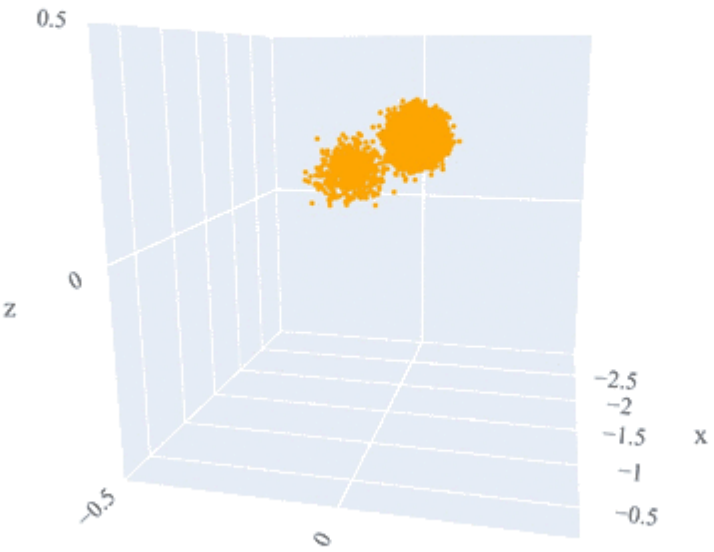
LCODE 3D

A New Open-Source Tool for 3D PWFA simulations



N. Okhotnikov, P. Tuev, I. Kargapolov, K. Lotov, I. Shalimova, A. Sosedkin
Novosibirsk State University

Who are we?

- A team from Novosibirsk
- We developing code to simulate wakefield acceleration (LCODE)
- We using LCODE to simulate experiments of the present and future


Lotov Konstantin


Petr Tuev


Roman Spitsyn


Nikita Okhotnikov


Ivan Kargapolov

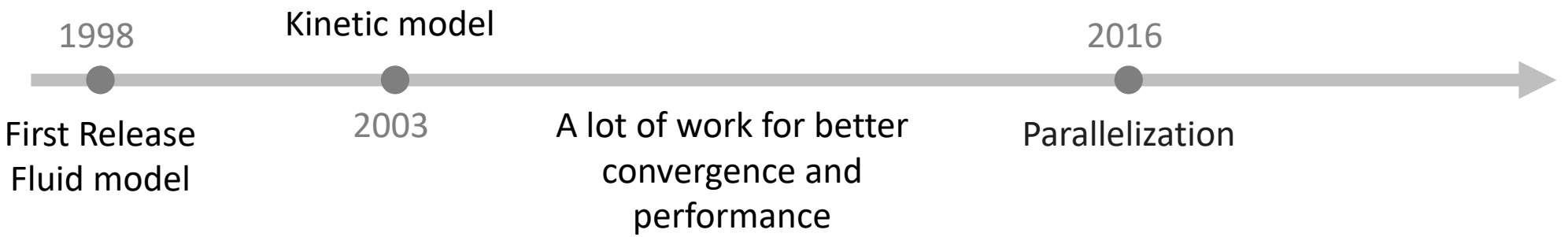

Vladimir Minakov


Irina Shalimova

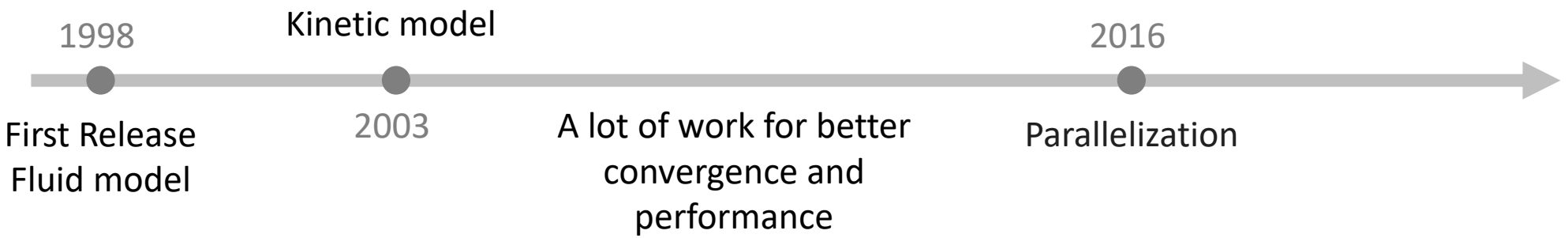

Vlada Yarygova

LCODE

- Has been in use for almost 30 years
- Used for modeling of AWAKE, FACET experiments
- Shows good agreement with experiment



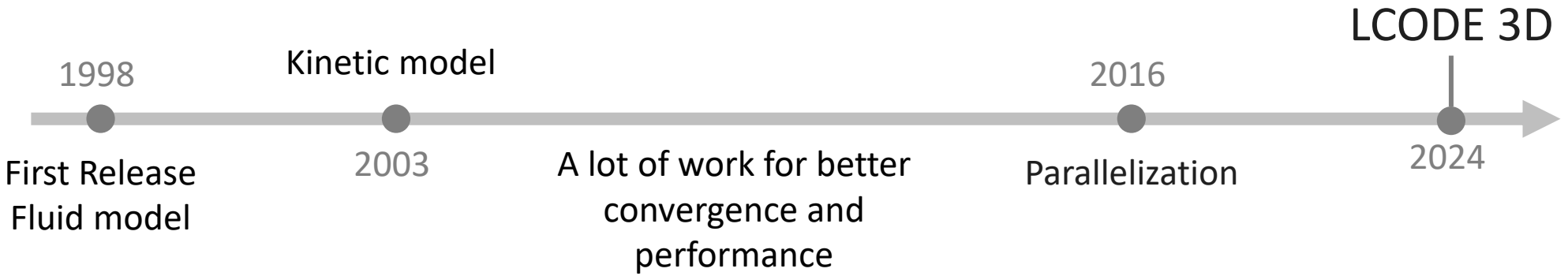
LCODE



But:

- There are new challenges (e.g. 3D)
- The code has become difficult to maintain

LCODE

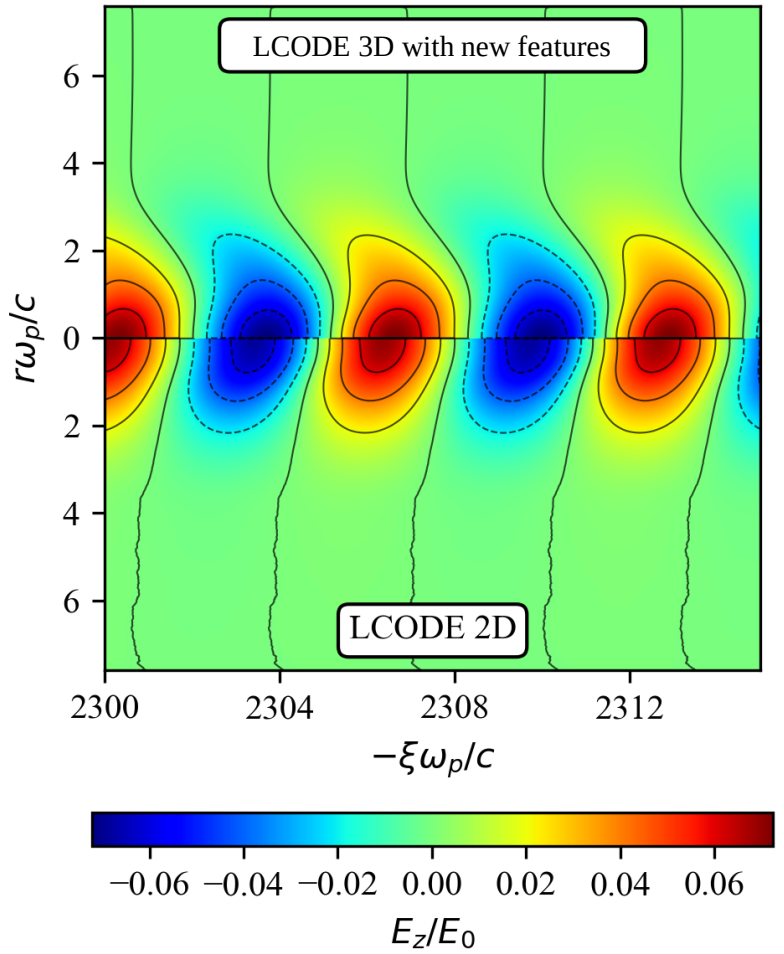


But:

- There are new challenges (e.g. 3D)
- The code has become difficult to maintain

LCODE 3D

- Written in Python
- Supports both 3D and 2D
- 2D works exactly like it used to
- Using NumPy&Numba / CuPy&Numba.cuda
- Improvements in the computing core
- Novel features (next presentation)



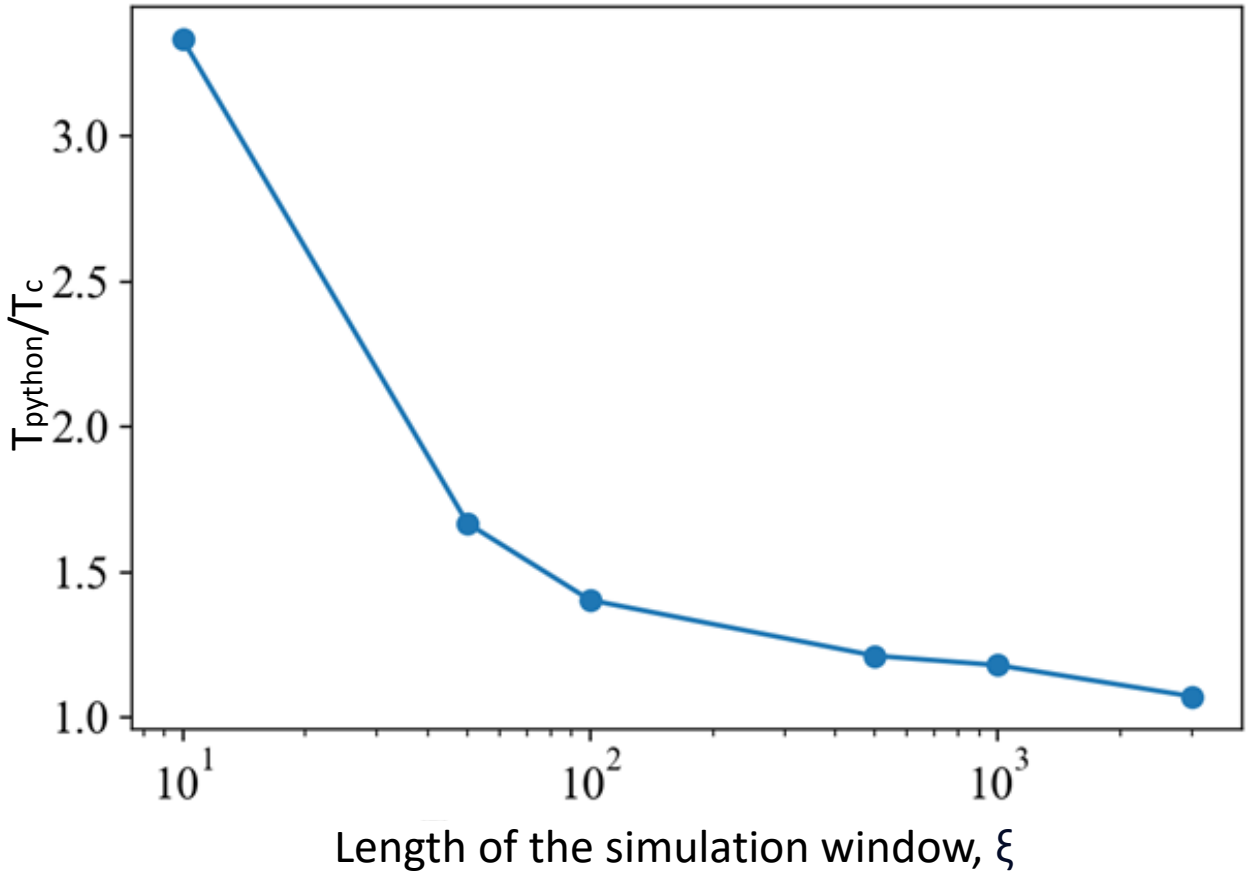
LCODE 3D

- Written in Python

Comparison of simulation time of the version written in C and the current Python version

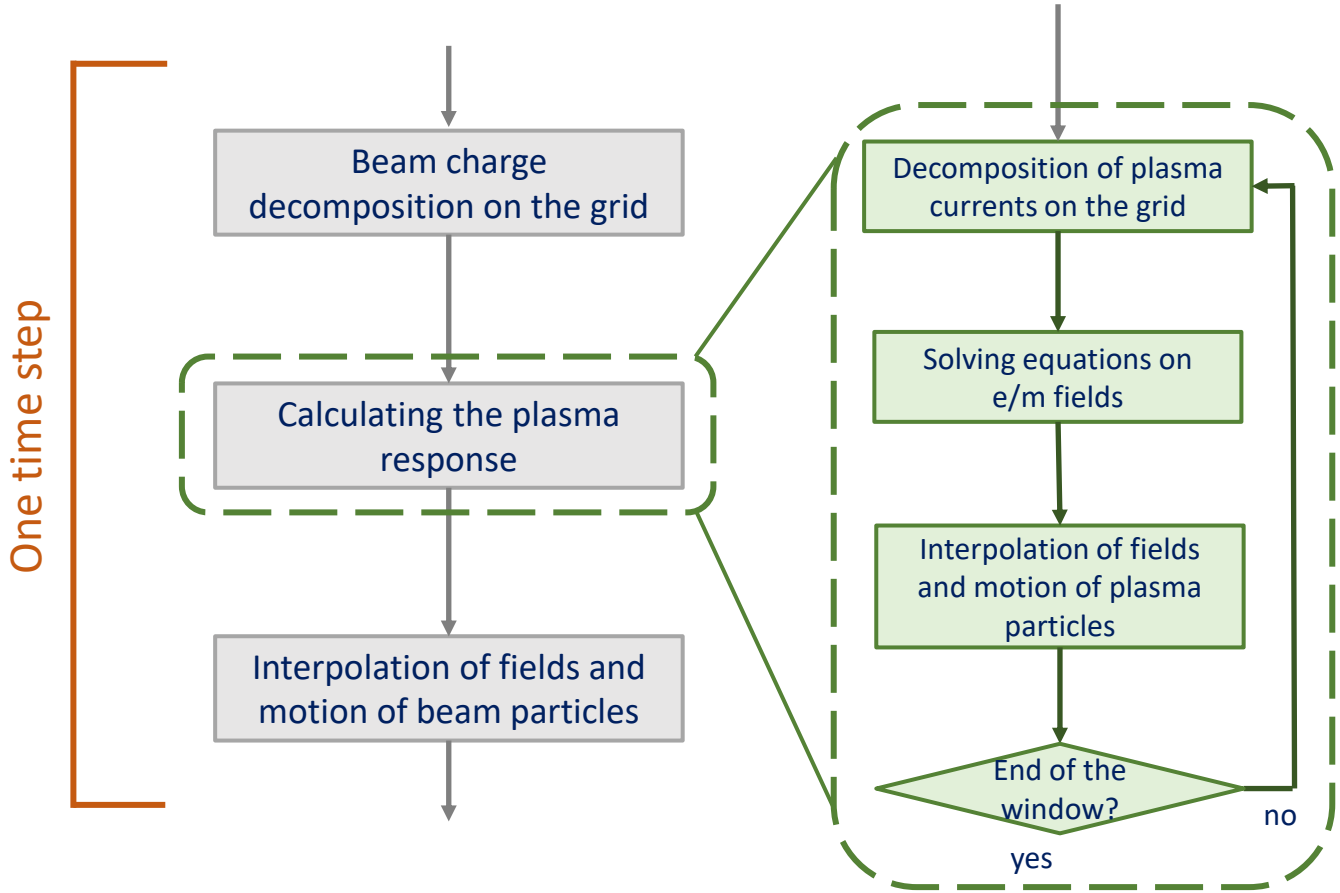
There is a big difference at the beginning due to the fact that JIT compilation takes place

AWAKE length: ≈50000 ξ



LCODE 3D

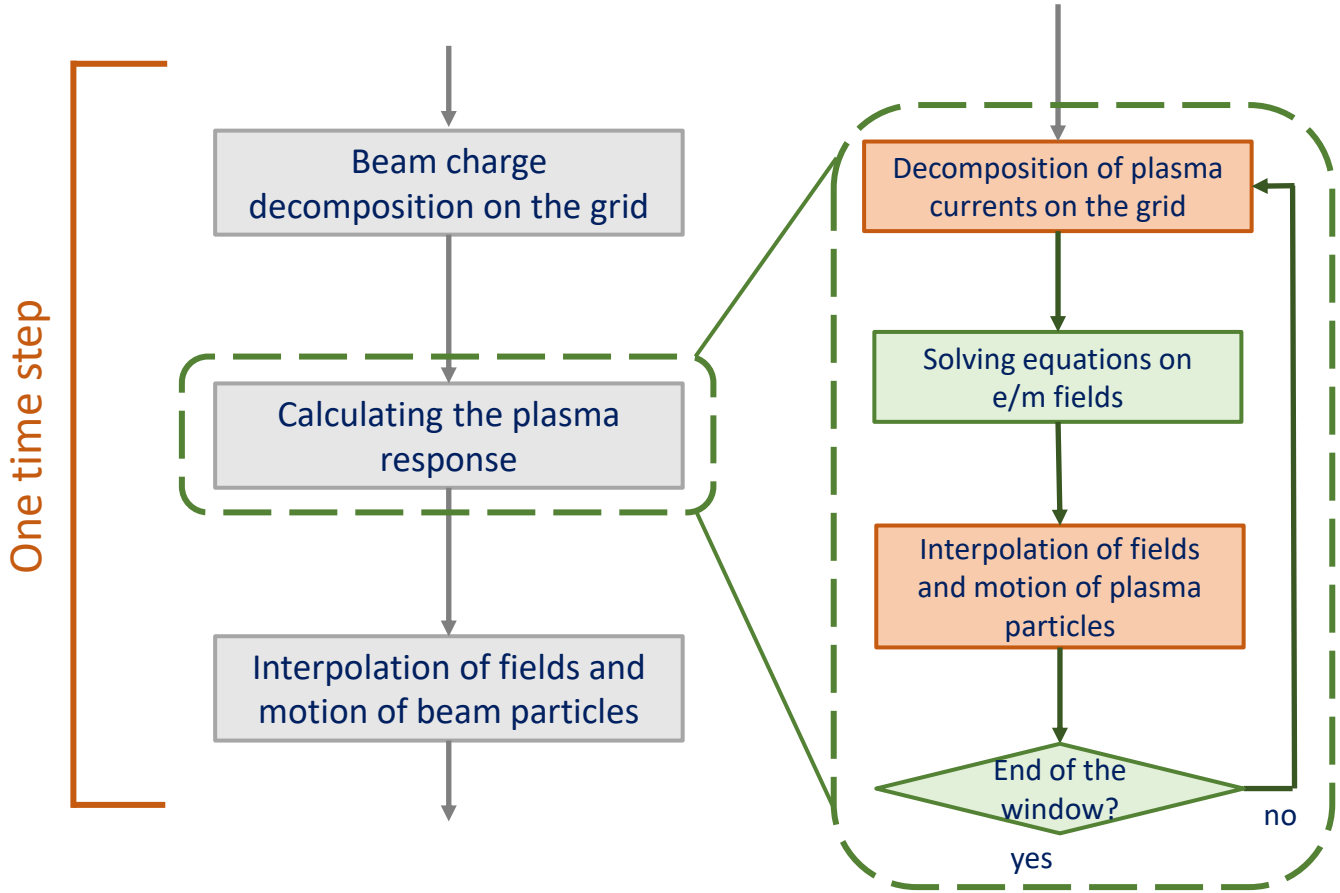
- Improvements in the computing core



LCODE 3D

- Improvements in the computing core

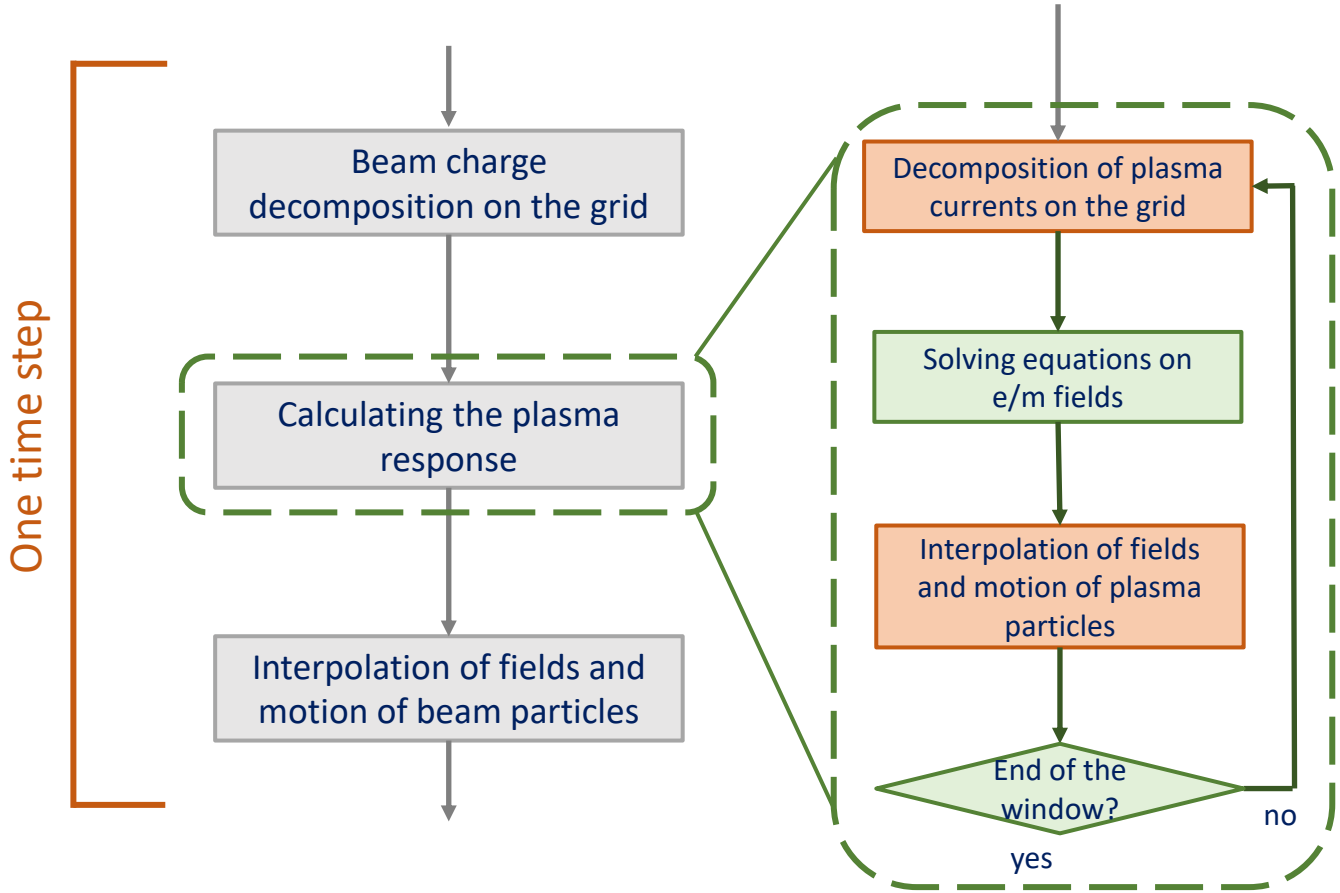
$$R_p(x) = \begin{cases} \frac{1}{h} - \frac{1}{h^3}\left(x^2 + \frac{h^2}{4}\right), & |x| \le \frac{1}{2}h, \\ \frac{1}{2h^3}\left(\frac{3}{2}h - |x|\right)^2, & \frac{1}{2}h < |x| \le \frac{3}{2}h, \\ 0, & |x| > \frac{3}{2}h, \end{cases}$$



LCODE 3D

- Improvements in the computing core

$$R_p(x) = \begin{cases} \frac{115}{192h} - \frac{5x^2}{8h^3} + \frac{x^4}{4h^5}, & |x| \le \frac{1}{2}h, \\ \frac{55}{96h} + \frac{5|x|}{24h^2} - \frac{5x^2}{4h^3} + \frac{5|x|^3}{6h^4} - \frac{x^4}{6h^5}, & \frac{1}{2}h < |x| \le \frac{3}{2}h, \\ \frac{1}{24h}\left(\frac{|x|}{h} - \frac{5}{2}\right)^4, & \frac{3}{2}h < |x| \le \frac{5}{2}h, \\ 0, & |x| > \frac{5}{2}h, \end{cases}$$

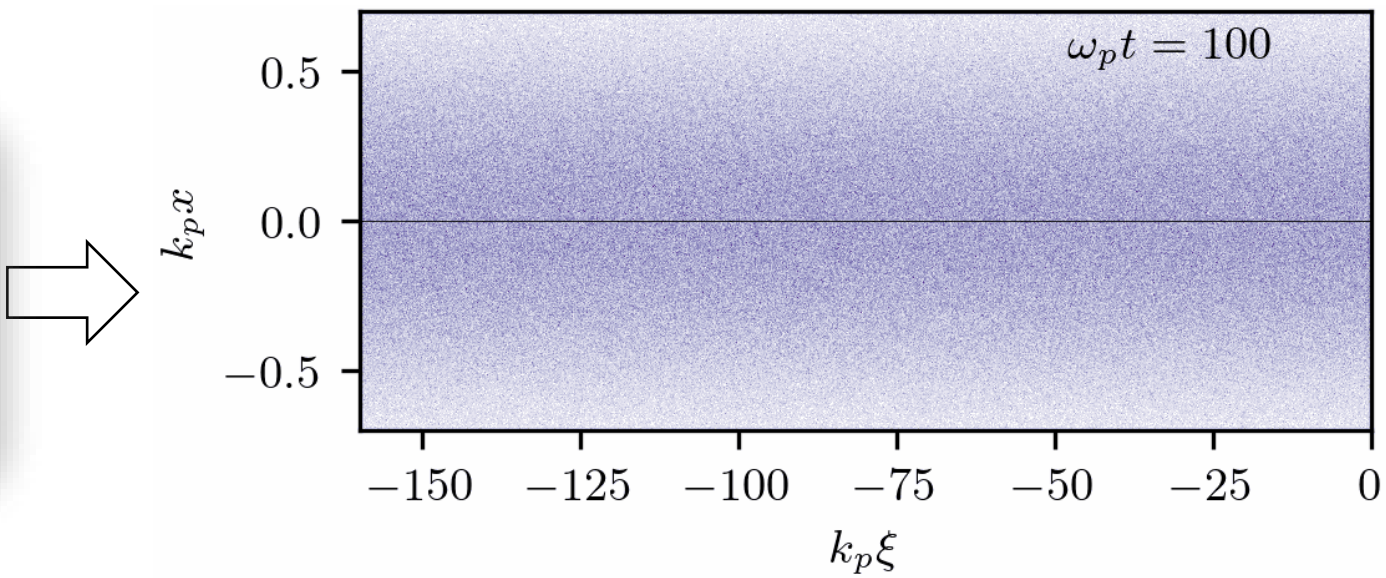


Easy to start

```
pip install lcode
python –m lcode get ssm
python run.py
```

Easy to start

pip install lcode
python –m lcode get ssm
python run.py



Easy to setup

- The simulation is easy to customize, due to the user-friendly interface
- The *run.py* on the right is completely ready to run

run.py

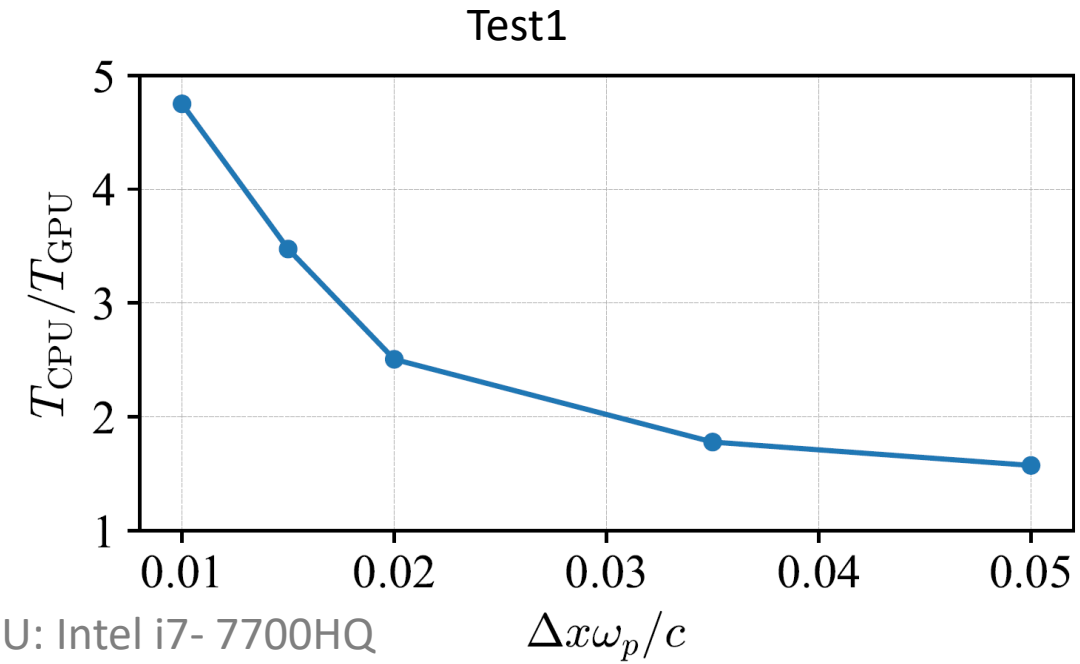
```
from lcode import Simulation
config = {
    'geometry': '3d',
    'processing-unit-type': 'cpu',
    'window-width-step-size': 0.0025,
    'window-width': 16,
    'window-length': 3002.506628274,
    'xi-step': 0.0025,
    'plasma-particles-per-cell': 1
}

beam = {'current': 0.05, 'particles_in_layer': 5000,
        'default' : {'length' : 5.013256548}}

sim = Simulation(config=config, diagnostics=[],
                 beam_parameters=beam)
sim.step()
```

Parallel computing on GPU

- Just change 1 word and get the power of the GPU (only for 3d)

```
config = {
    'geometry': '3d',
    'processing-unit-type': ‘cpu’,
    ...
```

```
config = {
    'geometry': '3d',
    'processing-unit-type': ‘gpu’,
    ...
```

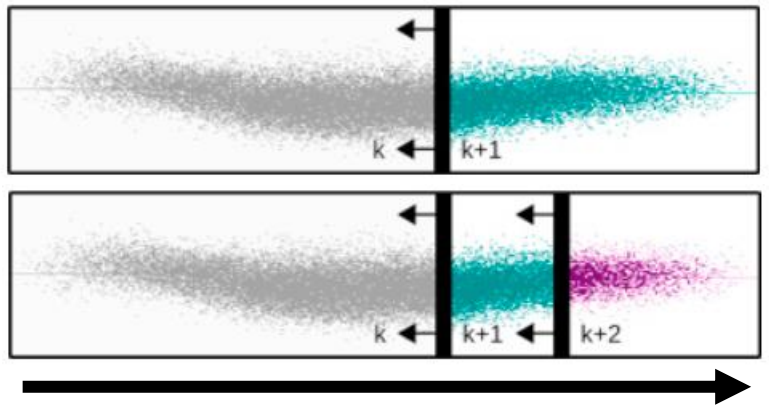


CPU: Intel i7- 7700HQ
GPU: NVIDIA GeForce GTX 1050 Ti

Acceleration depends on the task and GPU
Usually it's faster

Parallel computing on CPUs

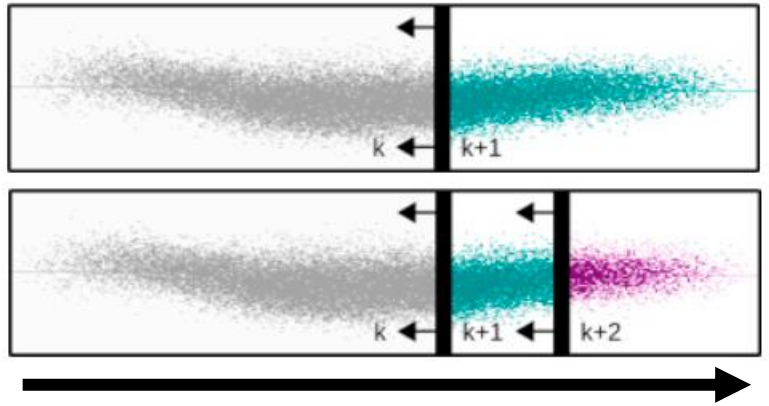
- Since the simulation window moves at the speed of light, information is only transmitted from the beam head to the tail
- This allows us to parallelize the calculation in the time coordinate



Parallel computing on CPUs

- LCODE supports this using MPI
- Works in both 2D and 3D
- Multi-GPU not supported yet

```
mpiexec –n N python run.py
# calculates N time steps in parallel
```

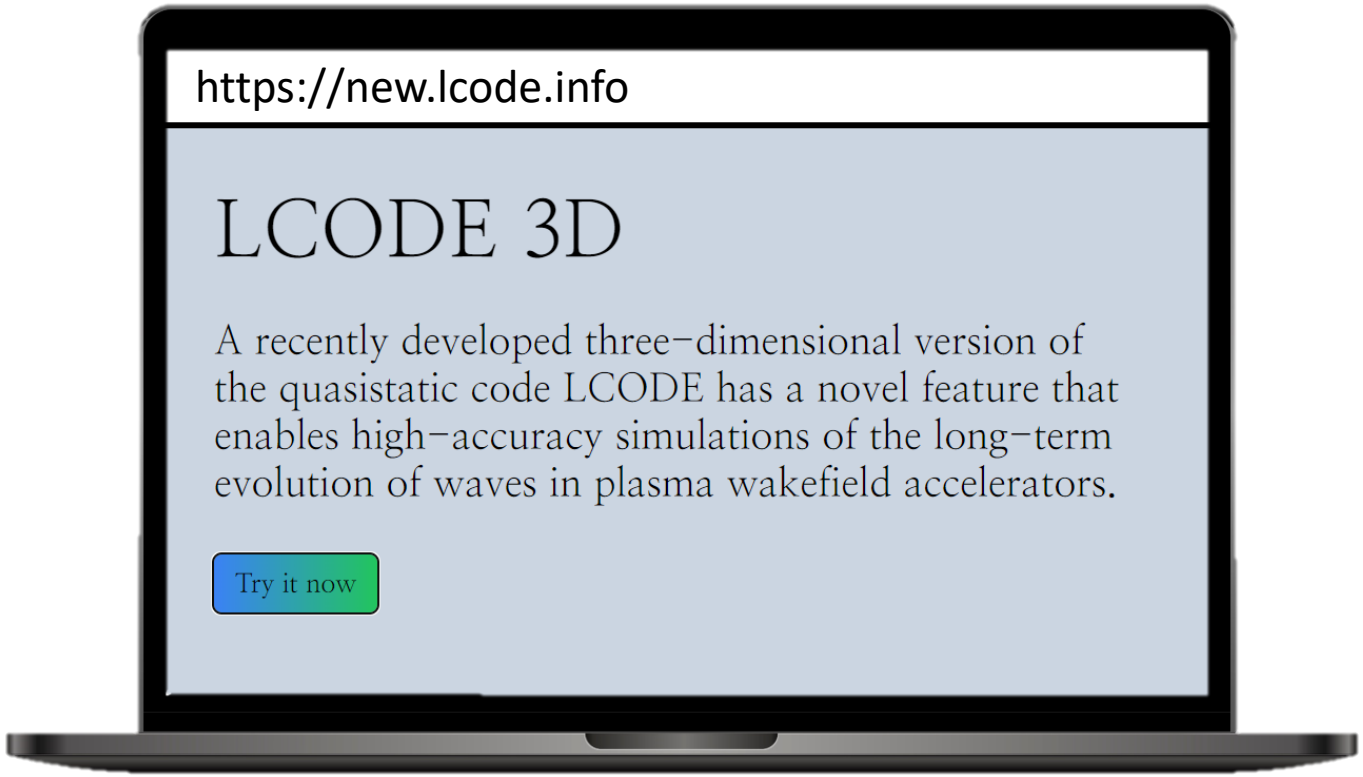


- This allows to achieve computation speedup up to N times

OpenSource

- LCODE is available on Github
- You can influence the development of the code by creating an issue
- You can propose your changes by creating a PR

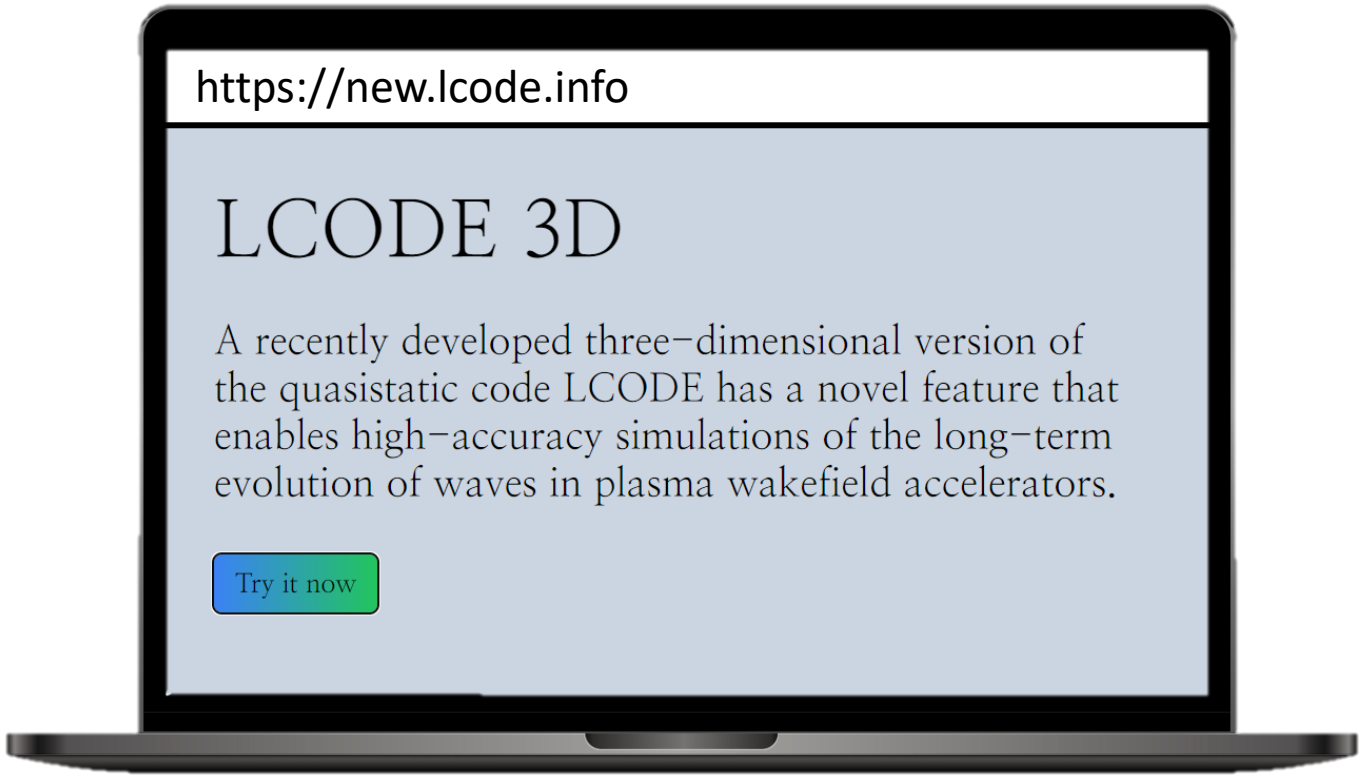
- We are looking forward to community support.



LCODE

new.lcode.info

lcodePy-team/lcodePy



LCODE

new.lcode.info

lcodePy-team/lcodePy

The release of version 1.0.0
will be by 03/27/24
Beta version is already available