



## Xsuite evolution for FCC studies

G. Iadarola, S. Łopaciuk, R. De Maria, F. Van. Der Veken

Big thanks to the Xsuite users and contributors  
for their enthusiasm and invaluable input.

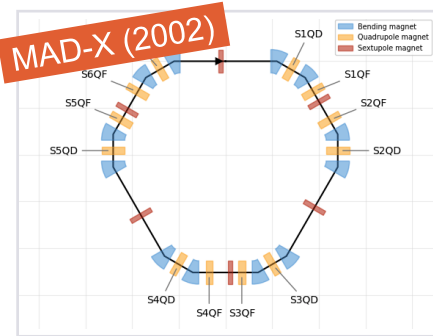
Work supported by:



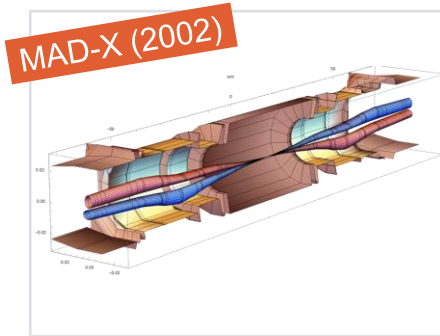


- **Introduction**
- **Xsuite capabilities in a nutshell**
- **Recent developments for FCC-ee**
  - Lattice and optics design
  - Global optimization for sextupole configuration
  - Spin tracking and polarization

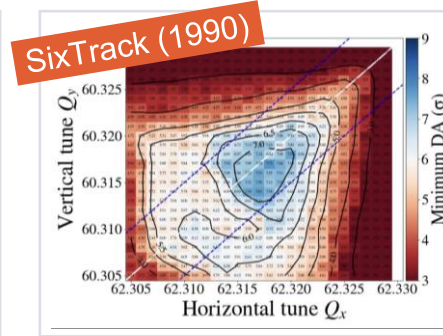
Until a few years ago beam dynamics studies at CERN were relying in a plethora of **legacy tools**:



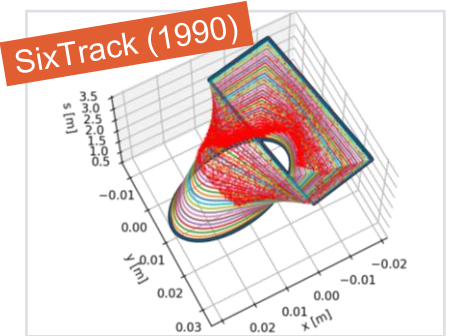
Lattice Design



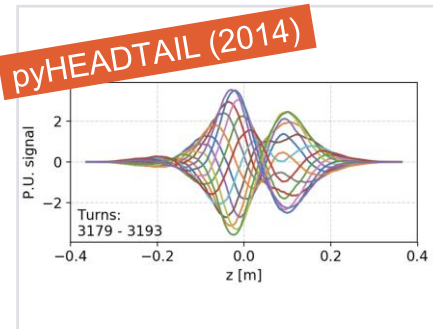
Optics (calculation & design)



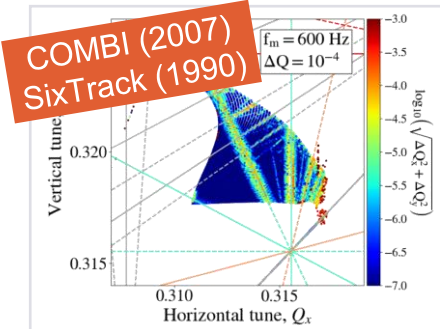
Tracking (dynamic aperture)



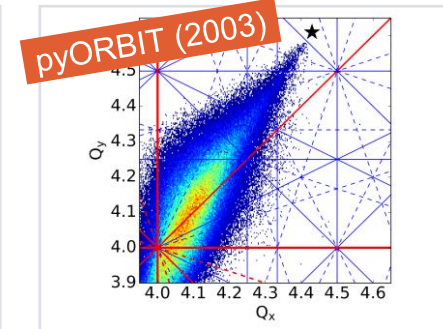
Collimation



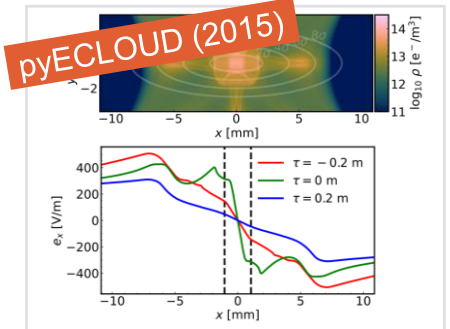
Impedances



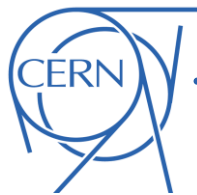
Beam-Beam Effects



Space Charge



Electron/Ion Cloud



Old tools had several limitations, which **became apparent when approaching FCC-ee**

- **Extendibility**

- Difficult and expensive to extend old tools for modeling additional effects
- Many experts have left or retired; working on old Fortran does not match skill set and interest of younger colleagues



- **Integration issues**

- Often addressing interplay of optics, non-linear tracking, collective effects, collimation, etc.
- With legacy tools this means dealing with model translations, ad-hoc interfaces, which is complicated, time-consuming, error prone.

- **Outdated user interfaces – today users want Python**

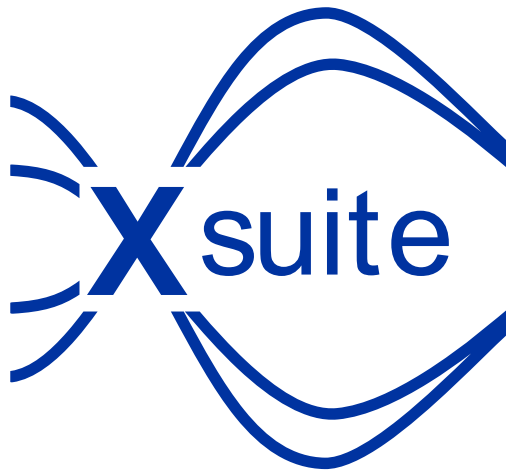
- Python is a standard in scientific computing, especially among younger researchers.
- Many high-quality tools are available in the Python ecosystem also due to ML boom.
- Maintaining custom tools for plotting, scripting, etc. wastes effort, increases complexity, and lowers quality.



- **Performance**

- GPU acceleration is a must-have in many applications, but often infeasible to retrofit.





Xsuite project **launched in 2021** to address these issues using the know-how acquired in the development of the earlier tools.

**Goals: 1st class Python, 1st class collective, 1st class GPU.**

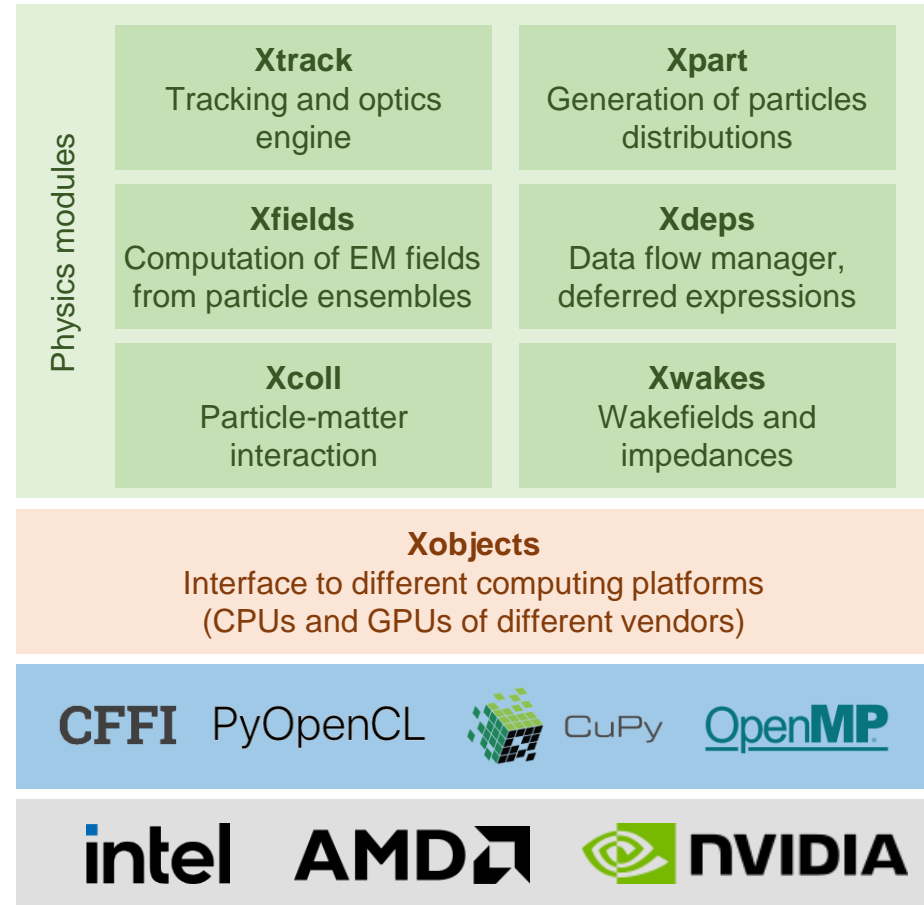
- One toolkit for many applications: from low-energy hadron rings to high-energy lepton colliders.
- Simulations involving heterogeneous effects made natural.
- Support different computing platforms – single- and multi-threaded CPU, GPU acceleration.

**References:**

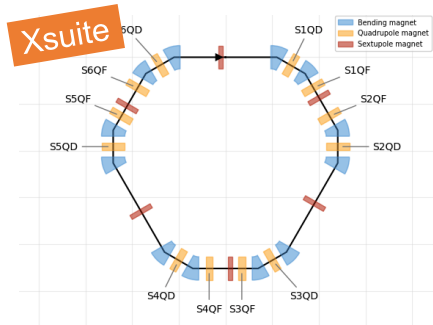
- Documentation: [xsuite.web.cern.ch](https://xsuite.web.cern.ch).
- [“Xsuite: An Integrated Beam Physics Simulation Framework,” JACoW HB2023 \(2024\), TUA211.](#)
- [“Xsuite: a multiplatform toolbox for optics design, fast tracking, collimation and collective effects,” ICAP’24](#)
- Seminar: <https://indico.cern.ch/event/1489714/>



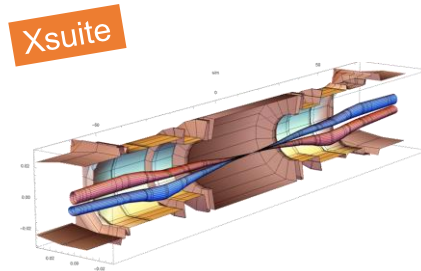
- **Orthogonal architecture: split the software into independent functional blocks at all levels.**
  - **Well structured codebase** scales well and makes it easier to add new features.
  - **Short learning curve** for users & developers → users can (and do!) become developers.
- **Practicing agile development:**
  - From the beginning big effort to **support users** ⇒ **users' feedback** visibly increases the quality of the tool
  - Strong investment in **automatic testing**, we can afford a **fast release cycle** with new versions coming out multiple times a month



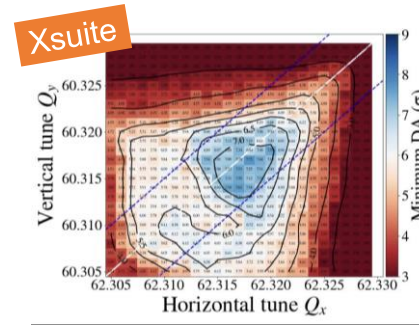
By now Xsuite became a **production tool** for many beam dynamics studies, allowing development discontinuation for many legacy tools.



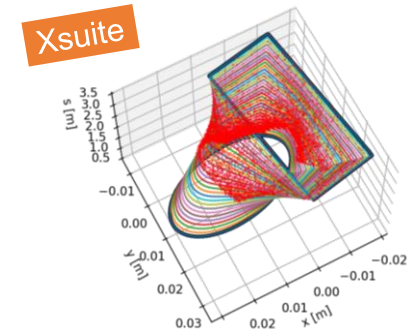
Lattice Design



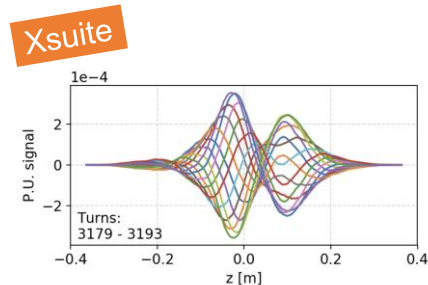
Optics (calculation & design)



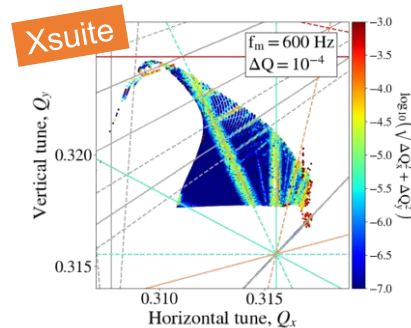
Tracking (dynamic aperture)



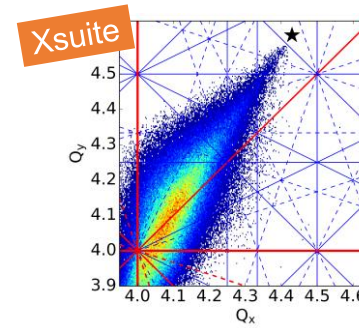
Collimation



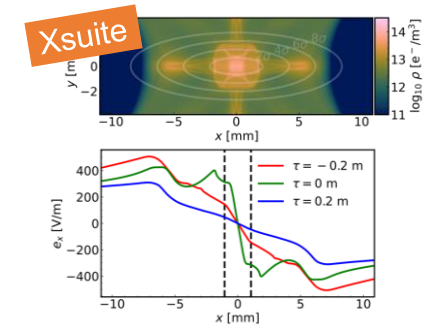
Impedances



Beam-Beam Effects



Space Charge



Electron/Ion Cloud



Users' and developers' response was well beyond our expectations!

**>30 contributors**

**>150 active users world-wide**

(60: ABP, 30: other CERN groups, 60: other labs)

**>200 Xsuite mentions in IPAC24 proceedings**

**Accelerator schools (USPAS, CAS) used Xsuite for tutorials in 2024.**

**Xsuite used for a large number of facilities**

## **CERN**

- AD
- ELENA
- LEIR
- PSB
- PS
- SPS, TI2, TI8
- LHC
- FCC-ee, FCC-hh
- Muon Collider

## **Fermilab**

- Main injector
- Recycler
- Booster
- IOTA

## **GSI**

- SIS-18
- SIS-100

## **BNL**

- RHIC
- Booster
- EIC

## **J-PARC**

- Main Ring
- KEK
- SuperKEKB

## **Medical facilities**

- HIT (Heidelberg)
- MEDAUSTRON
- PIMMS
- NIMMS

## **Light sources/damping rings:**

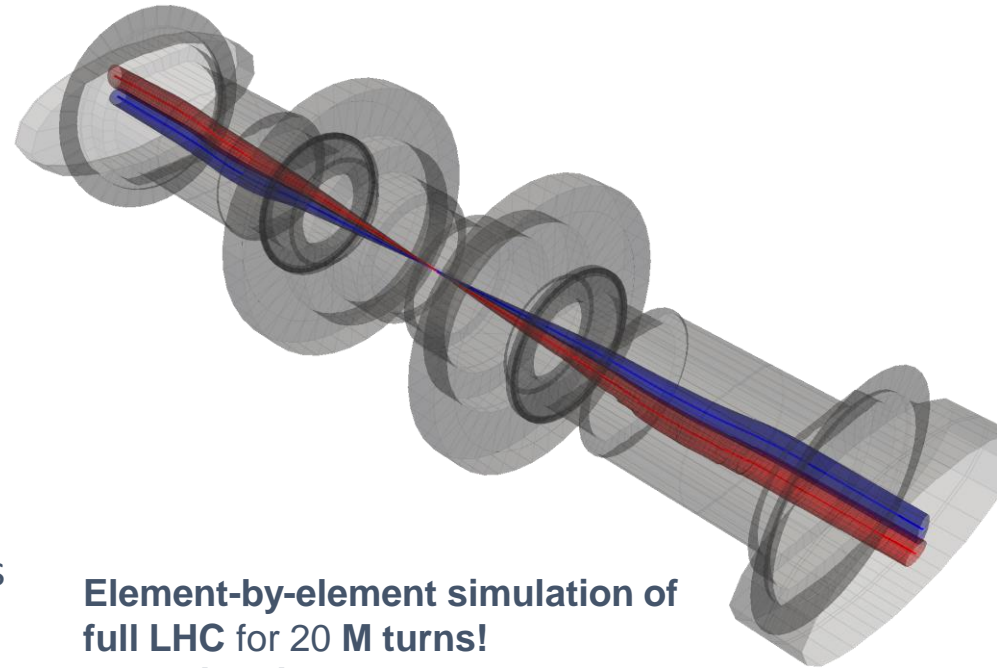
- PETRA
- DESY injector ring
- ELETTRA
- BESSY III
- PSI SLS 2.0
- Canadian Light Source
- CLIC-DR

**and more...**

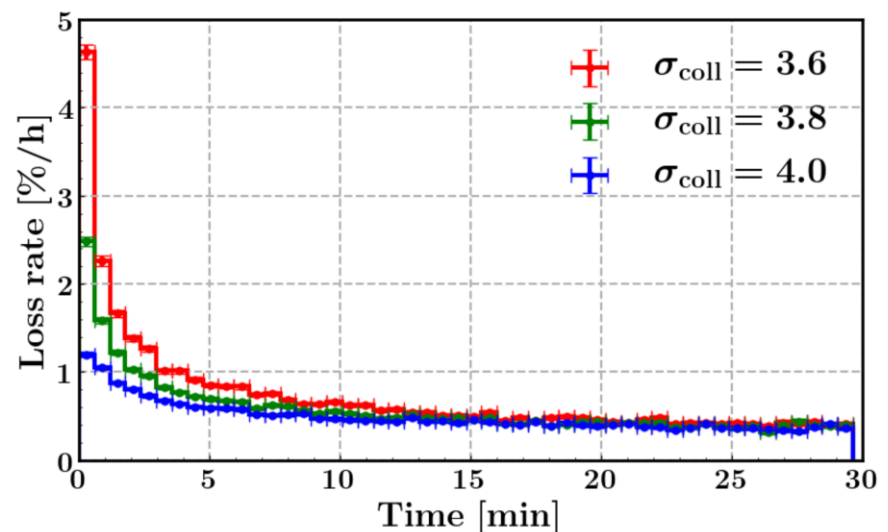


- Introduction
- **Xsuite capabilities in a nutshell**
- Recent developments for FCC-ee
  - Lattice and optics design
  - Global optimization for sextupole configuration
  - Spin tracking and polarization

- **Lattice design**
- Fast and accurate calculation of **optics and chromatic properties**
- **Optics design**
- **Field imperfections and misalignments**
- **Fringe fields, combined function-magnets, solenoids** with overlapping multipolar fields
- **Orbit and trajectory correction**
- Computation of **non-linear maps, normal forms and RDTs**
  - Through interface with MAD-NG
- Tracking through **fieldmaps**
  - Through interface with RF-Track (see also talk by A. Latina)
- **Fast tracking** on CPU and GPU

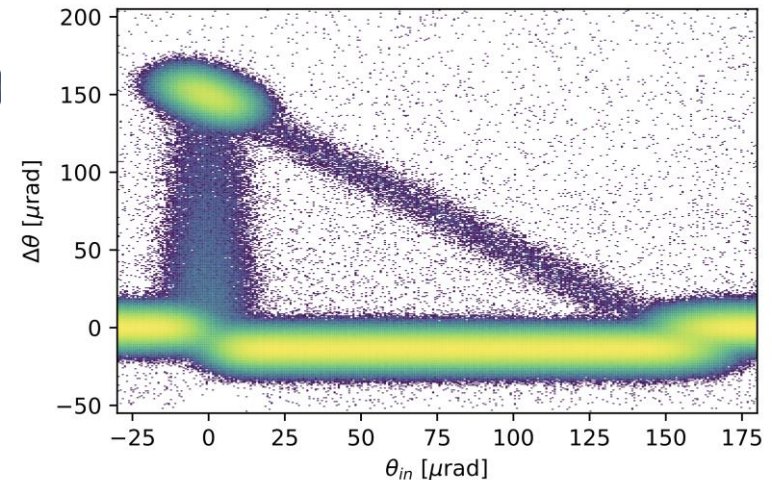


Element-by-element simulation of full LHC for 20 M turns!  
 < 3 d sim. time on GPUs

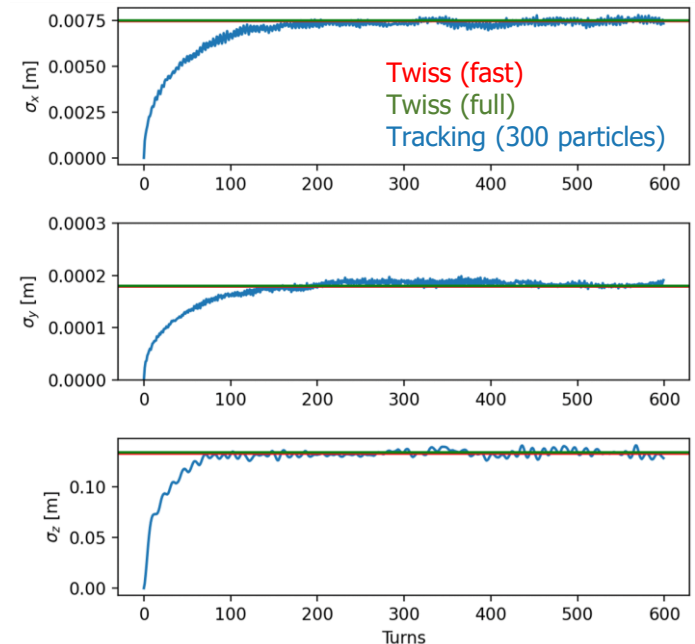



- **Particle-matter interaction** (e.g. for collimation)
  - Internal scattering engine (Everest)
  - Interfaces to Geant 4 and FLUKA
- **Synchrotron Radiation**
  - Mean and stochastic models for emission in all magnetic elements
  - Computation of damping constants and equilibrium emittances
  - Energy loss compensation (tapering)
  - Computation of radiation integrals
- **Spin and polarization**
  - Spin tracking
  - Computation of polarization buildup rate and equilibrium polarization level

Particle deflection from a bent crystal  
(Everest engine)

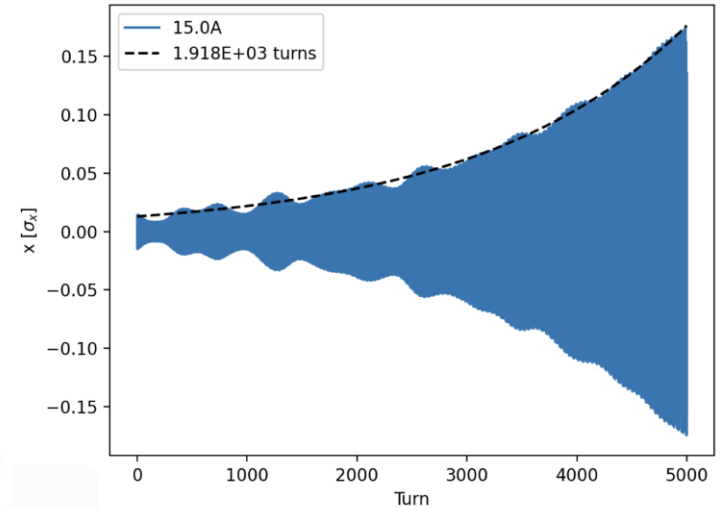


Eq. emittance from SR (twiss vs track)

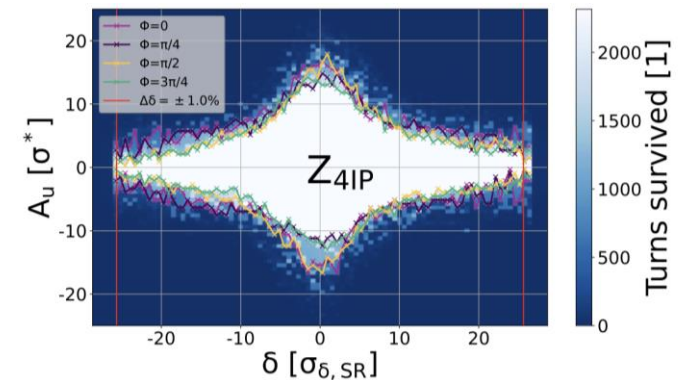


- **Space-charge**
  - Frozen and self-consistent PIC
- **Wakefields**
  - Dipolar and quadrupolar wakefields
  - Single-bunch and multi-bunch
  - Multi-turn effects
- **Beam-beam effects** (see talk by P. Kicsiny) 
  - Weak-strong and strong-strong modelling
  - Analytic distribution or Particle In Cell (new!)
  - Beamstahlung and Bhabha scattering
- **Intra-Beam Scattering (IBS)**
  - Computation of growth rates from twiss table
  - IBS kicks in tracking simulations

**Instability driven by wakfields**



**FCC-ee DA with BB**





- Introduction
- Xsuite capabilities in a nutshell
- **Recent developments for FCC-ee**
  - Lattice and optics design
  - Global optimization for sextupole configuration
  - Spin tracking and polarization

# Xsuite developments for lattice and optics design



- Over the last years we invested significant effort to improve Xsuite performance and interface in **describing accelerator lattices and design optics**
  - First full LHC cycle made with Xsuite tested in MD in 2024
- To probe and improve the tools capability also in the context of “**green-field**” **accelerator design** we tried to **reproduce reasonably from “scratch” the lattice building and matching of the FCC-ee LCC lattice**



The outcome of the exercise is a **tutorial in the form of Jupyter notebooks** illustrating

- How the lattice is “assembled”
- How the linear optics is matched (matching quad strengths)
- How chromatic and detuning properties are controlled (matching of sextupoles strengths)

Built with the aim of providing a **resource to colleagues learning the tool** and to **collect feedback on how to further improve**

Available [here](#)



## What you find in the tutorial

### Agile lattice definition and inspection

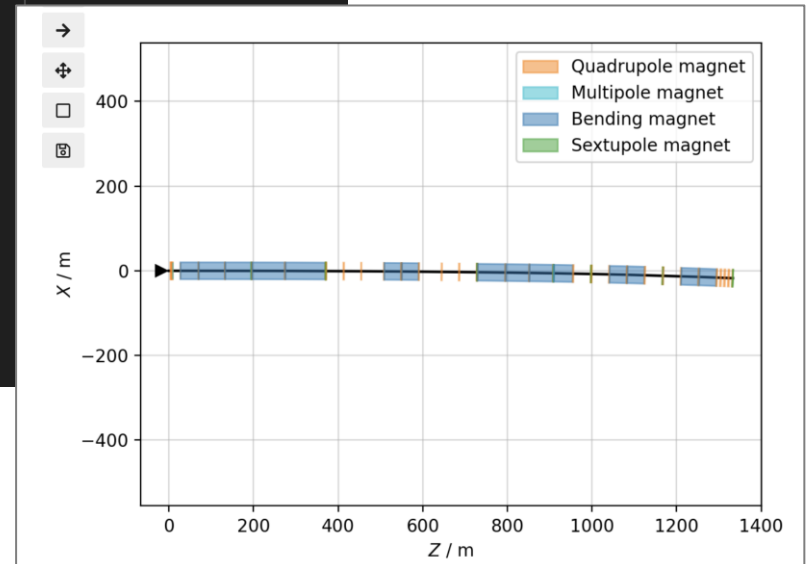
```
env['arc_octant'] = (12 * env['cell_u']
                    + env['mark_mid_cell_l']
                    + env['cell_u']
                    + env['mark_mid_cell_r']
                    + 12 * env['cell_u'])

env['ffl'] = env['ccs_xl'] + env['mark_e_ccsyl'] + env['ccs_yl']
env['ffr'] = env['ccs_xr'] + env['mark_e_ccsyr'] + env['ccs_yr']
env['experimental_insertion_l'] = (env['cell_l3'] + env['cell_uffl'] + env['mark_e_ffl'] + env['ffl'])
env['experimental_insertion_r'] = -(env['cell_r3'] + env['cell_uffr'] + env['mark_e_ffr'] + env['ffr'])

env['service_insertion_l'] = -(env['cell_ul'] + env['cell_us'] + env['straight_l'])
env['service_insertion_r'] = (env['cell_ur'] + env['cell_su'] + env['straight_r'])

env['fcc_quarter'] = (
    env['mark_ip_mid']
    + env['experimental_insertion_r']
    + env['arc_octant']
    + env['mark_ser_ins_r']
    + env['service_insertion_r']
    + env['mark_ser_ins_mid']
    + env['service_insertion_l']
    + env['mark_ser_ins_l']
    + env['arc_octant']
    + env['experimental_insertion_l'])

env['fccee_p_ring'] = 4 * env['fcc_quarter']
```





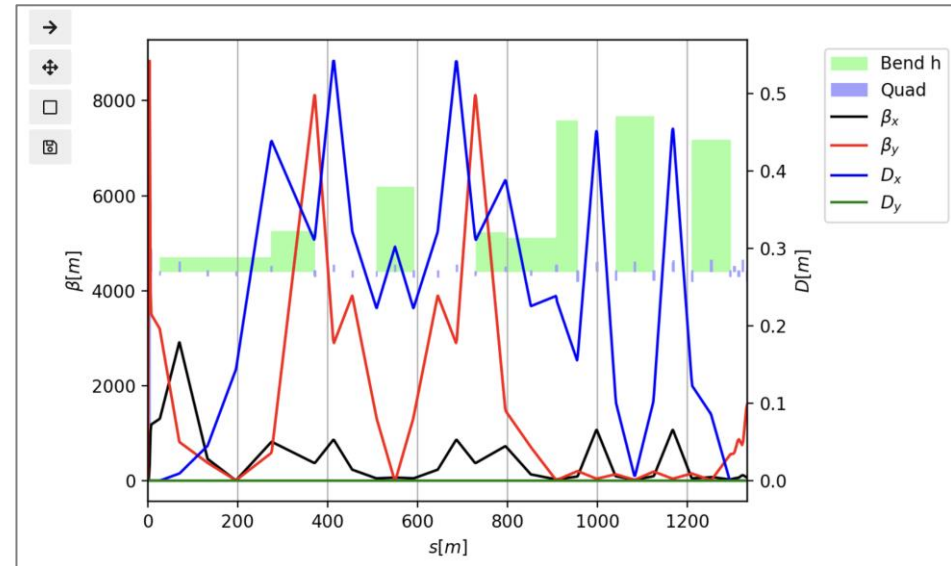
# Xsuite developments for lattice and optics design

## What you find in the tutorial

How to handle complex non linear optimization problems with up to 40 targets, including Twiss parameters, dispersions, phases, R-matrix



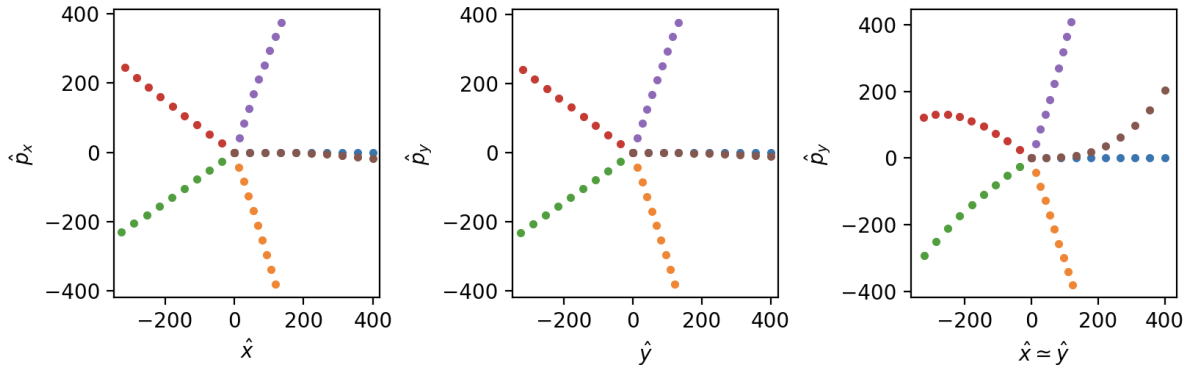
Target status:		enalty = 2.1749e-02					
id	state	tag	tol_met	residue	current_val	target_val	
0	ON	END_betx	True	2.46075e-10	66.667	66.667	
1	ON	END_bety	True	1.48475e-10	1600	1600.0	
2	ON	END_alfx	True	-4.32161e-11	-4.32161e-11	0.0	
3	ON	END_alfy	True	3.30344e-12	3.30344e-12	0.0	
4	ON	END_mux	True	3.06422e-14	3	3.0	
5	ON	END_muy	True	-8.88178e-15	2.75	2.75	
6	ON	END_dx	True	-1.80012e-10	-1.80012e-10	0.0	
7	ON	END_dpx	True	-3.40465e-12	-3.40465e-12	0.0	
8	ON	qf17l_dx	True	-5.26996e-11	-5.26996e-11	0	
9	ON	qf17l_dpx	True	-7.20583e-12	-7.20583e-12	0	
10	ON	qd18l_dx	True	-1.0819e-10	-1.0819e-10	0	
11	ON	qd18l_dpx	True	-5.7032e-12	-5.7032e-12	0	
12	ON	qf17l_bety	True	0	567.739	LessThan(2000)	
13	ON	qd18l_bety	True	0	844.968	LessThan(2000)	
14	ON	qf19l_bety	True	0	750	LessThan(2000)	
15	ON	qd20l_bety	True	0	1508.57	LessThan(2000)	
16	ON	qf19l_bety	True	-2.09184e-11	750	GreaterThan(750)	
17	ON	sfm2l::0_betx	True	0	25.4129	LessThan(27)	
18	ON	sfm2l::0_bety	True	0	13.6561	LessThan(15)	
19	ON	sfm2l::0_alfx	True	0	0.363027	GreaterThan(0)	
20	ON	sfm2l::0_alfy	True	0	0.82947	GreaterThan(0.5)	
21	ON	sfm2l::0_dx	True	-6.08149e-08	0.235	GreaterThan(0.235)	
22	ON	sfm2l::0_alfy	True	0	0.82947	LessThan(0.9)	
23	ON	sfm2l::0_dx	True	0	0.235	LessThan(0.25)	
24	ON	sfx1l::1_betx	True	0	1064.54	GreaterThan(1000)	
25	ON	sfx1l::1_bety	True	0	40.8916	LessThan(45)	
26	ON	sfx1l::1_alfx	True	6.16879e-06	6.16879e-06	0	
27	ON	sfx1l::1_mux	True	6.86118e-14	1.7407	1.7407	
28	ON	sfx1l::1_dx	True	0	0.451062	LessThan(0.6)	
29	ON	ipimag4_betx	True	0	19.1385	LessThan(20)	
30	ON	ipimag4_bety	True	0	16.2653	LessThan(25)	
31	ON	ipimag4_alfx	True	6.15045e-12	6.15045e-12	0	
32	ON	ipimag4_alfy	True	4.80171e-15	-0.2	-0.2	
33	ON	ipimag4_dx	True	0	0.0063794	GreaterThan(0.002)	
34	ON	ipimag4_dpx	True	5.00376e-17	1.1e-05	1.1e-05	
35	ON	r11	True	-3.41367e-08	-1	-1	
36	ON	r12	True	-1.52734e-05	-0.100015	-0.1	
37	ON	r33	True	0.000217492	-0.999783	-1	
38	ON	r34	True	-2.06912e-07	-2.06912e-07	0.0	



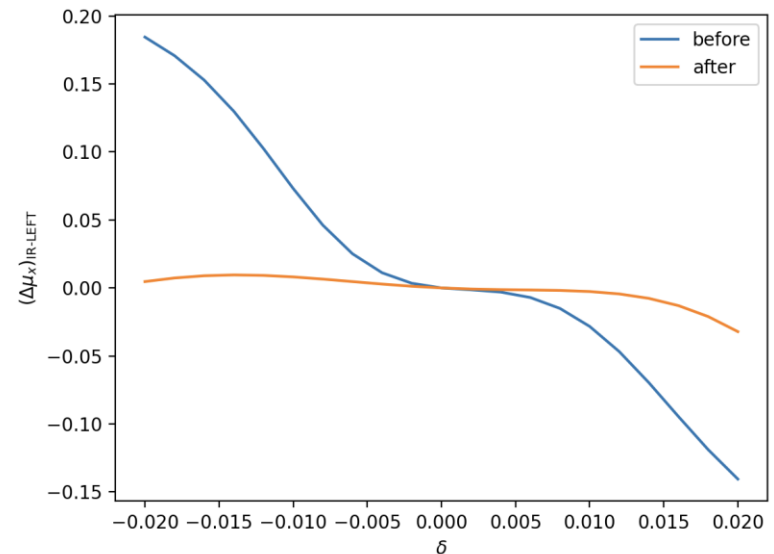
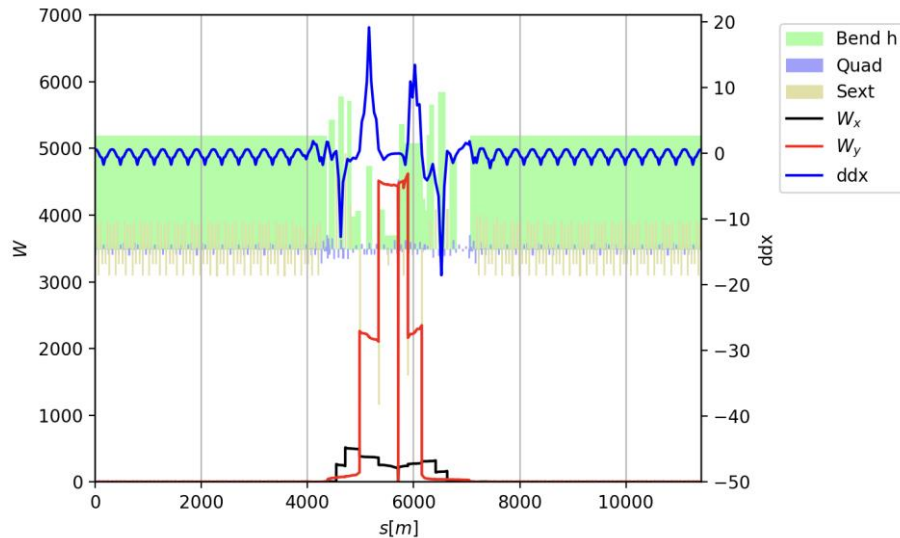


## What you find in the tutorial

### “Starfish” optimization of sextupole-induced amplitude detuning



### Matching of chromatic functions, second-order dispersion, non-linear chromaticity





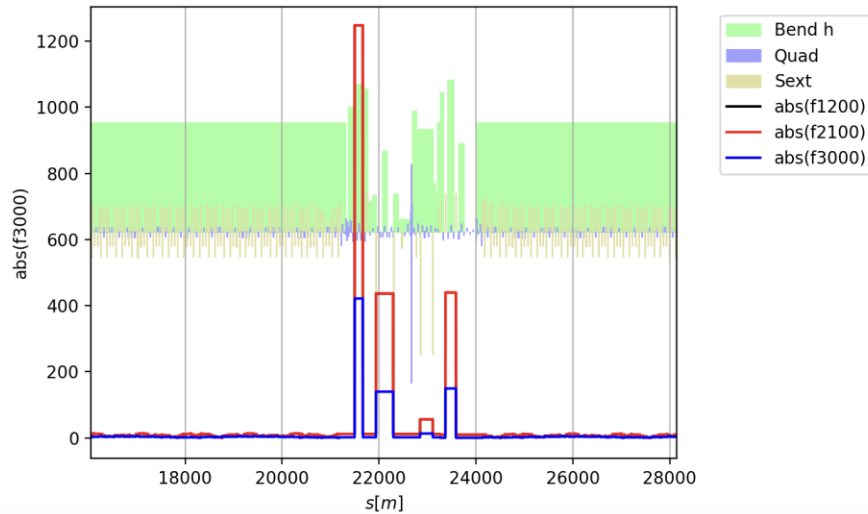
## What you find in the tutorial

### RDT computation (powered by MAD-NG)

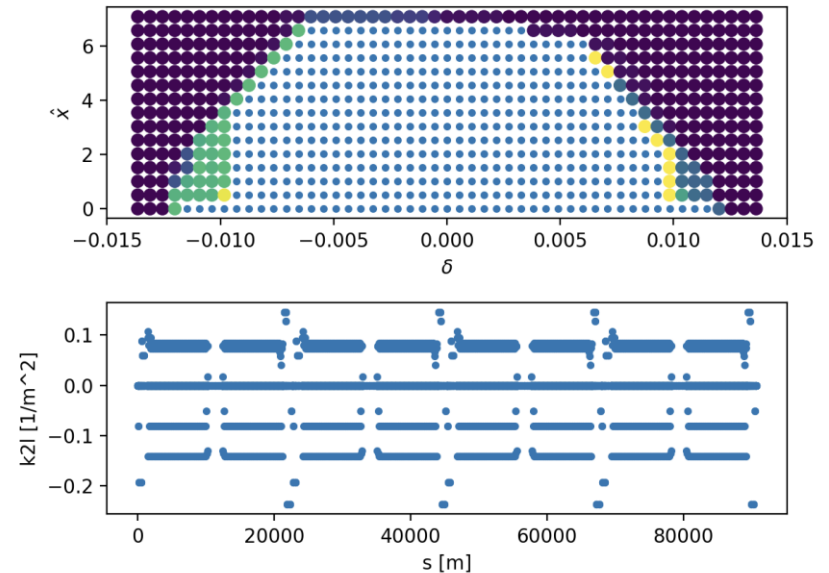
Use MAD-NG to study RDT cancellations

```
[9]: line.replace_all_repeated_elements() # Not yet supported by Xsuite MAD-NG interface
[10]: tw_ng = line.madng_twiss(rdts=['f1200', 'f2100', 'f3000'])
[11]: # Plot RDT
tw_ng.plot('abs(f1200) abs(f2100) abs(f3000)')
[11]: <xtrack.twissplot.TwissPlot object at 0x34aa6c1d0>
```

Figure 3



### Momentum acceptance computation



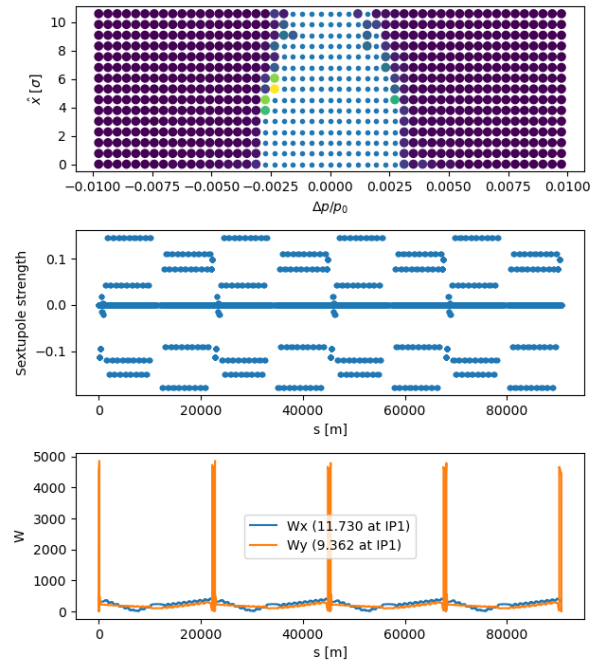


- **Introduction**
- **Xsuite capabilities in a nutshell**
- **Recent developments for FCC-ee**
  - Lattice and optics design
  - Global optimization for sextupole configuration
  - Spin tracking and polarization

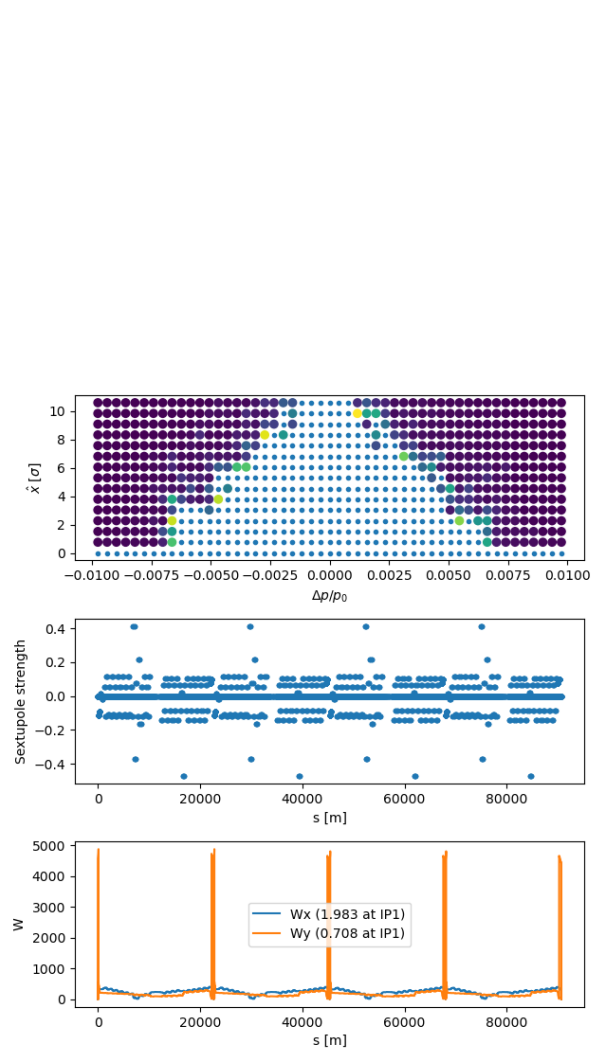


# Use of global optimization for MA improvement

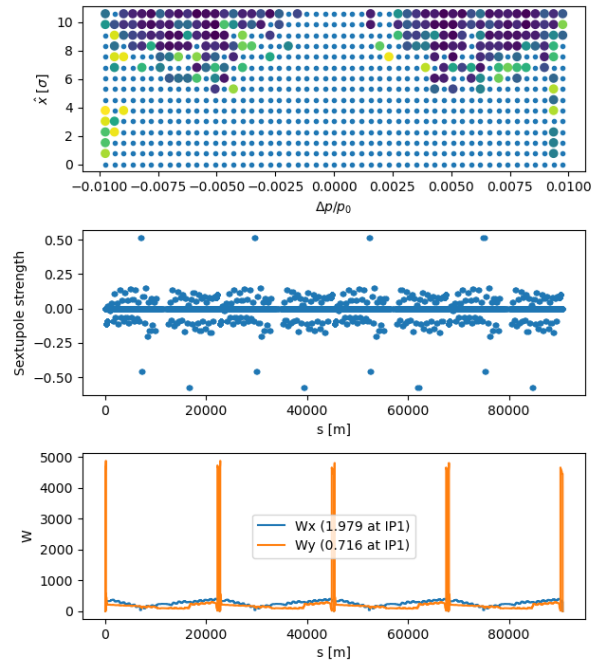
- Xsuite coupled to external Python optimization library ([pymoo](#)) for improving Momentum Acceptance on FCC-ee GHC lattice



DiRECT



Genetic





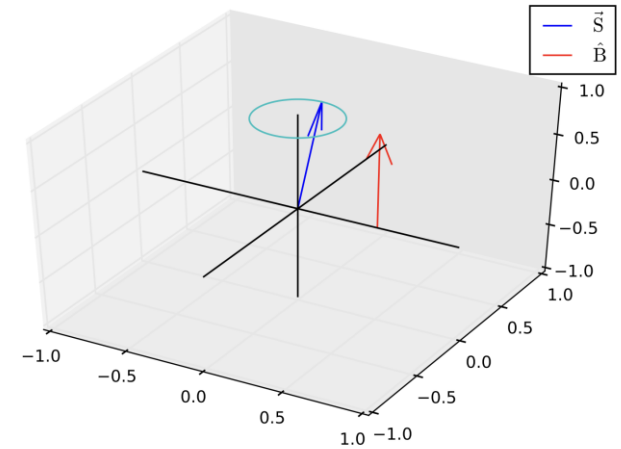
- Introduction
- Xsuite capabilities in a nutshell
- **Recent developments for FCC-ee**
  - Lattice and optics design
  - Global optimization for sextupole configuration
  - Spin tracking and polarization

- Spin tracking recently made available for all magnetic elements of Xsuite (see also presentation by K. Hock)

Spin precession equation:

$$\frac{d\vec{S}}{dt} = \frac{q}{\gamma m} \vec{S} \times \left[ (1 + G\gamma)\vec{B}_{\perp} + (1 + G)\vec{B}_{\parallel} \right]$$

- Spin tracking is by default off, **can be enabled by calling:**
  - `line.configure_spin(model="auto")`
- Spin tracking is used by the **twiss module** to compute:
  - The **Invariant Spin Field** on the closed orbit ( $n_0$ )
  - **Polarization/depolarization times** and **equilibrium polarization for lepton beams** in the linear approximation





- We add to the **coordinate space** the three cartesian component of the spin:

$$z = \begin{pmatrix} x \\ p_x \\ y \\ p_y \\ \zeta \\ \delta \\ s_x \\ s_y \\ s_z \end{pmatrix} \quad \text{We call } z_{\text{orb}} = \begin{pmatrix} x \\ p_x \\ y \\ p_y \\ \zeta \\ \delta \end{pmatrix}, \quad z_{\text{spin}} = \begin{pmatrix} s_x \\ s_y \\ s_z \end{pmatrix}$$

- Tracking particles for one turn defines a **9D non-linear one-turn map**:
  - The **closed orbit and the closed spin solution  $n_0$**  at the start of the machine are found by a numerical fixed-point searches
- Once the fixed point at one location is found the **9D orbit around the machine is obtained by tracking one particle**



The 9D one turn-matrix  $\mathbf{R}$  can be obtained from the tracking using **central differences** (as already done in 6D for computation of twiss parameters)

Since the spin motion does not influence the orbital motion the one turn matrix has the following **structure**:

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{\text{orb}} & 0 \\ \mathbf{D} & \mathbf{A} \end{pmatrix}$$

$\mathbf{R}_{\text{orb}}$  is the usual 6D one-turn matrix

$\mathbf{A}$  is the precession matrix for the spin on the closed orbit

$\mathbf{D}$  describes how the spin precession is perturbed when the particle coordinates move away from the closed orbit

Due to the structure of the matrix, its **eigenvalues and eigenvectors** have the following properties

The orbital part of the 9D eigenvector is the eigenvector of the orbital motion:

$$\mathbf{R}_{\text{orb}} \mathbf{e}_{i,\text{orb}} = \lambda_i \mathbf{e}_{i,\text{orb}}$$

$\mathbf{R} \mathbf{e}_i = \lambda_i \mathbf{e}_i \quad \rightarrow$  The spin part of the eigenvector can be calculated from the orbital part of the eigenvectors:

$$\mathbf{e}_{i,\text{spin}} = (\lambda_i \mathbf{I} - \mathbf{A})^{-1} \mathbf{D} \mathbf{e}_{i,\text{orb}}$$



The extended one-turn matrix can be used to compute the **Invariant Spin Field (ISF)** to first order:

$$\mathbf{n}(z_{\text{orb}}) = \mathbf{n}_0 + \mathbf{N} \left( z_{\text{orb}} - z_{\text{orb}}^{\text{CO}} \right)$$

One can prove that the **Jacobian of the invariant spin fields can be obtained from the eigenvectors of the one-turn matrix:**

$$\mathbf{N} = \mathbf{E}^{\text{spin}} \left( \mathbf{E}^{\text{orb}} \right)^{-1}$$



$$\mathbf{N} = \mathbf{E}^{\text{spin}} \left( \mathbf{E}^{\text{orb}} \right)^{-1}$$

Once the eigenvectors are known at one location in the machine **they can be efficiently obtained at all s position by tracking particles** having as coordinates the eigenvectors (suitably scaled to be in the linear regime).

→ It is possible (and fast) to get **N matrix at all s locations**

From these one can obtain the **polarization/depolarization times** and the **equilibrium polarization** using the following approximated expressions:

$$P_{\text{eq}} = \frac{8}{5\sqrt{3}} \frac{\alpha_-}{\alpha_+}$$

$$\alpha_+ \simeq \frac{1}{C} \oint \frac{ds}{|\rho(s)|^3} \left[ 1 - \frac{2}{9} (\mathbf{n}_0 \cdot \hat{\mathbf{i}}_v)^2 + \frac{11}{18} \left| \frac{\partial \mathbf{n}}{\partial \delta} \right|^2 \right]_s$$

$$\alpha_- \simeq \frac{1}{C} \oint \frac{ds}{|\rho(s)|^3} \left[ \hat{\mathbf{i}}_B \cdot \left( \mathbf{n}_0 - \frac{\partial \mathbf{n}}{\partial \gamma} \right) \right]_s$$

These quantities can be optionally returned as output of the Xsuite twiss.

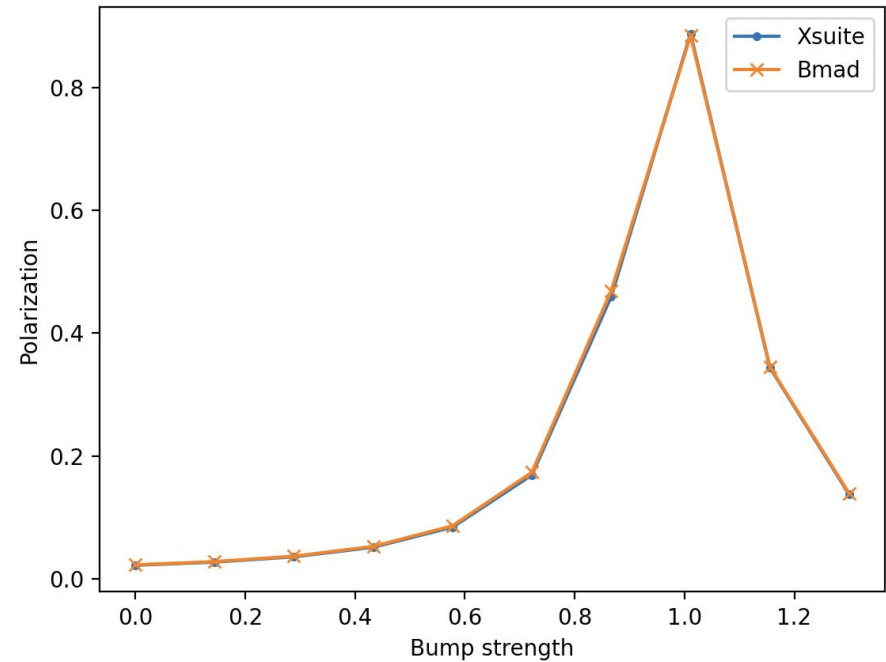
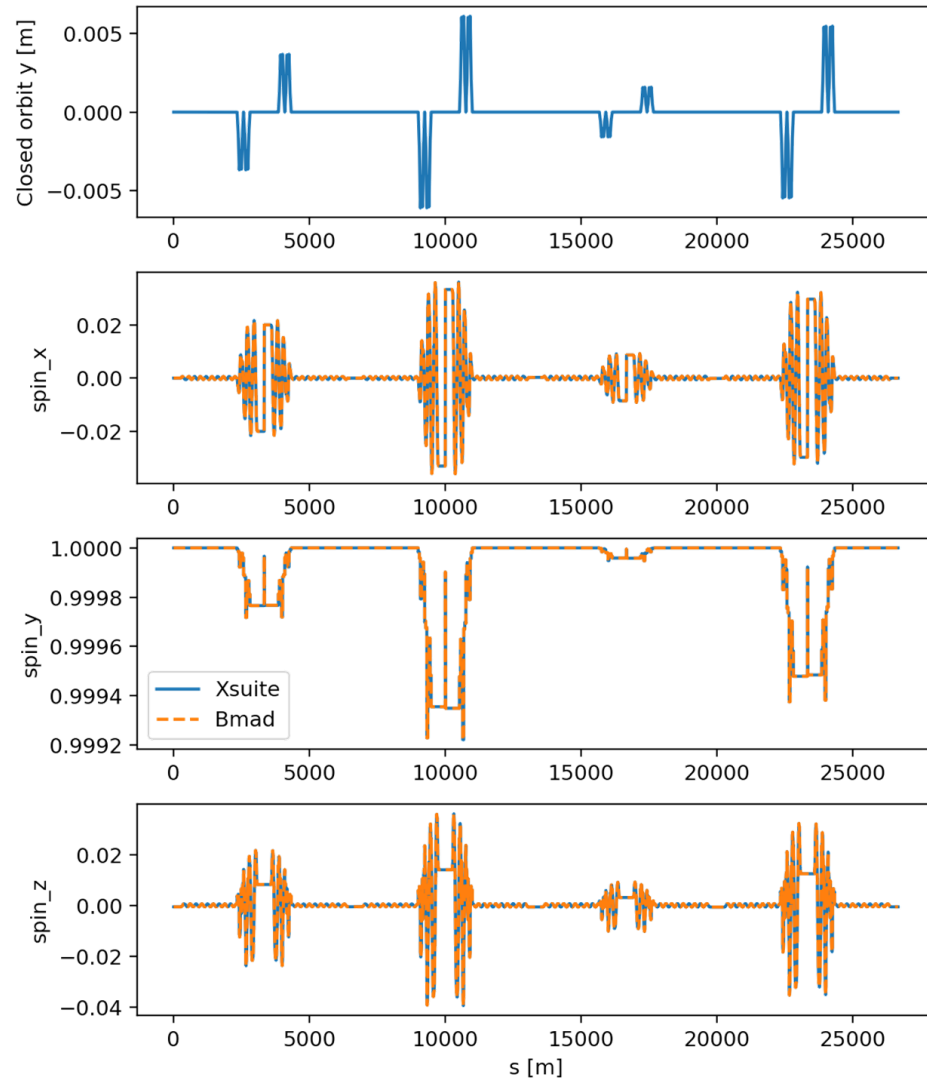
$$\tau_{\text{pol}}^{-1} = \frac{5\sqrt{3} r_e \hbar \gamma^5}{8 m_e C} \oint \frac{ds}{|\rho(s)|^3} \left[ 1 - \frac{2}{9} (\mathbf{n}_0 \cdot \hat{\mathbf{i}}_v)^2 \right]_s$$

$$\tau_{\text{depol}}^{-1} = \frac{5\sqrt{3} r_e \hbar \gamma^5}{8 m_e C} \oint \frac{ds}{|\rho(s)|^3} \left[ \frac{11}{18} \left| \frac{\partial \mathbf{n}}{\partial \delta} \right|^2 \right]_s$$



# Some comparisons against BMAD

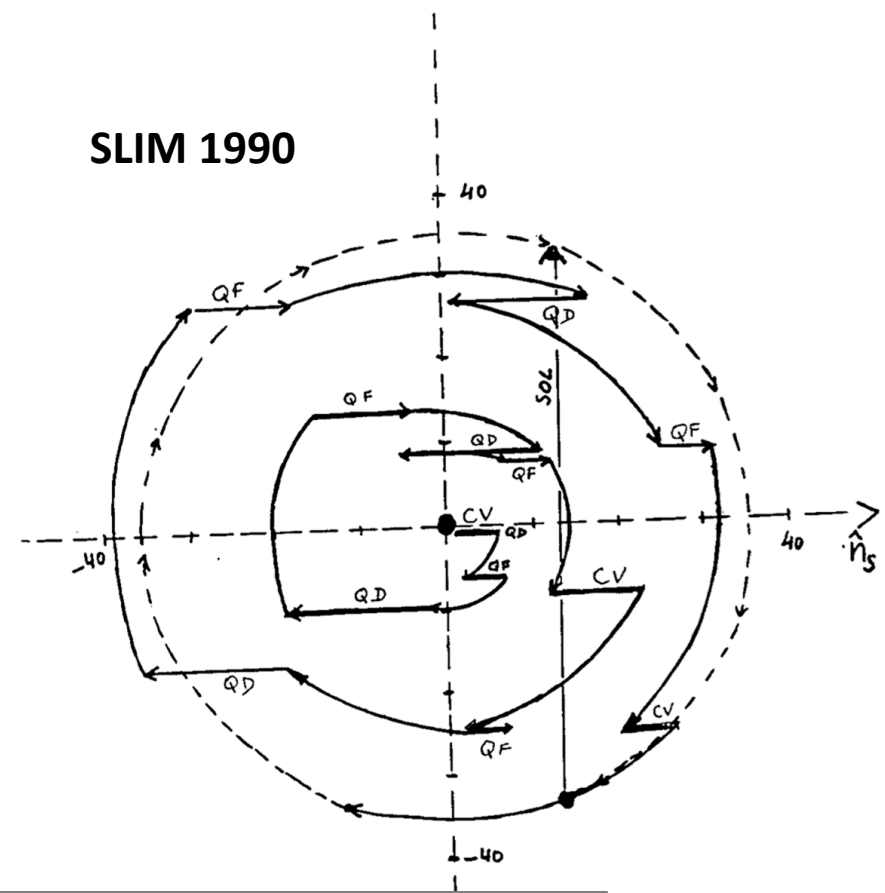
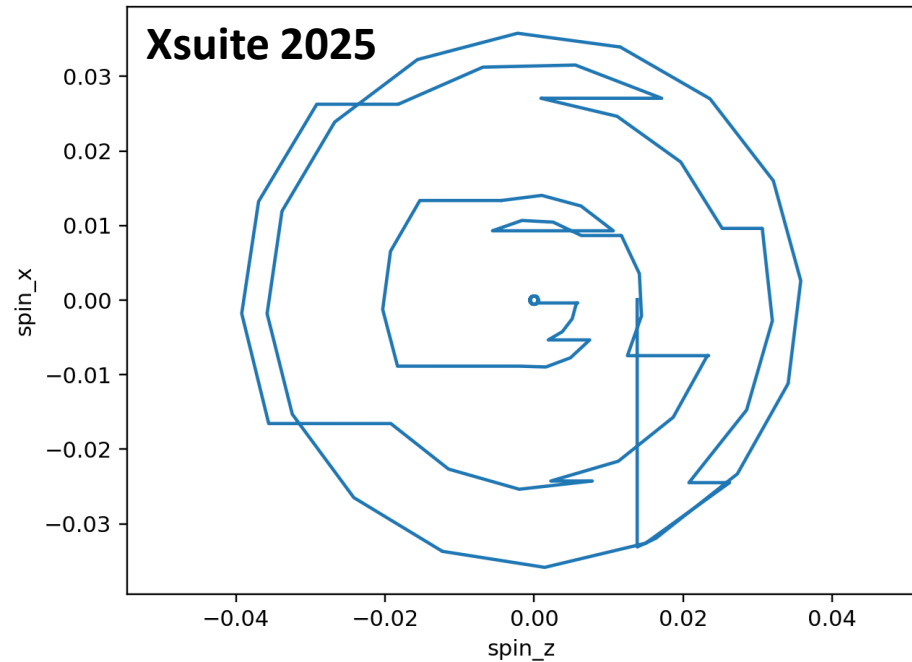
Studied the case of **LEP1** with orbit bumps to compensate precession in the experimental solenoids → **good agreement** found against BMAD





# ...and against 35-year-old SLIM simulation

Studied the case of **LEP1** with orbit bumps to compensate precession in the experimental solenoids → **good agreement** found against SLIM simulations



Compensation of Integer Spin Resonances Created by  
Experimental Solenoids

Alain Blondel  
L. P. N. H. E., Ecole Polytechnique, 91128 Palaiseau Cedex, France  
22 April 1990

10 mrad

Many thanks to J. Wenninger and Y. Wu for their input and guidance!



To account also for **non-linear effects**, the equilibrium polarization can be well approximated by the following expressions

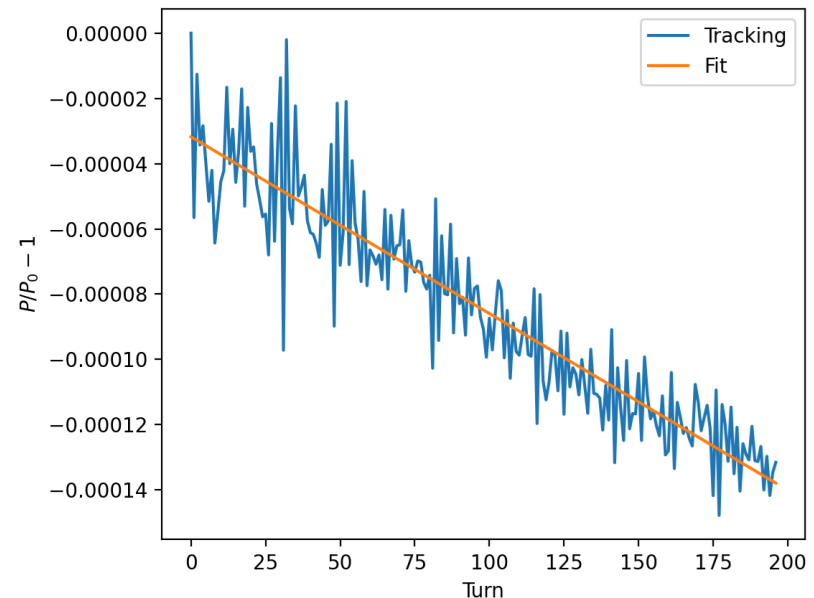
$$P_{\text{eq}} = P_{\infty} \frac{1}{1 + \frac{\tau_{\text{pol}}}{\tau_{\text{depol}}}}$$

$$P_{\infty} = \frac{8}{5\sqrt{3}} \frac{\oint \frac{ds}{|\rho(s)|^3} [\hat{\mathbf{i}}_B \cdot (\mathbf{n}_0)]_s}{\oint \frac{ds}{|\rho(s)|^3} \left[1 - \frac{2}{9}(\mathbf{n}_0 \cdot \hat{\mathbf{i}}_v)^2\right]_s}$$

$$\tau_{\text{pol}}^{-1} = \frac{5\sqrt{3} r_e \hbar \gamma^5}{8 m_e C} \oint \frac{ds}{|\rho(s)|^3} \left[1 - \frac{2}{9}(\mathbf{n}_0 \cdot \hat{\mathbf{i}}_v)^2\right]_s$$

The term  $\tau_{\text{depol}}$  can be obtained by tracking a bunch distributions in the presence of stochastic photon emission, while the other terms can be obtained by the knowledge of  $\mathbf{n}_0$  alone.

This approach is also supported by Xsuite





- Xsuite is by now the **workhorse for many FCC-ee beam dynamics studies**
- The code is **publicly available** on GitHub and can be installed through pip
  - You are **vey welcome to give it a try** for your use case
  - **Installation instructions and many examples** available in the [documentation pages](#)
  - We are very interested in getting your **feedback** (don't hesitate to contact us for issues, questions, suggestions)
- Code is **open-source** and is open to **developments from the community**
  - Get in touch if you are interested in contributing to the development

# Thanks for your attention!



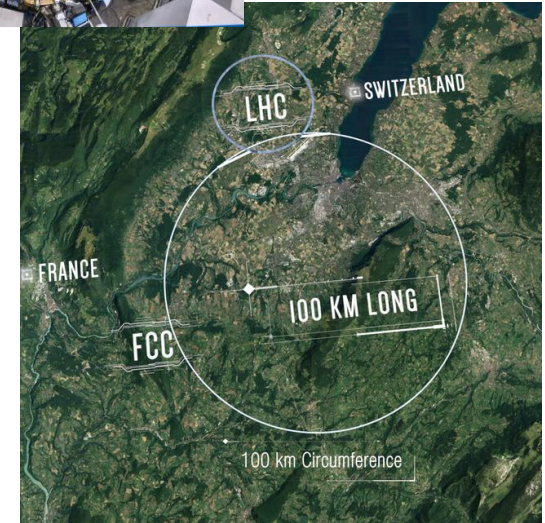
# Common toolkit for present CERN complex and FCC

Xsuite has given us the ability target with a single tool all CERN synchrotrons (present and future):  
**from ELENA to FCC-ee!**

## Tangible advantages:

- **Optimal usage of resources and expertise**
  - Efforts labelled/funded as LHC/LHC-injectors became automatically available to FCC (fast tracking, IBS, WS beam-beam, wakefields...)
  - Efforts labelled/funded as FCC automatically available to LHC and injectors (synchrotron radiation, strong-strong beam beam, Geant4 interface for collimation studies)
  - All infrastructure work is done only once and serves a very large community
- Using the same toolkit makes it **much more efficient for LHC/injectors experts to work on FCC and vice versa.**

0.03 KM LONG



- Keep up with **many (>100) small issues and requests from users**
  - Typically require ~1d of work each by a person who is skilled in accelerators and computing and has specific knowledge about our software
- Need to **introduce/complete missing key features in Xsuite** (timescale: before LS3):
  - Better aperture modelling and computation of aperture margins (Riccardo on that)
  - Missing collective effects features (monitors, feedbacks)
  - Additional features and interfaces for field imperfection and misalignments (including girders)
  - Consolidate Xsuite-MAD-NG interface for non-linear computation and optimization