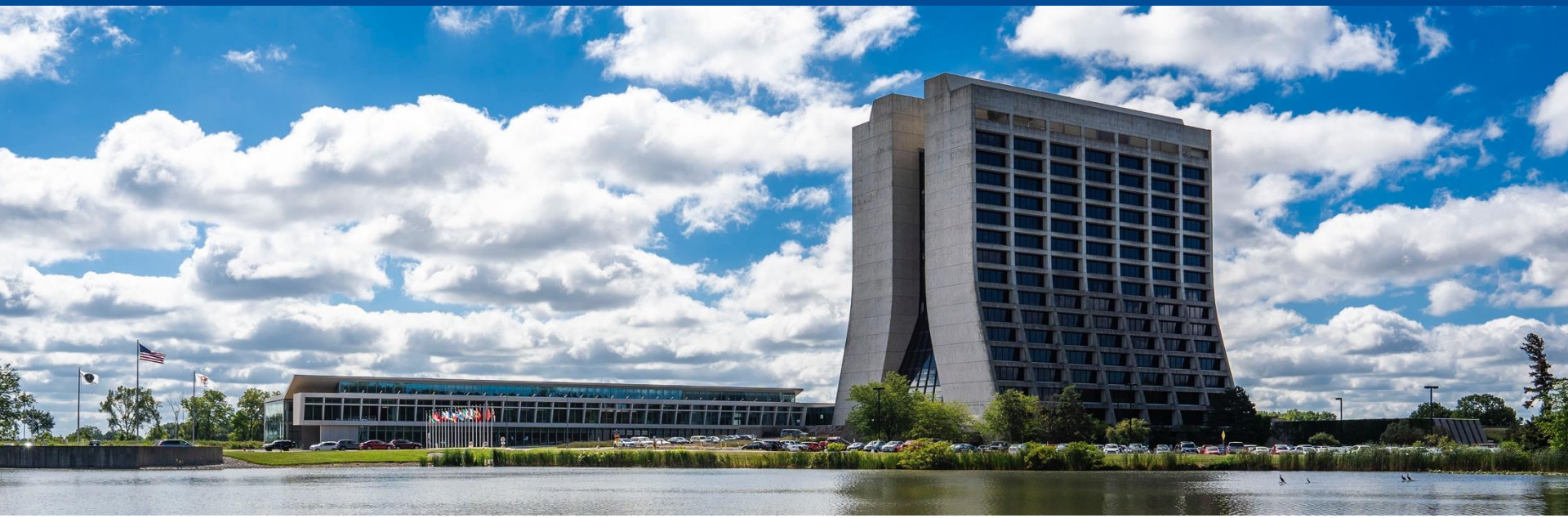


Fermilab-SLIDES-24-0117-AD

Status of the MARS code

Igor Rakhno

16th Workshop on Shielding aspects of Accelerators, Targets and Irradiation Facilities (SATIF-16)

28-31 May, 2024

In collaboration with

- Igor Tropin, Dali Georgobiani, Alajos Makovec (Fermilab)

Outline

- Introduction
- Geometry models, beamlines and transport in EM fields
- Meshes & scoring
- Residual dose – spatial distributions
- Update to TENDL-2023
- Work in progress:
  - (i) a new Python-based GUI
  - (ii) replacement of obsolete features and improved modularity
- Path forward

Introduction

- MARS code is a general purpose radiation transport Monte Carlo code with an emphasis on accelerator applications. Its development started by Nikolai Mokhov at the Institute for High Energy Physics (IHEP, Protvino, Russia) in early 1970s.
- In addition to a basic constructive solid geometry (CSG), the CERN ROOT geometry package is used. The latter is especially effective for complex geometry models.
- For accelerator applications, a beam line builder based on MADX is very helpful.
- Import of geometry models in GDML format is possible when the ROOT geometry is used.
- Structured and unstructured meshes can be used for scoring purposes. The scoring meshes can overlap.
- A recently implemented method to deal with residual radiation allows us to calculate 3D residual dose distributions in a single run without an intermediate source.
- At lower energies, an update to the most recent TALYS-based evaluated nuclear data library (TENDL 2023) is underway.

Introduction (2)

- Development of a new Python-based GUI is underway. Current GUI based on Tcl/Tk is about 25 years old.
- Work on replacement of obsolete features and improved modularity is ongoing.

Geometry models, beam lines & transport in EM fields

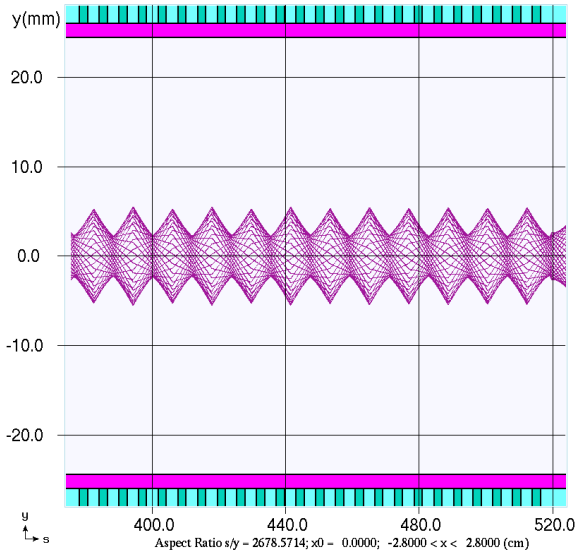
- The ROOT geometry package features about twenty basic shapes (primitives) that allow us to build complex geometry models as well as perform particle tracking and visualization.
- **Composite** shapes can be built out of the primitives using various translation and rotation matrices as well as Boolean operations.
- The shapes can be **divided** into slices, and **replicated** at other locations.
- In order to create such geometry models, users need to work with source code in C++ (no input file or GUI options in MARS).

```
Soil->AddNode(GalFlr,1,GalFlrBoxTr);
Soil->AddNode(LabyrinthAVol1,1,LabyrinthAExit1ConnTr);
// Soil->AddNode(LabyrinthAVol2,1,LabyrinthAExit3ConnTr);
Soil->AddNode(LabyrinthAVol2,1);
Soil->AddNode(LabyrinthCVol,1,LabyrinthCExit2ConnTr);
Soil->AddNode(LabyrinthCVol,2,LabyrinthCExit4ConnTr);
// Add SSR1/SSR2 penetrations
Int_t CavityNo=0;
bool InsertPenetration=true;
TGeoRotation*  PIP2TunRX90 = new TGeoRotation("PIP2TunRX90");
PIP2TunRX90->RotateX(90);
while(InsertPenetration) {
    CavityNo++;
     ostr.str("");
     ostr << "SSRPntr1H" <<  CavityNo << "Tube";
     TGeoTube* SSRPntr1HTube=(TGeoTube*) gGeoManager->GetListOfShapes()->FindObject(ostr.str().c_str());
     if(  SSRPntr1HTube == nullptr) {
       InsertPenetration=false;
```

- Visual geometry development is provided, e.g., by FreeCAD.

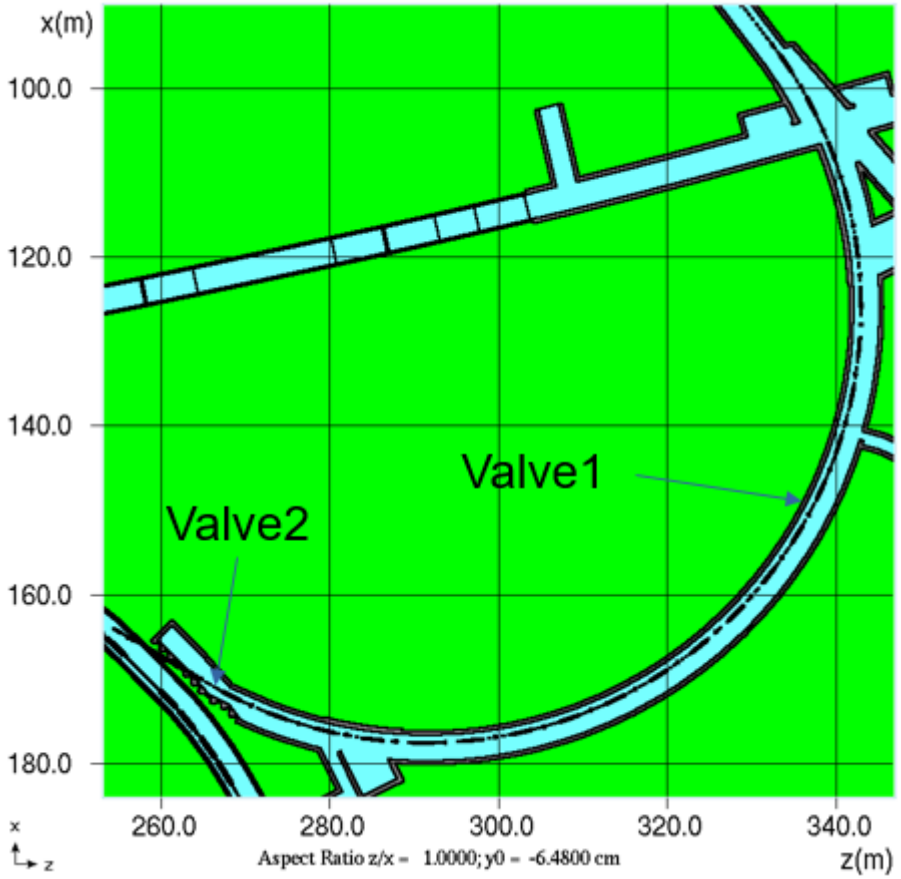
Geometry models, beam lines & transport in EM fields(2)

- For accelerator applications, an essential feature is a possibility to build models around realistic – often 3D – beam lines.
- In MARS code, a beam line builder has been developed which employs information from MADX input (usually supplied by beam physicists who design the beam line).
- All the components of the geometry model (tunnel, penetrations, …) are tied to the beam line itself which serves as a driver in the model.
- Distribution of electromagnetic field in beam line magnets allows us to perform precise particle tracking whenever necessary.

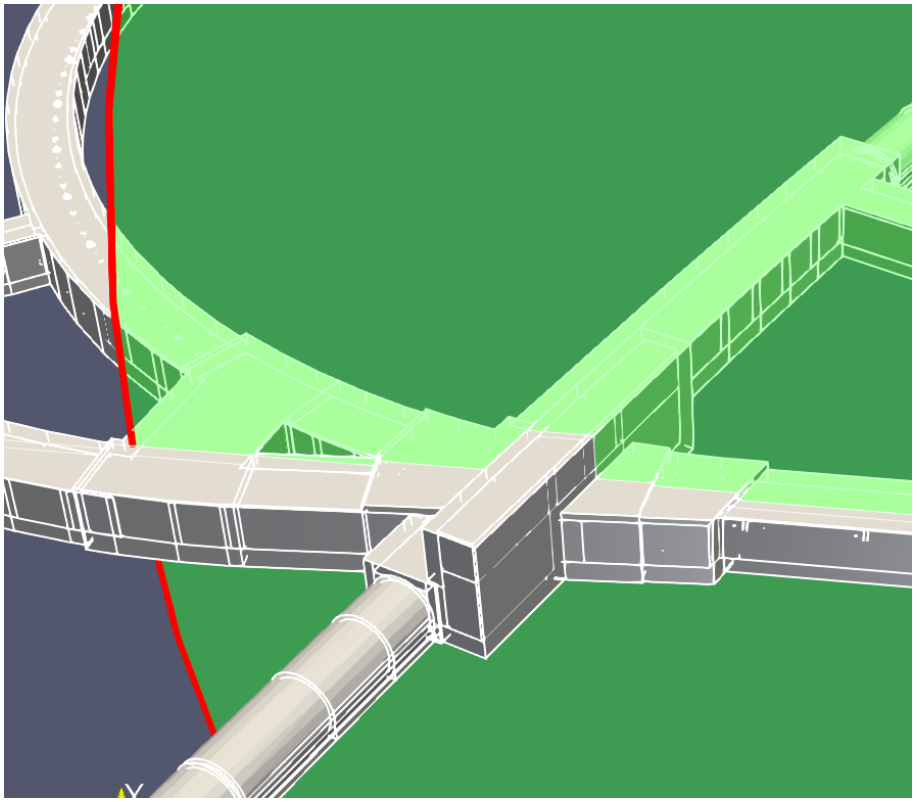


Geometry models and beam lines (3)

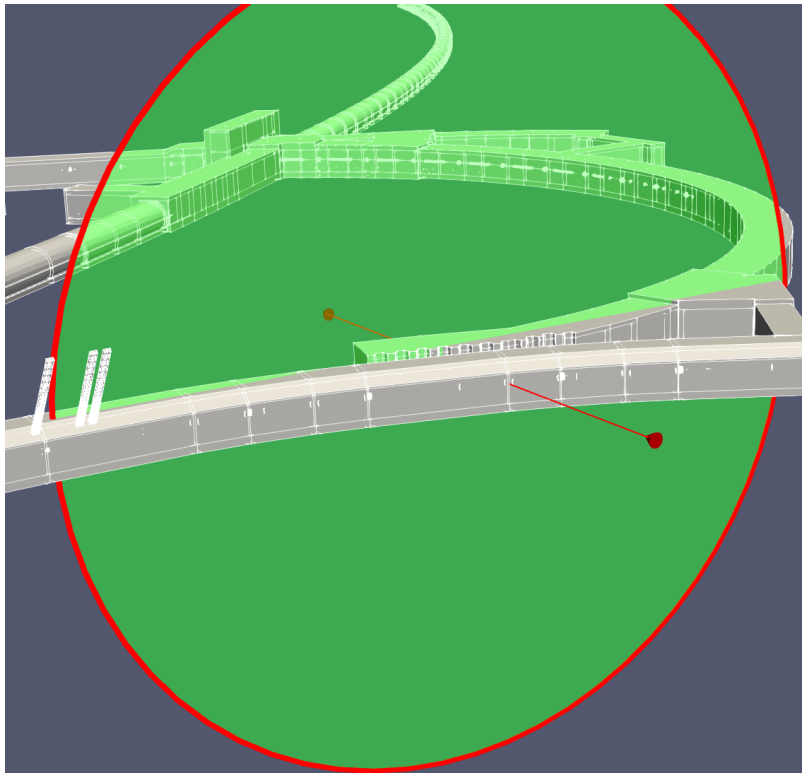
2D

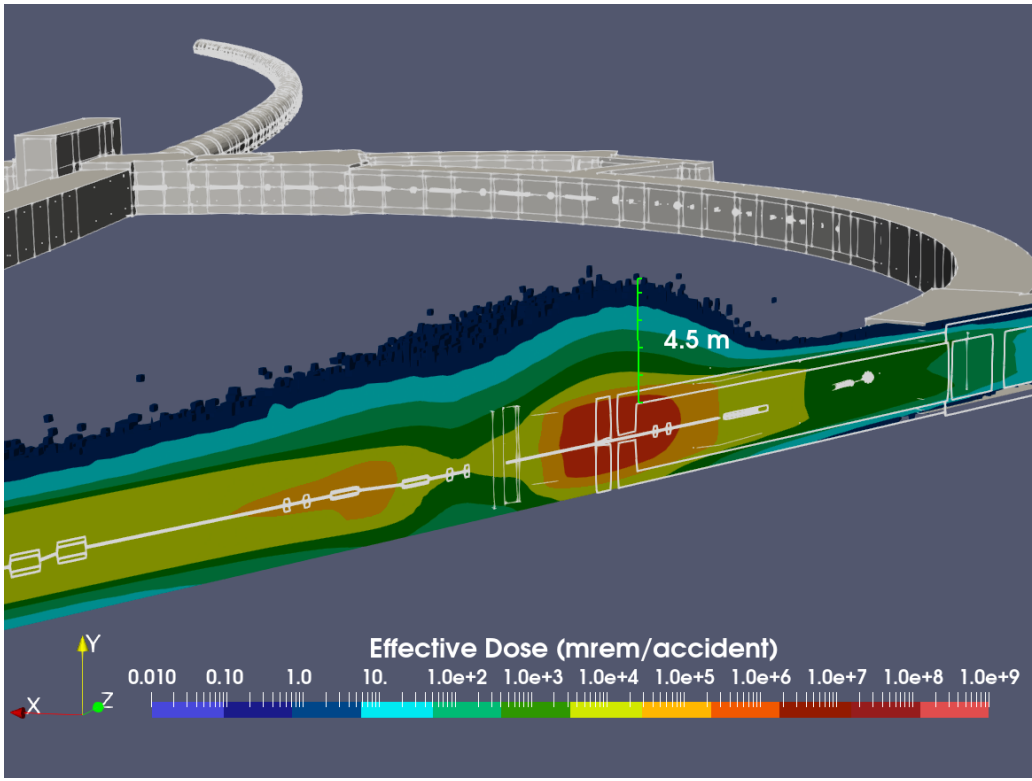


3D

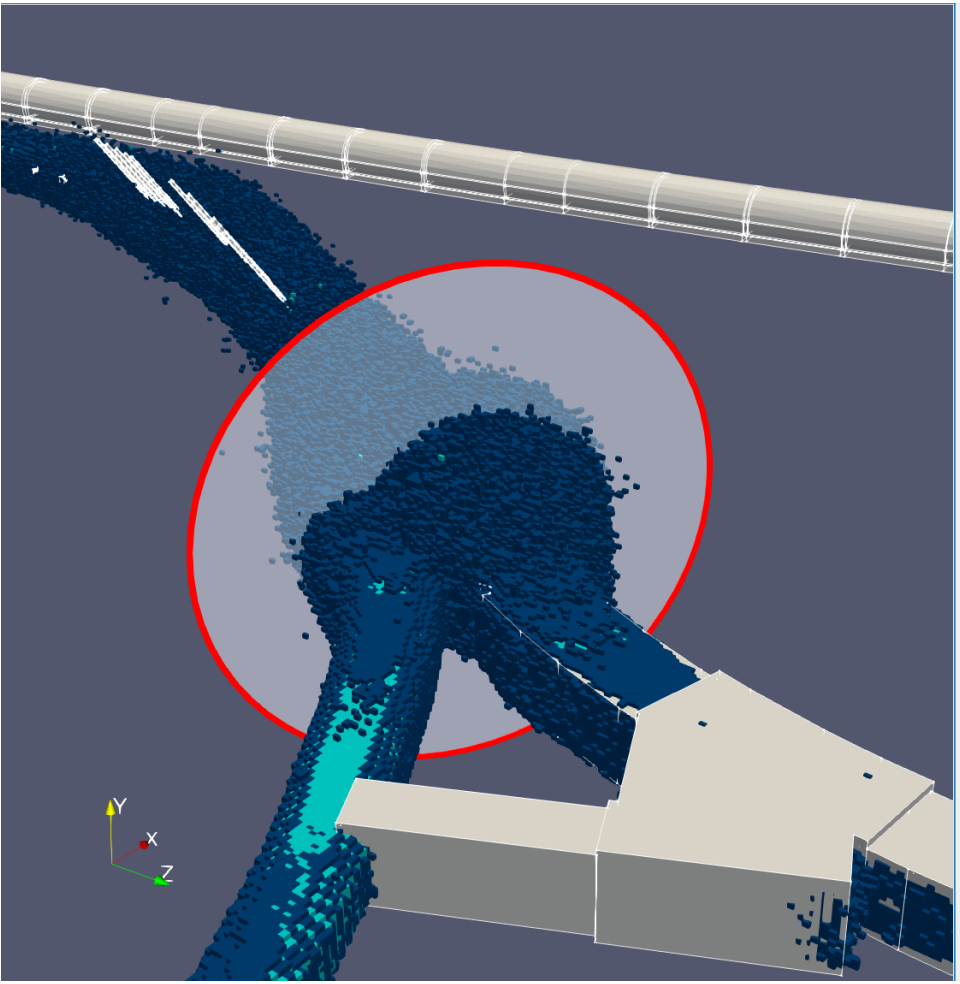


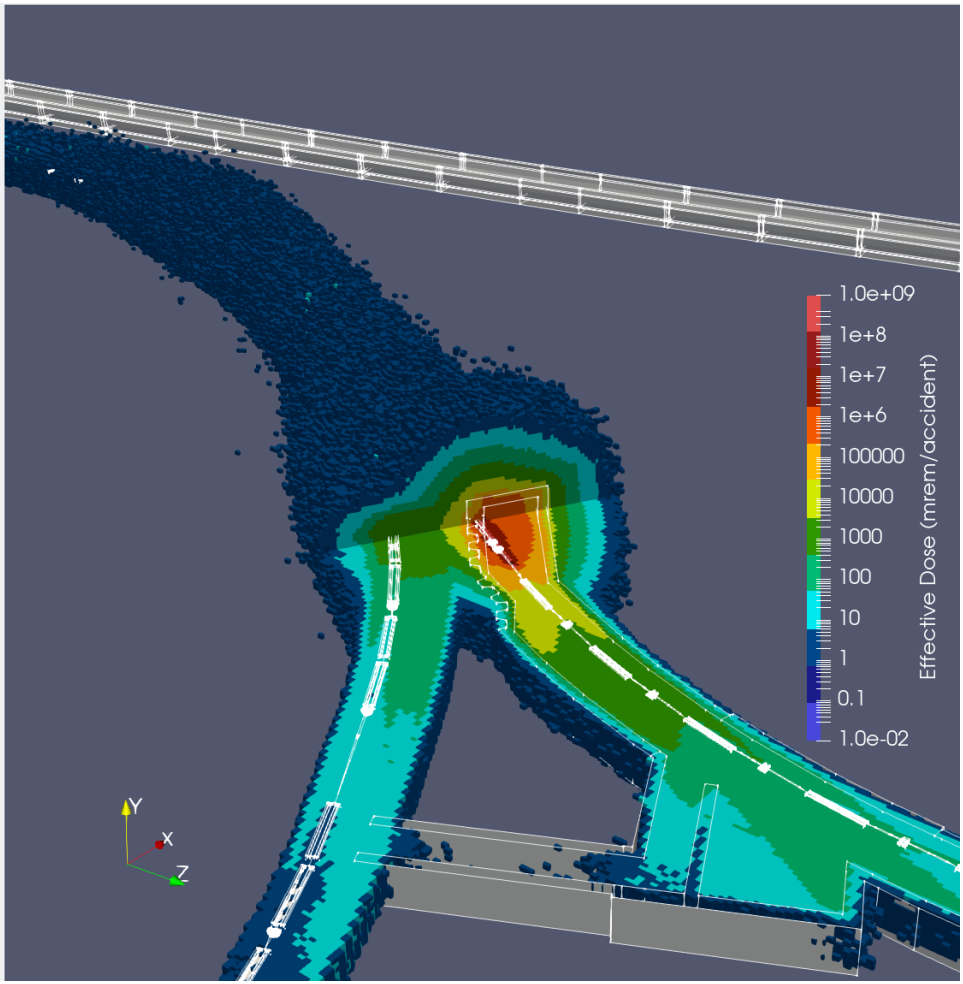
Geometry models and beam lines (4)





Geometry models and beam lines (5)



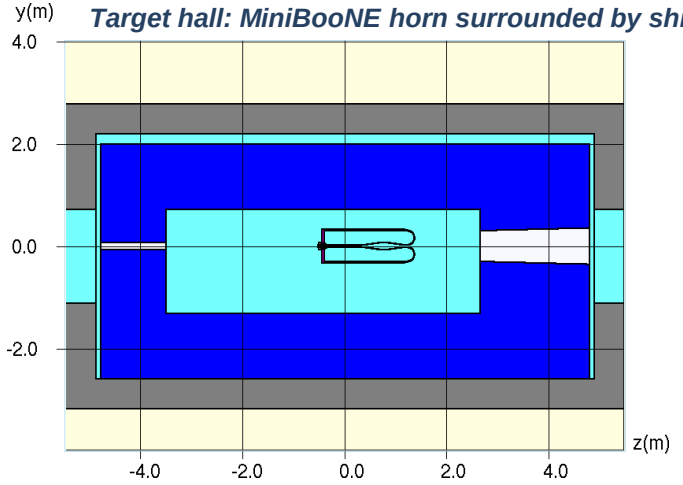


Geometry models and beam lines (6)

- Historically, the following geometry types exist in MARS code: (i) standard (axially symmetric cylindrical) geometry; (ii) extended (basic CSG geometry); (iii) non-standard geometry; (iv) ROOT geometry as an instance of non-standard geometry.
- Combinations of the geometry types are possible as well as a conversion from extended into ROOT geometry.
- Also, a conversion from ROOT to VTK format is possible using an in-house utility.
- The non-standard geometry option allows us to implement parametric additions and/or corrections to an initial complex model developed with any other geometry.
- Import of geometry models in GDML format is possible with proper extraction of information on materials employed in the GDML model.

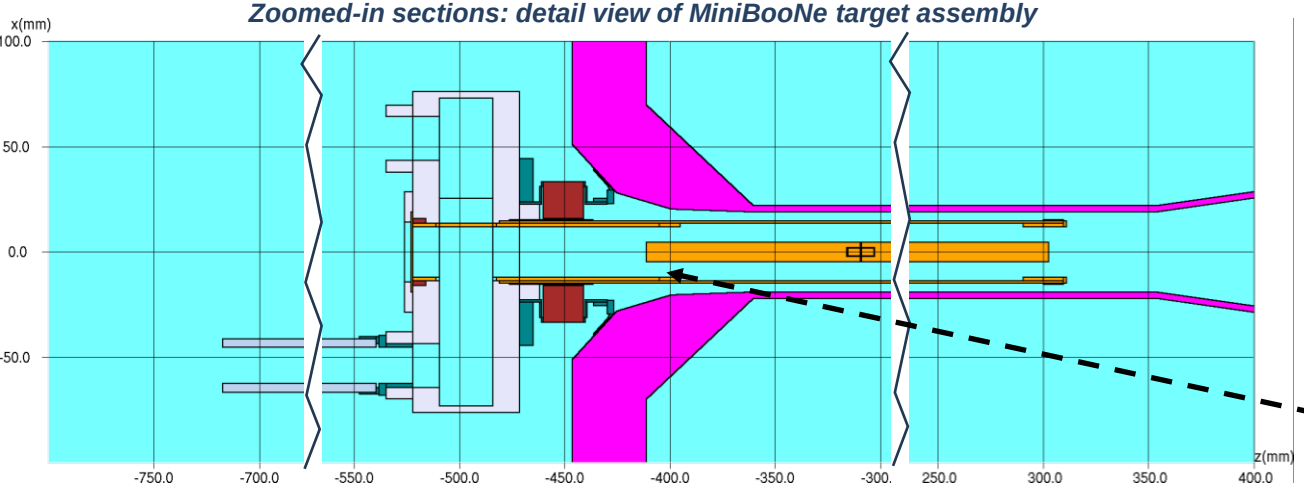
Meshes and scoring

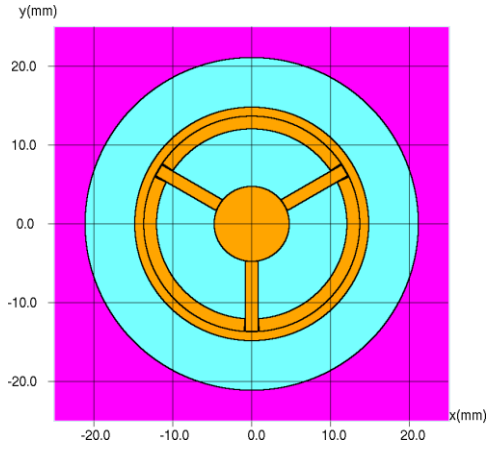
- Both 2D (hbook, root) and 3D (VTK/ParaView) meshes can be used (independently of geometry).
  - Target hall, MiniBooNE pulsed horn, and horn shielding.
  - MiniBooNe target assembly.
  - An 8-GeV proton beam.



Target hall: MiniBooNE horn surrounded by shielding

Zoomed-in sections: detail view of MiniBooNe target assembly

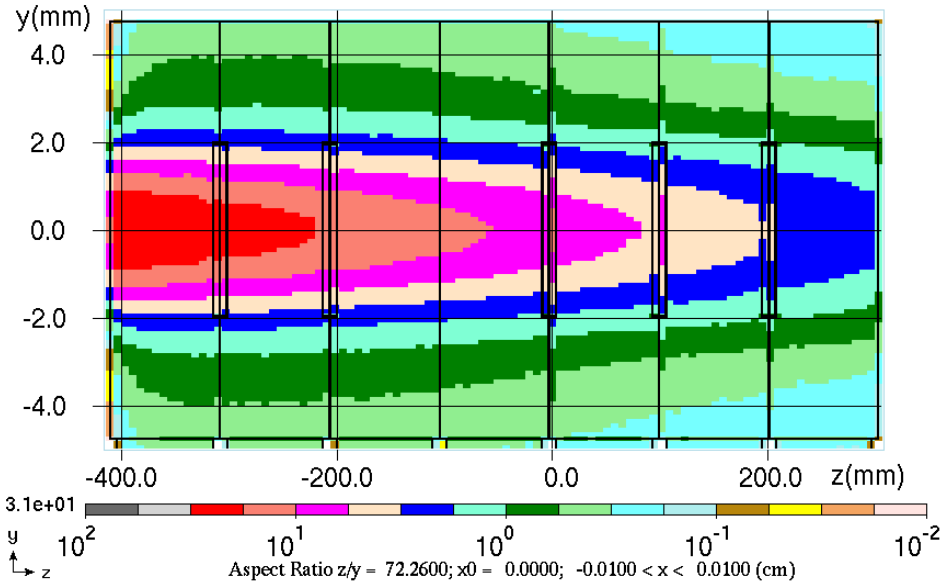




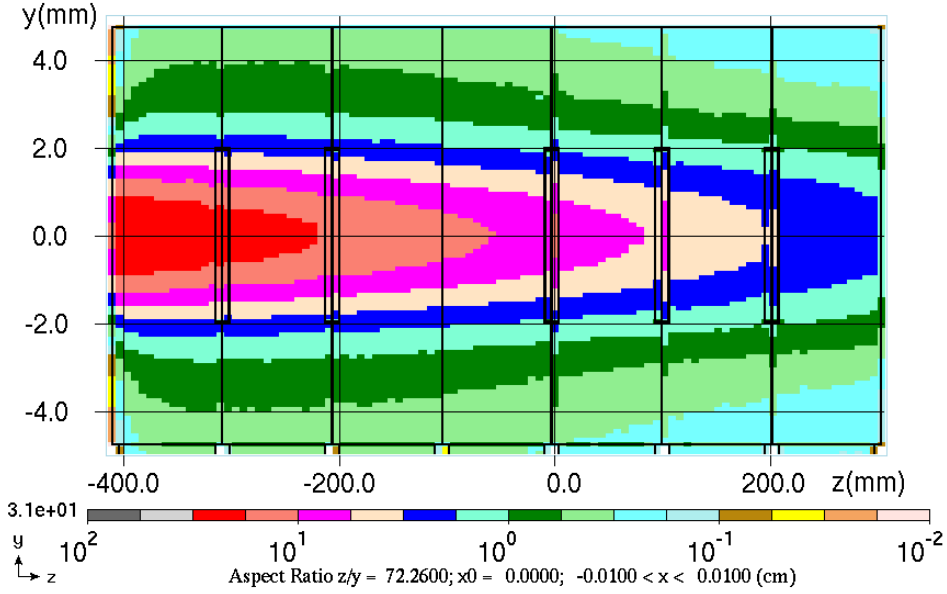
Upstream slug locator and locator fins

Meshes and scoring (2)

Energy deposition density (J/cm^3/pulse) in the target calculated with MARS histograms



HBOOK

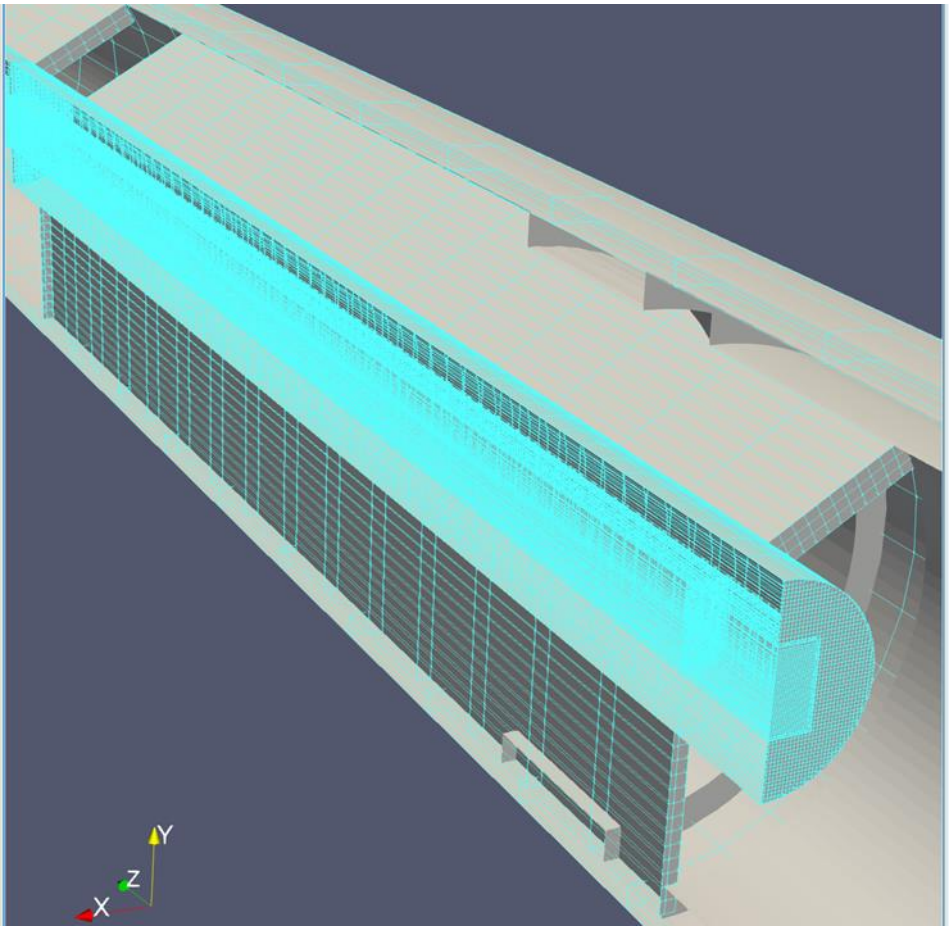


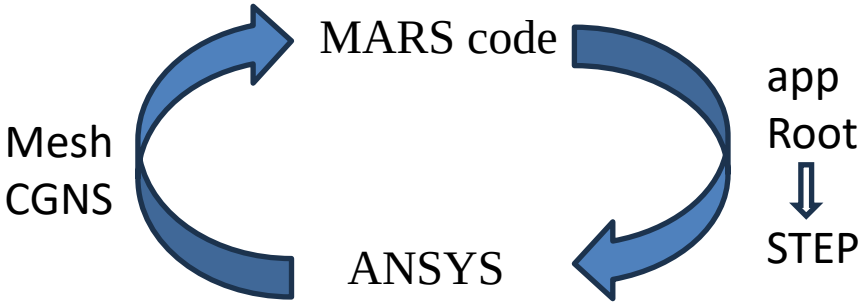
ROOT

Distributions obtained by means of built-in MARS estimators were used to validate results obtained on unstructured ANSYS mesh.

Meshes and scoring (3)

A mesh superimposed on geometry model

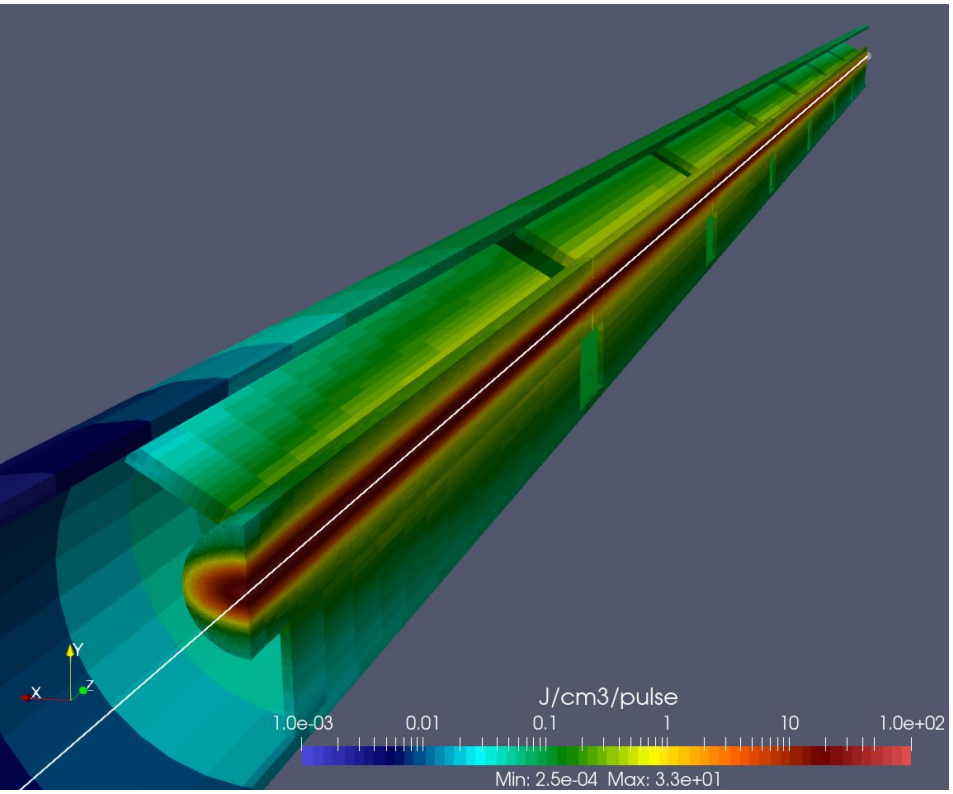




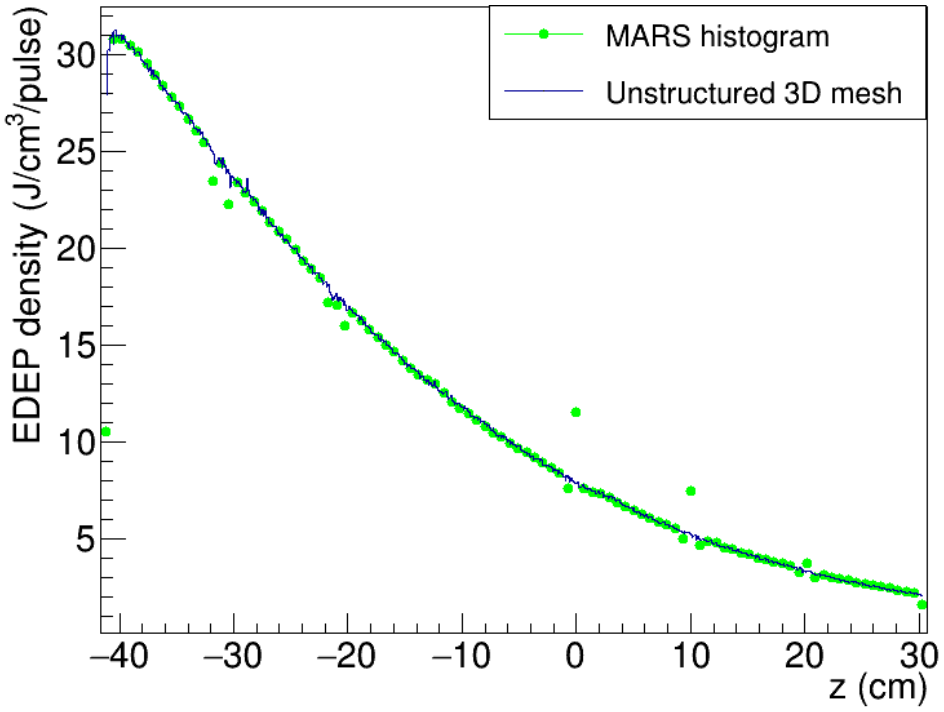
The spatial distribution of absorbed energy in the target is calculated on an unstructured mesh generated in ANSYS. To construct the mesh, the MARS geometry model converted to CAD format (STEP) was used. As a result of this approach, the boundaries of the mesh cells turn out to completely coincide with the geometric boundaries.

Longitudinal vertical X-section of the MARS model of target assembly with the ANSYS scoring mesh.

Meshes and scoring (4)



Spatial distribution of the energy deposition density in target assembly obtained on ANSYS unstructured grid, visualization in ParaView.

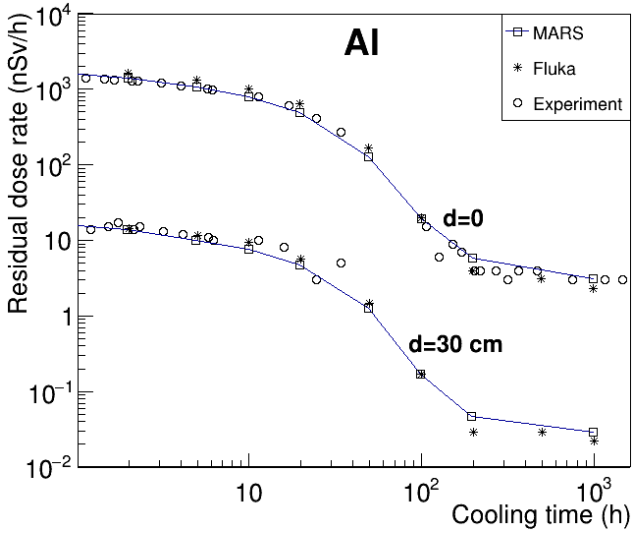


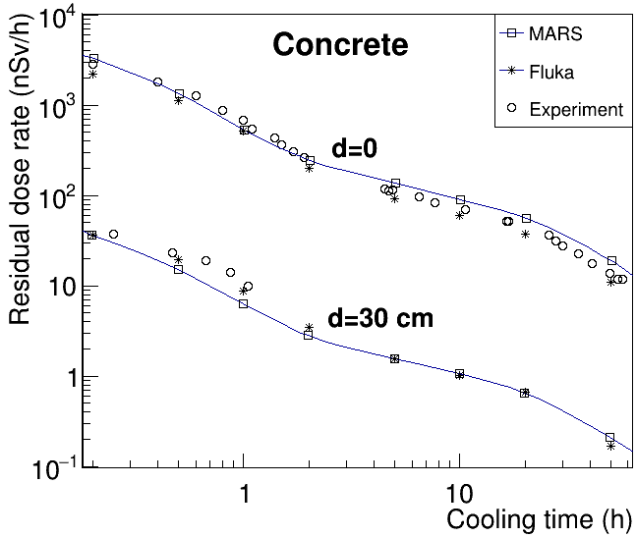
Energy deposition density in target along beam axis is compared with built-in MARS estimators (HBOOK/ROOT 2-D histogram).

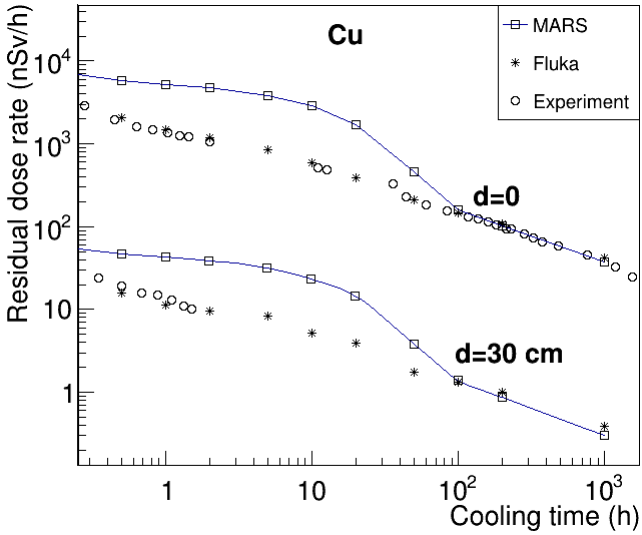
Residual dose distributions

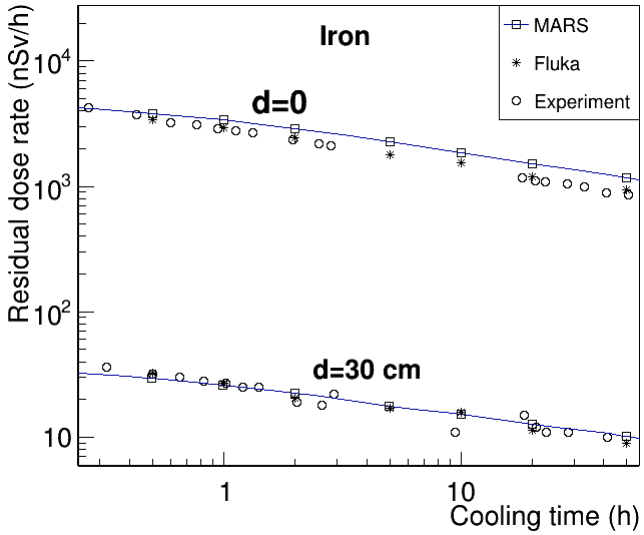
- Recently, a method has been implemented to deal with residual radiation that allows us to calculate 3D residual dose distributions in a single run without an intermediate source.
- It can be done in the same run for cases when irradiation and cooling geometries are different.
- The previously used contact residual dose option is also kept.
- SandiaDecay library has been used to simulate decay and generation of residual radiation.
- A detailed comparison with FLUKA as well as with SINBAD experimental data has been done (A. Makovec and I. Tropin, NIM **A1064**, 169423, 2024).

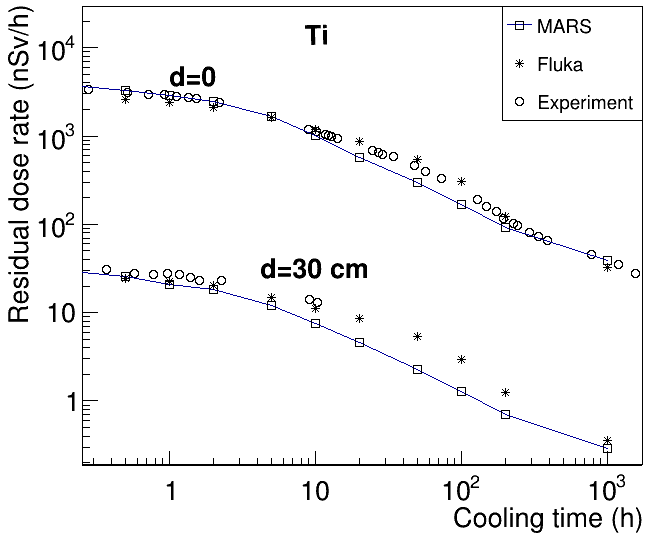
Residual dose distributions (2) – comparison with Sinbad





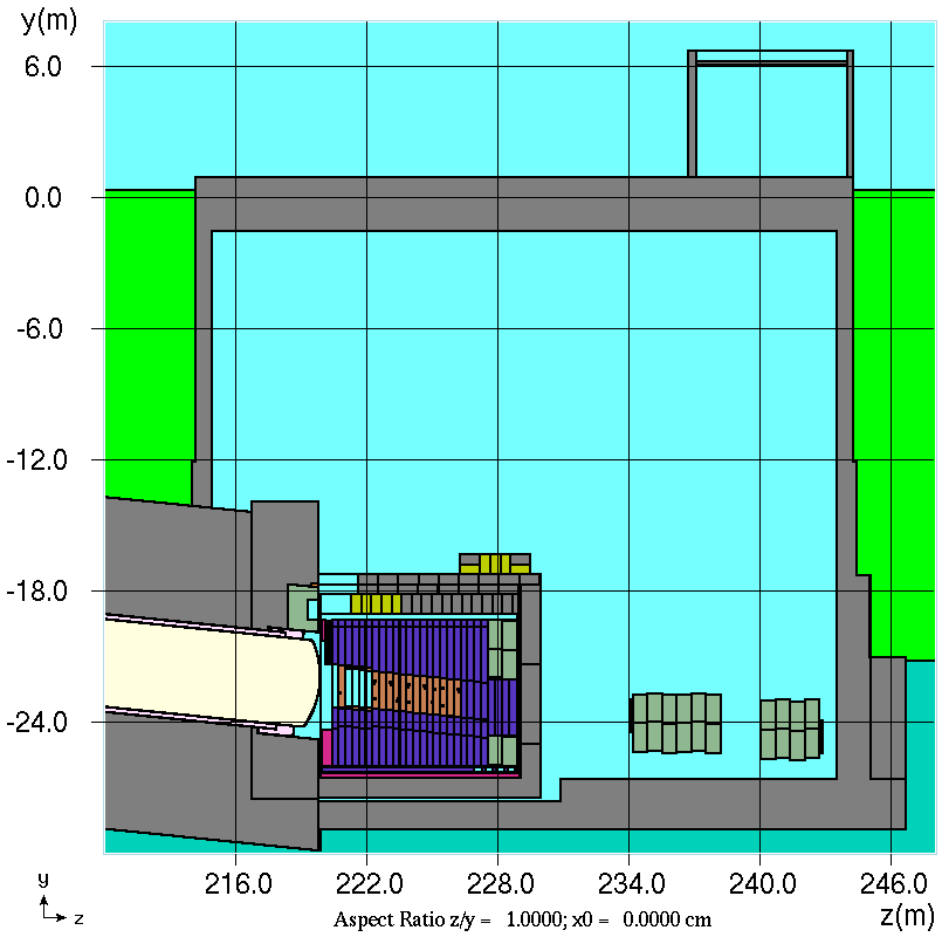


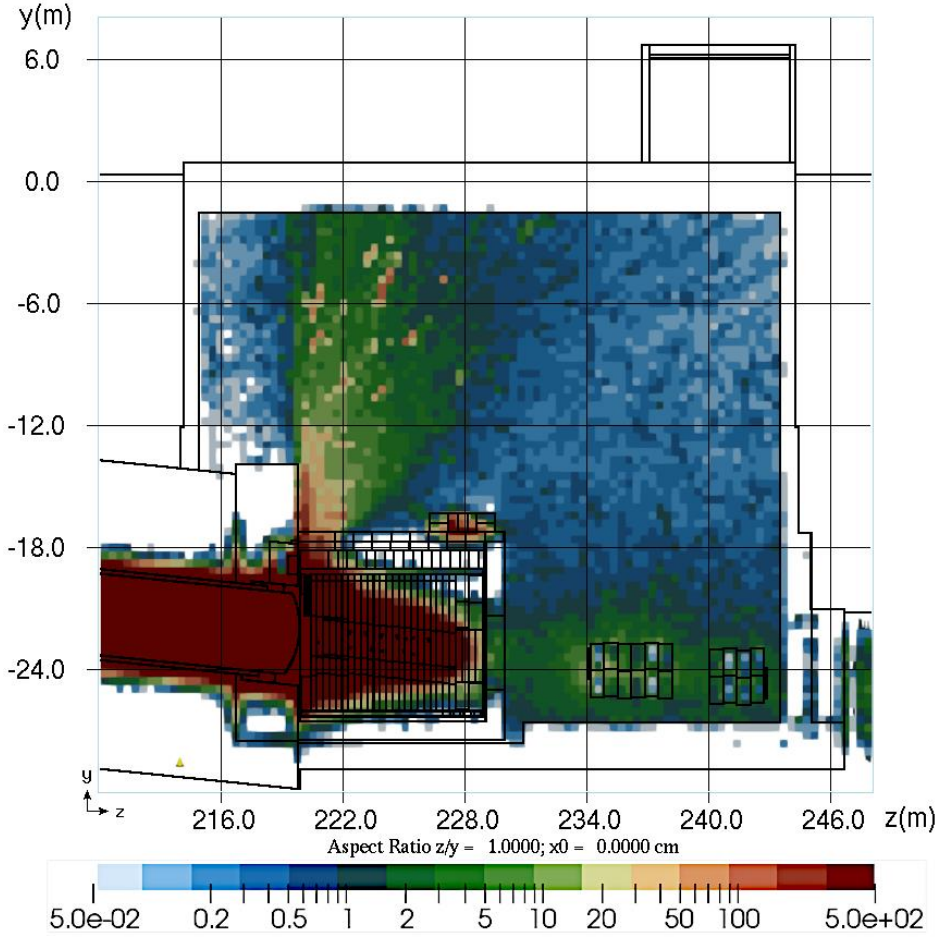




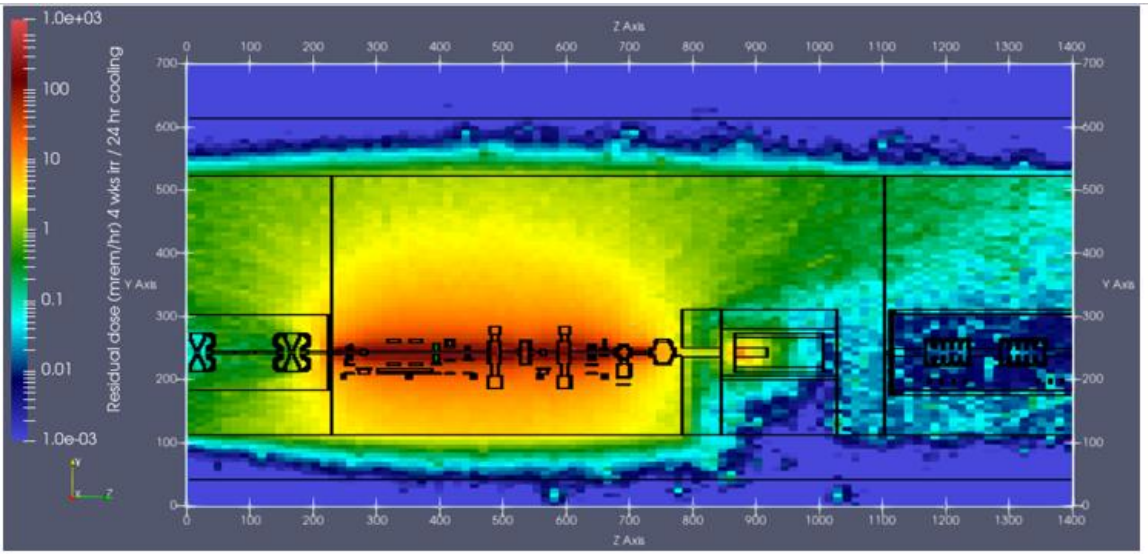
Residual dose rate distributions (3)

- Calculated residual dose rate (mrem/hr) distribution around LBNF Hadron Absorber for a scenario “200 days irradiation followed by 1 day cooling with removed concrete shielding blocks”.





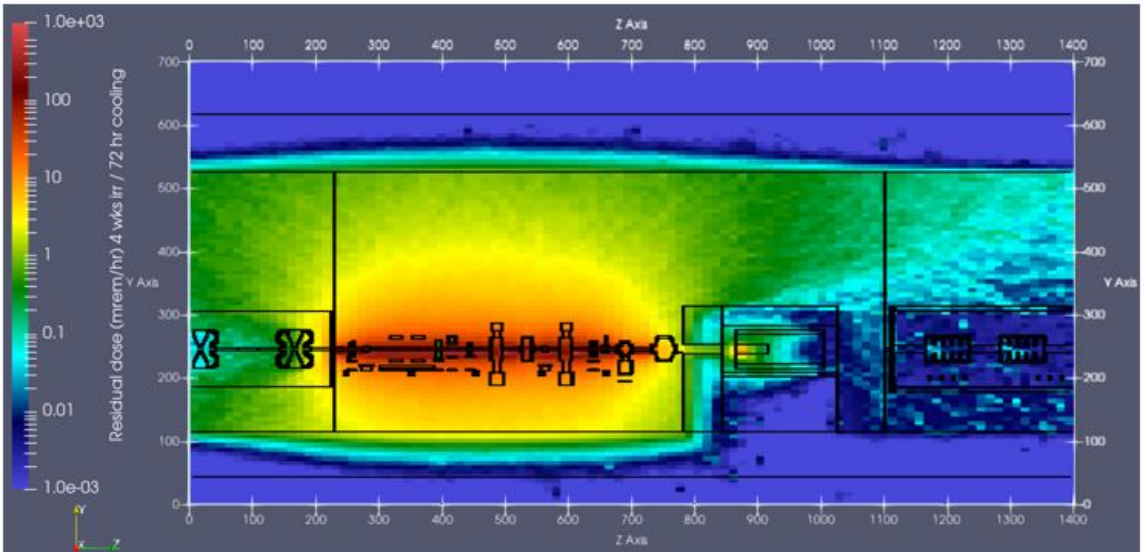
Residual dose rate distributions (4)



Beam loss (0.73 W) on Beam Pipe – Elevation view

24-hr cooling

PIP-II Instrumentation Cart at 177 MeV location (Commissioning & beam diagnostics)

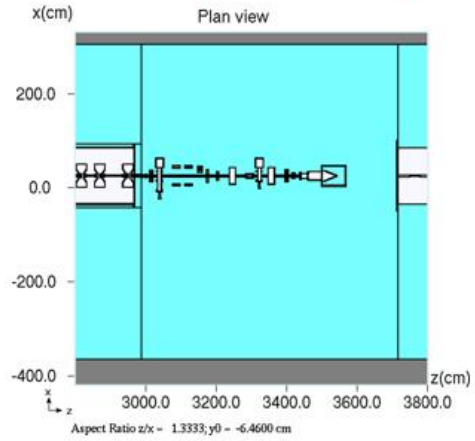


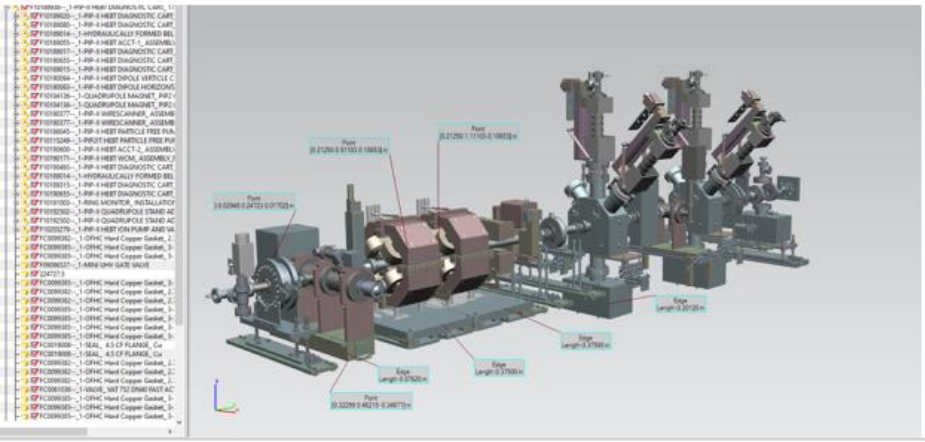
72-hr cooling

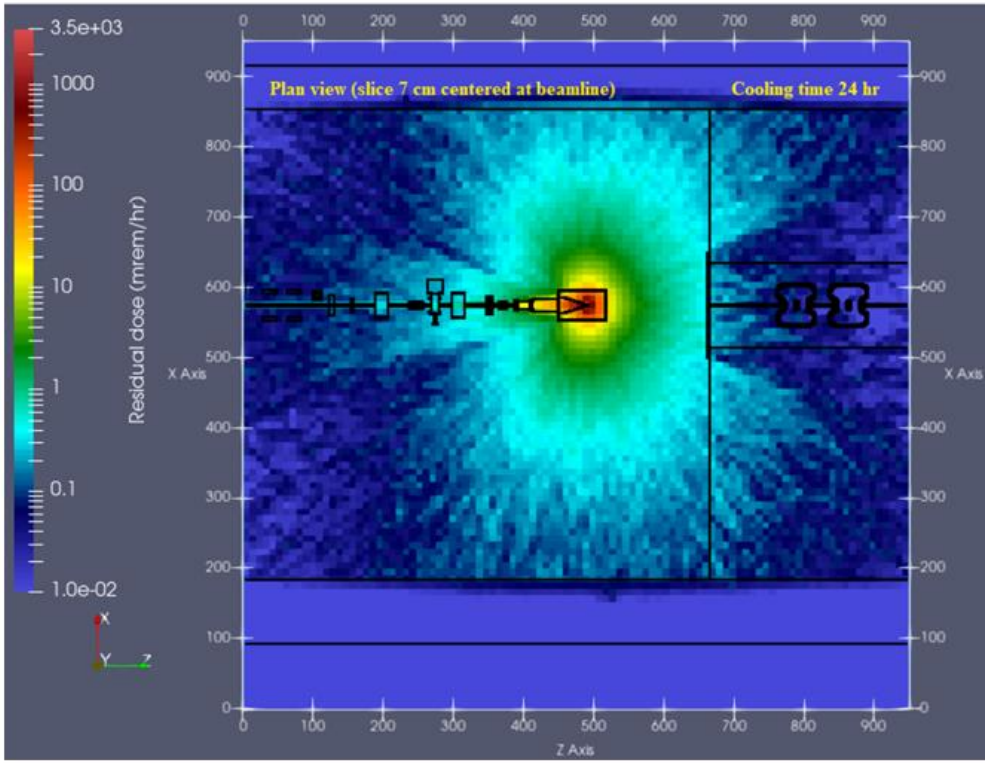
Does not buy much, ~ 30% reduction

Residual dose rate distributions (5)

Beam loss (0.363 W) on a Faraday Cup (Ni201) at 32 MeV location





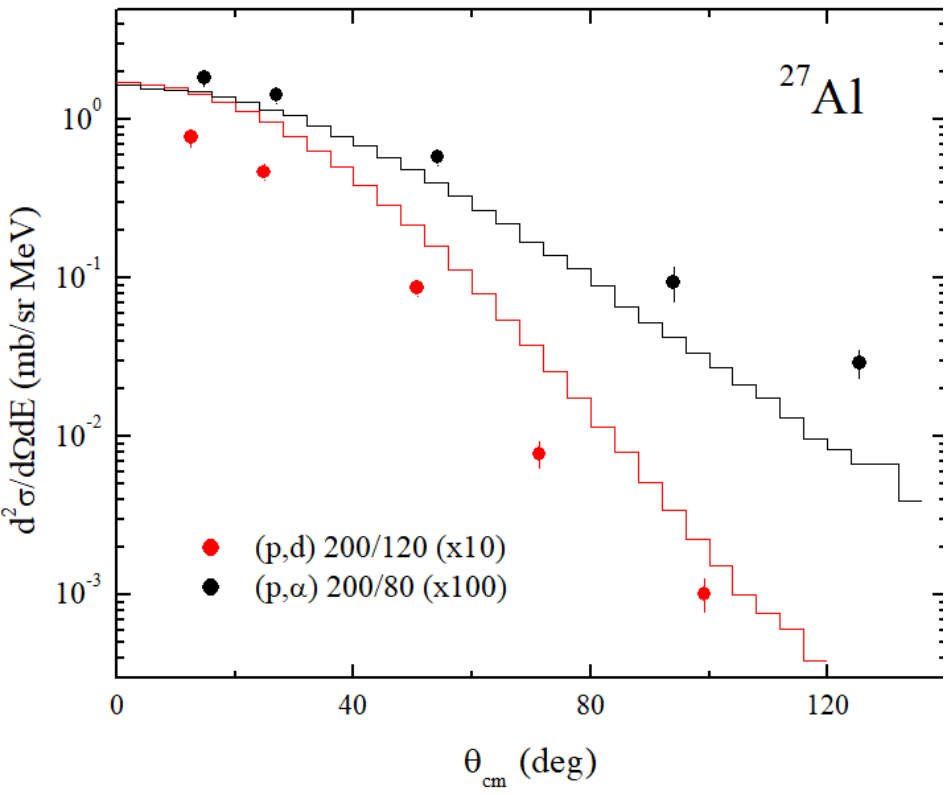


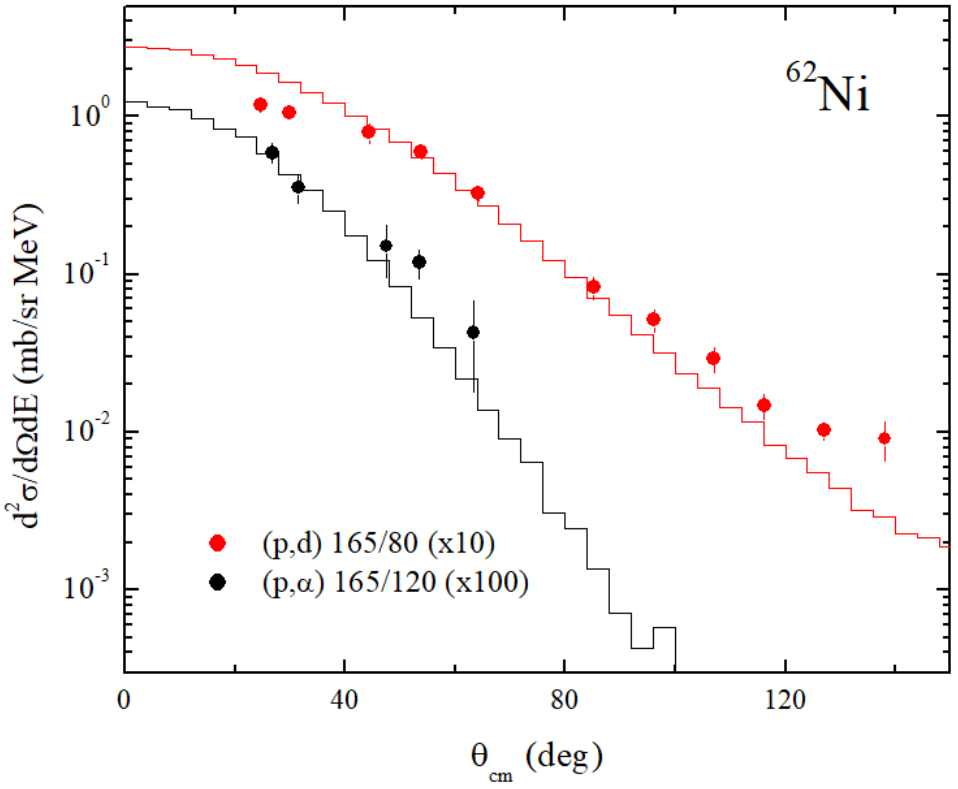
Update to the most recent library TENDL-2023

- Lower energy region 0.25 – 200/600 MeV.
- 7 projectiles: p, n, d, t, ^{3}He, ^{4}He, γ,
- Energy-angle distributions of all secondaries (p, n, d, t, ^{3}He, ^{4}He, γ as well as residual nuclei) are taken into account.
- Initial version of our in-house processing software and low energy event generator was developed in 2014 (SATIF-12). Inclusive version.
- This activity was driven, first of all, by Proton Improvement Plan-II (PIP-II).
- The update means, in fact, that we need to replace the TENDL-2021 source files (endfb) with newer ones.
- However, some tests are needed to make sure there are no surprises (for some projectile/target/secondary particle cases in the 2021 version, the secondary energies were too high – well beyond kinematically allowed region).
- The inclusive option (s0) – from 0.25 to 200 MeV – is being tested currently. The exclusive option is planned to be implemented soon.

Update to the most recent library TENDL-2023 (2)

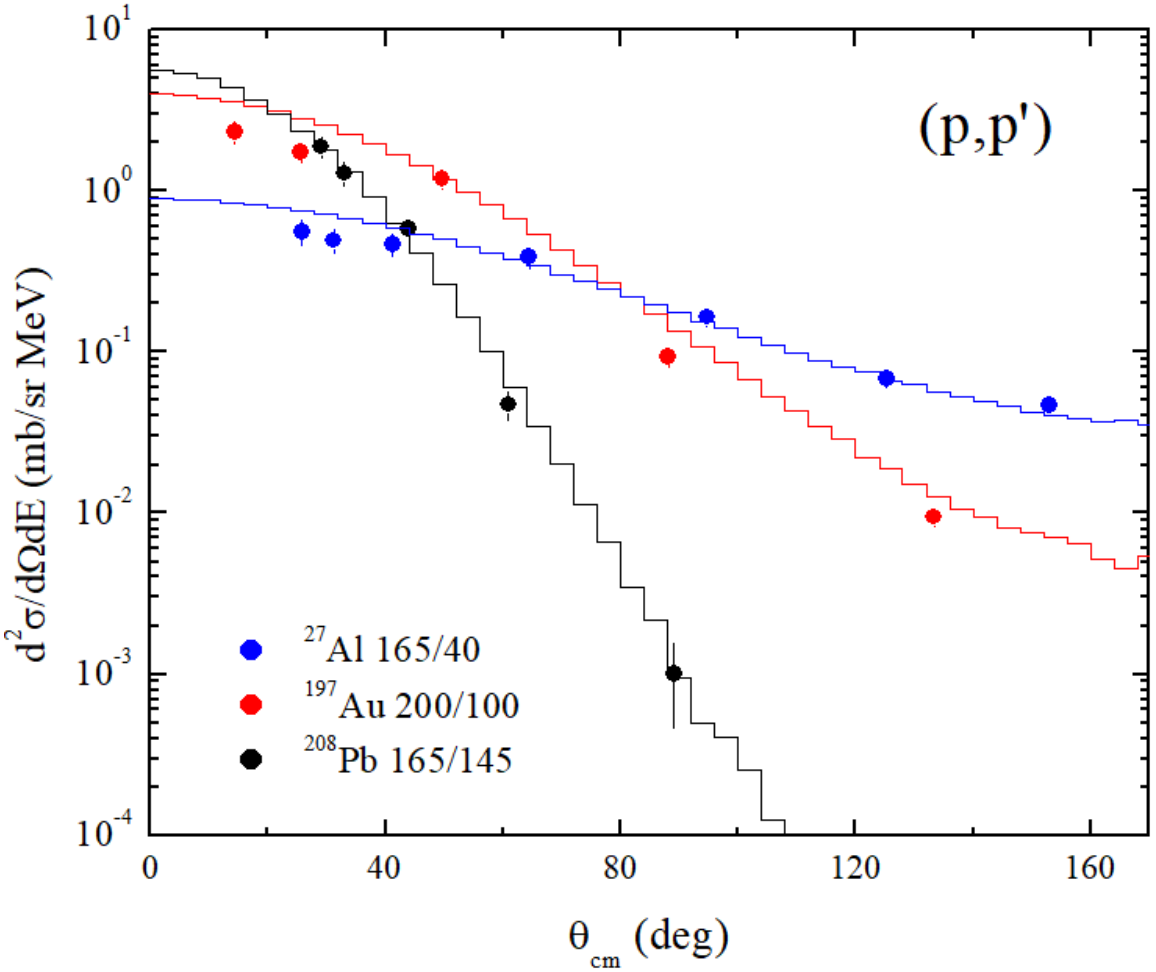
Lines – TENDL-2023, symbols – experimental data, projectile/secondary energies are given in MeV





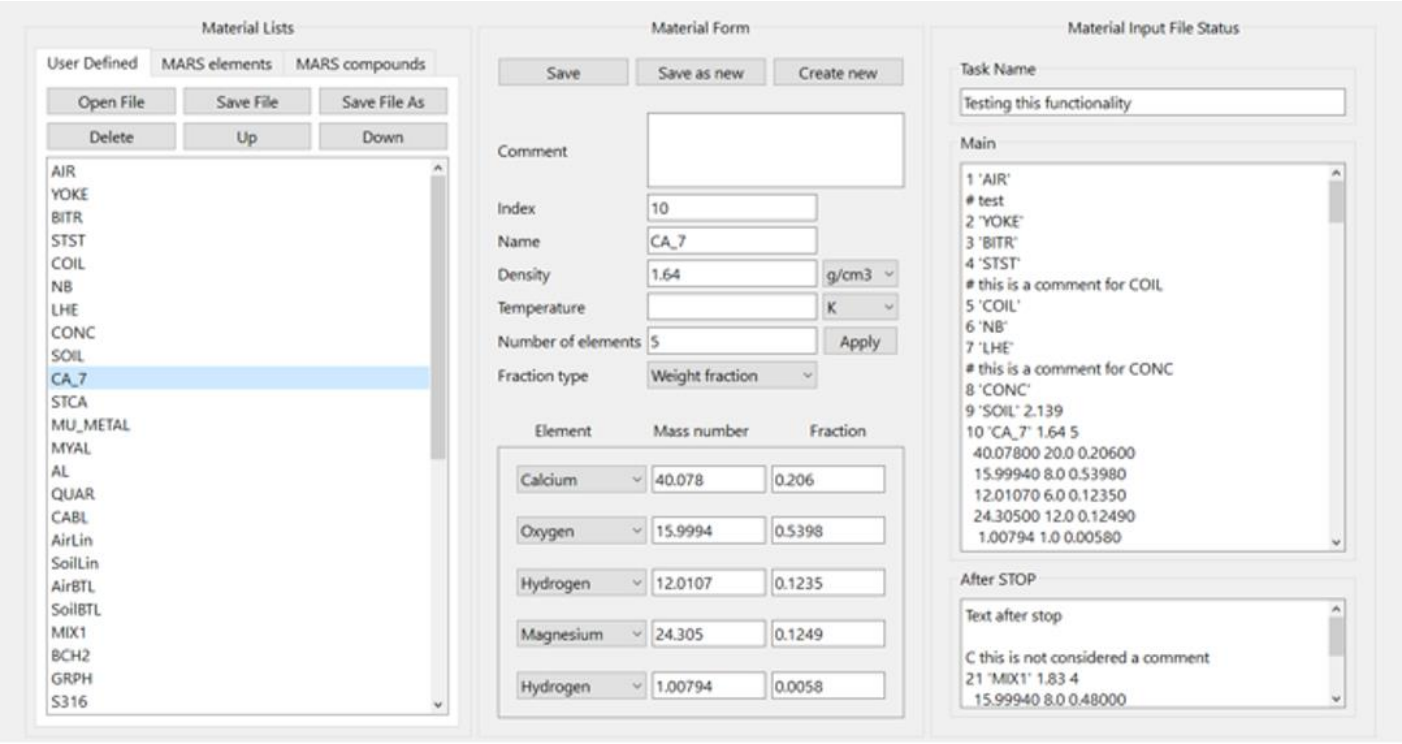
Update to the most recent library TENDL-2023 (4)

Lines – TENDL-2023, symbols – experimental data, projectile/secondary energies are given in MeV



Ongoing development of a new more user friendly GUI

- This work is in an initial stage.
- The main window facilitates switching between different windows.
- The material window (see the screenshot below) allows users to browse built-in elements and compounds, copy these into a material input file, import and export material input files, edit data for the compounds and so on.
- Tools to visualize data from MTUPLE output files, particle distributions from source files and so on.
- Demonstrated features to be added soon: 2D visualization geometry for interactive cross-sectional views.



Replacement of obsolete features and improved modularity

- A mixture of Fortran, C and C++.
- CERNLIB replacement with ROOT and VTK.
- Removal of COMMON-blocks.
- Improvement of the code structure (modules). Removal of global static data.
- Obsolete Fortran 4 features need to be removed (GO TO, BLOCK DATA, ...).

Path forward

- Replacement of obsolete features and improvement of the code structure will allow us to plan on utilizing the collection of Geant4 physics models.
- At the same time, we plan on further use and improvement of geometry options, estimators, variance reduction technique and graphical user interface developed for MARS code.
- Improvement of code efficiency. For example, instead of extensive use of virtual functions that leads to an overhead, an approach with use of templates looks preferable.
- By and large, it means that our path forward converges to development of an application that makes use of the collection of physics models from Geant4 as well as keeps the existing features of MARS code listed above.

Acknowledgements

- This work was produced by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics. Publisher acknowledges the U.S. Government license to provide public access under the DOE Public Access Plan.

- This research used Fermigrid at the Fermi National Accelerator Laboratory and ALCC allocations at the Argonne Leadership Computing Facility (ALCF), which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, and at the National Energy Research Scientific Computing Center (NERSC), which is a DOE Office of Science User Facility supported under Contract No. DE-AC02-05CH11231.