Status of SIXTRACK code with collimation

Adapting the code for our simulations

Guillaume Robert-Demolaize – AB/ABP
Ralph Aßmann – AB/ABP
Stefano Redaelli – AB/ABP
Jean-Bernard Jeanneret –AB/ABP
Frank Schmidt –AB/ABP

Contents

1- New Modules Needed

2- Collimation Routines

3- Specific Inputs

4- Status “as-of-today” & further improvements

1- New Modules Needed

Aim: track about 1 million particle along the LHC ring to obtain reliable statistical data for the collimation system.

Sixtrack provides a full realistic LHC tracking with input from MAD-X; what we need to implement is:

-- collimation routines (from J.B. Jeanneret & R.W. Aßmann)

-- new input block in SIXTRACK

-- collimators database file & unlimited number of particles

-- preliminary code version w/ collimation (R.W. Aßmann, 7/2001)

=> ONE unified SIXTRACK version

2- Collimation Routines

For collimation studies one needs a great number of particles (~ 1 million); only 64 maximum are available in standard SIXTRACK

=> create a loop over the number of turns of tracking

e.g.: 64 particles over 25 turns => nloop = 15000 to study $9.6*10^5$ particles, computing time is ~ 30-40 hours (estimate)

Good solution in terms of statistics, mainly for determination of collimation inefficiency (~ 10^{-4})

NAP-Cluster was upgraded by factor 10, nearly half of it for collimation

How the routines work:

Normal SIXTRACK run: element-by-element checking, then apply the effect of the current element where the beam is

=> for collimation studies, one looks for each element starting with "TCP" or "TCS" (to be expanded for other collimators/absorbers).

Once a collimator is found, each particle is **treated individually**. All kind of interactions can be treated, but no shower is generated.

When exiting these routines, each particle has a flag on it that allows the program to know if it has been absorbed (case of an inelastic interaction) or not, along with its impact parameter.

3- Specific Inputs

For collimation routines, a specific input block has been added to the fort.3 SIXTRACK input file

This block contains (among others):

-- the DO_COLL flag to make the collimation studies

-- the number of loops to be done (number of turns in usual input)

-- the opening (in sigmas) of the jaws of the TCPs & the TCSs

-- flags for radial distribution or pencil beam distribution

The other required input file (other than the geometry) is a database file for all collimators one wants to study, which includes:

-- the collimator name (full name as in LHC optic file)

-- its material (C, Cu, Pb, Be, Al and W available)

-- its length, orientation, tilt and offset

4- Status "as-of-today" & further improvements

By now, we are currently finishing a beta version of SIXTRACK for collimation, Colltrack.

We fixed mandatory to this code to be able to run on any platform: **a Windows® version has been developed** with the Compaq© Visual Studio software, a Lahey version is being prepared.

Under Linux® , the code is currently in its **final steps of development** as debugging is underway (mainly some output writing procedures are undergoing deep analysis).

Current results (obtained with the Visual Studio version) :

64 particles over 25 turns, simulations made 1000 times

Optic V6.4, C system, 0.5 mm impact parameter, 7 TeV

SAMPLE	# of impacts [cumulative]	# of surviving [per sample]	# of absorbed [cumulative]	Eff 8_ [10^{-4}]	Eff 10_ [10^{-4}]	Eff 10-20_ [10^{-4}]
1	64	20	44	7272.73	7045.45	0
11	502	21	481	291.06	291.06	20.79
117	5088	19	5069	27.62	27.62	1.97
548	23847	18	23829	5.88	5.88	0.42
985	42860	21	42839	3.27	3.03	0.23
1000	43523	21	43502	2.76	2.76	0

=> COMPATIBLE WITH EXPECTATIONS

Further work:

-- use thin lens version of optic V6.5 and update collimator database

-- make the code available on every platform

-- create some output files with histograms for complete analysis

-- couple with detailed aperture model (loss maps on 1 meter scale)

-- estimate LHC cleaning performance with realistic tracking and imperfections (7/2004)

-- estimate LHC performance reach (cleaning ⇔ quench limits)

-- develop set-up and optimization procedures