Status of Fluka coupling to Sixtrack

Tracking for Collimation Workshop (WP5)
5th Joint HiLumi LHC-LARP Annual Meeting
CERN, 26-30 October 2015

Vasilis.Vlachoudis@cern.ch
on behalf of the FLUKA & Collimation team

Introduction

- SixTrack and FLUKA are simulation tools regularly used at CERN to perform LHC collimation studies.
  - SixTrack → Single particle 6D tracking code for long term tracking in high energy rings
  - FLUKA → General purpose particle physics MonteCarlo code used for machine protection, design studies, R2E, activation, collimation,...
- Collimation studies requiring energy deposition calculations, description of EM showers,...
  as well multi-turn effects need synergy between the two codes.
- Standard work path for collimation loss maps up to now:
  - SixTrack, complemented with dedicated interaction routines, predicts losses in collimators.
  - FLUKA simulates the respective inelastic interaction (= loss) and transports the products.
- Prediction of losses depends on physics models!

History

2006 Network port communication between FLUKA and a toy tracker (single-turn matrix) with a dedicated server to displace in multiple SixTrack/FLUKA clients (*V.Vlachoudis*)

2007 First implementation (*S.Gilardoni & R.Bruce*) use IcoSim by *H.Braun* (MatLab based):

- 4D tracking
- only up to sextuples included: no higher order multipoles, no skew elements / solenoids / crab cavities…
- a FLUKA job per every turn
- I/O through files

2010 FlukaIO library (*V.Vlachoudis & D.Sinuela Pastor*) first study case: SPS scrapers by *F.Cerutti & A.Mereghetti* with IcoSim

>2010 SixTrack first implementation (*V.Vlachoudis, D.Sinuela Pastor, A.Mereghetti, P.Garcia Ortega*) with tests on β-beams, 4D tracking in thick lenses

Motivation

The Coupling defines a clean separation line between both codes:

- Particle tracking → SIXTRACK
- Beam-machine interaction → FLUKA

What the coupling can provide

- Efficient simulation of beam-machine interactions in a realistic multiturn approach with a state-of-art account of physics processes
  → More accurate predictions: the best of the two codes is used.
- Avoid simplifications in the modelling of physics processes, in particular for complex interactions such as single diffractive scattering or ion interactions
- Limit human intervention (no need to manually check files, units, etc.) → Overall process less error-prone.
- Handle moving beam intercepting devices runtime (Already used in Standard SixTrack for SPS in 2006. Now, possibility to use any roto-translation of 3D geometry)
- Deal with energy ramping
- More accurate heating calculations of collimators
- Benefit from further development from both codes and their tools

Heavy ions

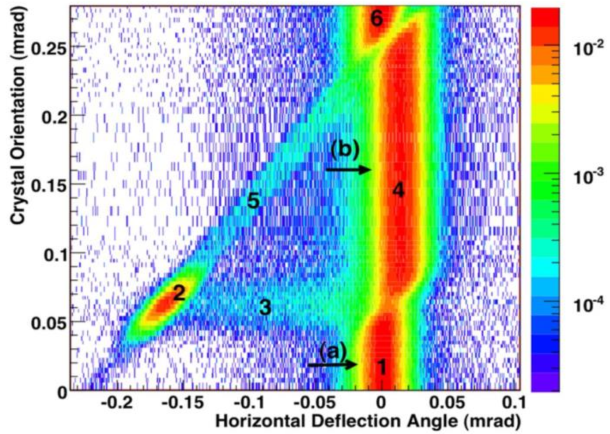
- Cross sections are energy dependent and ionization plays for ions a not negligible role in changing their energy along their path in matter
- energy loss evaluation needs a treatment significantly more sophisticated than the Bethe formula adopted in ICOSIM (e.g. including Mott corrections as well as pair production
- moreover, Landau fluctuations have to be taken into account
- it's not enough to know the probability for generating a given fragment, since its momentum is altered in the interaction; this makes fragments nominally far from the beam rigidity to fall in reality inside the machine acceptance (e.g. tritium)
- all fragments can re-interact in the collimator material

→ the coupling would intrinsically overcome all these issues

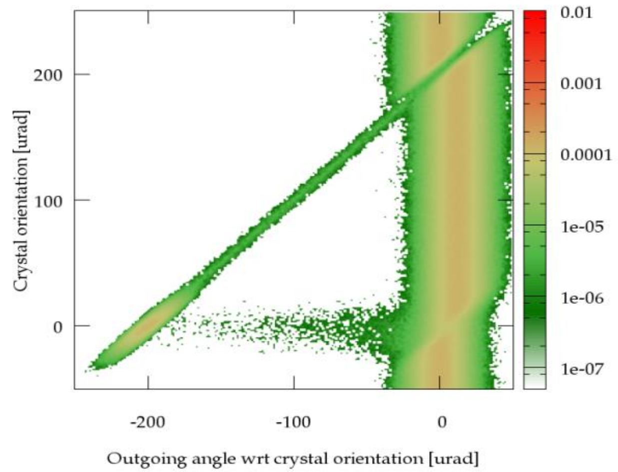
Crystal Channeling in FLUKA

- Benchmark against UA9 Experiment data (protons only)
- Qualitatively :

Experiment

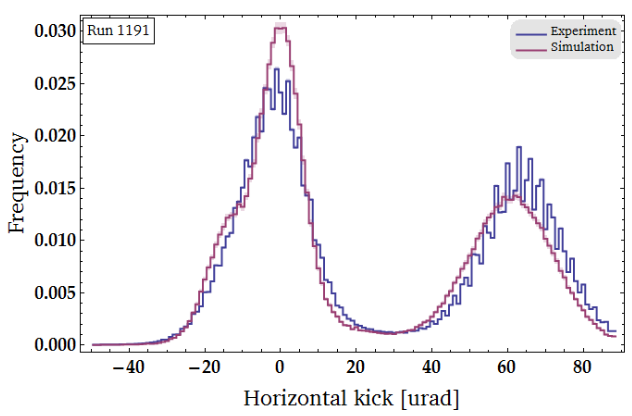


Simulation



- Quantitatively :

LHC strip 1191	VR/AM μ [urad]	VR/AM σ [urad]	CH σ [urad]	CH rate [%]	DC rate [%]
Experiment	-6.39(2)	9.97(1)	8.81(1)	39.58(6)	2.78(20)
Simulation	-3.32(4)	9.46(3)	10.90(4)	37.03(15)	2.80(5)



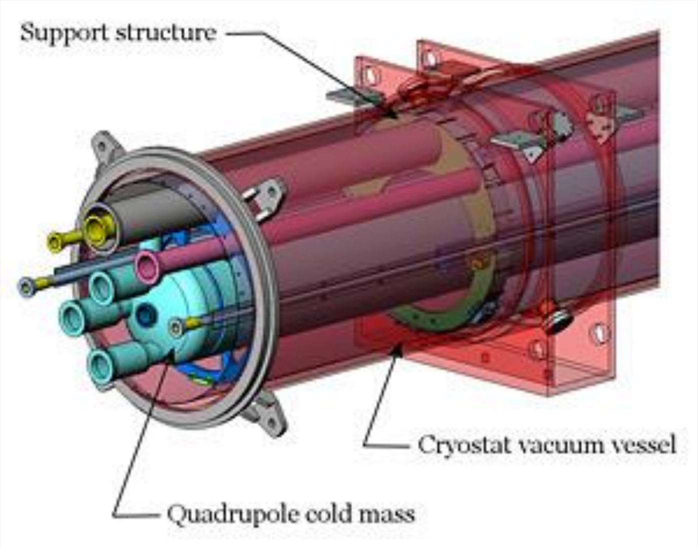
P.Schoofs

Efficient CPU time usage

- FLUKA simulations can be sometimes very CPU-intensive
  - For detailed energy deposition studies for large LHC geometries and TeV beam energies
    - where one applies low secondary production and transport cuts down to MeV energies or less (for a detailed description of shower development, particularly $e^-/e^+,\gamma$),
    - and many fine-grained scoring meshes (to calculate point-like quantities necessary to estimate e.g. quench levels),
  - It can be optimized
    - if one applies high cuts and/or if one switches off unnecessary physics processes (e^-/e^+, γ prod. and transport, generation of all inelastic collision products and their transport)
- When applied to the LHC, the coupling is meant to work in the latter mode
  - Focusing on particles which can still propagate in machine (able to create a lossmap) and not in the local shower development

The Fluka Element DataBase (FEDB)

Fluka Element DataBase

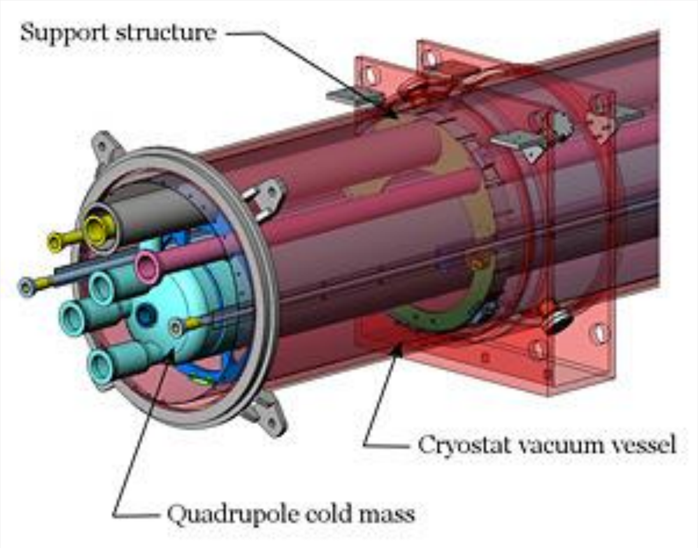


Database of Fluka models:

- **flexibility**: ease of modelling accelerator components, crucial when adding relevant details, following the evolution of the design of future devices, comparing different technical solutions...
- **consistency**: a common and shared resource where to keep track and monitor implementation details (SubVersioN repository);
- **portability**: prompt propagation of updates, with improvements being inherited by all the people involved;

The Line Builder (LB)

Fluka Element DataBase



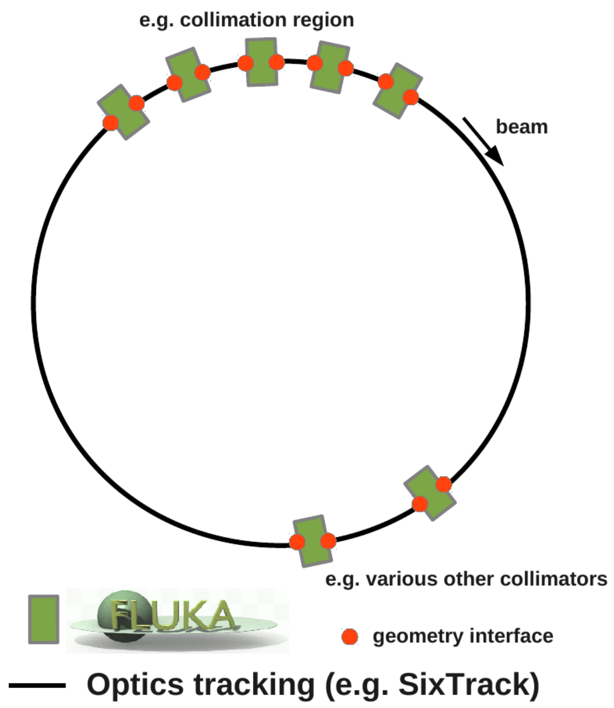
Line Builder

Assembler of geometries of actual beam lines:

- **flexibility**: automatic synchronization with the machine optics, essential ingredient for sound modelling of the line;
- **consistency**: protection against wrong settings of the beam line, e.g. misplacements, wrong orientations, mismatches in magnetic fields and collimator apertures…
- **portability**: the last updates of the FEDB are automatically used;

How

- FLUKA and SixTrack run at the same time, talking to each other.
- Exchange of particles at run-time, through a network port (dedicated FlukaIO library, TCP/IP)
- One or more portions of the accelerator lattice are labelled for transport in FLUKA the rest is handled by SixTrack.



- Primary particles are transported turn by turn by SixTrack throughout the lattice.
- When they reach a labelled element, they are transferred to
- FLUKA for transport in its 3D geometry and for simulating the interaction with accelerator components.
- At the end of the FLUKA insert, marked as a geometry interface, particles are sent back to SixTrack

Present Status

Code

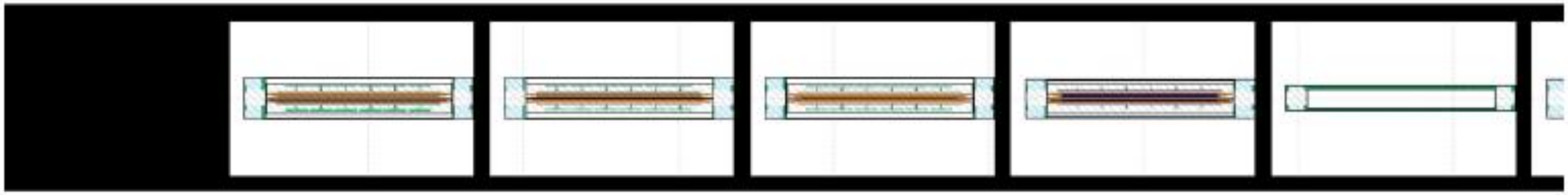
- coupling in both thin and thick lens, 4D, 6D (+acceleration)
- multiple elements can be flagged as FLUKA
- on-line aperture checks (native LIMI block) + addition of LHC aperture type (RECTELLIPSE)
- external generator of the beam to be tracked
- unique ID of new particles
- heavy ion exchange information

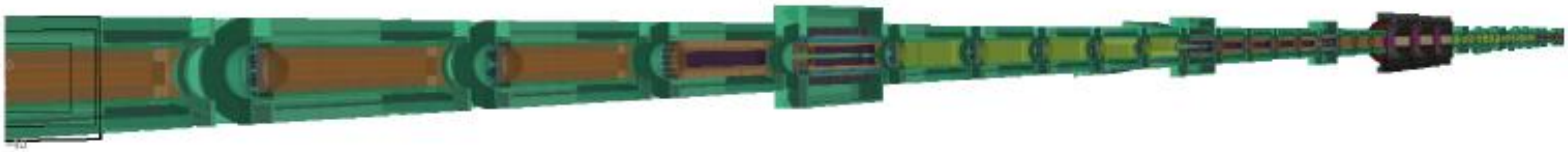
Additional

- dynamic kicks in SixTrack (magnetic bumps)
- moving FLUKA bodies run-time
- useful dumps for monitoring the simulation process
- could even be used to model complicated 3D magnetic fields

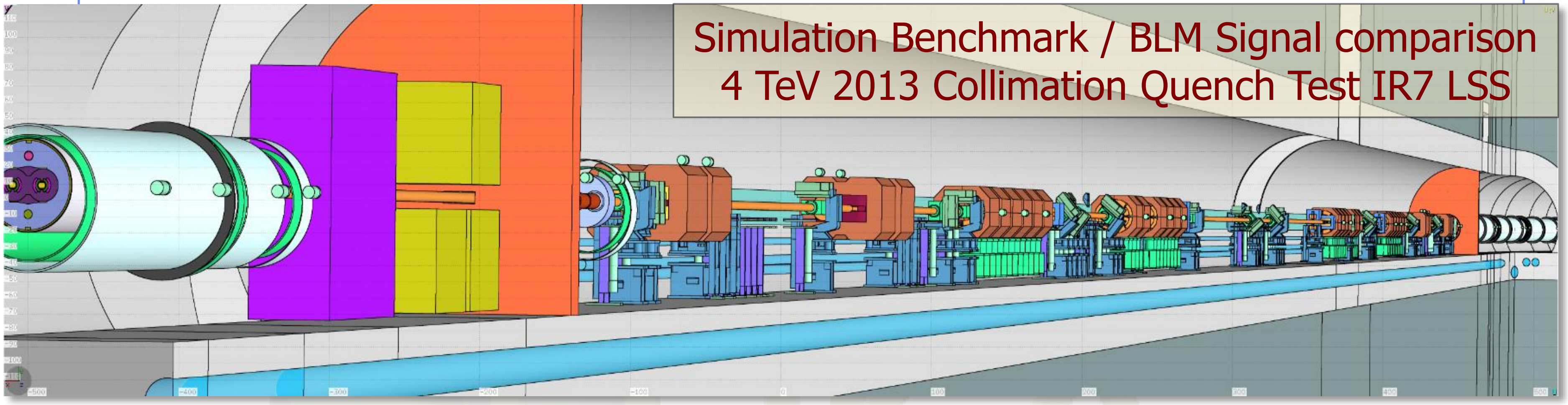
FLUKA Input

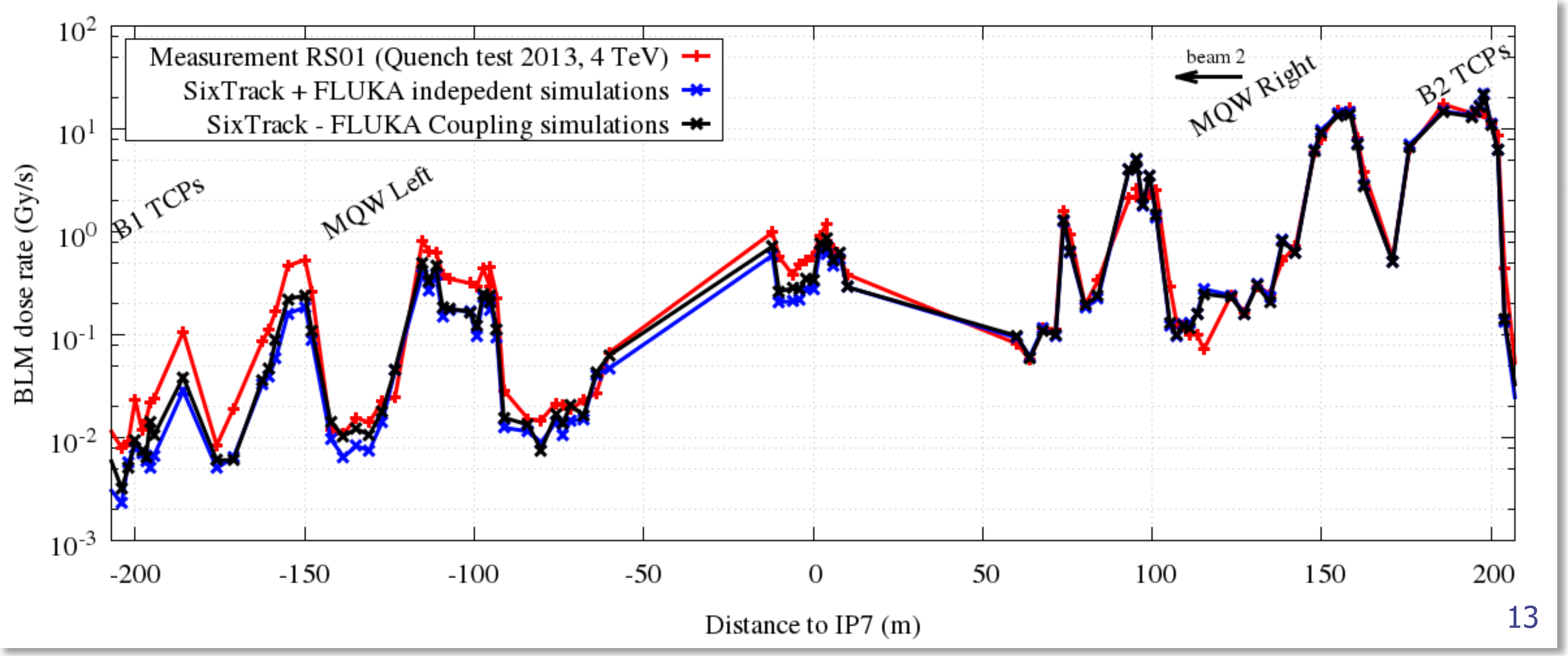
- Geometry: List of B2 collimators built in a sequence and isolated with BLACKHOLE regions.
- Fluka builder.py: Python script that automatically generates the Fluka input from FEDB and Collimation Database file (CollDB or Collgaps), developed in the framework of the LineBuilder.
  Available at the Coupling SVN Repository.
- Apertures, angles, jaw tilting, off-center deviations can be easily evaluated.
- Particles injected close to the beginning of the collimator tank.
- Plane that separates the collimator geometry from BLACKHOLE region used as the extraction plane.





Some results





Summary and Future

Prospects of integrated FLUKA/SixTrack calculations

- Offers realistic multi-turn simulations including sophisticated physics models (particularly for single diffractive scattering and ion interactions)
- CPU time appears not to be an issue for application to LHC (first feasibility tests)
- Coupling routines not only limited to SixTrack but also to other tracking codes: IcoSim, IcoSim++,…

Further Improvements

- Replace the user routines in FLUKA and integrate them as FLUKA cards
- Possible re-introduce the external server mechanism to handle anisotropic computation times between SixTrack-Fluka
- Use FLUKA as external beam generator (for beam-beam interactions)
- Heavy ions (further developments)

Present Status - Accessibility

Online Aperture Check

- essential ingredient for both loss maps around the ring and energy deposition onto beam-intercepting devices
- based on the SixTrack LIMI block
- the quality of the results relies on the slicing algorithm → limited advantage from thick lens tracking…

Repository

- Everything is accessible on lxplus (and svn)
  /afs/cern.ch/project/fluka
- Protected via an access list
  - Coupling
  - Fluka (development version)
  - LineBuilder
  - Flair

The user's point of view

SixTrack

activation and settings in fort.3

```
FLUK
bpm_scrh_b2m  bpm_scrh_b2m  1 4.0
bpm_scrh_beg  bpm_scrh_end  2 1.0
DEBU
NEXT
/
LIMI
qf.10010..1_AP  EL  76.0000 19.1500
mba.10030..1_AP RE 71.2000 17.2500
NEXT
```

FLUKA

- Isolate the each element geometry (BLCKHOLE)
- dedicated source and fluscw.f routines

insertion.txt (define the transformation)

```
1 INROT_1 OUTROT_1
2 INROT_2 OUTROT_2
```

it's the user's responsibility to guarantee the integrity of

- transformations from the local SixTrack refrence system to the FLUKA and vise-versa
- the synchronous length declared in fort.3 and the extension of the FLUKA geometry