

# Excellence Center in Plasma Simulations and Theory

for Discovery Science and Disruptive Technology in  
Plasma Accelerators

**Jorge Vieira, Luís O. Silva**

GoLP / IPFN, Instituto Superior Técnico, Lisbon, Portugal



TÉCNICO  
LISBOA



ipfn  
INSTITUTO DE PLASMAS  
E FUSÃO NUCLEAR

Experimental discovery strongly assisted by numerical simulations

- Purely theoretical models are not available
- Experiments explore sub-set of all available parameters

Recent technological and computational advances make simulations virtual experiments

- Explore the physics self-consistently, no approximations
- Bridge between theory and experiments

# Excellent plasma simulation tools ready for the exa-scale

## Simulations in the exascale computing era

- Simulations are computationally intensive
- Large scale simulations require large supercomputers and advanced models
- Exa-scale computing in Europe through **EuroHPC**

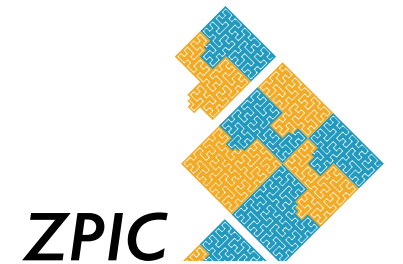


## *Collaboratorium*

- platform for **coordination, collaboration**, scientific and technical **exchange**
- **lead coordination efforts** for code development and integration
- **front end for the EU exascale** initiatives in plasma accelerators
- **hub for new and disruptive ideas**, and to explore the future directions for the facility

## The time is ripe

- Long tradition in plasma theory and simulation in Portugal
- Collaborations with the leading theory and experimental groups
- Code architects and co-developers of **Osiris**
- Developers of **ZPIC**



## IST nurtured plasma theory&simulation developments

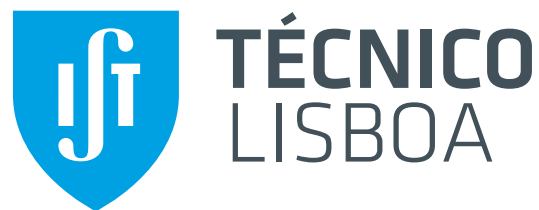
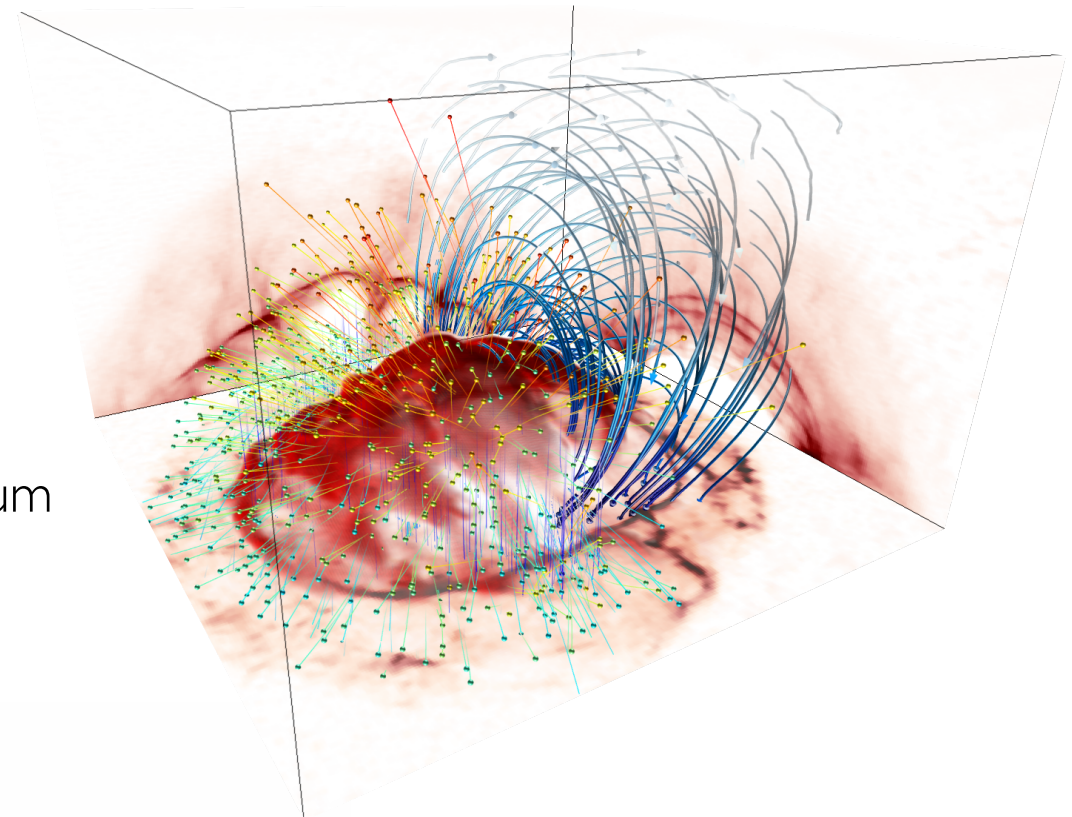
- Tools developed and used at **IST** and by many leading institutions in Europe, US and Asia
- Model for theory and simulation **collaboratorium**



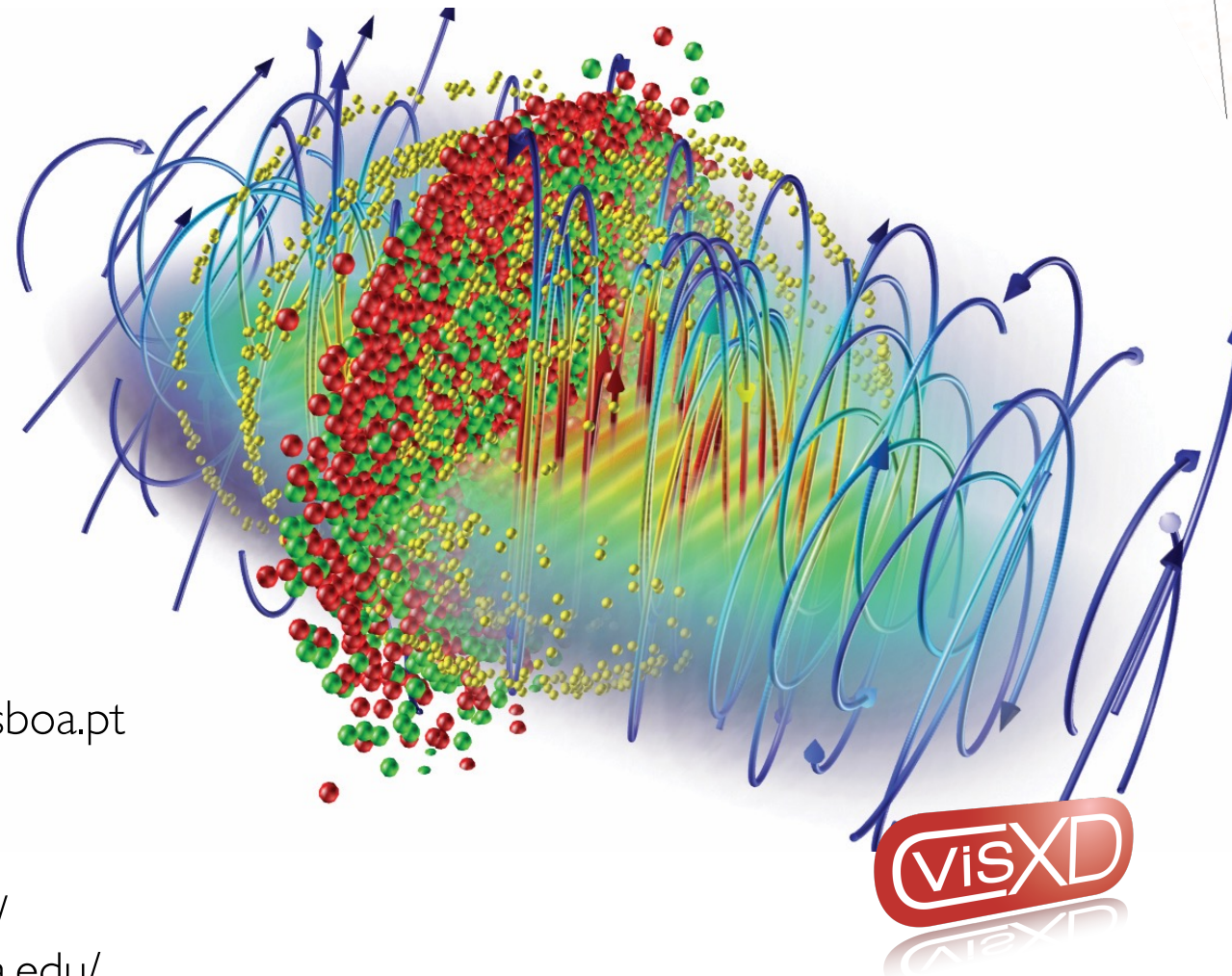


## osiris framework

- Massively Parallel, Fully Relativistic Particle-in-Cell (PIC) Code
- Visualization and Data Analysis Infrastructure
- Developed by the osiris.consortium  
⇒ UCLA + IST



**UCLA**



## code features

- **Scalability to ~1.6 M cores**
- SIMD hardware optimized
- Parallel I/O
- Dynamic Load Balancing
- **Ionization**
- **PGC support**
- Quasi-3d support
- QED module
- Particle merging
- Xeon Phi/GPGPU support

**Ricardo Fonseca**

ricardo.fonseca@tecnico.ulisboa.pt

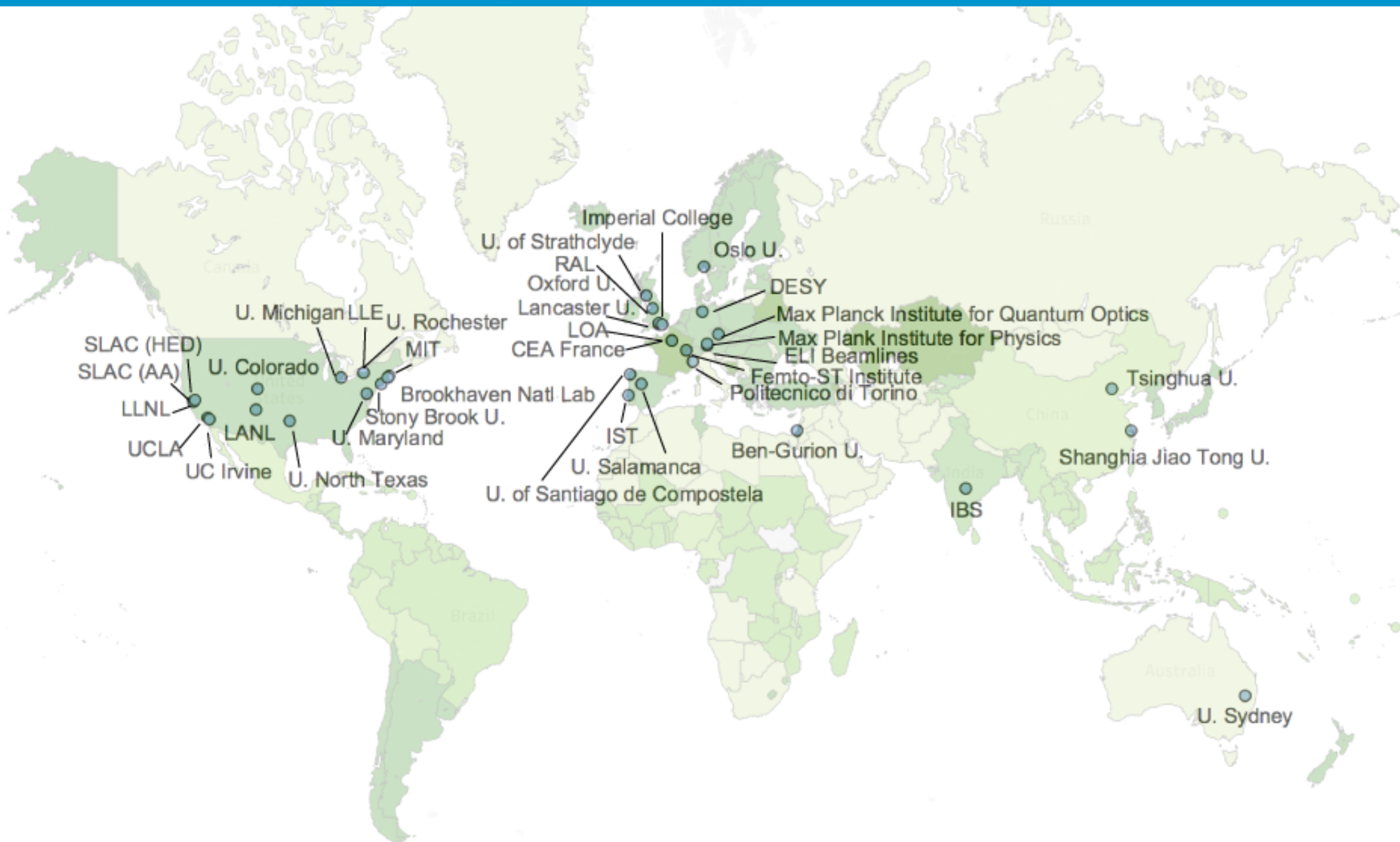
**Frank Tsung**

tsung@physics.ucla.edu

<http://epp.tecnico.ulisboa.pt/>

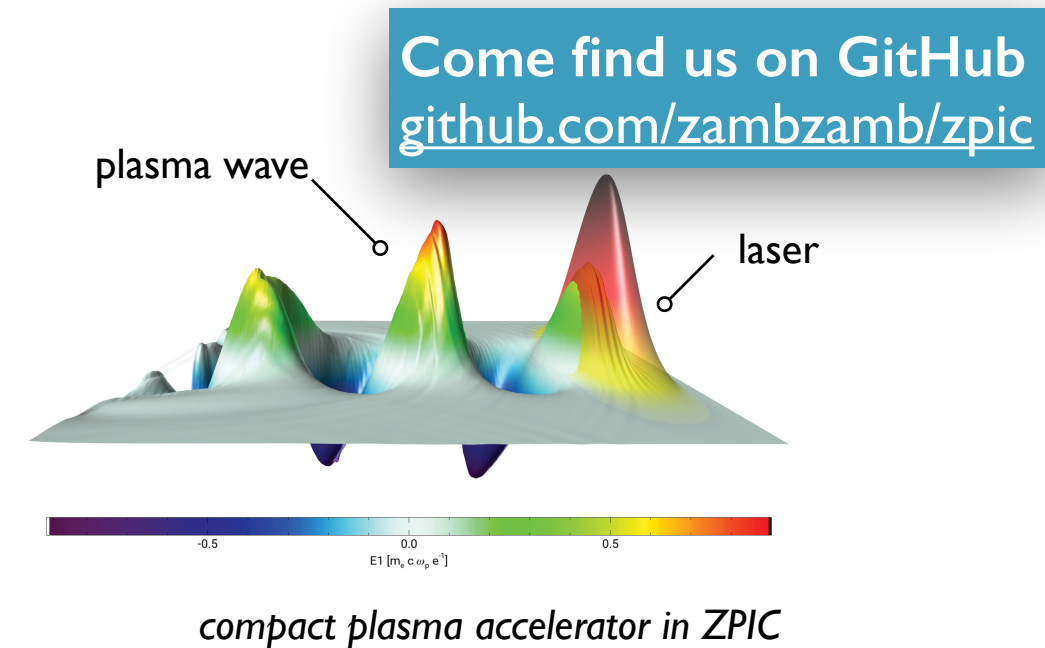
<http://plasmasim.physics.ucla.edu/>

# Osiris in the world






- **ZPIC** is a very simple PIC code (R. Fonseca et al) ideally suited for plasma physics and plasma acceleration education.
- **Employed** with success in the previous CERN accelerator school held in Sesimbra, Portugal (2019).
- **User friendly interface** written in c++ and controlled through Jupyter notebooks.



## Example

### Theoretical introduction



**Electron Plasma Waves**

Created by Rui Calado and Jorge Vieira, 2018

In this notebook, we are going to study the dispersion relation for electron plasma waves.

**Theory**

Electron plasma waves are longitudinal waves that may propagate in unmagnetized plasmas. To derive the dispersion relation for such waves let us start by considering the following setup:

- $\nabla \times \mathbf{E} = 0$  (Longitudinal waves)
- $T_i = 0$  (Cold plasma)
- $\mathbf{B} = 0$  (Unmagnetized)

We start by writing the continuity and momentum equations for the electron and ion species:

$$\begin{cases} \frac{\partial n_{e,i}}{\partial t} + \nabla \cdot (n_{e,i} \mathbf{v}_{e,i}) = 0 \\ \frac{\partial \mathbf{v}_{e,i}}{\partial t} = \mp \frac{e}{m_{e,i}} \mathbf{E} \end{cases}$$

Then we consider Poisson's equation:

$$\epsilon_0 \nabla \cdot \mathbf{E} = e(n_i - n_e)$$

Applying a time derivative twice,

$$\epsilon_0 \nabla \cdot \left( \frac{\partial^2 \mathbf{E}}{\partial t^2} \right) = e \left( \frac{\partial^2 n_i}{\partial t^2} - \frac{\partial^2 n_e}{\partial t^2} \right)$$

Using the continuity and momentum equations we get:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = \frac{e^2 n_0}{\epsilon_0} \left( \frac{1}{m_i} + \frac{1}{m_e} \right) \mathbf{E} = 0$$

This is the equation for a harmonic oscillator. Neglecting the  $1/m_i$  term since  $m_e \ll m_i$ , our oscillation frequency is the electron plasma frequency:

$$\omega = \frac{e^2 n_0}{\epsilon_0 m_e} \equiv \omega_p$$

**Warm plasma**

Now, what happens if we consider a warm plasma instead? Neglecting ion motion, the first step is to add a pressure term to the electron momentum equation:

$$\frac{\partial \mathbf{v}_e}{\partial t} = -\frac{e}{m_e} \mathbf{E} - \frac{\gamma k_B T_e}{m_e n_0} \nabla n_e$$

We also note that Poisson's equation now takes the form:

$$\nabla \cdot \mathbf{E} = -\frac{e}{\epsilon_0} n_i$$

### simulation initialisation

We also note that Poisson's equation now takes the form:

$$\nabla \cdot \mathbf{E} = -\frac{e}{\epsilon_0} n_i$$

Taking the divergence of the momentum equation, we get:

$$\nabla \cdot \left( \frac{\partial \mathbf{v}}{\partial t} \right) = -\frac{e}{m_e} \nabla \cdot \mathbf{E} - \frac{\gamma k_B T_e}{m_e n_0} \nabla \cdot (\nabla n_e)$$

Using the unchanged continuity equation and Poisson's equation:

$$\frac{\partial^2 n_i}{\partial t^2} + \omega_p^2 n_i - \frac{\gamma k_B T_e}{m_e} \nabla \cdot (\nabla n_i) = 0$$

Considering the high frequency regime, there will be no heat losses in our time scale, and so we will take the adiabatic coefficient  $\gamma = 3$  for 1D longitudinal oscillations. Additionally, we use the definition  $v_{th}^2 = k_B T_e / m_e$  to write:

$$\frac{\partial^2 n_i}{\partial t^2} + \omega_p^2 n_i - 3v_{th}^2 \nabla \cdot (\nabla n_i) = 0$$

The final step consists of considering sinusoidal waves such that  $n_i = n_1 \exp(i\mathbf{k} \cdot \mathbf{r} - i\omega t)$  and then Fourier analyzing the equation ( $\nabla = i\mathbf{k}$ ,  $\frac{\partial}{\partial t} = -i\omega$ ), which results in the dispersion relation:

$$\omega^2 = \omega_p^2 + 3v_{th}^2 k^2$$

**Simulations with ZPIC**

```
In [2]: import ems as zp
#v_th = 0.001
#v_th = 0.02
#v_th = 0.20

electrons = zp.Species("electrons", -1.0, ppc = 64, u0=[v_th, v_th, v_th])
sim = zp.simulation( nx = 500, box = 50.0, dt = 0.0999/2, species = electrons )
sim.emf.solver_type = "PSATD"

sim.emf.set_filter("none")
#sim.emf.set_filter("sharp", a = 0.99)
#sim.emf.set_filter("gaussian", a = 0.001)
```

We run the simulation up to a fixed number of iterations, controlled by the variable `niter`, storing the value of the EM field  $\mathbf{E}_i$  at every timestep so we can analyze them later.

```
In [3]: import numpy as np
niter = 4000
Ex_t = np.zeros((niter, sim.nx))
Ez_t = np.zeros((niter, sim.nx))
tmax = niter * sim.dt

print("\nRunning simulation up to t = {}s".format(tmax))
while sim.t <= tmax:
    print('t = {}s, t = {}s'.format(sim.t, tmax))
    Ex_t[sim.n, :] = sim.emf.Ex
    Ez_t[sim.n, :] = sim.emf.Ez
```

### analysis and questions for discussion

**Electromagnetic Plasma Waves**

To analyze the dispersion relation of the electrostatic plasma waves we use a 2D (Fast) Fourier transform of  $E_i(x, t)$  field values that we stored during the simulation. The plot below shows the obtained power spectrum alongside the theoretical prediction. Since the dataset is not periodic along  $t$  we apply a windowing technique (Hanning) to the dataset to lower the background spectrum, and make the dispersion relation more visible.

```
In [4]: import matplotlib.pyplot as plt
import matplotlib.colors as colors
# (omega, k) power spectrum
win = np.hanning(niter)
for i in range(sim.nx):
    Ez_t[:, i] *= win
sp = np.abs(np.fft.fft2(Ez_t))**2
sp = np.fft.fftshift(sp)
k_max = np.pi / sim.dx
omega_max = np.pi / sim.dt

plt.imshow(sp, origin = 'lower', norm=colors.LogNorm(vmin = 1e-7, vmax = 0.01),
           extent = (-k_max, k_max, -omega_max, omega_max),
           aspect = 'auto', cmap = 'gray')

k = np.linspace(-k_max, k_max, num = 512)
w = np.sqrt(1 + k**2)
plt.plot(k, w, label = '$\omega = \sqrt{\omega_p^2 + k^2 c^2}$', color = 'r', ls = '-')

plt.ylim(0, k_max)
plt.xlim(0, k_max)
plt.xlabel('$k$ [$\omega_p/c$]')
plt.ylabel('$\omega$ [$\omega_p$]')
plt.title('EM-wave dispersion relation')
plt.legend()
plt.show()
```

```
In [ ]:
```



## About IST

- **Largest** science&engineering school in Portugal
  - 11500 students
  - 870 research and teaching staff
  - 3 campi (alameda, taguspark, tecnológico e nuclear)
- **Attracts top** students in science&engineering

***Vibrant academic and scientific ecosystem on computational sciences and engineering***



## Light structure

- scientific **director**, 1-2 **admin**, 3-5 **computational scientists**, including **visualisation experts**, 1 **outreach/communication officer**
- **strong** connection with national and Iberian supercomputing infrastructures

## Main goals

- simulation and theoretical **support for EuPRAXIA** teams involved in computing
- **coordinate** virtual interactions in distributed Eupraxia configuration involved in theory and simulations
- **visiting/workshop program** on plasma accelerators (advanced training, convene PhD students, post docs and senior researchers)

- Take the form of a **Collaboratorium**
- **Unique location** of Lisbon (easy access from anywhere in Europe)
- **Institutional support** from the University of Lisbon
- Strongly **benefiting** from
  - integration of Portugal in the **EuroHPC** initiative,
  - **leading role** of plasma acceleration scientists in the Portuguese HPC community

