A Comparison of Interaction Physics for Proton Collimation Systems in Current Simulation Tools

J. Molson *et al.*

Laboratoire de l'Accélérateur Linéaire

May 17, 2017

The European Circular Energy-Frontier Collider Study (EuroCirCol) project has received funding from the European Union's Horizon 2020 research and innovation programme under grant No 654305. The information herein only reflects the views of its authors and the European Commission is not responsible for any use that may be made of the information.

Thank you to the many people involved:

J. Molson[1], A. Faus-Golfe[1], R. Bruce[2], F. Cerutti[2], A. Ferrari[2], A. Mereghetti[2], S. Redaelli[2], K. Sjobak[2], V. Vlachoudis[2], Y. Zou[3], R. Barlow[4], R. Appleby[5], H. Rafique[5], S. Tygier[5]

[1] LAL, Univ. Paris-Sud, CNRS/IN2P3, Université Paris-Saclay, Orsay, Fr

[2] CERN, Geneva, CH

[3] IHEP, Beijing, CN

[4] IIAA, Huddersfield, UK

[5] University of Manchester, Manchester, UK

Proton collider parameters

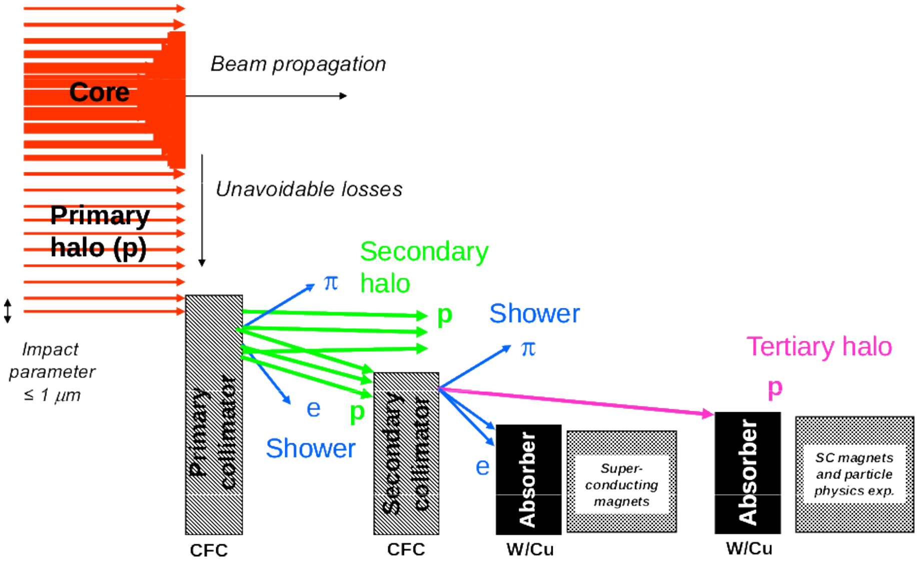
Parameter	LHC	HL-LHC	HE-LHC	SPPC	FCC-hh
Proton energy (TeV)	7	7	12.5-13.5	37.5	50
Number of bunches	2808	2808	2808	10080	10600
Protons per bunch ($\times 10^{11}$)	1.15	2.2	2.5	1.5	1
Stored energy (GJ)	0.36	0.69	**1.4**	**9.1**	**8.4**
Interaction energy (GeV)	115	115	153-159	265	306

- The interaction energy is the available energy when a proton collides with a fixed target nucleon in a collimator.
- Quench limits of superconducting magnets are $\approx 15 mWcm^{-3}$.

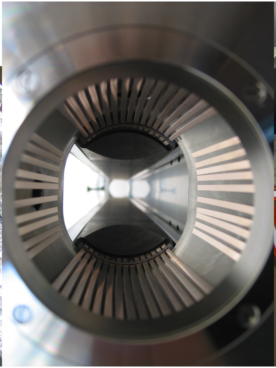
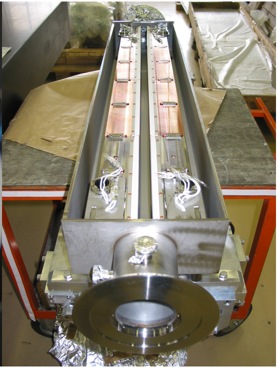
Why do we need to collimate?

- Collimators protect against two beam loss scenarios:
- 1: Slow continuous unavoidable beam halo loss.
- 2: Fast accident scenarios (magnet failures, cavity failures, injection errors, etc).
- Prevent magnet quenches.
- Reduce radiation damage to the tunnel environment and electronic systems.
- Reduce experimental backgrounds.
- High beam energy and high stored beam current make this difficult to achieve - **requires accurate simulations in order to make predictions**.

Current collimation systems



Current collimators (LHC TCSG)

Existing codes for collimation

A number of codes currently are used for collimation simulations:

- SixTrack/K2
- Merlin
- FLUKA
- BDSIM (geant4)

Are differences in codes due to tracking or to physics? Solution $\longrightarrow$ Implement all physics models in SixTrack so that the tracking, input files, post-processing will be the same.

- SixTrack is a fast symplectic beam tracking code.
- Extensively used for tracking studies at CERN for many years.
- Written in fortran and open source.

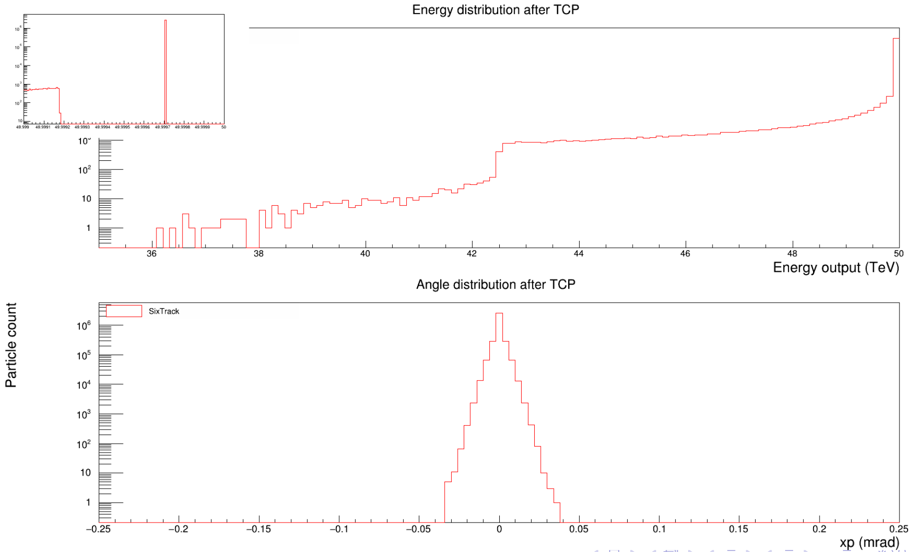
Perform simulations with each model and see what happens!

Single jaw test

- Always start with a simple test case, before any more in depth comparisons.
- Will re-create the scenario of the beam impacting the FCC-hh primary collimator.
- 60cm graphite collimator jaw
- Pencil beam - i.e. a point like distribution.
- 12.8 million protons at 50 TeV.
- Output phase space is dumped by SixTrack after the collimator.

- K2 was one of the first codes used to design the LHC collimation system.
- Eventually merged with SixTrack, and used for further collimation simulations.
- Simplified models, especially for nucleus modelling.
- Designed to be fast - only simulates protons.

`https://github.com/SixTrack/SixTrack`

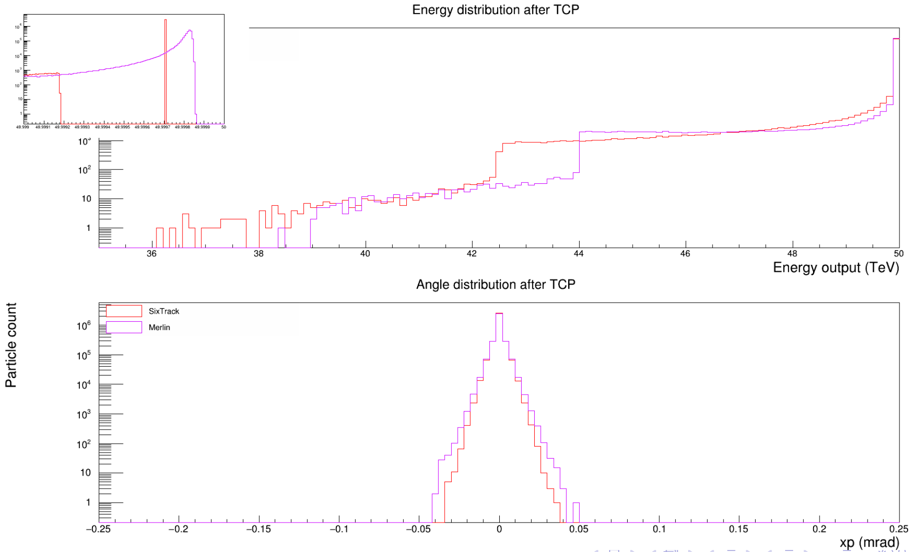

Energy distribution after TCP
Energy output (TeV)
Angle distribution after TCP
Particle count
SixTrack
xp (mrad)

Merlin

- Merlin is a C++ beam tracking code library initially used to simulate the ILC.
- Subsequently adapted to run with the LHC lattice and has been used to simulate the LHC and FCC collimation systems.
- Interaction physics is based off K2, but with a number of enhancements for calculation of the outgoing energy distribution - enhanced ionisation models, single diffraction, and elastic scattering.
- See R.Appleby *et al.* "The practical Pomeron for high energy proton collimation" Eur. Phys. J. C (2016) 76:520.

`https://github.com/MERLIN-Collaboration/MERLIN`

Merlin



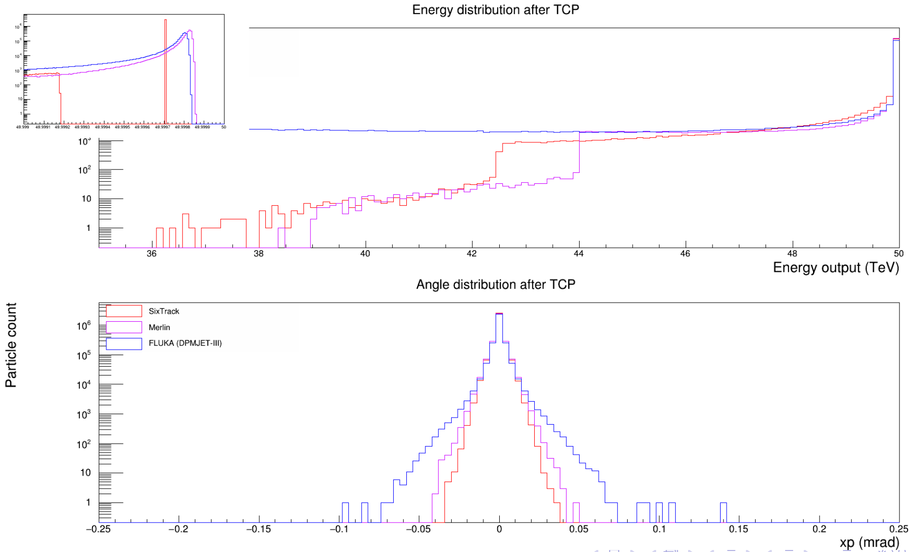
Energy distribution after TCP
Energy output (TeV)
Angle distribution after TCP
SixTrack
Merlin
Particle count
xp (mrad)

FLUKA

- FLUKA is a multipurpose code for radiation transport modelling.
- Will simulate all relevant interaction physics for collimation.
- Has already been run coupled with a special version of SixTrack for a number of years.
- Run with DPMJET-III in these simulations.
- These simulations will kill all non-proton secondary particles, and perform a cut on protons at 30% energy loss.
- Closed source and written in fortran.
- In these simulations the full collimator tank and jaw geometry is included.

`http://www.fluka.org/fluka.php`

FLUKA



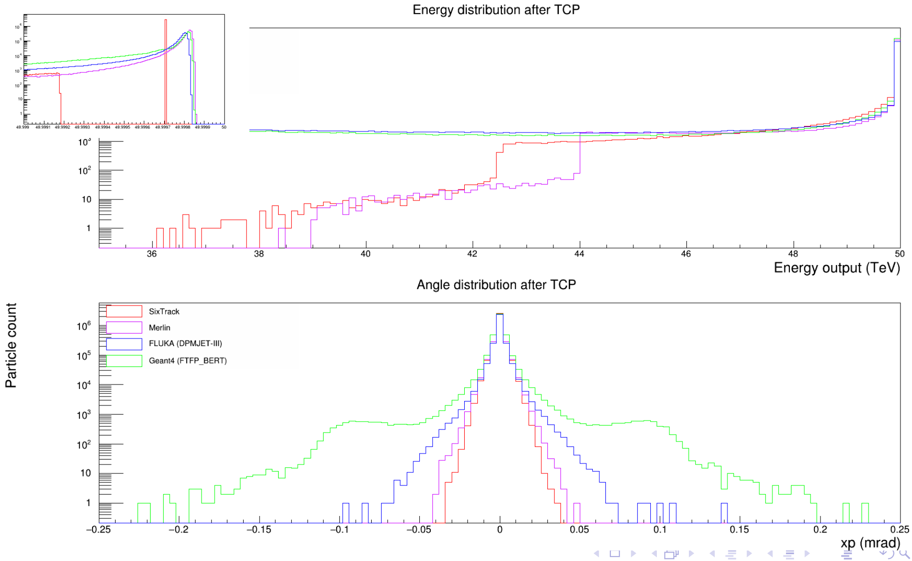
Energy distribution after TCP
Energy output (TeV)
Angle distribution after TCP
Particle count
SixTrack
Merlin
FLUKA (DPMJET-III)
xp (mrad)

Geant4

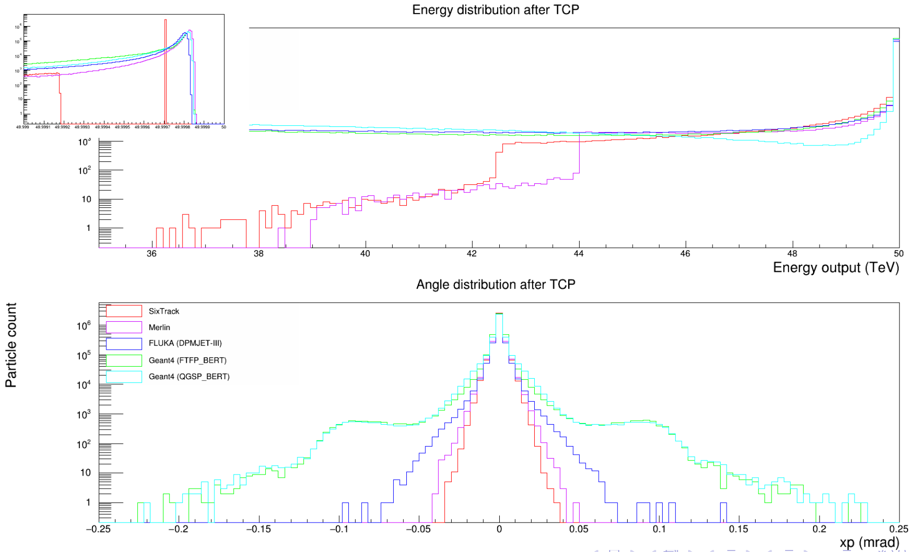
- Geant4 is a radiation transport code similar to FLUKA.
- Open source code and written in C++ as a library to link to.
- Multiple different internal physics models.
- Will use FTFP_BERT and QGSP_BERT for this work.
- Similar cuts to FLUKA are used.
- Simple “block” geometry used for the collimators.

http://geant4.web.cern.ch/geant4/

Geant4 FTFP_BERT

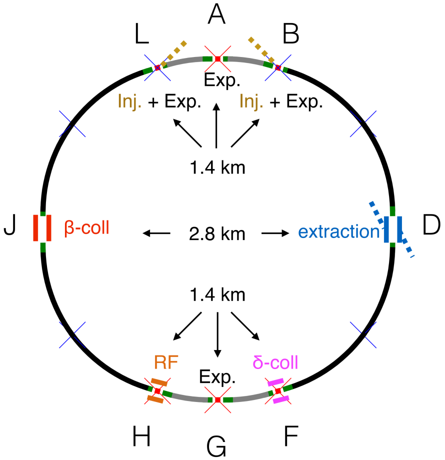


Geant4 QGSP_BERT

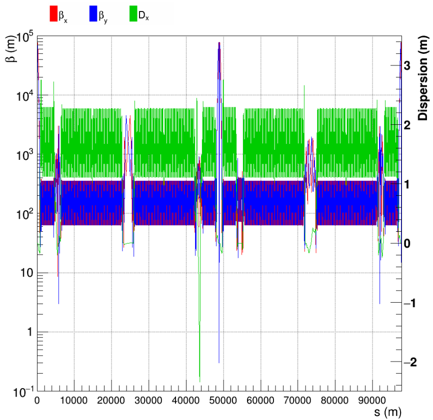


- We observe that each code produces different output distributions after interactions with a collimator.
- How important it this to the operation and effectiveness of the collimation system?
- Test with the current FCC-hh lattice.
- 200 turns, 12.8 million protons at 50 TeV, with a horizontal betatron halo distribution - a ring in x,xp that just touches the horizontal primary collimator jaw, normal distribution cut at 3 sigma in y,yp.

Loss maps show the distribution of losses around an accelerator ring. In an ideal world all losses will be confined to the collimators, and any normal conducting magnets. Losses into superconducting magnets are to be avoided.



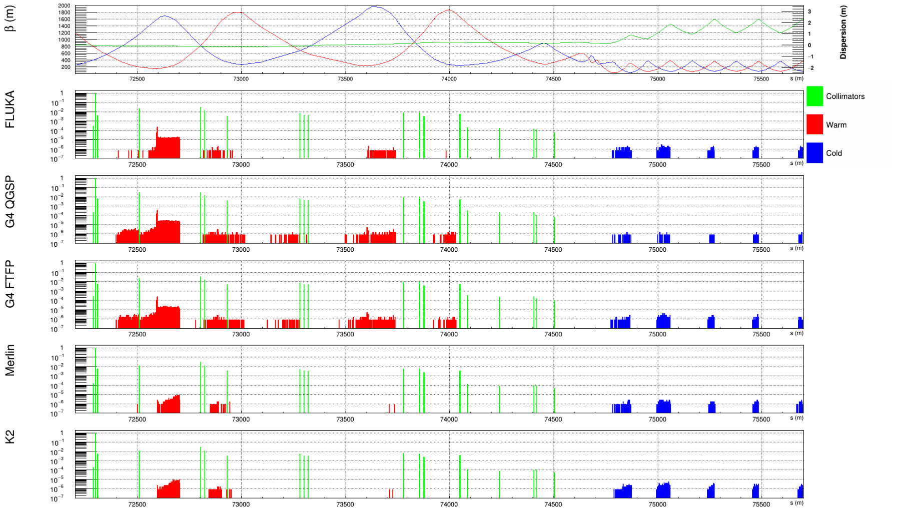
A
L
B
Exp.
Inj. + Exp.
Inj. + Exp.
1.4 km
J
β-coll
2.8 km
extraction
D
1.4 km
RF
Exp.
δ-coll
H
G
F



β_x β_y D_x
β (m)
Dispersion (m)
s (m)

Loss maps - full ring

Loss maps - betatron collimation



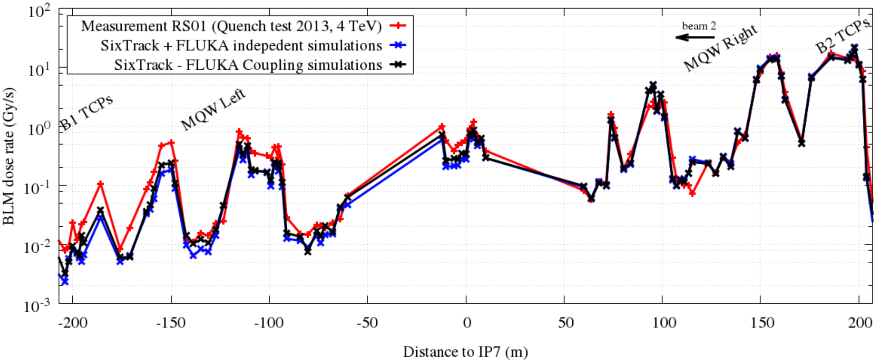
- Qualitatively there is an excellent agreement between each code's loss map.
- This gives good confidence in our simulation tools.
- When performing a more detailed quantitive comparison this changes.

Numerical comparisons

- Use K2 as a baseline since it was the first code used for collimation simulations.
- Look at the ratio of the number of losses in select regions vs K2.

Region	Merlin	FLUKA	G4 FTFP	G4 QGSP
β TCP	1.00	1.01	0.92	0.94
β TCSG	1.00	1.27	1.45	1.32
β TCLA	0.92	1.50	2.37	1.91
β DS1	0.51	0.57	0.68	0.066
β DS2	0.44	0.45	0.52	0.032
β DS3	0.41	0.43	0.51	0.027
β DS4	0.41	0.45	0.47	0.086
δ TCP	0.45	1.39	1.12	0.69
δ TCSG	0.49	1.36	1.24	0.79
δ TCLA	0.51	1.3	1.22	0.92
Total	1	1.05	0.99	0.99

Experimental data



FLUKA shows a good match with the losses measured at the LHC:

- E. Skordis *et al.* “Impact of beam losses in the LHC collimation regions”, IPAC 2015.
- E. Skordis *et al.*, “Study of the 2015 top energy LHC collimation quench tests through an advanced simulation chain”, IPAC 2017

Summary

- We can now run our simulations with more confidence of the loss locations - all codes generally agree in loss locations, differences are mostly in the magnitude of the losses.
- Therefore we know where to focus shielding efforts.
- Take care when using simulation tools, they may give unexpected results - see differences between geant4 physics lists.
- Always get a second opinion.
- More data input needed from the particle physics side!
- Many thanks to all involved.