Modelling of ultra-intense laser propagation in plasmas and laser-plasma accelerators: fundamentals

Laserlab-Europe

TÉCNICO LISBOA

ISCTE IUL
Instituto Universitário de Lisboa

Jorge Vieira[1] , R. A. Fonseca[1,2],

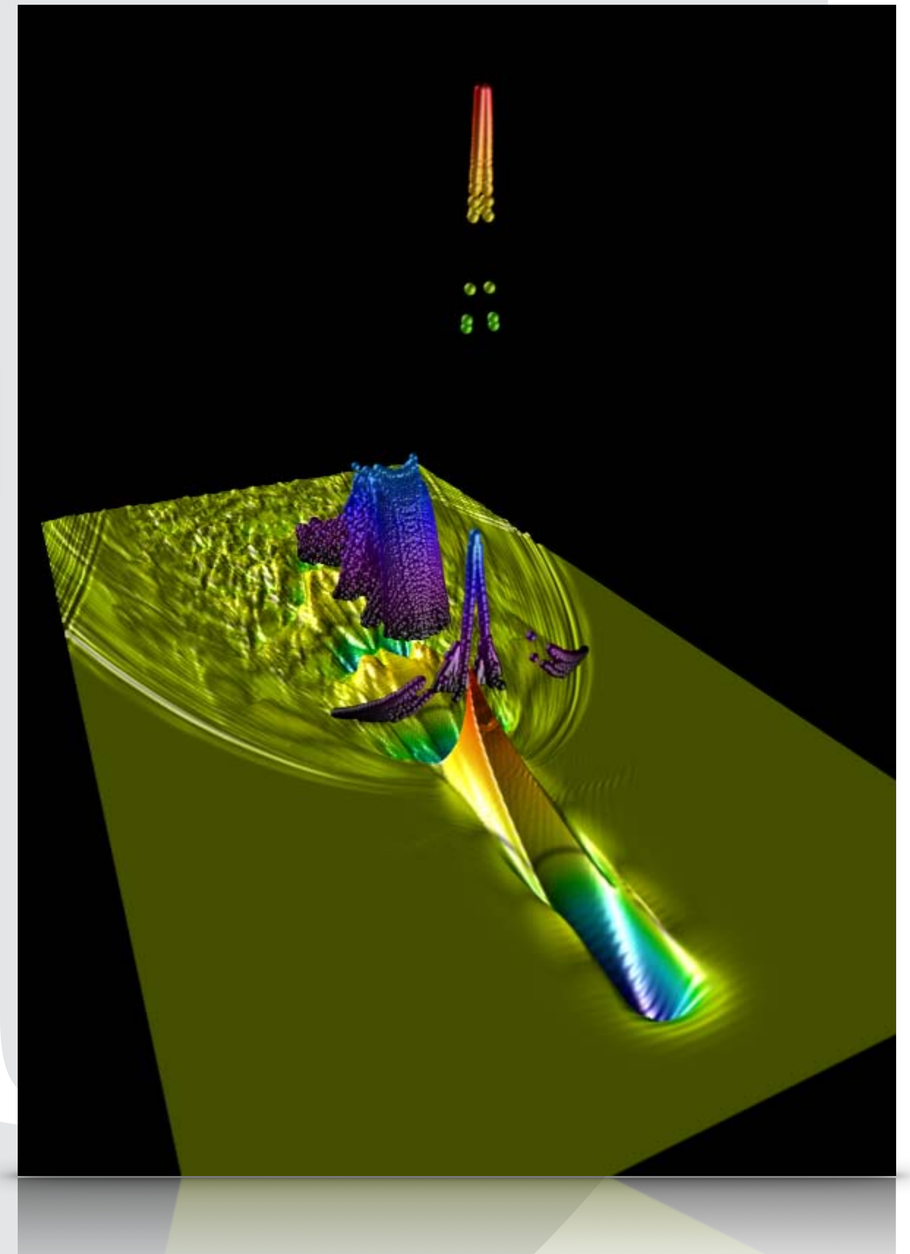
[1] GoLP/IPFN, Instituto Superior Técnico, Lisboa, Portugal

[2] DCTI, ISCTE-Instituto Universitário de Lisboa, Portugal

Outline

- **Advanced diagnostics**
  - Movies
  - Waterfall plots

- **Hands on examples**
  - *Propagation of e.m waves in a grid*
    - Do e.m. waves propagate as in vacuum?
  - *Beat wave accelerator*
    - Excite large amplitude plasma waves with low intensity lasers
    - Precursor of laser wakefield accelerator
  - *Self-modulated wakefield accelerator*
    - Create a beat-wave without an initial beat
    - Basis of AWAKE CERN experiment
  - *Controlled e- acceleration*
    - Plasma down ramp to induce electron trapping
  - *Avoiding laser diffraction*
    - Parabolic plasma channel



Advanced diagnostics

TÉCNICO LISBOA

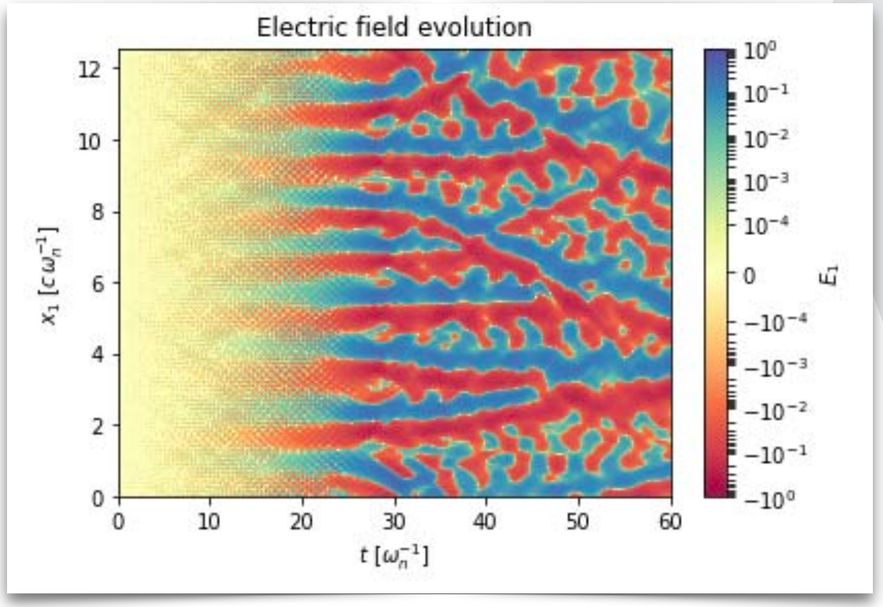
CERN Large Hadron Collider

Accelerator Tunnel

Waterfall plots

Two-stream instability example

- **Before running simulation**
  - Create 2D array to store simulation data (e.g. E_x)
- **Useful to (e.g.):**
  - Spatiotemporal evolution of instabilities
  - Determine temporal/spatial growth rates



Setup 2D array

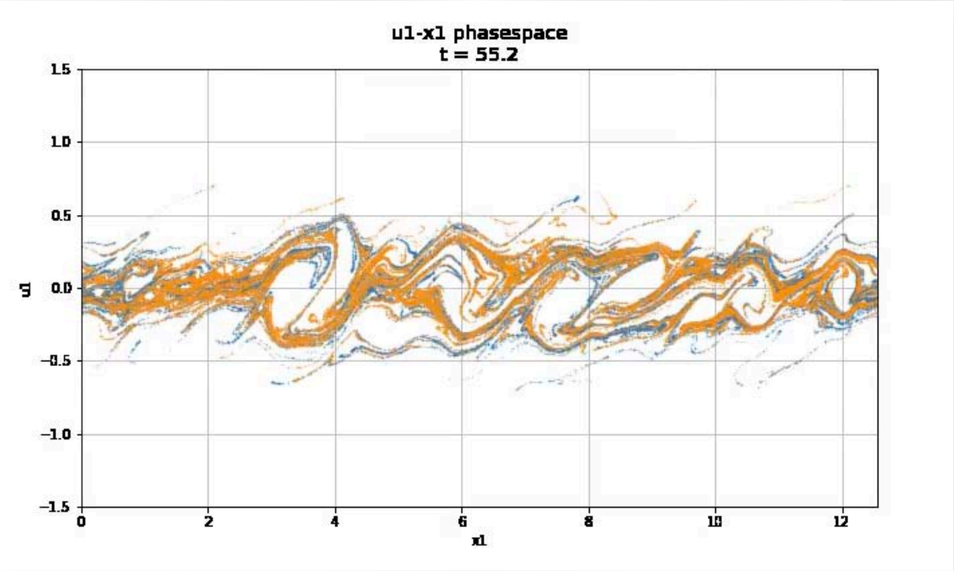
```
tmax = 60

niter = int(tmax / dt)

Ex_t = np.zeros((nx,niter))

print("\nRunning simulation up to t = {:g} ...".format(tmax))
while sim.t < tmax:
    print('n = {:d}, t = {:g}'.format(sim.n,sim.t), end = '\r')
    Ex_t[:,sim.n] = sim.emf.Ex
    sim.iter()

print("\nDone.")
```

Waterfall plot

```
import matplotlib.pyplot as plt
import matplotlib.colors as colors

range = [[0,tmax],[0,sim.box]]


plt.imshow( Ex_t, interpolation = 'bilinear', origin = 'lower',
          extent = ( range[0][0], range[0][1], range[1][0], range[1][1] ),
         norm=colors.SymLogNorm(base=10,linthresh = 0.0001, vmin=-1, vmax=1),
          aspect = 'auto', cmap = 'Spectral')

plt.colorbar().set_label('$E_1$')
plt.xlabel("$t$ [$\omega_n^{-1}$]")
plt.ylabel("$x_1$ [$c\,\omega_n^{-1}$]")
plt.title("Electric field evolution")

plt.show()
```

Movies

Animation.ipynb




```python
import matplotlib.pyplot as plt

from matplotlib import animation
from IPython.display import display
import ipywidgets

# Movie parameters
nframes = 200
tmax = 150.0
fps = 16

# Create progress bar
bar   = ipywidgets.FloatProgress( min = 0, max = 200 )
label = ipywidgets.HTML()
prog  = ipywidgets.VBox(children=[label, bar])
display(prog)

# Create plot
x = lambda s : (s.particles['ix'] + s.particles['x']) * s.dx

fig, ax = plt.subplots()

plt.rc('font', size=12)
fig.set_size_inches( (10.66,6.0) )

(p1,) = ax.plot([], [],  '.', ms=1,alpha=0.3, label = "Left")
(p2,) = ax.plot([], [], '.', ms=1,alpha=0.3, label = "Right")
ax.set_xlabel("x1")
ax.set_ylabel("u1")
ax.set_title("u1-x1 phasespace\nt = {:.1f}".format(sim.t))
ax.grid(True)

ax.set_xlim( (0,box ))
ax.set_ylim( (-1.5, 1.5) )

# Function to create each movie frame
skip  = np.int32(np.ceil((tmax / dt ) / (nframes-1) ))

def animate(i):
    label.value = "Generating frame {:d}/200 ...".format(i+1)
    bar.value = i

    if ( i > 0 ):
        for j in range(skip):
            sim.iter()

    p1.set_xdata(x(left))
    p1.set_ydata(left.particles['ux'])

    p2.set_xdata(x(right))
    p2.set_ydata(right.particles['ux'])

    ax.set_title("u1-x1 phasespace\nt = {:.1f}".format(sim.t))

    return (p1,p2)

# Create the movie
anim = animation.FuncAnimation( fig, animate, frames = nframes, repeat = False, blit = True, interval = 1000.0/fps )
movie = ipywidgets.HTML(anim.to_html5_video())

# Show the completed movie
label.value = "Done!"
bar.bar_style = "success"
display(movie)
```

Hands-on examples

CERN Large Hadron Collider
Accelerator Tunnel

Hands-on example templates

Short description

- **LWFA 1D -** Numerical simulations of a laser wakefield accelerator in 1D
- **LWFA 2D -** Numerical simulations of a laser wakefield accelerator in 2D
- **PWFA 1D -** Numerical simulations of a plasma wakefield accelerator in 1D
- **PBWA 1D -** Numerical simulations of a plasma beat wave accelerator in 1D
- **LWFA_1D-DownRamp –** Numerical simulations of down-ramp injection in a laser wakefield accelerator*
- **LWFA_1D-Self-Modulation –** Numerical simulations of the self-modulated laser wakefield accelerator*
- **Matching_parabolic –** Numerical simulations of laser propagation in a parabolic plasma channel*

***URLs**

https://raw.githubusercontent.com/ricardo-fonseca/zpic/master/doc/laserlab/LWFA_1D-DownRamp.ipynb
https://raw.githubusercontent.com/ricardo-fonseca/zpic/master/doc/laserlab/LWFA_1D-Self-Modulation.ipynb
https://raw.githubusercontent.com/ricardo-fonseca/zpic/master/doc/laserlab/Matching_parabolic.ipynb

Access hands-on example templates

1. Without Docker

Copy templates to computer and use

2. Docker with shared folder, i.e. launch Docker with:

```
> docker run -p 8888:8888 -t -v $PWD:/home/jovyan/work zamb/zpic
```

Copy templates to shared folder and use.

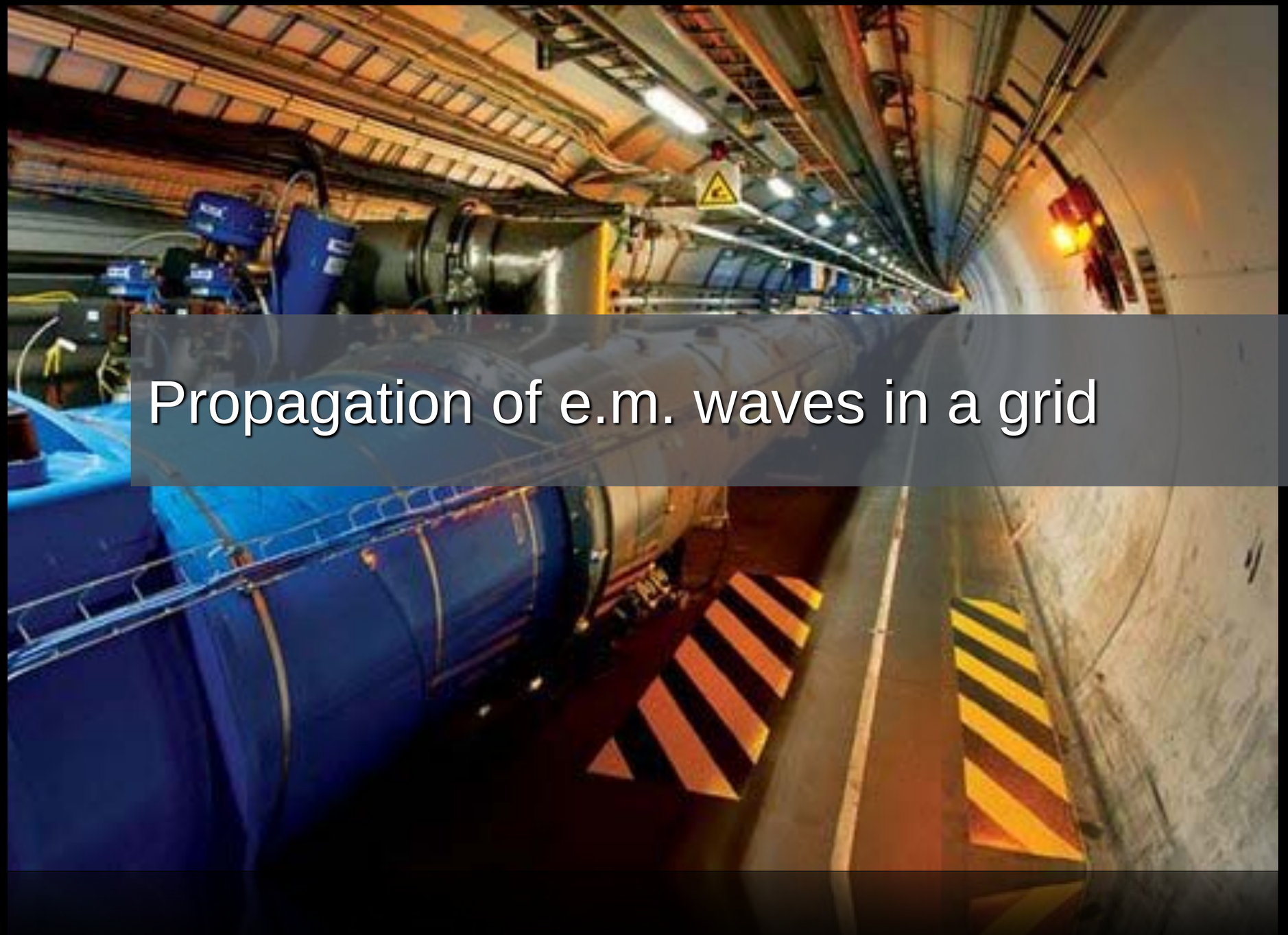
3. Docker without shared folder, i.e. launch Docker with:

```
> docker run -p 8888:8888 -t zamb/zpic
```

Use “File -> Open from URL"

Propagation of e.m. waves in a grid

TÉCNICO LISBOA



CERN Large Hadron Collider

Accelerator Tunnel

Modeling numerical dispersion relation

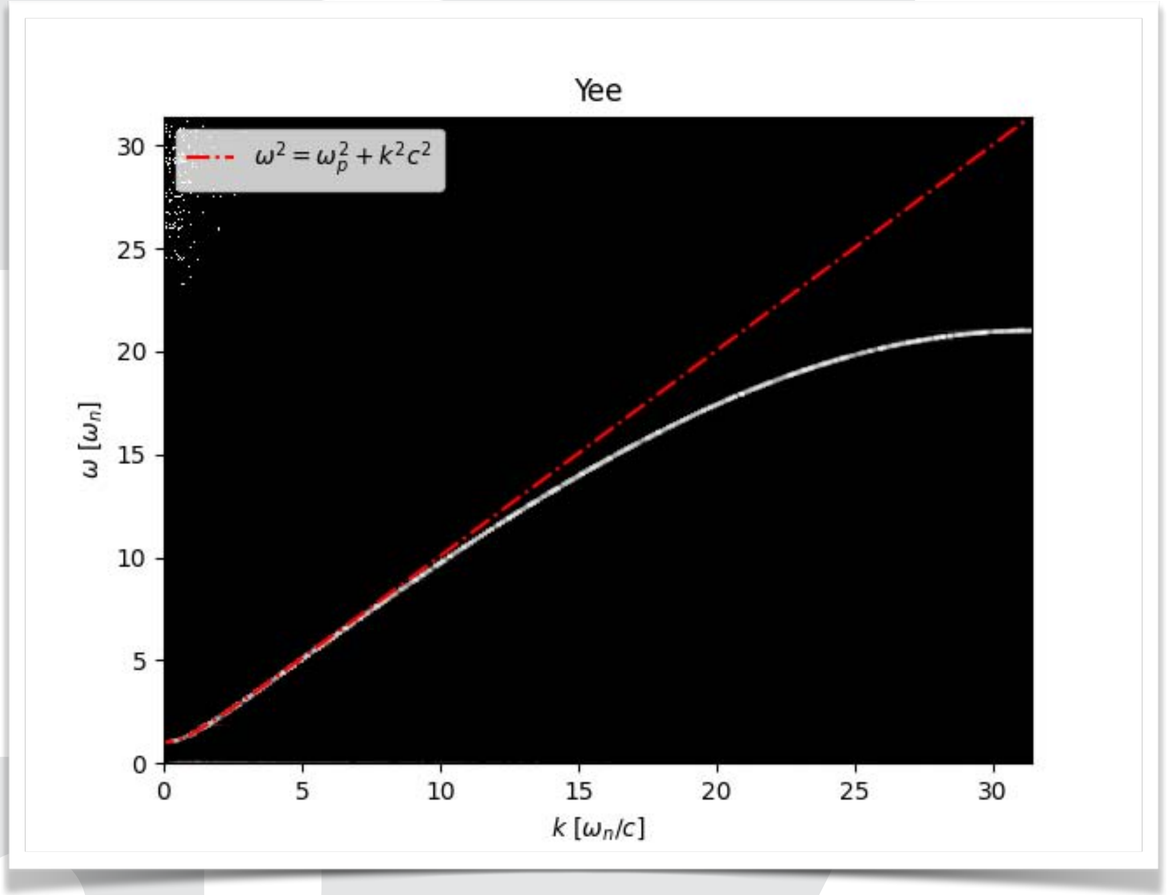
- **Inicialize warm plasma**
  - 1D simulation
  - Initial temperature is a seed for all possible modes in k space
  - Electrostatic waves appear in E_1
  - Electromagnetic waves appear in E_2 and E_3

```
# Choose between finite difference (em1d) and spectral (em1ds) codes
#import em1d as zpic
import em1ds as zpic

niter = 4000

electrons = zpic.Species( "electrons", -1.0, ppc = 64, uth=[0.001,0.001,0.001])
sim = zpic.Simulation( nx = 500, box = 50.0, dt = 0.5 * 0.1, species = electrons )
```

- **Modelling with ZPIC**
  - Run simulation over several plasma periods

```
# Run the simulation
import numpy as np
Ez_t = np.zeros((niter,sim.nx))

print("\nRunning simulation up to t = {:g} ...".format(niter * sim.dt))
while sim.n < niter:
    print('n = {:d}, t = {:g}'.format(sim.n,sim.t), end = '\r')
    Ez_t[sim.n,:] = sim.emf.Ez
    sim.iter()

print("\nDone.")
```

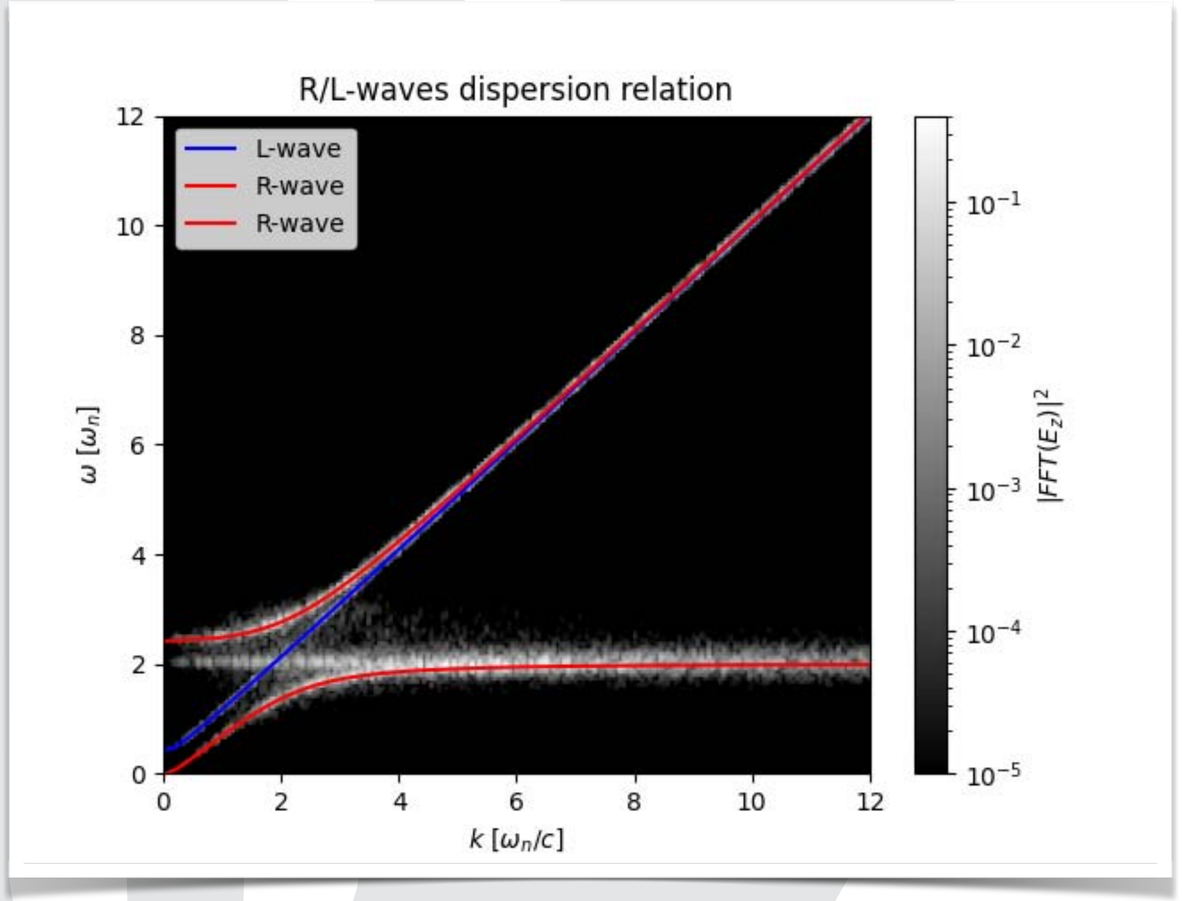
  - Produce a dispersion relation (ω vs. k) plot



Notebook: classroom/Field Solver Dispersion

Suggestions

- Change resolution, time step, temperature.
- Compare dispersion relation in E_2 and E_3.
- Plot dispersion relation for electrostatic modes instead.
- Include external magnetic field and interpret results.



Notebook: classroom/R-L Waves

Laser wakefield accelerator

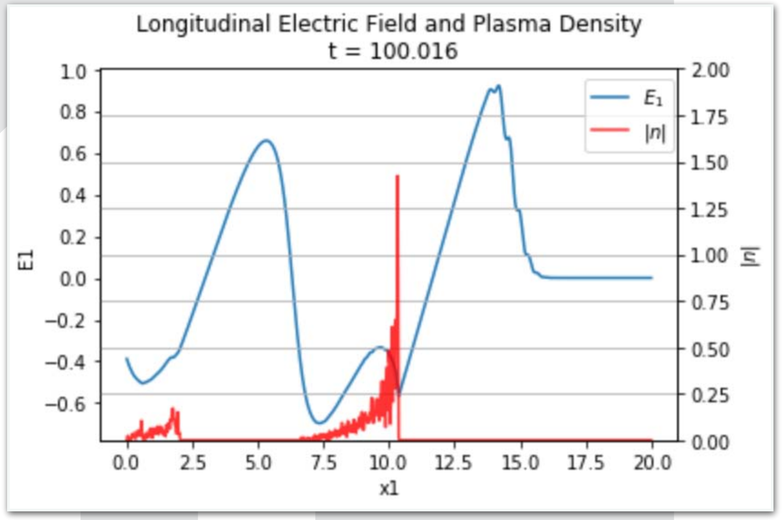
TÉCNICO LISBOA

CERN Large Hadron Collider
Accelerator Tunnel

Laser Wakefield Accelerator

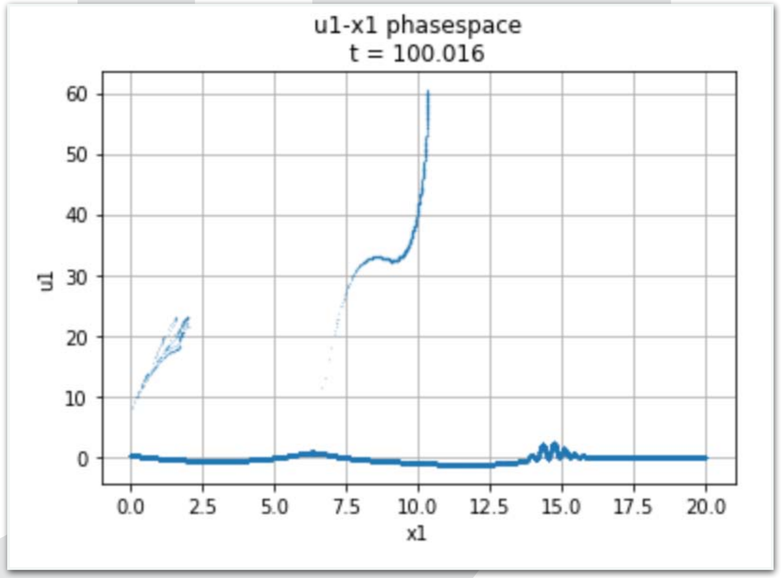
Simulate a laser wakefield accelerator:

- Add an ultra-intense laser beam as a driver (a_0~2)
- Choose laser length smaller than λ_p



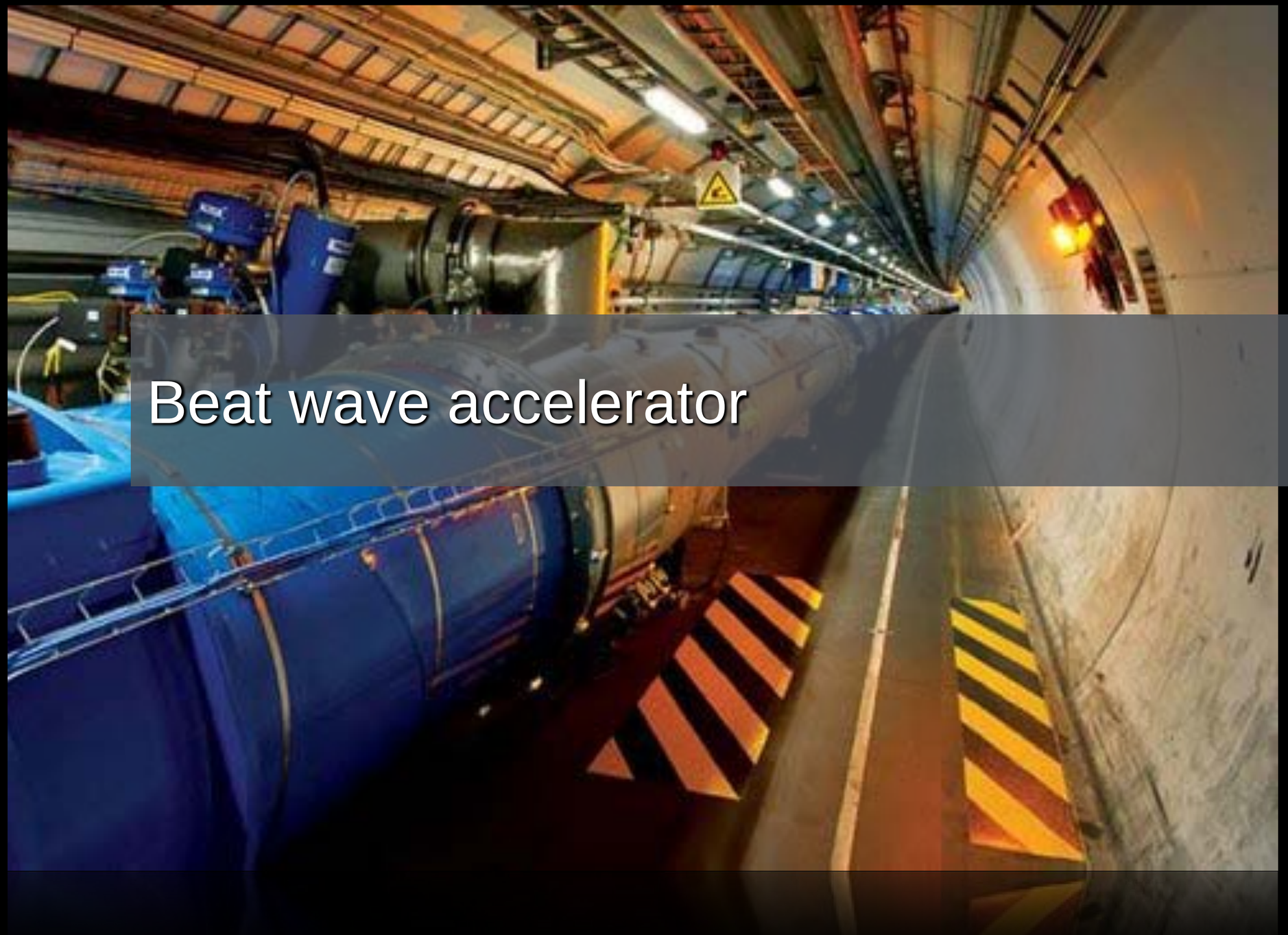
Questions:

1. can you observe particle injection and trapping?
2. is the energy gain consistent with the longitudinal electric field values?
3. describe and justify the shape for the plasma electric field in the region where particles accelerate
4. could you accelerate positrons in this plasma wave? where would you place them and with what initial velocity/energy? try to simulate!



Beat wave accelerator

TÉCNICO LISBOA



CERN Large Hadron Collider

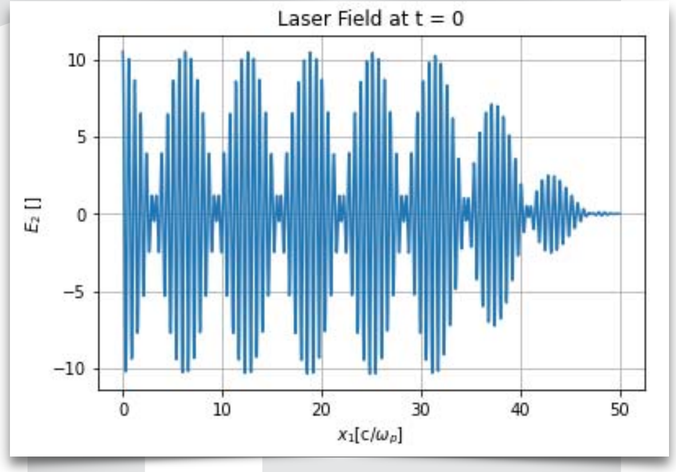
Accelerator Tunnel

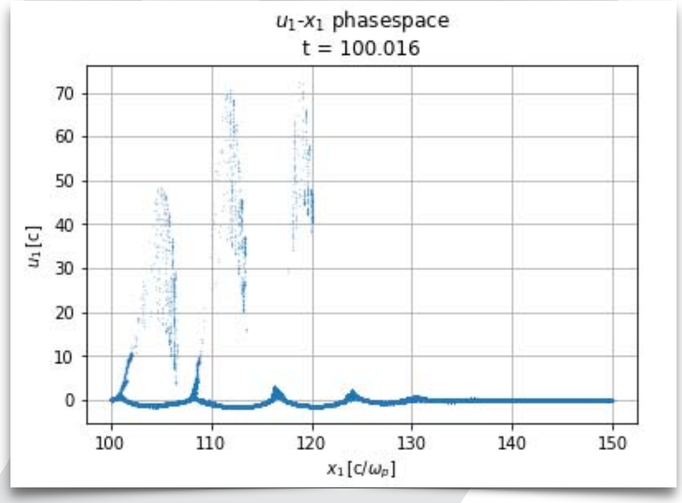
Plasma beat wave accelerator

Simulate a plasma beat-wave accelerator:

- Super-impose three laser modes with frequencies differing by ω_p (e.g. $\omega_0 = 10, 11\ \omega_p$)
- Choose Laser length $\gg \lambda_p$

```
# Add laser pulses
sim.add_laser( zpic.Laser( start = 50.0, rise = 20.0, flat = 40.0, fall = 1.0, a0 = 0.5, omega0 = 10.0 ))
sim.add_laser( zpic.Laser( start = 50.0, rise = 20.0, flat = 40.0, fall = 1.0, a0 = 0.5, omega0 = 11.0 ))
```

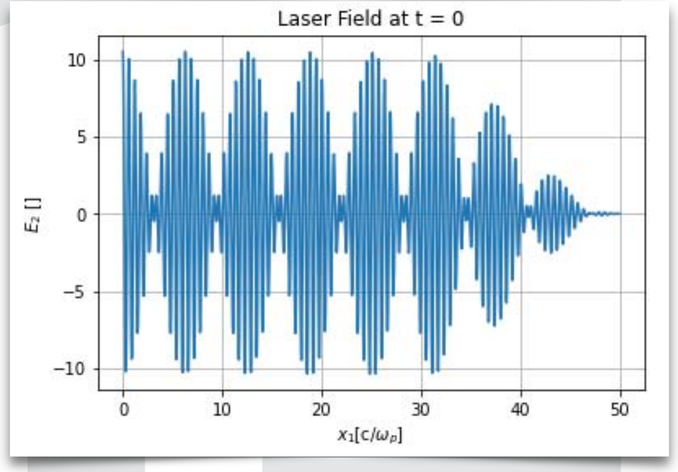


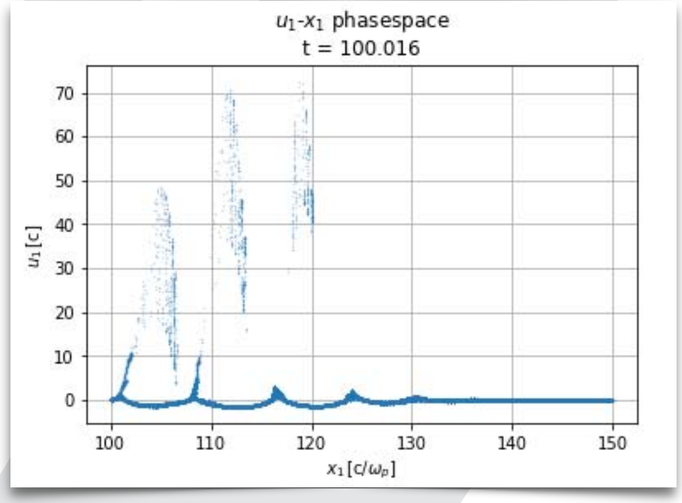


Suggestions

Questions:

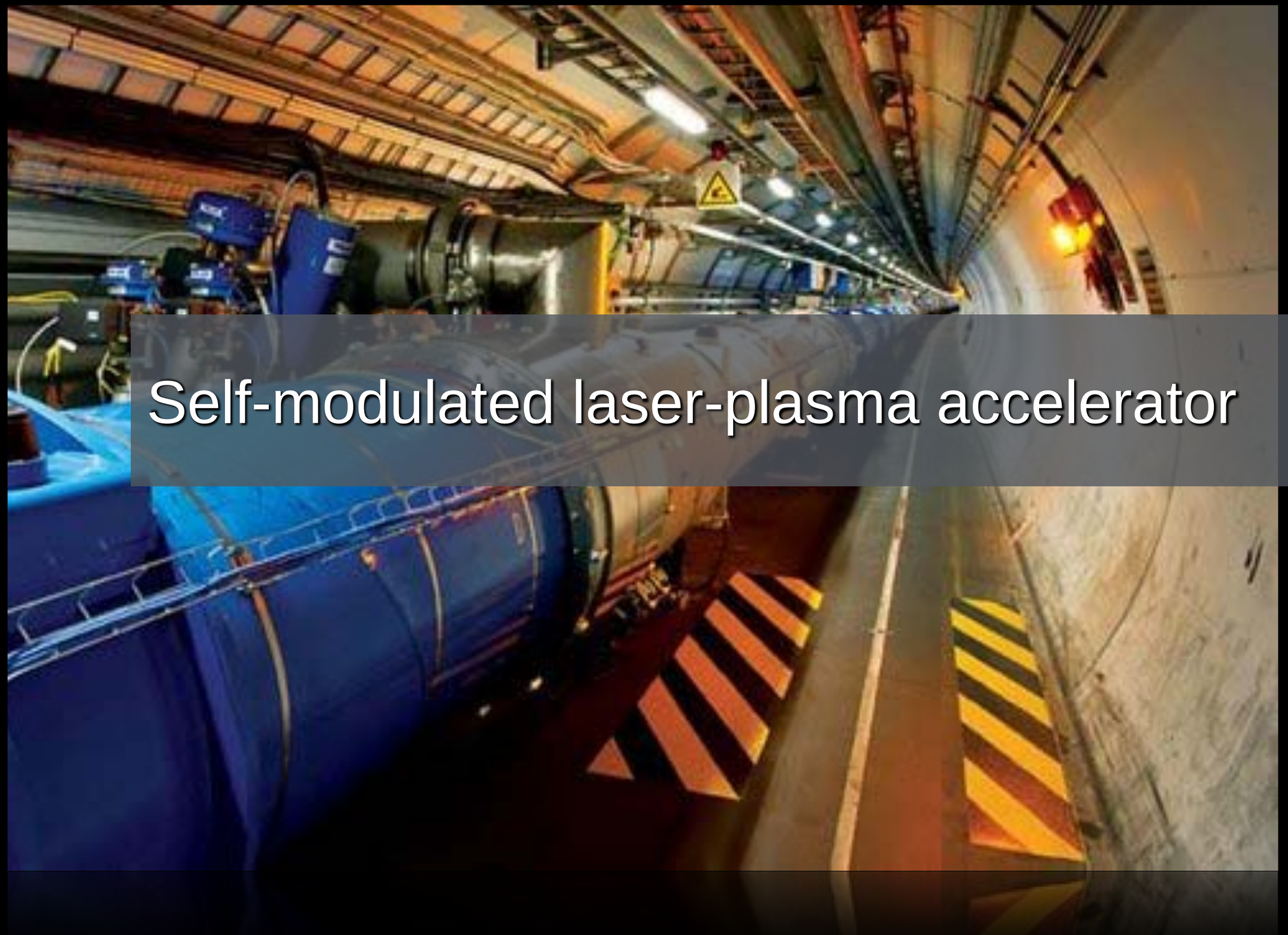
1. why does the plasma wave amplitude increase along the pulse?
2. what happens when the beat frequency is not ω_p? Why?
3. Try trapping electrons in the beatwave with a low intensity laser.
4. Decrease amplitude of laser frequency side bands. What happens to the laser?





Self-modulated laser-plasma accelerator

TÉCNICO LISBOA



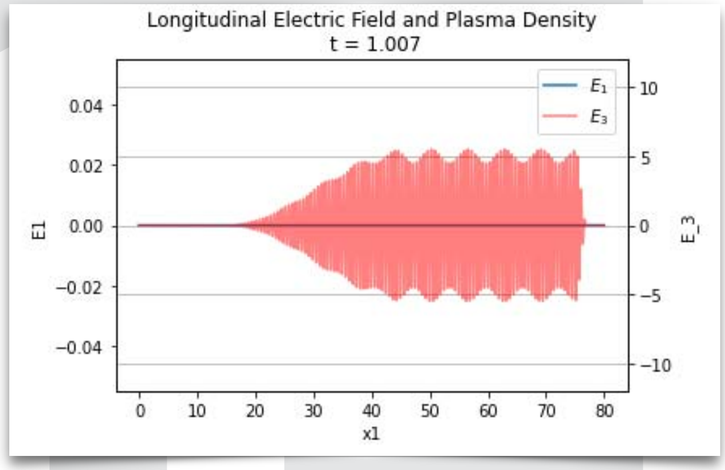
CERN Large Hadron Collider
Accelerator Tunnel

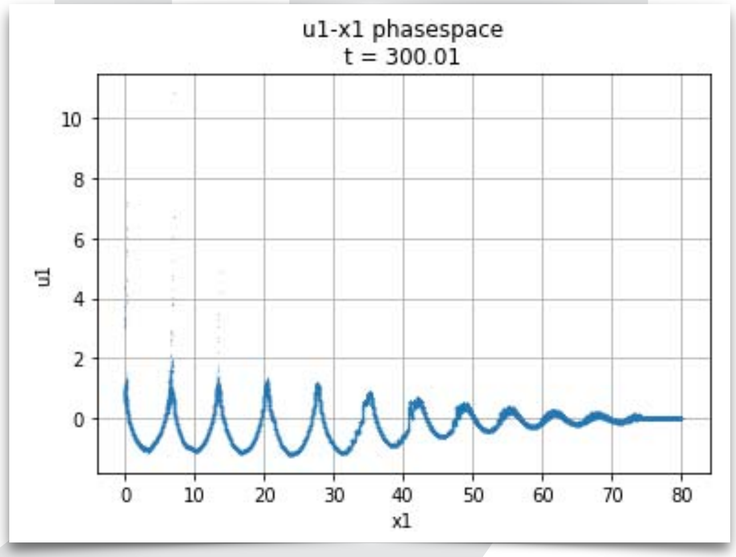
Self-modulated wakefield accelerator

Simulate a plasma self modulated wakefield accelerator:

- Choose Laser length $\gg\lambda_p$
- Super-impose three laser modes with frequencies differing by ω_p (e.g. ω_0 = 10, 11, 12 ω_p) with a small amplitude (seed)

```
# Add laser pulse 1
sim.add_laser( em1d.Laser( start = 77.0, rise = 2.0, flat = 30.0, fall = 30.0, a0 = 0.5, omega0 = 10.0,
                           polarization = numpy.pi/2 ))

# Add laser pulse 2
sim.add_laser( em1d.Laser( start = 77.0, rise = 2.0, flat = 30.0, fall = 30.0, a0 = 0.025, omega0 = 11.0,
                           polarization = numpy.pi/2 ))

# Add laser pulse 3
sim.add_laser( em1d.Laser( start = 77.0, rise = 2.0, flat = 30.0, fall = 3.0,a0 = 0.025, omega0 = 9.0,
                           polarization = numpy.pi/2 ))
```





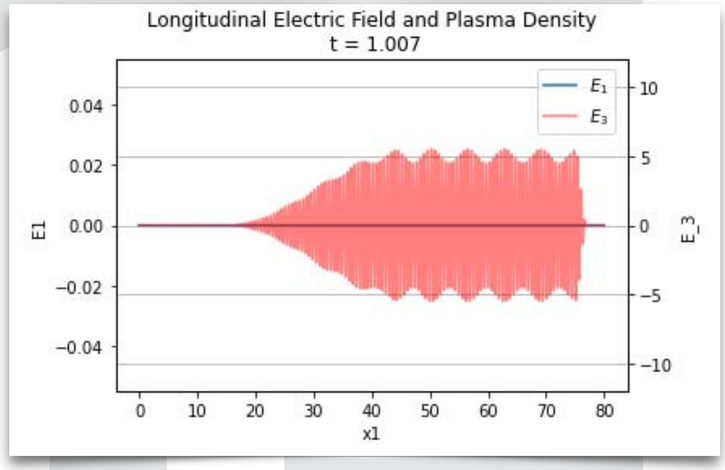
Self-modulated wakefield accelerator

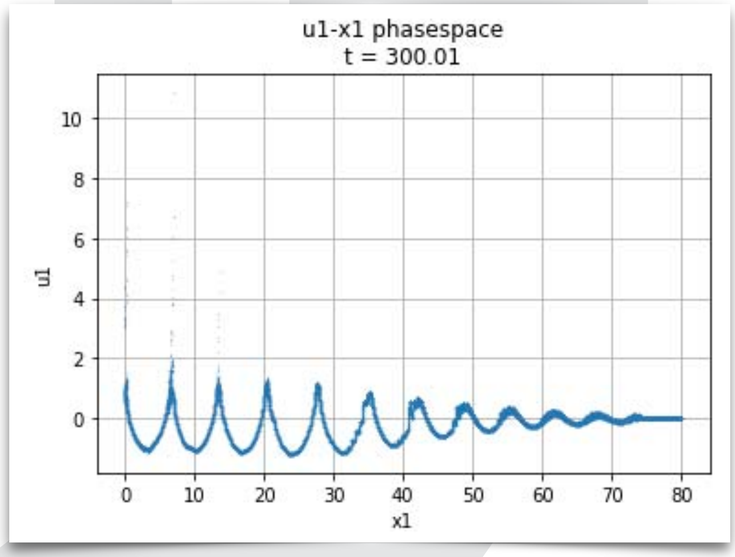
Questions:

1.why does the plasma wave amplitude increase with propagation distance?

2.what happens if the the initial seed frequencies of the lasers are not separated by the plasma frequency? Why?

3.find growth rate of self-modulation instability.





Controlled injection with plasma ramps

TÉCNICO LISBOA

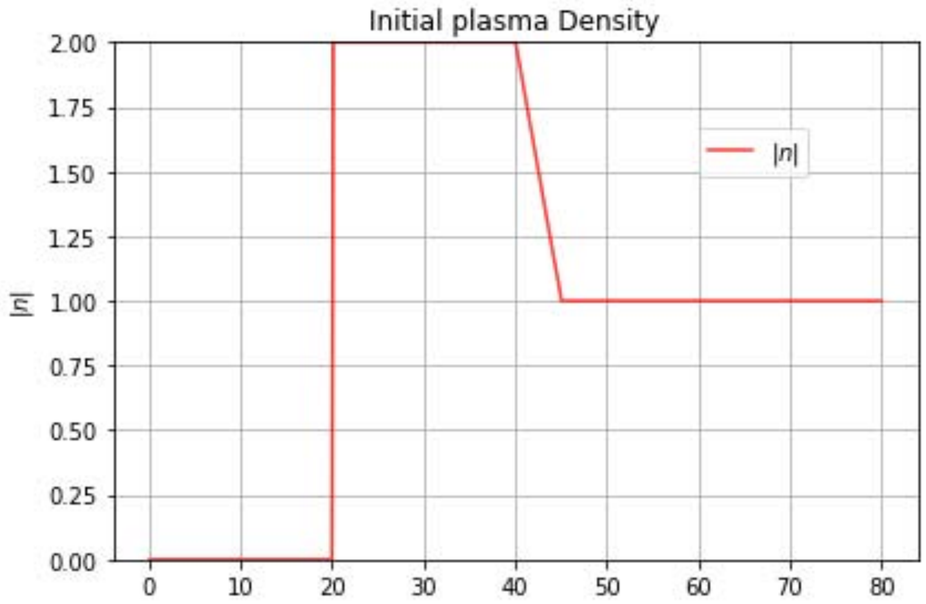
CERN Large Hadron Collider
Accelerator Tunnel

Down-ramp injection

Simulate down-ramp injection:

- Create a down-ramp
- e.g. super-impose three species (two slabs + ramp)

```
# Use a step density profile
electrons = em1d.Species( "electronsp", -1.0, ppc,
                          density = em1d.Density( type = "slab", start = 20, end = 400))

electronsp = em1d.Species( "electrons", -1.0, ppc,
                         density = em1d.Density( type = "slab", start = 20, end = 40))

electronsr = em1d.Species( "electronsr", -1.0, ppc,
                          density = em1d.Density( type = "ramp", start = 40, end = 45,ramp=[1.0,0.0]))
```



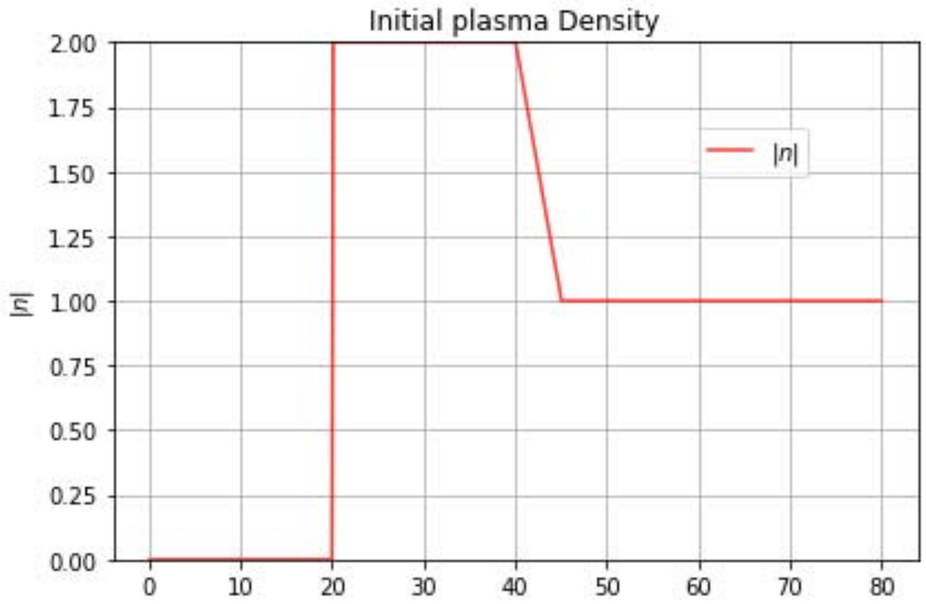
Down-ramp injection

Questions:

1.Compare the trapping thresholds with/without plasma ramp

2.What happens to the injected beams with multiple ramps?

3.What is the best ramp length for trapping?



Laser propagation in plasma channels

TÉCNICO LISBOA

CERN Large Hadron Collider
Accelerator Tunnel

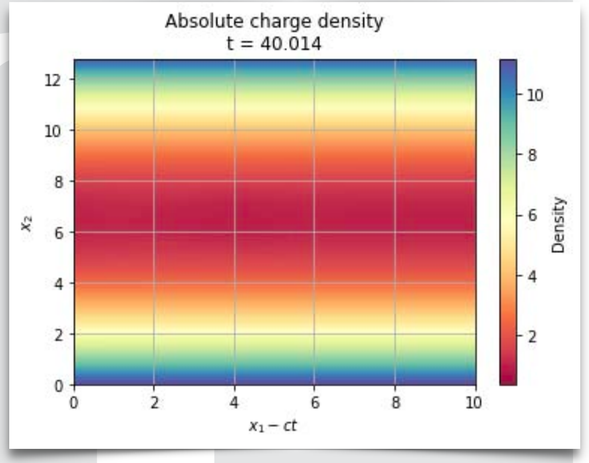
Propagation in parabolic plasma channel

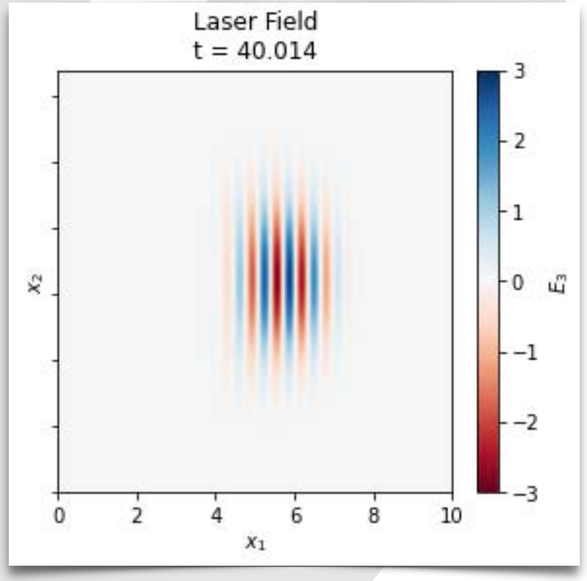
Avoid diffraction with a parabolic plasma channel

- plasma optical fiber $n(r) = n_0(1 + \Delta n/n_0 r^2)$
- Matched propagation condition (normalised) $\Delta n/n_0 = 4/w_0^4$

```
# Custom density profiles along x and y (density(x,y)=density_x(x)*density_y(y))
def density_x(x):
    if x>10:
        return 1.0
    return 0.0

def density_y(y):
    return 1.0+alpha*(y-axis_laser)*(y-axis_laser) #y^2
```

```
parabolic_density = zpic.Density( type = "custom", custom_x = density_x, custom_y = density_y )

electrons = zpic.Species( "electrons", -1.0, ppc,
                          density = parabolic_density)
```





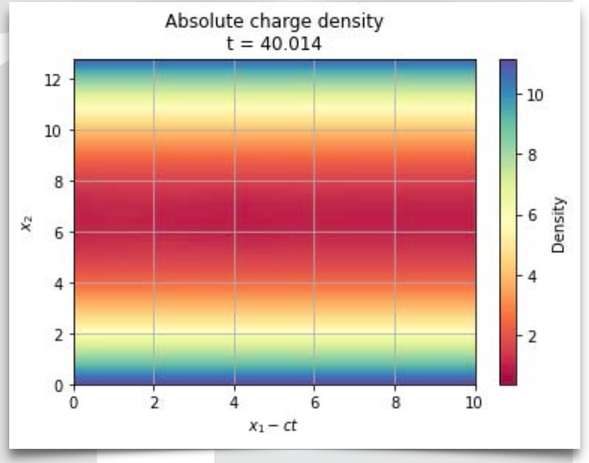
Propagation in parabolic plasma channel

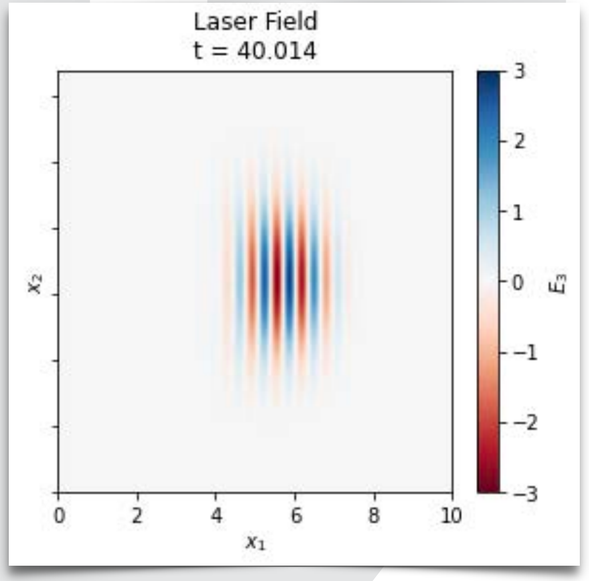
Questions:

1. compare propagation with/without channel

2.Vary laser focal plane. what happens? why?

3. Vary channel depth $\Delta n/n_0$





Plasma wakefield accelerator

TÉCNICO LISBOA

CERN Large Hadron Collider

Accelerator Tunnel

Plasma wakefield accelerator

Simulate a plasma wakefield accelerator:

- Use an ultra-relativistic particle beam as a driver
  - e.g. u_{fl} = 100, length ~ 10 c/ω_p, density ~ 0.3
- Choose plasma length $\gg\lambda_p$

```
# Use a step density profile
electrons = zpic.Species( "electrons", -1.0, ppc,
                          density = zpic.Density( type = "step", start = 100.0))

driver = zpic.Species( "driver", -1.0, ppc,
                          density = zpic.Density( n = 0.3 , type = "slab", start = 16 , end = 19 ),
                          ufl=[100.0,0.0,0.0])
```

VOLUME 54, NUMBER 7 PHYSICAL REVIEW LETTERS 18 FEBRUARY 1985

Acceleration of Electrons by the Interaction of a Bunched Electron Beam with a Plasma

Pisin Chen[(a)]
Stanford Linear Accelerator Center, Stanford University, Stanford, California 94305

and

J. M. Dawson, Robert W. Huff, and T. Katsouleas
Department of Physics, University of California, Los Angeles, California 90024
(Received 20 December 1984)

A new scheme for accelerating electrons, employing a bunched relativistic electron beam in a cold plasma, is analyzed. We show that energy gradients can exceed 1 GeV/m and that the driven electrons can be accelerated from $\gamma_0 mc^2$ to $3\gamma_0 mc^2$ before the driving beam slows down enough to degrade the plasma wave. If the driving electrons are removed before they cause the collapse of the plasma wave, energies up to $4\gamma_0^2 mc^2$ are possible. A noncollinear injection scheme is suggested in order that the driving electrons can be removed.

Plasma wakefield accelerator

Questions:

1. Why does the head of the driver loose energy?
2. What happens to the energy of the driver if it has a length comparable to λ_p?
3. What is the phase velocity of the plasma wave?
4. Can you observe plasma electron trapping and acceleration?