Status of coupling and linking between different codes

ColUSM #129

J. Molson et al
A brief SixTrack collimation history
The distant past

• Standard SixTrack
• Crystal collimation branch
• FLUKA coupling branch
• Ion branch (fork from above FLUKA branch)
• All used by different people for different tasks, with the core SixTrack at different development levels.
Past updates (2016-2019)

- In order to perform a comparison between Merlin and Sixtrack, the Merlin scattering physics were added into Sixtrack.
- Later, geant4 linking was also added to SixTrack (for simple proton simulations).
- A comparison with FLUKA, Merlin and geant4 was performed for IPAC 2017 - see the talk “A Comparison of Interaction Physics for Proton Collimation Systems in Current Simulation Tools”.
- The FLUKA coupling branch was merged into the mainline. The ion tracking branch was also merged in. A vast number of other code updates were performed by the Sixtrack team to update the full code base, build and test system to more modern standards (fortran 2008).
- Many other updates have been performed to SixTrack. This became SixTrack 5 - see past CoIUSM/SixTrack meetings and IPAC papers for details.
Recent updates (since SixTrack 5)

- Merge in crystal branch (Marco + Veronica + Daniele).
- Add more support for general particles (electrons, pions etc) - decide to use the PDG ID numbering scheme that other codes use.
- Update the FLUKA coupling communication layer to support generic particle transport.
- Not wanting to have to keep changing this in the future - add particle spin as a pass through for FCC-ee etc, and added particle charges.
- This gives support for partially stripped ions and BFPP (still pending).
- Then moved onto updating the geant4 linking (it was still the old version with only proton support).
FLUKA coupling status
General particle support

• Sixtrack modifications allow transport of any particle type.
• Changes to the FLUKA IO communication layer allows us to now send these to the coupling server.
• A layer exists to translate PDG IDs to FLUKA ID numbers. Changes were made to the source routine to support this and to inject the particles into FLUKA itself.
• Using USRBDX cards, any form of charged particle can be sent back to sixtrack. Of course it currently does not make sense to track neutrals here (for now).
• Can send back antiprotons, electrons, pions, and so on.
USRBDX examples

*All charged particles
USRBDX 99.0 ALL-CHAR -42.0 VAROUND TRANSF_D BACK2ICO

*Individual particle types
USRBDX 99.0 APROTON -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 ELECTRON -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 POSITRON -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 MUON+ -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 MUON- -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 PION+ -42.0 VAROUND TRANSF_D BACK2ICO
USRBDX 99.0 PION- -42.0 VAROUND TRANSF_D BACK2ICO

• See the template input files or the FLUKA manual for the naming convention of particles.
Partially stripped ion support

- Looked into partially stripped ions for the gamma factory project.
- We now send the particle charge (in addition to the ID number) to the coupling.
- This is saved at entry, and if a particle performs a transition from vacuum to any material, a flag is set.
- At extraction, if this is set, the ion is assumed fully stripped and the bare charge is used, if it is not, the initial saved charge state is set.
- Effectively assumes an infinite stripping cross section.
Other changes

- The coupling has been updated to use the CERN FLUKA release (Alessio).
- Support has been added into the source routine for beam gas and ip collisions (Alessio, Francesco).
- The coupling has been updated to save the energy that reached the extraction plane, but was not sent back to the tracker for specified particle types in a file (fort.66). This is controlled by USRBDX cards (Alessio, Nuria).
Geant4 collimation status
Geant4 linking updates

- The geant4 linking required a larger overhaul to make it compatible with the current Sixtrack release.
- The interface was updated to support passing the same additional information as is sent to the FLUKA coupling.
- This includes items such as the particle PDG id, the charge state, the spin vector, the Sixtrack particle and parent ID, and the statistical weighting in addition to the usual phase space.
- These changes allow collimation with all particles types, and not just protons as before.
- Rather than use fixed physics lists as in the previous version, any list supported by geant4 can now be used (via a physics list factory).
Geant4 linking updates

- Support for cuts in kinetic energy were added in addition to selecting particle types to send back to the tracker. Some additional changes would be needed to exactly match the cut methods used in the FLUKA coupling.

- Realistic collimator materials were added and updated as required (MoGr, CuD, etc)

- EMD was not enabled by default in any physics list, but this is now enabled in all cases (Laurie).

- Support for 1 sided collimators was added (e.g. TCDQ) using the same method as for standard K2 collimation.

- Added support for recording hits and saving these via the ROOT output.
Geant4 issues

- Geant4 supports tracking the charge states of ions, but due to a bug in one physics process this is lost (even in vacuum), therefore currently there is no PSI support.

- Geant4 can make ions that are not in the ground state (e.g. $L=1,2,3$ etc) which are then sent back to SixTrack. When these are put back into geant4 the code cannot support this and exits. A work around is to simply ignore these states. For larger ions at high $L$, the energy difference between the ground and upper state can be quite large (a few GeV), and the mass and energy must be rescaled (see the following slides).

- I’ve not yet added support for the longitudinal time coordinate from geant4, so tracking for now is 5D ($\Delta s$ is reset to 0 after each pass). Do not use this for studies during the ramp until this is implemented.
Geant4 usage

• To use the geant4 collimation, simply build Sixtrack with the G4COLLIMATION flag enabled.

• To have working EMD support at the current time a custom geant4 build is required.

• In the fort.3, use the standard COLL block, and add the GNT4 block with appropriate settings.

• That is it!

• The output is the same as used in the FLUKA coupling.
Geant4 block example

• Just take an existing collimation fort.3 and add:

GNT4
/Enable debugging
/DEBUG ON
/select physics list (see G4 documentation)
PHYSICS FTFP_BERT
/select particles to return
RETURN STABLE /Select stable particles only
RETURN IONS /select all ions
RETURN 211 /select via PDG id (e.g. pion)
RETURN -211 /select via PDG id (e.g. anti-pion)
ABSENERGYCUT 1000 /cut in GeV
RANGECUT_MM 2 /geant4 range cut for particle tracking
NEXT

• Many more options are documented in the Sixtrack manual.
Geant4 + FLUKA coupling status
Some additional code changes were made to support running with both Geant4 and FLUKA enabled at the same time.

This work was triggered in order to allow using the Geant4 crystal routines with all other collimators simulated by FLUKA.

Enabling this allows simulations of a Pb ion loss map with crystals using the more trusted FLUKA ion physics routines.
Geant4 + FLUKA

• A large number of safety checks were put in place at build and run time to prevent both codes running together, and these had to be removed.

• Some code was duplicated in multiple places and had to be made more generic to be shared between the two modules in a common way.

• This included items such as some file I/O, saving of aperture losses, each module now uses a common particle ID scheme, parent IDs, and statistical weights, tracking of maximum particle IDs, etc.

• These are now tracked between the codes, so the full particle history and weights can be preserved between the codes.

• Can now use collimation features such as the internal distribution generation with the FLUKA coupling. One could even mix K2 + FLUKA.
Issues encountered

- Checks exist in both the geant4 interface and the FLUKA coupling server that the mass of the input particle matches the expected internal mass of such a particle.

- Unfortunately the masses of particles in geant4 and FLUKA do not agree. Each code uses different PDG years for the masses, and different methods for calculating ion masses. These can sometimes disagree by quite a large number.

- We could just ignore this, but this would then violate conservation of energy.

- Instead we chose each time when injecting a particle to simply rescale the mass of the particles, ensuring the energy is conserved.

\[ E_{\text{internal}} = \frac{m_{\text{internal}}}{m_{\text{input}}} E_{\text{input}} \]
Geant4 + FLUKA setup

• To run with both Geant4 and FLUKA at the same time, first build Sixtrack with both the G4COLLIMATION and FLUKA flags enabled.

• Then ensure in the fort.3 that the COLL, GNT4 and FLUKA blocks are in place and enabled.

• DANGER: special care must take place when preparing the input files. For preparing the FLUKA input, use the full collimator database. Use the same full collimator database in the COLL block. Then in the FLUKA block comment out which collimators you wish to use in Geant4. Without this the collimator numbering scheme will have a mismatch.

• Then run the FLUKA coupling as normal.
FCC-hh 50 TeV horizontal loss map

Monday, 20th July 2020

Status of coupling and linking between different codes
Summary

- Have added support for any particle type to both FLUKA and geant4 collimation when running with SixTrack.
- Both codes can be used together for combined collimation simulations.
- We are well prepared to run with any envisaged future accelerator, and will be able to perform collimation simulations.