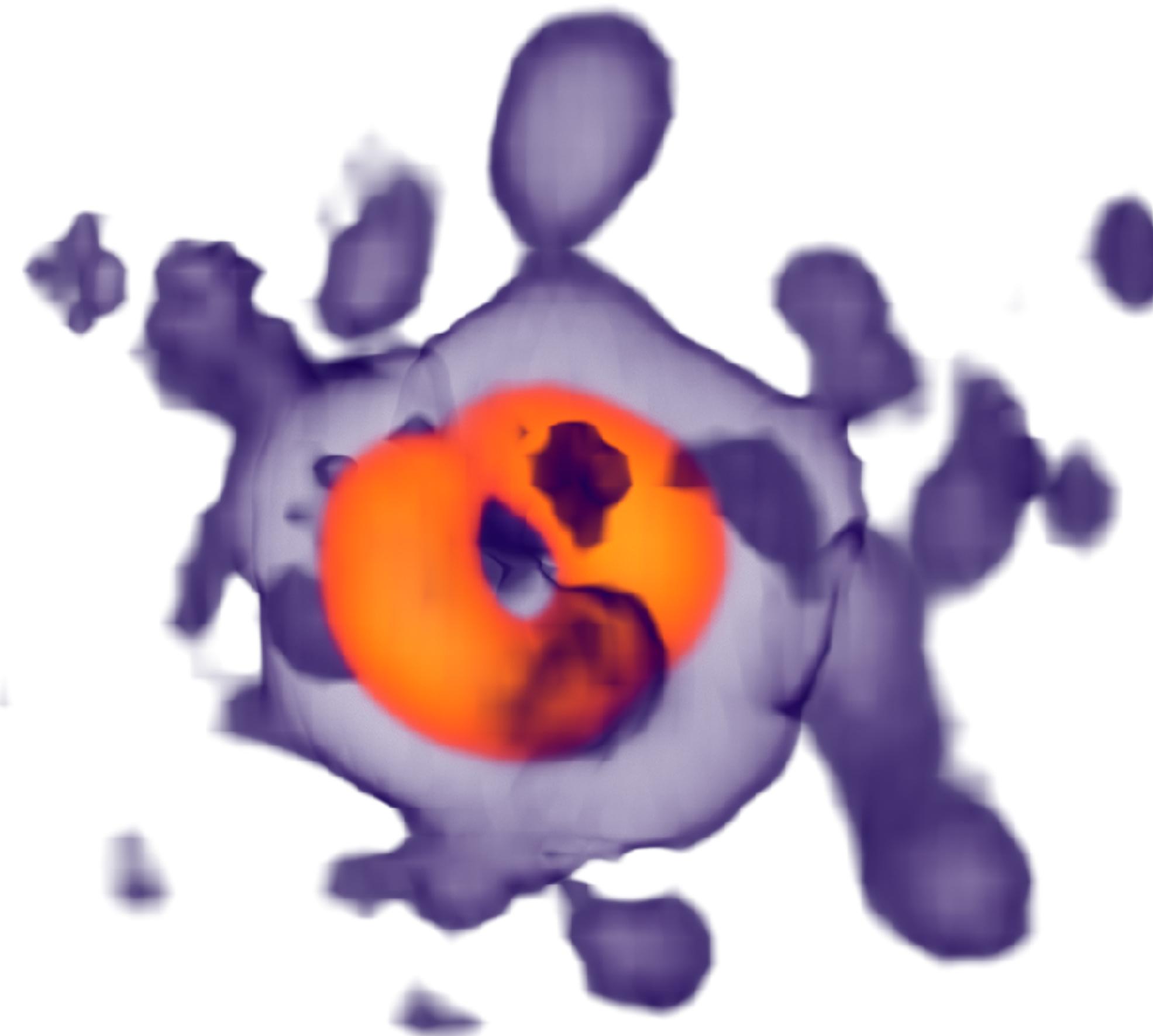
LASY allows leveraging existing openPMD infrastructure to visualise and simulate complex pulses

## Generate

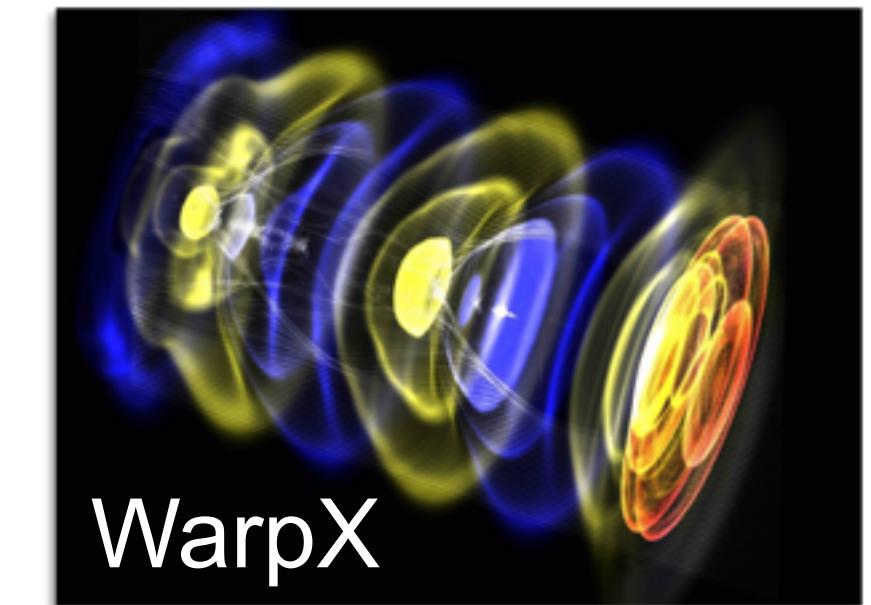


Longman et al, Phys Plasmas **29**, (2022)

## Visualise



## Simulate



## Measure



INSIGHT, Sourcelab

VisualPIC

# Examples

# Getting Started with LASY

## Installation and First Simulation

Installation Instructions

```
pip install lasy
```

```
from lasy.profiles.gaussian_profile import GaussianProfile  
from lasy.laser import Laser
```

```
wavelength      = 800e-9 # Laser wavelength in meters  
polarization    = (1,0)  # Linearly polarized in the x direction  
energy          = 1.5    # Energy of the laser pulse in joules  
spot_size       = 25e-6 # Waist of the laser pulse in meters  
pulse_duration  = 30e-15 # Pulse duration of the laser in seconds  
t_peak          = 0.0    # Location of the peak of the laser pulse in time  
  
laser_profile = GaussianProfile(wavelength,polarization,energy,spot_size,pulse_duration,t_peak)
```

```
dimensions      = 'rt'           # Use cylindrical geometry  
lo              = (0,-2.5*pulse_duration) # Lower bounds of the simulation box  
hi              = (5*spot_size,2.5*pulse_duration) # Upper bounds of the simulation box  
num_points      = (300,500)       # Number of points in each dimension  
  
laser = Laser(dimensions,lo,hi,num_points,laser_profile)
```

```
z_R             = 3.14159*spot_size**2/wavelength # The Rayleigh length  
laser.propagate(-z_R)                         # Propagate the pulse upstream of the focal plane
```

100%  00:00<00:00 [2429.64it/s]

```
file_prefix     = 'test_output' # The file name will start with this prefix  
file_format     = 'h5'         # Format to be used for the output file  
  
laser.write_to_file(file_prefix, file_format)
```

**Import functions and classes**

**Define the laser based on physical characteristics**

**Define the grid initialise the laser object**

**Propagate or manipulate as you want**

**Dump to file**

# Initialisation of Laser Pulse with Experimental Data

## Example

```
● ● ●

Imports
from lasy.profiles.transverse.transverse_profile_from_data import TransverseProfileFromData
from lasy.profiles.longitudinal.longitudinal_profile_from_data import LongitudinalProfileFromData
from lasy.profiles.CombinedLongitudinalTransverseProfile import CombinedLongitudinalTransverseProfile
from lasy.laser import Laser
import numpy as np

Define Laser Parameters
fluence      = np.load('laser_fluence.npy')          # Import the fluence data
calib        = 0.3e-6                                # Camera calibration in um/pixel
temporalData = np.load('temporal_profile.npy')       # Import temporal data
wavelength   = 800e-9                               # Central Wavelength
pol          = (1,0)                                 # Polarisation
laser_energy = 0.5                                  # Laser Energy

Import Experimental Data
Create: Longitudinal Profile
Transverse Profile
Combine them
Initialise Laser Object
Propagate and dump to file

transProf    = TransverseProfileFromData(fluence, (0 , 0), (cols*calib, rows*calib))
longProf     = LongitudinalProfileFromData(temporalData, (-150e-15,150e-15))
profile      = CombinedLongitudinalTransverseProfile(wavelength, pol, laser_energy, longProf, transProf)

lo = (-75e-6,-75e-6,-150e-15) ; hi = (75e-6,75e-6,150e-15) ; npts = (100,100,100)
laser = Laser('xyt',lo,hi,npts,profile)

laser.propagate(-500e-6)
laser.write_to_file('reconstructed')
```

# LASY enables contemporary high quality simulations

Flexible open-source toolkit to simulate realistic laser pulses in most efficient manner



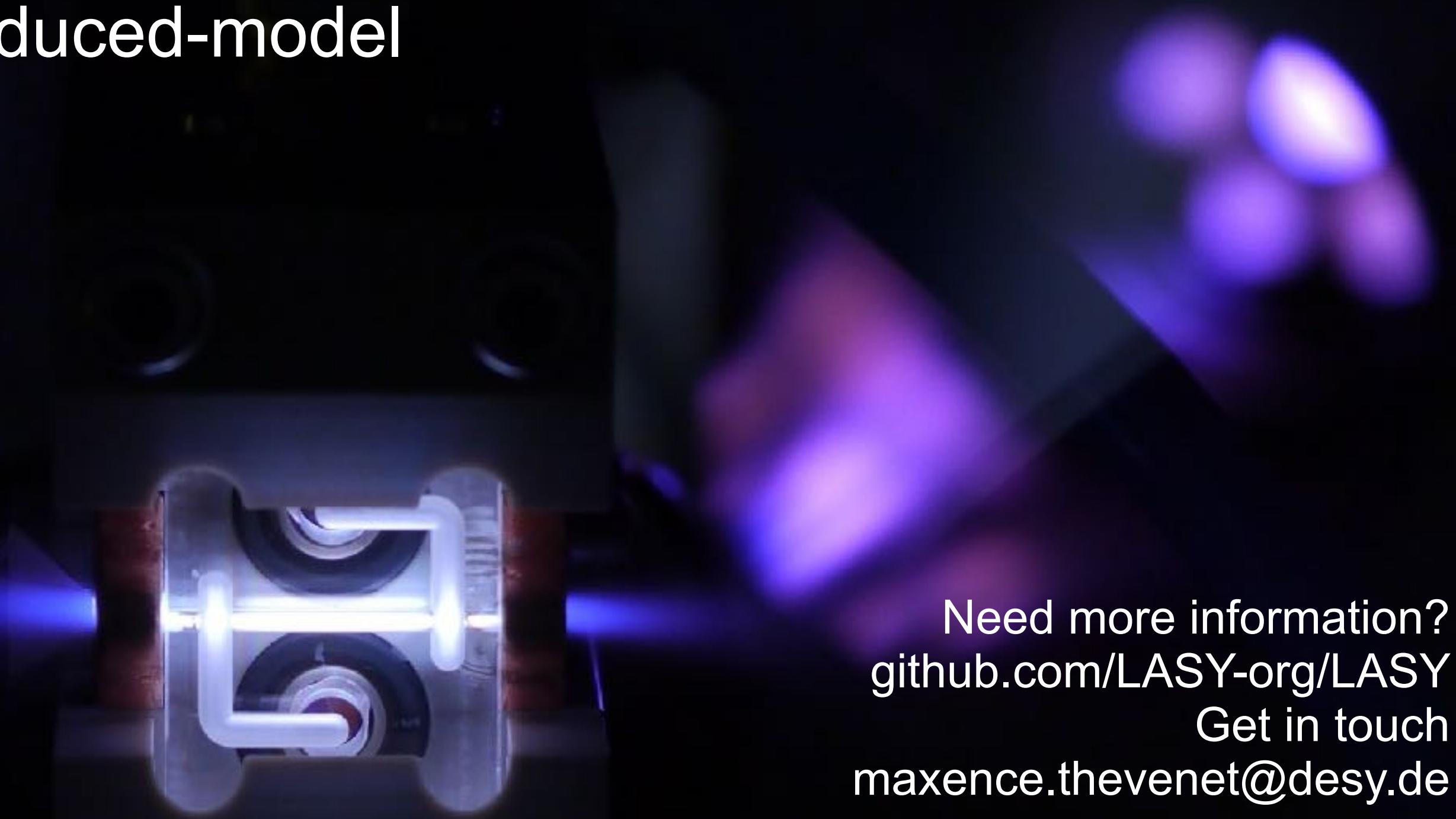
Check out the code

LASY allows

- Creating and manipulating (complex) laser pulses
- Importing and manipulating experimentally measured spatial, temporal or spatio-temporal laser profiles
- Interfacing full EM codes with efficient, reduced-model codes

in a modern, standard-embracing codebase.

Contributions to the code (*very*) welcome!



Need more information?  
[github.com/LASY-org/LASY](https://github.com/LASY-org/LASY)  
Get in touch  
[maxence.thevenet@desy.de](mailto:maxence.thevenet@desy.de)