U.S. Particle Accelerator School
Education in Beam Physics and Accelerator Technology

Simulations of Beam and Plasma Systems

Steven M. Lund, David Bruhwiler, Rémi Lehe, Jean-Luc Vay and Daniel Winklehner

Sponsoring University: Old Dominion University

Hampton, Virginia – January 15-26, 2018

Example Warp Simulations – part 1

Jean-Luc Vay

Lawrence Berkeley National Laboratory

This material is based upon work supported by the U.S. Department of Energy, Office of Science and the National Nuclear Security Administration, under Award Number(s) DE-AC02-05CH11231 and 17-SC-20-SC. Used resources of the National Energy Research Scientific Computing Center.

U.S. DEPARTMENT OF ENERGY
Office of Science

EXASCALE COMPUTING PROJECT

NERSC

BERKELEY LAB
Lawrence Berkeley National Laboratory

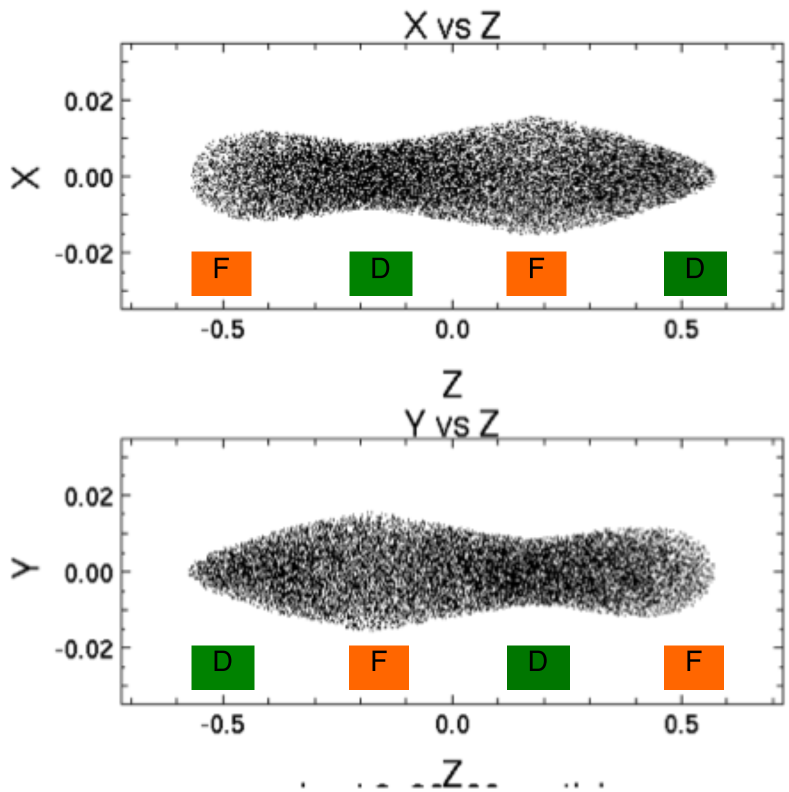
Outline

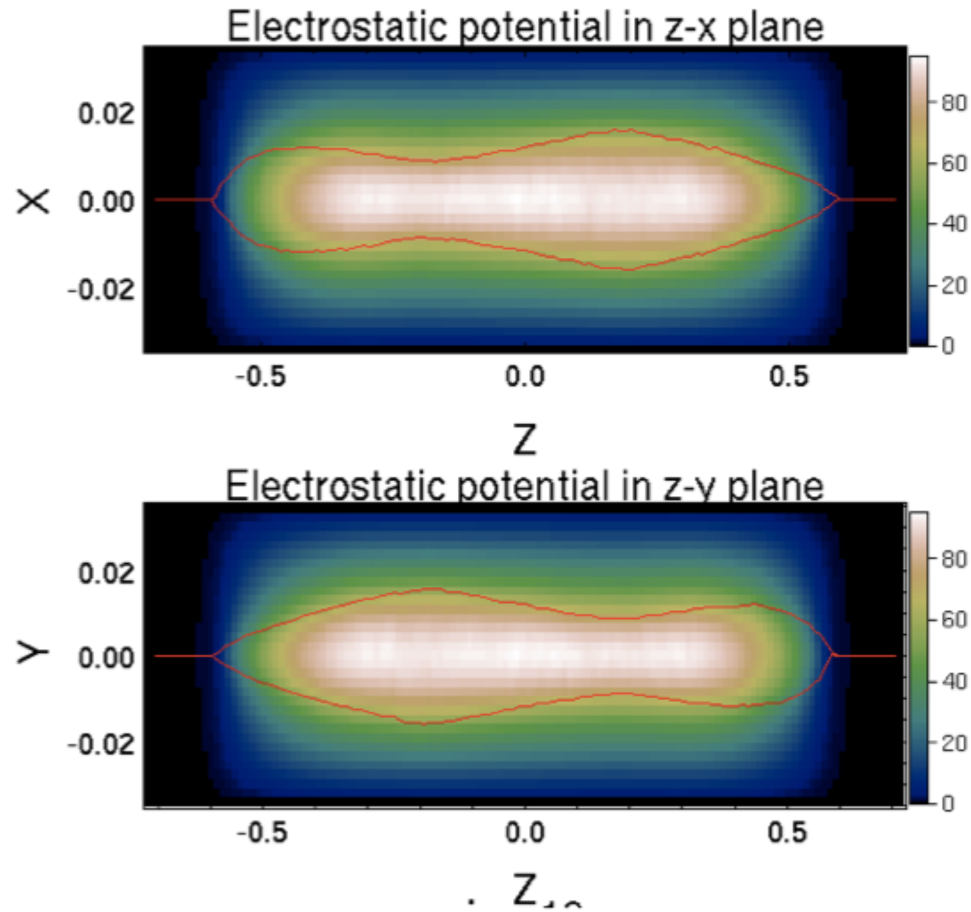
- Quadrupole transport 3D
- Quadrupole transport 2D-XY
- Plasma acceleration

Examples are in simbeamplasma18/warp_scripts from github repository :

- git clone https://github.com/uspas/simbeamplasma18.git.

Quadrupole transport – 3D





File FODO3D.py - basic 3D simulation of an ion beam in a periodic FODO lattice:

- Sets up a periodic FODO lattice and creates a beam that is matched to the lattice.
- The beam is propagated one lattice period.

FODO3D: tasks

① Into FODO3D, open FODO3D.py

② Execute file: “python –i FODO3D.py”

③ Open cgm file and explore:

  a) “gist FODO3D.000.cgm &”

④ Compare the slices emittance diagnostics to the whole emittance diagnostic. Can you explain the differences (i.e. one is constant, the other oscillates)?

⑤ Read input script and try to understand every command

⑥ Change ‘w3d.distrbtn = "semigaus”’ to ‘w3d.distrbtn = ”KV"’ ; rerun & observe

⑦ Change ‘w3d.distr_l = "gaussian”’ to ‘w3d.distr_l = ”neuffer”’ ; rerun & observe

⑧ Insert “beam.x0 = beam.a0/2” on the line following “beam.a0 = …”; rerun & observe

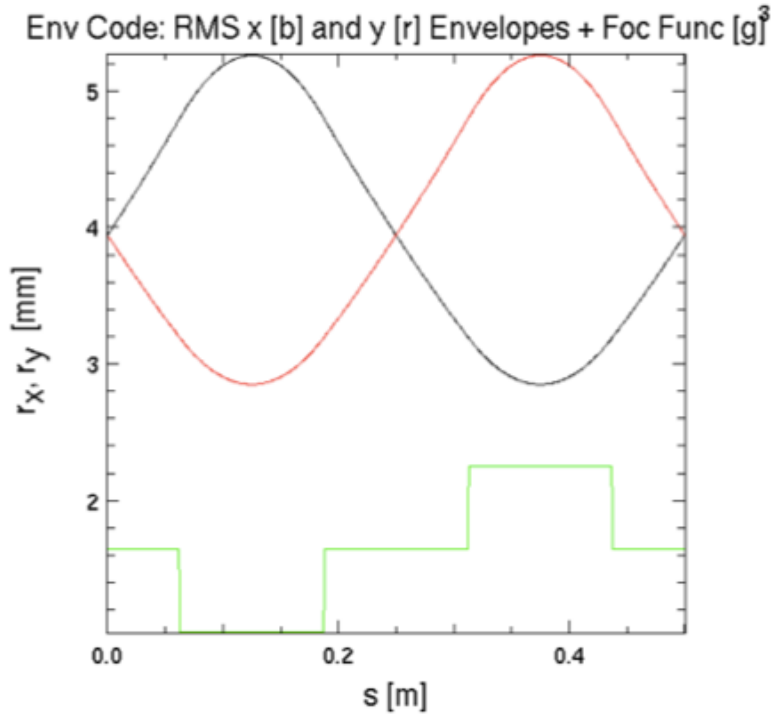
⑨ Check that you have the “ffmpeg” software installed: “which ffmpeg”

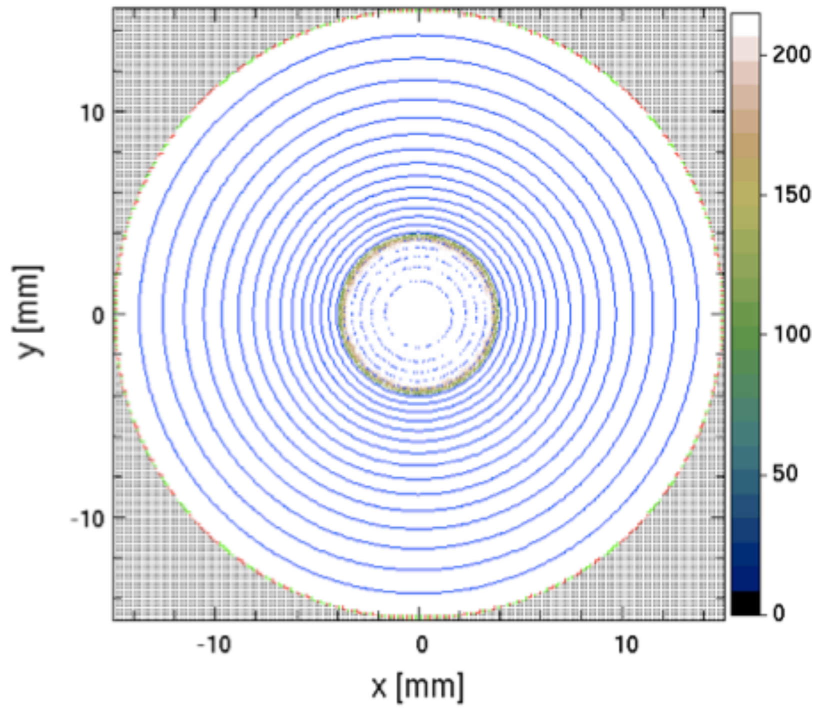
- If not, download and install ffmpeg

⑩ Change “l_movieplot = False” to “l_movieplot = True” & rerun

- If all goes well, after a few minutes, you should have a movie “movie.mp4”

Quadrupole transport – XY





File xy-quad-mag-mg.py:

nonrelativistic Warp xy slice simulation of a K^+ ion beam with intense space-charge focused by a hard-edge magnetic quadrupole doublet focusing lattice.

xy-quad-mag-mg: tasks

① Into XY-quad, open xy-quad-mag-mg.py

② Execute file: “python –i xy-quad-mag-mg.py”

③ Open cgm file and explore: “gist xy-quad-mag-mg.000.cgm &”

④ Read input script and try to understand every command.

⑤ Comment ‘w3d.distrbtn = "SG”’ and uncomment ‘w3d.distrbtn = ”KV”’, rerun and compare to results using the KV vs SG distributions.

⑥ Change the initial emittance “emit = 10.e-6” to “emit = 10.e-7”, rerun and observe effect on matching and emittance preservation.

⑦ Change switch “l_automatch = False” to “l_automatch = True”, rerun and observe difference with previous run.

⑧ Change n_grid=200 to 400, rerun and observe differences. Can you explain?

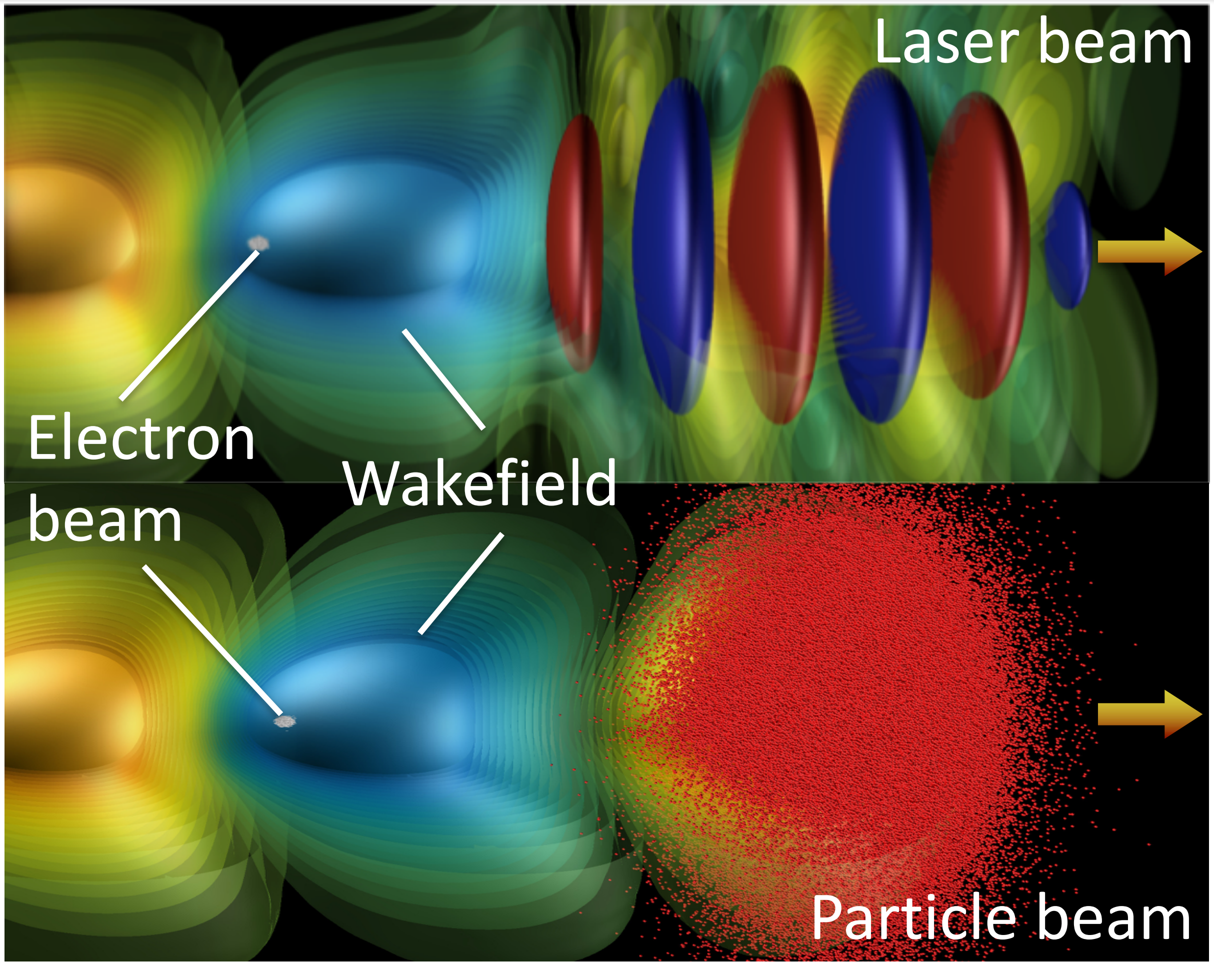
⑨ With the simulation back at the python prompt, type ‘dump()’, then run for another 500 steps: “step(500)”.

⑩ In another terminal, start python and type:

- from warp import *
- restart(‘xy-quad-mag-mg001000.dump’)
- step(500)

Reopen “xy-quad-mag-mg.000.cgm” and compare to “xy-quad-mag-mg.001.cgm”.

Plasma acceleration



Scripts lpa_script.py, lpa_script_2d.py, pwfa_script.py – basic plasma acceleration runs:

- Generate plasma, laser or beam driver, and injected electron beam and follow self-consistent evolution.

Plasma acceleration tasks

① In Plasma_acceleration, open LWFA_3D/lwfa_3d.py and PWFA_3D/pwfa_3d.py

② Execute the files "python –i lwfa_3d.py" and "python –i pwfa_3d.py" separately.

③ It takes some time to run. While it runs, you may open periodically the cgm files ***_script.000.cgm and see the progress. In the meantime, also go through the input scripts and try to understand all the commands.

④ At the end of the run, a plot displays the energy of the accelerated beam versus z.

⑤ Explore data using the OpenPMD notebook viewer by running 'openPMD_notebook' in the run directory. While exploring data, run 2000 additional time steps.

⑥ In plasma_acceleration, open LWFA_2D/lwfa_2d.py and execute.

⑦ Open cgm file and explore:

  a) "gist lpa_basic_2d.000.cgm &"

⑨ Read input script and try to understand every command.

⑩ Run the script ptime displaying the history of the elapsed time and the time per step. Observe the spikes from the diagnostics.