# LONGITUDINAL TRACKING HANDS-ON

## ADVANCED TRACKING

### RF CAS 2023

S. Albright, A. Lasheen,
F. Batsch, K. Iliakis, G. Papotti, D. Quartullo, H. Timko,
L. Intelisano, B. Karlsen-Baeck, I. Karpov, O. Naumenko, A. Vanel, M. Zampetakis

# COPYRIGHT

The CERN Accelerator School

**Copyright statement and speaker's release for video publishing**

# OUTLINE OF THE HANDS-ON

## FOR EACH HANDS-ON BLOCK

## FIRST AFTERNOON: INTRODUCTION TO TRACKING

- Develop a multiparticle tracking code, without intensity effects
- Two options are offered:
    - Make your own tracker!
    - Use the BLonD simulation code

## SECOND AFTERNOON: ADVANCED TRACKING

- Learn how to include intensity effects (wakefields) using BLonD
- Check if instabilities are a limitation for scSPS
- Get insights on advanced topics like potential well distortion, as you would measure it in a real accelerator

# OUTLINE OF THE HANDS-ON

## FOR EACH HANDS-ON BLOCK

## TODAY!

## SECOND AFTERNOON: ADVANCED TRACKING

- Learn how to include intensity effects (wakefields) using BLonD
- Check if instabilities are a limitation for scSPS
- Get insights on advanced topics like potential well distortion, as you would measure it in a real accelerator

# ADVANCED TRACKING

## ALL YOU NEED TO GET STARTED

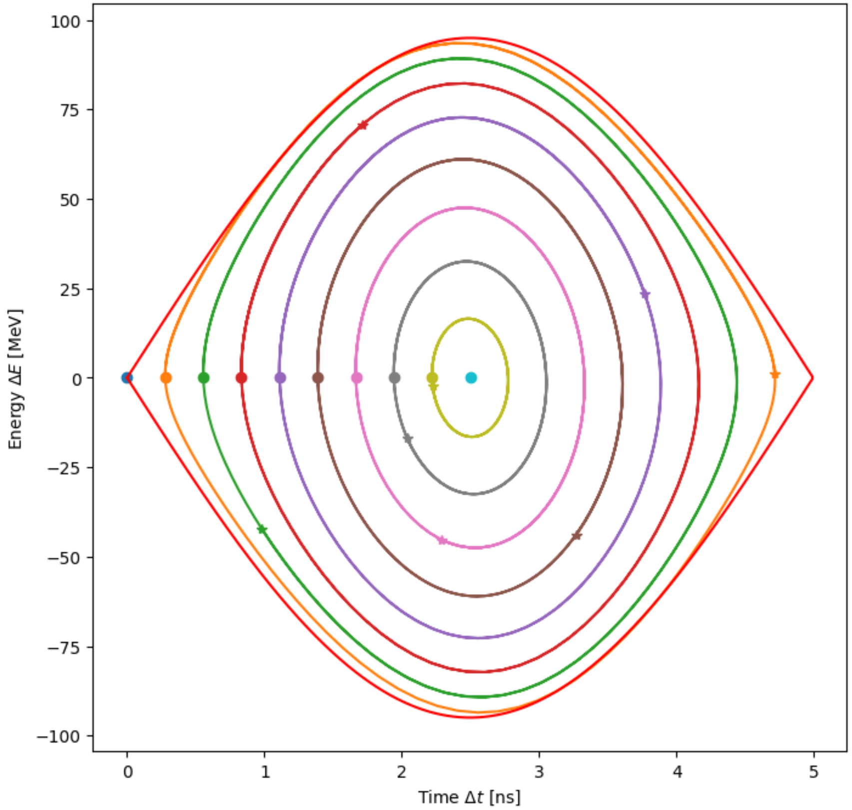
→ **Resuming from the introduction session**

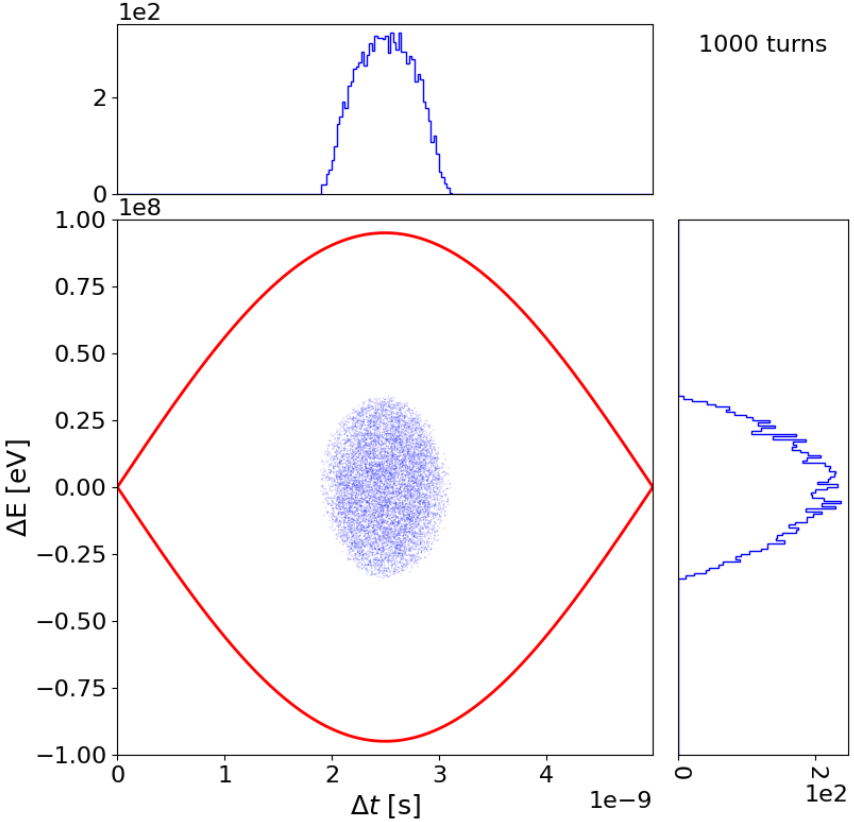
→ **Wakefield, impedance, induced voltage**

→ **Intensity effects using BLonD**

# RESUMING FROM INTRODUCTION

## PREVIOUSLY, AT THE RF CAS...




- Yesterday, we have experimented with longitudinal tracking by applying the two discretized equations of motion (energy gain and relative drift).

# TRACKING USING BLOND

## SOLUTION TO INTRODUCTION EXERCISE (1/3)

- We first establish the layout of the accelerator and base parameters

```
ring = Ring(circumference, alpha_0, momentum, particle,
            n_turns=n_turns)
rf_station = RFStation(ring, harmonic, voltage, phase)
beam = Beam(ring, n_macroparticles, intensity)
```

- An initial particle distribution is needed (defined from pre-injectors; starting from matched conditions...)

```
parabolic(ring, rf_station, beam, bunch_length)
```

- The longitudinal profile is generated to monitor bunch parameters, which can be compared with measurements

```
cut_opt = CutOptions(
    cut_left=0,
    cut_right=rf_station.t_rf[0, 0],
    n_slices=60)
fit_opt = FitOptions(fit_option='gaussian')
profile = Profile(beam, CutOptions=cut_opt, FitOptions=fit_opt)
```

# TRACKING USING BLOND

## SOLUTION TO INTRODUCTION EXERCISE (2/3)

- A monitoring object is made to record the simulated bunch parameters on disk

```
h5file = './data/saved_data'
bunchmonitor = BunchMonitor(
    ring, rf_station, beam,
    h5file,
    Profile=profile)
bunchmonitor.track()
```

- And some plotting parameters specified for visualization

```
dt_plt = 200
plots = Plot(ring, rf_station, beam,
             dt_plt, ring.n_turns,
             0, rf_station.t_rf[0, 0],
             -100e6, 100e6,
             Profile=profile,
             h5file=h5file,
             separatrix_plot=True)
```

# TRACKING USING BLOND

## SOLUTION TO INTRODUCTION EXERCISE (3/3)

- The effective tracking objects are generated

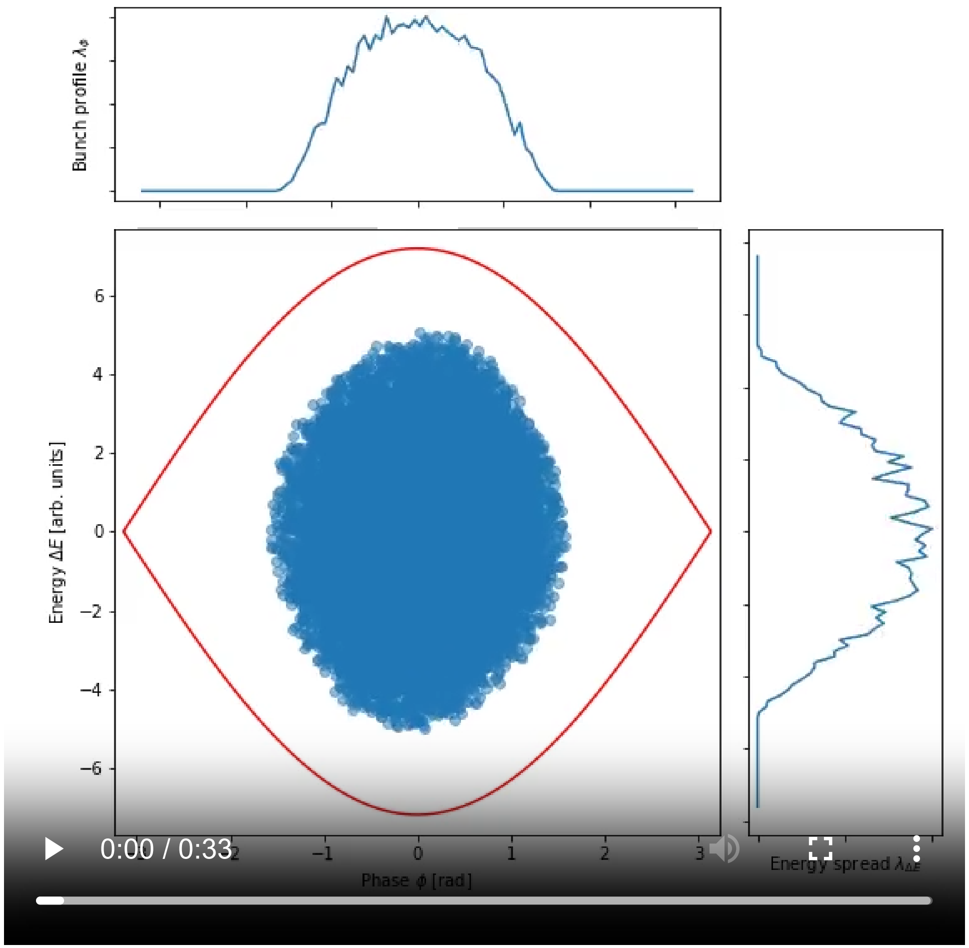
```
rf_tracker = RingAndRFTracker(rf_station, beam)
full_tracker = FullRingAndRF([rf_tracker])
```

- **Now is time to track!!**

```
for turn in range(ring.n_turns):
    full_tracker.track()
    profile.track()
    plots.track()
    bunchmonitor.track()
```

- The *track* routines in BLonD represents the main action of objects but do not always actuate on the particles (e.g. plots and bunch monitoring)

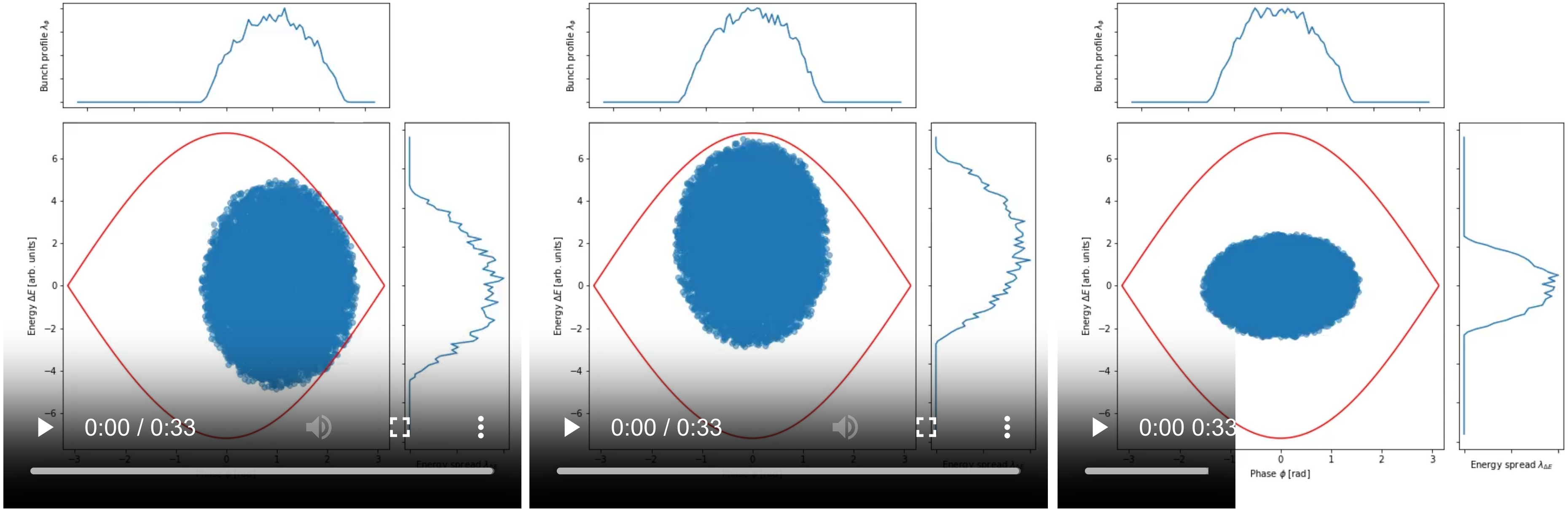
# EXAMPLE OF APPLICATIONS




- The effective physics in the tracking code relies on two simple functions (kick and drift).
- This is nonetheless sufficient to do quite extensive studies, for the design of future accelerators and operational optimization of the existing ones.
- **Let's check injection oscillations!**

# EXAMPLE OF APPLICATIONS

## INJECTION




→ Example of error in injection phase (left), energy (middle), voltage (right).

# ADDING COMPLEXITY TO THE TRACKING

- We previoulsy considered a simple ring with only a single RF system and constant parameters, giving the energy change

$$\Delta E^{n+1} = \Delta E^{n} + qV_{\mathrm{rf}} \sin\left(\omega_{\mathrm{rf}} \Delta t + \phi\right) - \delta E_{d}^{n \rightarrow n+1}$$

# ADDING COMPLEXITY TO THE TRACKING

- We previoulsy considered a simple ring with only a single RF system and constant parameters, giving the energy change

$$\Delta E^{n+1} = \Delta E^{n} + qV_{\mathrm{rf}} \sin\left(\omega_{\mathrm{rf}} \Delta t + \phi\right) - \delta E_{d}^{n \rightarrow n+1}$$

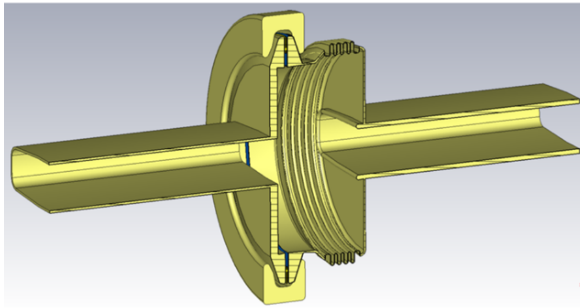
- We can extend our model by summing other contributions

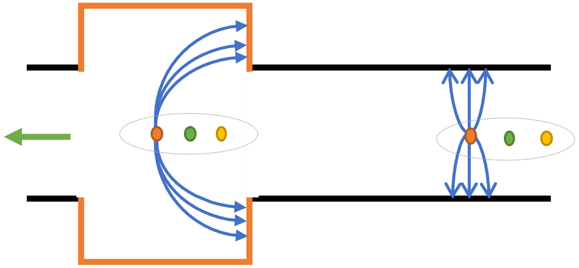
$$\Delta E^{n+1} = \Delta E^{n} + q \sum_{i} V_{\mathrm{rf,i}}\left(t\right) g\left[\omega_{\mathrm{rf,i}}\left(t\right) \Delta t + \phi_{i}\left(t\right)\right]$$
$$-\delta E_{d}^{n \rightarrow n+1} + \delta E_{\mathrm{ind}} + \delta E_{\mathrm{sr}}$$

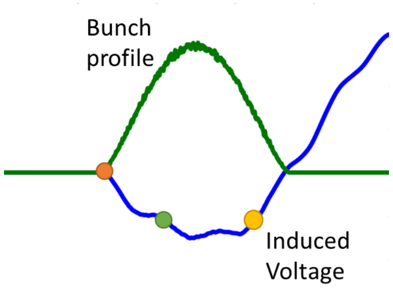
where $g$ can be an arbitrary RF waveform, multiple RF systems $i$ can be used with time varying programs (feedback systems), self induced fields ($\delta E_{\mathrm{ind}}$) and synchrotron radiation ($\delta E_{\mathrm{sr}}$) are also included.

# WAKEFIELDS IN ACCELERATORS

- A real accelerator is composed of many equipment, which can lead for example to discontinuities in the beam pipe aperture.





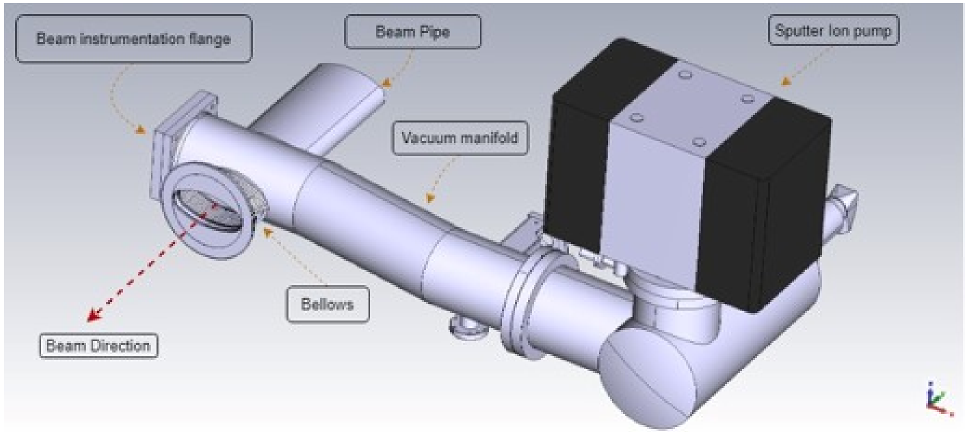


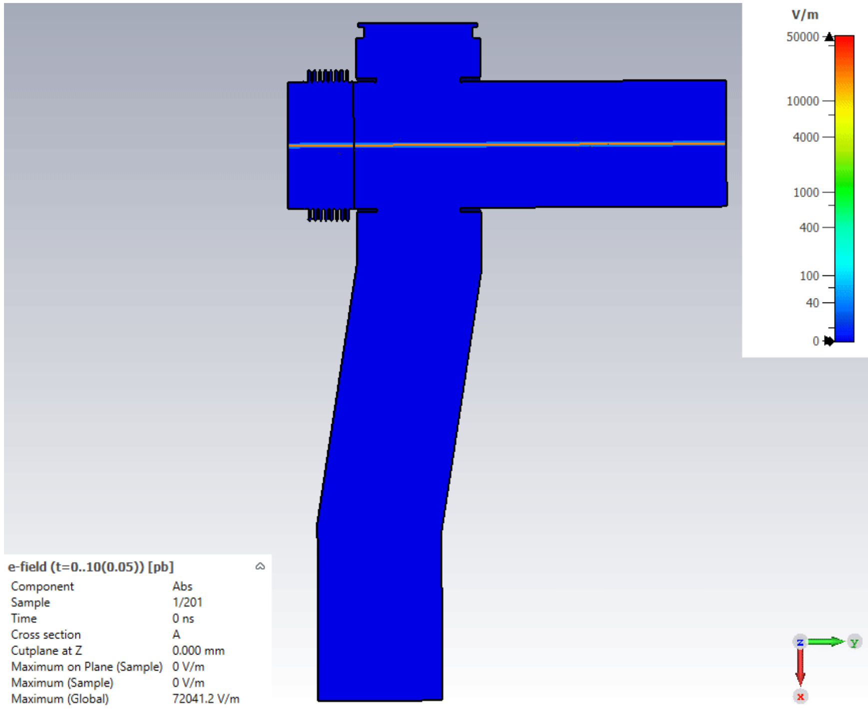

- A single particle passing through a cavity-like gap will induce a wakefield $\mathcal{W}\left(\tau\right)$. A bunch with a longitudinal charge density $\lambda\left(\tau\right)$ (number of particles $N_b$) will induce a voltage $V_{\mathrm{ind}}\left(\tau\right)$, as a convolution product of all the particles single wakes

$$\delta E_{\mathrm{ind}}\left(\tau\right) = qV_{\mathrm{ind}}\left(\tau\right) = -qN_b\left(\lambda * \mathcal{W}\right)$$

# EXAMPLE OF SIMULATED WAKEFIELDS

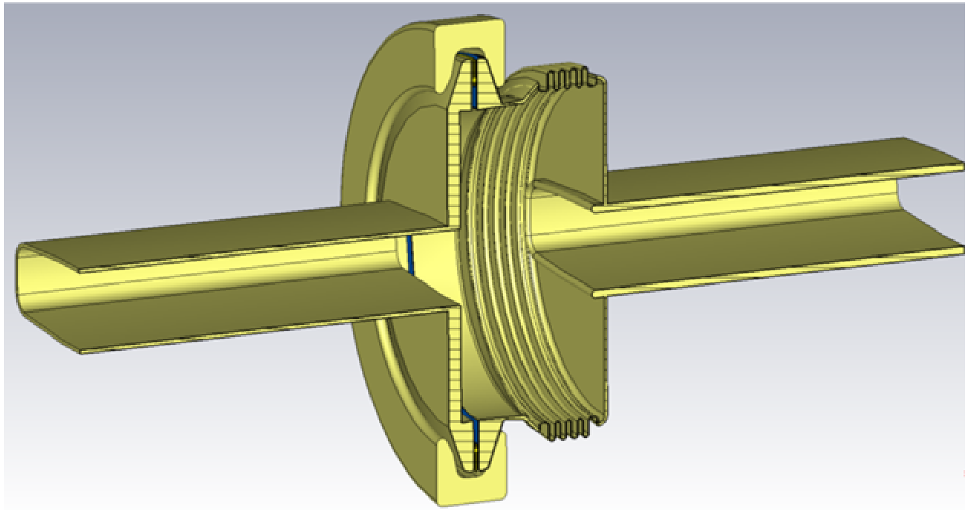
Wakefields simulated for a single passage into a vacuum chamber with a discontinuity (pumping manifold). Simulation with CST Studio Suite®

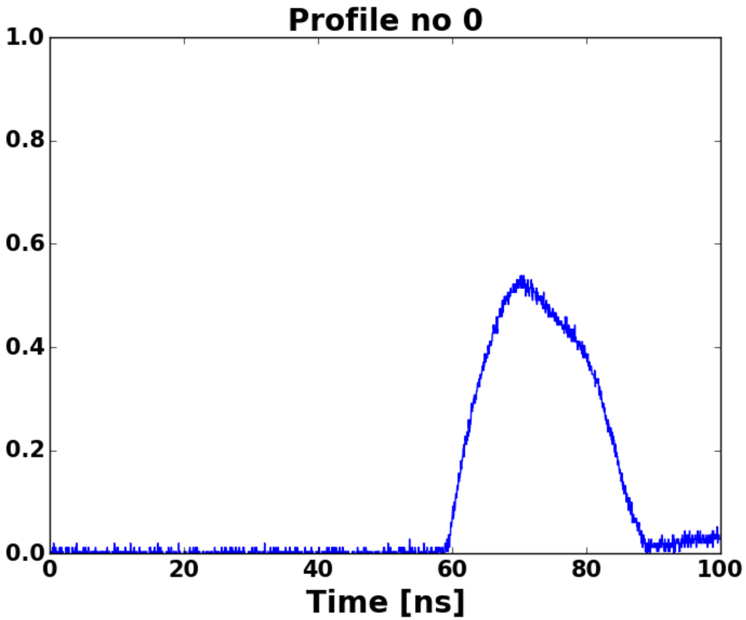


# BEAM INDUCED VOLTAGE EFFECT ON THE BEAM

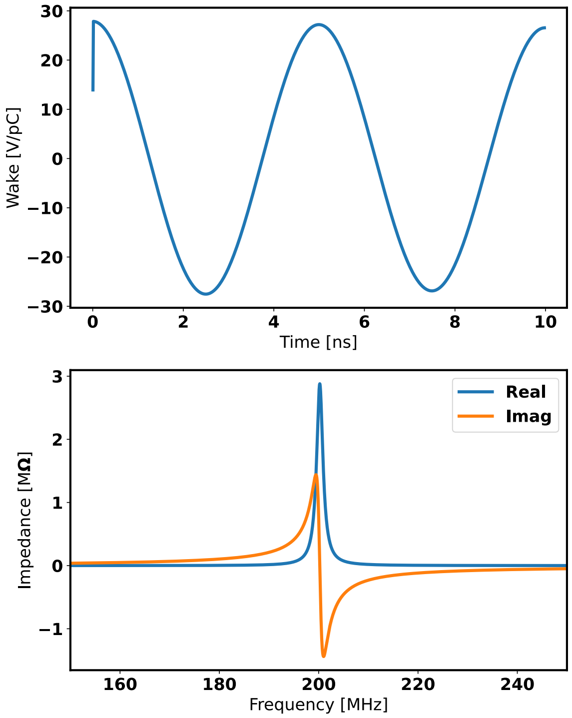
Example of a bunch is injected with RF off, the bunch profile is modulated only by wakefields (many passages).





# MODELING IMPEDANCE SOURCES

## EXAMPLE OF THE RESONATOR




- Wakefields can often be represented by a resonator. In time domain, we have

$$W(t>0) = 2\alpha R_s e^{-\alpha t}\left(\cos\tilde{\omega}t - \frac{\alpha}{\tilde{\omega}}\sin\tilde{\omega}t\right)$$

$$W(0) = \alpha R_s \quad , W(t<0) = 0$$

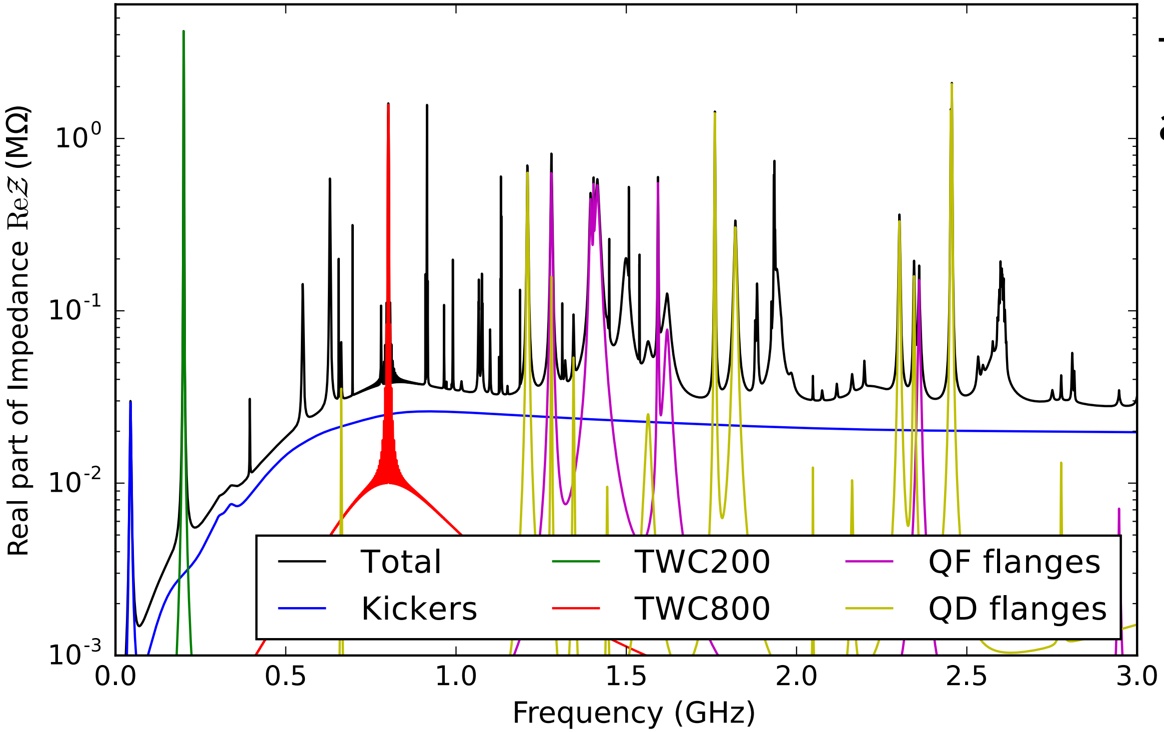
- In frequency domain, refered to as impedance

$$Z(\omega) = \frac{R_s}{1 + jQ\left(\frac{\omega}{\omega_r} - \frac{\omega_r}{\omega}\right)}$$

*Shunt impedance* $R_s$, *resonant angular frequency* $\omega_r$, *quality factor* $Q$, *decay time* $\alpha = \omega_r / (2Q)$, $\tilde{\omega} = \sqrt{\omega_r^2 - \alpha^2}$

# EXAMPLE MACHINE IMPEDANCE MODEL

## THE SPS IMPEDANCE MODEL (2018)




The impedance aims at including all equipment in the ring.

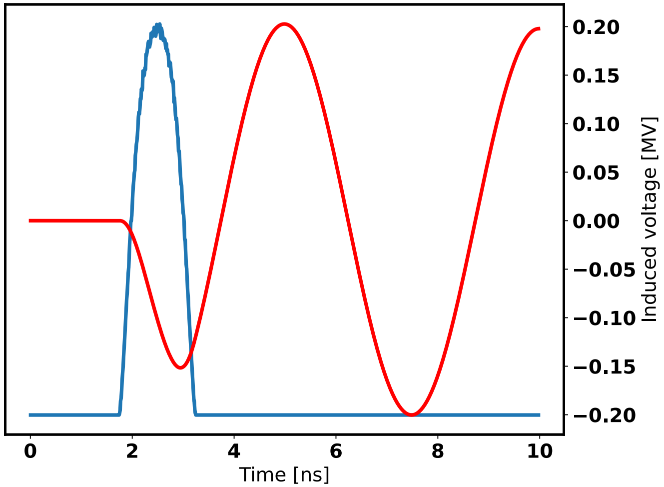
- Cavities (here Traveling Wave Cavity TWC)
- Injection/Extraction systems (kickers)
- Vacuum equimpment and chamber conductivity (e.g. flanges, resistive wall)
- Beam instrumentation
- ...

→ Highly complex configurations can be used in tracking simulations!

# NUMERICAL IMPLEMENTATION

## IN TIME DOMAIN

*A bunch profile $\lambda[n]$ is obtained from an histogram of the bunch distribution in phase space ($n$ is a time index), and is updated every turn during tracking.*




- The induced voltage can then be computed in time or frequency domain. In time domain, a numerical convolution is applied (e.g. *numpy.convolve*)

$$V_{\mathrm{ind}}[n] = -q\,N_p \mathrm{CONV}\left(W[n], \lambda[n]\right)$$

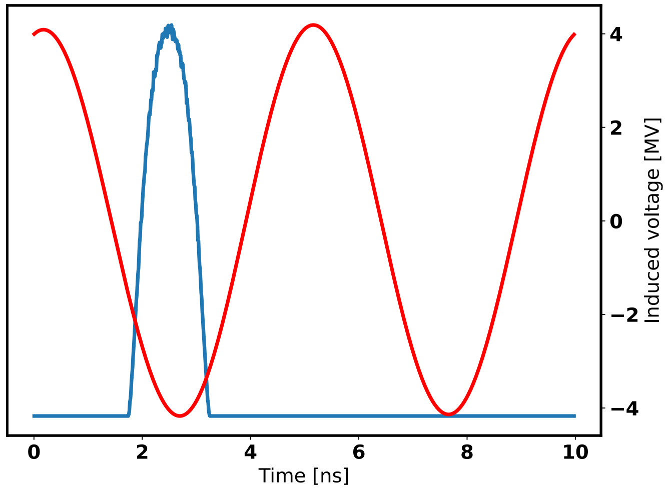
- Linear convolution, the method is non-periodic.

→**Useful for a single bunch or un-even filling of a ring!**

# NUMERICAL IMPLEMENTATION

## IN FREQUENCY DOMAIN

*A bunch profile $\lambda[n]$ is obtained from a histogram of the bunch distribution in phase space ($n$ is a time index), and is updated every turn during tracking.*




- In frequency domain, (fast) Discrete Fourier Transforms (DFTs) are used

$$V_{\mathrm{ind}}[n] = -q\,N_p\,\mathrm{IDFT}\left(Z[k]\cdot\mathrm{DFT}\left(\lambda[n]\right)\right)$$

- Circular convolution, the method is periodic through DFTs.

→ **Useful for multi-bunch, one bunch to represent them all! (one bunch every 10 ns for this example plot)**

# USING BLOND OBJECTS

## ON TOP OF OUR EXISTING SCRIPT (1/2)

- Defining the parameters of the impedance source (e.g. 6 cavities, can be linearly summed!)

```
Rs_cavities = 6*480e3;f_r_cavities = 200.222e6;Q_cavities = 130
```

- Defining the type of impedance sources (here resonators)

```
cavities = Resonators(Rs_cavities, f_r_cavities, Q_cavities)
```

- Then the method to compute the induced voltage, passing a list of resonators (can be extended with more impedance sources)

```
ind_volt_time = InducedVoltageTime(beam, profile, [cavities])
```

- Finally summing all induced voltage contribution into one for computational efficiency

```
total_ind_voltage = TotalInducedVoltage(beam, profile, [ind_volt_time])
```

# USING BLOND OBJECTS

## ON TOP OF OUR EXISTING SCRIPT (2/2)

- This can be finally added to the tracking loop

```
for turn in range(ring.n_turns):
    full_tracker.track()
    profile.track()

    '''
     HERE WE ADD INTENSITY EFFCTS
     '''
    total_ind_voltage.track()

    plots.track()
    bunchmonitor.track()
```

**→ The increased complexity comes with the cost of an increased runtime. How to find the most optimal parameters for computation?**

# WHERE TO FIND THE EXERCISES

**Longitudinal Tracking**

- 0_cheat_sheet.html
- 0_cheat_sheet.pdf
- 0_Foreword.html
- 0_Foreword.pdf
- 0_Installation_instructions
- 0_PythonExample.ipynb
- 1_Introduction.html
- 1_Introduction_optionA_empty.ipynb
- 1_Introduction_optionA_solutions.ipynb
- 1_Introduction_optionB_empty.ipynb
- 1_Introduction_optionB_solutions.ipynb
- 1_Introduction.pdf
- 2_Advanced.html
- 2_Advanced.pdf
- 2_AdvancedTracking_empty.ipynb
- 2_AdvancedTracking_solutions.ipynb
- all_files.tar.gz
- all_files.zip
- all_in_one_tracker.py
- PythonExampleFunctions.py
- support_functions.py

Find the exercises in
Indico (Timetable → Introduction Afternoon courses)

We will need

- 2_AdvancedTracking_empty.ipynb
- all_in_one_tracker.py
- support_functions.py

The all_files archive files allows to get them all at once!

# YOUR MISSION

## WE WOULD LIKE TO INCLUDE INTENSITY EFFECTS IN OUR SIMULATION AND CHECK BEAM STABILITY IN SCSPS.

### CHOOSE ONE APPROACH BELOW AND DON'T HESITATE TO ASK US FOR HELP! TEAM-UP WITH YOUR NEIGHBORS IF YOU WANT.

→ **Use the premade `blond_complete_tracker` (when relevant)**

→ **Build experience using BLonD**

→ **You can also continue with the Introduction to Tracking if you prefer!**

## DON'T LET THE "ADVANCED" SCARE YOU. YOU'VE ALREADY CLIMBED HIGHER MOUNTAINS!!