

**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 2

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.

## Outline

- Emission between parallel plates
- Pierce diode
- Solenoid transport

Examples are in simbeampasma18/warp\_scripts from github repository :

- git clone <https://github.com/uspas/simbeampasma18.git>

## 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.

File Pierce\_diode.py

## Pierce diode: tasks

- ① In Pierce\_diode, open file Pierce\_diode.py and execute: "python -i Pierce\_diode.py"
- ② Open cgm files and explore:
  - a) "gist Pierce\_diode.000.cgm &"
  - b) "gist current.cgm &"
- ③ Read input script and try to understand every command
- ④ Comment "w3d.solvegeom = w3d.rzgeom", uncomment "w3d.solvegeom = w3d.xyzgeom" and rerun; observe longer runtime but similar result
- ⑤ Reverse to RZ geometry
- ⑥ Set "steady\_state\_gun=True" and rerun. Simulation is now generating traces, converging to steady-state solutions faster than with time-dependent mode.
- ⑦ Set "w3d.l\_inj\_regular = True", "top.npinject = 15" and rerun with regularly spaced traces. This option can be used to enable faster simulations.
- ⑧ 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?



5

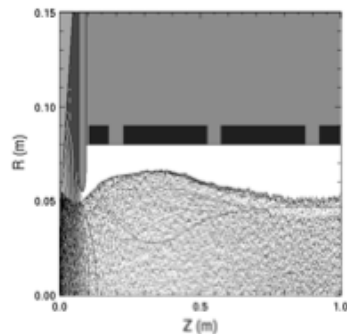
## Pierce diode: tasks

- ⑨ Go back to original settings
  - steady\_state\_gun=False
  - diode\_current = pi\*source\_radius\*\*2\*j
  - (optional) w3d.l\_inj\_regular = False and top.npinject = 150
 then change
  - top.inject=1 → top.inject=2 so that extracted current is automatically set at the Child-Langmuir limit, for a given voltage drop.
 Rerun. Open the latest cgm file, page through and observe how the head of the beam has a larger current and touches the extractor. Can you explain 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.



6

## Solenoid transport



File Solenoid\_transport.py:

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



7

## Solenoid transport: tasks

- ① In Solenoid\_transport, open Solenoid\_transport.py
- ② Execute file in interactive mode: "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(view=10)" to plot particle projections of azimuthal velocity versus z.
- ⑩ Notice the correlations between the extremas of the azimuthal velocity and the positions of the solenoids.
- ⑪ Here again, faster simulations can be performed by setting "w3d.l\_inj\_regular = True", "top.npinject = 15".



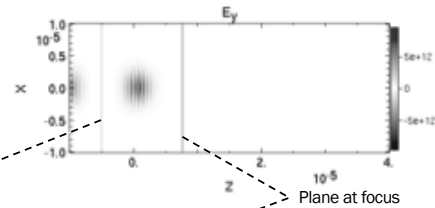
8

## Laser injection & propagation in vacuum/dielectric

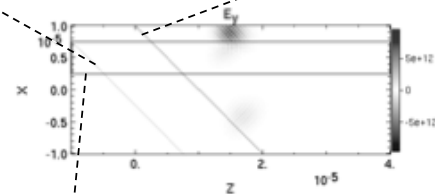
File laser\_injection.py:

- Laser injected with antenna
- "backward" laser exits
- Dielectric region can be set

Antenna emission plane



Plane at focus



Contours of dielectric box



9

## Laser injection: tasks

- ① In Laser\_injection, open Laser\_injection.py
- ② Execute file in interactive mode: "python -i Laser\_injection.py"
- ③ Read input script and try to understand every command
- ④ Check at beginning of scripts how to add optional arguments and their definitions
- ⑤ Rerun with a longer wavelength:
  - python -i laser\_injection.py -ll 2.e-6
- ⑥ Rerun with the laser impinging a dielectric at an angle of 45 degree:
  - python -i laser\_injection.py -lv '[1.,0.,1.]' -lp '[-5.e-6,0.,2.5e-6]' -er 1.5
- ⑦ Rerun with the laser born inside the dielectric:
  - python -i laser\_injection.py -lv '[1.,0.,1.]' -bp '[-5.e-6,5.e-6]' -er 1.5
- ⑧ Reducing the angle of incidence:
  - python -i laser\_injection.py -lv '[1.,0.,2.]' -bp '[-5.e-6,5.e-6]' -er 1.5
  - python -i laser\_injection.py -lv '[1.,0.,3.]' -bp '[-5.e-6,5.e-6]' -er 1.5
  - Note: you may propagate the laser further with 'step(200)'
  - Observe the total reflection with the latest run. What happens with -ll 2.e-6?
- ⑨ In the script, change laser\_source\_v from 0. to  $0.5 \cdot c$ light and run
  - python -i laser\_injection.py (observe the Doppler effect)



10