U.S. Particle Accelerator School

Education in Beam Physics and Accelerator Technology

Self-Consistent Simulations of Beam and Plasma Systems

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

Colorado State U., Ft. Collins, CO, 13-17 June, 2016

W3. Examples Warp Simulations

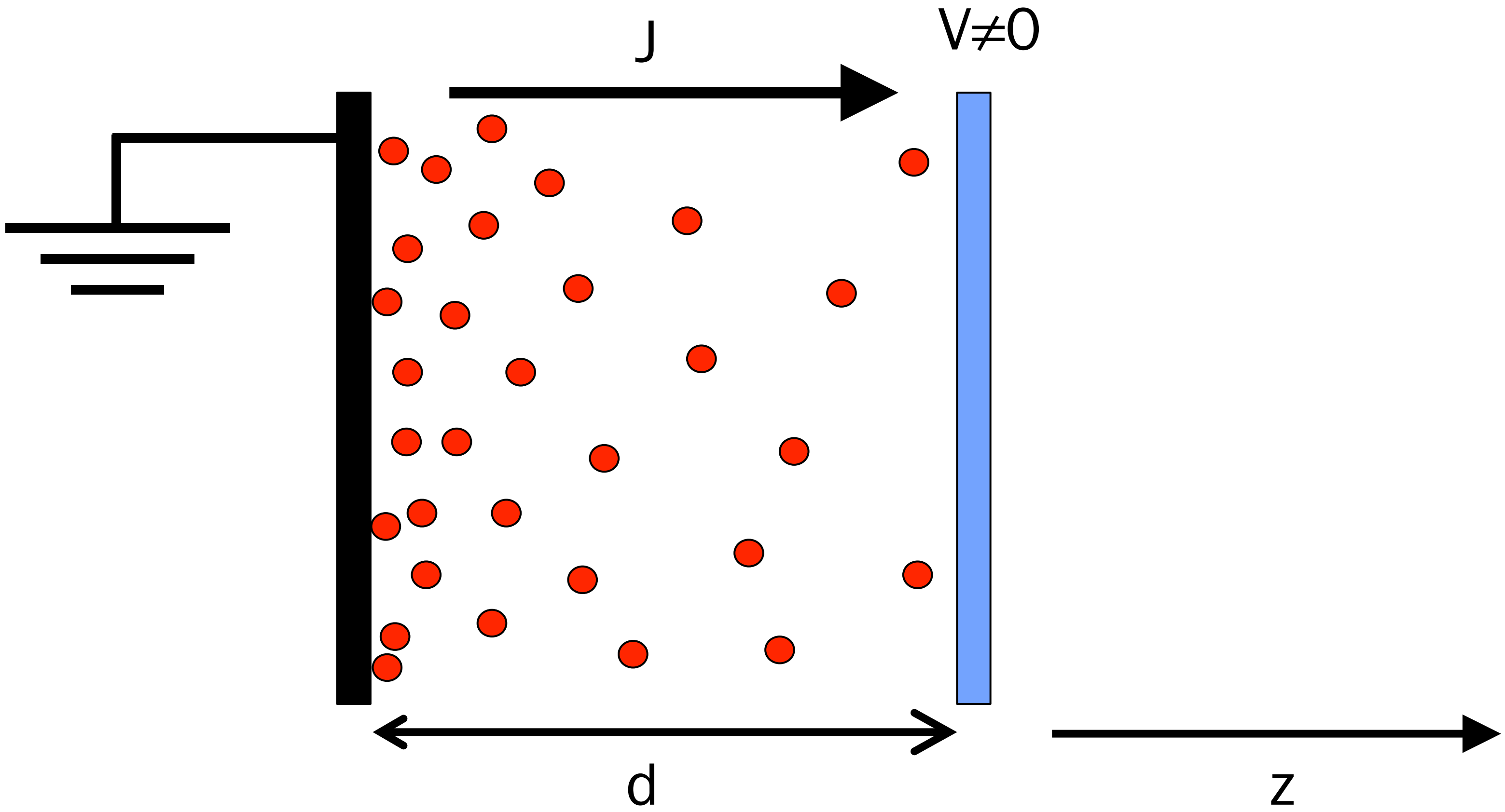
Jean-Luc Vay

Lawrence Berkeley National Laboratory

Outline

- Emission between parallel plates
- Pierce diode
- Quadrupole transport
- Solenoid transport
- Plasma acceleration

Emission between parallel plates

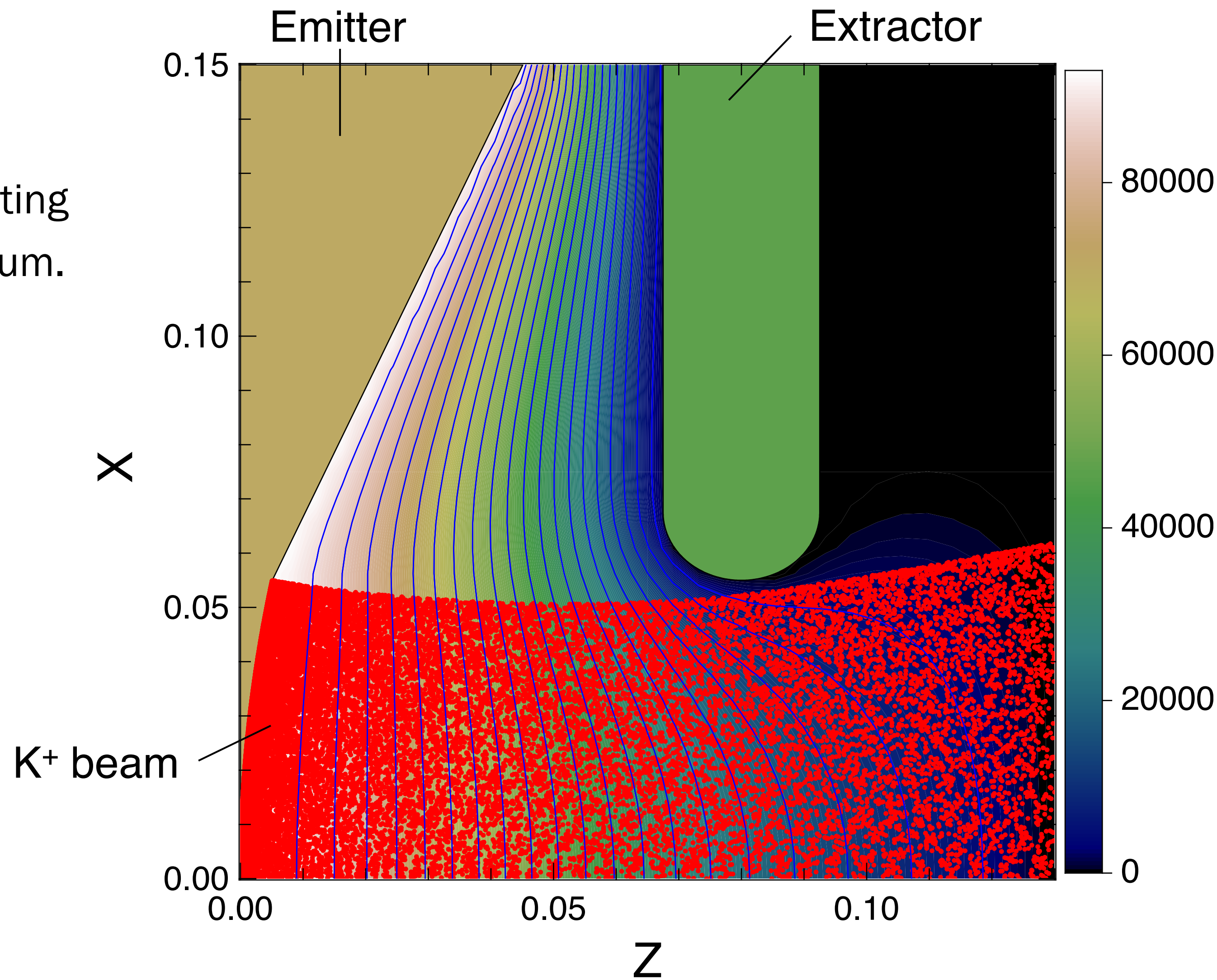


Was given as a problem yesterday. Any question?

Pierce diode: intro

File Pierce_diode.py

Hot plate source emitting singly ionized potassium.



Pierce diode: tasks

① Open Pierce_diode.py

② Execute file: "python –i Pierce_diode.py"

③ Open cgm files and explore:

   a) "gist Pierce_diode.000.cgm &"

   b) "gist current.cgm &"

④ Read input script and try to understand every command

⑤ Comment "w3d.solvergeom = w3d.rzgeom", uncomment "w3d.solvergeom = w3d.xyzgeom" and rerun; observe longer runtime but similar result

⑥ Reverse to RZ geometry

⑦ Set "steady_state_gun=True" and rerun. Simulation is now generating traces, converging to steady-state solutions faster than with time-dependent mode.

⑧ Set "w3d.l_inj_regular = True", "top.npinject = 15" and rerun with regularly spaced traces. This option can be used to enable clean and fast convergence to steady-state.

⑨ Change "diode_current = pi*source_radius**2*j" to "0.5*pi*source_radius**2*j", then "2*pi*source_radius**2*j" and rerun each time. What do you observe?

Pierce diode: tasks

⑨ Go back to original settings

- steady_state_gun=False
- w3d.l_inj_regular = False
- top.npinject = 150
- diode_current = pi*source_radius**2*j

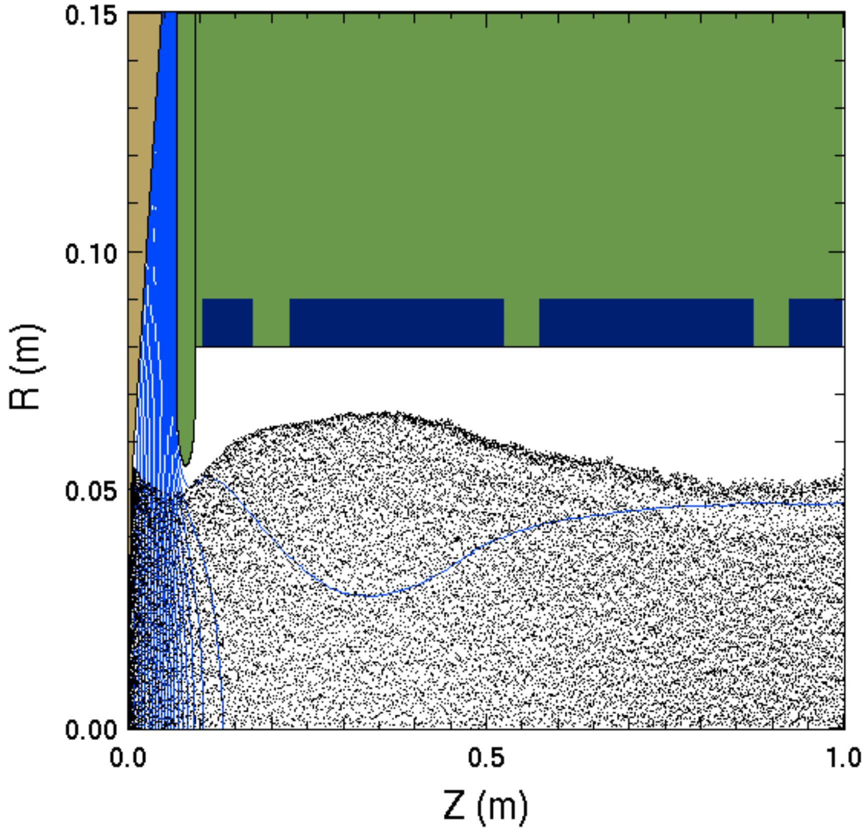
then change

- beamplots(False) ➔ beamplots(True)
- top.inject=1 ➔ top.inject=2 so that extracted current is at Child-Langmuir limit to given voltage

Rerun. Open the latest cgm file, page through and observe how the head of the beam has a larger current and touches the extractor. Why?

⑩ Set "l_constant_current = True" and rerun, observing how the injected current is now constant. Also observe the history of the applied voltage versus time.

Solenoid transport



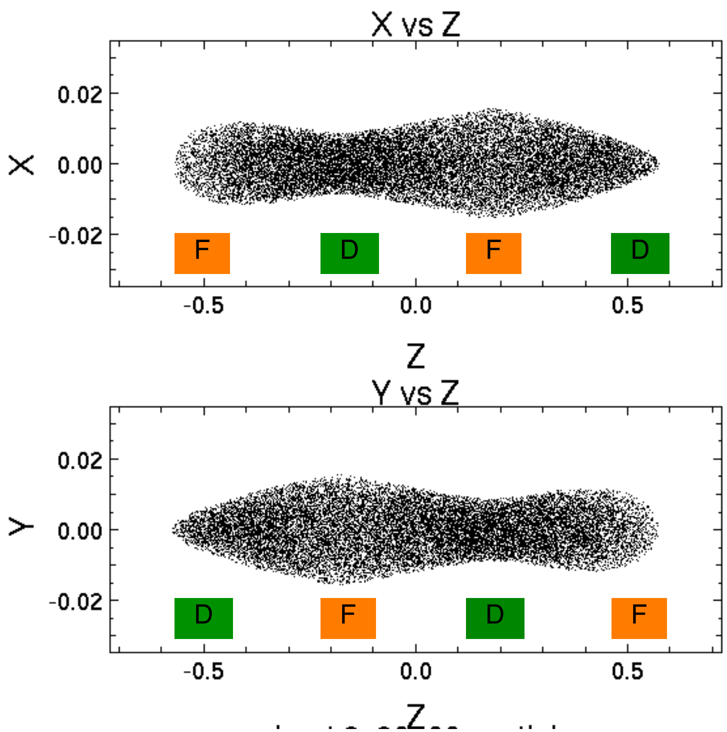
File Solenoid_transport.py:

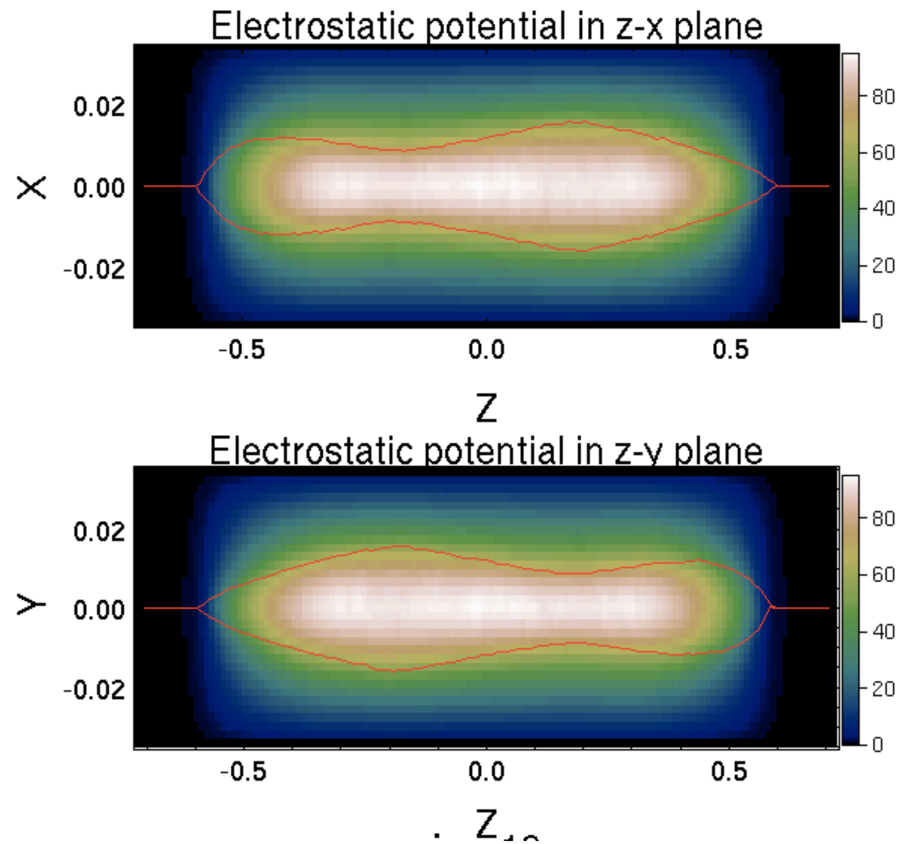
- Example Pierce diode with subsequent solenoid transport.
- Hot plate source emitting singly ionized potassium.

Solenoid transport: tasks

① Open Solenoid_transport.py
② Execute file: "python –i Solenoid_transport.py"
③ Open cgm file and explore:
   a) "gist Solenoid_transport.000.cgm &"
④ Read input script and try to understand every command
⑤ Change "l_solenoid = False" to "l_solenoid = True". Rerun.
⑥ Select window(1)
⑦ Type "fma()" to start next plot from empty page.
⑧ Type "rzplot(9)" to plot RZ view of beam, pipe and solenoids in upper half.
⑨ Type "ppzvtheta(10)" to plot particle projections of azimuthal velocity versus z.
⑩ Notice the correlations between the maximums of the azimuthal velocity and the positions of the solenoids.

Quadrupole transport – 3D





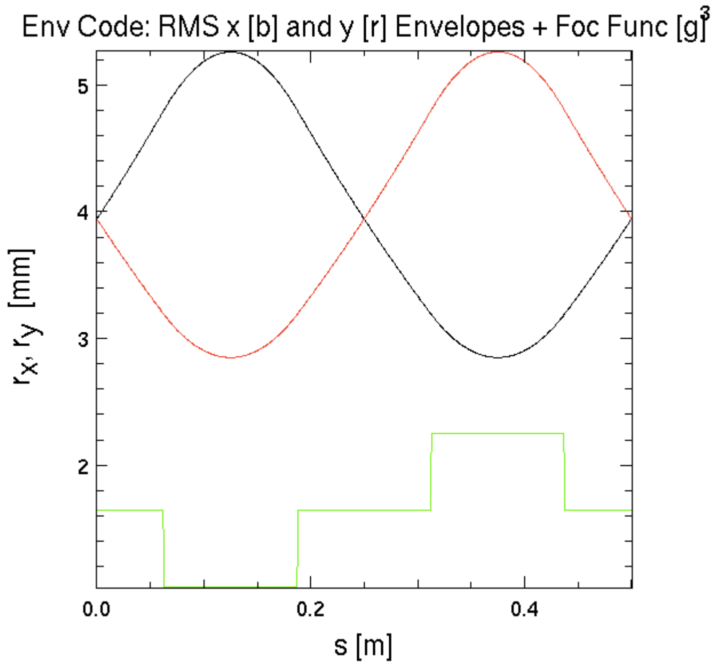
File FODO3D.py - basic 3D simulation of an ion beam in a periodic FODO lattice:

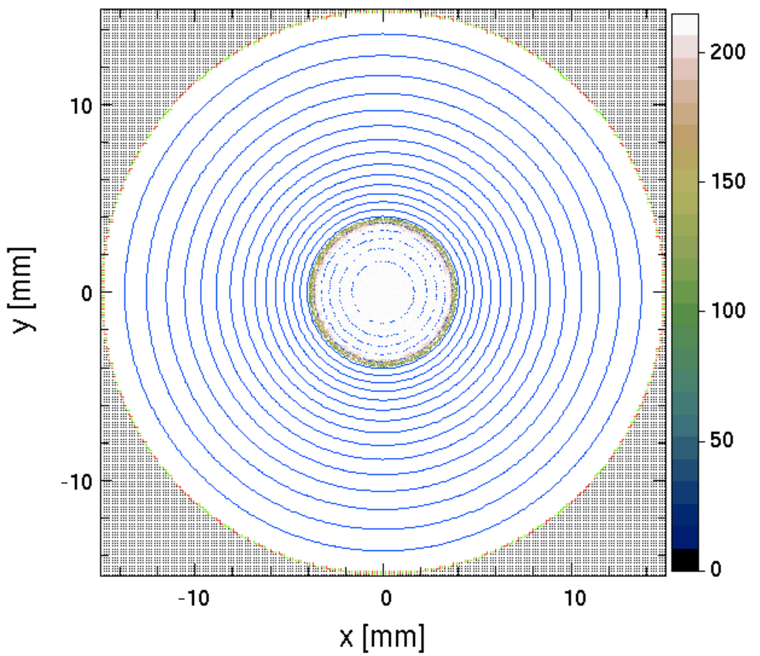
- Sets up a periodic FODO lattice and creates a beam that is matched to the lattice.
- The beam is propagated one lattice period.

FODO3D: tasks

① 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 Open xy-quad-mag-mg.py

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

③ Open cgm file and explore:

   a) “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.

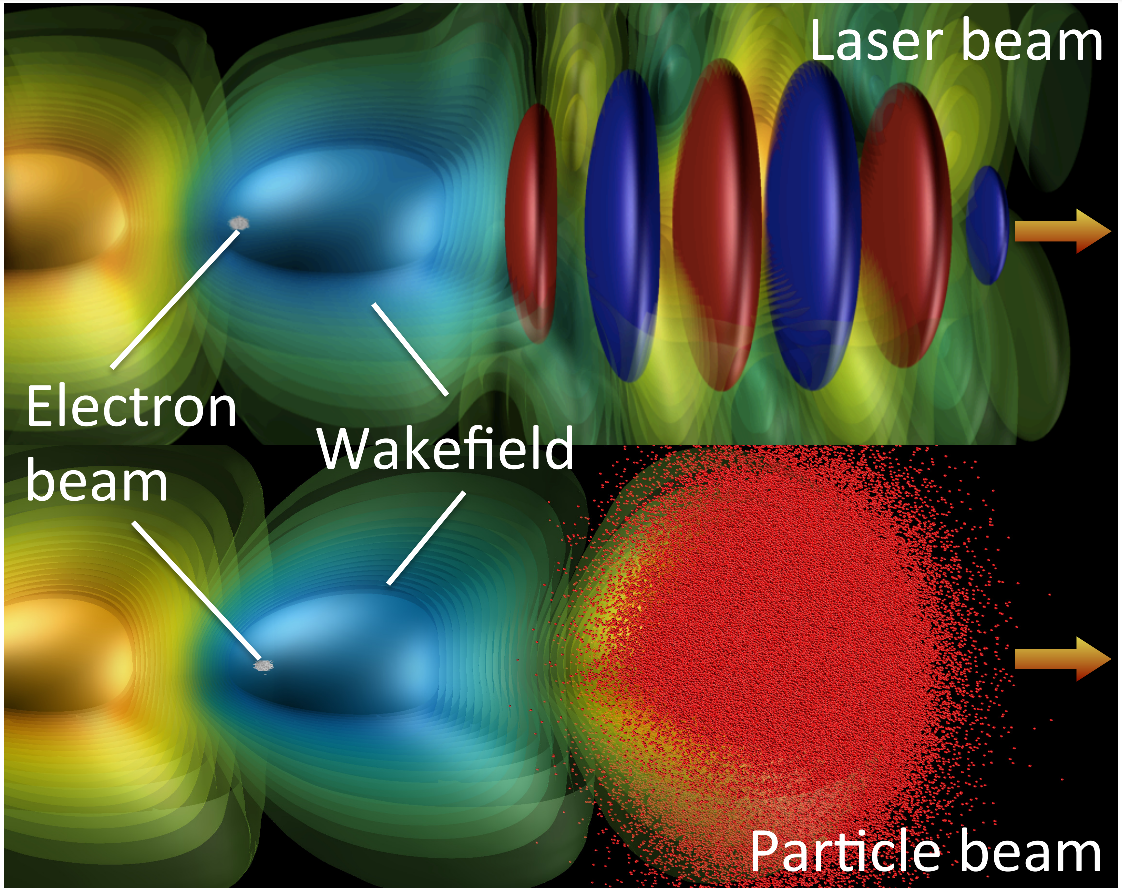
⑧ With the simulation back at the python prompt, type ‘dump()’, then run for another 500 steps: “step(500)”.

⑨ 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 Open lpa_script.py: the first command reads “from warp_init_tools import *”

- the warp_init_tools package contains utility subroutines for easy setup of lasers and continuous injections of plasmas.

② Download the archive
https://github.com/RemiLehe/uspas_exercise/raw/master/warp-init-tools.tar
then execute:

- untar the file: “tar –xvf warp-init-tools.tar”
- cd warp-init-tools
- python setup.py install

③ Execute the files “python –i lpa_script.py” and “python –I pwfa_script.py” separately.

④ 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 and try to understand all the commands.

⑤ At the end of the run, a plot display the energy of the accelerated beam versus z.

⑥ Install OpenPMD notebook viewer and start. While exploring data, run 2000 additional time steps.

Laser plasma acceleration tasks

⑦ Open the file lpa_script_2d.py and execute

⑧ 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.