Advanced topics

A brief overview of some advanced features

Lecture overview

- Introduction

- Geometry replication (LATTICE)

- Accessing detailed particle information

- Medical applications

- Special sources

- Other advanced features and conclusion

Advanced topics

- We have covered the fundamental capabilities of FLUKA during this training

- These features are controlled through the default options and can be combined to cover a wide variety of problems, including complex geometries, scoring needs etc.

- Still, more complicated/specialised requirements may arise, such as:
  - Replicating geometries
  - Advanced sources
  - Custom scoring, extraction of detailed particle information (on an event-by-event basis)
  - Medical applications
  - And more…

- Some of these advanced capabilities require modified user routines and compilation of custom FLUKA executables
  - Default versions of the user routines can be found in the `pathtofluka/src/user/` directory

- We will cover a few examples in this lecture

Geometry replication: LATTICE

Geometry with replicated regions

- Complex geometries often exhibit replicated regions

- E.g. in particle accelerator beamlines: dipoles, quadrupoles, beam-loss monitors, etc.
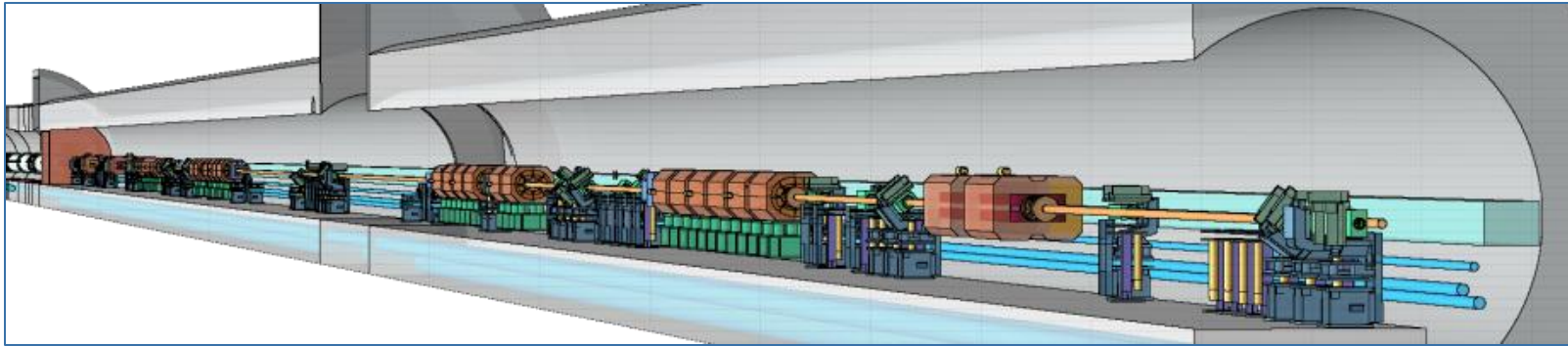
- Instead of defining every replicated region (and their interior complexity) explicitly, one may define a prototype region and then use FLUKA's LATTICE capabilities to replicate it as many times as needed elsewhere in the geometry
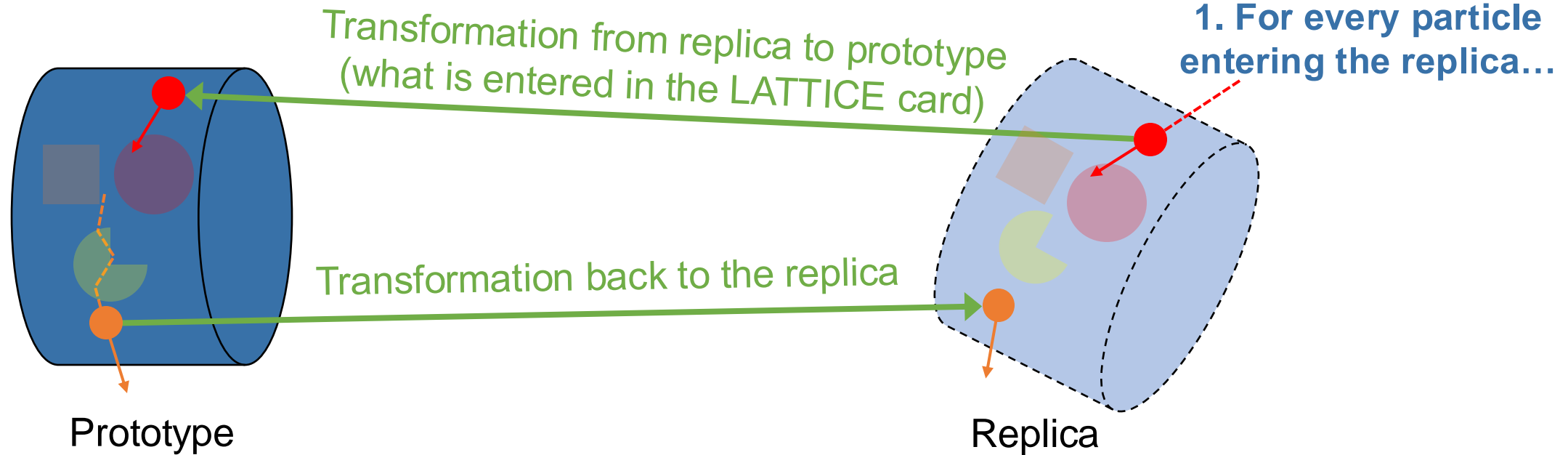
Lattice: the concept

- **Prototype** region: a region enclosing a complex geometry (full detail of various bodies, regions, material definitions, etc). Typically, the prototype container is a closed body (RCC, SPH, etc.)

- **Lattice cell** region: empty transformed prototype region where prototype region contents should be effectively replicated (see next slide).

- The **LATTICE** card links lattice cell regions to prototype regions, effectively replicating the prototype region content in each lattice cell (see next slide).

- Replica-prototype link is established via transformation (defined with ROT-DEFI), which exactly transforms the lattice cell into the prototype (!)

Prototype   Empty lattice cell   **LATTICE**   Prototype   Replica
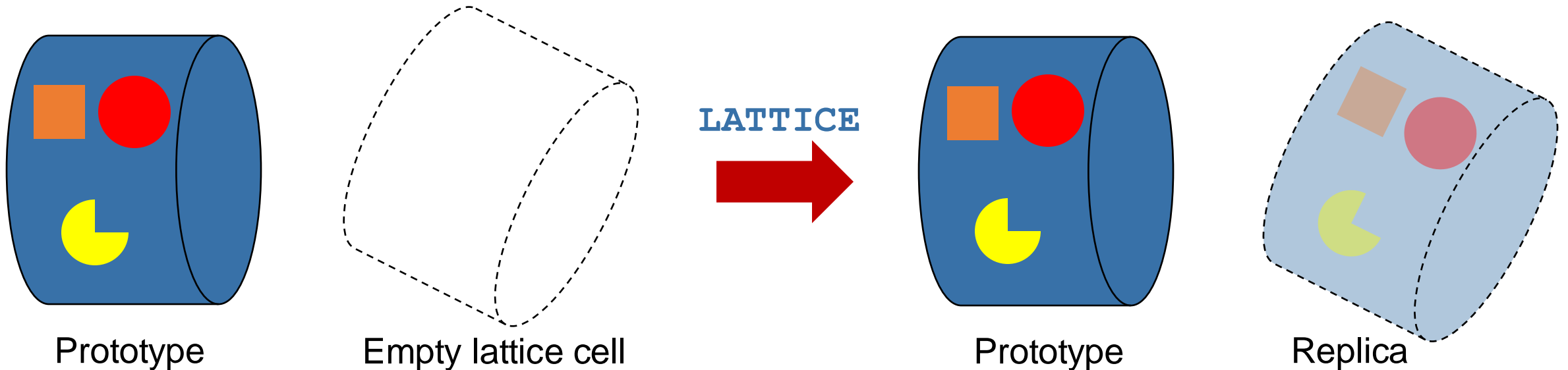
Behind the scenes

2. …its coordinates are transformed back to the prototype, where FLUKA performs the tracking

Transformation from replica to prototype (what is entered in the LATTICE card)

1. For every particle entering the replica…

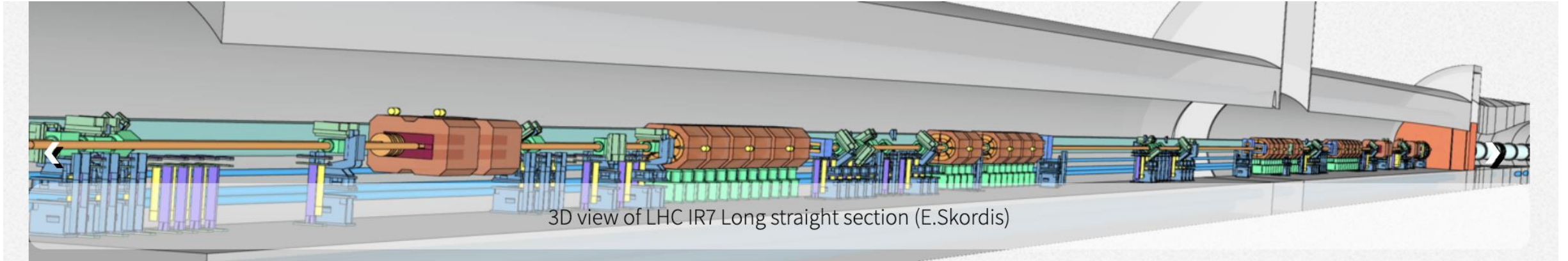Transformation back to the replica

Prototype

Replica

3. A resulting particle exiting the prototype…

4. …is transformed back to the replica boundary to be tracked in the surrounding geometry
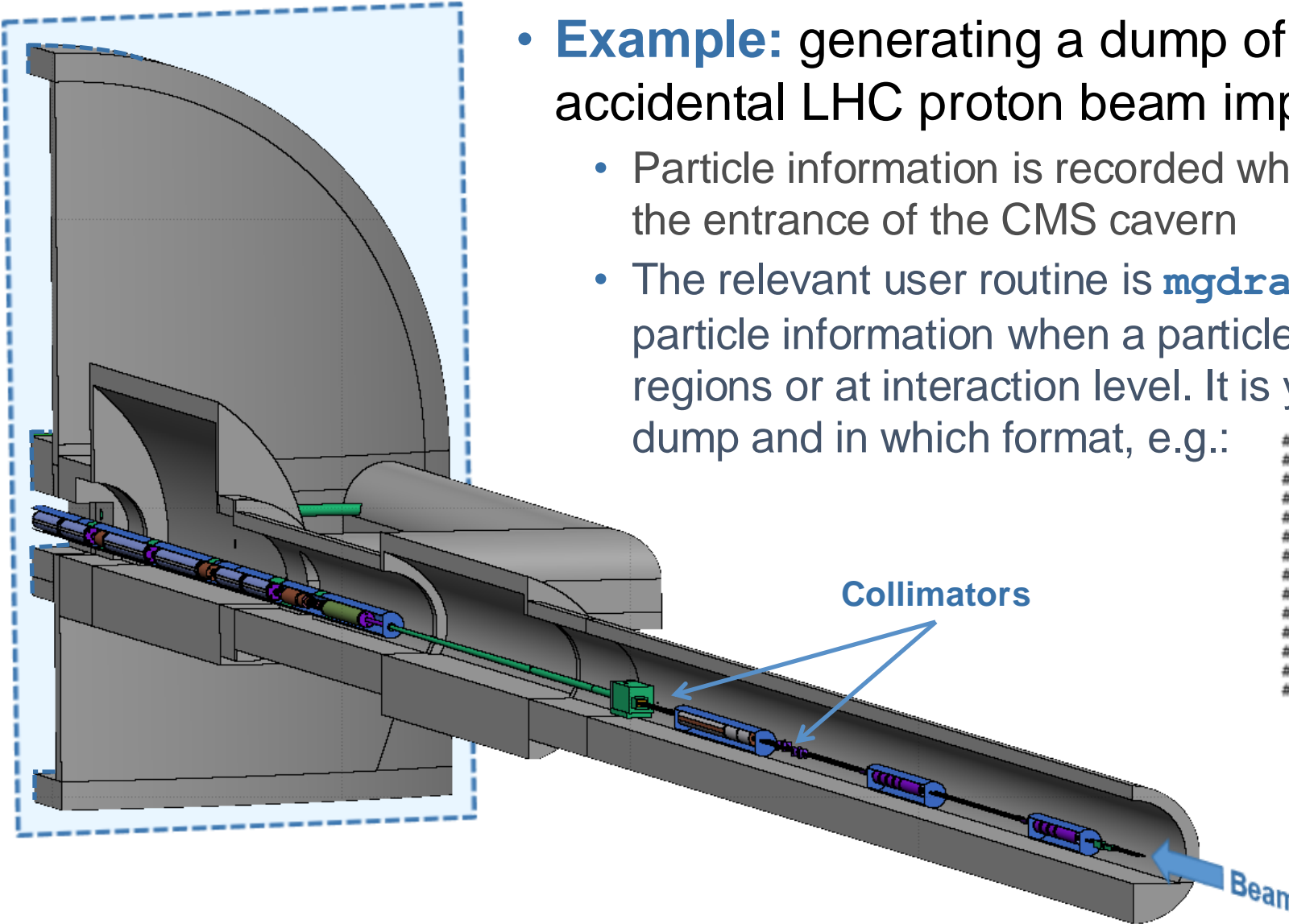
LATTICE example

- Building on the LATTICE functionality, one may construct arbitrarily complex geometries:

3D view of LHC IR7 Long straight section (E.Skordis)

- For accelerator lines: LINE BUILDER (makes heavy use of LATTICE capabilities) [A. Mereghetti et al., IPAC2012, WEPPD071, 2687]

Accessing particle information

Accessing particle information

- **Example:** generating a dump of particles produced from an accidental LHC proton beam impact on collimators
  - Particle information is recorded when crossing an interface plane at the entrance of the CMS cavern
  - The relevant user routine is `mgdraw.f` which allows to intercept particle information when a particle crosses the boundary between two regions or at interaction level. It is you, the user, who tells what to dump and in which format, e.g.:

Collimators

Beam

```
# Scoring particles entering Region No  2126
# Col  1: FLUKA run number
# Col  2: primary event number
# -- Particle information --
# Col  3: FLUKA particle type ID
# Col  4: Kinetic energy (GeV)
# Col  5: Statistical weight
# -- Crossing at scoring plane --
# Col  6: x coord (cm)
# Col  7: y coord (cm)
# Col  8: x dir cosine
# Col  9: y dir cosine
# Col 10: Particle age since primary event (sec)
# Col 11: Total energy (GeV)
      0    1     7  1.0685640710268577E-03  1.0000000000000000E+00  5.80402
      0    1     7  2.0869470497192035E-04  1.0000000000000000E+00  6.14101
      0    1     8  3.4885316857469206E-11  1.0000000000000000E+00 -8.73614
      0    1     7  7.0087267686422025E-02  1.0000000000000000E+00 -4.77469
      0    1     8  5.0713988564154988E-11  1.0000000000000000E+00 -7.06454
      0    1     7  3.1391604740674895E-03  1.0000000000000000E+00 -3.73255
      0    1     7  8.2376767285229514E-03  1.0000000000000000E+00 -4.63773
      0    1     4  2.3790418550118337E-02  1.0000000000000000E+00  5.64980
      0    1     7  2.0485104425603303E-02  1.0000000000000000E+00 -4.90033
      0    1     7  1.1054655441846896E-03  1.0000000000000000E+00  2.43893
      0    1     3  2.2395286454039216E-02  1.0000000000000000E+00  3.63118
```

Accessing particle information

- This information can be processed to study the properties of the particle population
  - e.g. the spatial distribution near the beam-line and the energy distribution of particles for different choice of collimator materials and beam impacts:
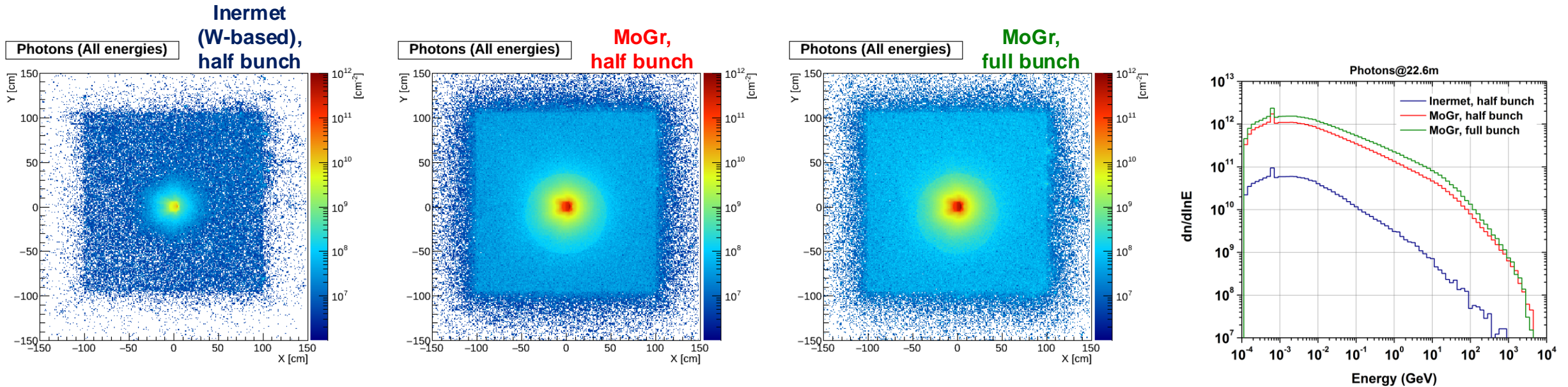

- **Note:** The great flexibility of this type of scoring comes with the penalty of ad-hoc post-processing with custom tools!
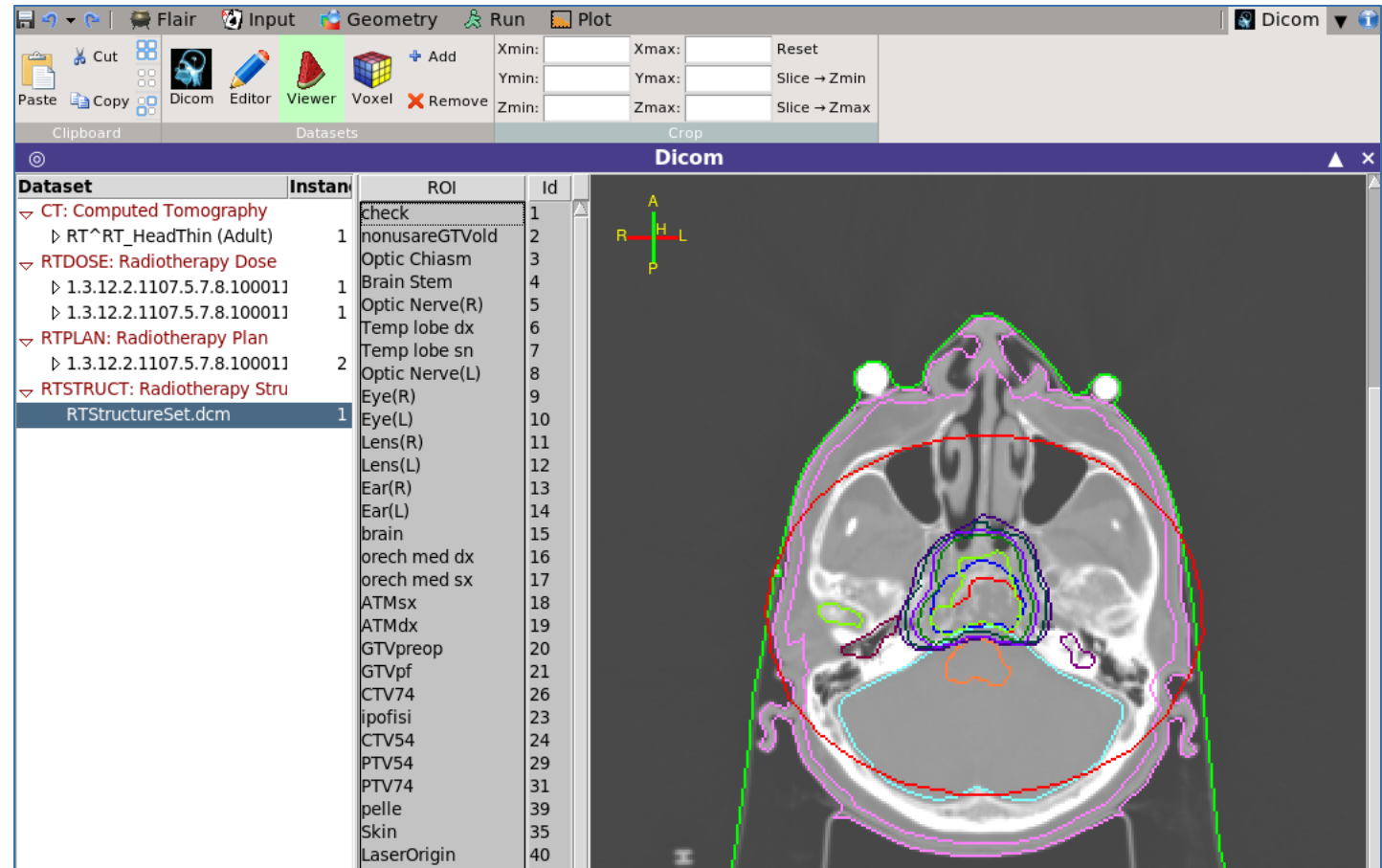
Medical applications

Medical applications

- A geometry can be described in terms of voxels, tiny parallelepipeds of equal size forming a 3-dimensional grid

- Voxel geometries are especially useful for importing CT scans, e.g. for dosimetric calculations of radiotherapy treatments

- Flair can process CT scans in the DICOM(*) format using the pydicom module and convert them to FLUKA voxel geometries or USRBIN-compatible files

  (*) DICOM (Digital Imaging and Communications in Medicine) is a medical standard for distributing any kind of medical image.
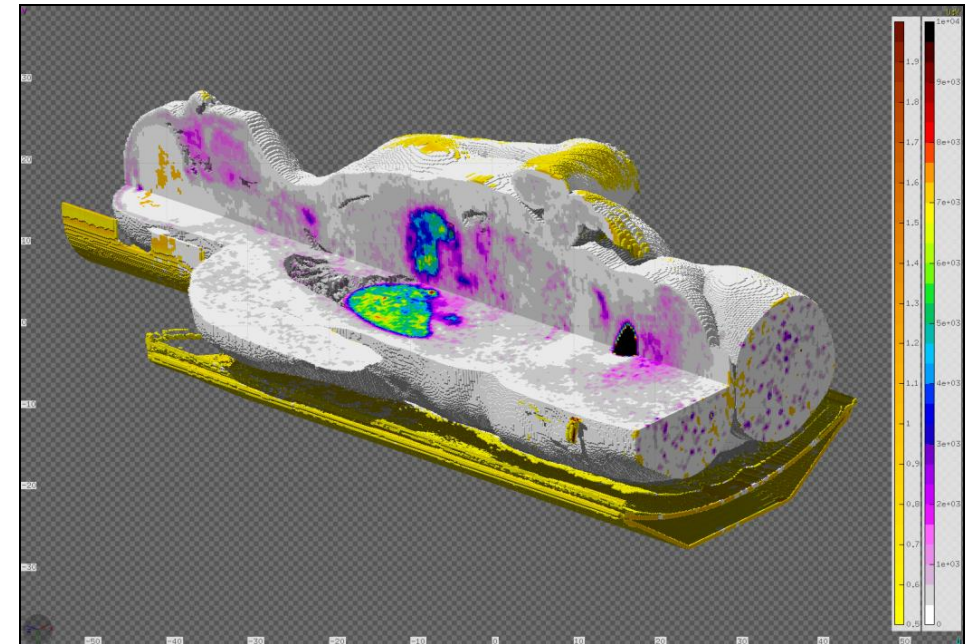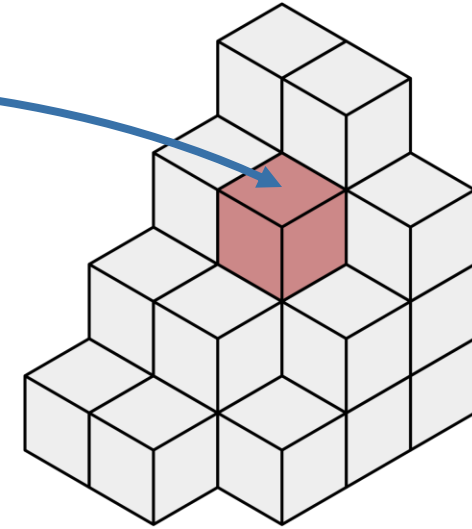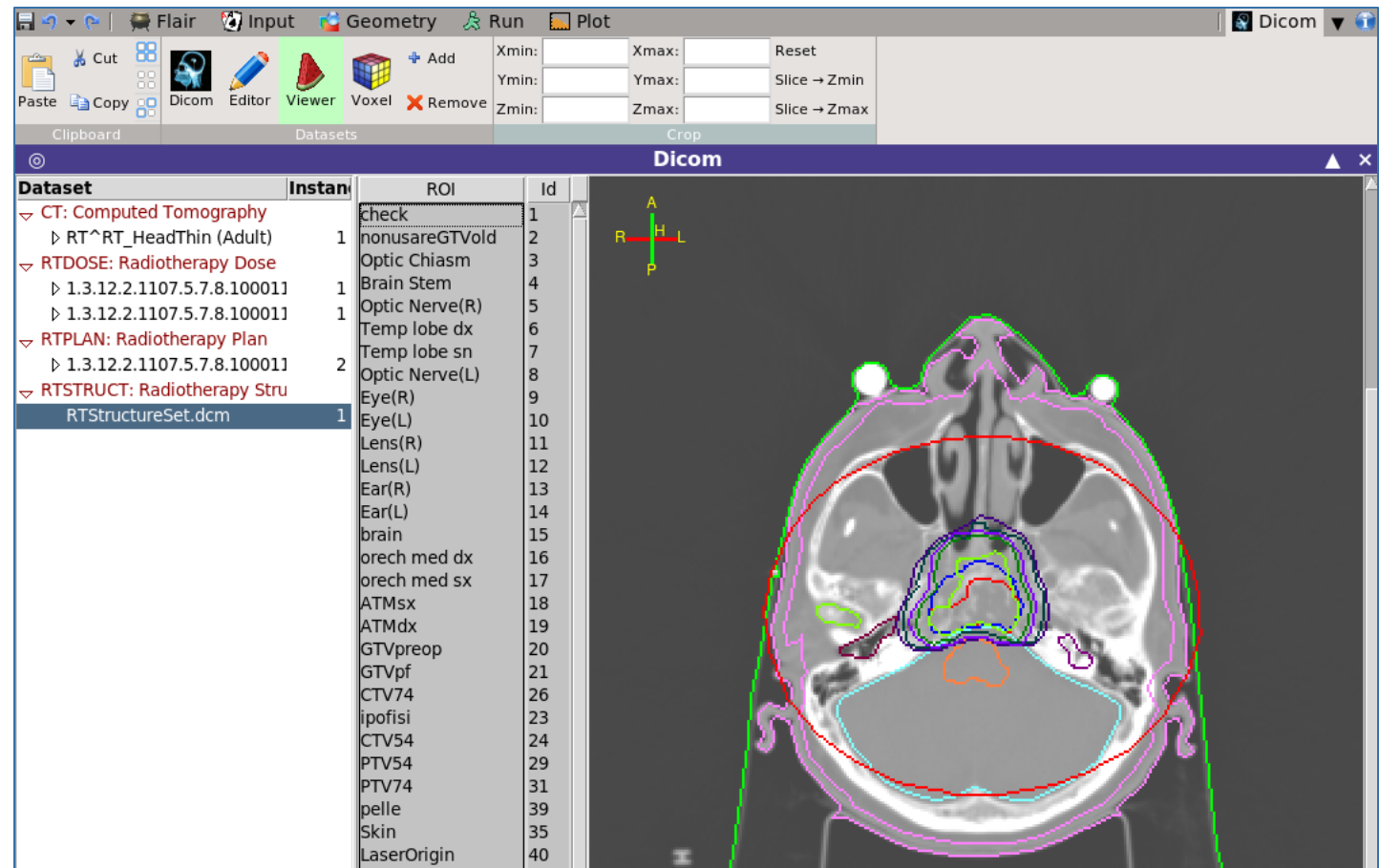
Medical applications

- DICOM files can be browsed, visualised and edited (e.g. anonymised)

- Voxels can be grouped into "organs" or regions of interest (RTSTRUCT in a pre-existing DICOM file)
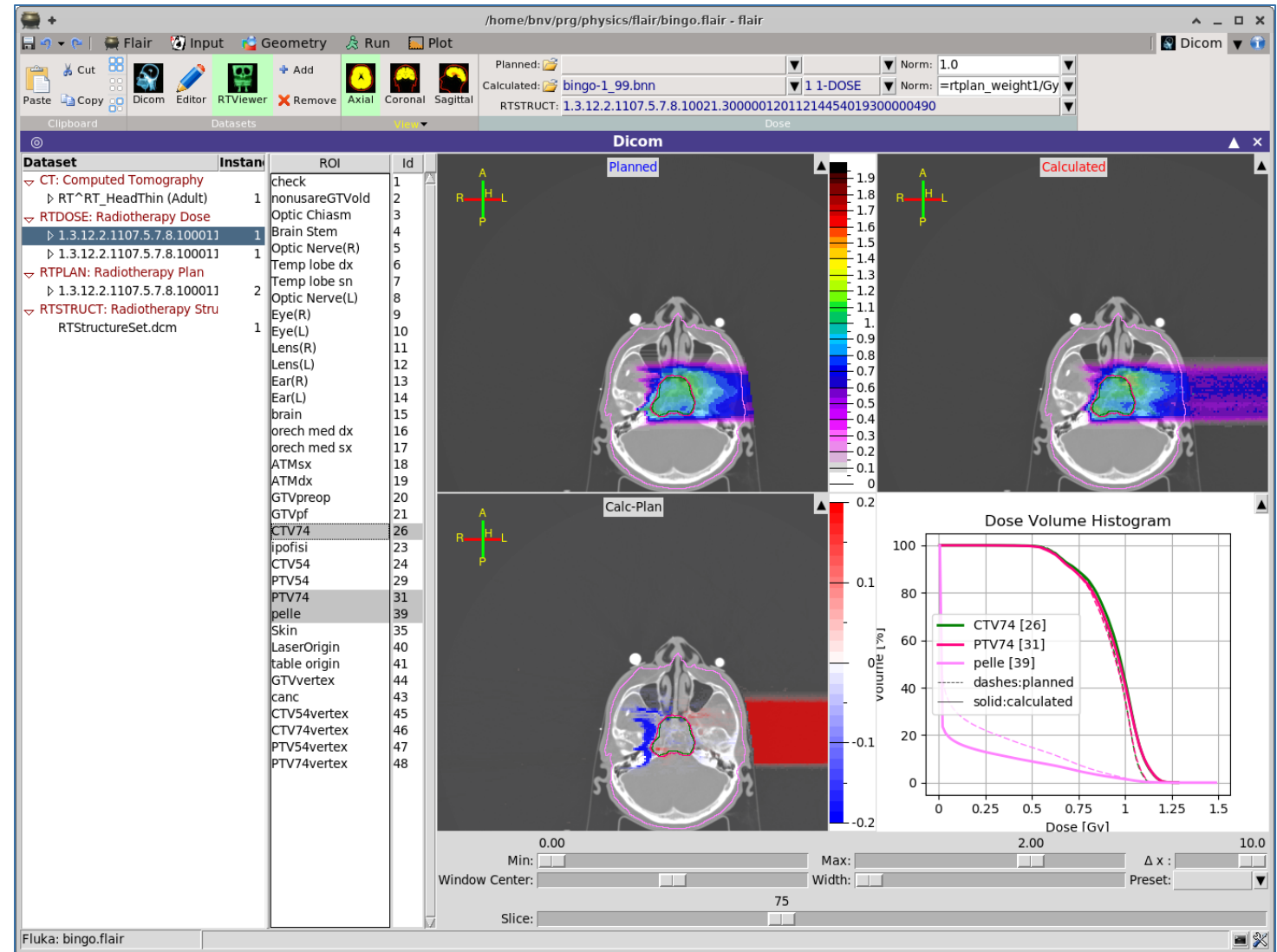
Medical applications

- The DICOM CT file together with RTSTRUCT can be converted to a FLUKA voxel geometry.

- This voxel geometry is contained in an RPP and can be placed within a larger combinatorial FLUKA geometry

Medical applications

- Schneider parametrization to convert HU in the CT file to actual materials (external file in flair website). This conversion is machine-dependent!

- Ranges of HU are assigned to various materials. To smoothly account for density variation with HU, Flair applies additional correction factors. Additionally, one may manually apply dE/dx correction factors

- The DICOM RTPLAN can be converted to a FLUKA beam input

- RTDOSE: the planned dose can be compared to USRBIN scoring

- Automatic generation of DVH (Dose Volume Histogram)

- Relevant cards: **VOXELS**, **CORRFACT**, **RAD-BIOL**, **TPSSCORE**

Medical applications & RP

- An intervention planner & path definition.

- This tool allows for calculating the dose incurred due to interventions in an activated area, taking into account not only working in specific places but also the traversal of activated zones.

- Workflows can consist of: routes, workplaces, times for traversal as well as activities

- Also instead of moving dosimeter, you can use it to define movement of radioactive sources or waste via **SOURCE** card combined with **source_newgen.f**

- Relevant functions: **:spline**, **USRBIN**, **SOURCE**
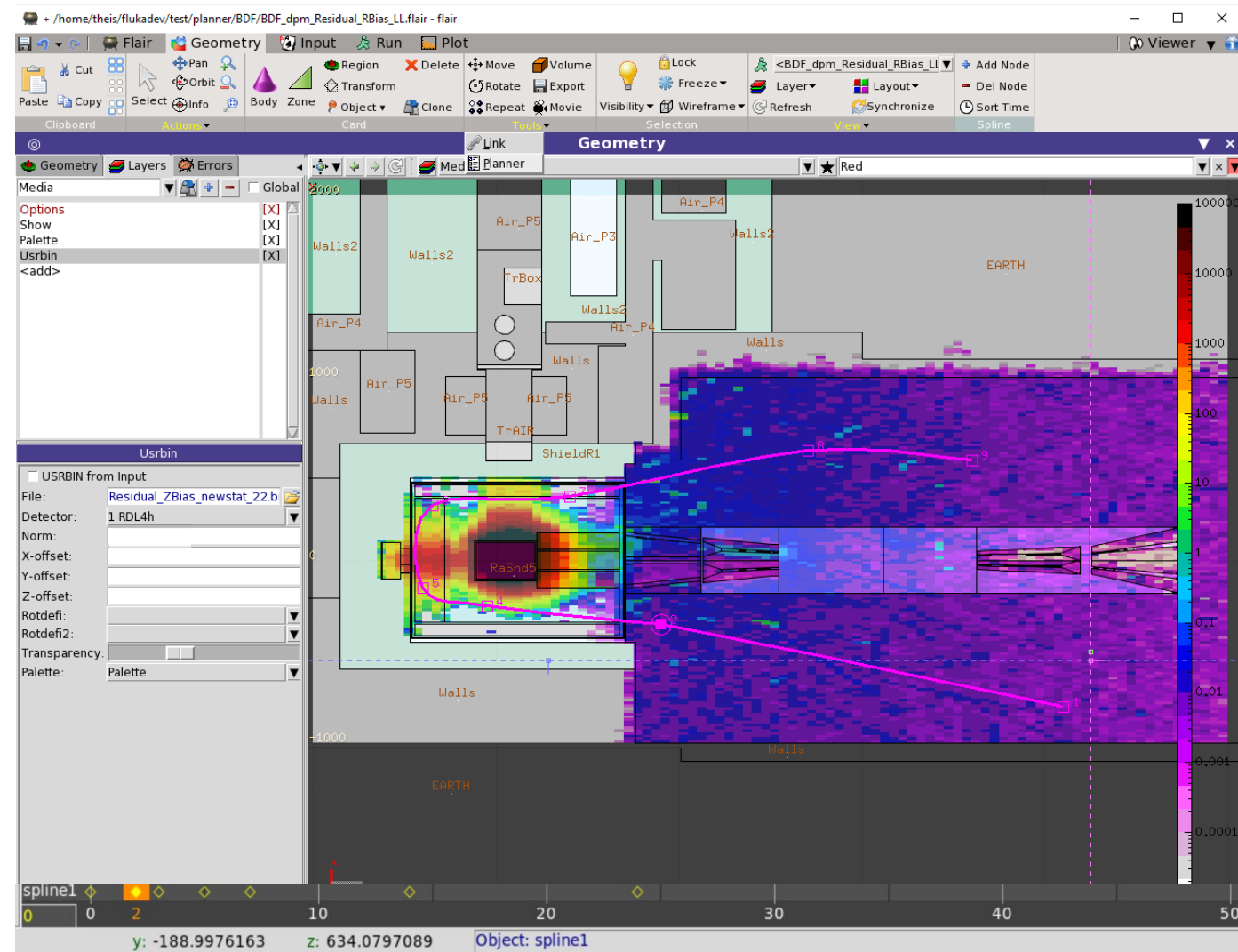
Special sources

Type: PPSOURCE, BIN-SOUR

Radiation sources

- Radiation sources we've seen thus far:
    - Particle beams
    - Radioactive isotopes

- FLUKA offers special sources:
    - Hadronic collision as a source term (p-p, p-Pb, etc)
    - USRBIN from another simulation as an extended source
    - SPOTBEAM: multiple beams
    - Cosmic rays
    - …

- And, if you want full customization: `source_newgen.f`

Special sources – *Colliding beams*

Input card: `SPECSOUR` – **PPSOURCE**

Beam momentum and direction:

- Based on the **Type**, 3 different options

       Type: **PPSOURCE**

- Momentum and direction are defined with the **x**, **y** and **z** components of the **total laboratory momentum** of either beam [GeV/c]

Special sources – *Colliding beams*

Input card: `SPECSOUR` – **PPSOURCE**

Simulates a collision between two beams:

- 1st beam: Hadrons (including protons and heavier nuclei)
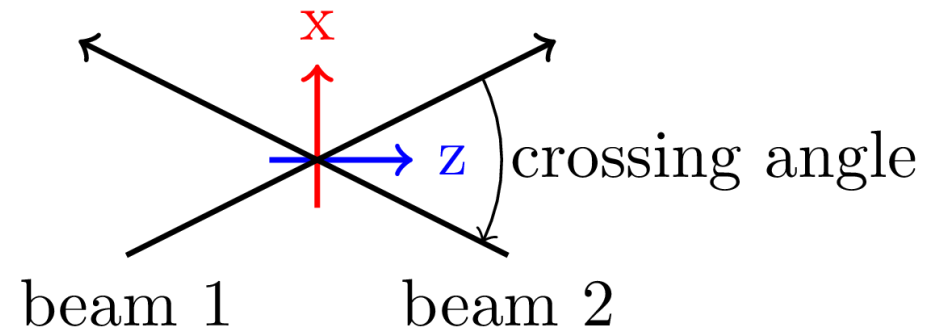- 2nd beam: Only proton or heavier nuclei

Two section of the card:

- **Top**: Beam momenta and directions of the two colliding beams
- **Bottom**: Volume of interaction, beam particles and divergences, physics interactions

Special sources – *Colliding beams*

Input card: **SPECSOUR** – **CROSSSYM**

Beam momentum and direction:

Type: **CROSSSYM**

beam 1   beam 2

x   z   crossing angle

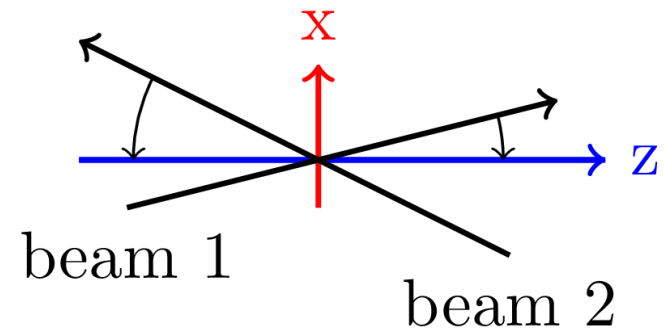| 🔥 **SPECSOUR**	Type: CROSSSYM ▾		
	Plab:	CrossAng/2:	Azimuthal:

- **Plab**: **Total laboratory momentum** of beam 1 and 2 [GeV/c]
- **CrossAng/2**: Half of the crossing angle [radians] (between: 0 … π/2)
- **Azimuthal**: Azimuthal angle defining the crossing plane [*degrees (!)*]

Special sources – *Colliding beams*

Input card: `SPECSOUR` – **CROSSASY**

Beam momentum and direction:

Type: **CROSSASY**

| SPECSOUR	Type: CROSSASY ▼		
P1lab:		Polar1:	Azimuthal:
P2-lab:		Polar2:	

- **P1lab, P2lab**: **Total laboratory momentum** of beam 1 and 2, respectively [GeV/c]
- **Polar1, Polar2**: Polar angle between beam 1 (2) direction and positive (negative) z direction [radians] (between: 0 … π/2)
- **Azimuthal**: Azimuthal angle defining the crossing plane [*degrees (!)*]

Example: LHC BLM signal benchmark

- Source term: p-p collision at several TeV per beam

- Beam-loss monitor (BLM) dose signal vs FLUKA prediction

- At various distances downstream (20 – 560 m)

- Note the excellent agreement!

• Fill #4919 (May 2016)
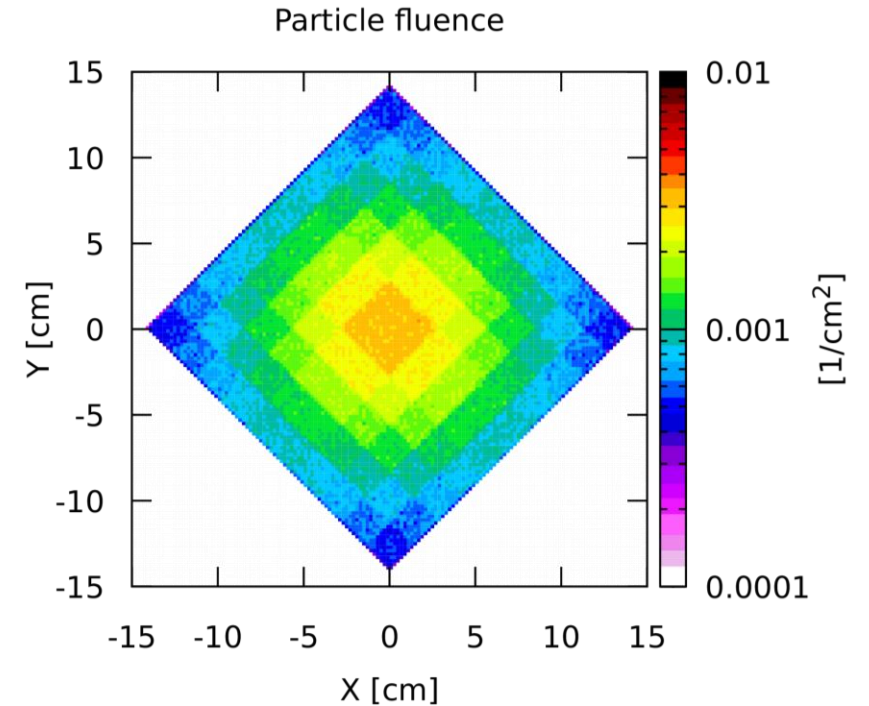

PHYS. REV. ACCEL. BEAMS 22, 071003 (2019)
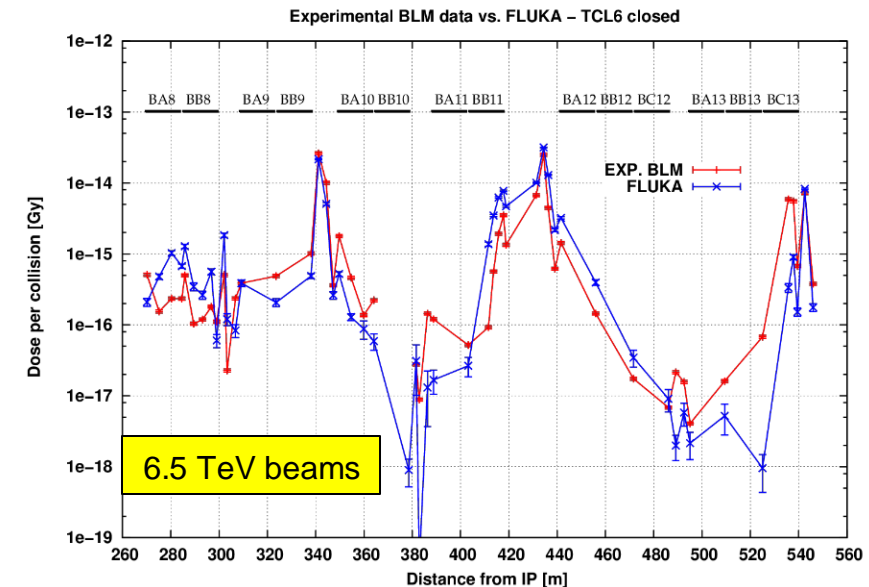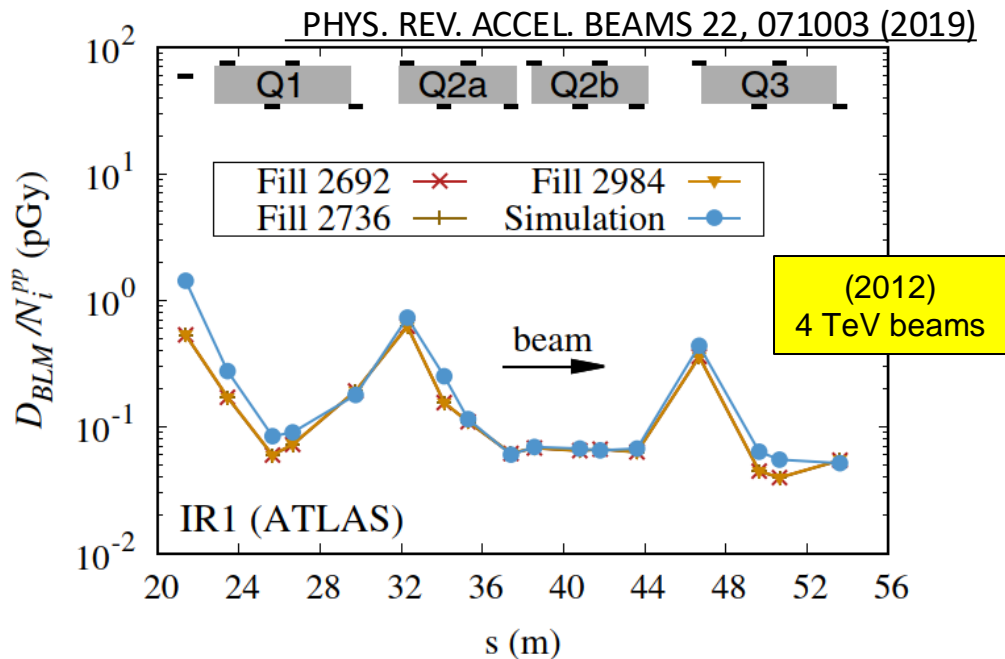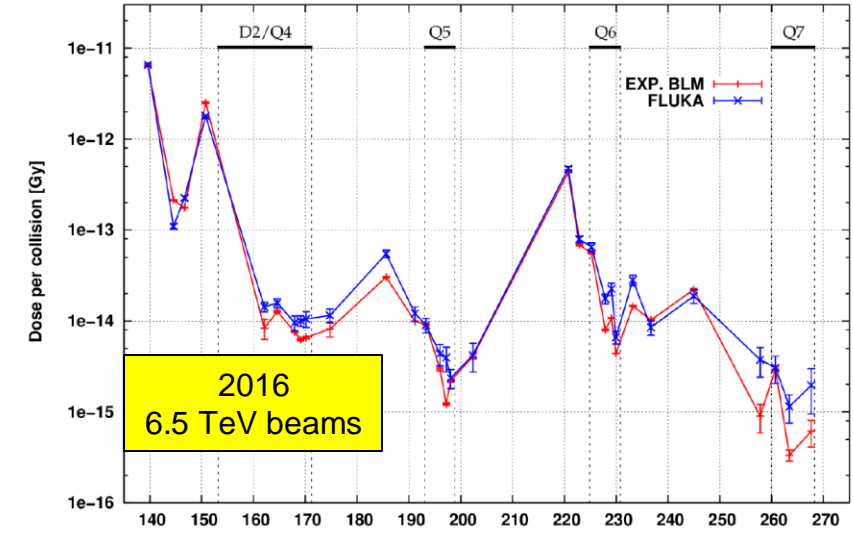
Special sources – *USRBIN source*

Input card: `SPECSOUR`

Example:

USRBIN file opened with logical unit 99 and rotated around the Z axis.

(Beam direction along Z axis)

Particle fluence

📂 **OPEN**	Unit: 99 BIN ▼	Status: OLD ▼	
	File: bin-sour_source.bnn ▼		
🔥 **SPECSOUR**	Type: BIN-SOUR ▼	Unit: 99 ▼	Det Id: 1
	Rot-before: Pre ▼	Rot-after: ▼	
⬦ **ROT-DEFI**	Axis: Z ▼	Id: 0	Name: Pre
	Polar:	Azm: 45	
	Δx:	Δy:	Δz:

Use case

- Suppose you have a PET image from a patient (^{68}Ga radionuclide source)

- What about dose to nearby healthy tissue?

- DICOM processed into a FLUKA VOXEL geometry

- RTDOSE can be converted to a `USRBIN` in Flair

- You can take the resulting `USRBIN` as a `BIN-SOUR` for such a simulation, putting ^{68}Ga as "beam" particle

- Positions are sampled from the `USRBIN` weights

- Directions as per `BEAM` card (you'd put isotropic emission)
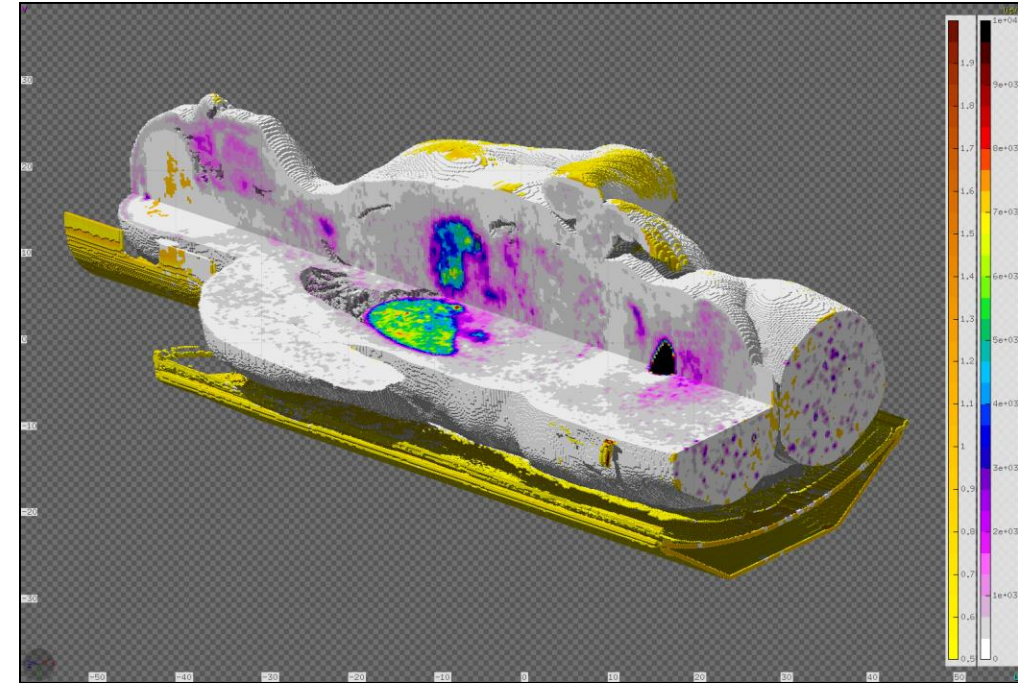
Conclusion

All good things…

- Many other advanced/specialised topics can be studied with FLUKA…
  - Cosmic rays
  - Neutrino interactions
  - Optical photons
  - Crystal channeling
  - …and more!

- …and a lot more flexibility can be achieved via user routines

- This course was meant to get you started on FLUKA (building geometries, defining materials and sources, scoring), while also introducing more advanced concepts and techniques (biasing, advanced geometry features, advanced sources)
  - We would like to hear your feedback, positive and negative!

- **Consider following a FLUKA advanced course!** ☺
  - **Programmable user routines, advanced geometry, tracking, scoring, optical photons, and much more!**

Lattice: basic usage

- Very useful for geometries where models are used multiple times (e.g. beamlines)
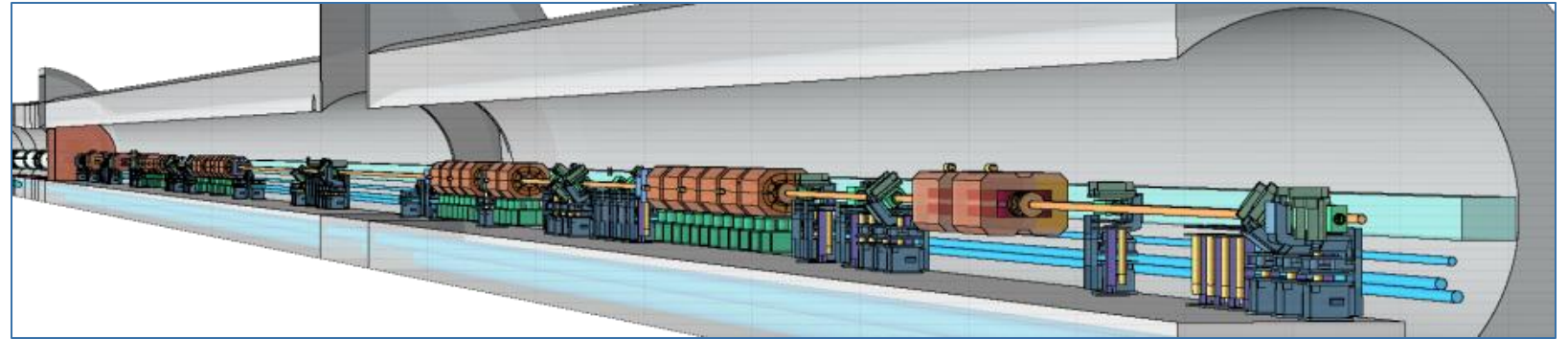
- The prototype is defined in detail with all the necessary information (geometry, materials etc.) inside a closed container body (`RCC`, `RPP` etc.)

- The lattices (replicas) are defined as "empty" regions in their correct location and declared as such via `LATTICE` card

- The transformations exactly mapping the replicas onto the prototype are defined using `ROT-DEFI` cards
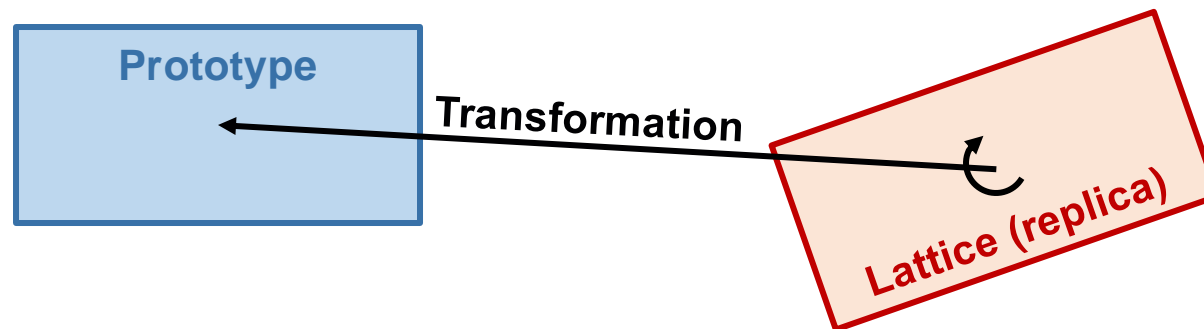
- *Note: You can load the lattice template in Flair for a simple working example!*

Lattice example

◇ **ROT-DEFI** Axis: Z ▾   Id: 0   Name: transf
           Polar: 0.0   Azm: 0.0   ⎫
           Δx: 0.0   Δy: 0.0   Δz: -30.0   ⎬ **A**
◇ **ROT-DEFI** Axis: X ▾   Id: 0   Name: transf   ⎫
           Polar: 0.0   Azm: 30.0   ⎬ **B**
           Δx: 0.0   Δy: 0.0   Δz: 0.0   ⎭
◇ **ROT-DEFI** Axis: Z ▾   Id: 0   Name: transf   ⎫
           Polar: 0.0   Azm: 0.0   ⎬ **C**
           Δx: 0.0   Δy: 20.0   Δz: 10.0   ⎭

Note: if a transformation R consists of 3 roto-translations A, B and C applied in this order, then $R = CBA$ and the inverse transformation is $R^{-1} = A^{-1}B^{-1}C^{-1}$

Here, R maps the replica onto the prototype, whereas R^{-1} is used to transform the replica container from the prototype to the replica

A — Translation to origin

B — Rotation around x-axis

C — Translation to prototype position