Introduction to: wakis

3D Electromagnetic Time-Domain wake and impedance solver

10th Low Emittance Rings Workshop, Hamburg, Germany

Elena de la Fuente[1,2] &

(Wakis team) Lorenzo Giacomel, Giovanni Iadarola[1], Carlo Zannini[1], Giovanni Rumolo[1], Manuel Cotelo[2,]
(Iddefix team) Sébastien Joly[3], Malthe Raschke, Bernardo Figueiredo

Extended thanks to the users and people that has already shown interest in the code ☺

[1]CERN (BE-ABP-CEI), Geneva Switzerland
[2]IFN-GV, Polytechnic University of Madrid, Madrid, Spain
[4]Helmholtz-Zentrum, Berlin, Germany

Outline

1. Introduction to beam-coupling impedance simulations
2. Overview of `Wakis`:
   1. Numerical method: the Finite Integration Technique
   2. CSG, CAD geometry import and materials
   3. Code structure overview
   4. Results and benchmarking against CST® Studio
3. Recent advancements to tackle impedance challenges
4. Some user↔developer examples ☺
5. Conclusions and future work

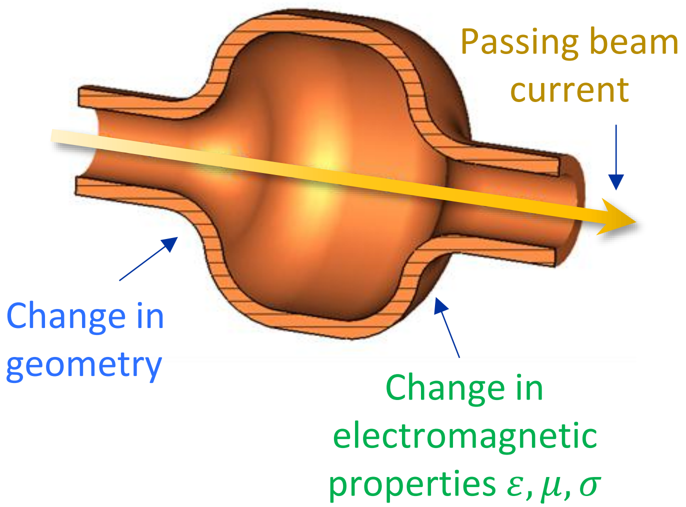
Outline

1. **Introduction to beam-coupling impedance simulations**
2. Overview of `Wakis`:
   i. Numerical method: the Finite Integration Technique
   ii. CSG, CAD geometry import and materials
   iii. Code structure overview
   iv. Results and benchmarking against CST® Studio
3. Recent advancements to tackle impedance challenges
4. Some user↔developer examples ☺
5. Conclusions and future work

Introduction to beam-coupling impedance

- Electromagnetic wakefields are generated as the particle beam traverses the different accelerator devices with discontinuities in the geometry (cavities, transitions…) or the electromagnetic properties (ε, μ, σ).
  - These will affect the trailing particles/bunches and can generate instabilities and beam-induced heating

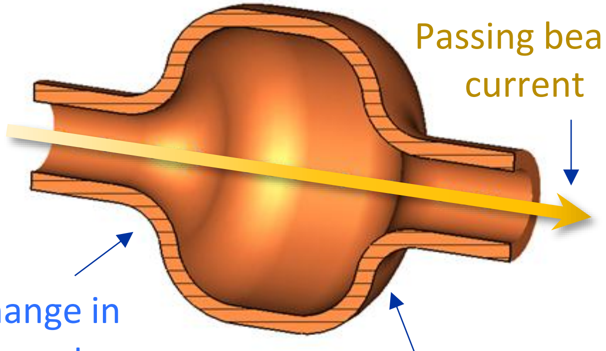
Accelerator device



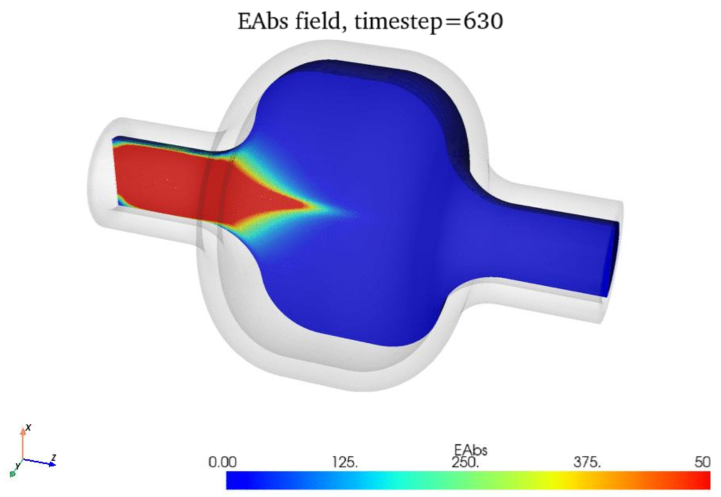
Introduction to beam-coupling impedance

- Electromagnetic wakefields are generated as the particle beam traverses the different accelerator devices with discontinuities in the geometry (cavities, transitions…) or the electromagnetic properties (ε, μ, σ).
    - These will affect the trailing particles/bunches and can generate instabilities and beam-induced heating

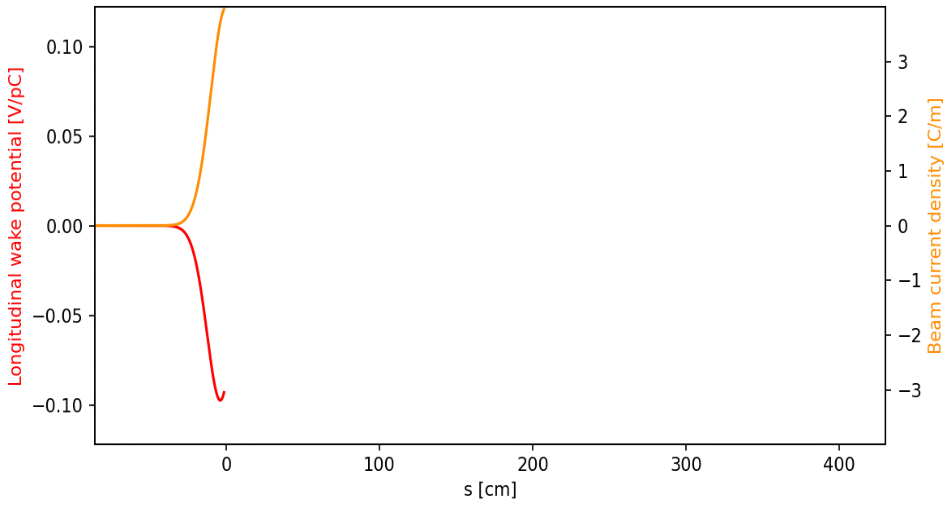
Accelerator device



Electromagnetic fields



Wake potential & Impedance



- The beam-coupling impedance $\boldsymbol{Z}$ is a frequency-dependent property of each accelerator device, used to quantify the wakefields' effects

$$\boldsymbol{Z}_{||}(\omega) = -\frac{\int_{-\infty}^{\infty} W_{||}(s) e^{-i\omega s} ds}{\int_{-\infty}^{\infty} \beta c \lambda(s) e^{-i\omega s} ds}$$

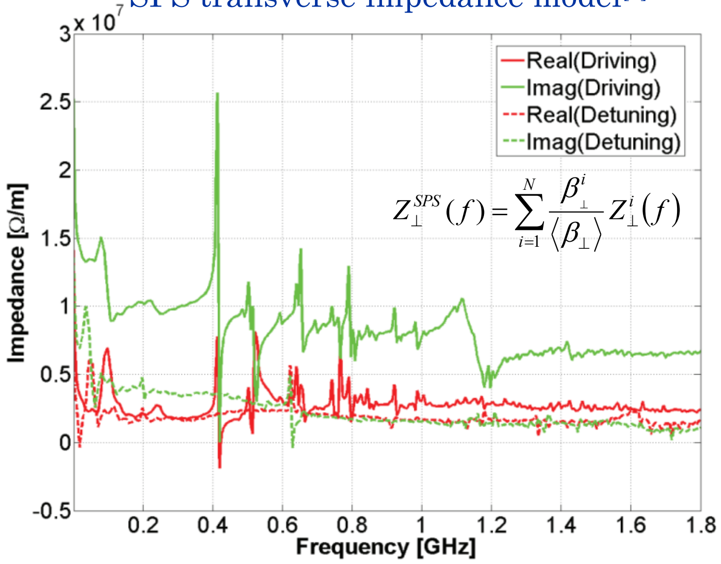
[1] C. Zannini, MOPJE049, IPAC'15
[2] E. de la Fuente et al., THAFP01, HB'23
[3] C. Antuono et al., TUAN3, IPAC'24

Importance of beam-coupling impedance

Impedance model of the accelerator

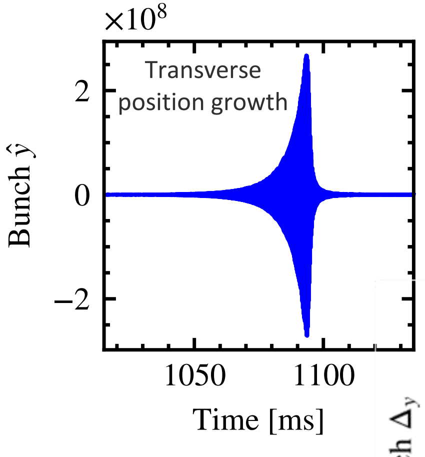
The impedance of all relevant accelerator components is gathered in each accelerator's impedance model

SPS transverse impedance model[1]



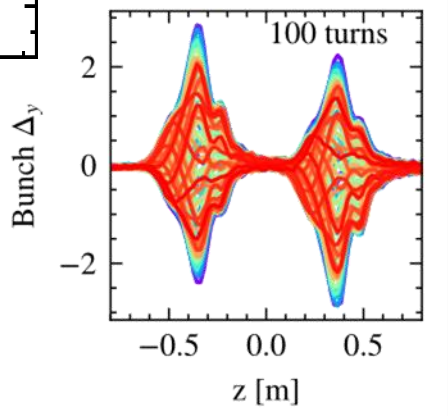
Beam stability

And used in beam-dynamics codes (e.g., pyHeadtail, Xsuite) to predict beam behaviour and stability



SPS Head-Tail mode zero instability [2]

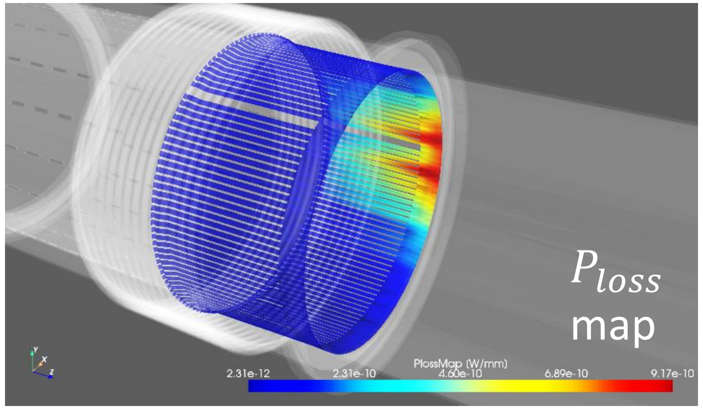
Intrabunch motion (mirrored)



Beam induced heating

Assess beam-induced heating of individual accelerator components and propose mitigation solutions

LHC Warm Vacuum modules Power lost map [3]

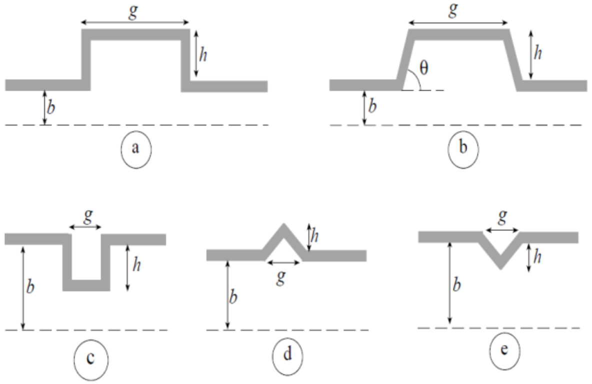


How to compute beam-coupling impedance?

Analytical calculations

Beam-coupling impedance can be derived analytically for simple geometries and material configurations[4]:

- e.g., cylindrical pillbox cavities, pipe transitions, single-layer resistive wall

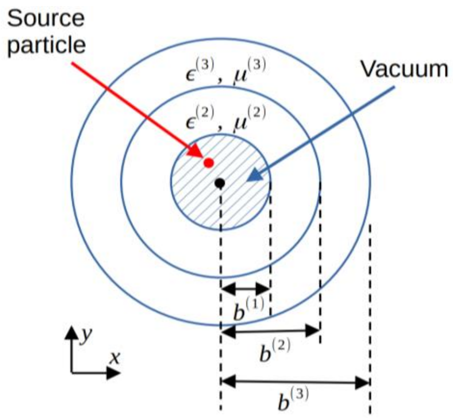


[4] Wake fields and impedance,
L. Palumbo, V.G. Vaccaro, CERN-1995-006.331

Numerical computations

For more complex accelerator devices, the numerical solution of Maxwell's equations in frequency or time-domain is required.

- In frequency domain:
  - 2D models for vacuum chamber multi-layered elements e.g., ImpedanceWake2D[5]
  - Commercial CST® Eigenmode solver → loss-less resonant modes of general 3D structures without excitation + impedance in post-processing
- In time-domain:
  - Time-domain allows for full-wave electromagnetic simulations including losses and excitation bunch. Several codes available following different approaches e.g.:
    - ABCI (axi-symmetric), CST® Wakefield solver (3D), GdfidL®, PBCI, ECHO-3D (3D, moving window) → not open-source



[5] ImpedanceWake2D,
N. Mounet, TUO1C02 HB'10
https://gitlab.cern.ch/IRIS/IW2D

How to compute beam-coupling impedance?

Analytical calculations

Beam-coupling impedance can be derived analytically for simple geometries and material configurations[4]:

- e.g., cylindrical pillbox cavities, pipe transitions, single-layer resistive wall

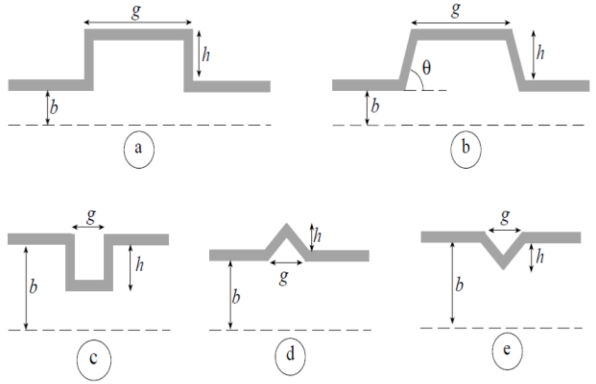
Numerical computations

For more complex accelerator devices, the numerical solution of Maxwell's equations in frequency or time-domain is required.

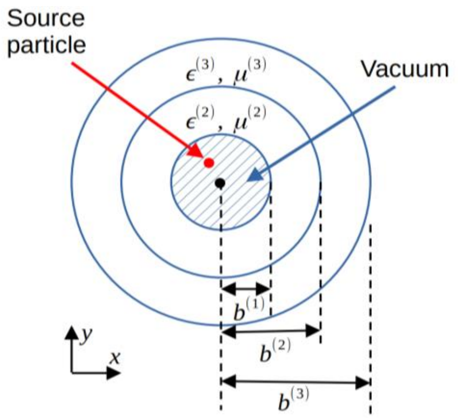
Traditionally handled only by *commercial / closed-source* software such as SIMULIA CST STUDIO SUITE

✓ Well-stablished & robust

× Flexible / extendable for new accelerator challenges



[4] Wake fields and impedance, L. Palumbo, V.G. Vaccaro, CERN-1995-006.331



[5] ImpedanceWake2D, N. Mounet, TUO1C02 HB'10 https://gitlab.cern.ch/IRIS/IW2D

Outline

1. Introduction to beam-coupling impedance simulations
2. **Overview of `Wakis`:**
   i. **Numerical method: the Finite Integration Technique**
   ii. **CSG, CAD geometry import and materials**
   iii. **Code structure overview**
   iv. **Results and benchmarking against CST® Studio**
3. Recent advancements to tackle impedance challenges
4. Some user↔developer examples ☺
5. Conclusions and future work

How to compute beam-coupling impedance?

Time-domain Maxwell's equations....

$$\oint_{\partial A} \boldsymbol{E} \cdot d\boldsymbol{s} = -\iint_A \frac{\partial \boldsymbol{B}}{\partial t} \cdot d\boldsymbol{A}$$

$$\oint_{\partial A} \boldsymbol{H} \cdot d\boldsymbol{s} = -\iint_A \left(\frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}\right) \cdot d\boldsymbol{A}$$

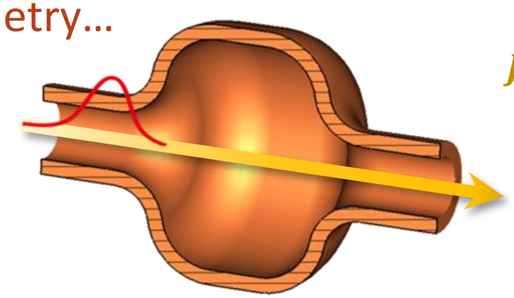
$$\oiint_{\partial V} \boldsymbol{B} \cdot d\boldsymbol{A} = \boldsymbol{0}$$

$$\oiint_{\partial V} \boldsymbol{D} \cdot d\boldsymbol{A} = \iiint_V \boldsymbol{\rho}\, d\boldsymbol{V}$$

$$\boldsymbol{D} = \varepsilon \boldsymbol{E}, \quad \boldsymbol{B} = \mu \boldsymbol{H}, \quad \mathbf{J} = \sigma \boldsymbol{E} + \boldsymbol{J_{src}}$$

on each accelerator device geometry...

...with a passing beam current



$$J_{z,\,src}(x_s, y_s, z) = \frac{q\beta c}{\sqrt{2\pi\sigma_z}} e^{\frac{-(s-s_0)^2}{2\sigma_z^2}}$$

To obtain figures of merit...

- **Wake potential** in time-domain

r_s: beam source path
r_t: integration path

$$\boldsymbol{W}(x_s, y_s, x_t, y_t, s) = \frac{1}{q_1}\int_{-\infty}^{\infty} dz[\boldsymbol{E_d}(x_s, y_s, x_t, y_t, z, t) + c\boldsymbol{e_z} \times \boldsymbol{B_d}(x_s, y_s, x_t, y_t, z, t)]_{t=\frac{(s+z)}{c}}$$

- **Beam-coupling impedance** in frequency-domain

$$Z_{||}(\omega) = -\frac{\int_{-\infty}^{\infty} W_{||}(s) e^{-i\omega s} ds}{\int_{-\infty}^{\infty} c\lambda(s) e^{-i\omega s} ds}$$

$$Z_{\perp}(\omega) = i\frac{\int_{-\infty}^{\infty} W_{\perp}(s) e^{-i\omega s} ds}{\int_{-\infty}^{\infty} c\lambda(s) e^{-i\omega s} ds}$$

Deconvolution with charge line distribution $\lambda(s)$

Finite Integration Technique

Maxwell Equations (Integral form)

$$\oint_{\partial A} \boldsymbol{E} \cdot d\boldsymbol{s} = -\iint_A \frac{\partial \boldsymbol{B}}{\partial t} \cdot d\boldsymbol{A}$$

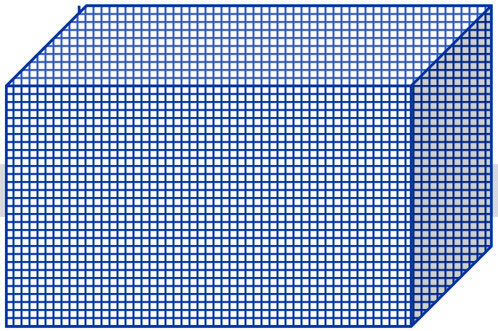
$$\oint_{\partial A} \boldsymbol{H} \cdot d\boldsymbol{s} = -\iint_A \left(\frac{\partial \boldsymbol{D}}{\partial t} + \boldsymbol{J}\right) \cdot d\boldsymbol{A}$$

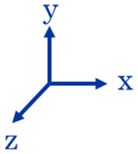
$$\oiint_{\partial V} \boldsymbol{B} \cdot d\boldsymbol{A} = 0$$

$$\oiint_{\partial V} \boldsymbol{D} \cdot d\boldsymbol{A} = \iiint_V \boldsymbol{\rho}\, d\boldsymbol{V}$$

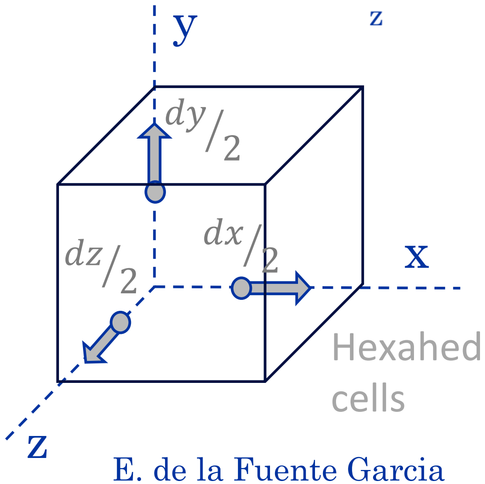
$$\boldsymbol{D} = \varepsilon \boldsymbol{E}, \quad \boldsymbol{B} = \mu \boldsymbol{H}, \quad \boldsymbol{J} = \sigma \boldsymbol{E} + \boldsymbol{J}_{\mathbf{src}}$$

1st approximation
Domain discretization
dx, dy, dz





$$N_{cells} = N_x N_y N_z$$



Hexahedral cells

Maxwell Grid Equations

$$\boldsymbol{C}\boldsymbol{D}_s\boldsymbol{e} = -\boldsymbol{D}_A \frac{\partial \boldsymbol{b}}{\partial t}$$

$$\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{D}}_s\boldsymbol{h} = \widetilde{\boldsymbol{D}}_A \left(\frac{\partial \boldsymbol{d}}{\partial t} + \boldsymbol{j}\right)$$

$$\boldsymbol{S}\boldsymbol{D}_A\boldsymbol{b} = \mathbf{0}$$

$$\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{D}}_A \left(\frac{\partial \boldsymbol{d}}{\partial t} + \boldsymbol{j}\right) = \mathbf{0}$$

$$\boldsymbol{d} = \widetilde{\boldsymbol{M}}_{\boldsymbol{\varepsilon}}\boldsymbol{e}, \quad \mathbf{b} = \boldsymbol{M}_{\boldsymbol{\mu}}\boldsymbol{h}, \quad \mathbf{j} = \widetilde{\boldsymbol{M}}_{\boldsymbol{\sigma}}\boldsymbol{e} + \boldsymbol{j}_{\boldsymbol{src}}$$

- Operators
- Grid areas and lengths
- Material tensors

Finite Integration Technique (II)

Maxwell Grid Equations

$$\boldsymbol{C}\boldsymbol{D_s}\boldsymbol{e} = -\boldsymbol{D_A}\frac{\partial \boldsymbol{b}}{\partial t}$$

$$\widetilde{\boldsymbol{C}}\widetilde{\boldsymbol{D}}_{\boldsymbol{s}}\boldsymbol{h} = \widetilde{\boldsymbol{D}}_{\boldsymbol{A}}\left(\frac{\partial \boldsymbol{d}}{\partial t} + \boldsymbol{j}\right)$$

$$\boldsymbol{d} = \widetilde{\boldsymbol{M}}_{\boldsymbol{\varepsilon}}\boldsymbol{e}, \quad \mathbf{b} = \boldsymbol{M_\mu}\boldsymbol{h},$$

$$\mathbf{j} = \widetilde{\boldsymbol{M}}_{\boldsymbol{\sigma}}\boldsymbol{e} + \boldsymbol{j_{src}}$$

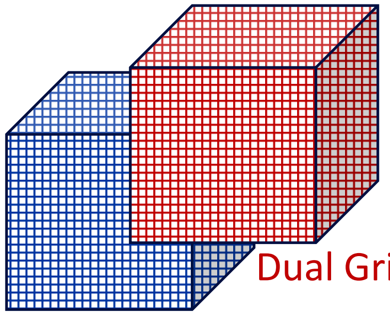
$$\boldsymbol{S}\boldsymbol{D_A}\boldsymbol{b} = 0$$

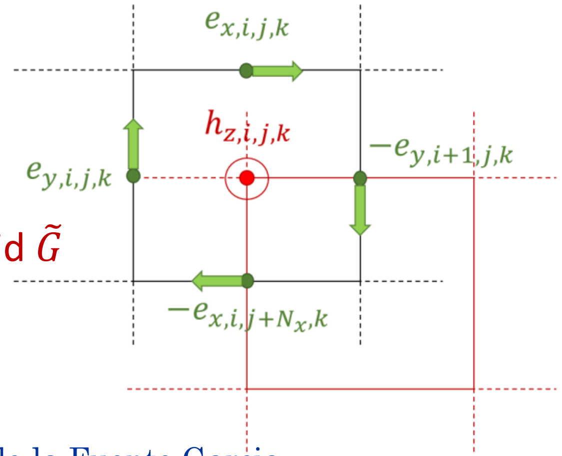
$$\widetilde{\boldsymbol{S}}\widetilde{\boldsymbol{D}}_{\boldsymbol{A}}\left(\frac{\partial \boldsymbol{d}}{\partial t} + \boldsymbol{j}\right) = 0$$

2nd approximation
Time-evolution in timestep Δt

$$\frac{\partial \mathrm{a}}{\partial t} = \frac{\mathrm{a}^{n+1} - \mathrm{a}^{n}}{\Delta \mathrm{t}}$$

Domain discretization
With Yee grid





Time-stepping scheme (leapfrog)

$$\boldsymbol{h}^{n+1} = \boldsymbol{h}^{n} - \Delta t\, \widetilde{\boldsymbol{D}}_{\boldsymbol{s}}\boldsymbol{M}_{\boldsymbol{\mu}}^{-1}\boldsymbol{D}_{\boldsymbol{A}}^{-1}\boldsymbol{C}\boldsymbol{e}^{n+0.5}$$

$$\boldsymbol{e}^{n+1.5} = \boldsymbol{e}^{n+0.5} + \Delta t\boldsymbol{D_s}\widetilde{\boldsymbol{D}}_{\boldsymbol{\varepsilon}}^{-1}\widetilde{\boldsymbol{D}}_{\boldsymbol{A}}^{-1}\widetilde{\boldsymbol{C}}\boldsymbol{h}^{n} - \widetilde{\boldsymbol{M}}_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{j}_{\boldsymbol{src}}^{\boldsymbol{n}} - \widetilde{\boldsymbol{M}}_{\boldsymbol{\varepsilon}}^{-1}\,\widetilde{\boldsymbol{M}}_{\boldsymbol{\sigma}}\,\boldsymbol{e}^{n+0.5}$$

- ✓ Symplectic integrator, 2nd order
- ✓ No matrix inversion needed during time loop
- ✓ Can pre-compute most of the quantities: ↑ speed

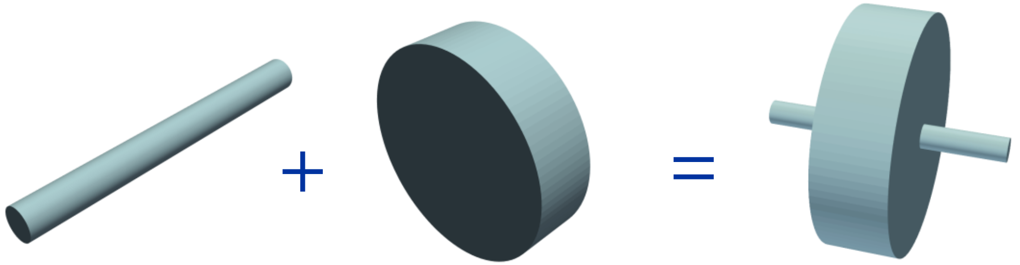
  Taste of speed: 20M cells, 22.3 it/s on Titan V GPU, ~8 Gb memory
- ✓ CFL stability condition
- ▪ Gauss Laws (for E and H) considered satisfied* $\forall\, t$ for no charges j_{ext}

CAD geometry import w/ PyVista

[1] Sullivan, C., & Kaszynski, A. (2019). *PyVista*
https://doi.org/10.21105/joss.01450

- Building on PyVista [1], wakis can:
  - Build the geometry using **Constructive Solid Geometry (CSG)** principles:

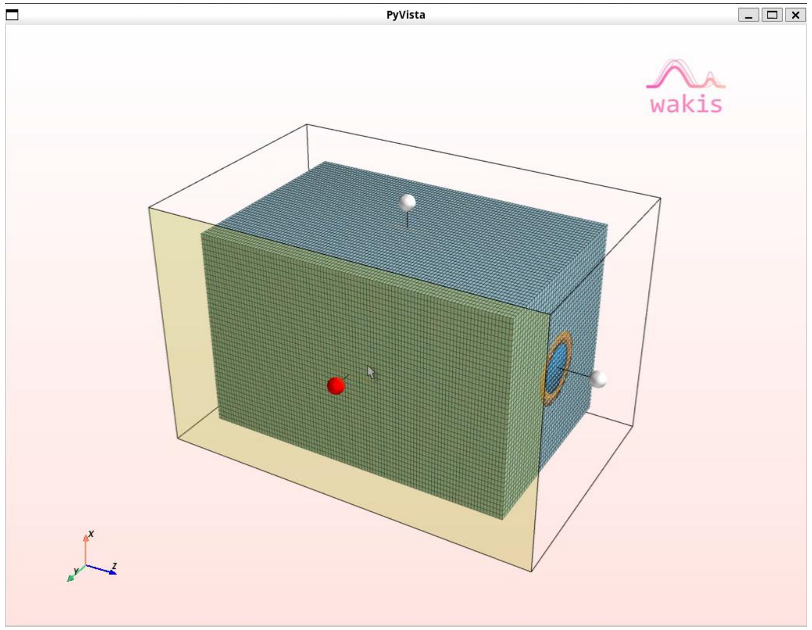
```
cavity = pv.Cylinder(radius=5, height=100) +
    pv.Cylinder(radius=50, height=30)
```



All from python notebooks! ☺

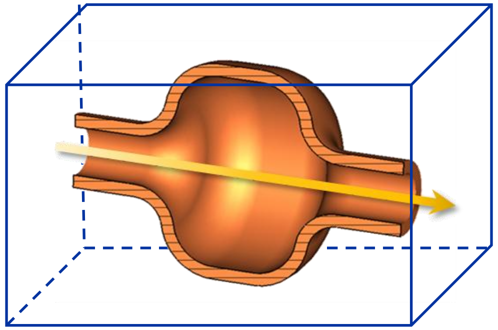
  - Built-in interactive 3D plotting

```
grid.inspect(add_stl='all')
```



  - Read CAD geometry and bounds:

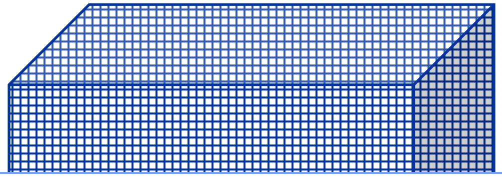
```
geometry = pv.read(CAD_file)
domain_bounds = geometry.bounds
```



  - Generate the domain grid

```
X, Y, Z = np.meshgrid(x, y, z)
grid = pv.StructuredGrid(X, Y, Z)
```

e.g.

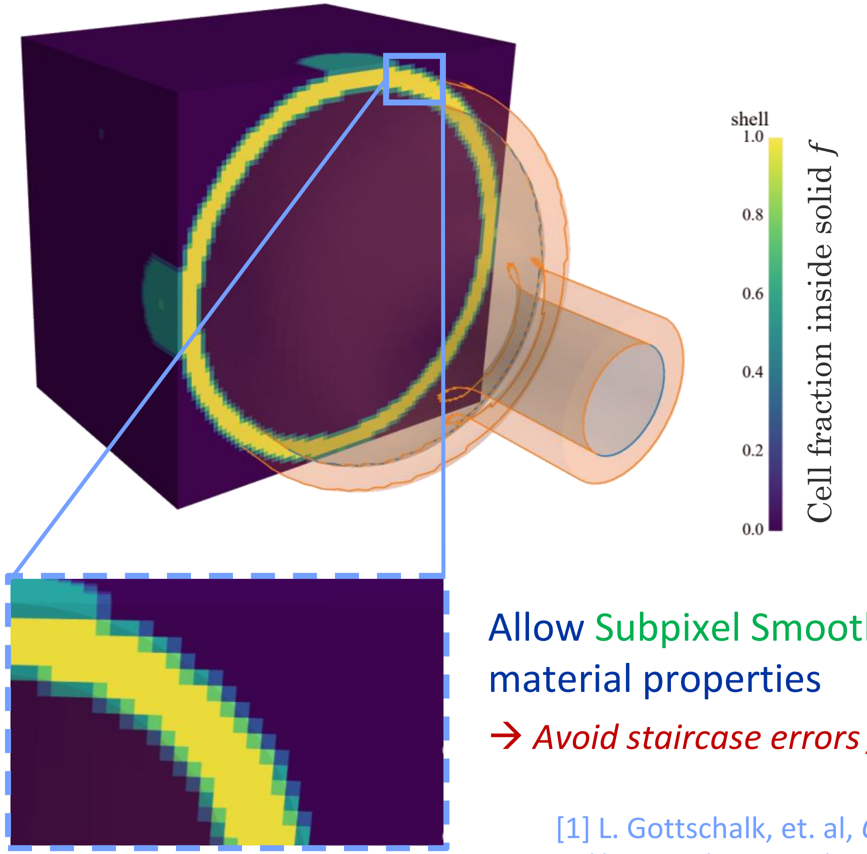


StructuredGrid	Information
N Cells	902400
N Points	931662
X Bounds	-2.600e-01, 2.600e-01
Y Bounds	-2.600e-01, 2.600e-01
Z Bounds	-2.500e-01, 5.500e-01
Dimensions	142, 81, 81

Allocating material properties $\widetilde{M}_{\varepsilon}^{-1}, M_{\mu}^{-1}, \widetilde{M}_{\sigma}$

- Find the domain cells inside each CAD solids with optimized STL↔grid collision filters[1]

```
grid.plot_stl_mask('shell')
```

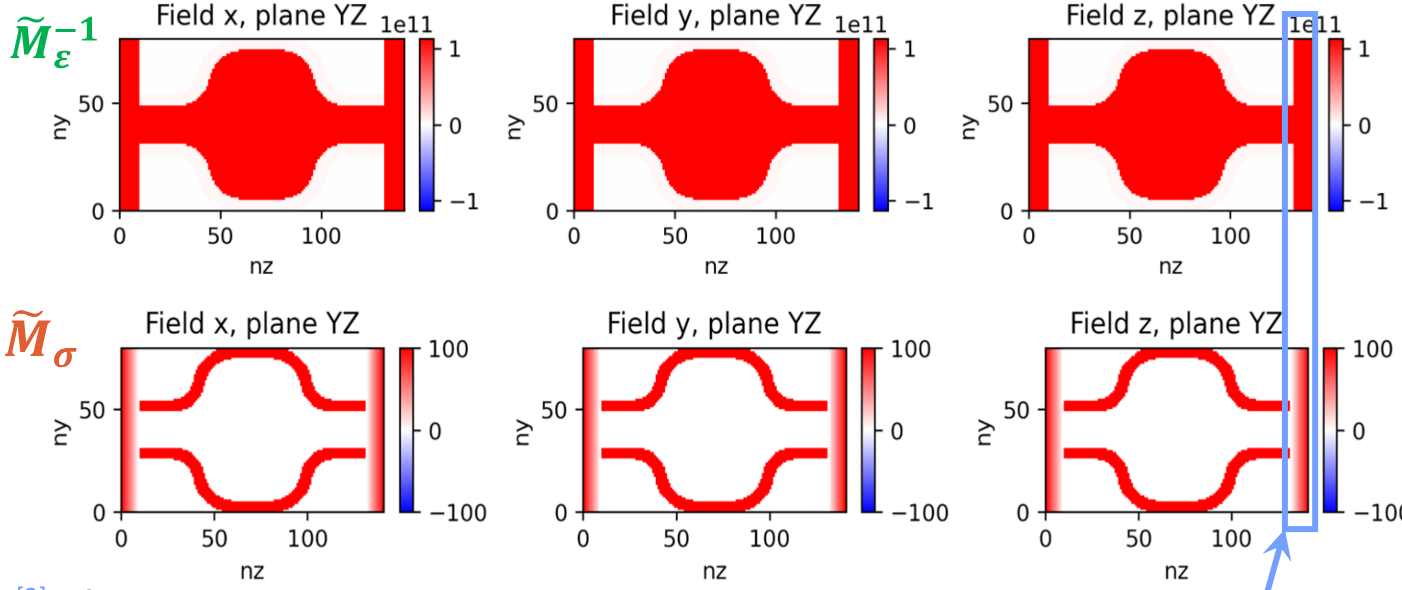


Allow Subpixel Smoothing[2] of material properties

→ *Avoid staircase errors / false PEC cells*

- Assign material properties to each cell inside solid
  → possibility of anisotropic tensors

```
solver.sigma.inspect(plane='YZ')
```



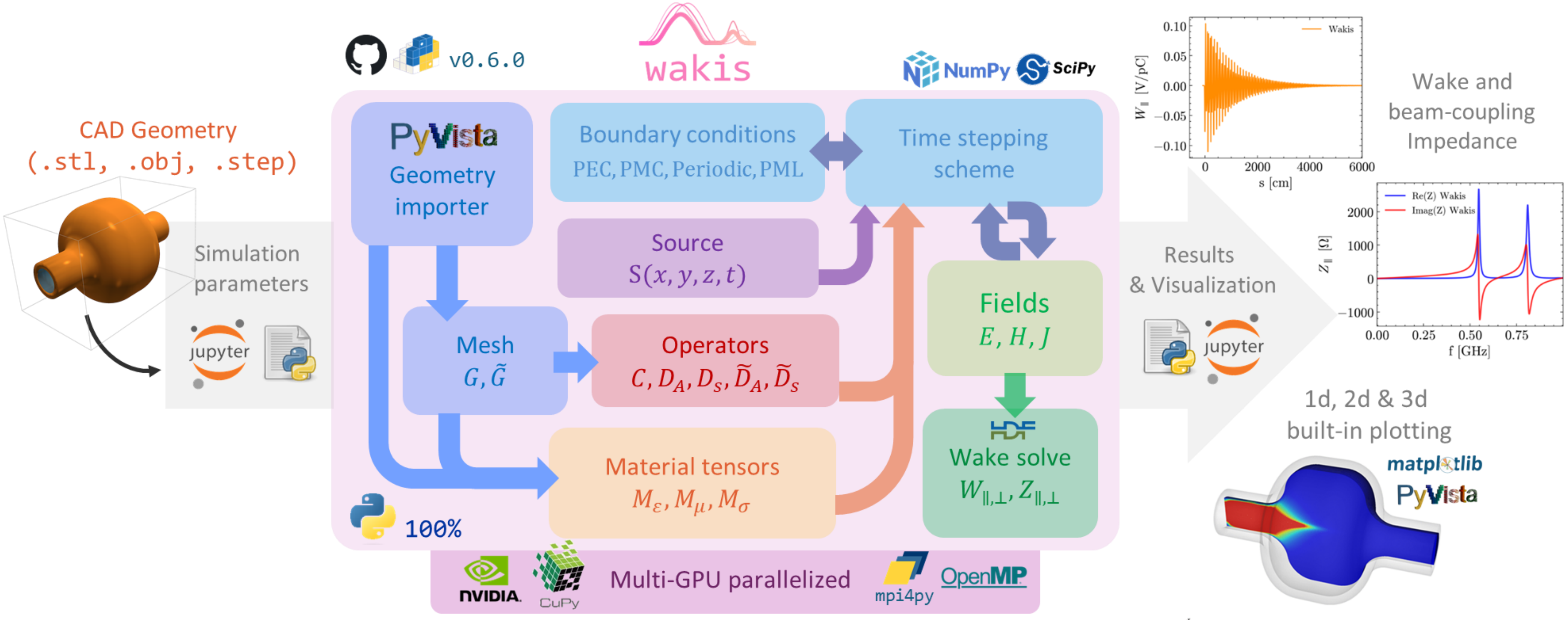
- PML boundary condition region

[1] L. Gottschalk, et. al, *OBBTree*, https://doi.org/10.1145/237170.237244

[2] A. Farjadpour et. al, *Subpixel Smoothing FDTD*, https://doi.org/10.1364/OL.31.002972

Wakis code overview

- Wakis is the 1st **open-source time-domain wakefield solver** tailored to compute **beam-coupling impedance**.
- It computes the integral **general form of Maxwell's equations** using the **Finite Integration Technique (FIT)**



Wakis code overview (II)

Benchmarks against CST studio®

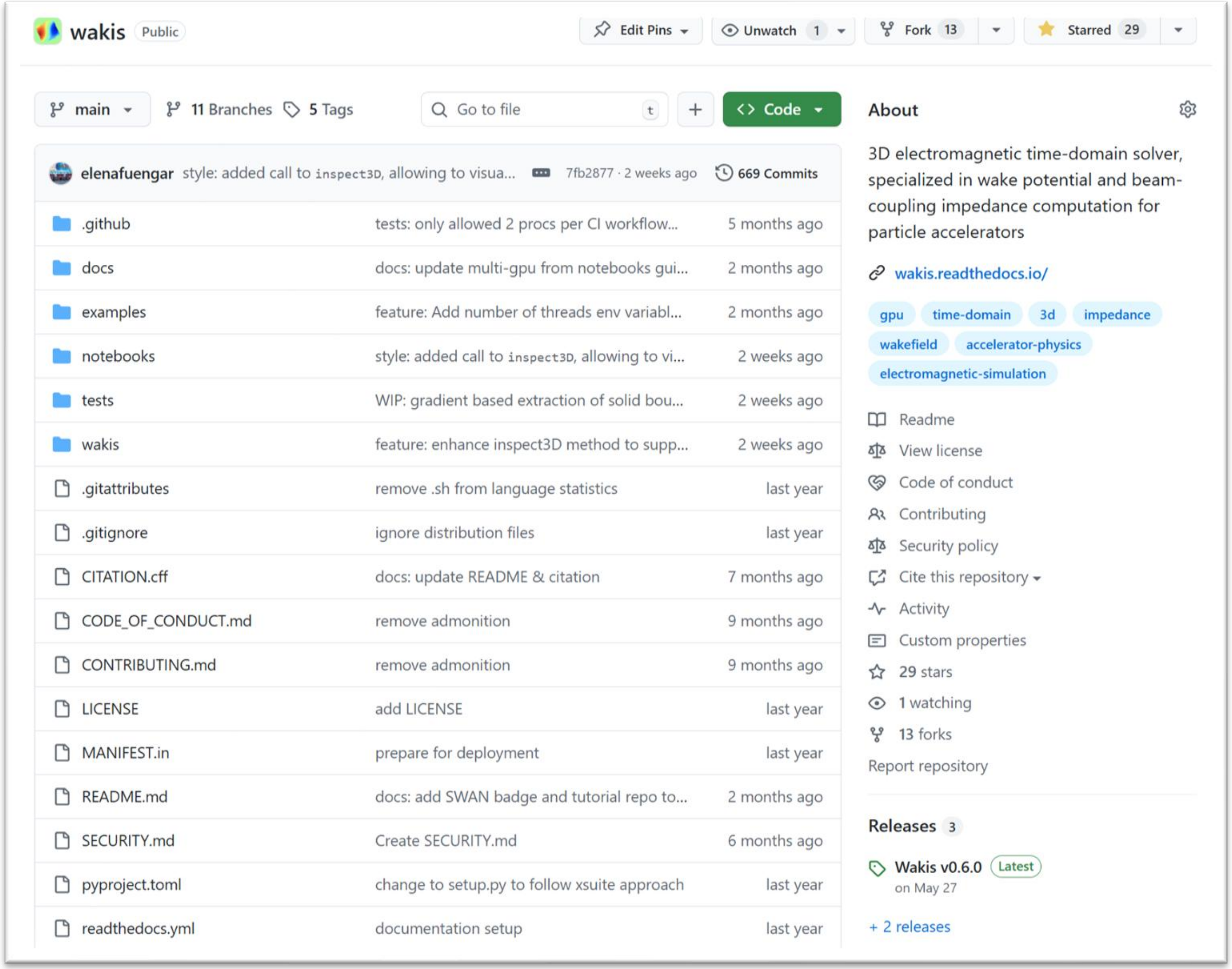
CPU-GPU-MPI Simulation examples ⏳ ~2’ to 20’

Nightly tests CI/CD pipeline

Open source code

[Documentation](#) with Sphinx

Complete impedance workflow playground notebooks



Agile CI/CD

Documented

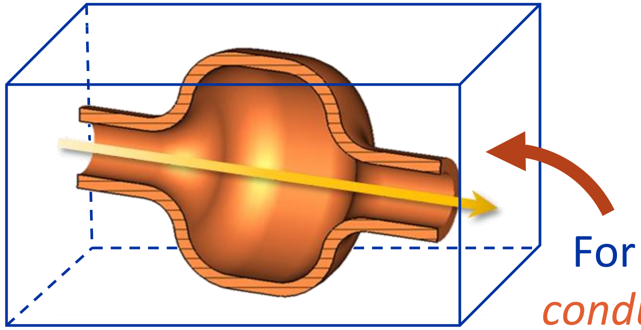
Nightly tested

Check the repository [@ImpedanCEI/wakis](#)

Results & benchmarking

*presented at IPAC25

- 1st Benchmarking campaign conducted on a *lossy Pillbox cavity*:

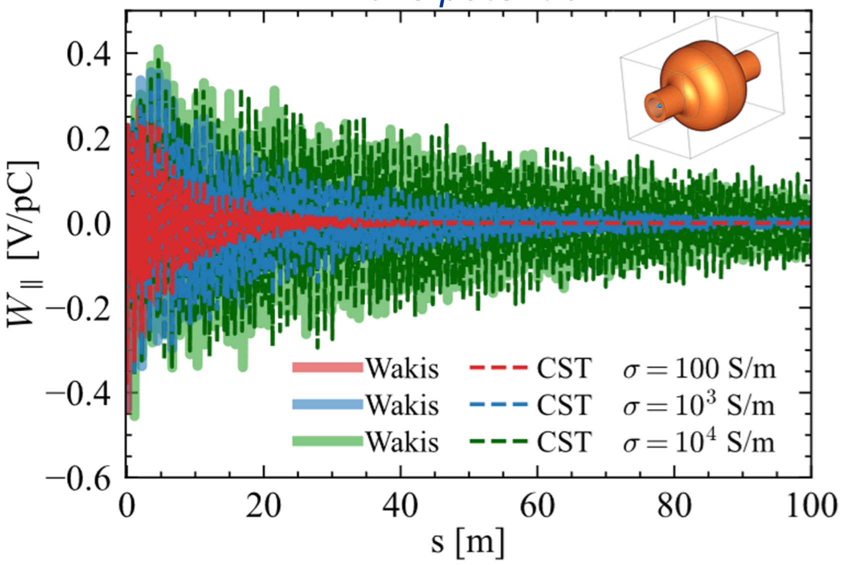


For different *conductivities σ*

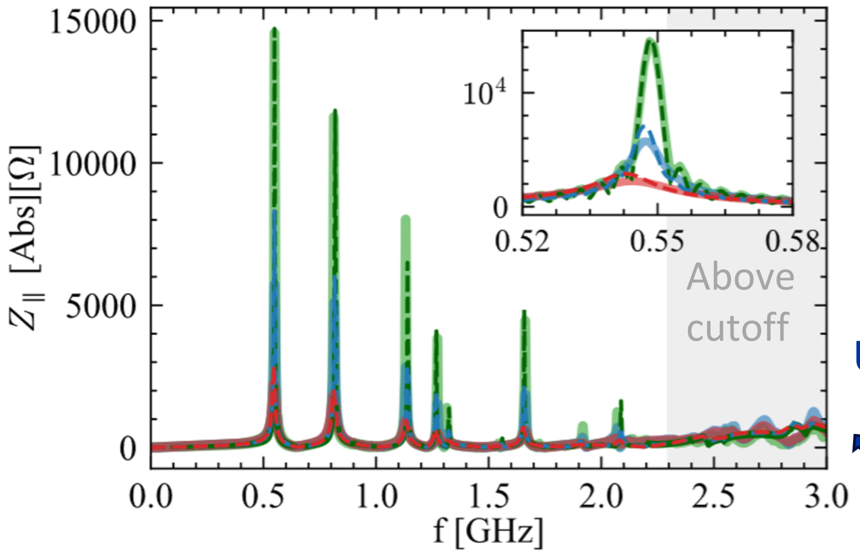
CST STUDIO SUITE — 16-core CPU — 5h 20’ 11’’ — 20M cells

wakis — 1 Tesla T4 GPU — 56’ 10’’ *x5 faster* — Wakis: peak memory 8 Gb

Wake potential



Impedance



Using the PML

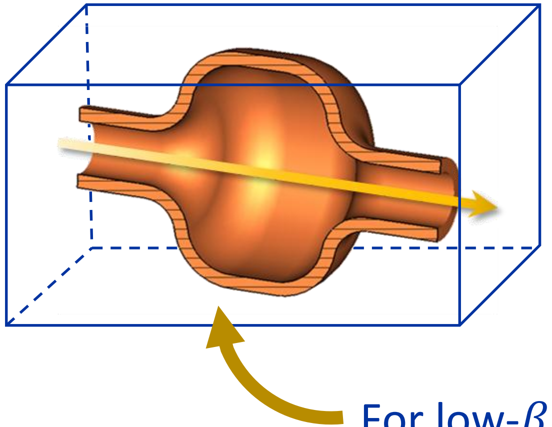
* $\sigma = 10^5 - 10^9$ S/m conductivities require Surface Impedance Boundary Condition (SIBC)

WORK IN PROGRESS

Results & benchmarking (II)

*presented at IPAC25

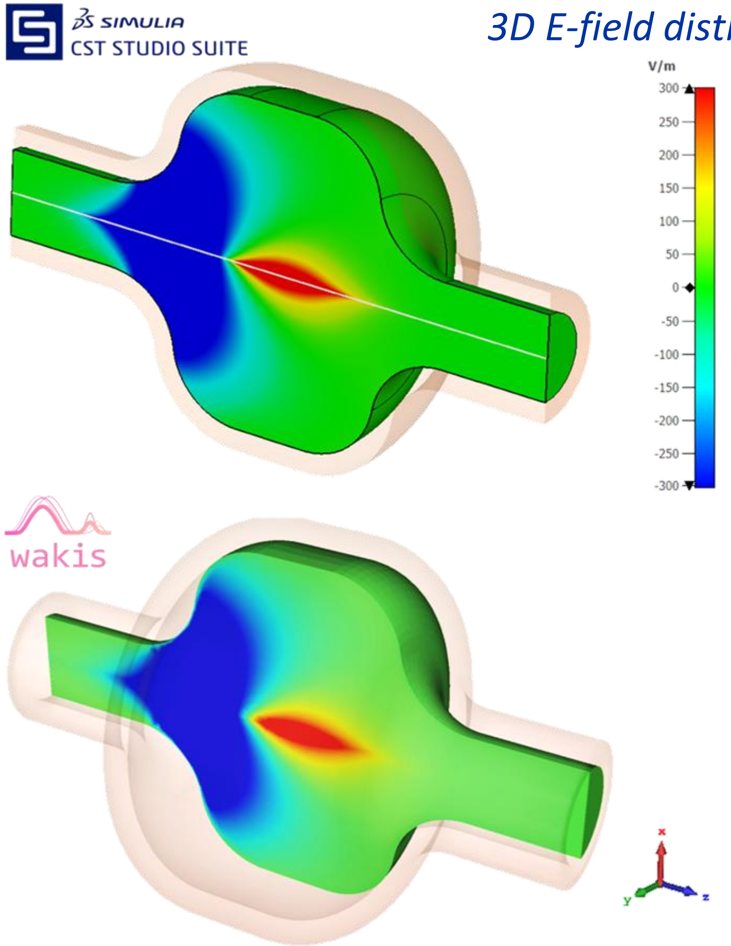
- 1st Benchmarking campaign conducted on a *lossy Pillbox cavity*:



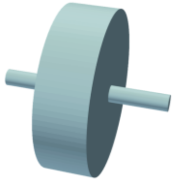
For low-β beam source regime

$\beta = 0.6$

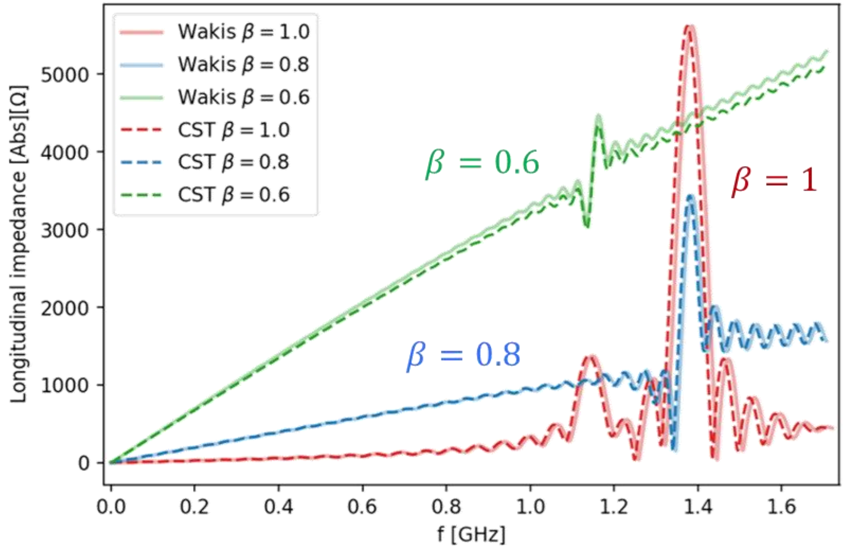
3D E-field distribution



Benchmark 006



Longitudinal impedance
(partially decayed simulation)



Publicly available:
ImpedanCEI/wakis-benchmarks

Outline

1. Introduction to beam-coupling impedance simulations
2. Overview of `Wakis`:
   i. Numerical method: the Finite Integration Technique
   ii. CSG, CAD geometry import and materials
   iii. Code structure overview
   iv. Results and benchmarking against CST® Studio
3. **Recent advancements to tackle impedance challenges**
4. Some user↔developer examples ☺
5. Conclusions and future work

Impedance assessment challenges

FUTURE CIRCULAR COLLIDER

CLIC

International MUON Collider Collaboration

Fields excited above >>pipe cutoff

Absorbing boundary conditions, element crosstalk

Simulations with very short bunches

Dense meshes > 1000 M cells

Slowly decaying wakefields

Simulation length & stability

Impedance in matter

Model & correction of direct space charge impedance

Simulations with non-ultrarelativistic (low β) beams

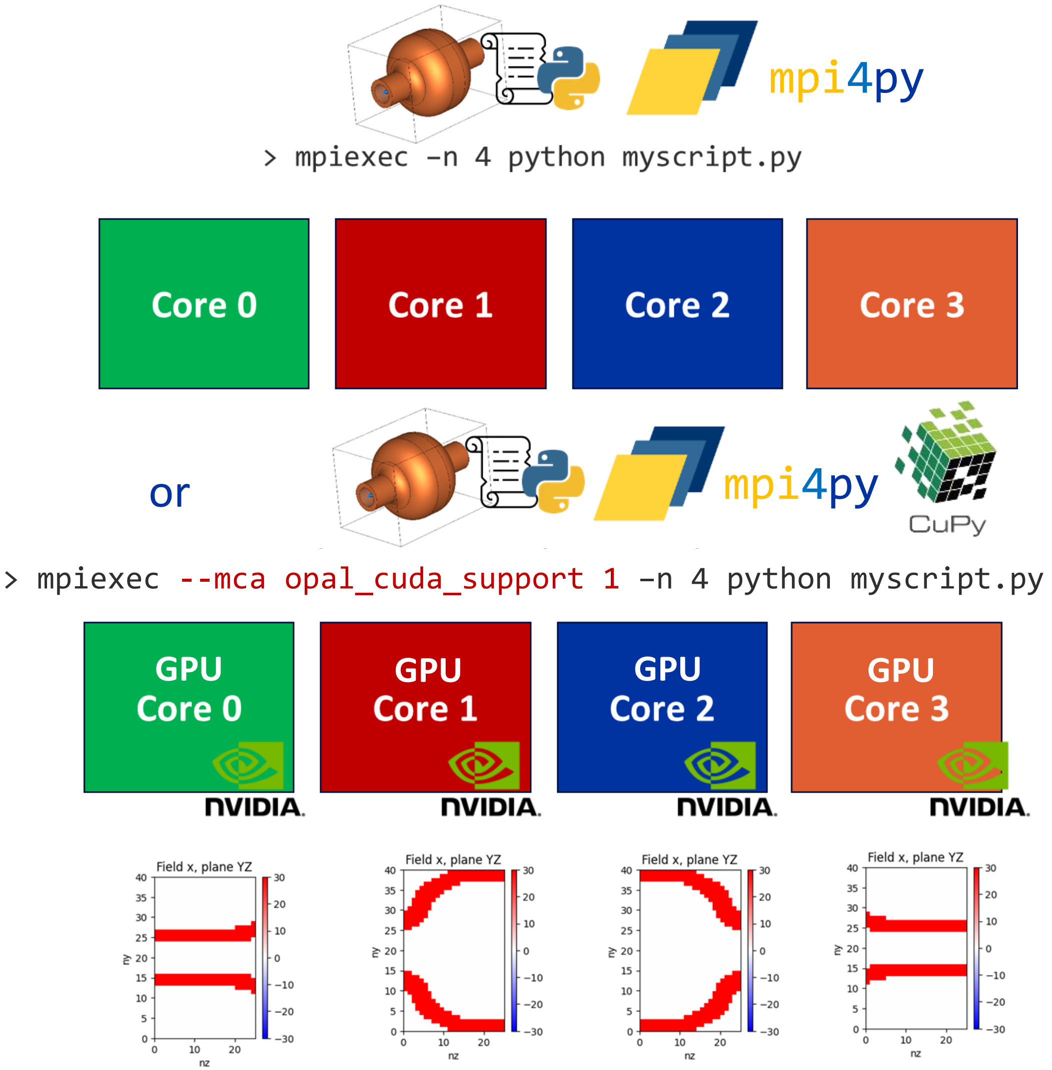
Physics models development

CPU/GPU acceleration in Wakis

- To tackle present & future impedance *challenges*:

Dense meshes > 1000 M cells   **Simulation length & stability**

- Wakis numerical implementation can run on:
  - CPU multi-thread with `numpy/scipy` MKL backend
  - Single GPU with `cupy/cupyx`
  - Multi-CPU nodes with `mpi4py & ipyparallel`
  - Multi-GPU nodes with `mpi4py,ipyparallel & CUDA`
  - Double FP64 and **single precision FP32:**
    - With np.float32 → ½ memory, x30 speedup on TeslaT4 e.g., 20M cells, 4Gb, 560 it/s

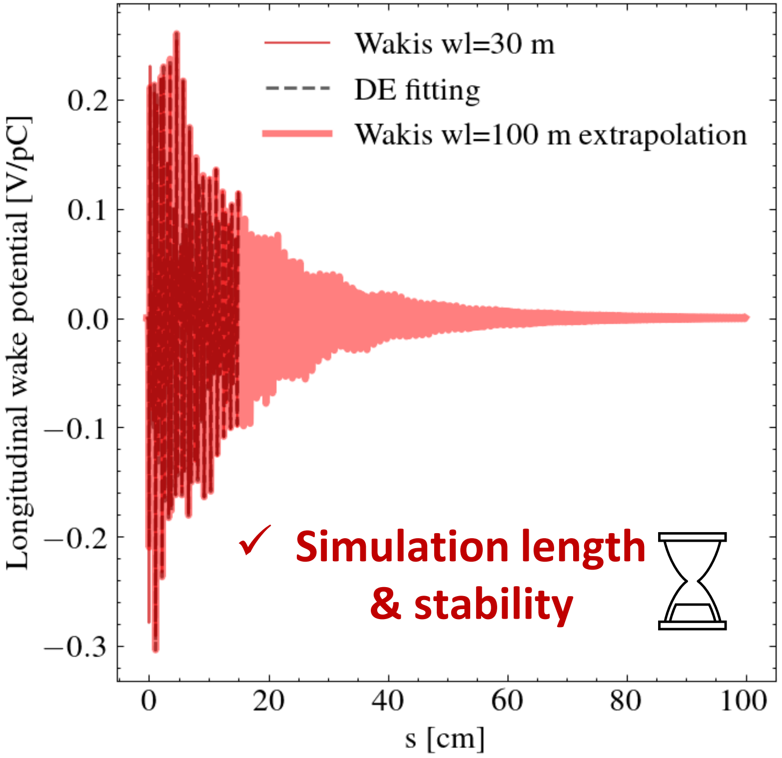


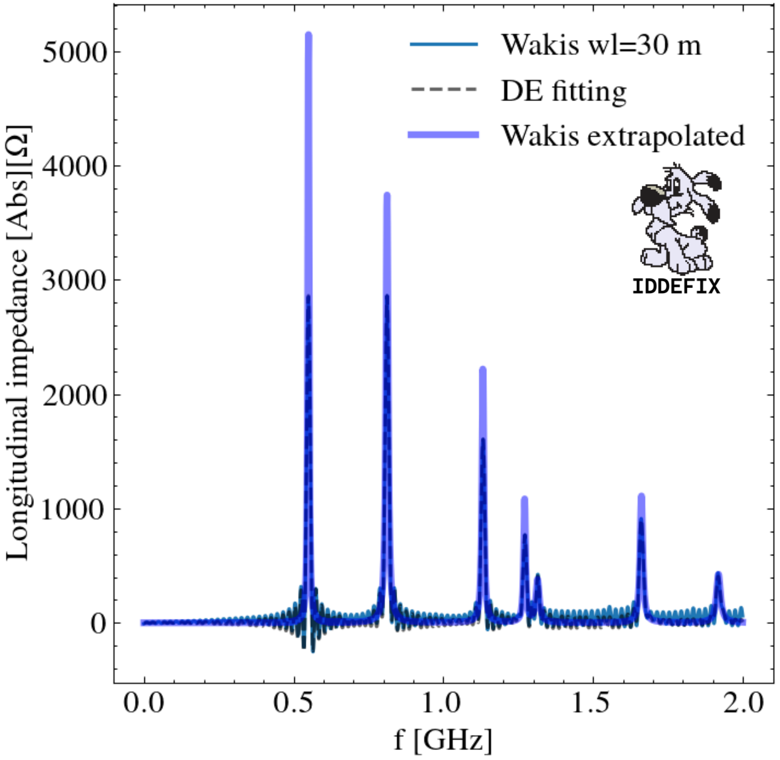
Extrapolation of partially decayed simulations

IDDEFIX

Collaboration with **S. Joly (CERN/Helmholtz)** and 6-month student **Malthe Raschke**

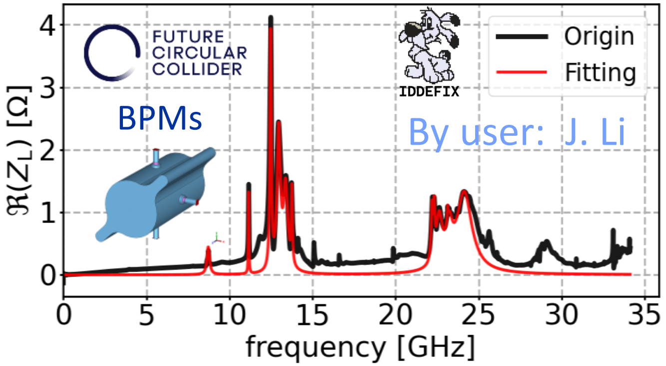
- `IDDEFIX` add-on to `Wakis`: using **resonator formalism + evolutionary algorithms** to **fit and extrapolate analytically** partially decayed simulations



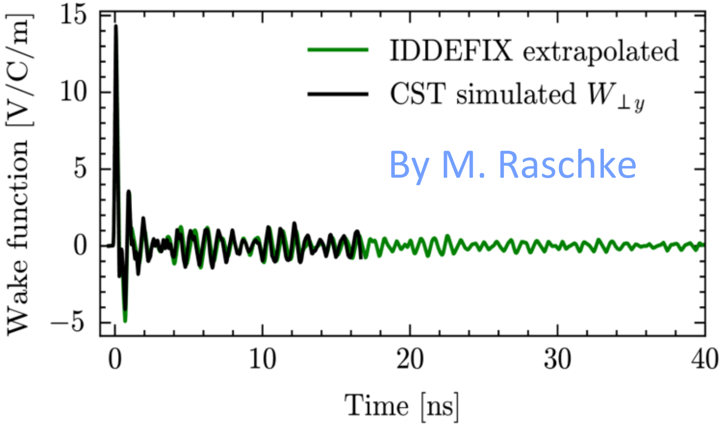


✓ **Improve impedance model flexibility**

E.g.: FCC BPMs Impedance model



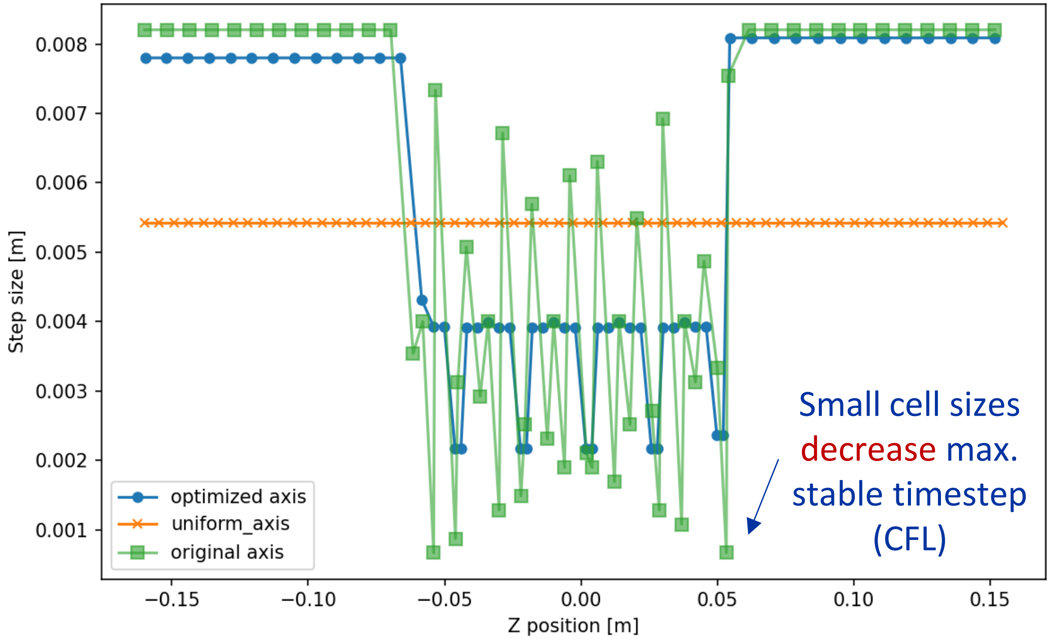
E.g.: Extrapolation of SPS transition wake



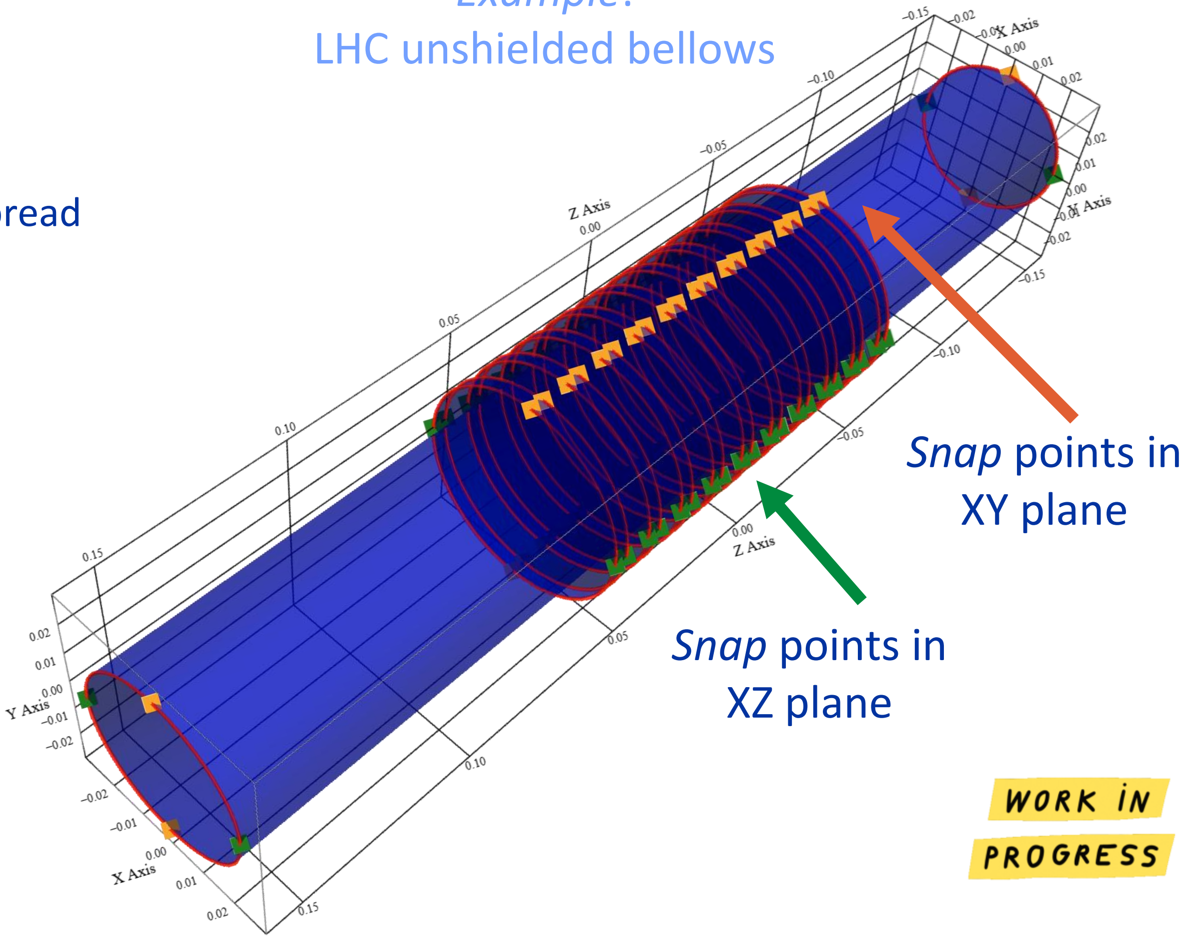
Towards smart meshing

Mesh refinement strategy adopted by OpenFoam & Ansys:

i. Find imported STL surface edge features
ii. Project points to XZ, YZ, XY planes to get *snap points*
iii. Refine mesh around snap points using an *optimizer*
   → *Opt. penalties:* small cell sizes, abrupt changes, spread



Example:
LHC unshielded bellows



WORK IN PROGRESS

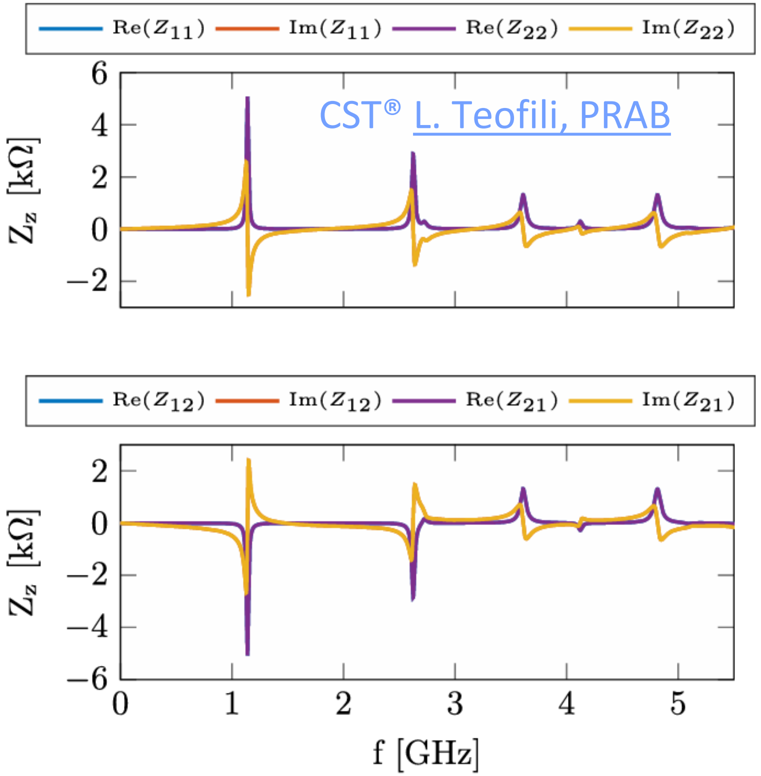
Outline

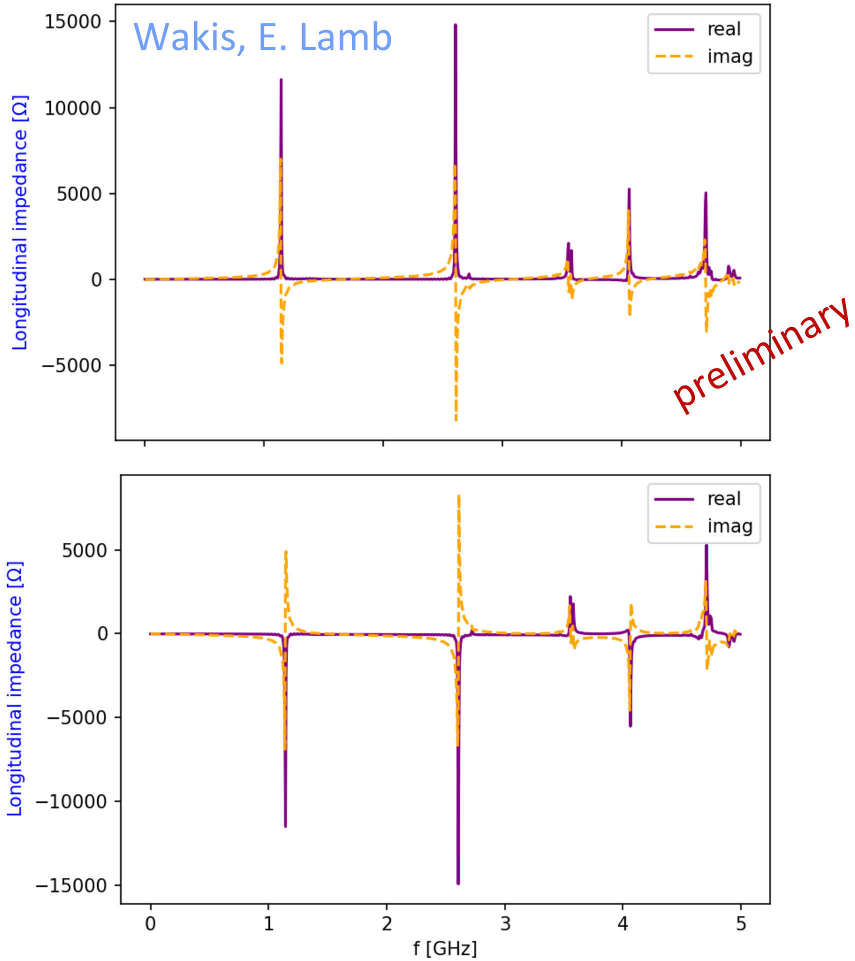
1. Introduction to beam-coupling impedance simulations
2. Overview of `Wakis`:
   i. Numerical method: the Finite Integration Technique
   ii. CSG, CAD geometry import and materials
   iii. Code structure overview
   iv. Results and benchmarking against CST® Studio
3. Recent advancements to tackle impedance challenges
4. **Some user↔developer examples ☺**
5. Conclusions and future work

Users and applications

Thanks to the open-source approach, users can become developers

- **Collaboration with E. Lamb (CERN-SY-RF):** Study of the modes excited by two counter-rotating beams w/ Wakis
  + autonomously implemented Frequency domain monitors

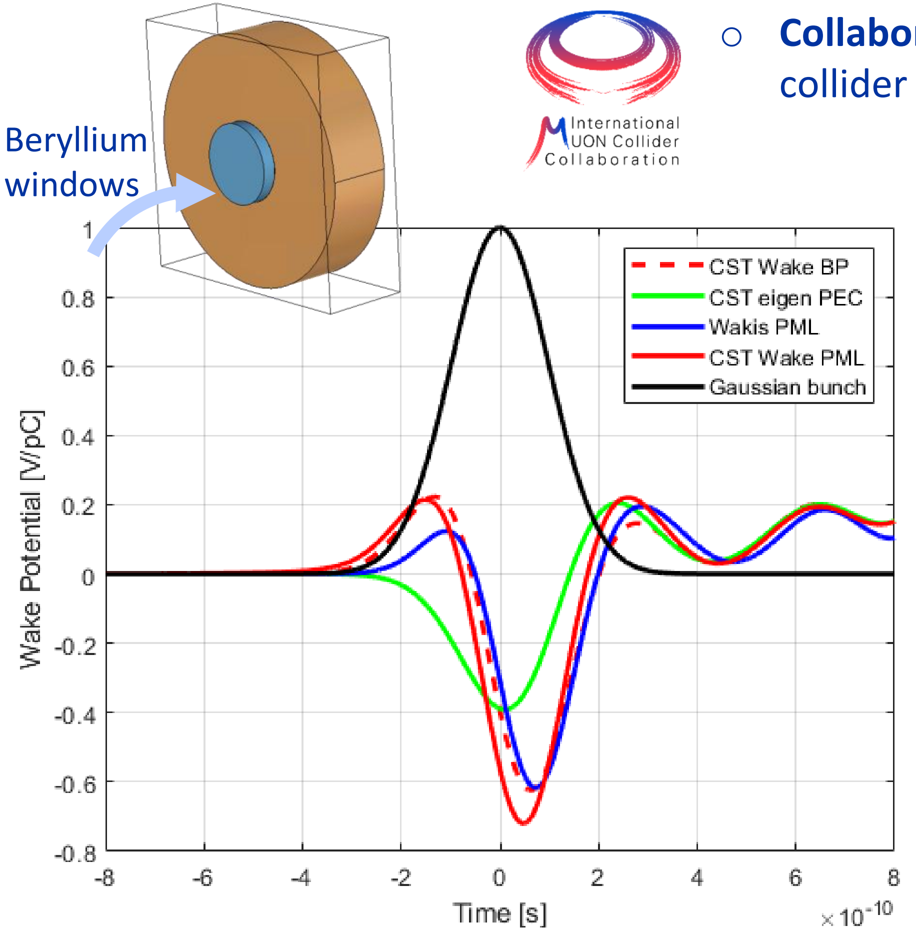


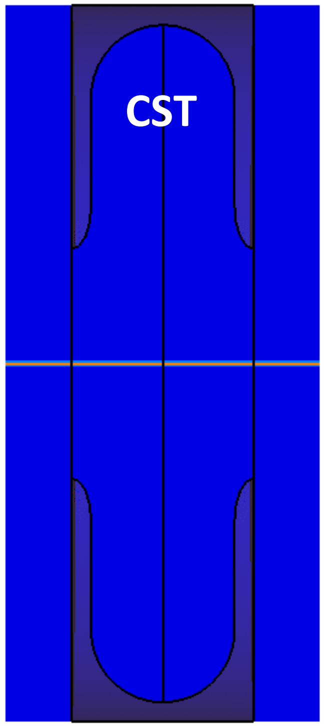


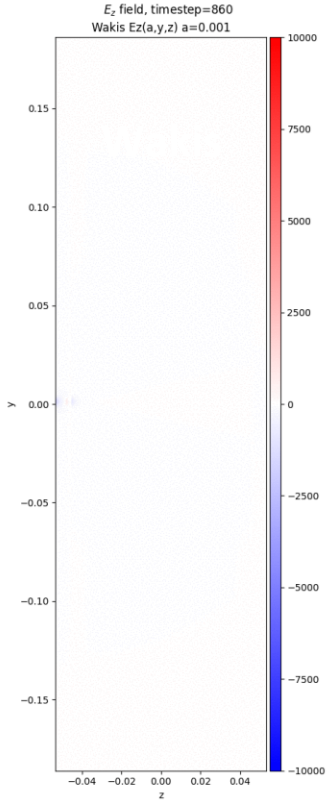
Users and applications (II)

Thanks to the open-source approach, users can become developers

- **Collaboration with C. Barbagallo (CERN-SY-RF):** Study of the voltage of the muon collider RF cavities with Beryllium windows







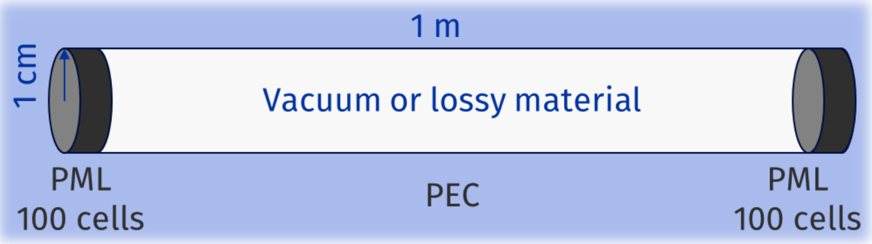
challenging to simulate in CST®

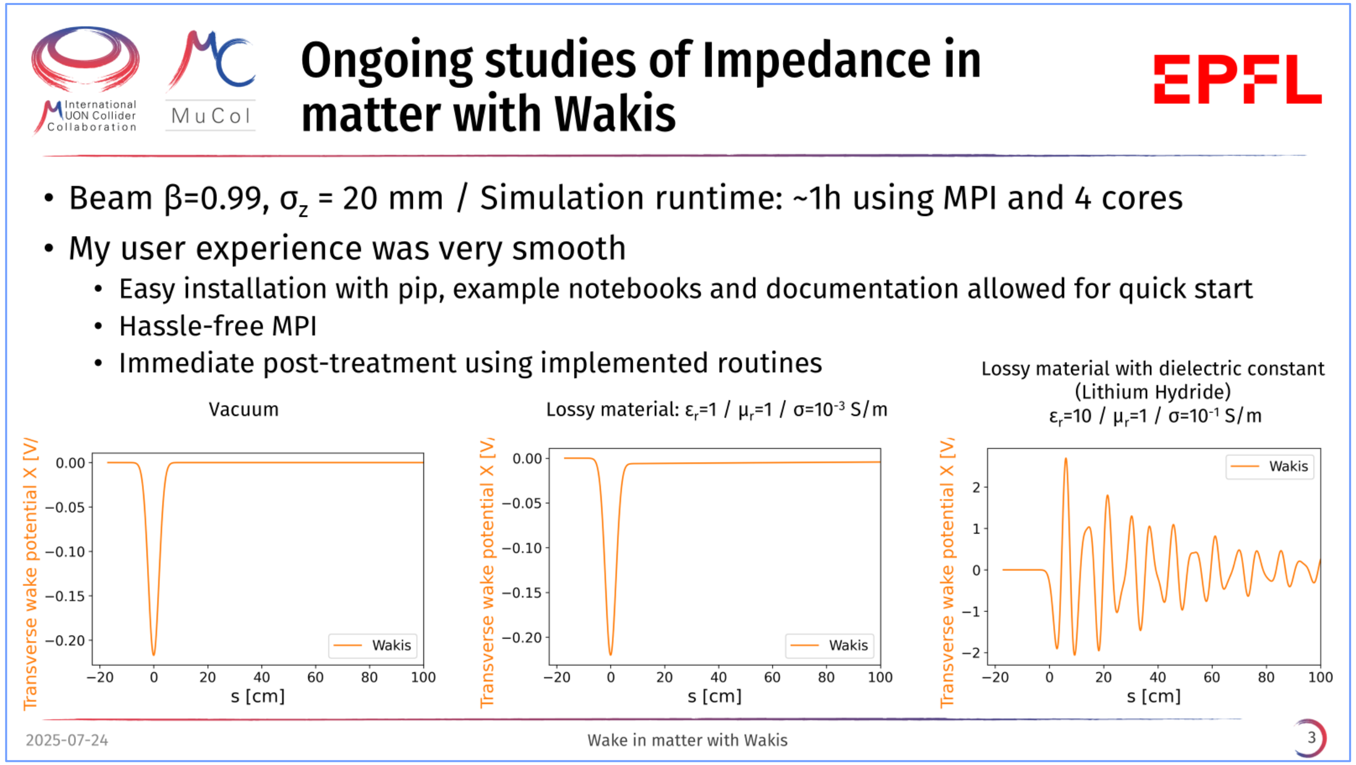
Users and applications (III)

Thanks to the open-source approach, users can become developers

- **Collaboration with D. Amorim (CERN-BE-ABP):** compare 3D EM simulations with model derived from resistive-wall theory by E. Métral (presentation at IMCC annual meeting)

+ upgraded PML implementation to match arbitrary lossy materials





Outline

1. Introduction to beam-coupling impedance simulations
2. Overview of `Wakis`:
   i. Numerical method: the Finite Integration Technique
   ii. CSG, CAD geometry import and materials
   iii. Code structure overview
   iv. Results and benchmarking against CST® Studio
3. Recent advancements to tackle impedance challenges
4. Some user↔developer examples ☺
5. **Conclusions and future work**

Conclusions and future work

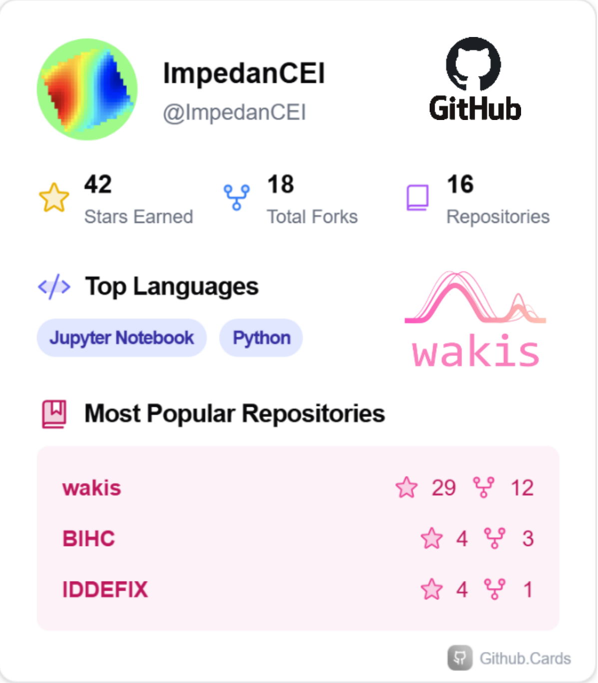
Wakis is **gaining maturity** as the 1st open-source wakefield solver

- **Done**
  - FIT operators based on numpy & scipy
  - BCs: PEC, PMC, Periodic, uniaxial-PML
  - Wake potential and impedance
  - Anisotropic materials ε, μ, σ
  - CAD geometry import
  - State of the art 2d, 3d plotting
  - Automatic testing & CI/CD
  - EM Sources & beam current with β
  - Parallelized & GPU accelerated
  - Wakefield extrapolation with EA (w/ S. Joly)

- **To do:**
  - *Smart refinement for grid (WIP)*
  - *SIBC for high conductive metals (WIP)*
  - Benchmark campaign on complex devices: LHC TCPC, LHC WVM *(WIP)*
  - Improve robustness of PML at high-frequency (**TECH student**, WIP)
  - Dispersive materials + *other user needs!*

- **Users↔Developers:**
  - Ports and field monitors in frequency domain (*E. Lamb*)
  - Impedance in matter (muCol) (*D. Amorim, E. Kvinke*)
  - Correction of direct space charge for low-β simulations (muCol) (C. Barbagallo)
  - *Hopefully more to come!* ☺

STAY TUNED



Community interest

Thank you for the attention ☺

Introduction to: wakis

3D Electromagnetic Time-Domain
wake and impedance solver

Elena de la Fuente García
elena.de.la.fuente.garcia@cern.ch

Acknowledgements

Code development is a big collaborative team effort:

Thanks to contributors (code or expertise):

Wakis team: Lorenzo Giacomel, Giovanni Iadarola, Carlo Zannini, Manuel Cotelo, Giovanni Rumolo

Iddefix team: Malthe Raschke Nielsen, Sébastien Joly, Bernardo Abreu Figueiredo

BIHC team: Leonardo Sito, Francesco Giordano, Jintao Li, Benoit Salvant, Carlo Zannini

Neffint team: Eskil Vik, Nicolas Mounet

+ extended thanks to:
Chiara Antuono, Elena Macchia, David Amorim, Szymon Lopaciuk (CERN),
Jean Luc Vay, Weiqun Zhang (LBNL), Erion Gjonaj (TU-Darmstadt)
& Eduardo Oliva (UPM)
for the very useful discussions

Bibliography sources

IDDEFIX

I. S. Joly, ”Resonator impedance extrapolation of a partially decayed wake” presentation @ CERN ABP-CEI section meeting link
II. S. Joly,”Recent advances in the CERN PS impedance model and instability simulations following the LHC Injectors Upgrade project” PhD Thesis, Sapienza Universita di Roma, link
III. M. Raschke, ”Evolutionary Algorithms for Wakefields” @ CERN ABP-CEI section meeting link
IV. N. Mounet, “The LHC Transverse Coupled-Bunch Instability” PhD Thesis EPFL, link

wakis

I. T. Weiland and R. Wanzenberg, “Wake fields and impedances,” Lect. Notes Phys., vol. 400, pp. 39–79, 1992, doi: 10.1007/3-540-55250-2_26
II. I. Zagorodnov and T. Weiland, “TE/TM field solver for particle beam simulations without numerical Cherenkov radiation,” Phys. Rev. ST Accel. Beams, vol. 8, p. 042001, 2005, doi: 10.1103/PhysRevSTAB.8.042001
III. M. C. Balk, R. Schuhmann, and T. Weiland, “Open boundaries for particle beams within fit-simulations,” Nuclear Instruments and Methods in Physics Research, vol. 558, no. 1, pp. 54–57, 2006, doi: https://doi.org/10.1016/j.nima.2005.11.076.
IV. K. Klopfer, “Computation of Complex Eigenmodes for Resonators Filled With Gyrotropic Materials,” PhD thesis, Technische Universität Darmstadt, Darmstadt, 2014. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/4210/
V. J. A. Moreno, E. Oliva, and P. Velarde, “EMcLAW: An unsplit Godunov method for Maxwell ’s equations including polarization, metals, divergence control and AMR,” Computer Physics Communications, vol. 260, p. e107268, Mar. 2021, doi: 10.1016/j.cpc.2020.107268.
VI. U. Niedermayer, “Determination of Beam Coupling Impedance in the Frequency Domain,” PhD thesis, Technische Universität Darmstadt, Darmstadt, 2016. [Online]. Available: http://tuprints.ulb.tu-darmstadt.de/5157/

About the project:

Project start:

wakis as **post-processing tool**

Computation of wake potential and impedance from **pre-computed fields** from other EM solvers[5]

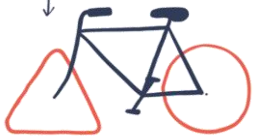
2022

Towards open-source Wakefield solver:

Coupling post-processing **wakis with open-source PIC-FDTD solver WarpX**[6]

Successfully benchmarked lossless pillbox cavity below cutoff[7]

Don't reinvent the wheel !



2023

Exploring FIT mock-up in Python

Implementing the **Finite Integration Technique (FIT) fully in Python** to explore computational viability and benefit from the open-source community

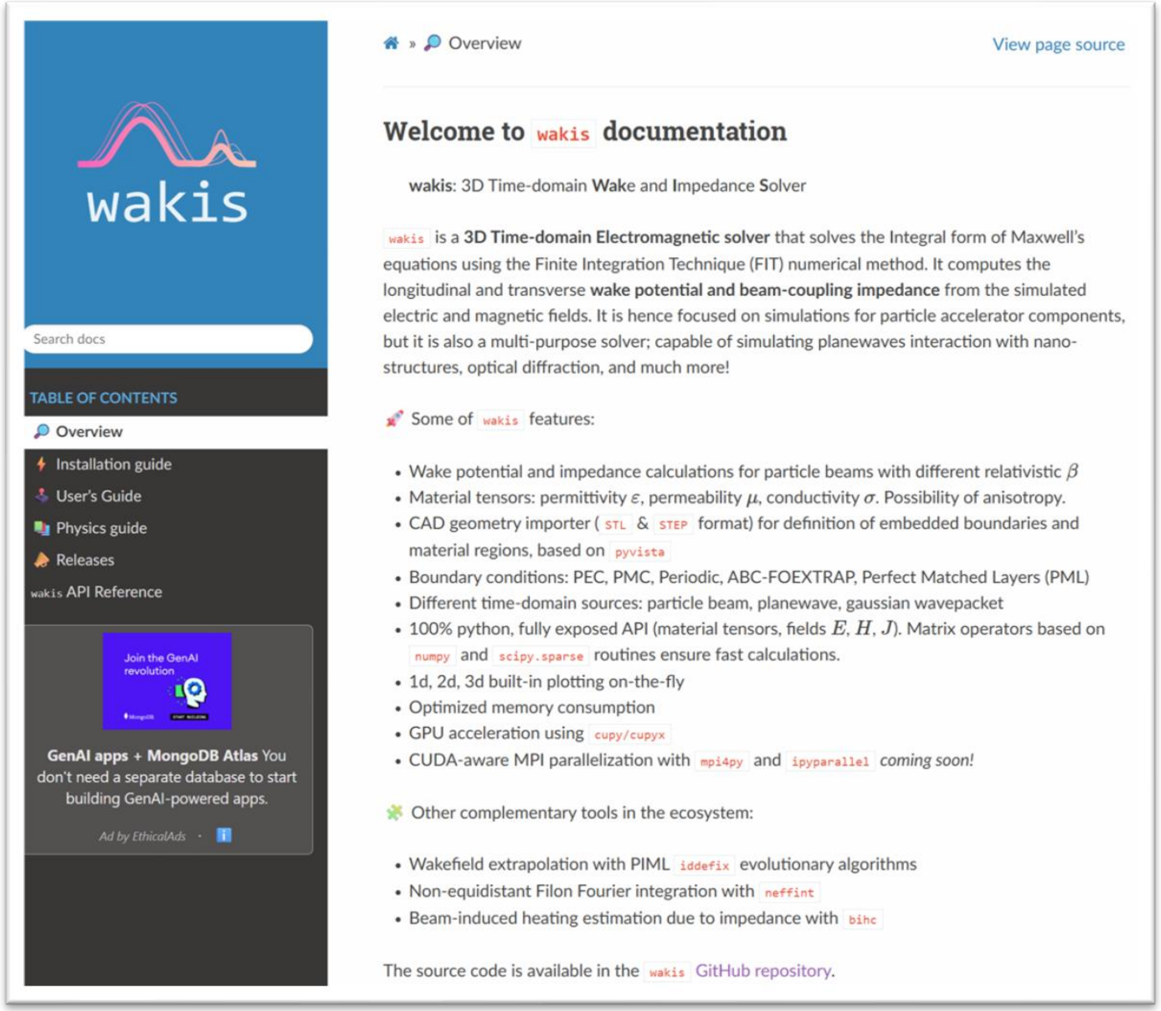
Present:

wakis as an open-source **3D time-domain electromagnetic wake and impedance solver**

[5] ABP-GIM: https://indico.cern.ch/event/1154158/
[6] ECP-WarpX: https://ecp-warpx.github.io/
[7] E. de la Fuente, WEPL170 IPAC'23

Wakis Documentation

CHECK THIS out



🏠 » 🔎 Overview   View page source

wakis

Search docs

TABLE OF CONTENTS

- 🔎 Overview
- Installation guide
- User's Guide
- Physics guide
- Releases
- wakis API Reference

Welcome to `wakis` documentation

wakis: 3D Time-domain **Wake** and **I**mpedance **S**olver

`wakis` is a **3D Time-domain Electromagnetic solver** that solves the Integral form of Maxwell's equations using the Finite Integration Technique (FIT) numerical method. It computes the longitudinal and transverse **wake potential and beam-coupling impedance** from the simulated electric and magnetic fields. It is hence focused on simulations for particle accelerator components, but it is also a multi-purpose solver; capable of simulating planewaves interaction with nano-structures, optical diffraction, and much more!

Some of `wakis` features:

- Wake potential and impedance calculations for particle beams with different relativistic β
- Material tensors: permittivity ε, permeability μ, conductivity σ. Possibility of anisotropy.
- CAD geometry importer (`STL` & `STEP` format) for definition of embedded boundaries and material regions, based on `pyvista`
- Boundary conditions: PEC, PMC, Periodic, ABC-FOEXTRAP, Perfect Matched Layers (PML)
- Different time-domain sources: particle beam, planewave, gaussian wavepacket
- 100% python, fully exposed API (material tensors, fields E, H, J). Matrix operators based on `numpy` and `scipy.sparse` routines ensure fast calculations.
- 1d, 2d, 3d built-in plotting on-the-fly
- Optimized memory consumption
- GPU acceleration using `cupy/cupyx`
- CUDA-aware MPI parallelization with `mpi4py` and `ipyparallel` *coming soon!*

Other complementary tools in the ecosystem:

- Wakefield extrapolation with PIML `iddefix` evolutionary algorithms
- Non-equidistant Filon Fourier integration with `neffint`
- Beam-induced heating estimation due to impedance with `bihc`

The source code is available in the `wakis` GitHub repository.

wakis

Search docs

TABLE OF CONTENTS

- Overview
- Installation guide
- User's Guide
- Physics guide
- Releases
- wakis API Reference

Physics guide

This section provides the theoretical foundations behind Wakis, offering a clear and concise explanation of the physics principles and numerical methods that power the simulation engine.

We begin by revisiting Maxwell's equations in their integral form and explaining how they are discretized using the Finite Integration Technique (FIT) on a structured Cartesian grid. The formulation naturally leads to Maxwell Grid Equations (MGEs), which are solved in time using a leapfrog scheme. The fields and material properties are represented on a Yee-type staggered lattice, and anisotropic or spatially varying materials are handled via sparse metric tensors.

Boundary conditions (PEC, PMC, periodic, PML) are discussed along with the treatment of sources and initial conditions. We also highlight how the solver supports geometry import and sub-pixel smoothing for embedded CAD models.

Finally, we cover the implementation aspects, including GPU acceleration with CuPy and parallelization with mpi4py, enabling high-resolution 3D simulations across multiple devices.

Contents

New!

Powered by Sphinx, hosted on ReadTheDocs

Finite Integration Technique (III)

$$h^{n+1} = h^{n} - \Delta t\, \tilde{D}_s M_\mu^{-1} D_A^{-1} C e^{n+0.5}$$

$$e^{n+1.5} = e^{n+0.5} + \Delta t D_s \tilde{M}_\varepsilon^{-1} \tilde{D}_A^{-1} \tilde{C} h^{n} - \tilde{M}_\varepsilon^{-1} j_{src}^{n} - \tilde{M}_\varepsilon^{-1}\, \tilde{M}_\sigma\, e^{n+0.5}$$

scipy.sparse
Operators
$C, D_A, D_s, \tilde{D}_A, \tilde{D}_s$

NumPy
Fields
$E(t), H(t), J(t)$

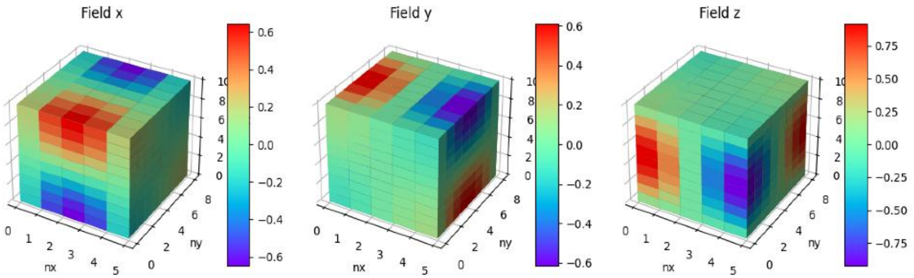
scipy.sparse
Materials
$M_\varepsilon^{-1}, M_\mu^{-1}, M_\sigma$

- $C, \tilde{C}$ **Curl operator:** [3N$_{cell}$ x 3N$_{cell}$] sparse matrix that relates the E and H fields and defines PEC, PMC or Periodic boundary conditions (BCs)
- $D_A, D_s, \tilde{D}_A, \tilde{D}_s$: 3N$_{cell}$ diagonal matrices containing the grid (primal and dual) discretization in terms of cell lengths and areas

- $e^{n+1.5}, h^{n+1}, j^{n}$ **Fields:** saved in memory as [3N$_{cell}$] 1d lexicographic arrays, updated every simulation timestep.

Fully-exposed, modifications on-the-fly (e.g. addition of **sources or initial conditions**)

wakis `E.inspect3d()` method

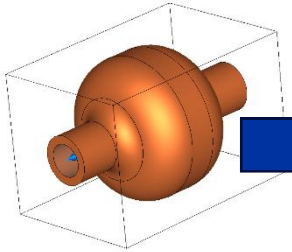


- $\tilde{M}_\varepsilon^{-1}, M_\mu^{-1}$: 3N$_{cell}$ diagonal matrices for the anisotropic inverse permittivity ε and permeability μ rank-2 tensors
- $\tilde{M}_\sigma$: 3N$_{cell}$ diagonal matrix for the anisotropic electric conductivity rank-2 tensor. The current can be computed as: $j^{n+1} = \tilde{M}_\sigma\, e^{n+0.5} + j_{src}^{n}$

CERN POLITÉCNICA IFN-GV

Wakis+IDDEFIX simulation workflow

Acc. Device under study



wakis

IDDEFIX

Partially decayed wakefield simulation (e.g. -30dB)

Resonator fitting with IDDEFIX

Table of resonator parameters
$\{R_{s,n}, Q_n, f_{r,n}\}$

Fully decayed, analytically represented impedance & wake

Verification with longer simulation (-60dB) if needed

Acc. Device's beam-induced power loss

Acc. Impedance model

Xsuite pyWIT

Beam dynamic simulations

Xsuite xwakes

Deployment pipeline

Code development is tracked with GitHub releases

On each release (CI/CD pipeline)**:**

a. All the tests are run with GitHub Actions

b. Main branch is tagged with a version e.g. #v0.5.1

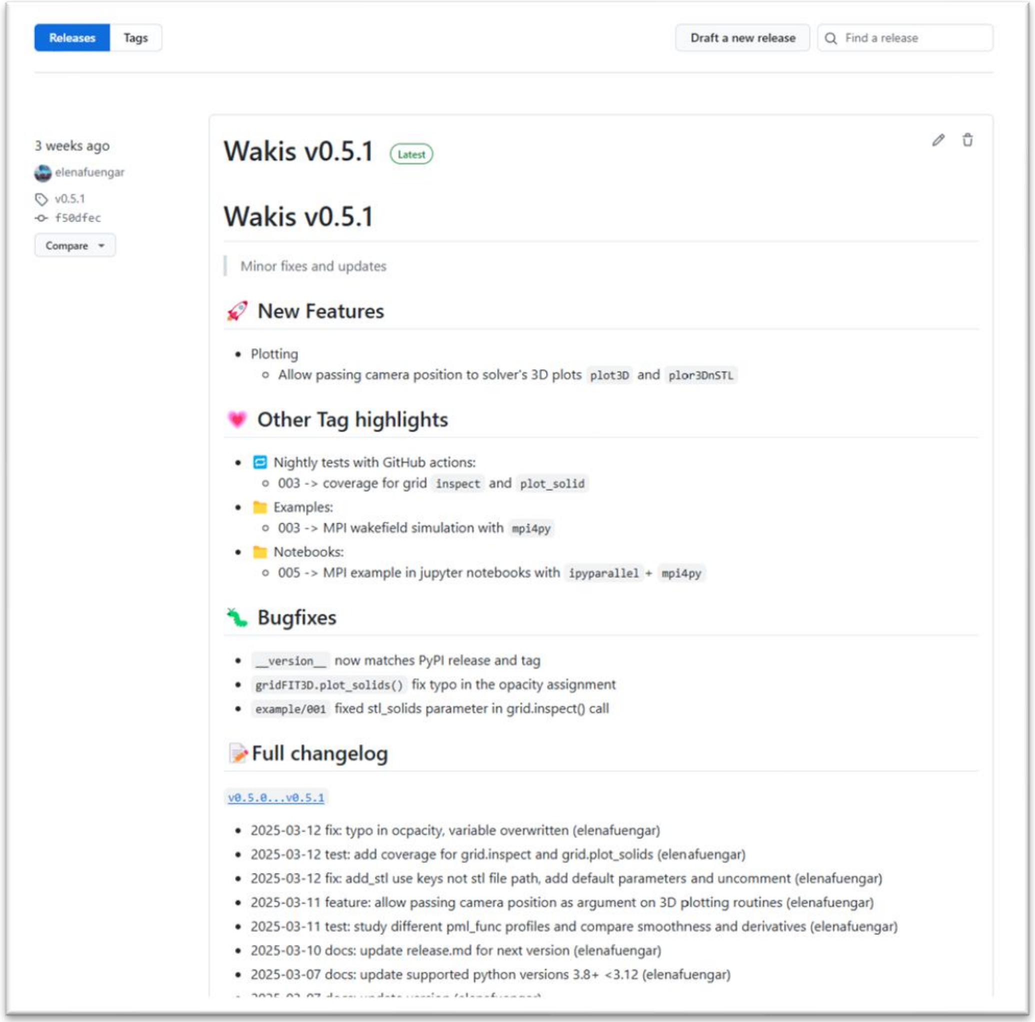
c. Create a summary of features, bugfixes and contributions and changelog

d. Deploys a stable version to PyPI `> pip install wakis`

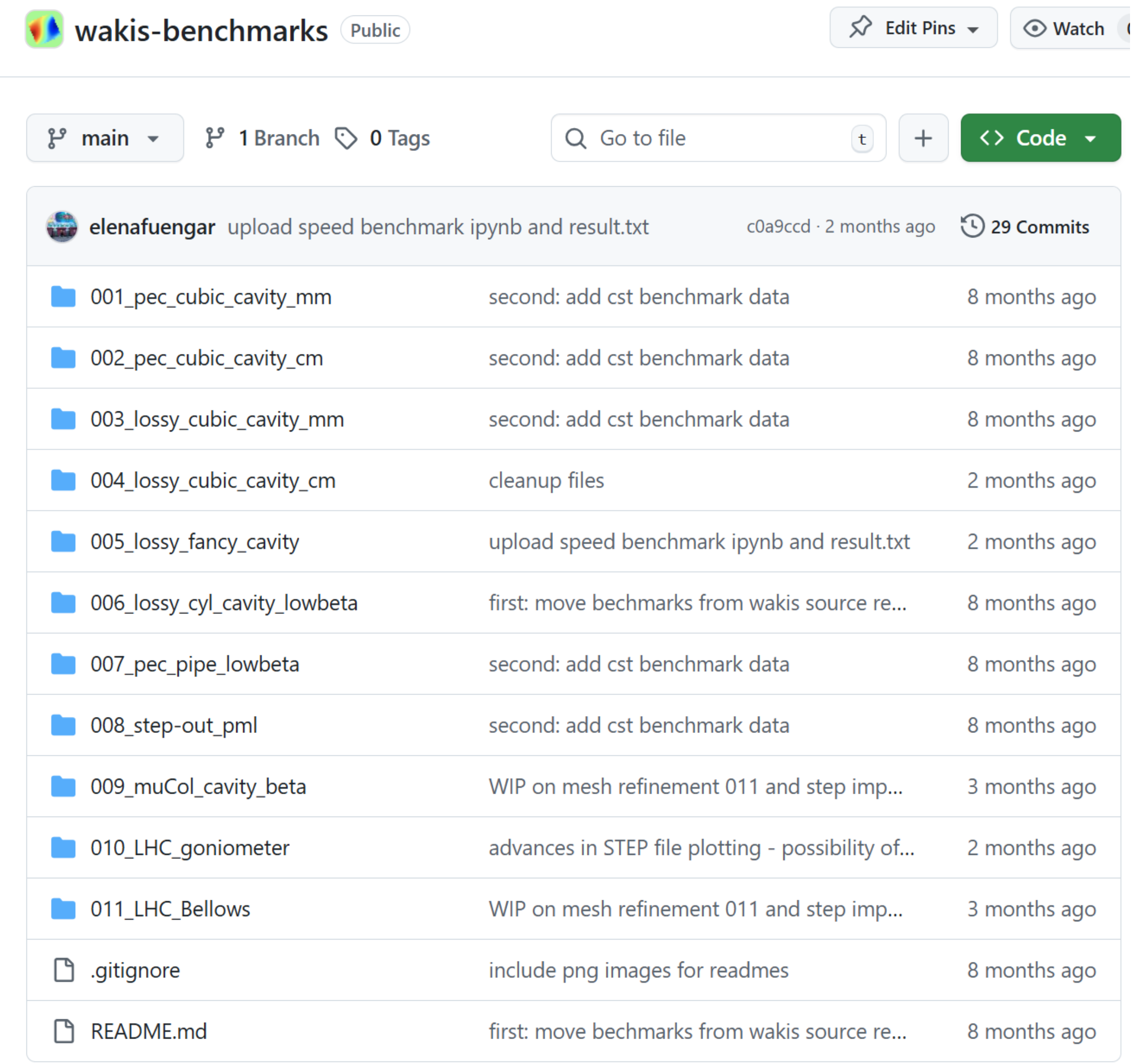
e. Updates documentation on ReadTheDocs

f. Generates a Zenodo DOI that hosts a snapshot of the code at that stage in a .zip file

The work in progress is tracked in a release draft `release.md`



Wakis benchmarks



wakis-benchmarks Public

Edit Pins | Watch

main | 1 Branch | 0 Tags | Go to file | t | + | Code

elenafuengar upload speed benchmark ipynb and result.txt c0a9ccd · 2 months ago 29 Commits

001_pec_cubic_cavity_mm	second: add cst benchmark data	8 months ago
002_pec_cubic_cavity_cm	second: add cst benchmark data	8 months ago
003_lossy_cubic_cavity_mm	second: add cst benchmark data	8 months ago
004_lossy_cubic_cavity_cm	cleanup files	2 months ago
005_lossy_fancy_cavity	upload speed benchmark ipynb and result.txt	2 months ago
006_lossy_cyl_cavity_lowbeta	first: move bechmarks from wakis source re...	8 months ago
007_pec_pipe_lowbeta	second: add cst benchmark data	8 months ago
008_step-out_pml	second: add cst benchmark data	8 months ago
009_muCol_cavity_beta	WIP on mesh refinement 011 and step imp...	3 months ago
010_LHC_goniometer	advances in STEP file plotting - possibility of...	2 months ago
011_LHC_Bellows	WIP on mesh refinement 011 and step imp...	3 months ago
.gitignore	include png images for readmes	8 months ago
README.md	first: move bechmarks from wakis source re...	8 months ago