



Managed by Fermi Research Alliance, LLC for the U.S. Department of Energy Office of Science

Simulation of Particle-Material Interactions: 2. Advanced Implementation in the Monte-Carlo Codes

Nikolai Mokhov

CERN Accelerator School

Thessaloniki, Greece

November 11-23, 2018

This Course at CAS-Thessaloniki / Nov. 2018

- Nov. 20: Simulation of Particle-Material Interactions:
 1. Basics
- **Nov. 20: Simulation of Particle-Material Interactions:
 2. Advanced Implementation in the Monte-Carlo Codes**
- Nov. 22: Comparison of Various Codes for Particle Interaction with Material

Outline

- Magnetic and Electric Fields
- Atomic Displacements (DPA) and Radiation Damage
- Five General-Purpose Particle Interaction and Transport Codes: FLUKA, GEANT4, MARS15, MCNP6 and PHITS
- MARS15-MADX-PTC Integration
- Compilers, ROOT, MPI
- Multithreading and Multi-Core Approaches
- LBNF/DUNE Application Example

Effects of Magnetic and Electric Fields

As described before, the *charged particle* pathlength x is usually subdivided into **steps** s to account for quasi-continuous effects as ionization energy loss and multiple Coulomb scattering (MCS). If a magnetic field B is present, the particle will further change its direction upon passing the step s . If an electric field E is present, the particle can in addition change its energy. These effects are governed by the Lorentz force $F = q[E + v \times B]$, where q and v are particle electric charge and velocity vector.

If the step s is already small enough – due to ionization loss and MCS constraints – one can neglect the variations of B and E on the step. Then, the new direction Ω of the particle after the step is accurately derived from the equation of helical motion in a constant field. Otherwise – especially in complex geometry and field configurations - it is found by solving a high-order Runge-Kutta equation. The particle energy gain or loss in an electric field is calculated from the field component co-linear with the particle direction of motion and – in a case of a RF cavity – taking into account the distribution of the full E phase.

Atomic Displacements (DPA) in MARS15

- Atomic displacement cross-section σ_{DPA} is a reference way to characterize the radiation damage induced by neutrons and charged particles in crystalline materials. To evaluate a number of displaced atoms, Norget, Torrens and Robinson proposed in 1975 a standard (so-called, NRT-DPA), which has been widely used since. DPA is the left side of Eq. (3) while $D = \Sigma_{\text{DPA}}$ in the right side of Eq. (3) of the first lecture.
- Energy of recoil fragments and new charge particles in (elastic and inelastic) nuclear interactions is used to calculate atomic displacement cross sections σ_{DPA} for the NRT model – w/o or with Nordlund/Stoller damage efficiency $\xi(T)$ – for a number of stable defects
- Atomic screening parameters are calculated using the Hartree-Fock form-factors and recently suggested corrections to the Born approximation
- NJOY2016+ENDF/B-VIII.0(2018) is used to generate an NRT/Nordlund/Stoller database for 490 nuclides for neutrons from 10^{-5} eV to 200 MeV; DPA in neutron-nuclear interactions above 200 MeV are treated the same way as described in the second bullet

NRT “Standard” Model to Calculate a Number of Frenkel Pairs and Damage Energy

M.J. Norgett, M.T. Robinson, I.M. Torrens Nucl. Eng. Des 33, 50 (1975)

$$N_d = \frac{0.8}{2E_d} T_d$$
$$T_d = \frac{T_r}{1 + k(Z_t, A_t, Z_r, A_r)g(T_r, Z_t, A_t, Z_r, A_r)}$$

T_r, Z_r, A_r - recoil fragment energy=primary knock-on (PKA) energy, charge and atomic mass

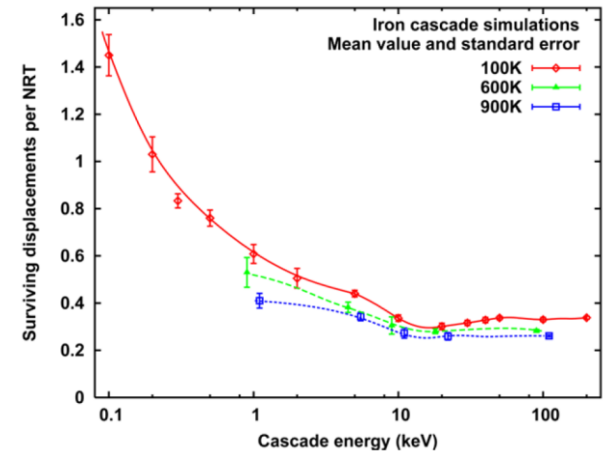
Z_t, A_t - charge and atomic mass of irradiated material

Nuclear physics (T_r, T_d) + solid state physics (N_d)

NRT-DPA is successfully applied to correlate data from many studies involving direct comparison from different irradiation environments

Efficiency Function (1): Stoller MD Parametrization

Corrections to NRT to account for atom recombination in elastic cascading. Database based on MD simulations. Its parametrization, efficiency function $\xi(T) = N_D / N_{NRT}$, is used for several years in MARS15 (=1 if >1, since 2016).



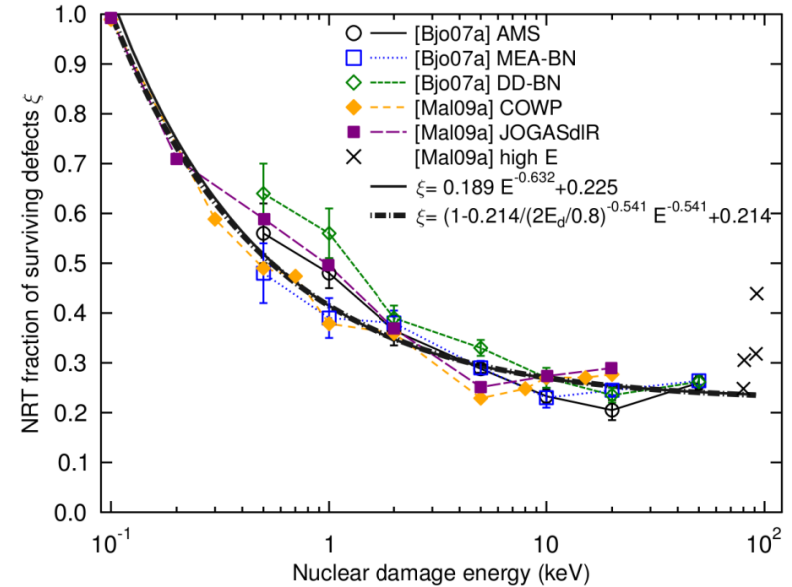
Temperature dependence. The calculations of Stoller (J. Nucl.Mater. 276 (2000) 22) for iron at 100-900K show some temperature dependence of the number of stable defects. At the same time the comparison of displacement cross-sections for p+Fe calculated using Stoller defect generation efficiency with the displacement cross-sections derived by Jung (J. Nucl. Mater. 117 (1983) 70) from low temperature experiments shows very good agreement.

Efficiency Function (2): Nordlund ARC-DPA

Nordlund's the ARC-DPA concept (athermal recombination-corrected DPA, in MARS15 since 2016):

“The recombination process does not require any thermally activated defect migration (atom motion is caused primarily by the high kinetic energy introduced by the recoil atom), this recombination is called “athermal” (i.e. it would also happen if the ambient temperature of the sample would be 0 K).”

“The arc-dpa concept allows empirical validation against frozen defects at cryogenic temperature (whereas NRT is an unobservable quantity).”



Modified NRT

$$N_d = \begin{cases} 0 & T_d < E_d \\ 1 & E_d < T_d < 2.5E_d \\ \frac{T_d}{2.5E_d} \xi(T_d) & 2.5E_d < T_d \end{cases}$$

with efficiency function

$$\xi(T) = 0.214 + 0.786 \times (2.5E_d / T)^{0.541}$$

Experimental Data Relevant to DPA Analysis

Jung et al, Greene et al and Iwamoto measured electrical resistivity change due to protons, electrons, light ions, fast and low-energy neutrons at low temperatures and low doses. It is connected to displacement cross section σ_d

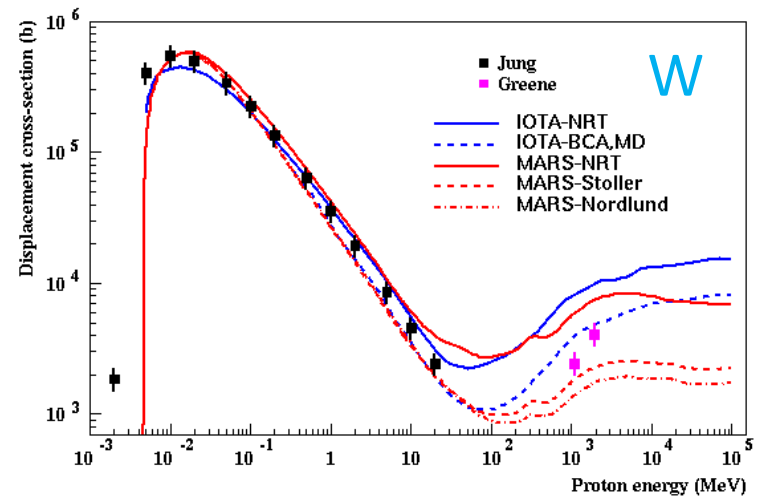
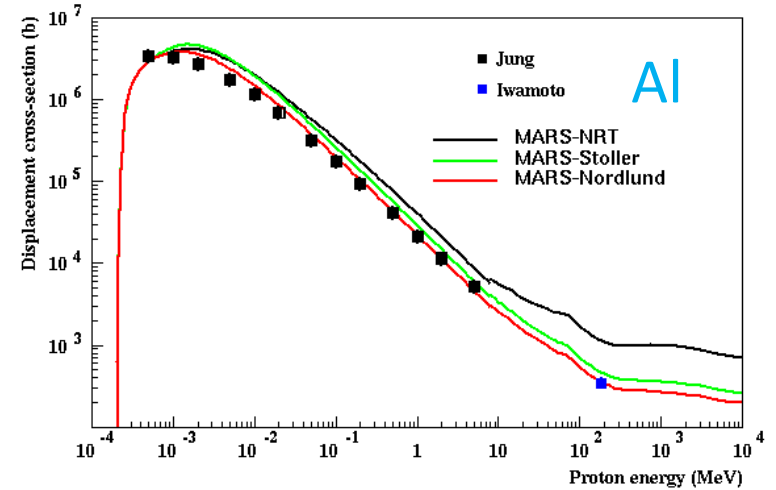
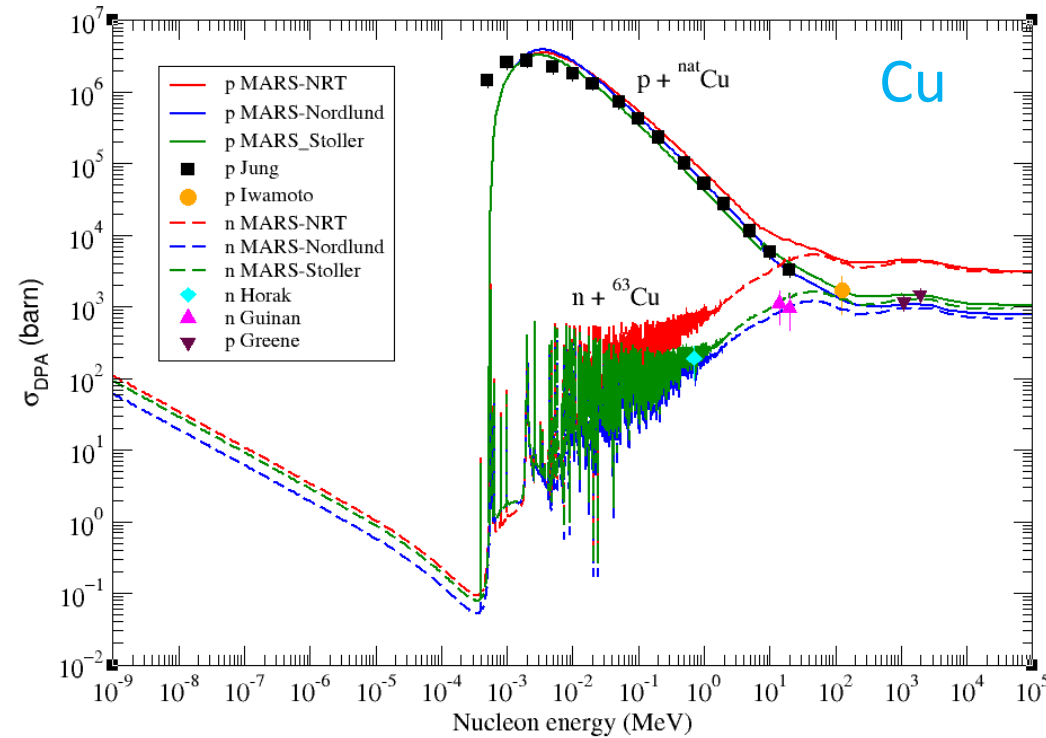
$$\Delta\rho_d(\Phi, E) = \Phi \sigma_d(E) \rho_F$$

ρ_F is a resistivity per unit concentration of Frenkel defects. This constant cannot be accurately calculated and is determined from measurements. Jung and Greene groups choose different ρ_F ($\mu\Omega\text{m/u.c.}$) for the same material

	Jung	Greene	Iwamoto
Cu	2.5 ± 0.3	2	2 ± 1
W	27 ± 6	14	-

Konobeev, Broeders and Fisher (IOTA) note that Greene's choice for W seems questionable taking into account later analysis

Proton and Neutron DPA Verification



DPA in FLUKA

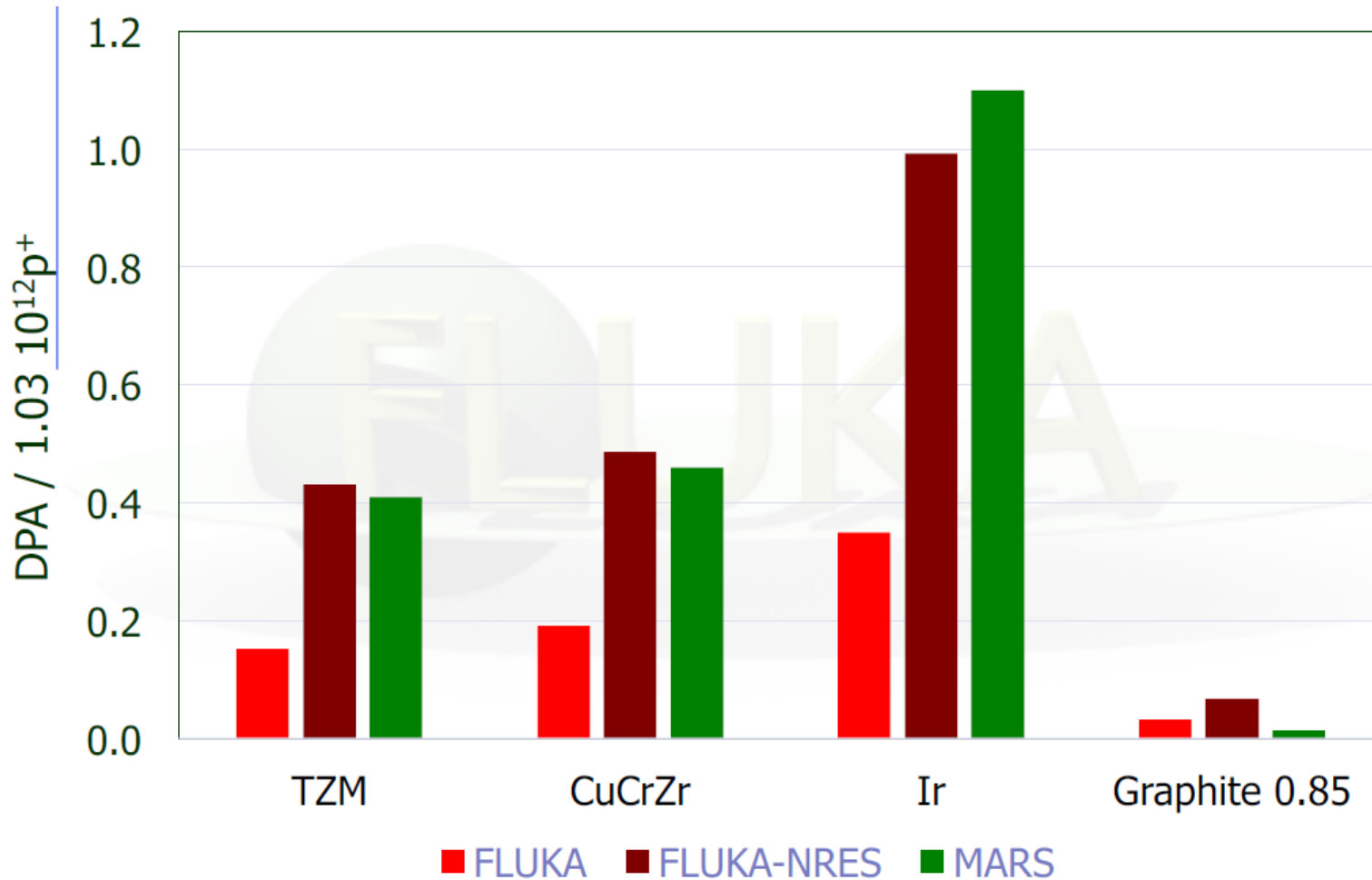
Charged particles and heavy ions

- During transport: The restricted non-ionizing energy loss
- Below threshold: The integrated nuclear stopping power with Lindhard partition
- At elastic and inelastic interactions: The is transported and treated as “below threshold”

Neutrons

- High energy $E > 20$ MeV: treat recoils after interaction as a “normal” charged particle/ion
- Low energy $E \leq 20$ MeV (group-wise): NIEL from NJOY
- Low energy $E \leq 20$ MeV (point-wise): treat recoil (if created) as a “normal” charged particle/ion

DPA at High-Z BLIP: FLUKA vs MARS15



Note: NRT | model of MARS

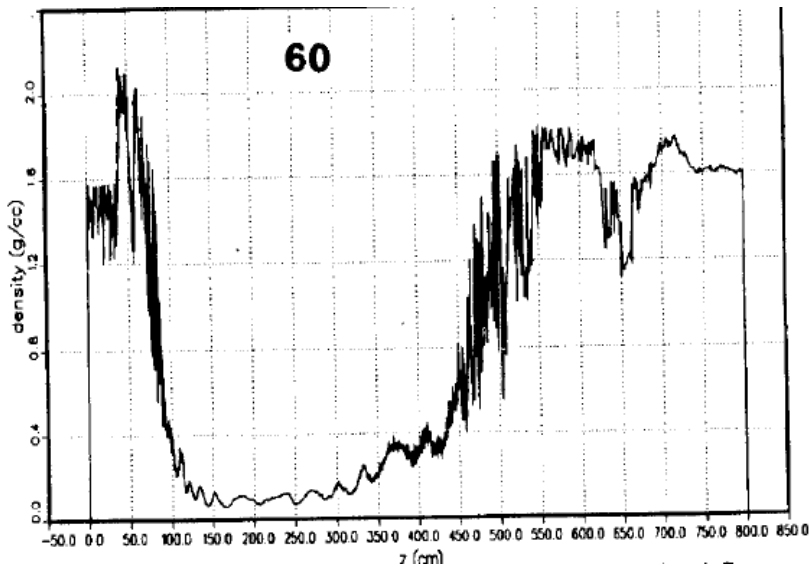
Code Benchmarking Definition

- **Debugging:** The code should calculate what is supposed to calculate
- **Validation:** Results should agree with established (or analytic) result for the specific case
- **Inter-comparison:** Two codes should agree if the model is the same
- **Verification:** The code should agree with (reliable) measurements

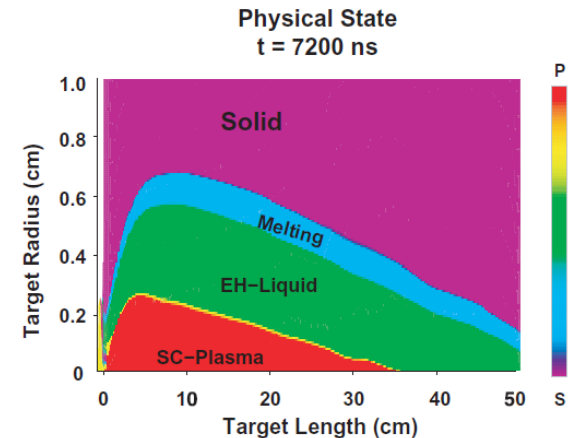
Hydrodynamic Tunneling in Solid Materials

Pulses with EDD >15 kJ/g: hydrodynamic regime.

First done for the 300- μ s, 400-MJ, 20-TeV proton beams for the SSC graphite beam dump, steel collimators and tunnel-surrounding Austin Chalk by SSC-LANL Collaboration (D. Wilson, ..., N. Mokhov, PAC93, p. 3090). Combining MARS ED calculations at each time step for a fresh material state and MESA/SPHINX hydrodynamics codes.



The hole was drilled at the 7 cm/ μ s penetration rate. Shown is axial density of graphite beam dump in 60 μ s after the spill start

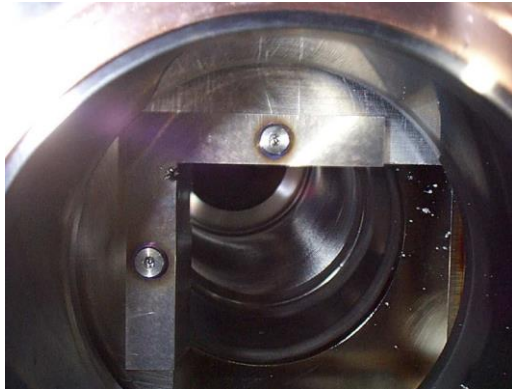


Later, studies by N. Tahir et al with FLUKA+BIG2 codes for SPS & LHC

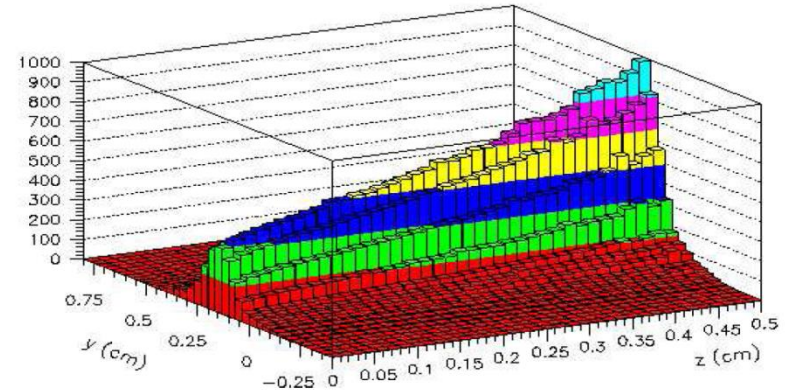
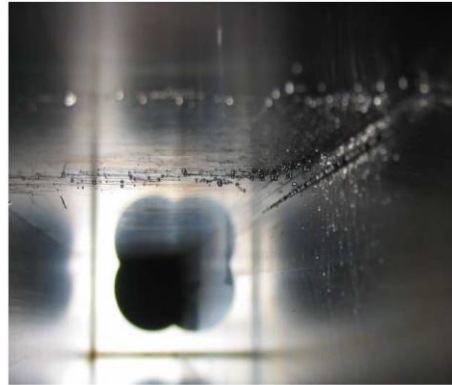
These days we use MARS+FRONTIER

Tevatron Tungsten Collimator Ablation

Hole in 5-mm W



25-cm groove in SS



Detailed modeling of dynamics of beam loss (STRUCT), energy deposition (MARS15) as high as 1 kJ/g, and time evolution over 1.6 ms of the tungsten collimator ablation, fully explained what happened

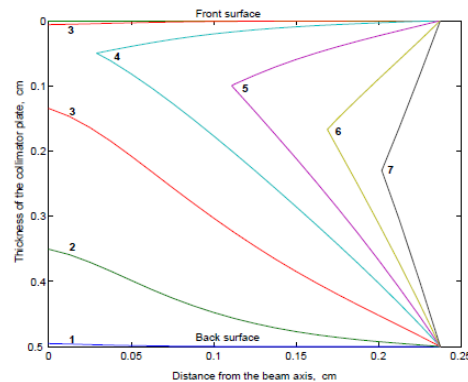
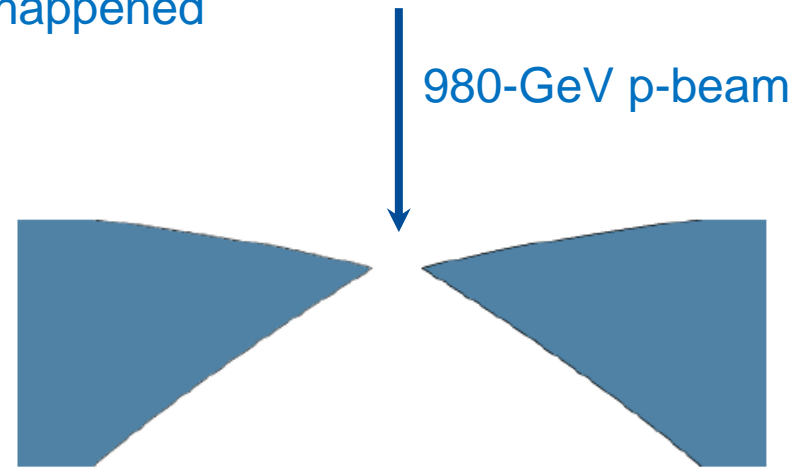
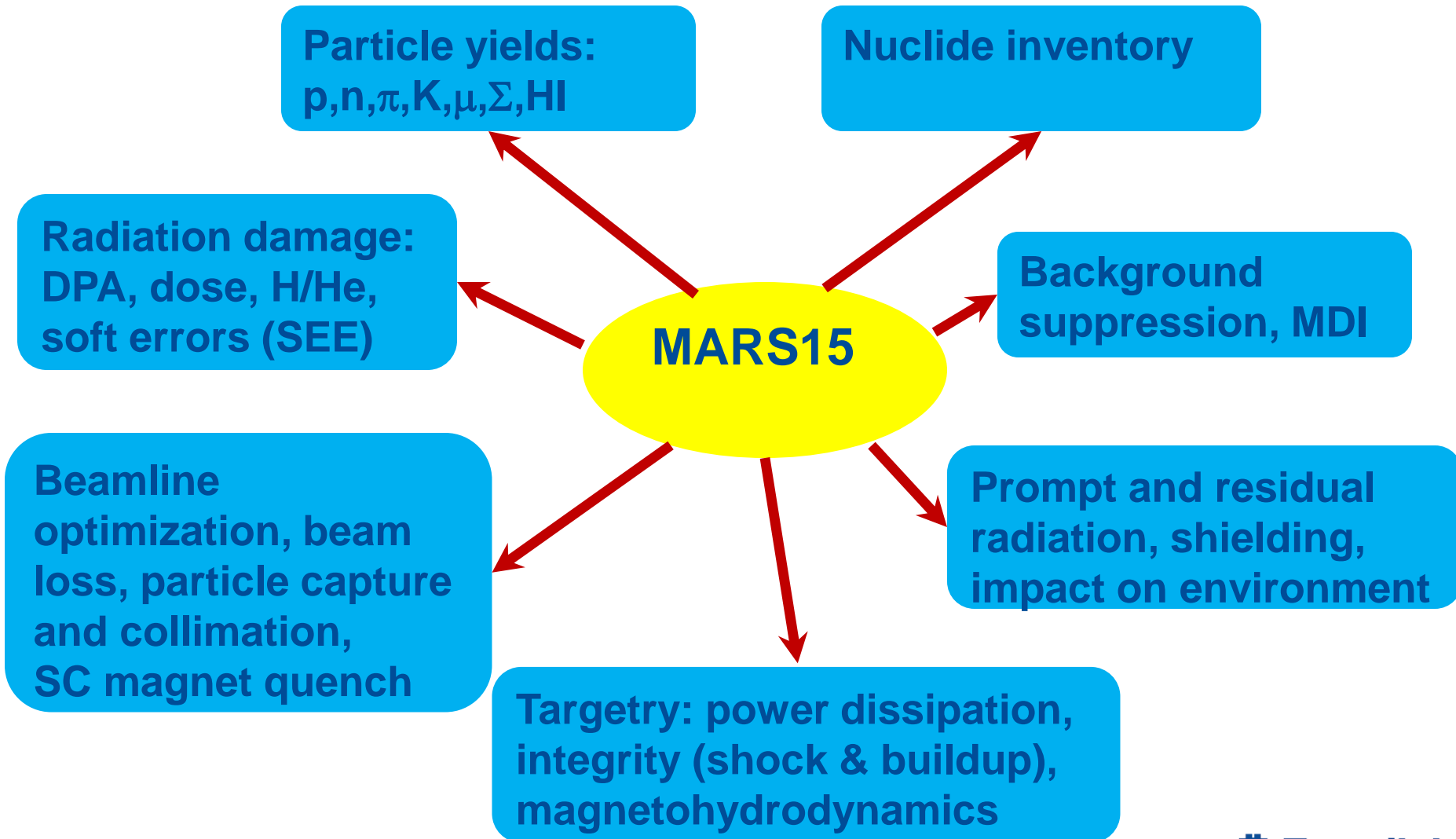


Figure 7: Evolution of the front and back surfaces of the collimator plate at $t = 0.4_{[1]} - 1.6_{[7]} ms$ with $\Delta t = 0.2 ms$.



Example of Simulation Code Applications (true for the codes considered here)



Five Codes Widely Used Around the Globe

The use of **general-purpose particle interaction and transport Monte Carlo codes** is nowadays the most accurate and efficient choice for assessing impact and consequences of particle-matter interactions at accelerators. Due to the vast spread of such codes to all areas of particle physics and the associated extensive benchmarking with experimental data, **the modeling has reached an unprecedented accuracy.**

Furthermore, most of these codes allow the user to simulate all aspects of a high energy particle cascade in one and the same run: from the first interaction of a primary beam (of up to TeV energies) over the transport and re-interactions (hadronic and electromagnetic) of the produced secondaries, to detailed nuclear fragmentation, the calculation of radioactive decays, secondary electromagnetic showers, muon and neutrino generation and their interaction with surroundings.

A brief account of these codes – taken from Review of Particle Physics, Phys. Rev. **D98, 030001 (2018)** and extended with principal features and examples – is given in the following slides for

FLUKA, GEANT4, MARS15, MCNP6 and PHITS

FLUKA (www.fluka.org)

FLUKA is a general-purpose particle interaction and transport (**Fortran 77**) code. It comprises all features needed for radiation protection, such as detailed hadronic and nuclear interaction models up to 10 PeV, full coupling between hadronic and electromagnetic processes and numerous variance reduction options. The latter include weight windows, region importance biasing, and leading particle, interaction, and decay length biasing (among others).

The capabilities of FLUKA are very good for studies of induced radioactivity, especially with regard to nuclide production, decay, and transport of residual radiation (cite from RPP). In particular, particle cascades by prompt and residual radiation are simulated in parallel based on the microscopic models for nuclide production and a solution of the Bateman equations for activity build-up and decay. **FLUKA is de facto the official code in numerous LHC and other applications at CERN.**

FLUKA's Features (1)

The highest priority in the design and development of FLUKA has always been the implementation and improvement of sound and modern physical models. Microscopic models are adopted whenever possible, consistency among all the reaction steps and/or reaction types is ensured, conservation laws are enforced at each step, results are checked against experimental data at single interaction level.

As a result, final predictions are obtained with a minimal set of free parameters fixed for all energy/target/projectile combinations. Therefore results in complex cases, as well as properties and scaling laws, arise naturally from the underlying physical models, predictivity is provided where no experimental data are directly available, and correlations within interactions and among shower components are preserved.

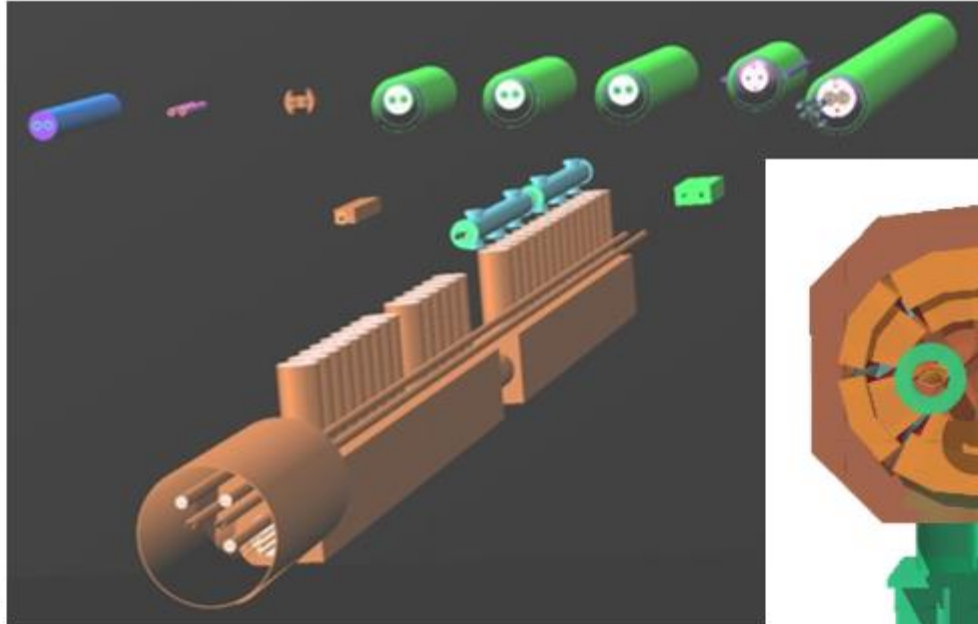
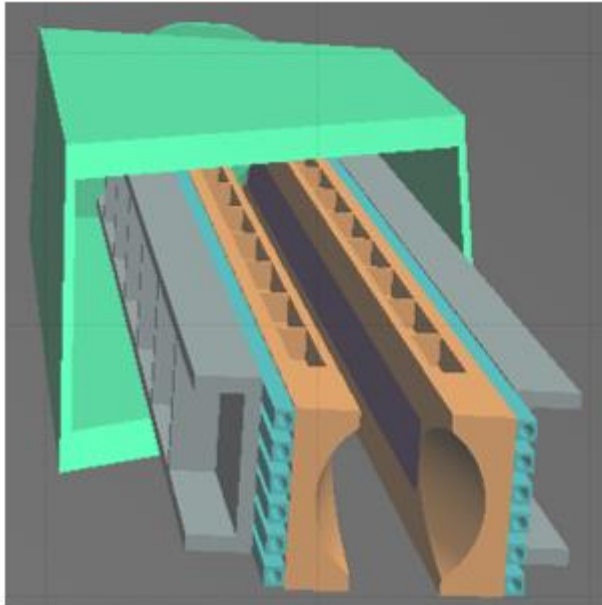
FLUKA's Features (2)

FLUKA can handle very complex geometries, using an improved version of the well-known Combinatorial Geometry (CG) package. The FLUKA CG has been designed to track correctly also charged particles (even in the presence of magnetic or electric fields). Various visualisation and debugging tools are also available.

Similar to the MARS15 code, FLUKA has a double capability to be used in a biased mode as well as a fully analogue code. That means that while it can be used to predict fluctuations, signal coincidences and other correlated events, a wide choice of statistical techniques are also available to investigate punch-through or other rare events in connection with attenuations by many orders of magnitude.

FLUKA Geometry Modeling (1)

NEED FOR DETAILED MODELS OF ACCELERATOR COMPONENTS WITH ASSOCIATED SCORING

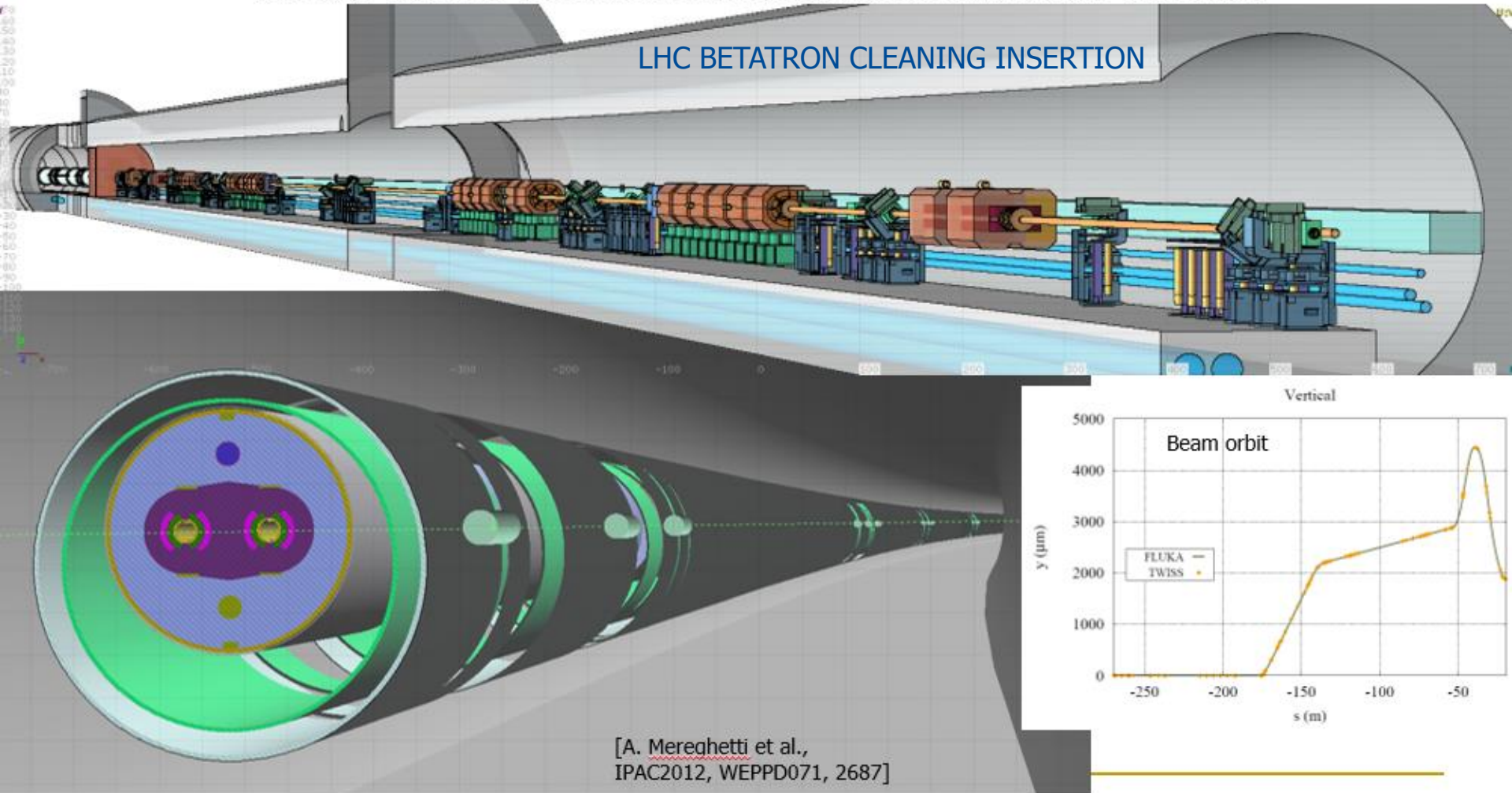


ELEMENT SEQUENCE AND RESPECTIVE MAGNETIC STRENGTHS
IN THE MACHINE OPTICS (TWISS) FILES

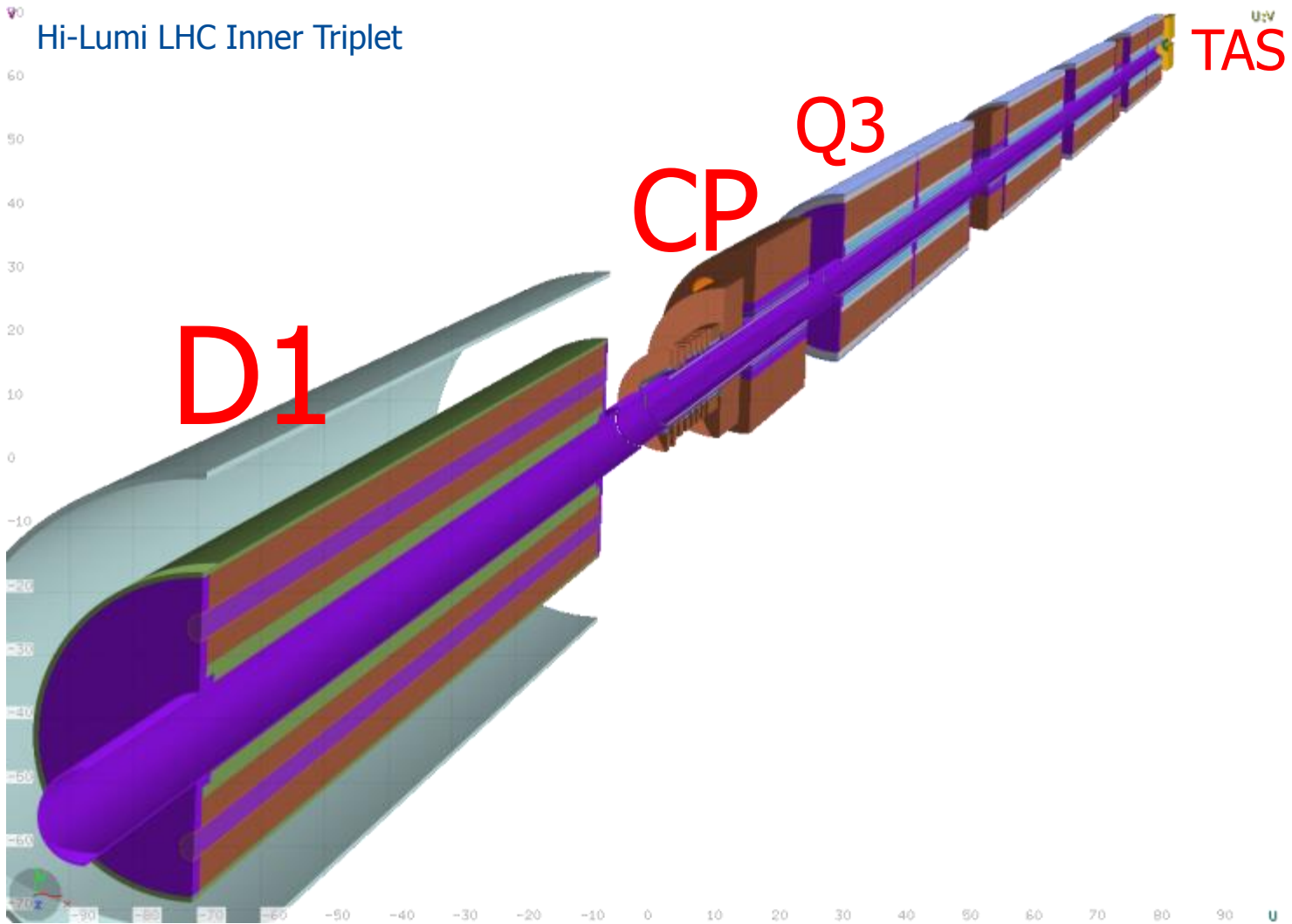
F. Cerutti

FLUKA Geometry Modeling (2)

Profiting from roto-translation directives and replication (lattice) capabilities, the AUTOMATIC CONSTRUCTION OF COMPLEX BEAM LINES, including collimator settings and element displacement (BLMs), is achievable



FLUKA Geometry Modeling (3)



F. Cerutti



GEANT4 (geant4.cern.ch)

GEANT4 is an **object-oriented toolkit** consisting of a kernel that provides the framework for particle transport, including tracking, geometry description, material specifications, management of events and interfaces to external graphics systems. The kernel also provides interfaces to physics processes. It allows the user to freely select the physics models that best serve the particular application needs. Implementations of interaction models exist over an extended range of energies, from optical photons and thermal neutrons to high-energy interactions required for the simulation of accelerator and cosmic ray experiments.

G4 is the industry standard for HEP detector simulation. To facilitate the use of variance reduction techniques, general-purpose biasing methods such as importance biasing, weight windows, and a weight cut-off method have been introduced directly into the toolkit. Other variance reduction methods, such as leading particle biasing for hadronic processes, come with the respective physics packages.

GEANT4 Physics Models

A comprehensive set of the well-established models comprises GEANT's physics lists for users to choose from. Substantial efforts were and still are put by the GEANT4 team on validation and verification of electro-magnetic physics in the code and hadronic physics loosely defined to cover any reaction which can produce hadrons in final state: purely hadronic interactions, lepton- and gamma-induced nuclear reactions, and radioactive decay.

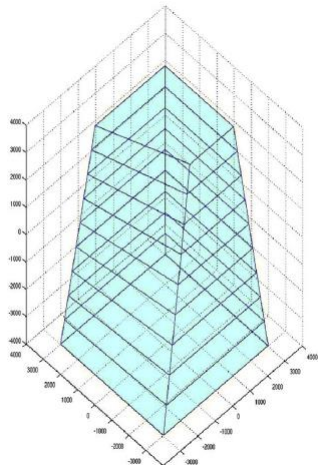
Models and x-sections are provided which span an energy range from sub-eV to TeV. Following the toolkit philosophy, more than one model or process is usually offered in any given energy range in order to provide alternative approaches for different applications.

GEANT4's performance was noticeably improved after several international benchmarking campaigns over last 15 years.

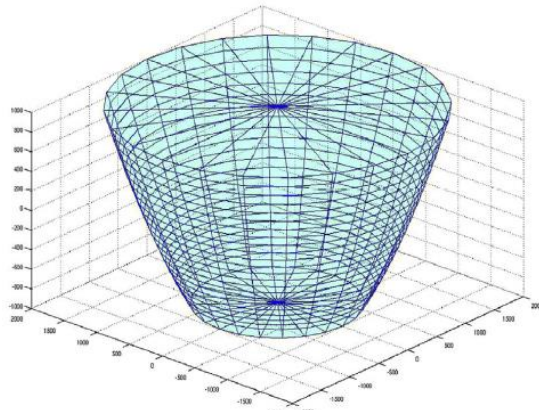
GEANT4 Geometry

There are several ways to build a geometry model in GEANT4. The standard one is to write a C++ code that contains all the definitions, material, dependence, position and hierarchy assignments, and arranges all these in the model. The shapes or geometrical primitives can be taken from a comprehensive library built in the toolkit. Fragments of the model can be imported and exported from external files according to two different formats: GDML or plain ASCII text (next slide).

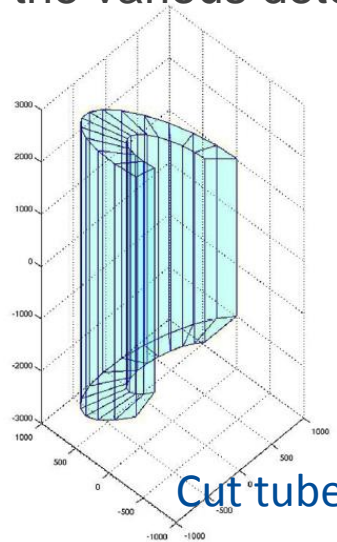
GEANT4 provides internal modules which allow the interpretation and conversion of these formats to and from the internal geometry representation, without the need for C++ programming for the implementation of the various detector description setups.



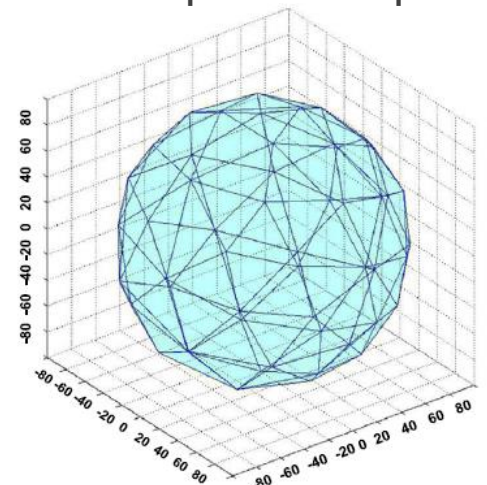
Trapezoid



Parabolic solid



Cut tube



Sphere



GEANT4: Example of Geometry ASCII Text Format

```
// Define a parameter for later use
:P DIMZ 5.

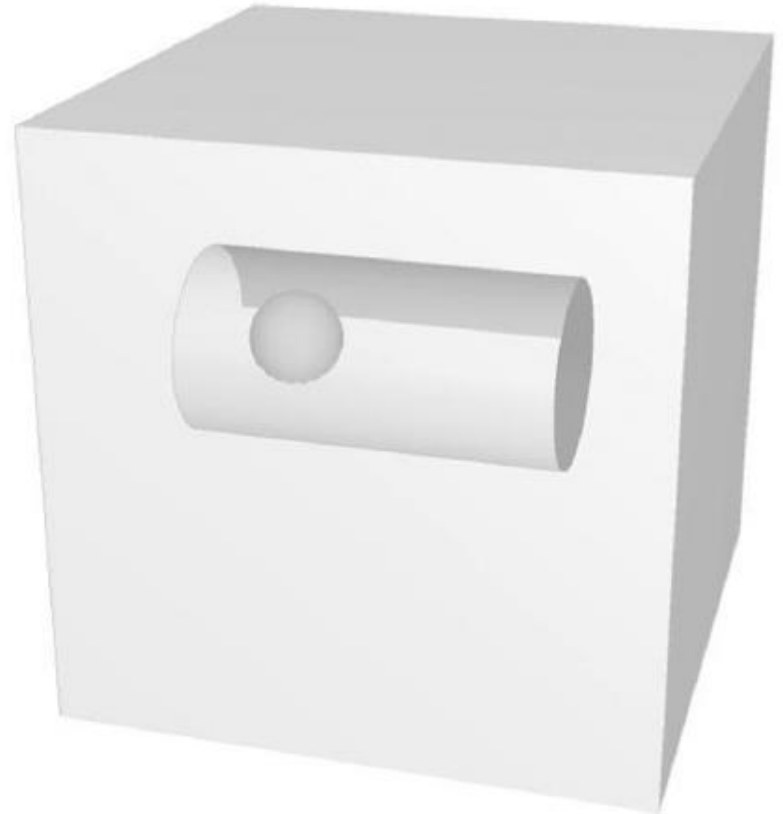
// Define materials
:ELEM Hydrogen H 1. 1.
:ELEM Oxygen O 8 16.
:ELEM Nitrogen N 7 14.
:MIXT Air 1.214E-03 2
      Nitrogen 0.75
      Oxygen 0.25

// Define rotation matrix
:ROTM R00 0. 0. 0. // unit matrix

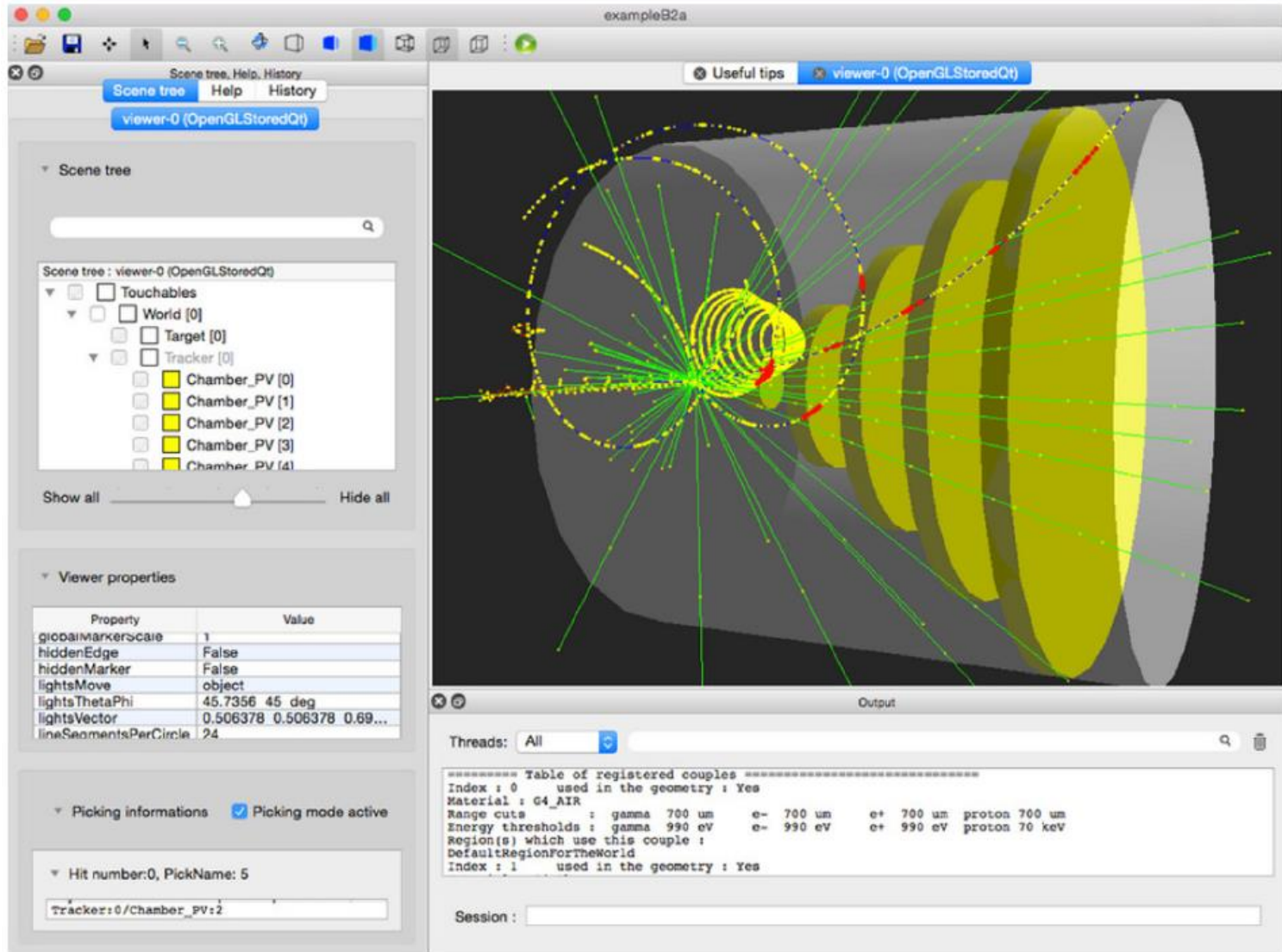
// Define volumes and place them
:VOLU world BOX 30. 30. 30. Air

:VOLU "my tube" TUBE 0. 10. $DIMZ*4 G4_WATER
:PLACE "my tube" 1 world R00 0. 0. $DIMZ

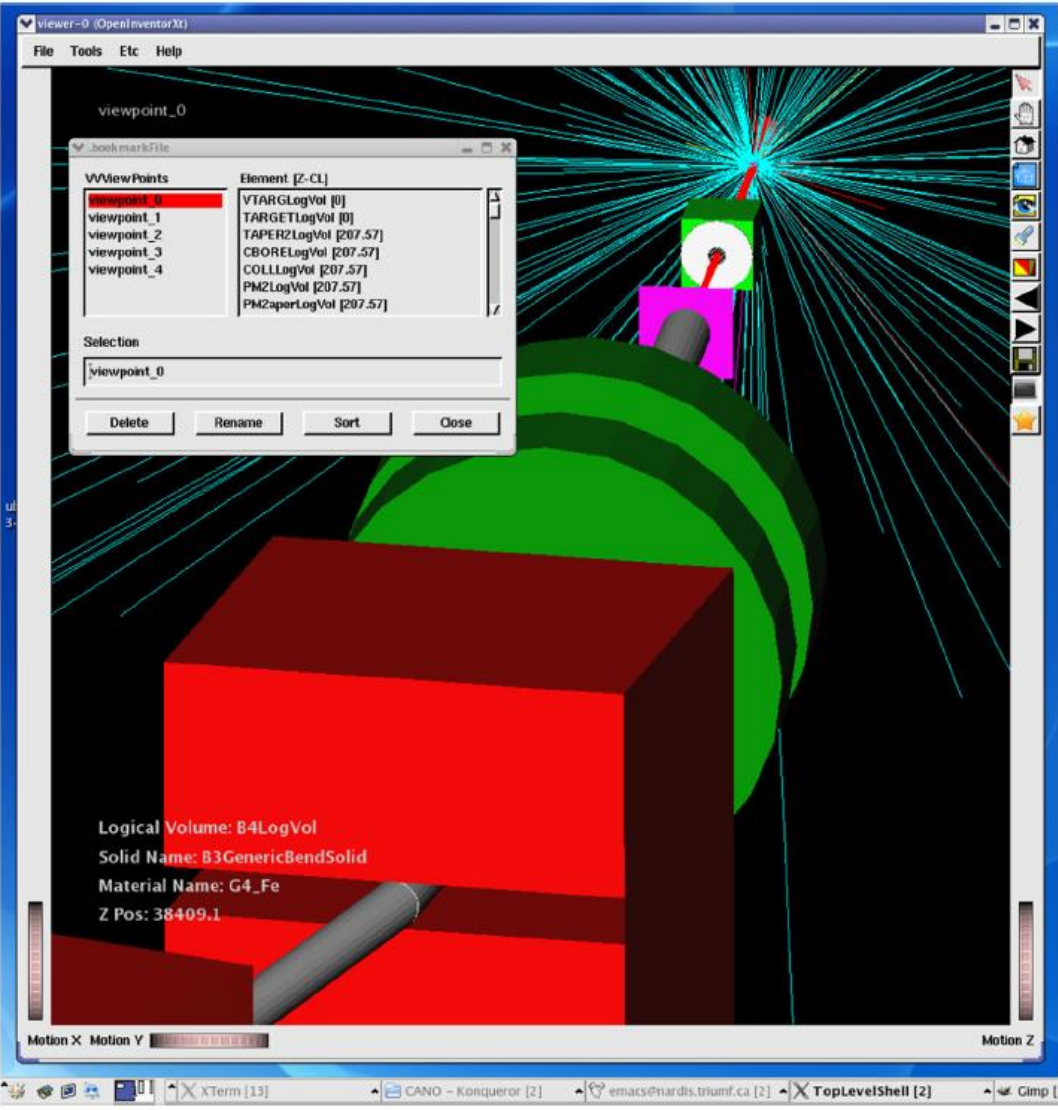
:VOLU sphere ORB 5. G4_AIR
:PLACE sphere 1 "my tube" R00 0. 1. 10.
```



GEANT4: OpenGL Viewer Wrapped in Qt



GEANT4: Open Inventor Extended Viewer



MARS15 (mars.fnal.gov)

- MARS15 is a set of **Fortran 77** and **C++** programs for Monte-Carlo simulations of coupled hadronic and electromagnetic cascades, with heavy ion, muon and neutrino production and interaction. It covers a wide energy range: 1 keV to 100 TeV for muons, hadrons, heavy ions and electromagnetic showers; and 10^{-5} eV to 100 TeV for neutrons.
- Nuclear interactions as well as practically all other strong, weak and electromagnetic interactions in the entire energy range can be simulated either inclusively or exclusively – i.e, in a biased mode or in a fully or partially analogue mode.
- Nuclide production, decay, transmutation and calculation of the activity distribution is done with the built-in DeTra code.
- MARS15 uses ENDF/B-VIII.0(2018) nuclear data to handle interactions of neutrons with energies below 14 MeV and derive the NRT/Stoller/Nordlund DPA x-sections below 200 MeV. The elemental distributions are automatically unpacked into isotope distributions for both user-defined and those from the 172 built-in materials.

MARS15: Other Features

- A tagging module allows one to tag the origin of a given signal for source term or sensitivity analyses. Several variance reduction techniques, such as weight windows, particle splitting, and Russian roulette are possible.
- Six ways to describe geometry are offered, with a basic solid body representation and a ROOT-based powerful engine among them.
- ***The powerful capabilities of MARS15 for simulation in accelerator environment*** with the MARS-MAD Beamline Builder (**MMBLB**) working in concert with an accelerator tracking code (since almost 20 years ago) and with a recent active merge with MADX-PTC for a convenient creation of accelerator models and multi-turn tracking and cascade simulation in accelerator and beamline lattices.
- MARS15 is routinely used in concert with ANSYS for iterative studies of thermo-mechanical problems and can be interfaced to a hydrodynamic code to study phase transition and “**hydrodynamic tunneling**” – first done at SSC for a 20-TeV proton beam in 1993.

Geometry Description and ROOT-based Beamline Builder

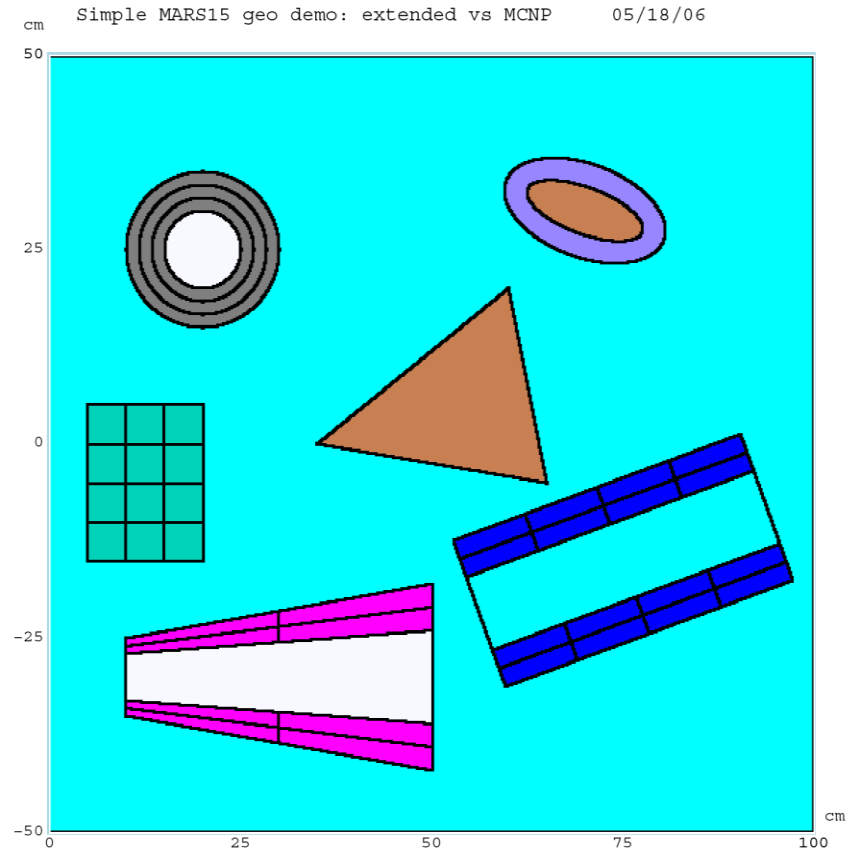
1. User-generated via MARS extended geometry input files
2. User-generated ROOT files
3. GDML files (two-way exchange with Geant4 teams)
4. G4beamline's BruitDeFond can generate MARS's input files MARS.INP, GEOM.INP and FIELD.INP
5. STEP files from project CAD models used to generate ROOT geometry modules
6. Lattice and beamline components such as dipole and quadrupole magnets, correctors, accelerating cavities, cryomodules and tunnel with all the details available on geometry, materials and electromagnetic fields by means of the advanced ROOT-based Beamline Builder

MARS15 Simple GEOM.INP Example

Extended Demo 05/17/06

OPT

```
box-1 1 0 2 0. -5. 5. 10. 10. 15. 1 4 3
cyl-1a -2 1 7 0. 0. 0. 0. 5. 20.
cyl-1a -2 1 1 0. 0. 0. 0. 5. 10. 20. 4 2
ball-a 3 0 8 0. 25. 20. 0. 5.
ball-b 3 0 3 0. 25. 20. 5. 10. 3
cone-in -4 0 0 0. -30. 30. 0. 3. 0. 6. 20.
cone-out -4 0 4 0. -30. 30. 3. 5. 6. 12. 20. 2 2
th 5 0 6 0. 0. 35. 5. 3. 55. 0. 20. 60. 0. -5. 65.
ell-tub1 -6 2 6 0. 0. 0. 0. 8. 3. 0. 40.
ell-tub2 -6 2 5 0. 0. 0. 0. 8. 3. 3. 40.
TR1 0. -15. 75. -20.
TR2 0. 30. 70. 20. 90.
stop
```



Aspect Ratio: Y:Z = 1:1.0

Comparing MARS and MCNP Geo Descriptions

Extended & MCNP Demo 05/17/06
 /home/mokhov/restricted/mars15/dat
 INDX 3=F 5=T 16=T

CTRL 1
 NEVT 50
 ENRG 10.
 ZSEC 100.
 RSEC 50. 101=7
 NMAT 7
 MATR 'NBS2' 'SCI' 'CONC' 'MRBL' 'CAST' 'S316' 'AIR'
 NHBK 1
 STOP

*MCNP START

```

1 7 -0.00129 2 -1 (-26:27:25) 8 17 (-19:21:23) imp:n,p=1
2 2 -0.7903 -2 -3 -6 imp:n,p=1
3 2 -0.7903 -2 3 -4 -6 imp:n,p=1
4 2 -0.7903 -2 4 -5 -6 imp:n,p=1
5 2 -0.7903 -2 5 -6 -6 imp:n,p=1
6 2 -0.7903 -2 -3 6 -7 imp:n,p=1
7 2 -0.7903 -2 3 -4 6 -7 imp:n,p=1
8 2 -0.7903 -2 4 -5 6 -7 imp:n,p=1
9 2 -0.7903 -2 5 6 -7 imp:n,p=1
10 2 -0.7903 -2 -3 7 imp:n,p=1
11 2 -0.7903 -2 3 -4 7 imp:n,p=1
12 2 -0.7903 -2 4 -5 7 imp:n,p=1
13 2 -0.7903 -2 5 7 imp:n,p=1
14 7 -0.00129 -8 -9 imp:n,p=1
15 1 -7.0 -8 9 -10 -11 imp:n,p=1
16 1 -7.0 -8 9 -10 11 -12 imp:n,p=1
17 1 -7.0 -8 9 -10 12 -13 imp:n,p=1
18 1 -7.0 -8 9 -10 13 imp:n,p=1
19 1 -7.0 -8 10 -11 imp:n,p=1
20 1 -7.0 -8 10 11 -12 imp:n,p=1
21 1 -7.0 -8 10 12 -13 imp:n,p=1
22 1 -7.0 -8 10 13 imp:n,p=1
23 0 -14 imp:n,p=1
24 3 -2.35 14 -15 imp:n,p=1
25 3 -2.35 15 -16 imp:n,p=1
26 3 -2.35 16 -17 imp:n,p=1
27 0 -18 19 -21 imp:n,p=1
28 4 -2.7 18 -22 19 -20 imp:n,p=1
29 4 -2.7 18 -22 20 -21 imp:n,p=1
30 4 -2.7 22 -23 19 -20 imp:n,p=1
  
```

```

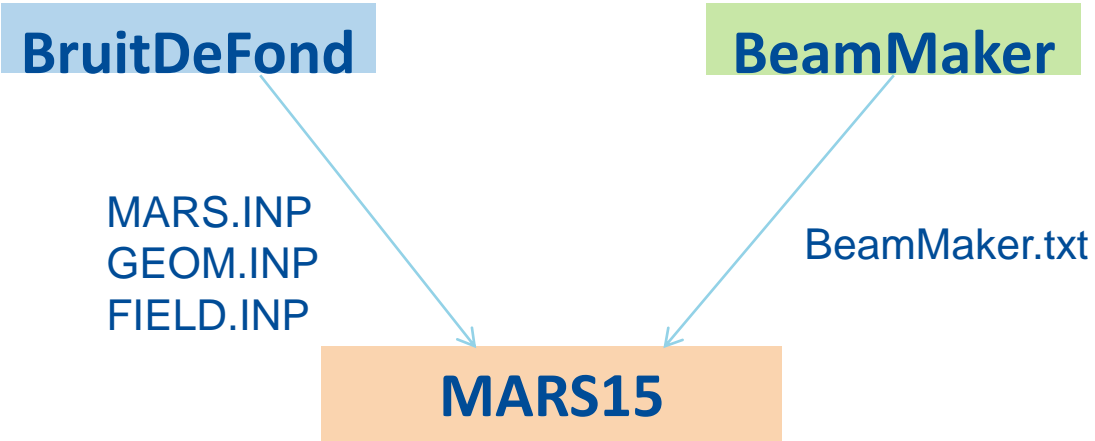
31 4 -2.7 22 -23 20 -21 imp:n,p=1
32 6 -7.92 -24 26 -27 imp:n,p=1
33 5 -7.31 24 -25 26 -27 imp:n,p=1
34 0 1 imp:n,p=0
1 rcc 0 0 0 0 0 100 50
2 rpp -10 10 -15 5 5 20
3 py -10
4 py -5
5 py 0
6 pz 10
7 pz 15
8 1 rcc 0 0 -20 0 0 40 10
9 1 cz 5
10 1 cz 7.5
11 1 pz -10
12 1 pz 0
13 1 pz 10
14 s 0 25 20 5
15 s 0 25 20 6.667
16 s 0 25 20 8.333
17 s 0 25 20 10
18 k/z 0 -30 -30 .005625
19 pz 10
20 pz 30
21 pz 50
22 k/z 0 -30 -22 .015625
23 k/z 0 -30 -18.57 .030626
24 2 sq 0 64 9 0 0 0 -576 0 0 0
25 2 sq 0 121 36 0 0 0 -4356 0 0 0
26 px -40
27 px 40
mode n p
  
```

```

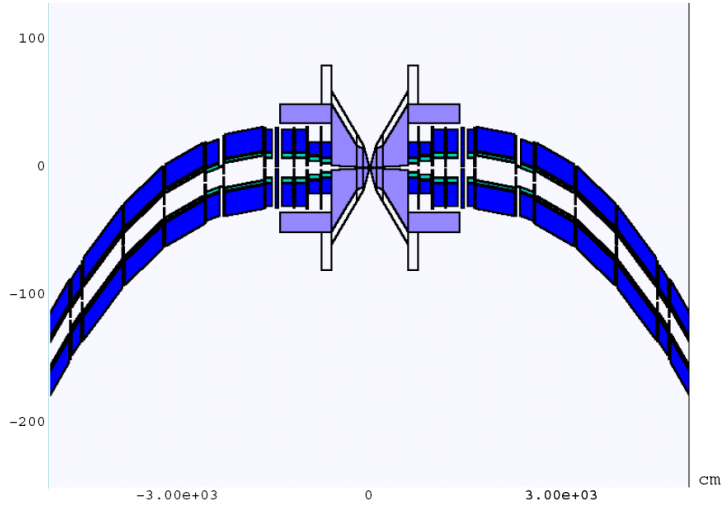
m1 2004 -.02 29000 -.38 13027 -.2 41093 -.280522 50000 -.119478 $ NBS2
m2 1001 -.13314 6000 -.86651 7014 -.00016 8016 -.00019 $ SCI
m3 1001 -.006 6000 -.030 8016 -.500 11023 -.010 13027 -.003 & $ Conc
    14000 -.200 19000 -.010 20000 -.200 26000.42c -.014
m4 20000 -.400431 6000 -.120005 8016 -.479564 $ MRBL
m5 6000 -.0365 14000 -.025 25055 -.0018 26000.42c -.9347 29000 -.002 $ CAST
m6 24000 -.17 25055 -.02 26000.42c -.655 28000 -.12 14000 -.01 42000 -.025 $ S316
m7 7014 .78443 8016 .21076 18000.42c 4.671E-3 6000 1.39E-4 gas=1 $ Air
vol 1 33r
*MCNP END
  
```



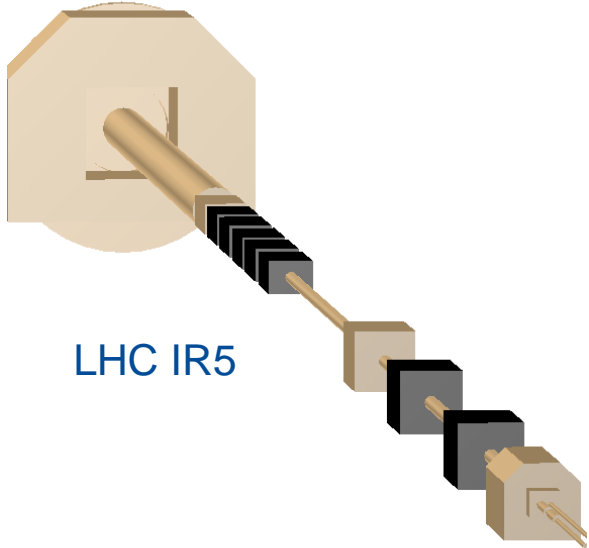
G4beamline's BruitDeFond Can Generate MARS's Input Files



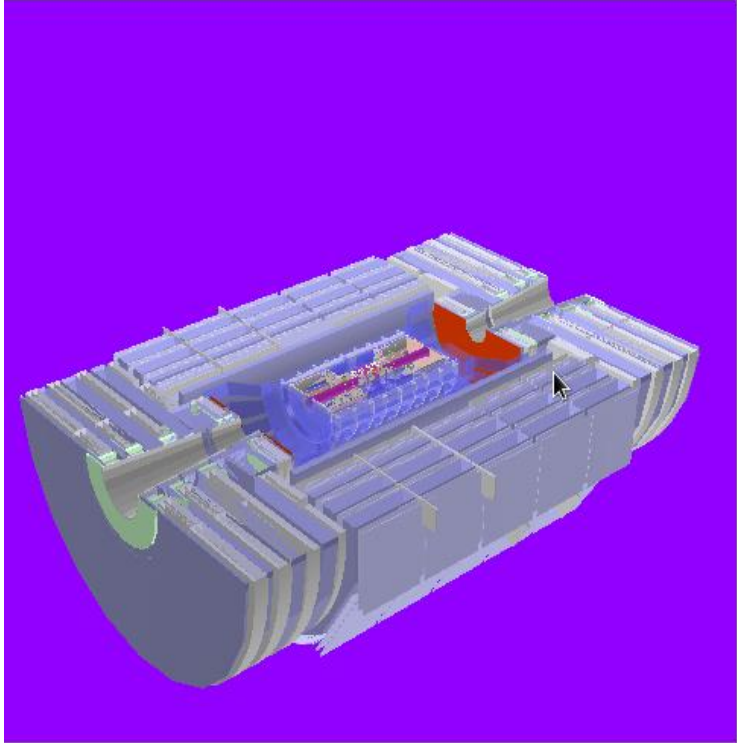
- *BruitDeFond* can generate MARS.INP and GEOM.INP files.
- The *extended* geometry description is used.
- The field is described by the FIELD.INP file which is read by the MARS user subroutine "field" that we wrote.
- We use the same BeamMaker.txt file for the MARS input.



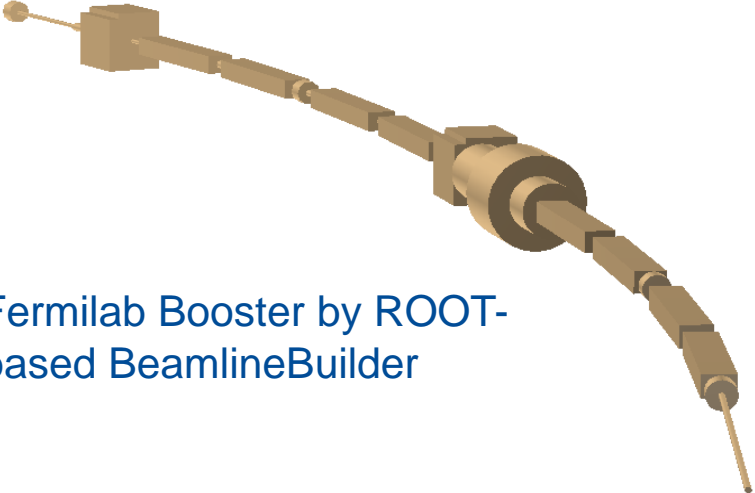
MARS15 Models



LHC IR5



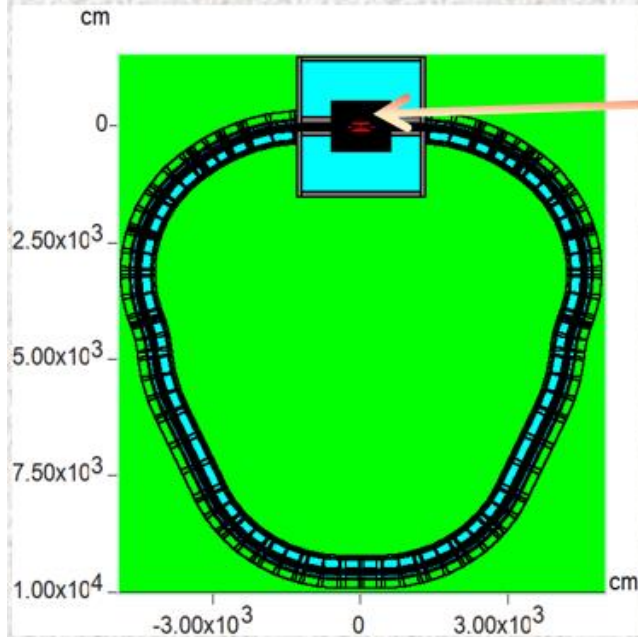
LHC CMS detector



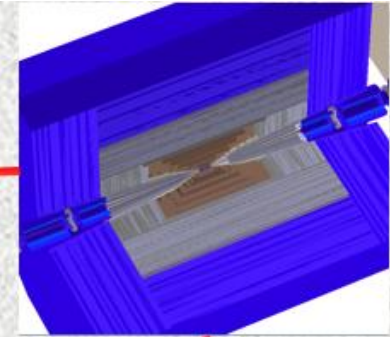
Fermilab Booster by ROOT-based BeamlineBuilder

MARS15 Model of the Higgs Factory Muon Collider

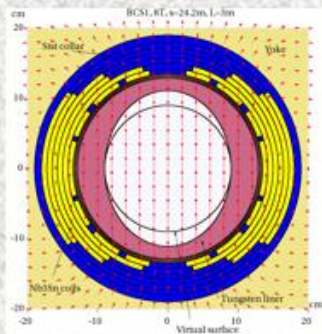
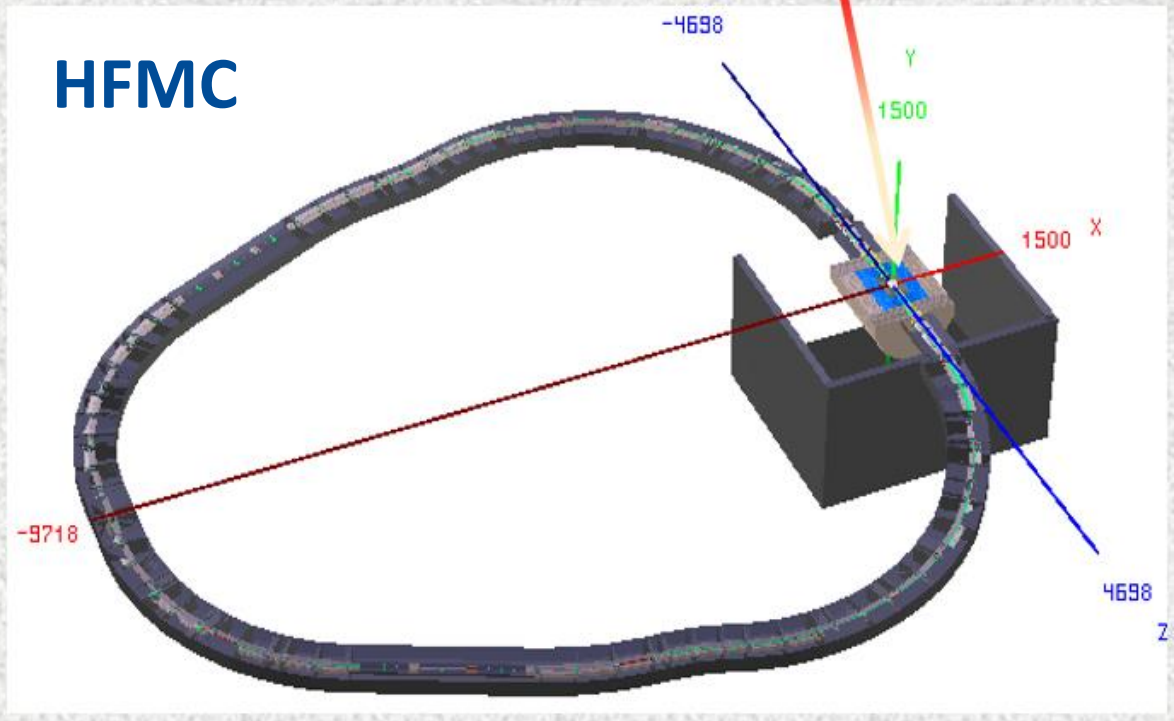
300-m $\mu^+\mu^-$ Collider



SiD-like detector with CMS upgrade-type tracker

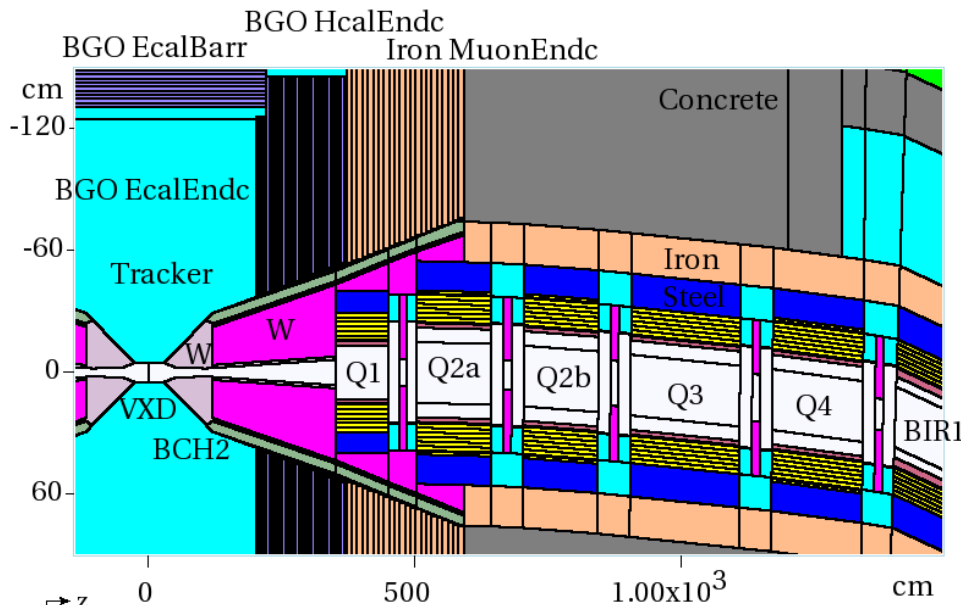


HFMC

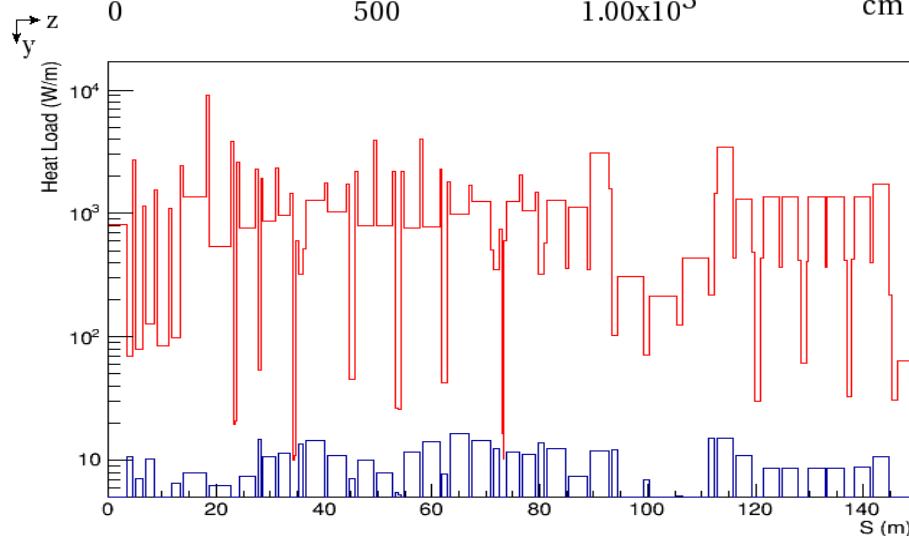


8-T
SC
dipole

HFMC: Protection of SC magnets and Detector

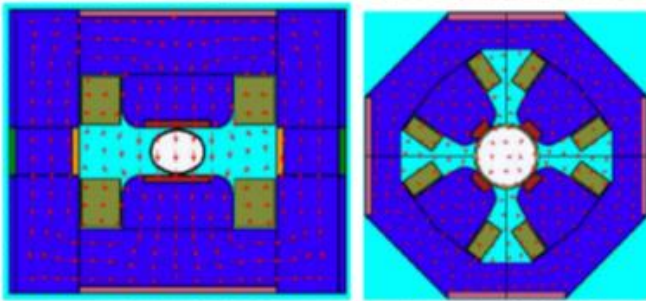
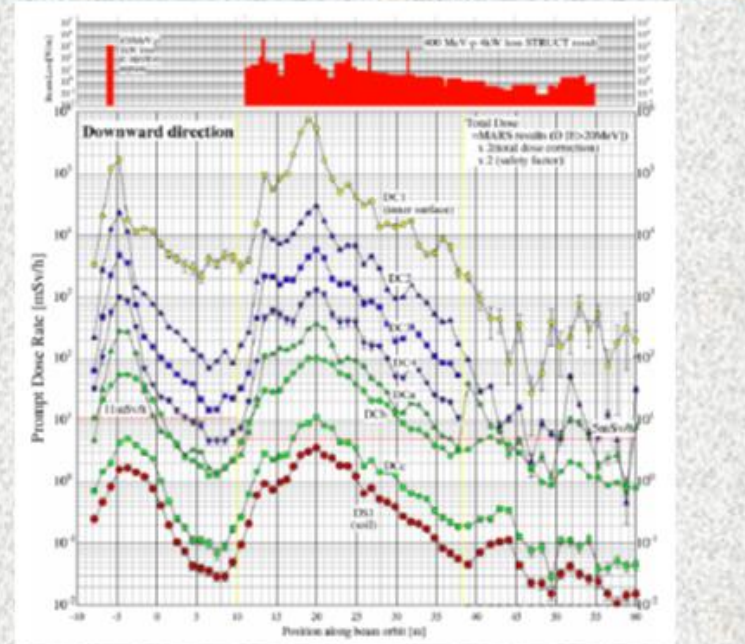
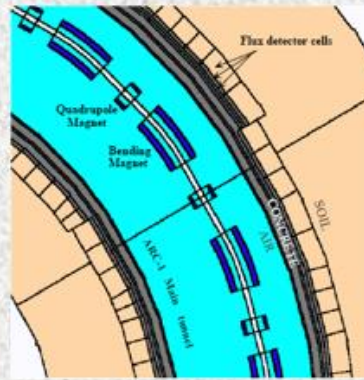
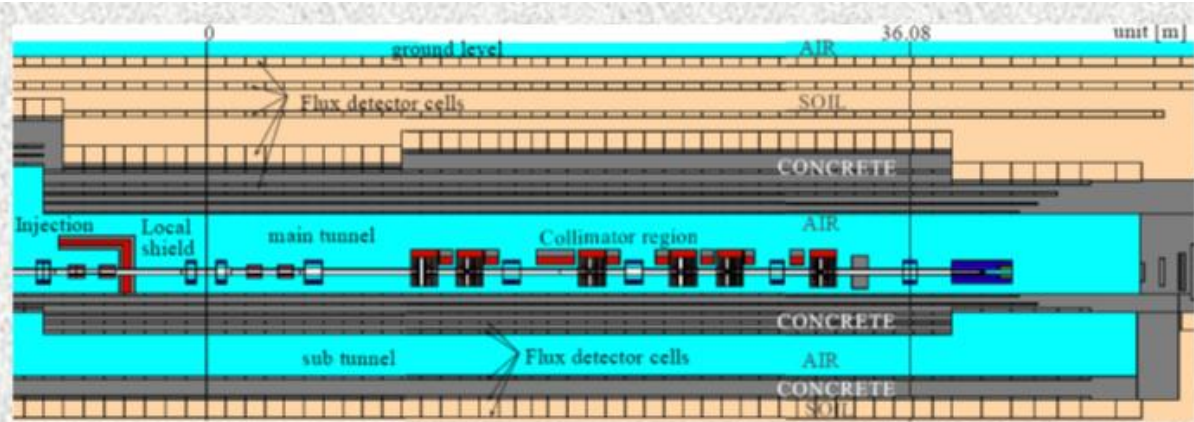
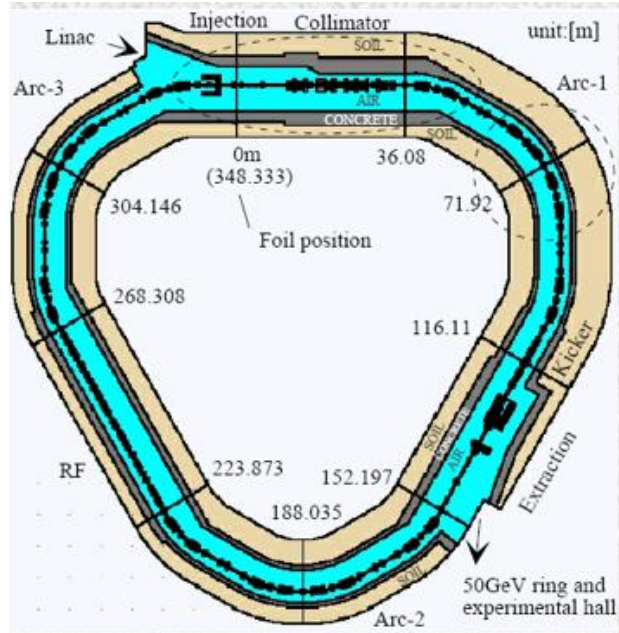


MARS15 model of the HFMC Machine-Detector Interface with a sophisticated tungsten nozzle near IP, collimators in interconnect regions and liners inside SC magnets – all resulted in reduction of detector backgrounds to the design limits.



Collimators in interconnect regions and liners inside magnets in the ring have resulted in x100 reduction of dynamic heat loads in the magnet superconducting coils.

MMBLB Model: J-PARC 3-GeV Ring

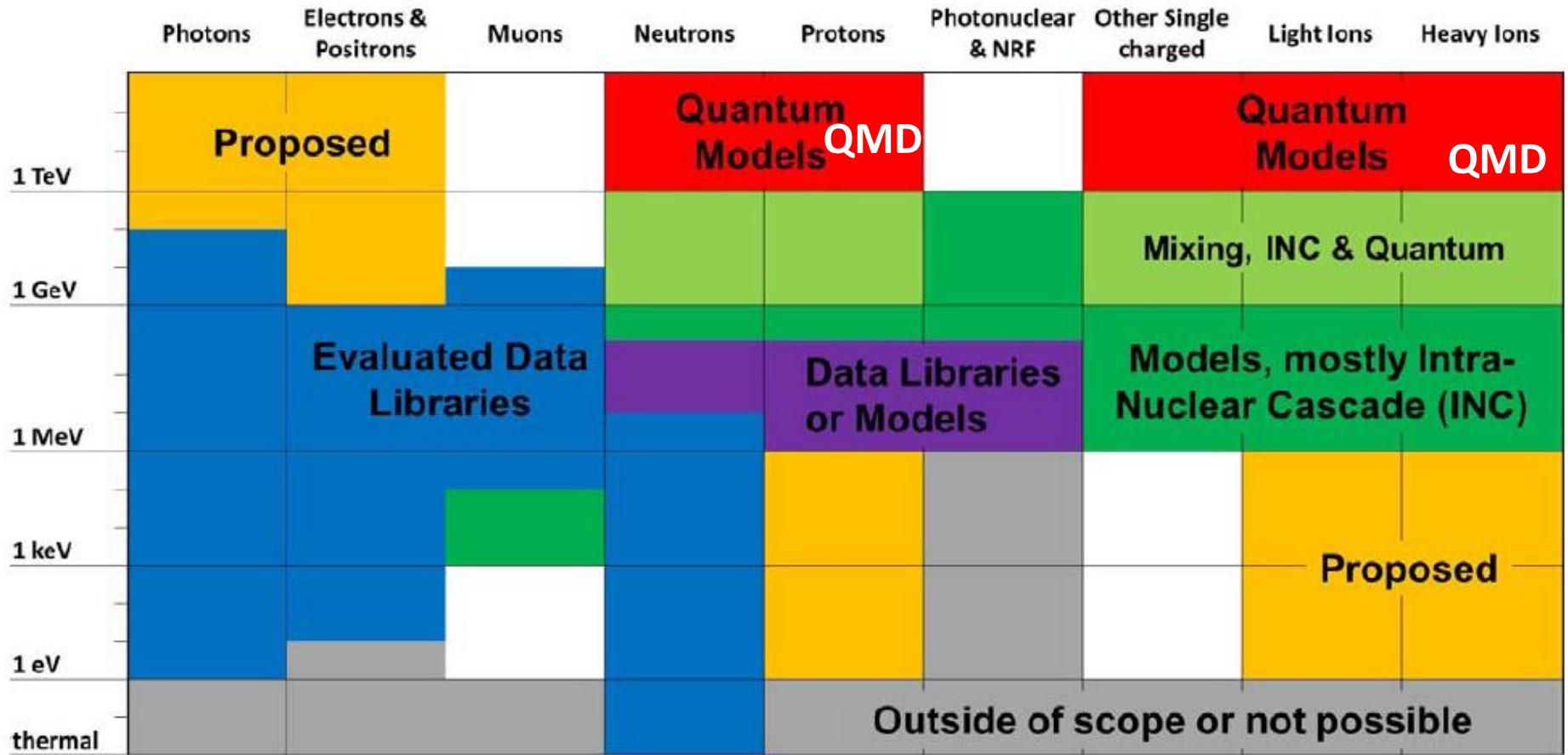


MCNP6 (mcnp.lanl.gov)

MCNP6 is the latest version of the Monte Carlo N-Particle transport (MCNP) family of particle interaction and transport codes (**Fortran-90**) and features comprehensive and detailed descriptions of the related physical processes. It transports 37 different particle types, including ions and electromagnetic particles. The neutron interaction and transport modules use standard evaluated data libraries mixed with physics models where such libraries are not available. ***Considered by many as the industry standard for simulation in reactor, medical, space and low- and medium-energy accelerator applications.***

The transport is continuous in energy. MCNP6 contains one of the most powerful implementations of variance reduction techniques. Spherical mesh weight windows can be created by a generator in order to focus the simulation time on certain spatial regions of interest. In addition, a more generalized phase space biasing is also possible through energy- and time-dependent weight windows. Other biasing options include pulse-height tallies with variance reduction and criticality source convergence acceleration.

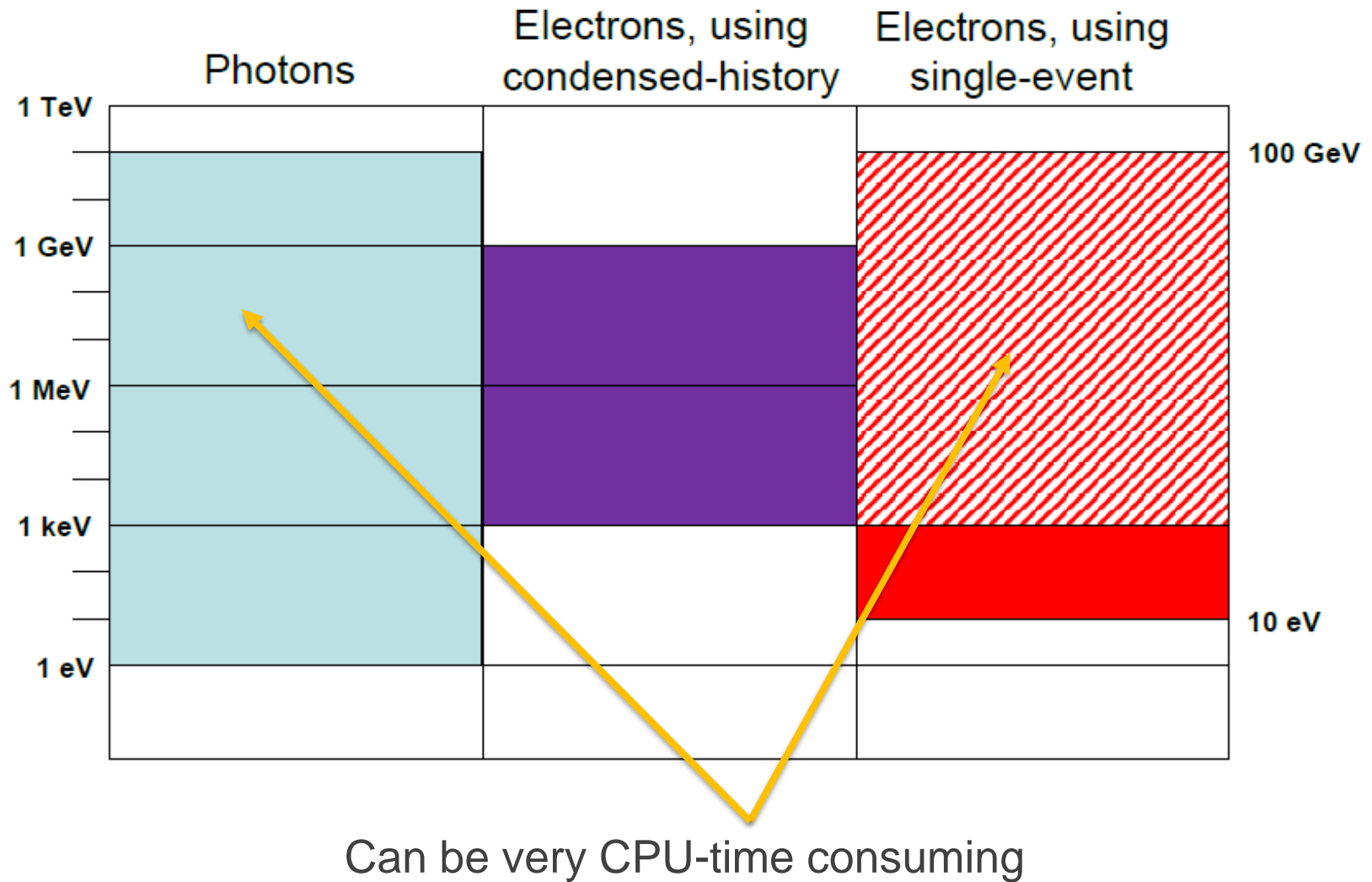
MCNP6 Models



MCNP6 Geometry, B-Field and Tests

- Geometry options: traditional surface-based, voxel lattice, constructive solid and unstructured mesh
- Magnetic fields: (1) Constant dipole, square-edge quadrupole and quadrupole with a fringe-field kick – all in low-density materials, such as air; (2) COSY maps only in vacuum and specific to one particle type; both are rather limited compared to four other codes with the arbitrary EM field capability in arbitrary geometry/materials
- **Unique feature:** MCNP6 is considered risk level two software (death is risk level one), i.e. is treated as if failure of the software could result in temporary injury or illness to workers or the public. Therefore, a set of hundreds automated verification, validation and regression tests. Latter is detecting unintended changes to the code and installation testing.
- Super-precise simulation of EMS at 1 eV to 100 GeV

MCNP6: Super-Precise Simulation of EMS

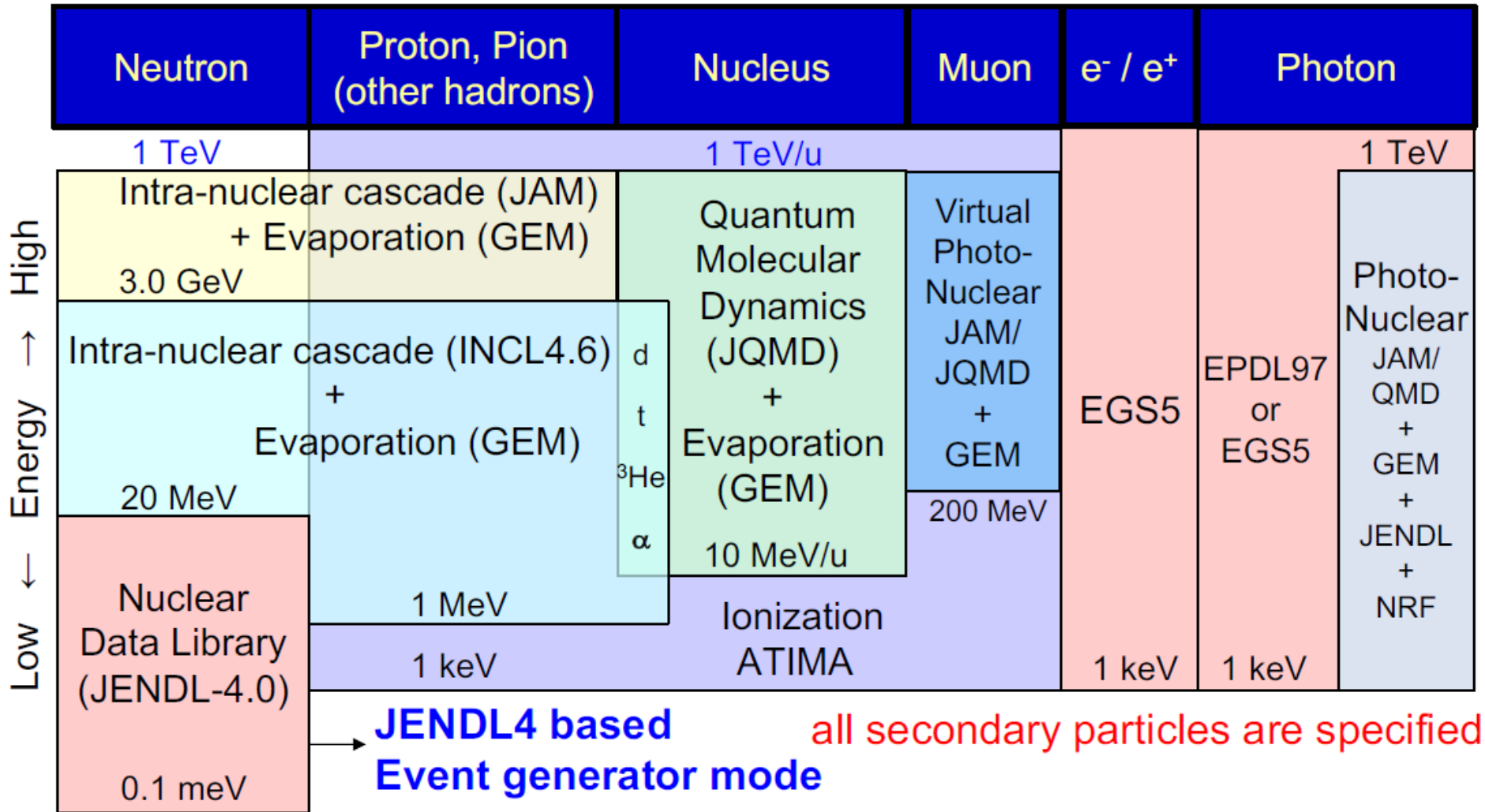


PHITS (phits.jaea.go.jp)

PHITS is the Particle and Heavy-Ion Transport code System (**Fortran 77**). *It was among the first general-purpose codes to simulate the transport and interactions of heavy ions in a wide energy range, from 10 MeV/nucleon to 100 GeV/nucleon.* It is based on the high-energy hadron transport code NMTC/JAM that was extended to heavy ions.

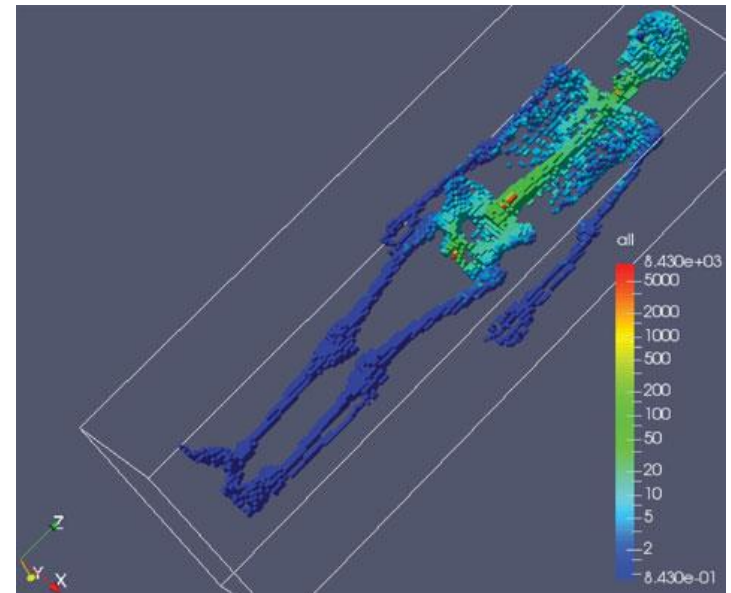
The transport of low-energy neutrons employs cross sections from evaluated nuclear data libraries such as ENDF and JENDL below 20 MeV. Electromagnetic interactions are simulated based on the EGS5 code in the energy range between 1 keV and 100 MeV for electrons and positrons and between 1 keV and 100 GeV for photons. Several variance reduction techniques, including weight windows and region importance biasing, are available. An accurate calculation of DPA supported by dedicated experiments with medium-energy protons.

PHITS Models



PHITS Geometry, Fields, Nuclides & Tallying

- The geometrical configuration of a simulation is set with general geometry (GG) in a manner similar to MCNP. The interactive solid modeler Simple-Geo (FLUKA) can be used for generating the geometries written in PHITS-readable GG format.
- Computer-aided Design (CAD)-based geometries can be incorporated into PHITS by converting CAD data into tetrahedral-mesh geometries. In addition, CAD geometries can be directly converted into the PHITS-readable GG format by using SuperMC.
- Electromagnetic fields and gravity can be considered in transport simulation of all particles.
- The time evolution of radioactivity is estimated by built-in DCHAIN-SP module.
- An example of tallying is shown



MARS15-MADX-PTC Integration (1)

In accelerator simulations, one provides – one way or another - a cross-talk between an accelerator tracking code and a code (MARS, FLUKA) for particle-matter interactions

On the accelerator side, we used for more than two decades a home-made STRUCT code working in concert with MARS. Recently, we have successfully switched to MADX-PTC for accelerator tracking

MARS15-MADX-PTC Integration (2)

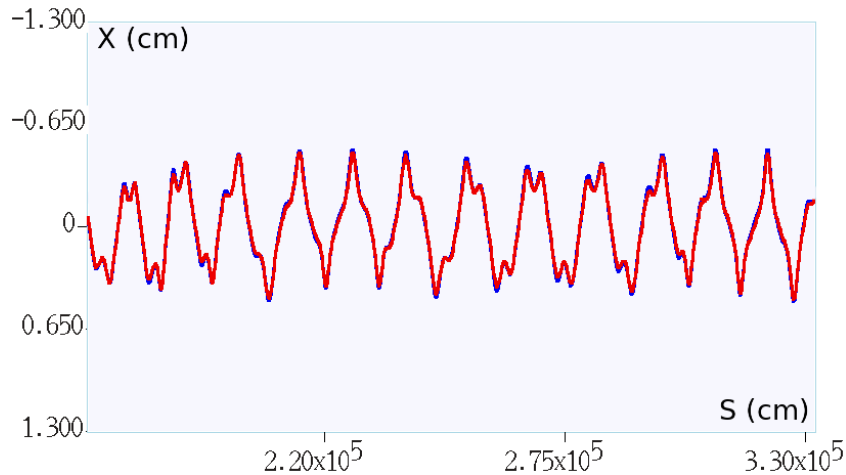
Basic integration outline:

1. A library containing functions and C++ classes which interfaces MARS with MADX is now packed with the MARS15 distribution. The library allows to
 - Create a 3-D TGeo ROOT geometry model for the sequence described in a MADX-PTC input file. Alignment of elements is performed by means of the MAD-X survey table.
 - Define transformation for each point in the phase space used in the PTC module to the phase space used in MARS15 and vice-versa.
 - Inject particles transported by MARS15 to MADX-PTC module using a formulated acceptance for the accelerator code model.
 - For particles transported in PTC, perform check of boundary crossing against the ROOT geometry in MARS15; the particle is forwarded to the MARS15 stack.

MARS15-MADX-PTC Integration (3)

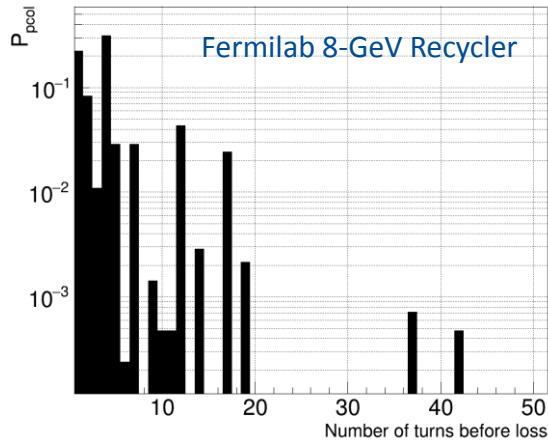
2. The original PTC code was modified in order to

- Allow a particle to start from the upstream end of an arbitrary element in the sequence. Originally, it always started from $S=0$.
- Check the aperture crossing against the MARS15 geometry not only at the entrance and exit of the element, but also all along a curved track (e.g., in dipoles).

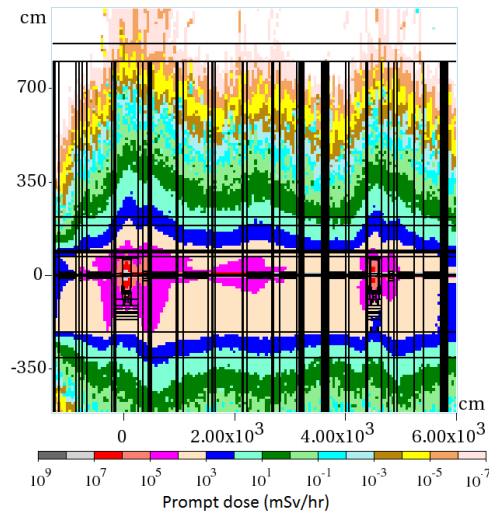


Beam orbits in the Fermilab 8-GeV Recycler calculated with MAD-X PTC module (blue) and MARS15 native stepper (red)

Examples: Fermilab Recycler and ILC



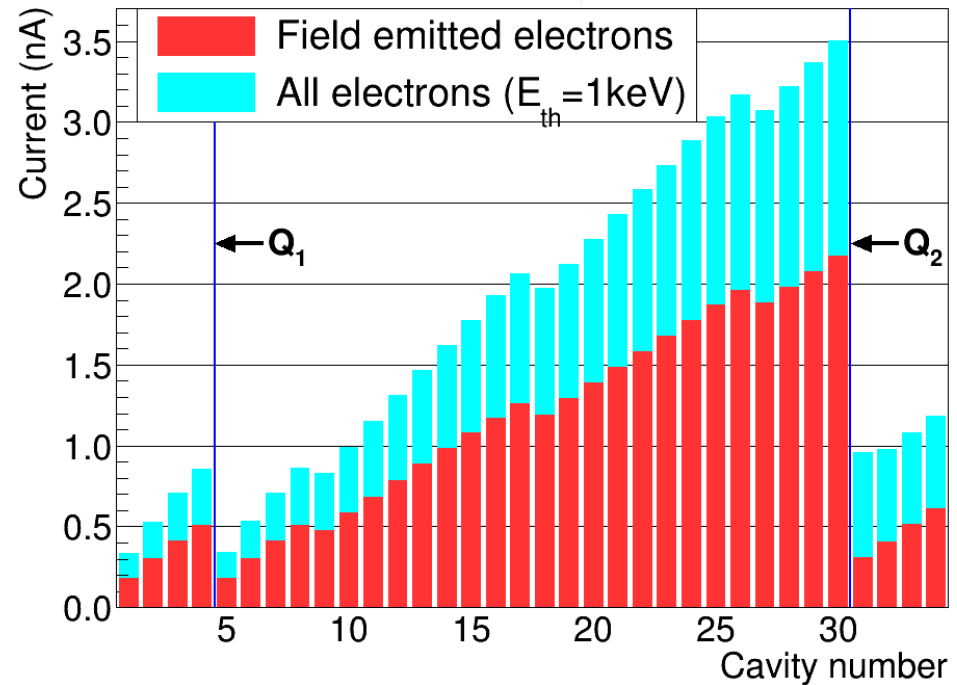
Probability to be lost for beam halo protons passed through the primary collimator vs #turns



Prompt dose in collimation region



MARS15 SRF model



Dark current electrons and EMS electrons in ILC aperture with their loss responsible for radiation load to components and radiation field in ILC tunnel

MARS15 Platform, Compilers, ROOT and MPI

- Linux \geq gcc-4.8 on 64 bit architectures; our compiler policy: use the latest stable release, e.g., currently, gcc-8.2, which is ideal for modern c++ codes
- C++11 standard
- gsl-2.4 or 2.5
- ROOT-6.14
- ISO standard Fortran \Leftrightarrow C interface
- Many-core jobs (standard for decade); 10^2 to 10^5 cores routinely; improved submission scripts and built-in averaging
- Genuine MPI mode in MARS15(2018) is now used more and more often; to eliminate discovered scalability bottleneck, common physics data (x-sections etc.) are accessed via a shared memory window (MPI-3 feature)

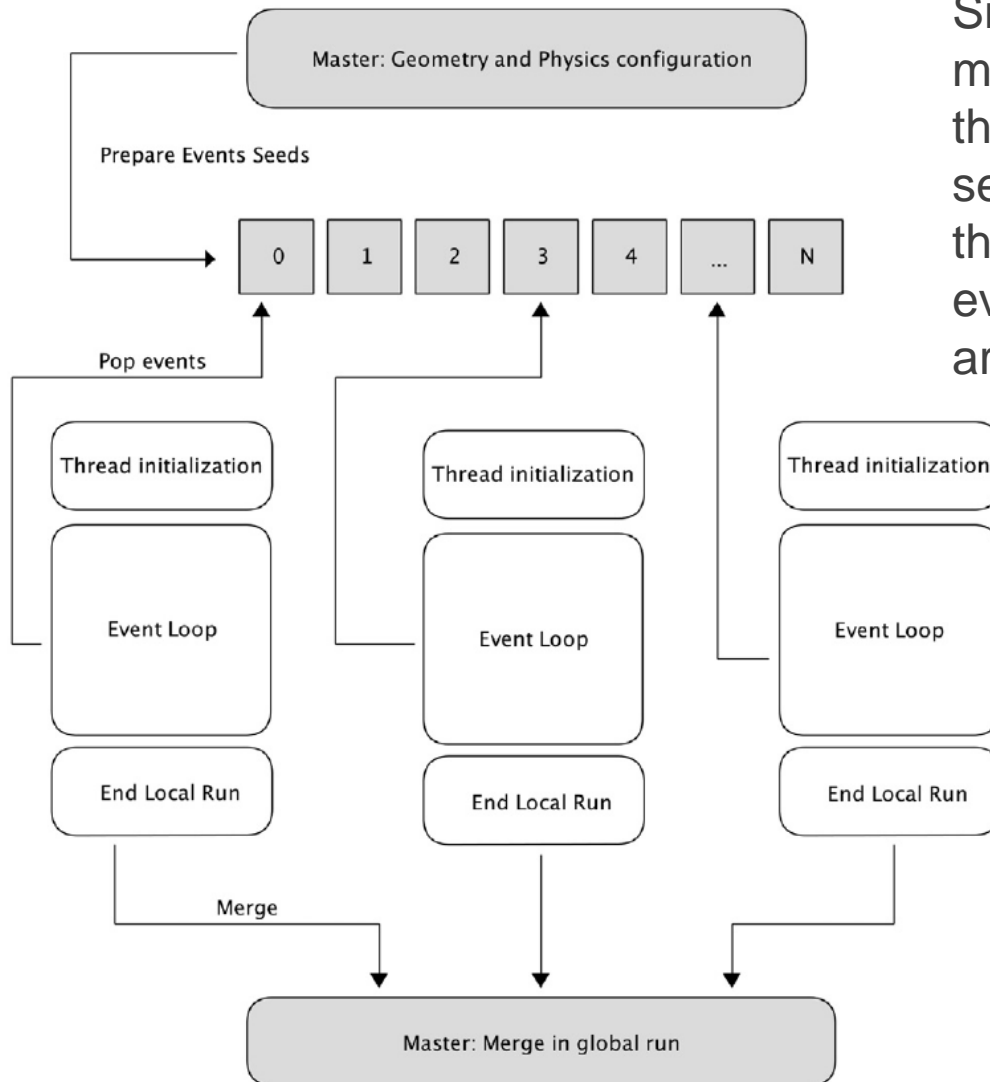
Multithreading and Multi-Core Approaches

Multithreading is usually found in multitasking operating system. Multithreading is a widespread programming and execution model that allows multiple threads to exist within the context of one process. These threads share the process's resources, but are able to execute independently

Nowadays, multithreading is a common techniques used by particle-matter interaction codes in CPU-hungry applications. It is applied to one process to enable parallel execution on a multiprocessing system

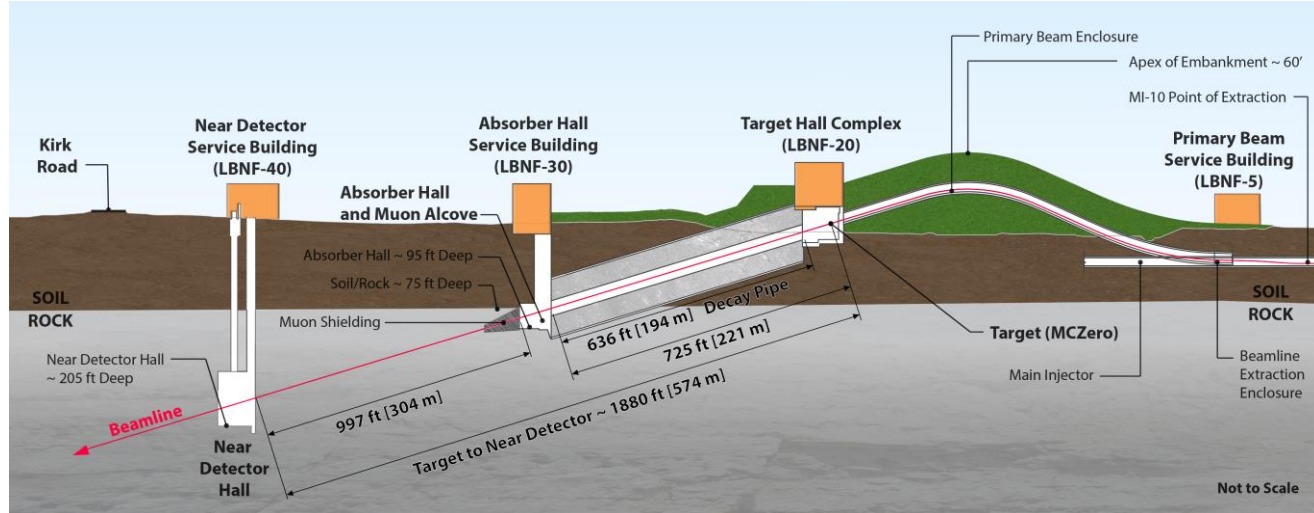
Multithreading is a user-friendly alternative to a **multiple-core failproof approach** (used, for example, in MARS for decades) with thousands independent jobs on a cluster submitted with a user-created script and with results averaged at a post-processing stage

Flow Chart of Multithreading

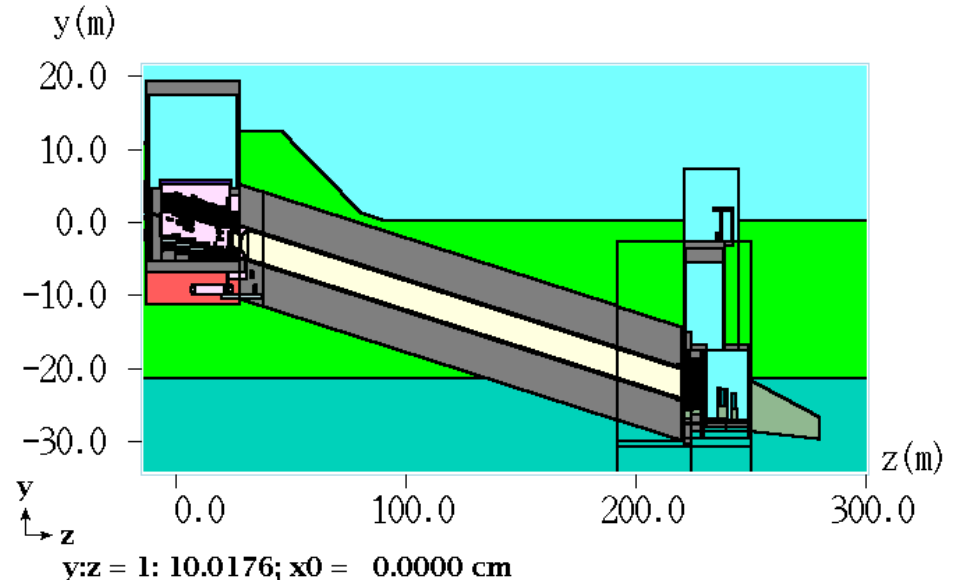


Simplified description of a GEANT4 multithreaded application: the master thread prepares geometry and physics setups for the simulation, and the worker threads compete for the next (group of) events to be simulated; otherwise they are independent.

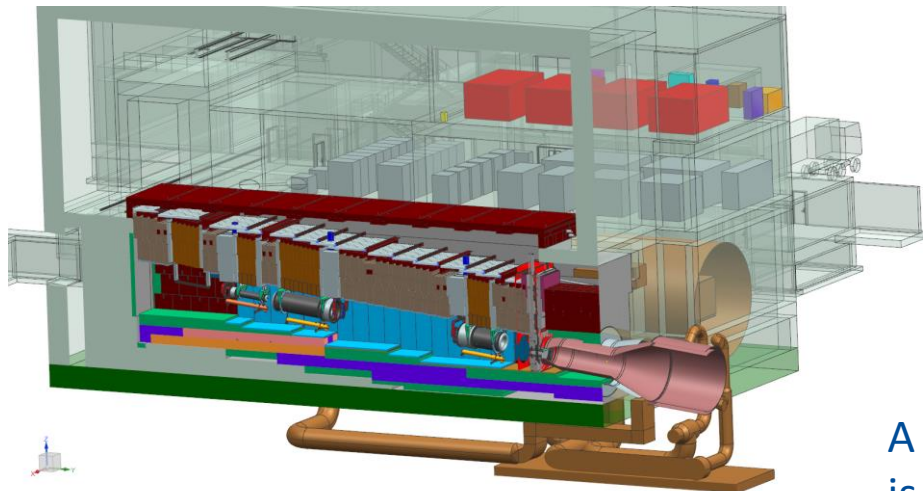
LBNF Beamline: Schematic & MARS Model



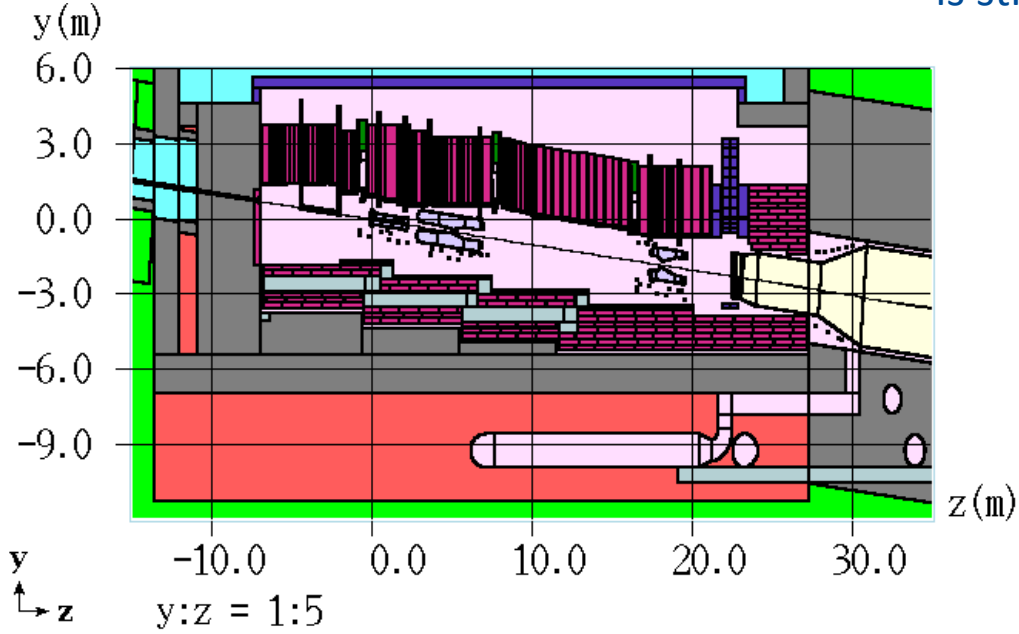
- LBNF and DUNE
- Neutrinos from 60 to 120 GeV proton beam
- **1.2 MW from day one**; upgradeable to 2.4 MW
- **Near detector** to characterize the beam
- Massive underground LAr TP Chambers
- **4 x 17 kton** (fiducial mass of more than 40 kton)



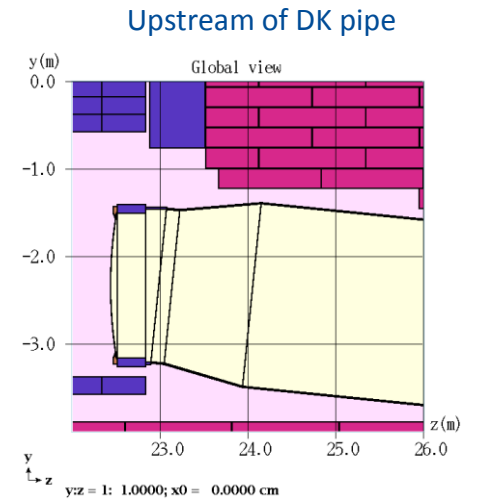
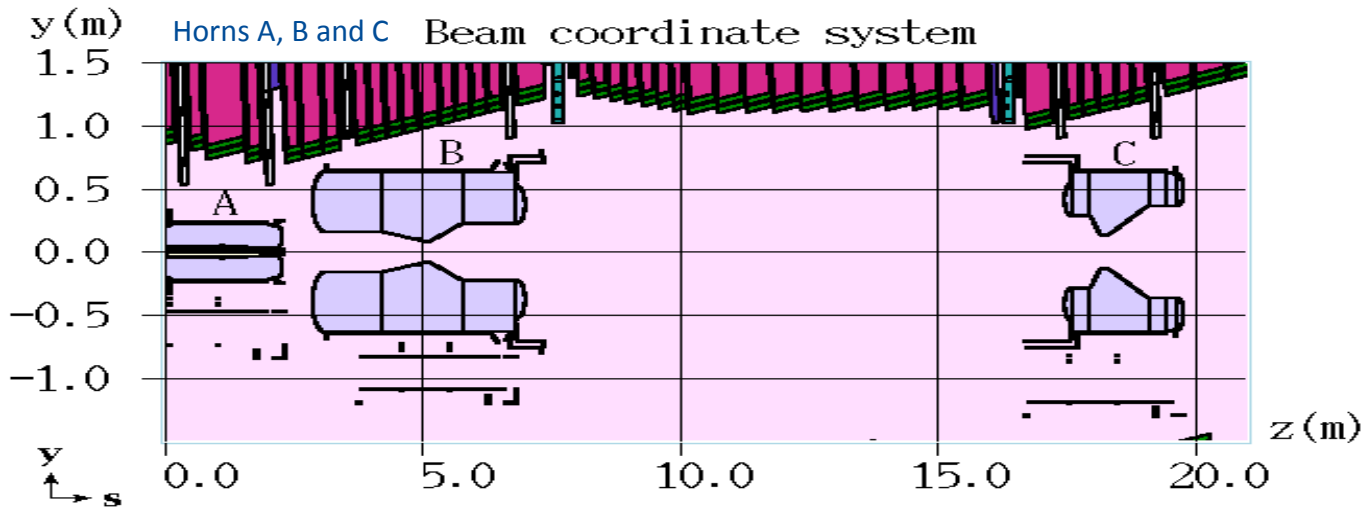
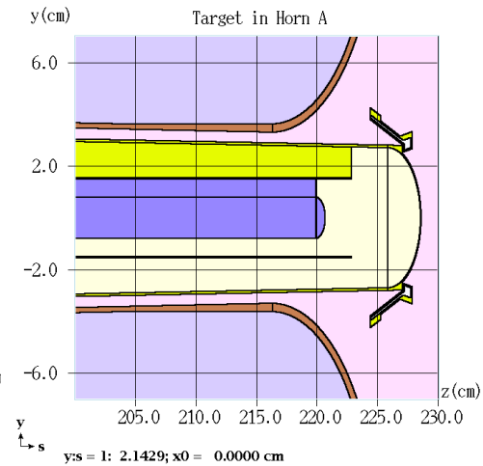
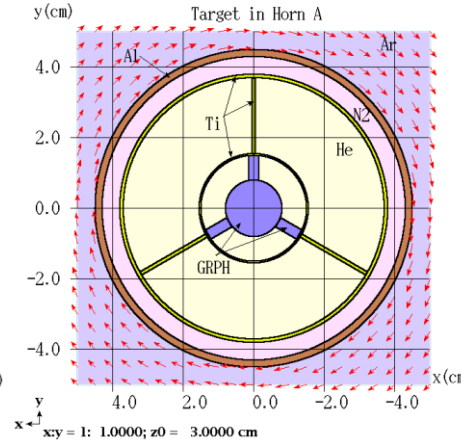
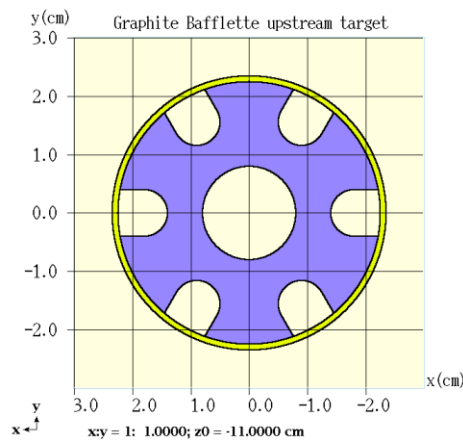
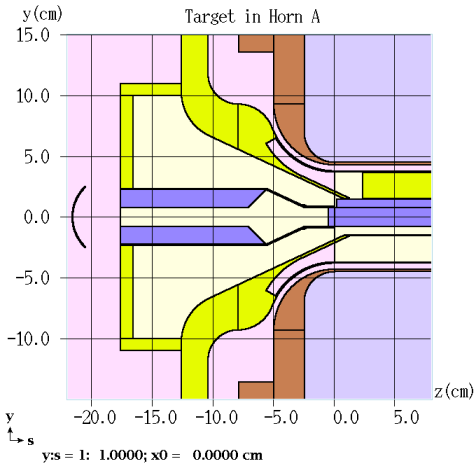
Target Station (TS): CAD & MARS15 Models



A replaceable target design is still in a preliminary stage

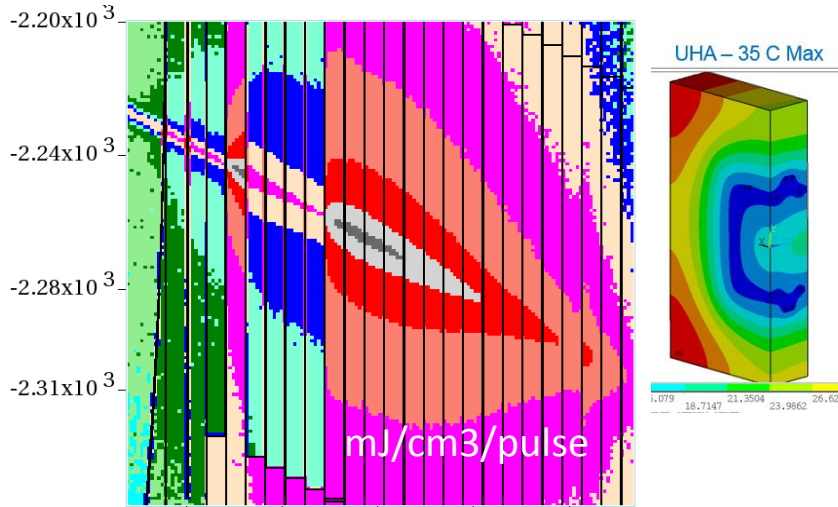


Details of the LBNF-MARS TS Model



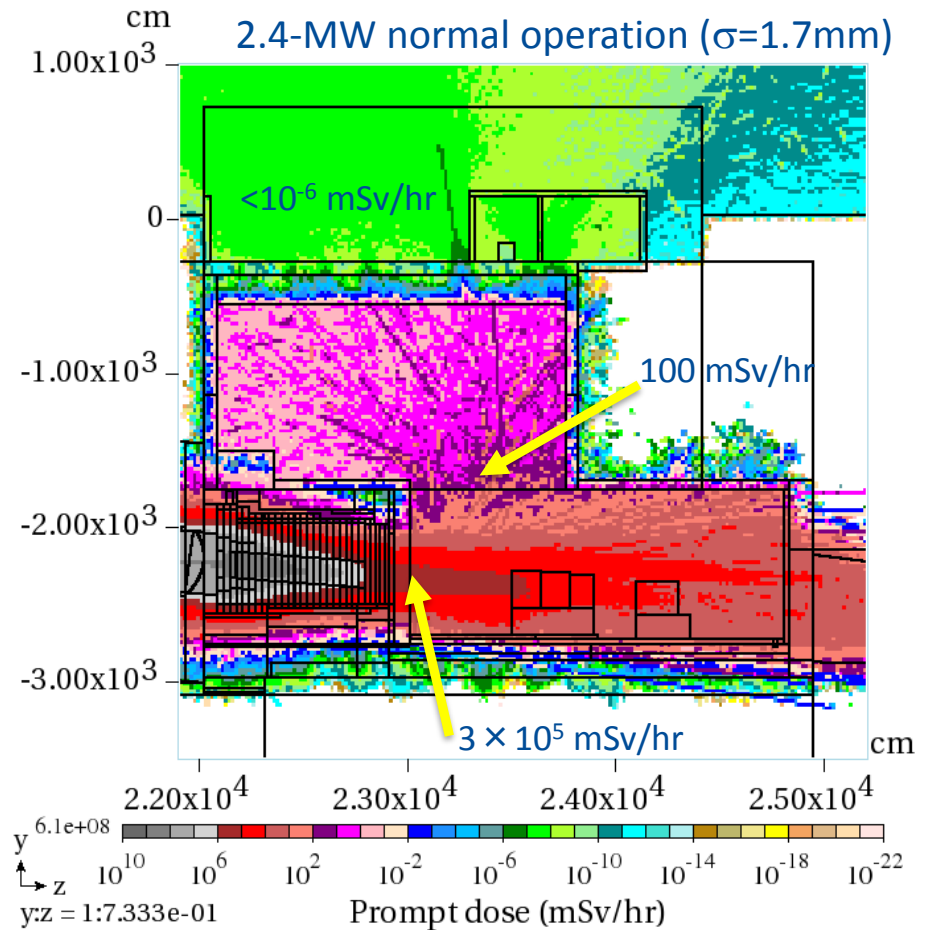
Hadron Absorber Complex: EDEP & Prompt Dose

No-target beam accident ($\sigma=2.4\text{mm}$)



Temperature and stresses below the limits

2.4-MW normal operation ($\sigma=1.7\text{mm}$)



Carefully designed shielding guarantees prompt and residual dose, air and cooling water activation as well as radiation load on ground water to be below the administrative limits with corresponding safety margins

Neutrino Fluxes at Far Detector (1300 km)

Thorough search and elimination of differences in MARS15LBNF and G4LBNF models were performed on the optimized ν -flux: geometry, materials, magnetic fields etc.

ν -fluxes at the Far Detector calculated with MARS15 and Geant4 now agree within 10%. The code related uncertainties were reduced to the differences in the event generators, especially for K^- and K^0 mesons (need data!)

