Flair Intermediate

An invitation to discover further features

Starting flair

- `flair -h`

  shows command line arguments

- Two examples of interesting options:
  - `-r` loads the last opened project
  - `-l` shows the last 10 opened projects
- [filename] doesn't need to be complete,

  flair will find the best match, e.g.

  `flair example.fl`

  is equivalent to

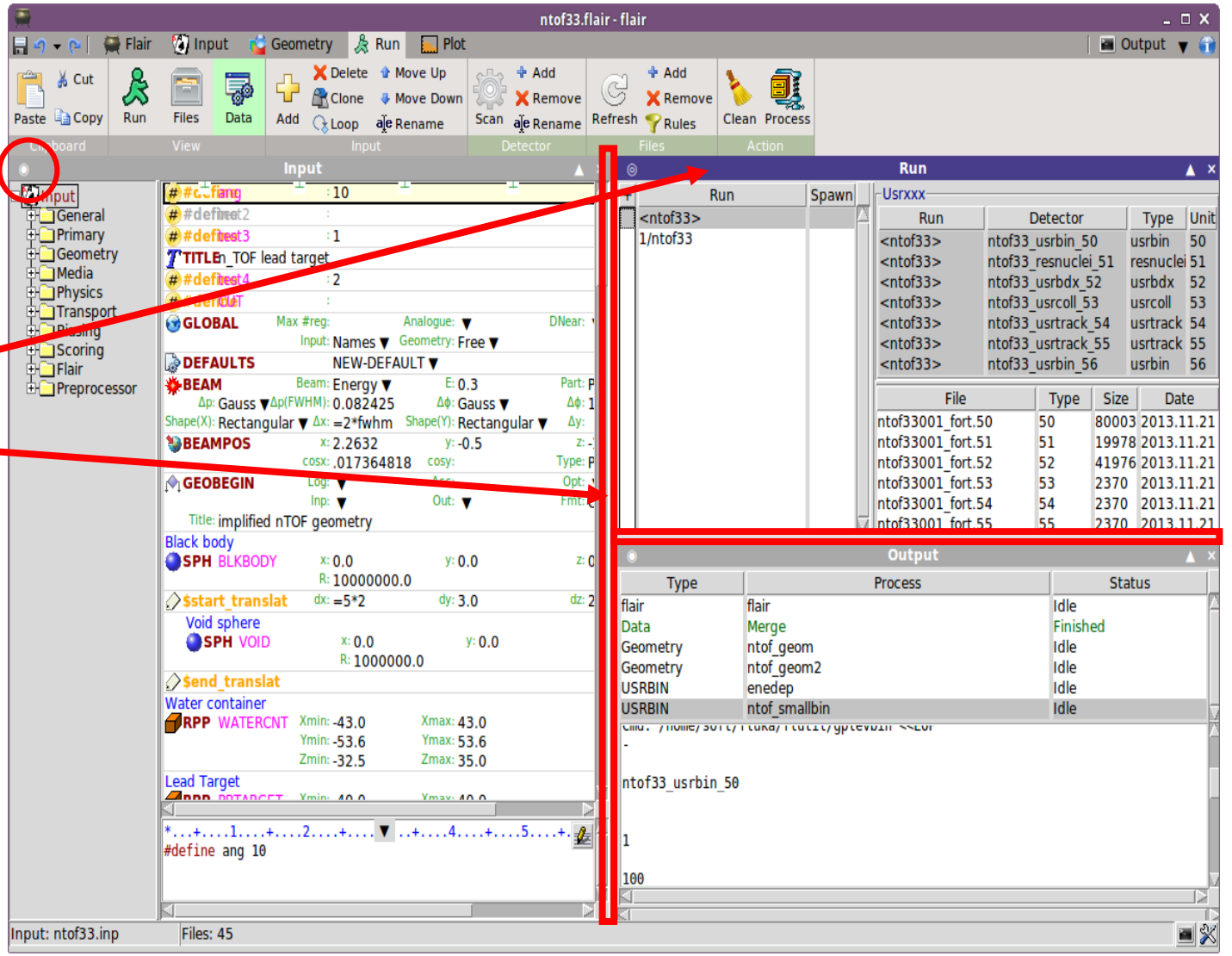
  `flair example.flair`

```
Usage: flair [options] [filename]...
 Options:
-1          Load the first flair file in the folder
--compile   Compile executable
-d/-D       Activate/Deactivate the beta-development features
--data #    Process/Merge the data files of all or specific
            runs (no default). Accepts patterns or comma
            separated names e.g. *  or foo* or foo,bar*
--exe file  Fluka executable. (default: {FlukaDir}/bin/fluka)
-h|-?|--help print this help page
-i/--ini file           Alternative configuration file
            (default: $HOME/.flair/flair.ini)
-l | --list List recent projects
-p/-P       enable/disable profiling
-m #        Open a new project in mode
--plots     Do all plots and save the files
-r|--recent Load most recent project
-R #        Load recent project (number 1..10 or filename)
-s          Skip About dialog
-t # | --type # force import file type to load if different
            than .flair. Accepted types: flair fluka gdml
            mcnp moira penelope pickle
-u | --update           Recalculate and save input file variables
-v/-V | --verbose       Increase/Decrease verbosity level
--noansi    Disable ansi coloring on dumps

filename[s] flair project, input, imported files or directory
automatically detect type from file extension
extensions supported: <none|.flair>, .inp, .fluka, .moira,
.mac[ro], .gdml, .pickle, .mcnp
```

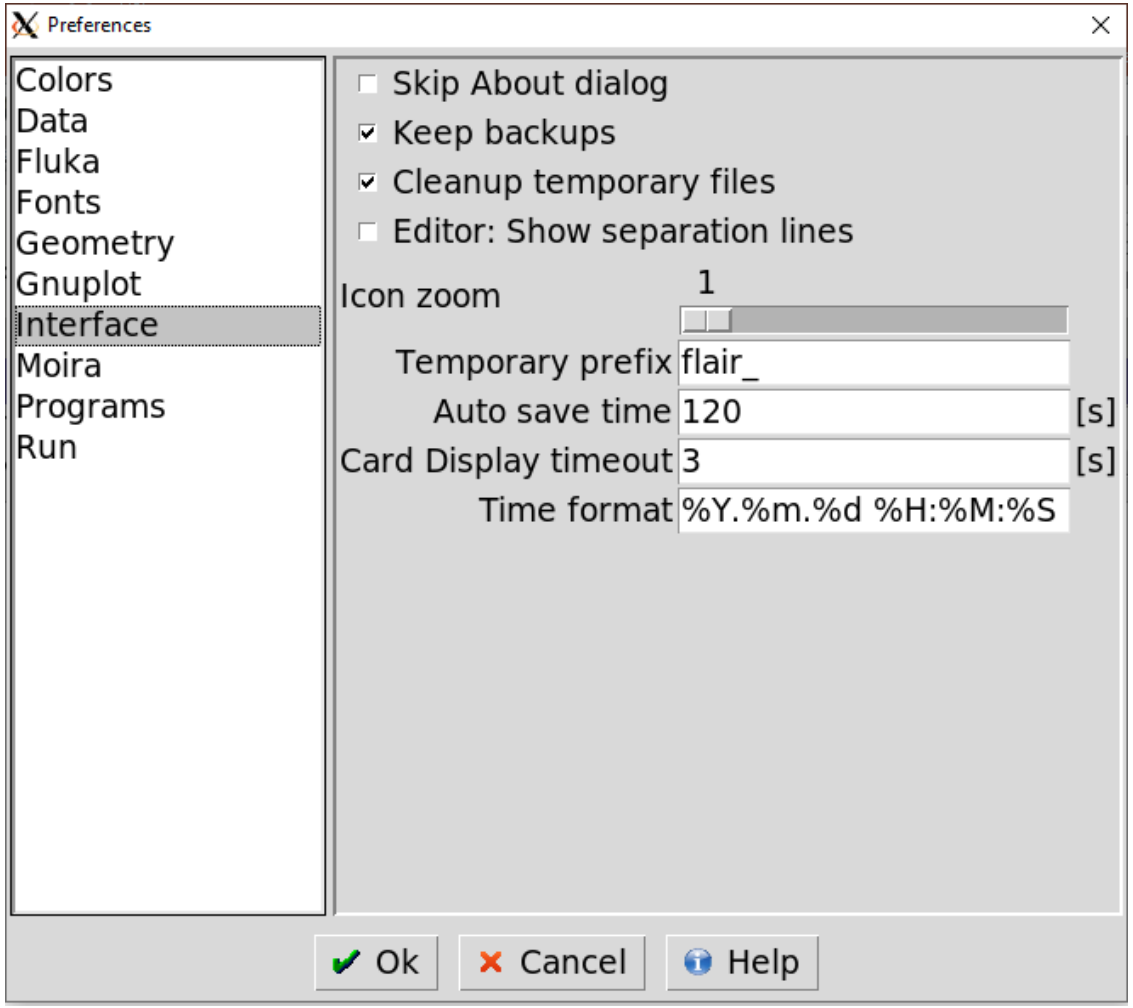
Interface customization – Undocking & Tiling

- Tiling
  - Multiple tabs can be displayed at the same time
    1. Pin the tab(s)
    2. Select a new tab
    3. Drag the tab title bar to rearrange
    4. Resize the separation
- Undocking
  - Drag the tab outside flair to open it in a new window.



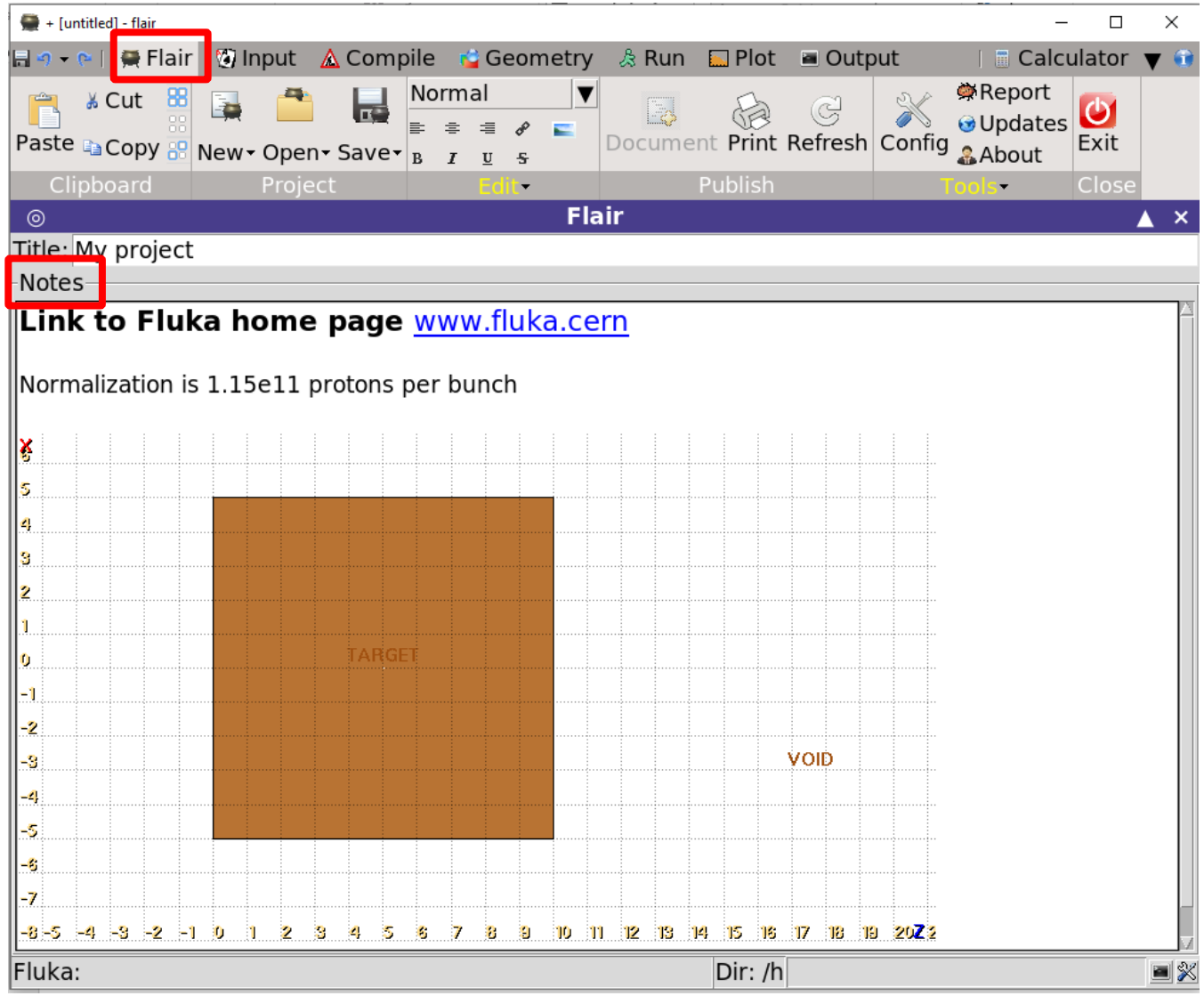
Configuration

- Almost everything can be configured
  - Colors — all colors used by flair
  - Data — processing programs and filters
  - Fluka — Fluka programs configuration
  - Fonts — all flair fonts (except Geometry)
  - Geometry — CPUs and precision
  - Gnuplot — global commands and terminal
  - Moira — [still in development]
  - Programs — editor, terminal, debugger…
  - Run — Spawning rules & Queues



Project notes

- Notes can be included in the project
- Basic formatting capabilities
- Plots and geometry viewports
  can be exported to the notes

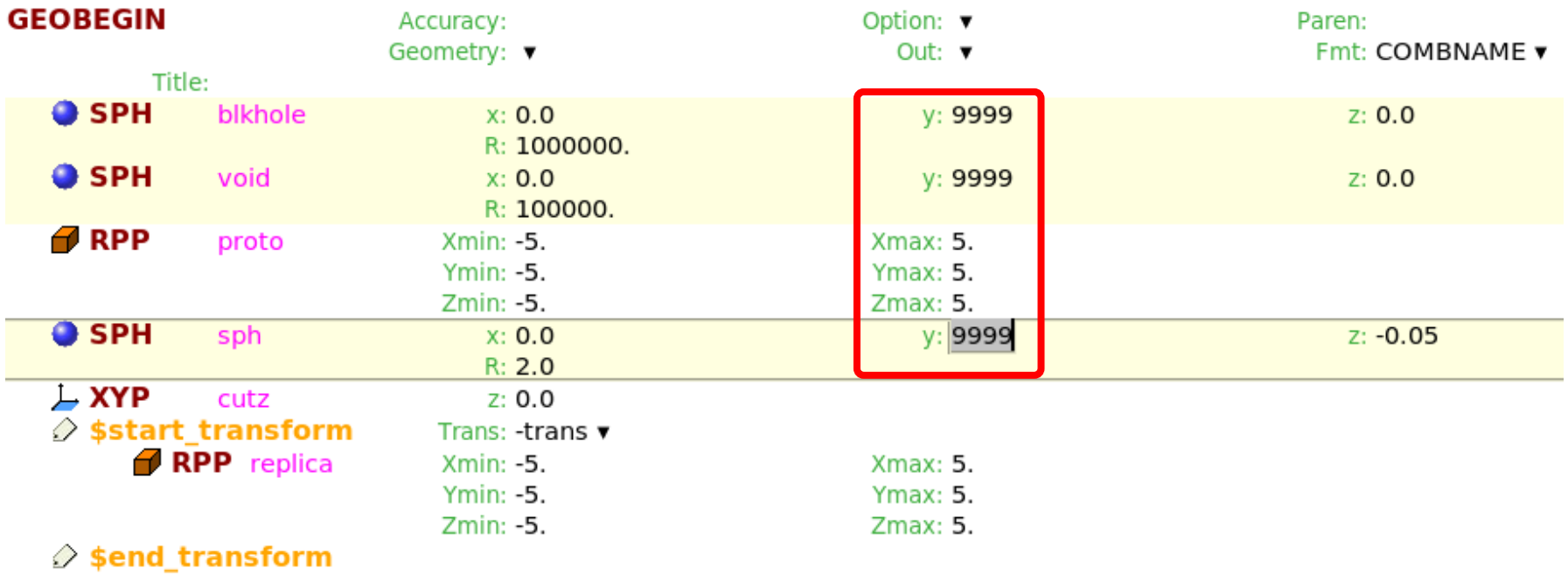


Input tab

Input editor: multi-card editing

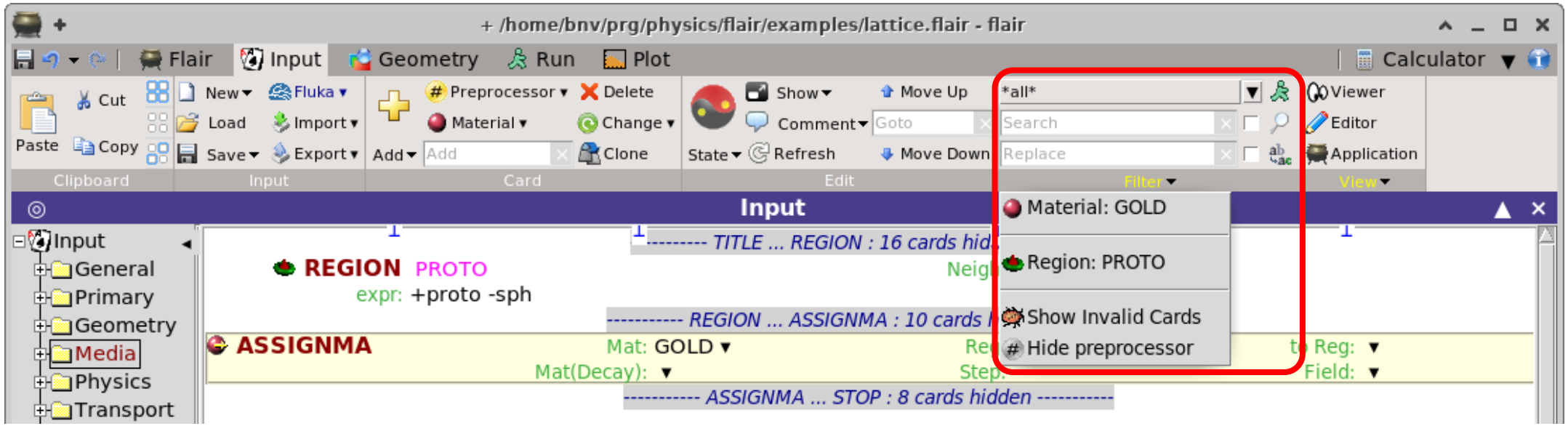
- Possible to edit multiple card at the same time
- Any change in the selected cards will apply to all similar card for the same what

  e.g.: very useful when changing the binning of multiple scoring cards
- Two undo commands registered
  - 1st for all the other cards, 2nd for the currently active card



Input editor: filtering cards

- Possible to filter cards for a specific run or all runs (different **`#define`**, different cards)
- Possible to filter cards by string, body, region, material, transformation, detector
- Possible to filter cards with errors
- Possible to filter cards “related” to the selected one



Input editor: expressions

- It is possible to specify values using expressions
- Possible to make parametric runs
- Fields starting with “=“ will be evaluated by flair, e.g.:

```
BEAMPOS      x: =2+10*length
```

- Expressions are stored in the `.flair` file
- Expressions are also stored in the `.inp` file as comments, e.g.:

```
!@what.1=2+10*length
```

- The cards in the `.inp` file contain the evaluated values

Do not change by hand, they will be overwritten by flair!!!

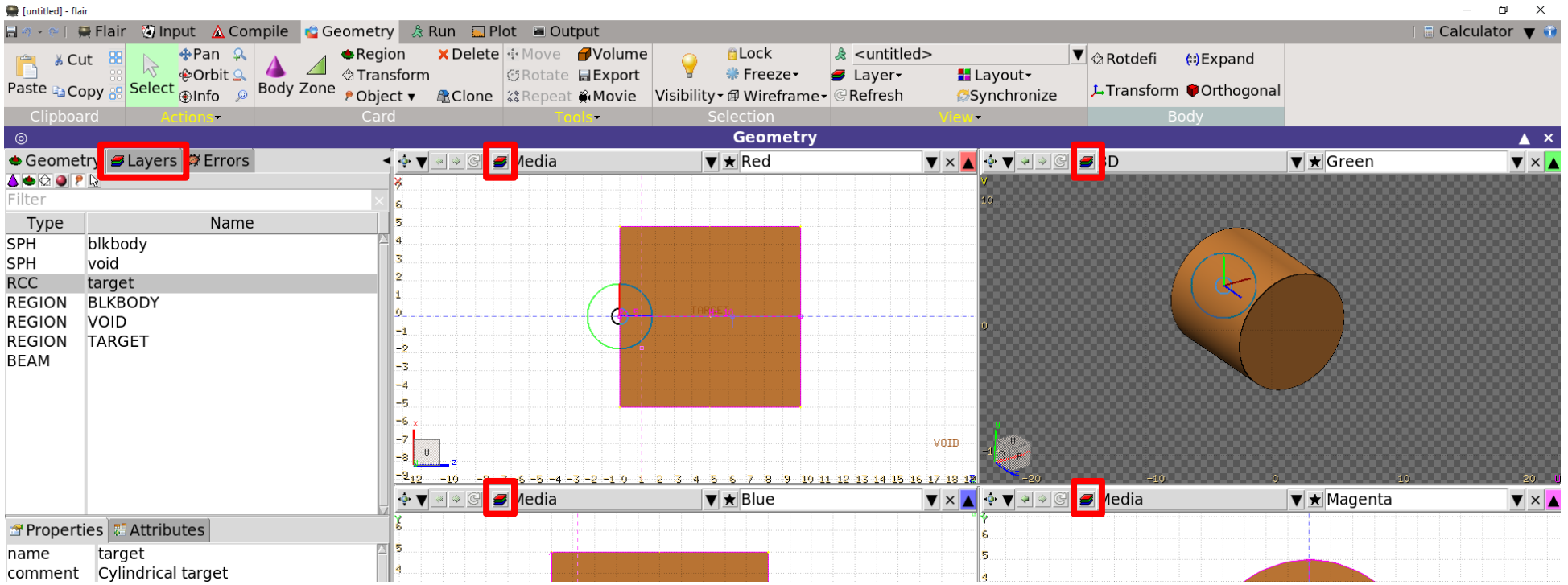
Input editor: expressions

- See manual F3.6} for details
- Useful predefined quantities
  - Units, e.g.: *MeV, mm, ms…* (warning: only threated as conversion factors)
  - Constants: *fwhm, c, qe…*
  - Particle masses: *Mp, Me…*
- All common mathematical functions: *sin(x), cos(x), exp(x)…*
- Some physics functions
- Card reference functions
  - *what(n)*
  - *body(name,what)*
  - *card(tag,sdum/id, what)*

Geometry tab

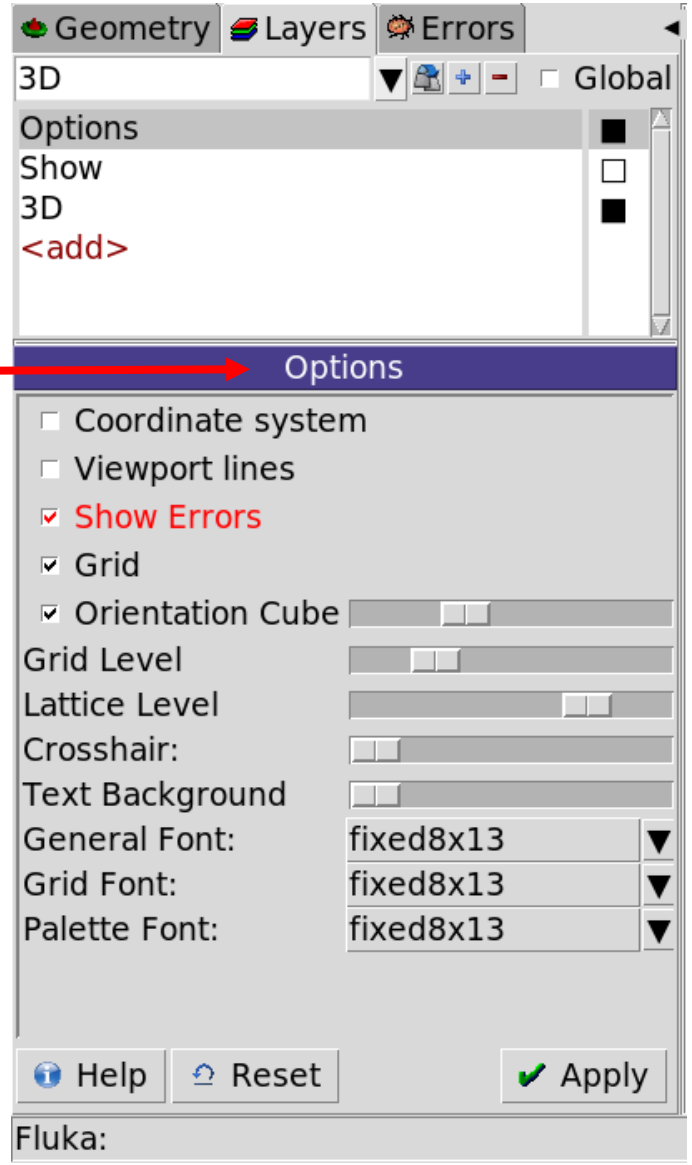
Geometry tab: Layers

- Somewhat introduced in the first Flair lecture
- Custom layers can be defined in the “Configure layer menu” 



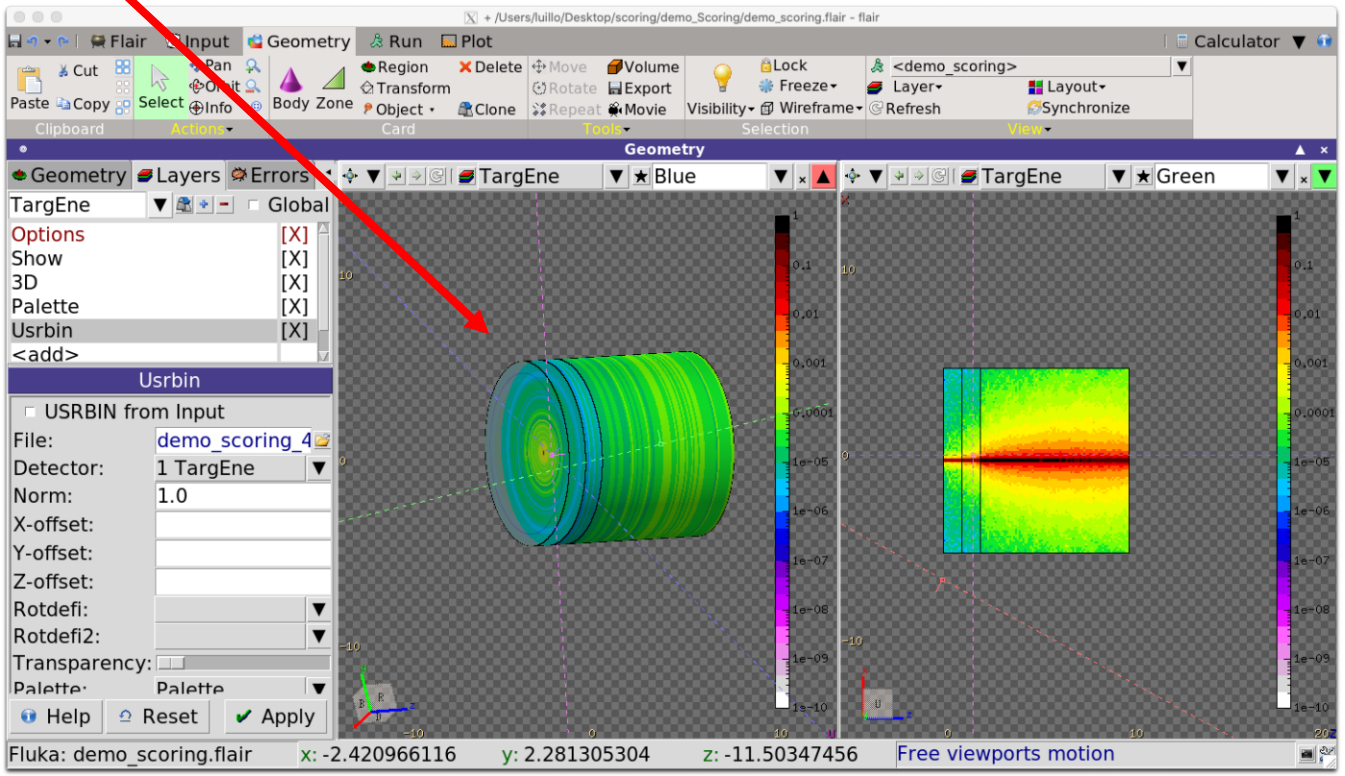
Geometry tab: Layers

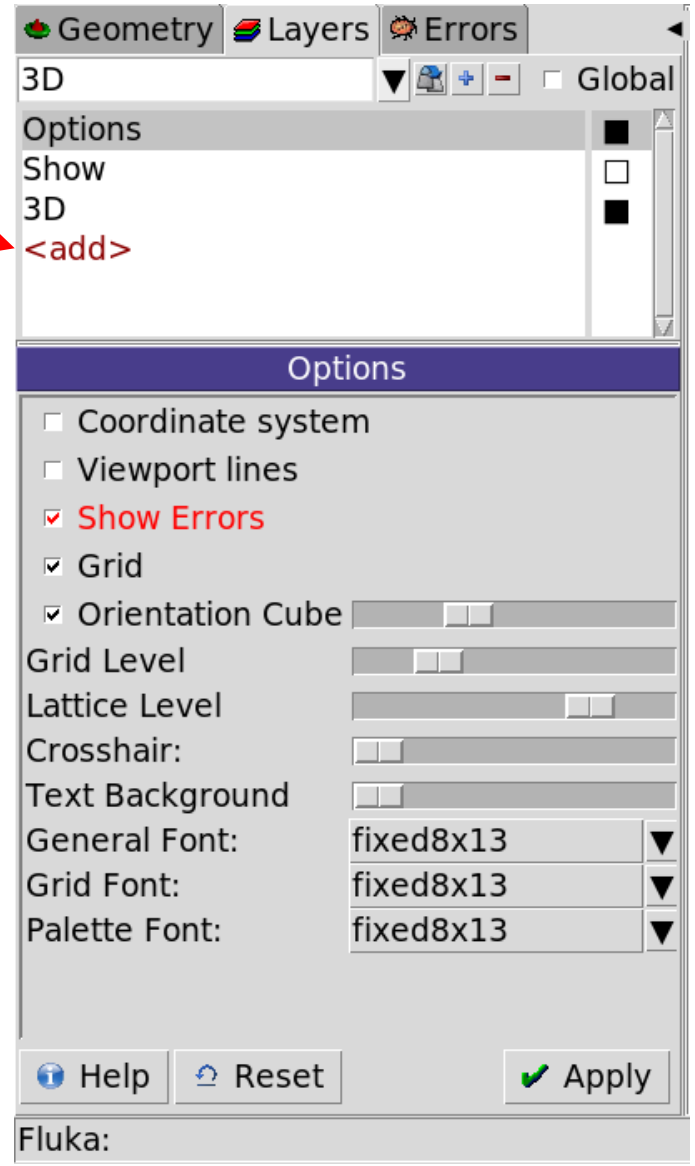
- Toolbar
  - Add / delete / rename / clone layers
  - Global: to make a layer available on every project
- Options
  - Enable/disable
    - coordinate system, viewport lines, grid, orientation cube
  - Adjust:
    - Grid level (set grid intensity)
    - Lattice level (set lattice hash intensity)
    - Crosshair (set dimension of the cross in the viewport center)
    - Orientation cube size
    - Fonts



Geometry tab: Layers combination

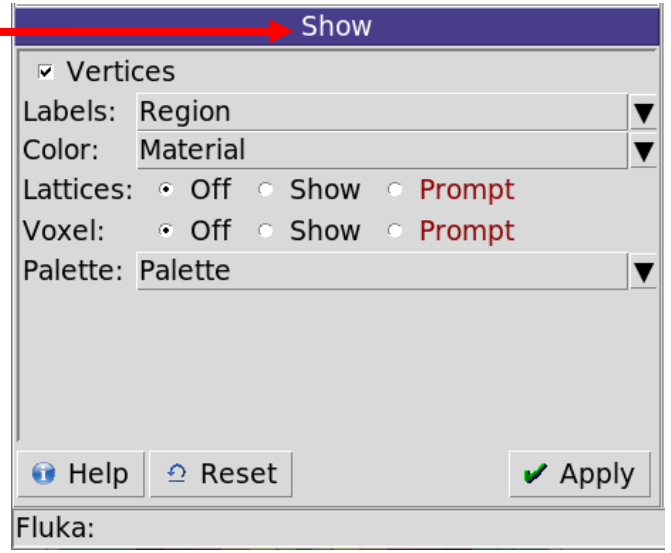
- “(Sub-)layers” can be combined together via <add> button
  - Image & USRBIN
  - Custom color values & 3D
  - USRBIN & 3D
  - …

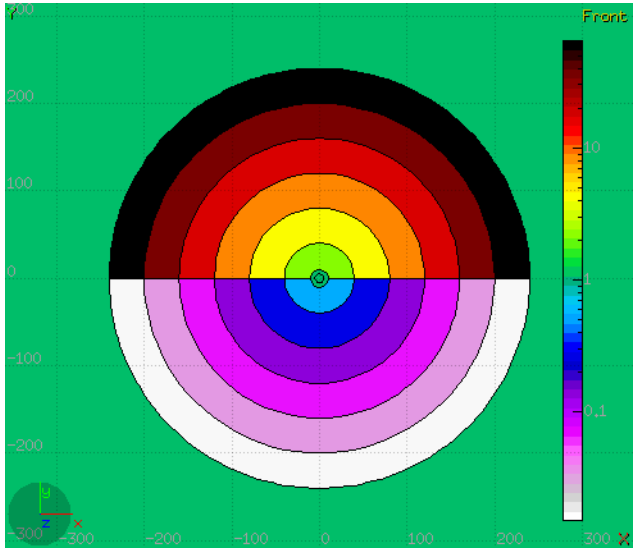




Geometry tab: Layers

- Show (to set visualization details)
  - Visibility of Vertices, Lattices, and Voxels
  - Set label (Region, Material, Value, or none)
  - Choose palette
  - Associate region color to:
    - Regions
    - Material
    - Density
    - Region Importance
    - Thresholds
    - …

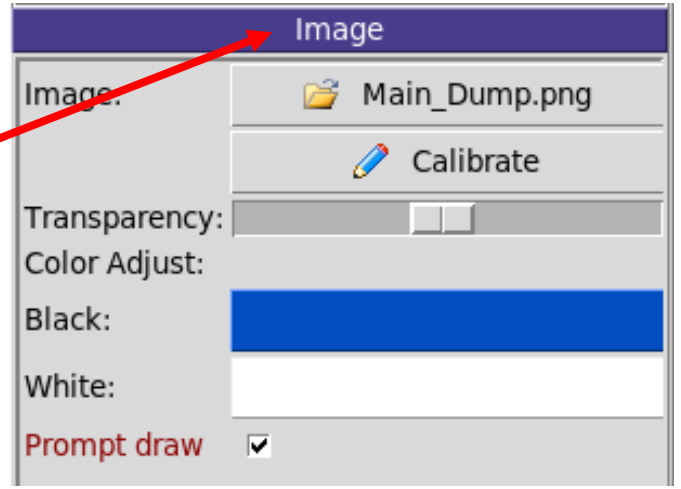


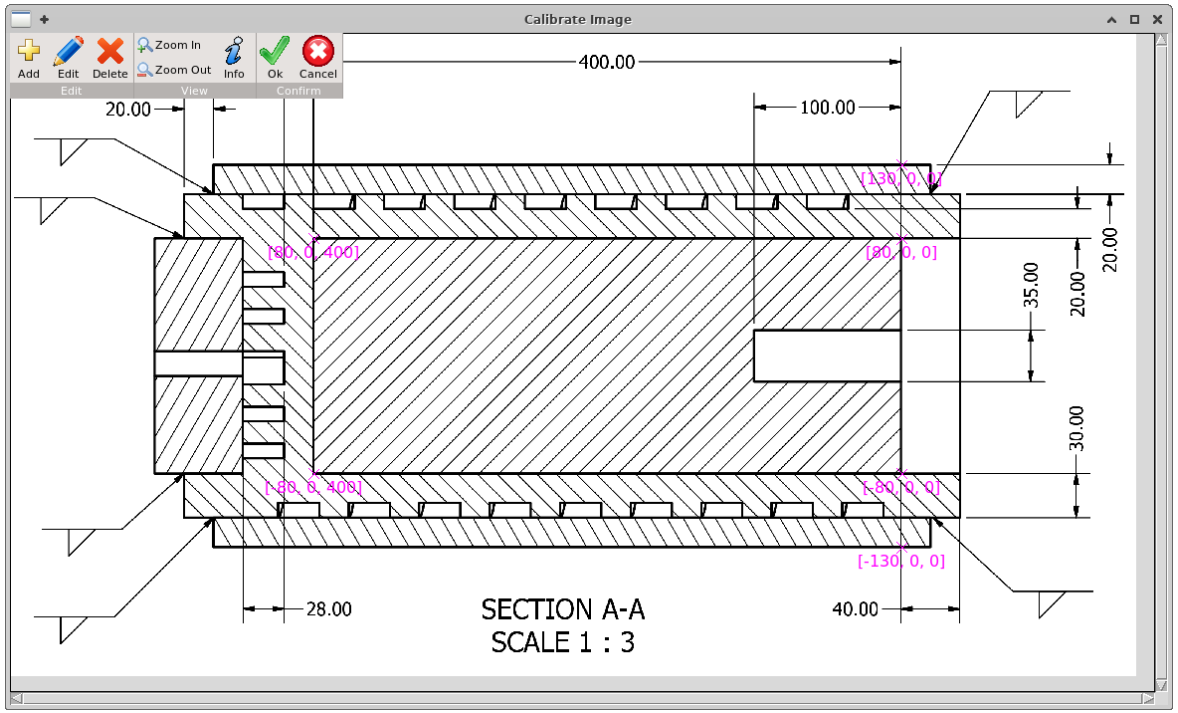


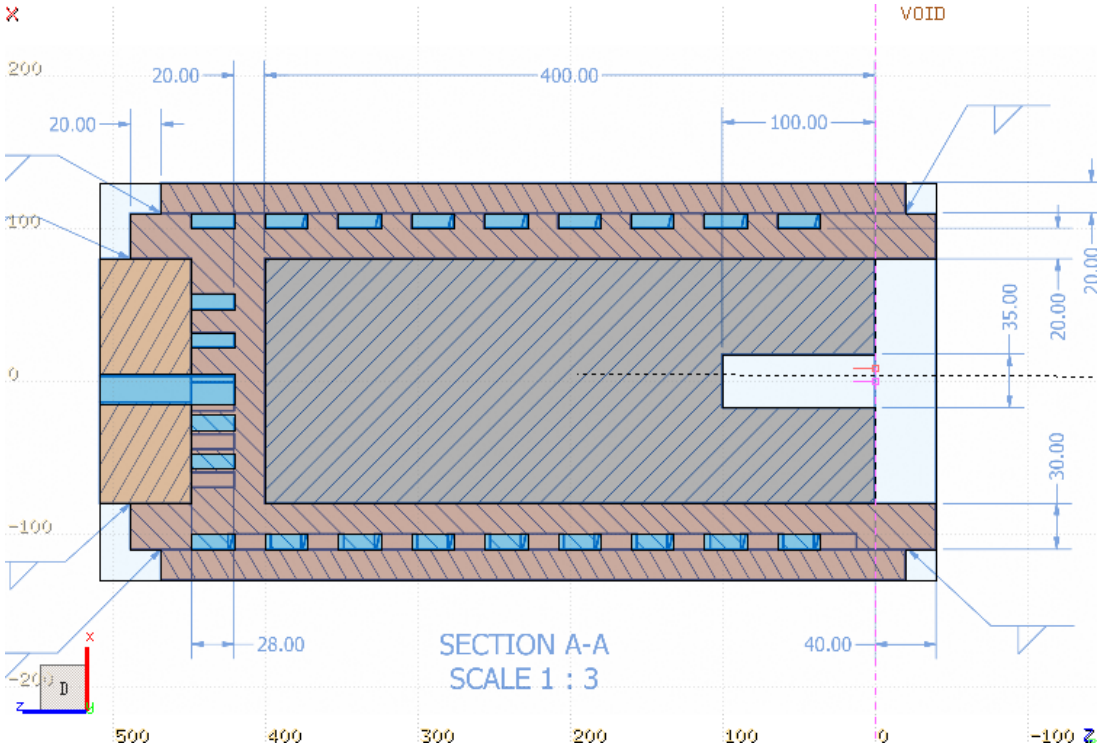
Geometry tab: Layers

- Image (to set a background image, i.e. a CAD drawing)
  - Image: select **`.png`**, **`.gif`**, or **`.jpg`** file
  - Calibrate: to input coordinates of specific points
  - Prompt draw: immediate image draw for idle display

    (warning: it slows flair)

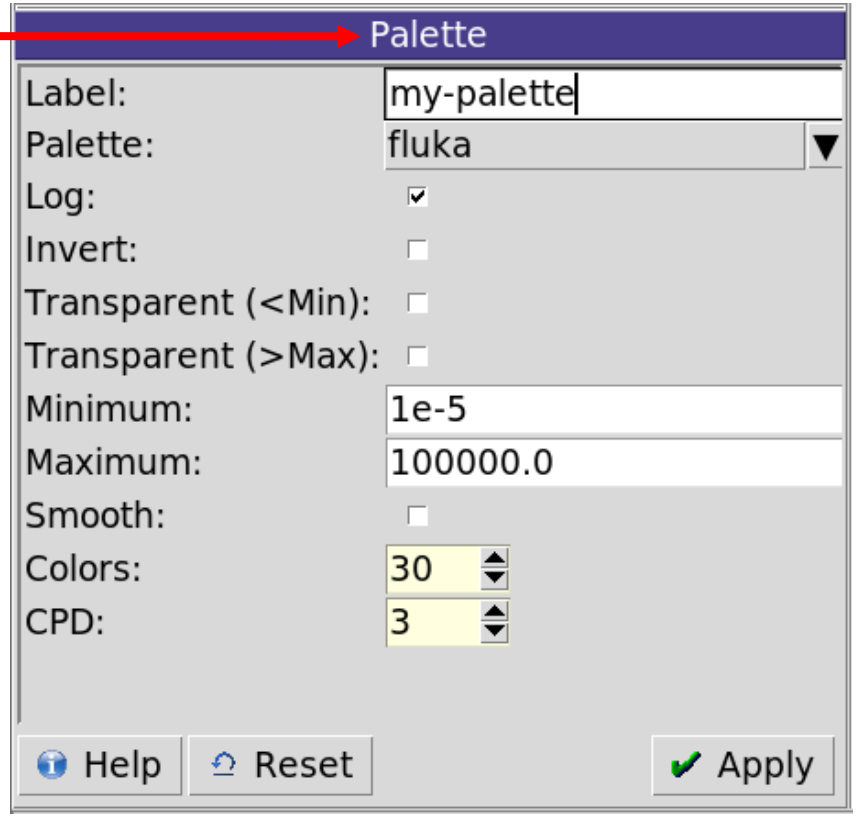






Geometry tab: Layers

- Palette (to set palette properties)
  - Assign label
  - Change palette colors
  - Set linear or log scale
  - Set minimum and maximum values
  - Set color range

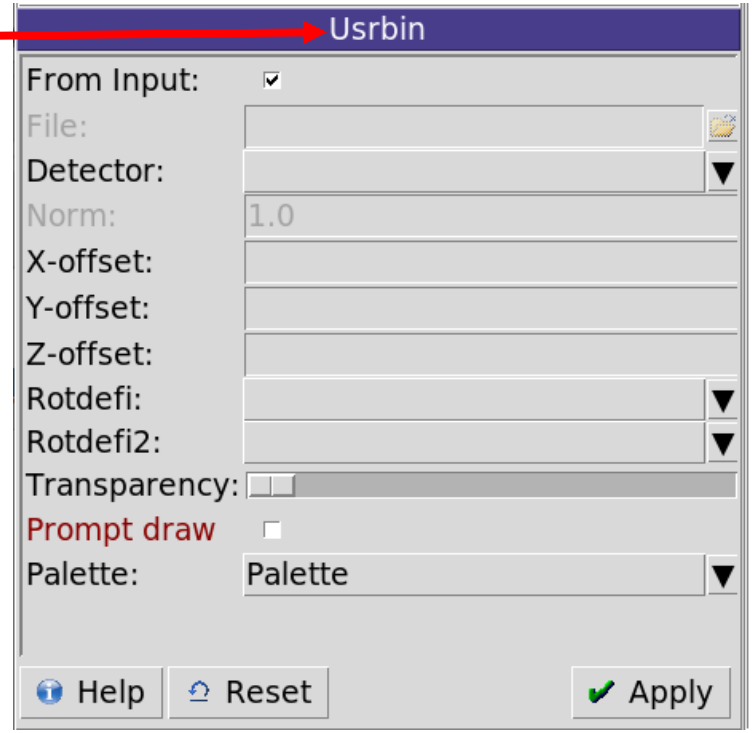


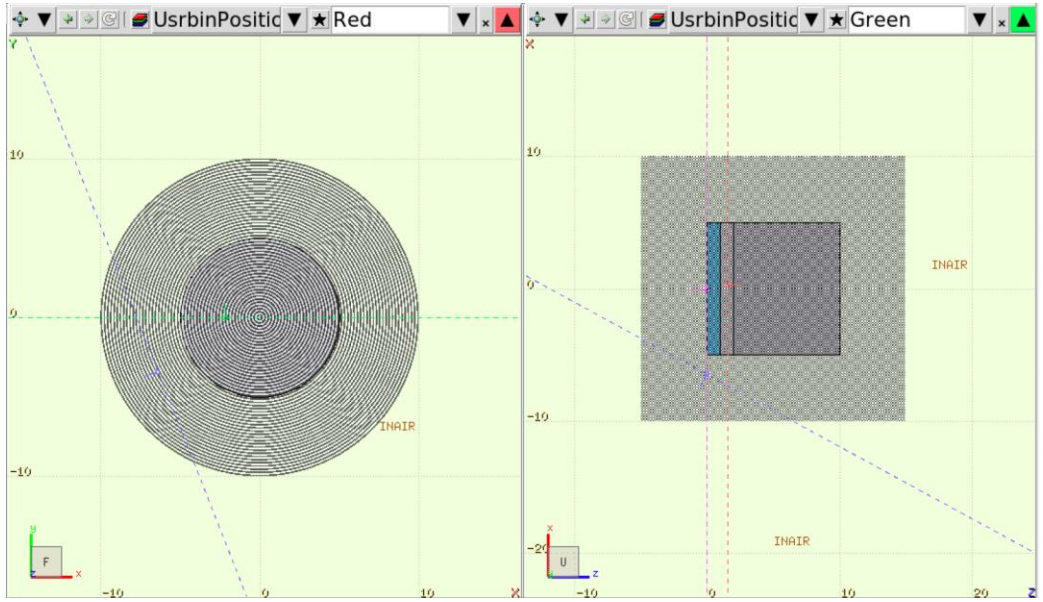
Geometry tab: Layers

- USRBIN (to show a USRBIN scoring)
  - Already seen in the first Flair lecture
  - Possible to add more USRBIN on the same layer
  - Possible to load a USRBIN from the input

    Displayed with a checker pattern

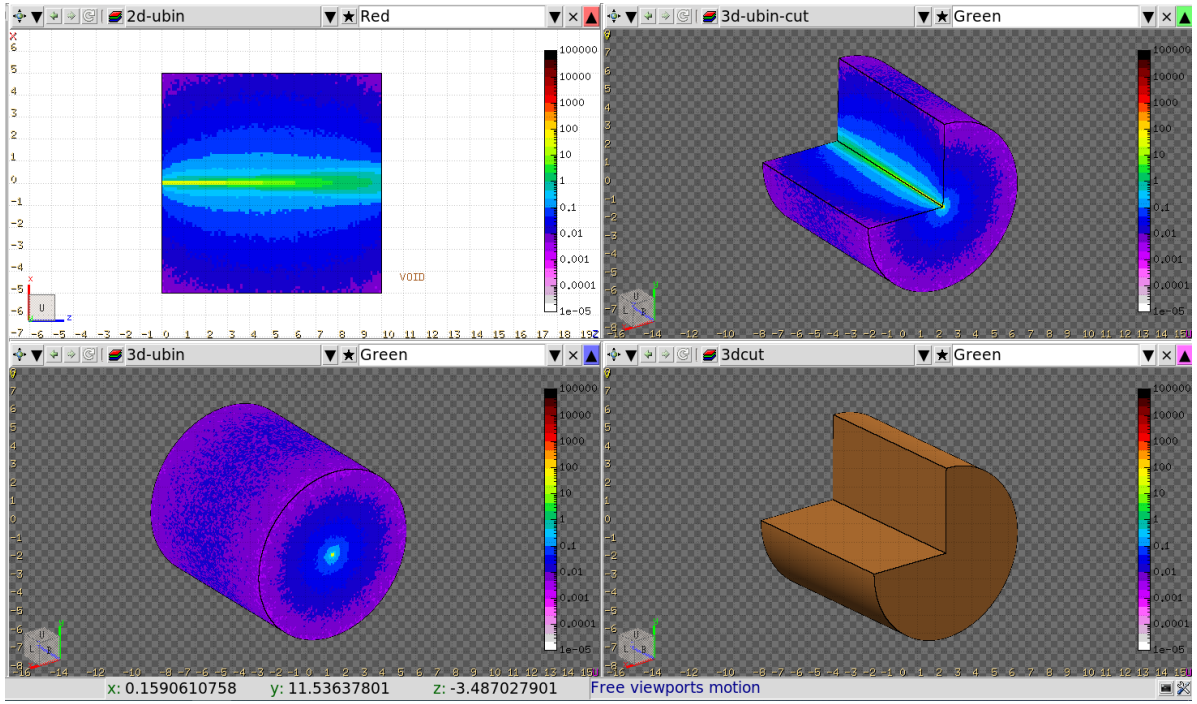
    Useful to the its location before running

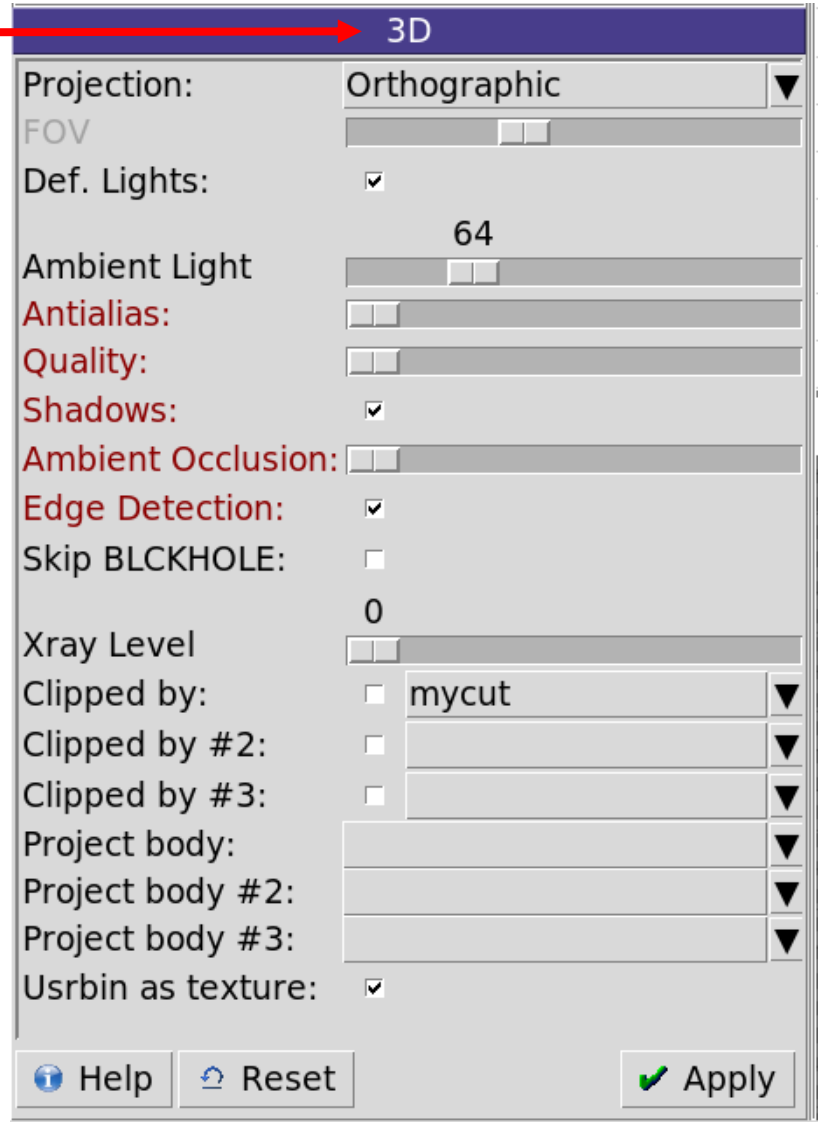




Geometry tab: Layers

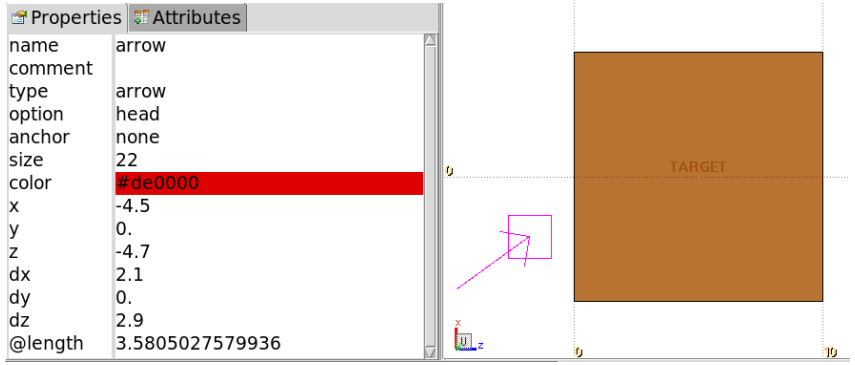
- 3D (to enable 3D rendering)
  - Plenty of options
  - Red indicate time consuming options
  - Select up to 3 bodies to cut the view
  - Project USRBIN info on up to 3 bodies

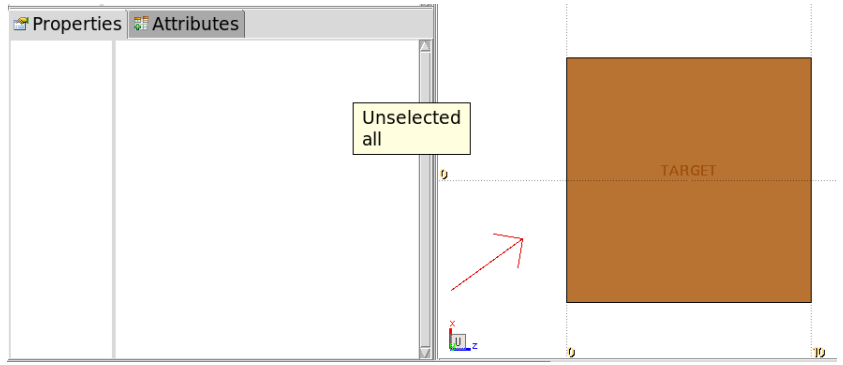




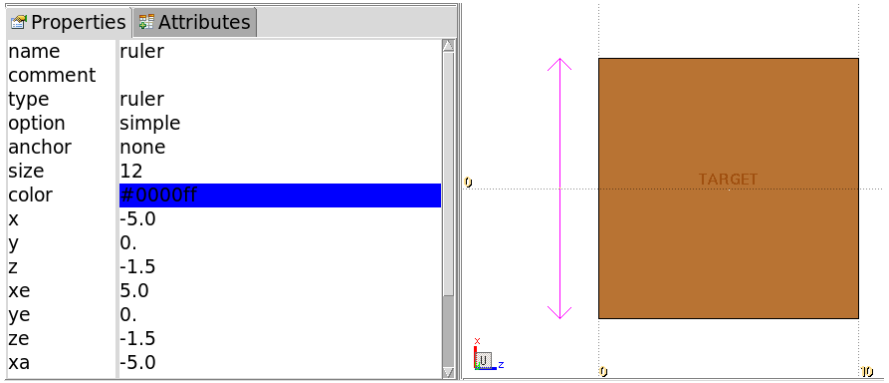
Objects

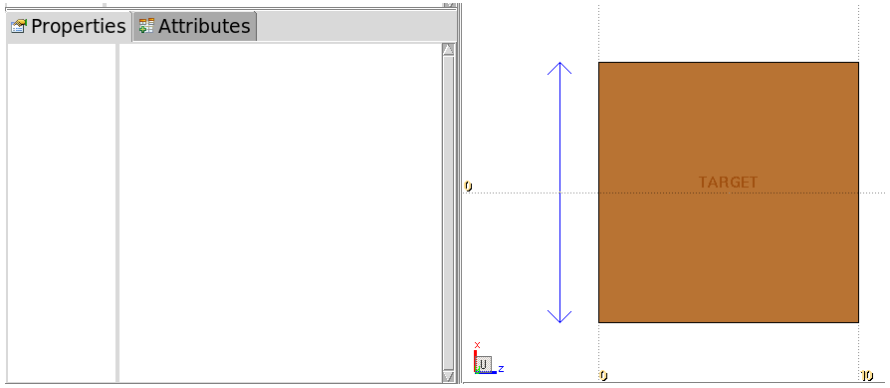
- Arrow
  - Provides basic drawing/pointing means
  - Can be used as snapping point





- Ruler (simple or angle)
  - Used to measure distances and angles
  - Can be used as snapping point
  - Used to project snapping point to different locations





- Point
- Light
- Camera
- Spline

Material database

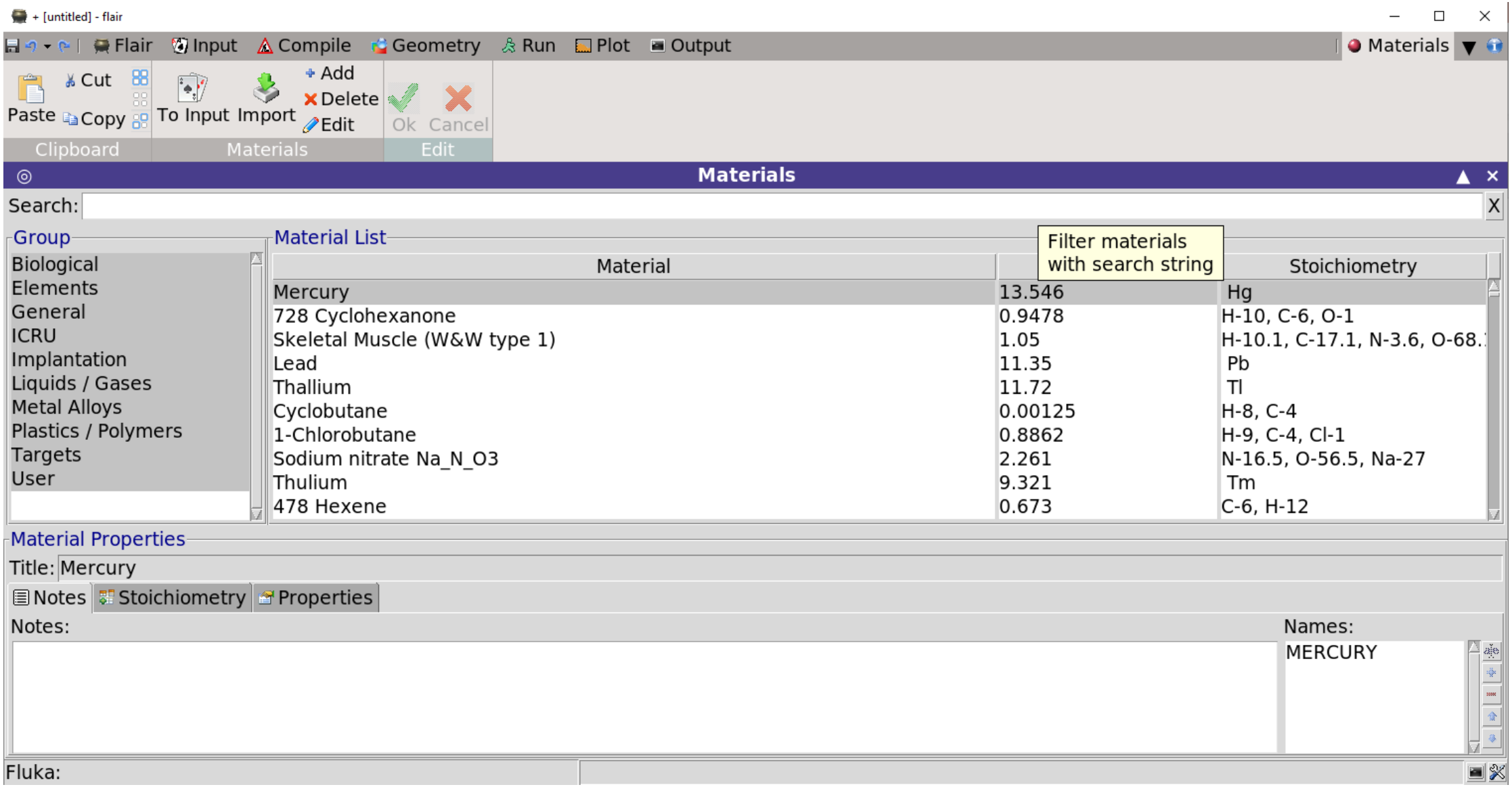
Material Database

- Flair has an internal database of ~500 predefined materials/compounds
- About 300 with Sternheimer parameter

Provided as reference only

- User must always verify the correctness of the data
- The database can be edited and populated with user-defined materials/compounds
- The database can be shared among different flair projects
- Materials can be edited by hand or imported from existing inputs

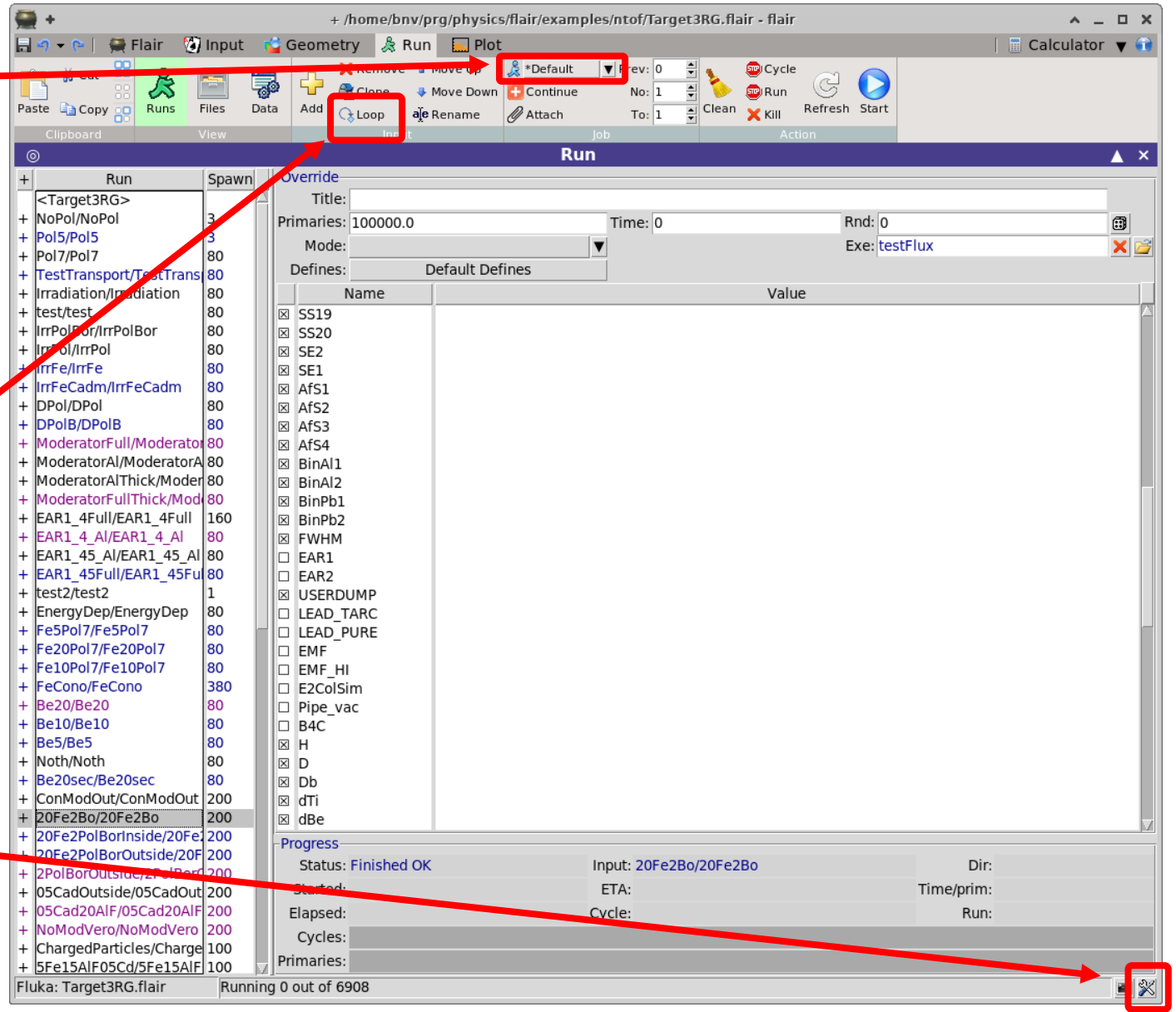
Material Database



Run, Plot, and Output tabs

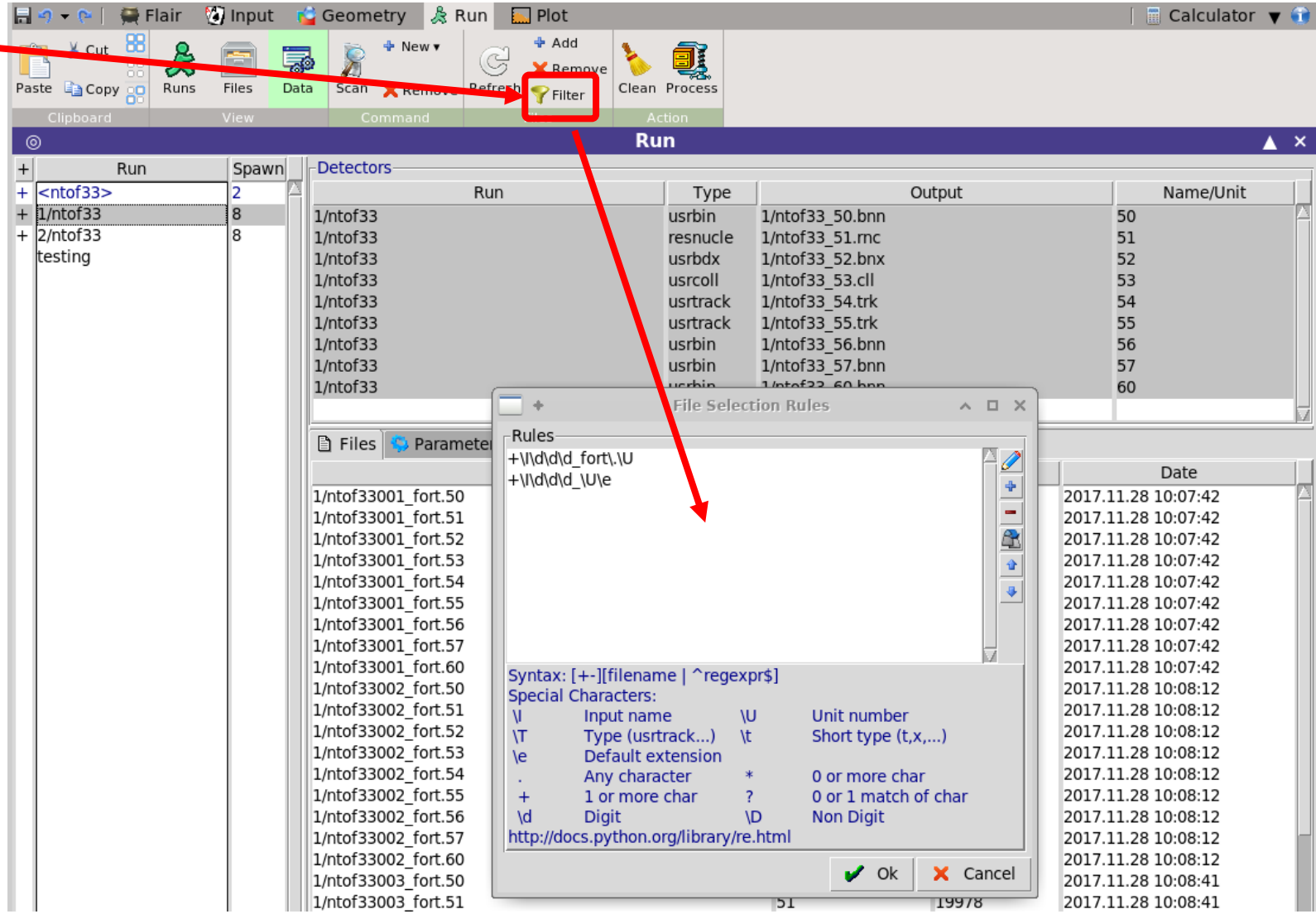
Customization

- Select job queue
  - default: =nohup
  - local: attached to flair
  - nohup: detached from flair
  - tsp: task spooler (simple single-machine task scheduler)
- Create loop of runs over **`#define`** variable
- Direct access to configuration e.g. change spawn name



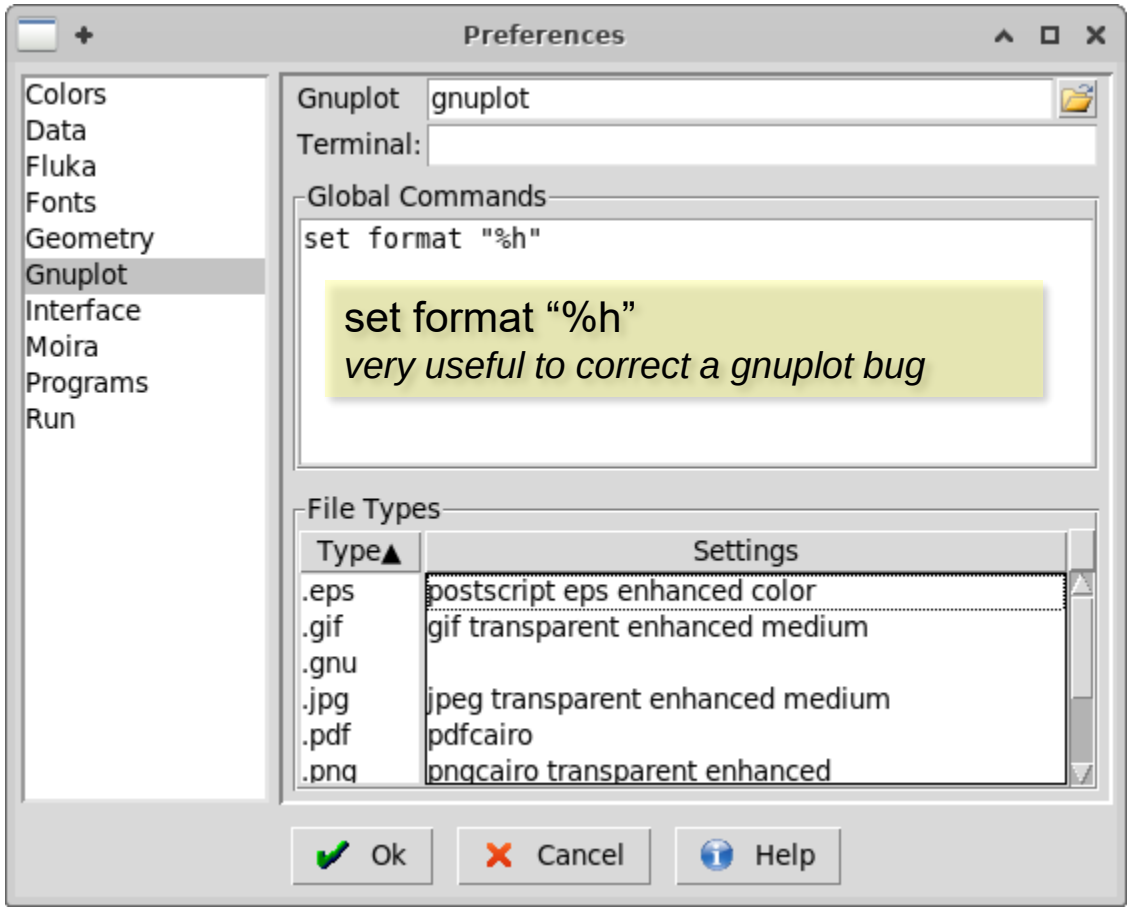
Customization

- Possible to customize filtering rules
- Possible to create processing commands in flair.ini (examples inside)



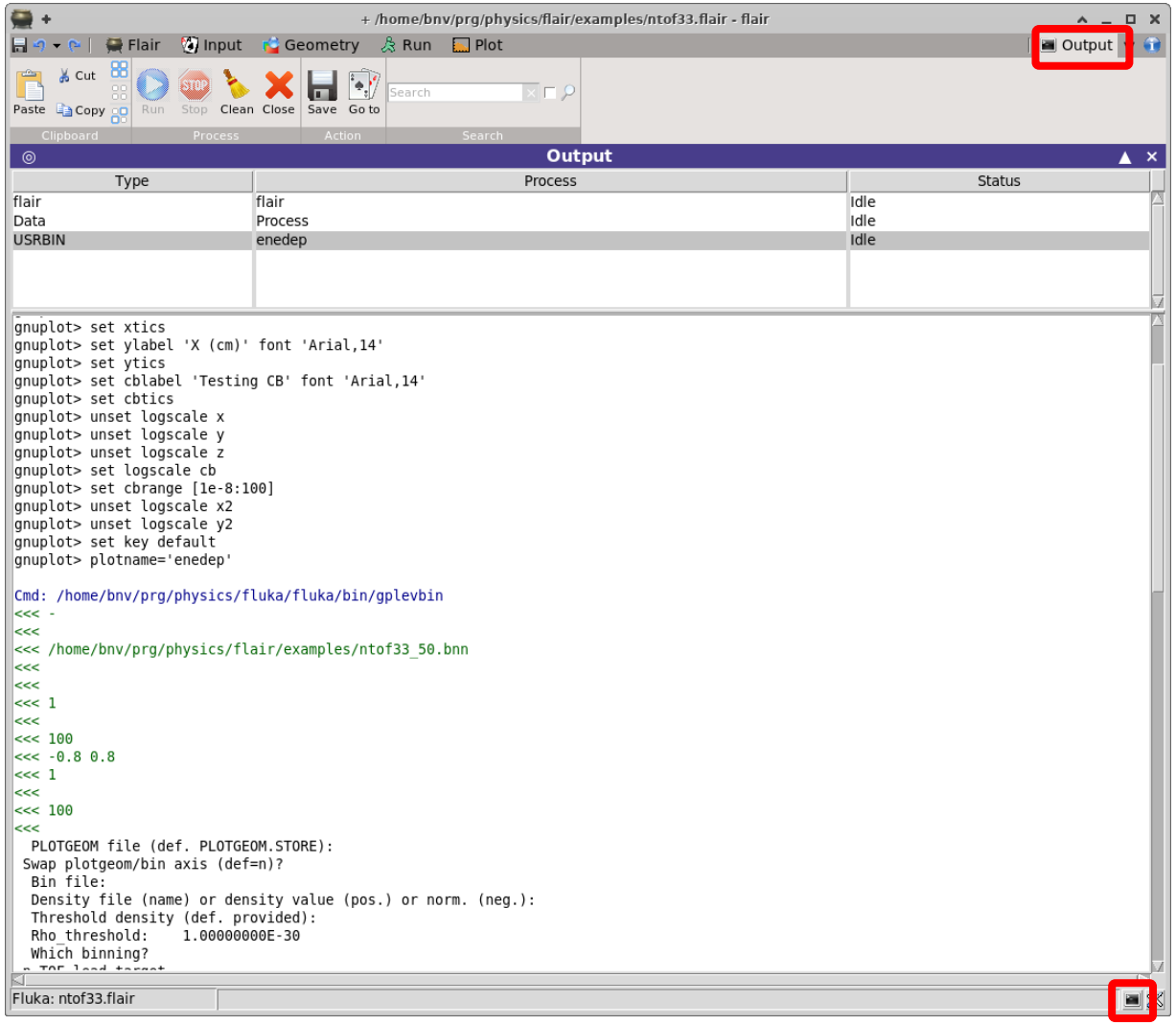
Plotting configuration

- Configure → Plot
  - Specify gnuplot path
  - Select favorite gnuplot terminal
  - Set gnuplot global commands
    (similar to ${HOME}/.gnuplot)
  - Customize export file type



Output tab

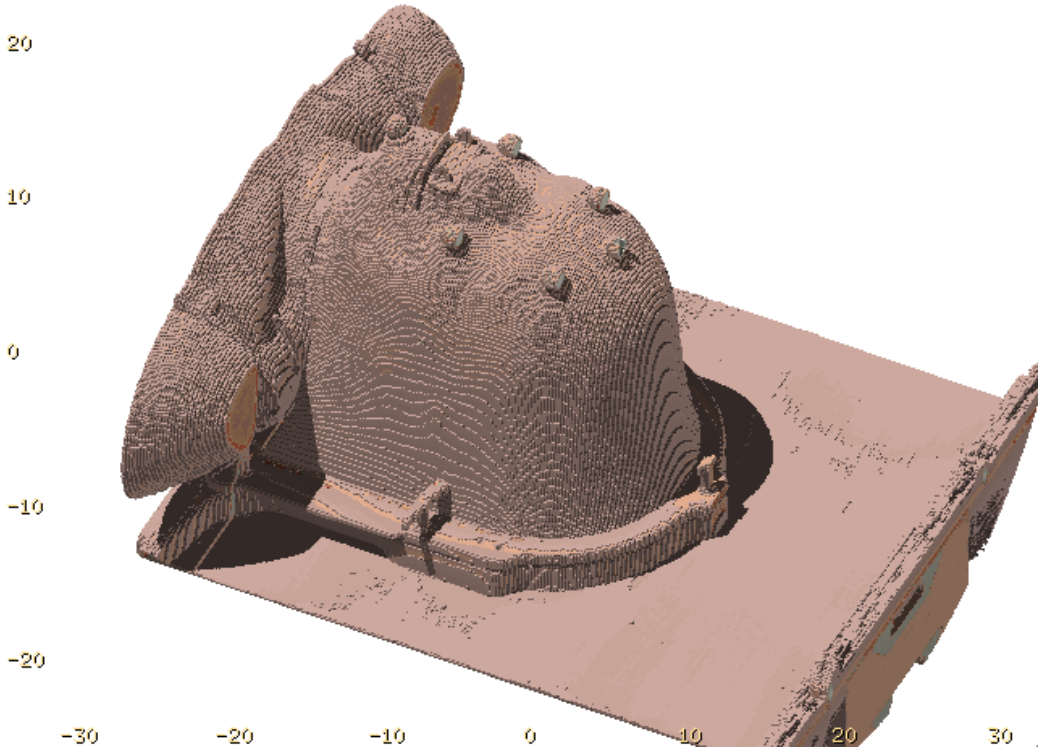
- Source of useful information
  - Displays every command executed
  - Classified by processes starting the command
- Various extra commands:
  - Clean display
  - Save output as text
  - Search string
- ERROS & WARNINGS
  - Clickable
  - Point to faulty cards

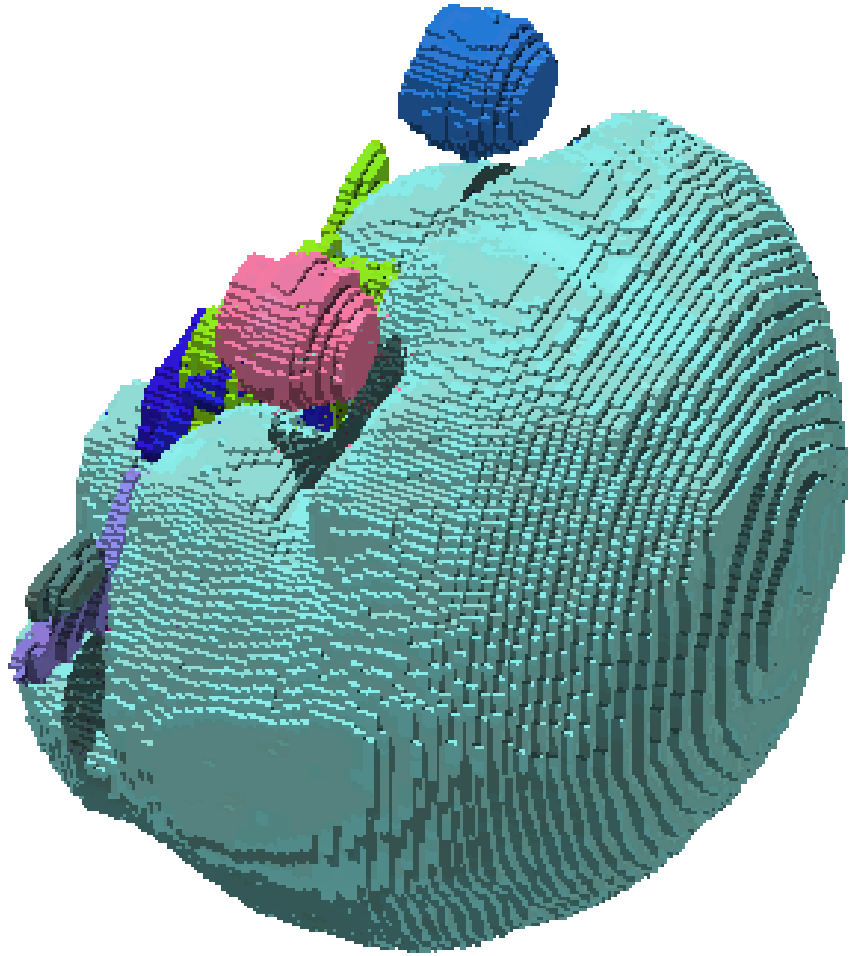


More advanced topics

Voxel

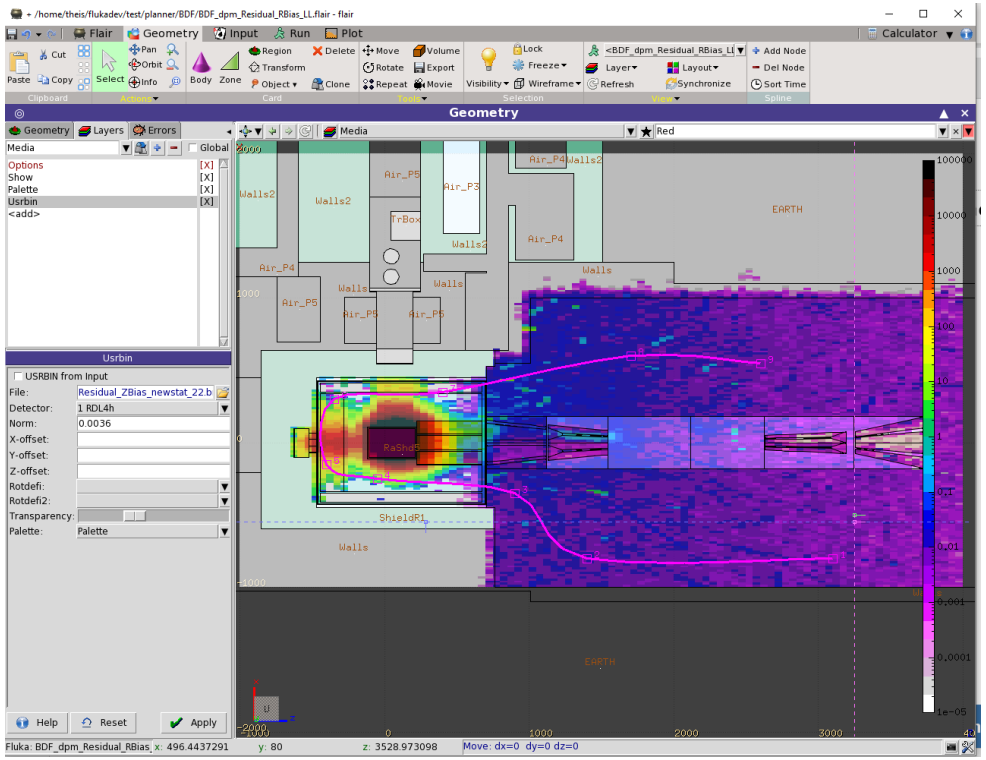
- Advanced topic
- It is possible to visualize voxel
- Dedicated layer

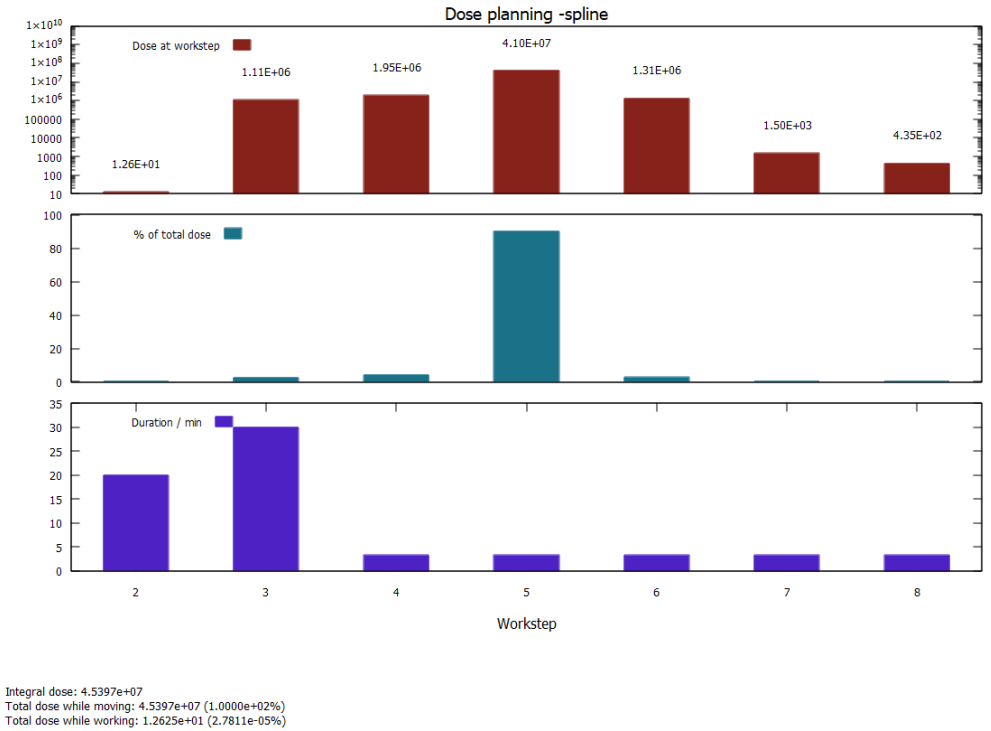




Planner

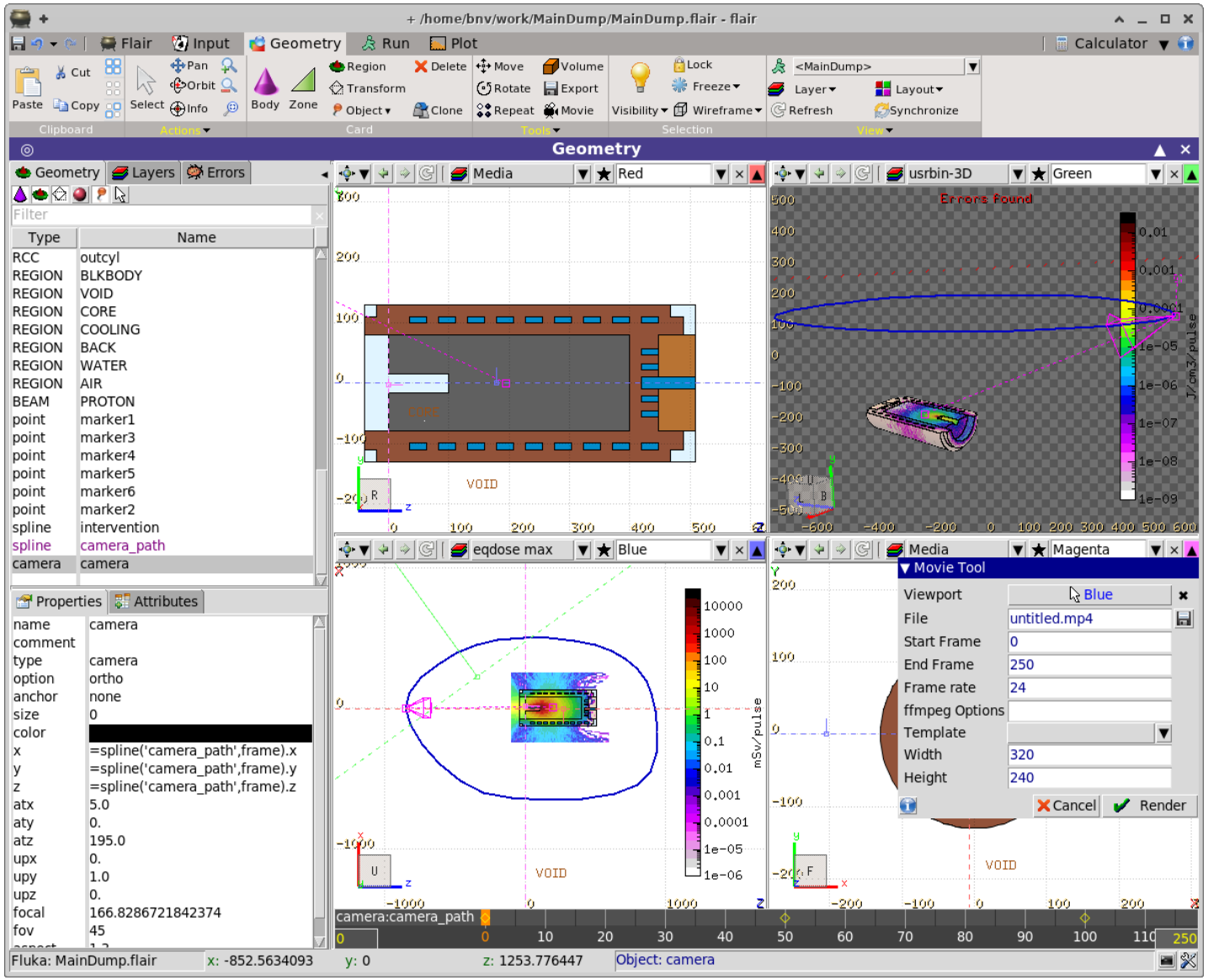
- Advanced topic
- Allows to calculate the dose incurred due to interventions in activated areas
- Takes into account time to perform work and time for movement





Movies

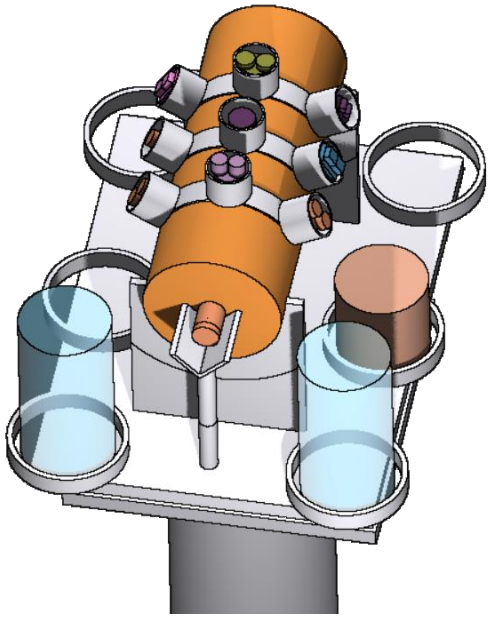
- Advanced topic
- Flair can generate movies
  - Need to define a spline
  - Need to define a camera

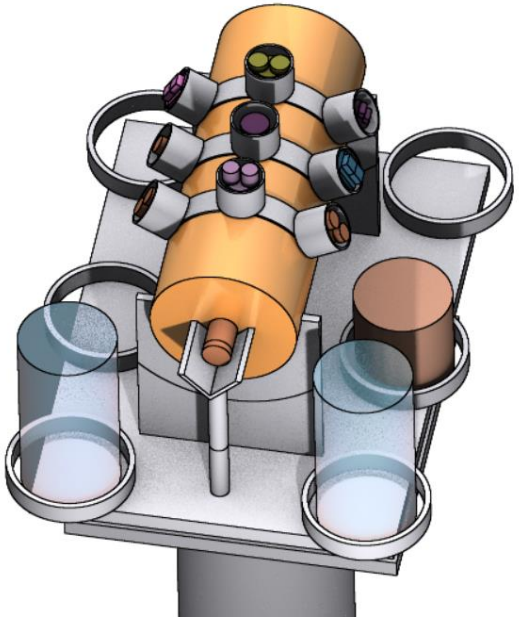


FARM – Flair Advanced Render Module

- Advanced topic
- Developed by C. Theis and V. Vlachoudis
- Two render mode available:
  - Speed: optimized for 3D previews during geometry construction
  - Quality: optimized for high-quality photorealistic renderings for presentations & publications

Speed
interactive rendering

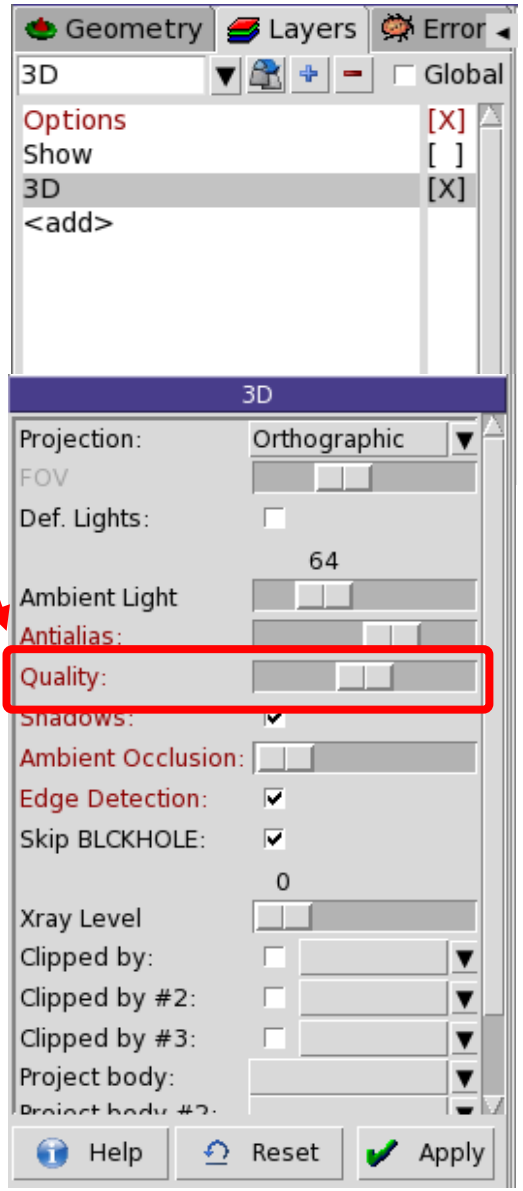




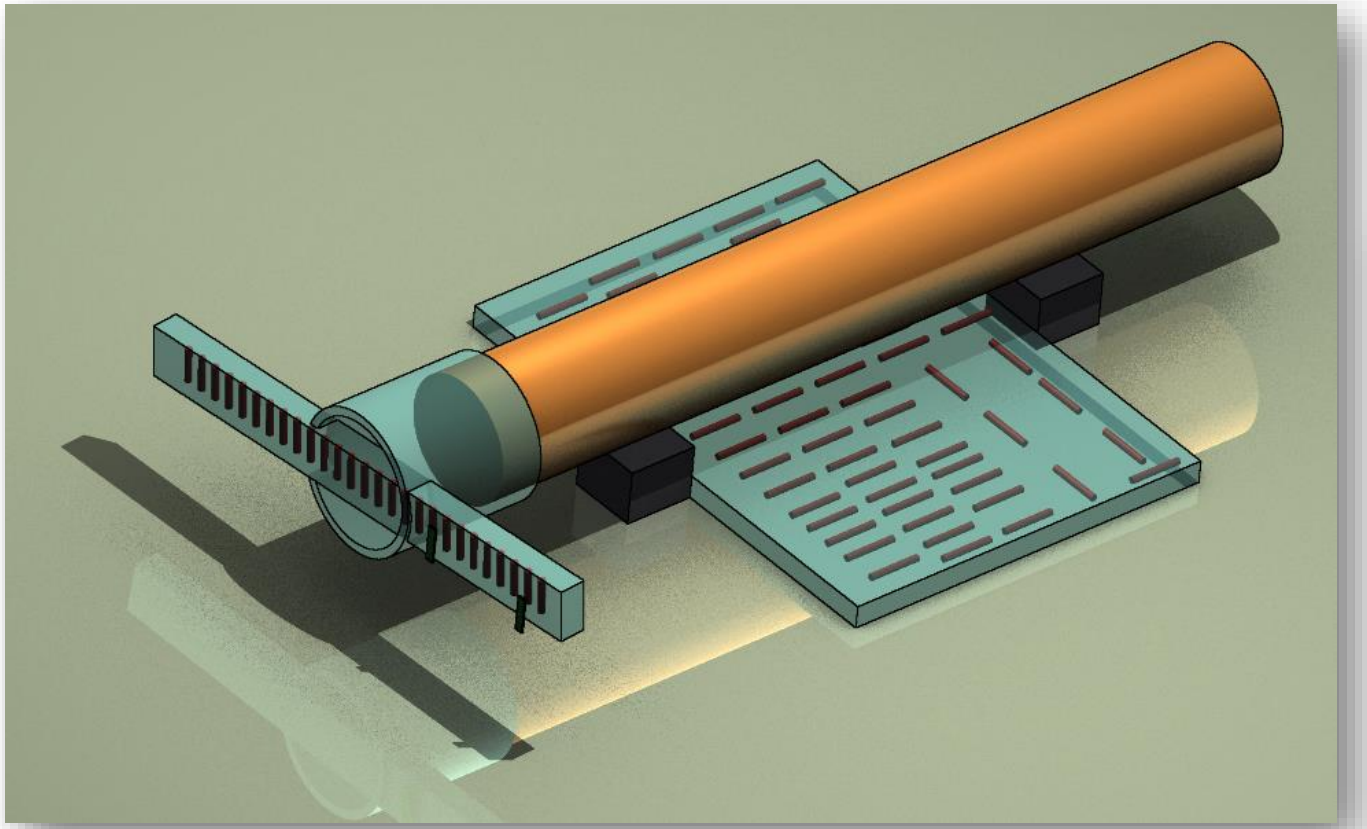
Quality
15s to render

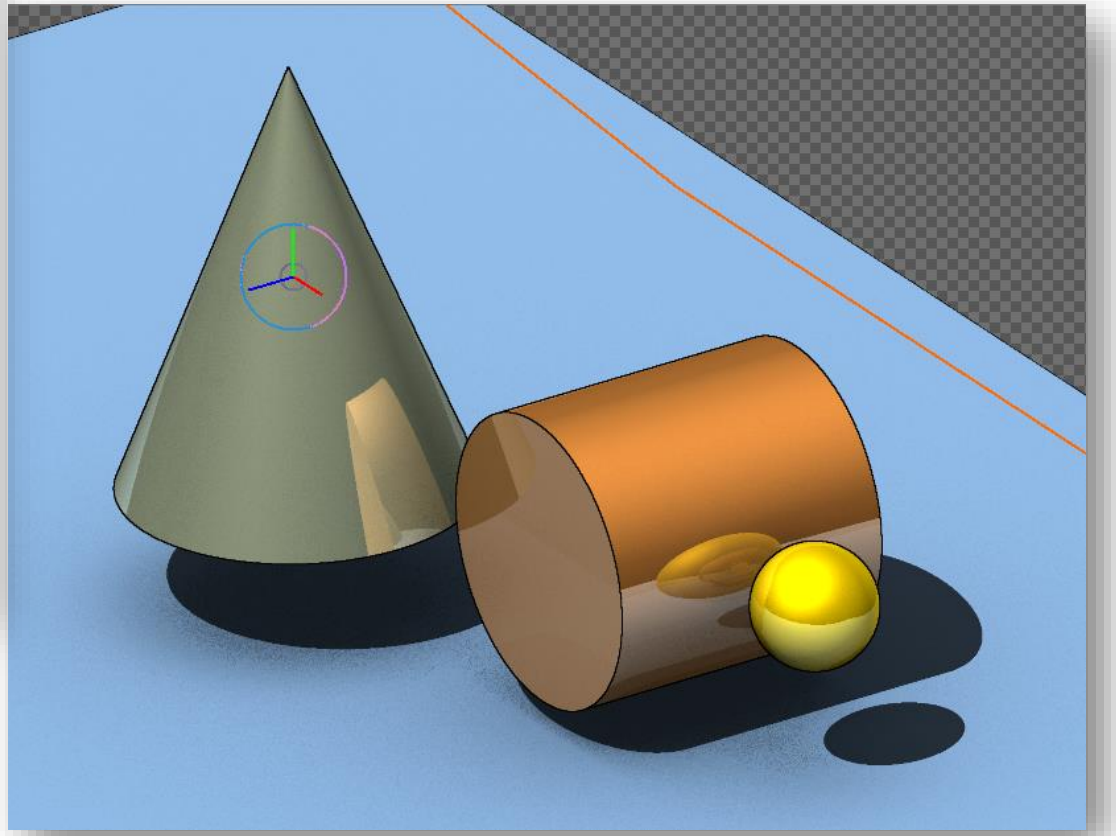
FARM – Flair Advanced Render Module

- “Quality” mode activated by the slider in the 3D layer options
- “Quality” mode takes into account:
  - Material dependent scattering & reflection models
  - Light reflection & refraction
  - Fresnel reflection for conductors & di-electrics
  - Disperision
  - Beer’s absorption
  - Microfacet distribution for polished metals
- Important: it is much slower as it invokes a physically based renderer requiring more resources to correctly account for all the effects

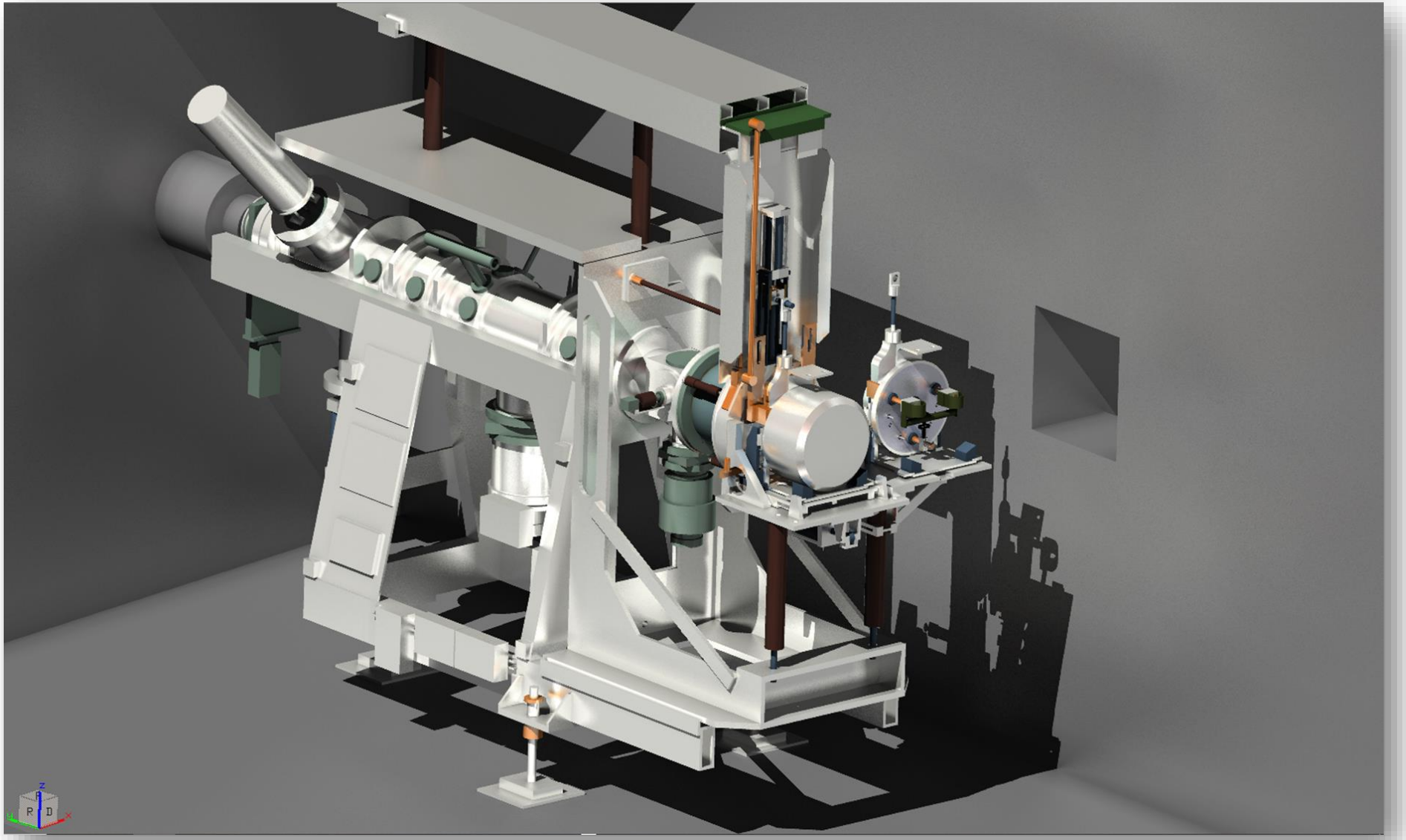


FARM – Flair Advanced Render Module

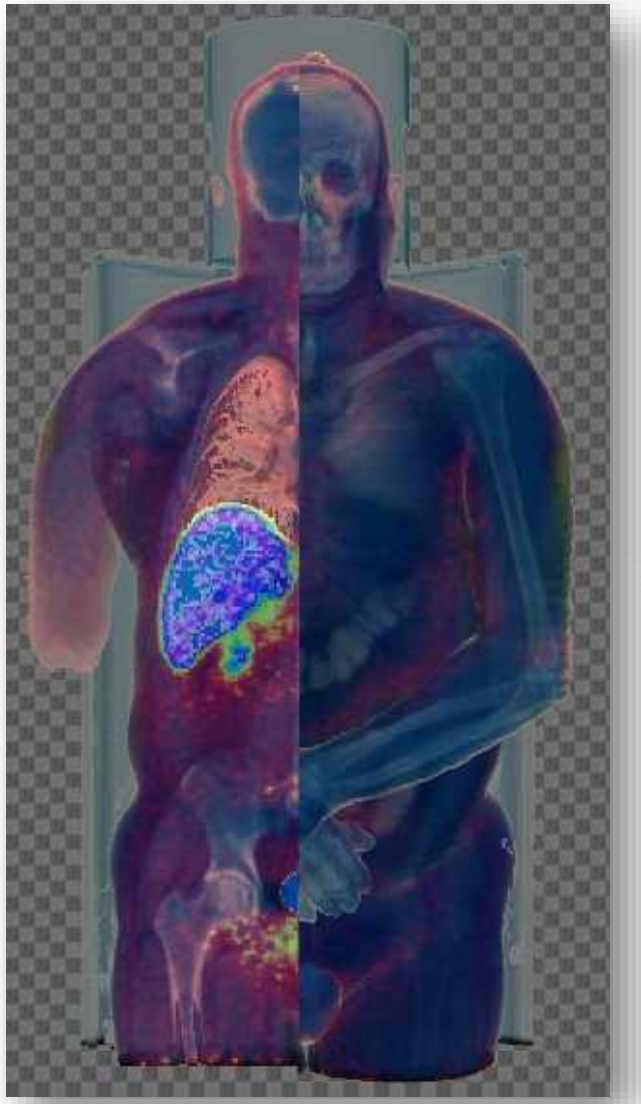


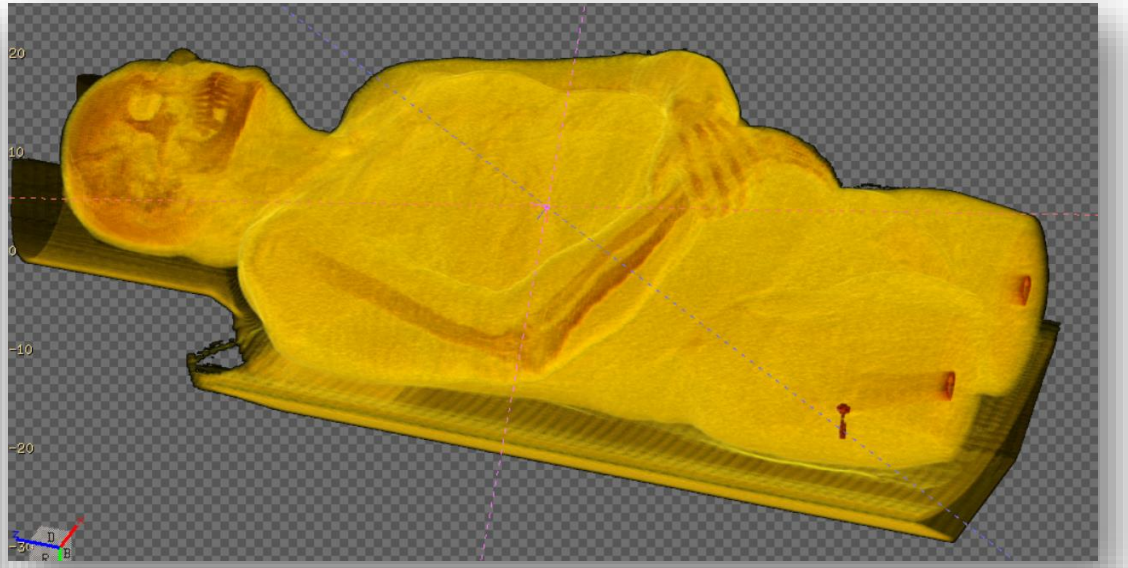


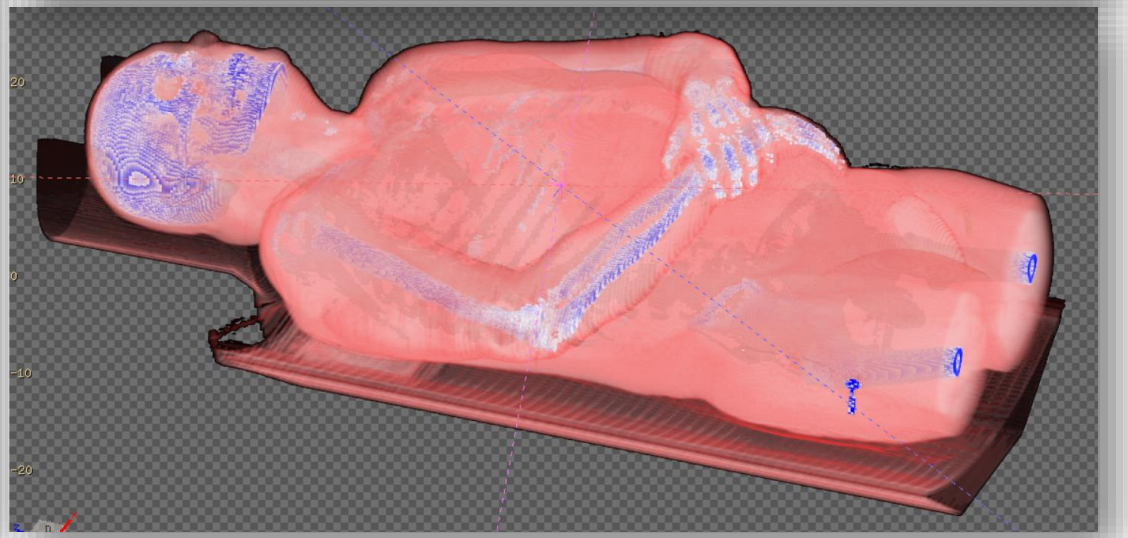
FARM – Flair Advanced Render Module

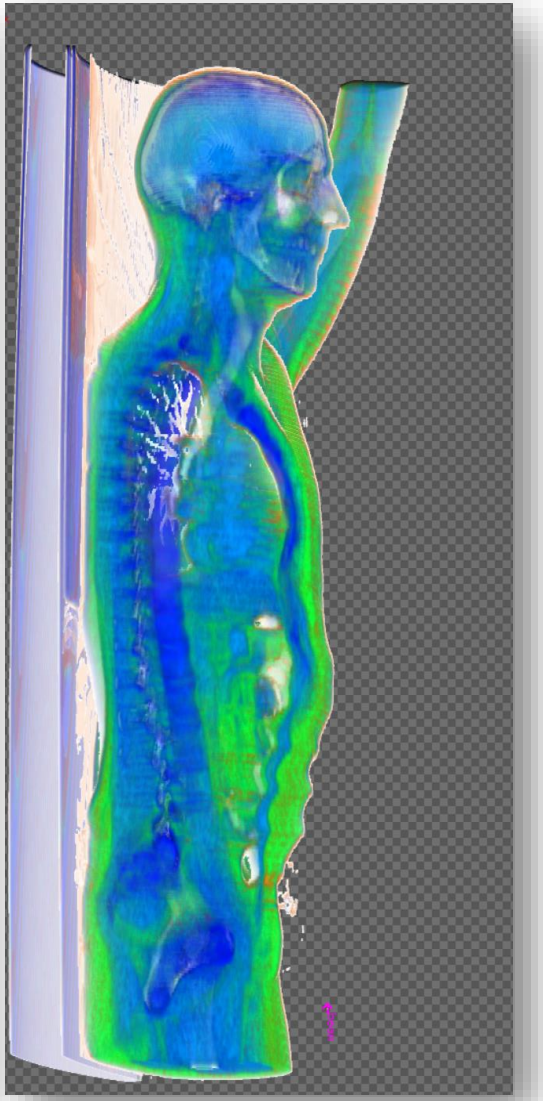


FARM – Flair Advanced Render Module





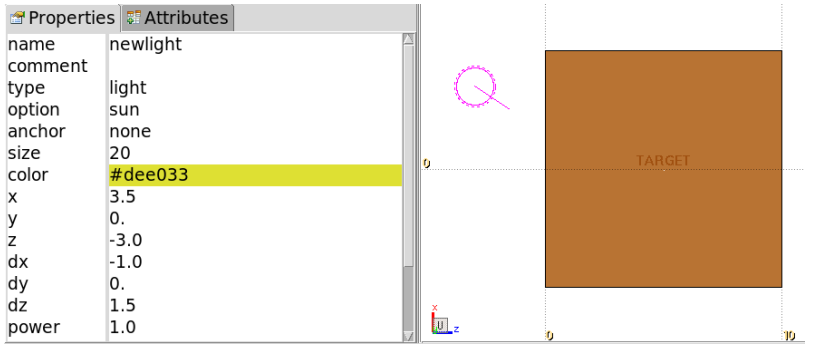


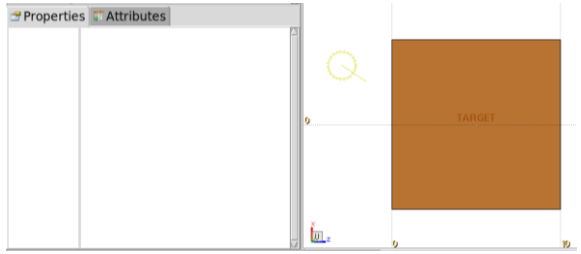


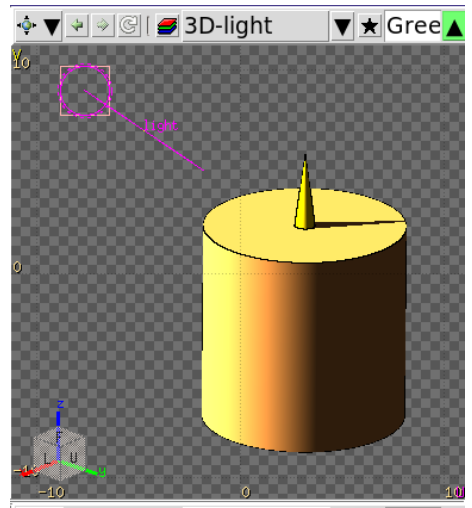
Spare slides

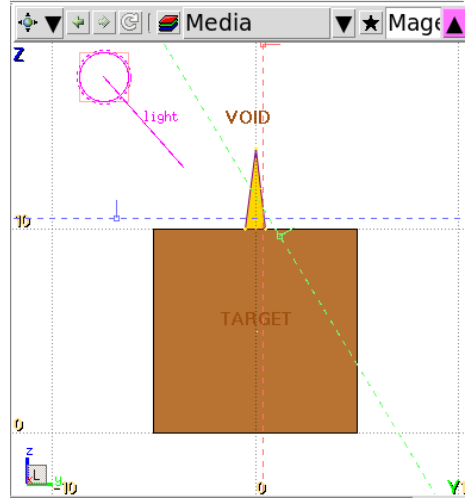
Objects

- Light
  - Light source to illuminate 3D rendering
  - More lights can be defined

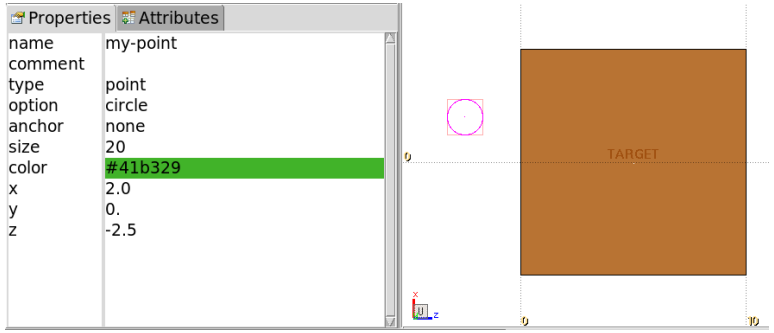


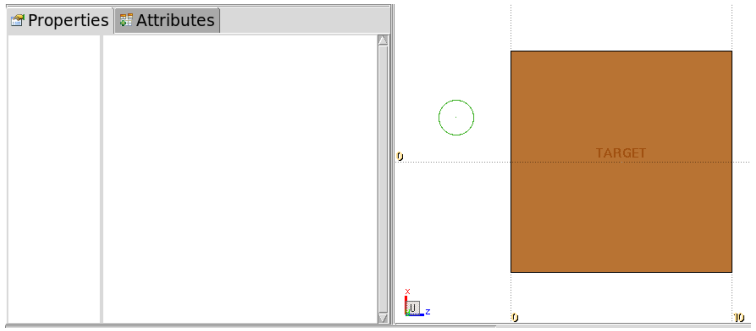






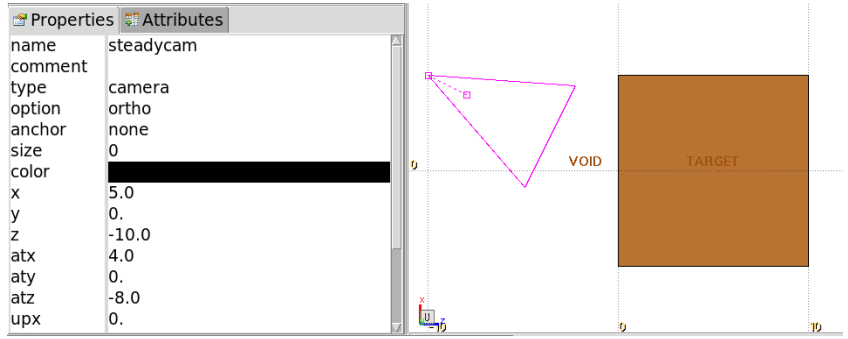
- Point
  - Can be used as snapping point
  - Automatically generated after image calibration
  - Allows to add text on the viewport

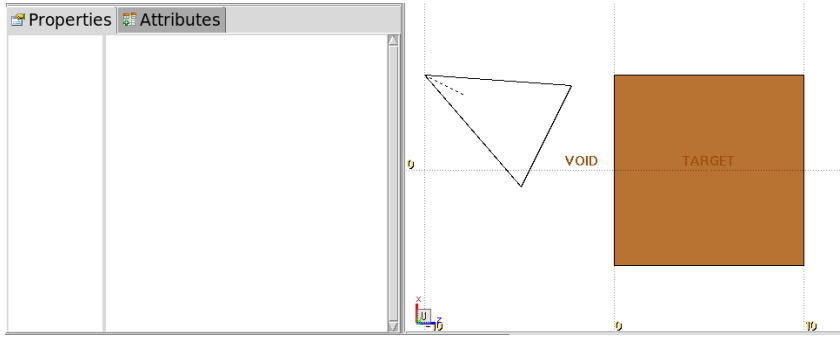




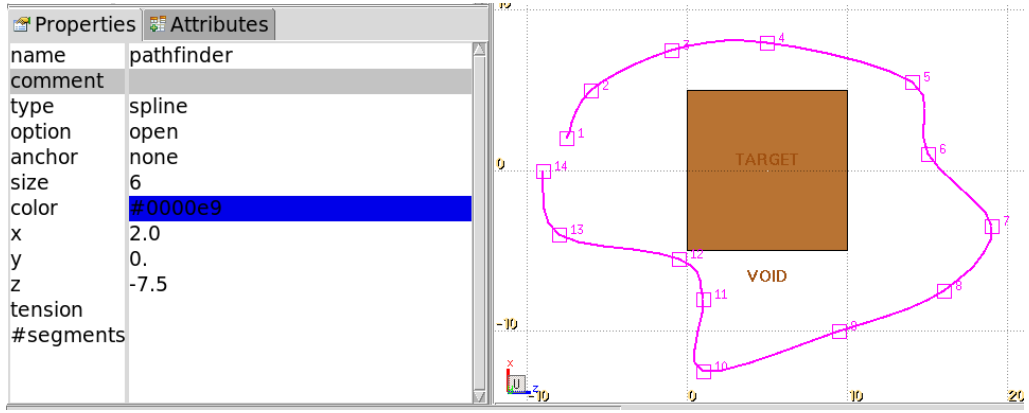
Objects

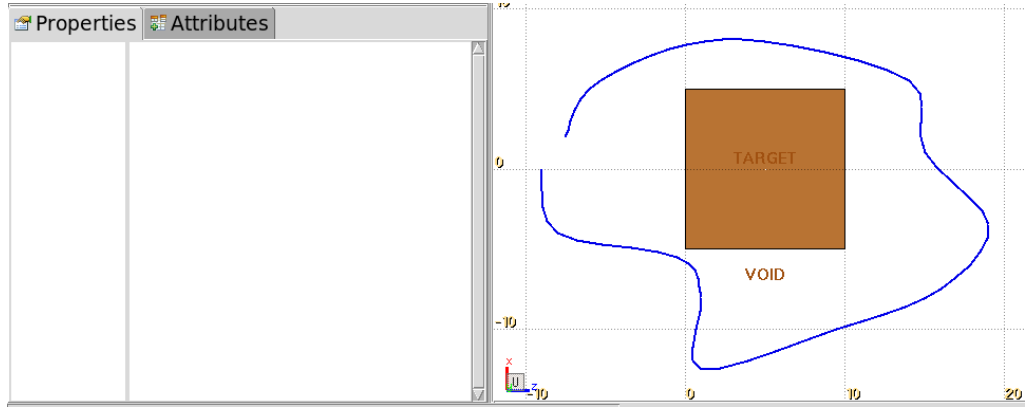
- Camera
  - Used to create movies
  - Can move along a spline





- Spline
  - Creates a cardinal cubic time-spline path for a set of nodes (closed or open)
  - Used to define the path of moving objects (e.g. camera)

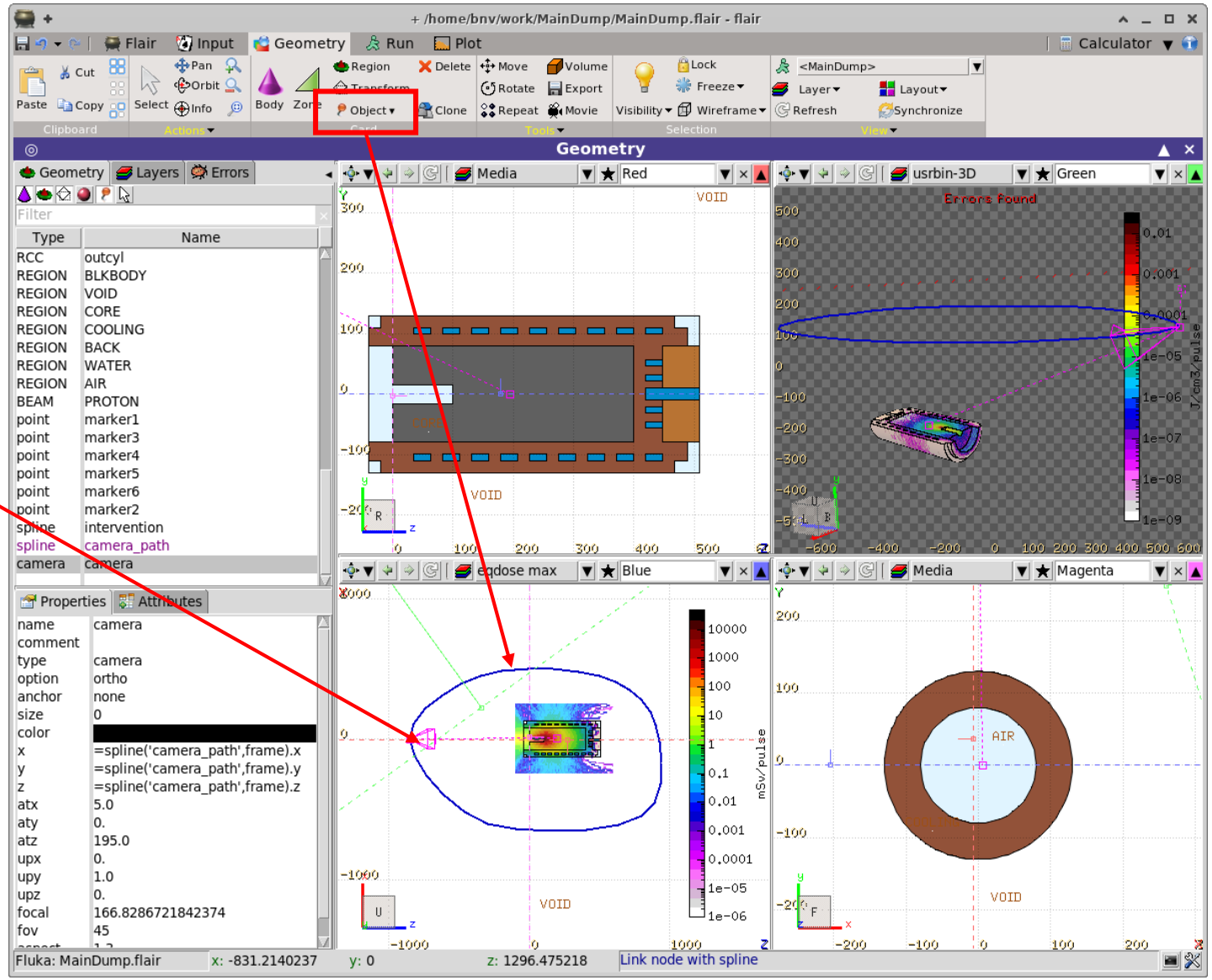




Geometry Editor: movie creation

Steps

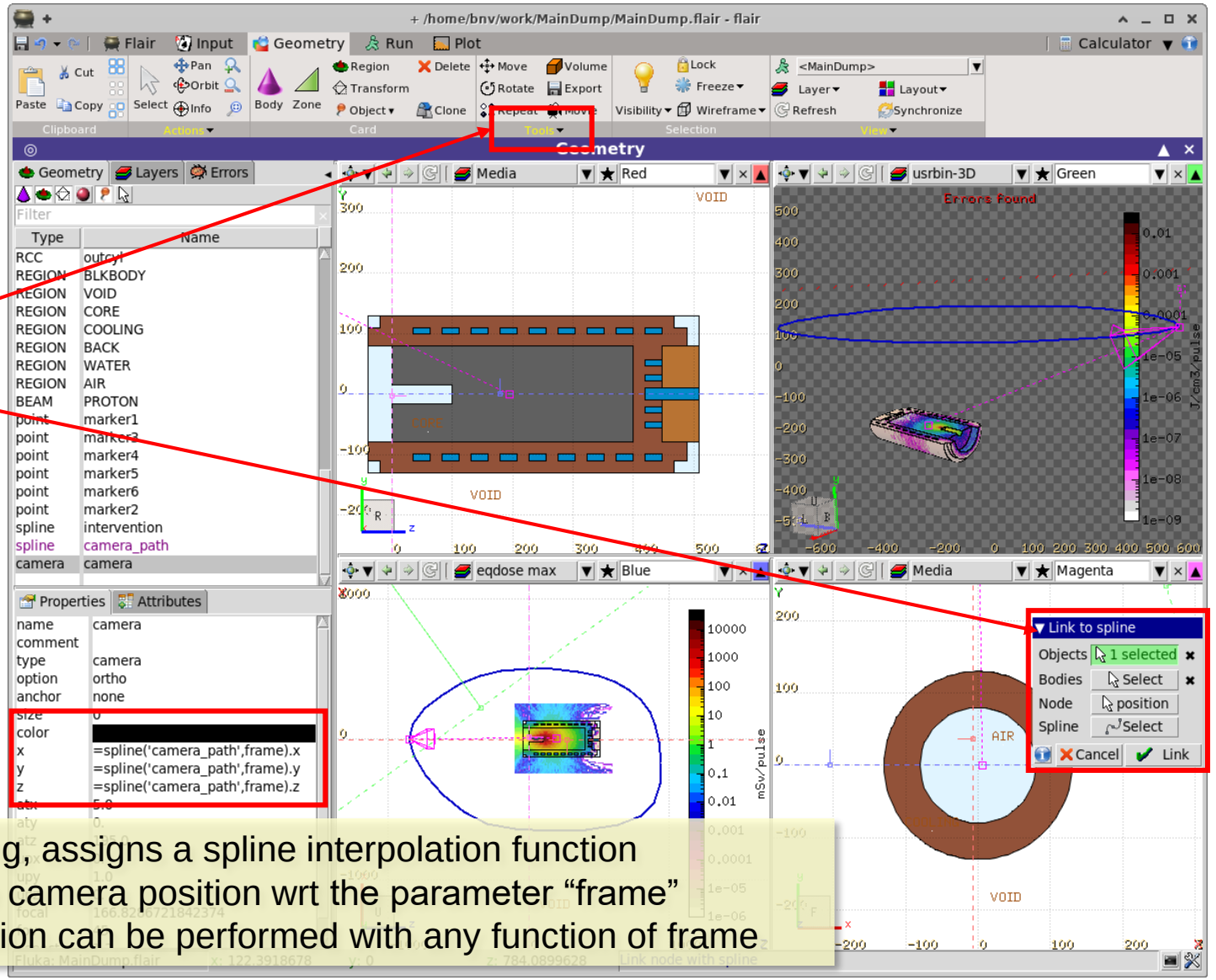
1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



Geometry Editor: movie creation

Steps

1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time
   (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool

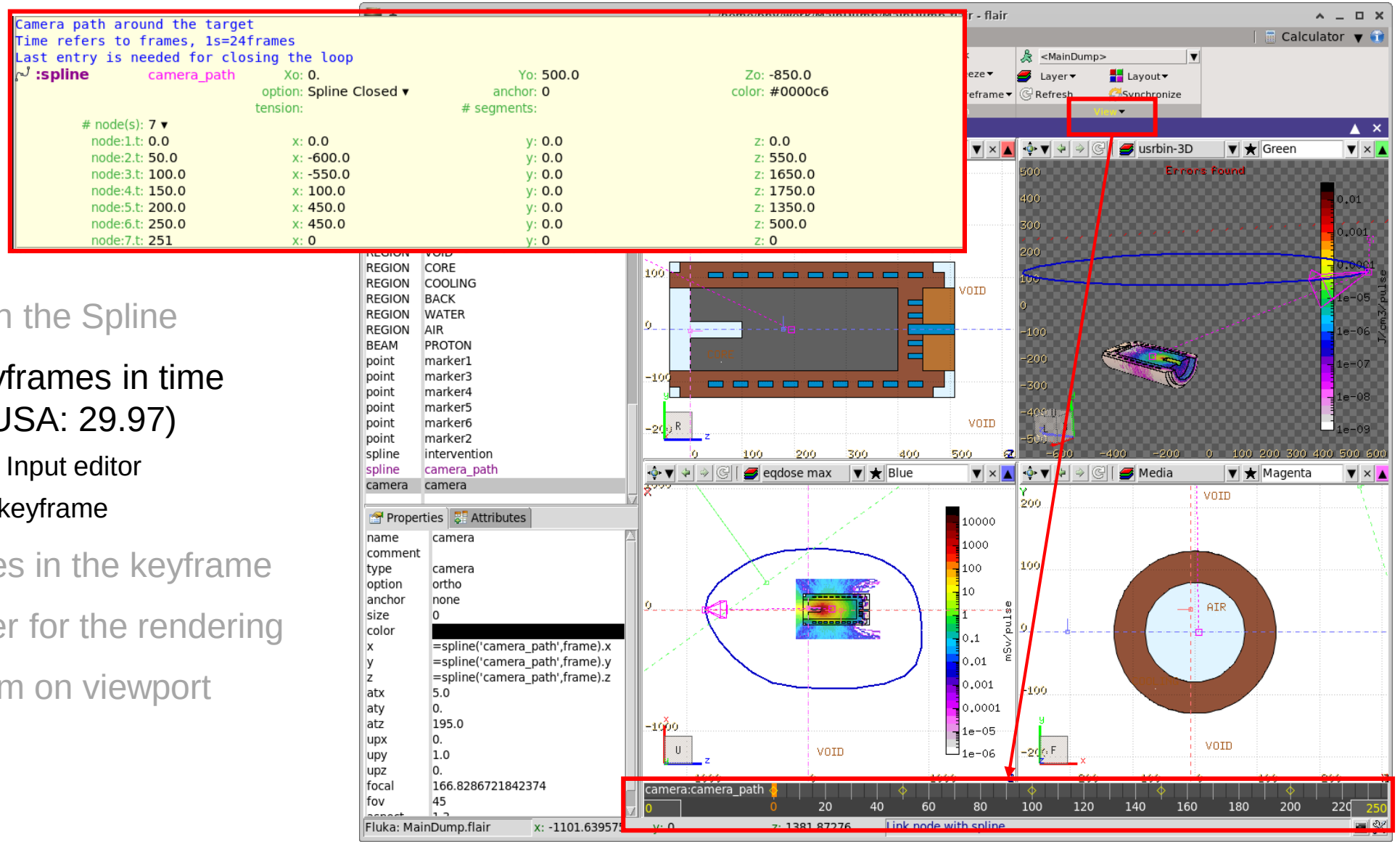


Linking, assigns a spline interpolation function to the camera position wrt the parameter “frame”
A motion can be performed with any function of frame

Geometry Editor: movie creation

Steps

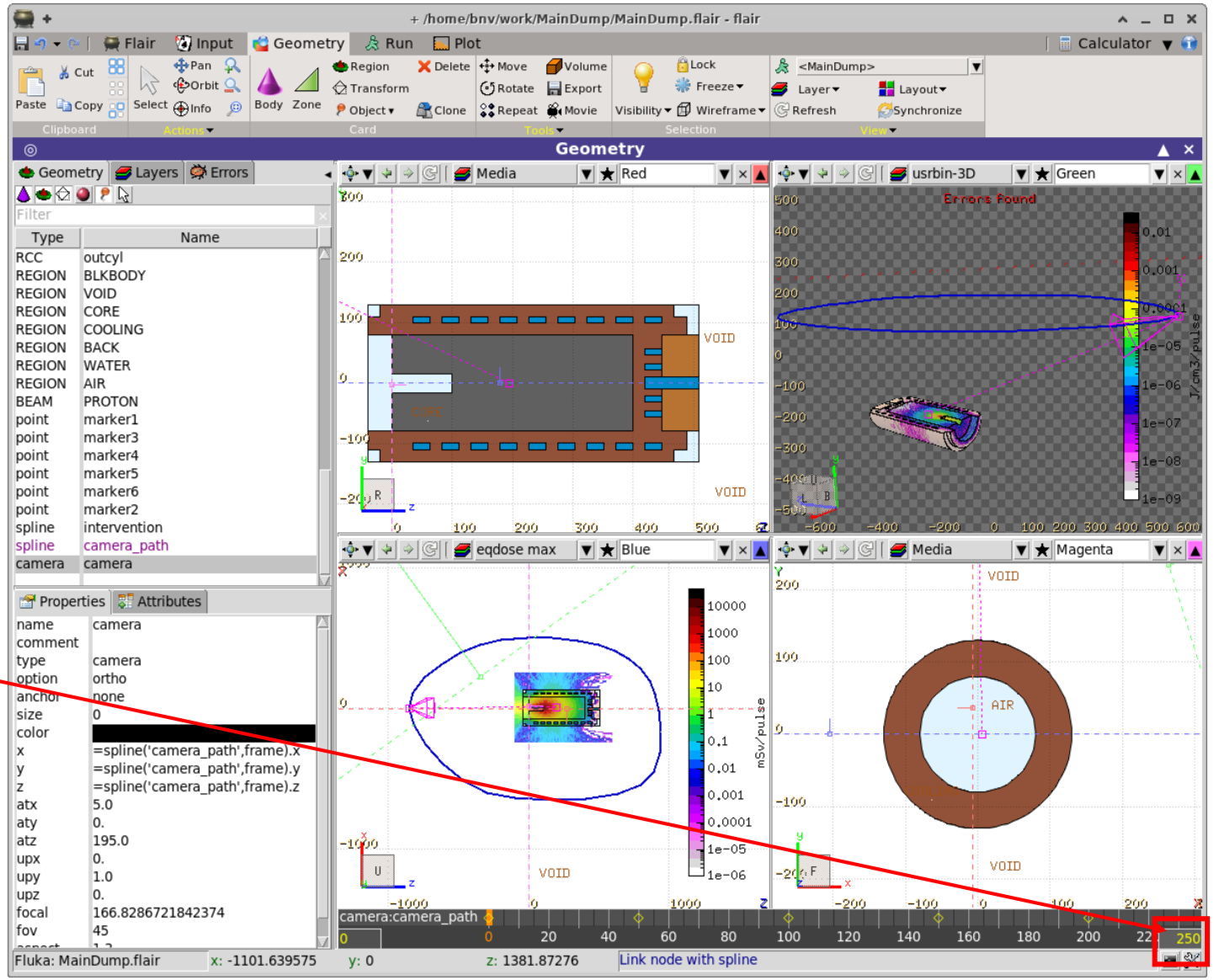
1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



Geometry Editor: movie creation

Steps

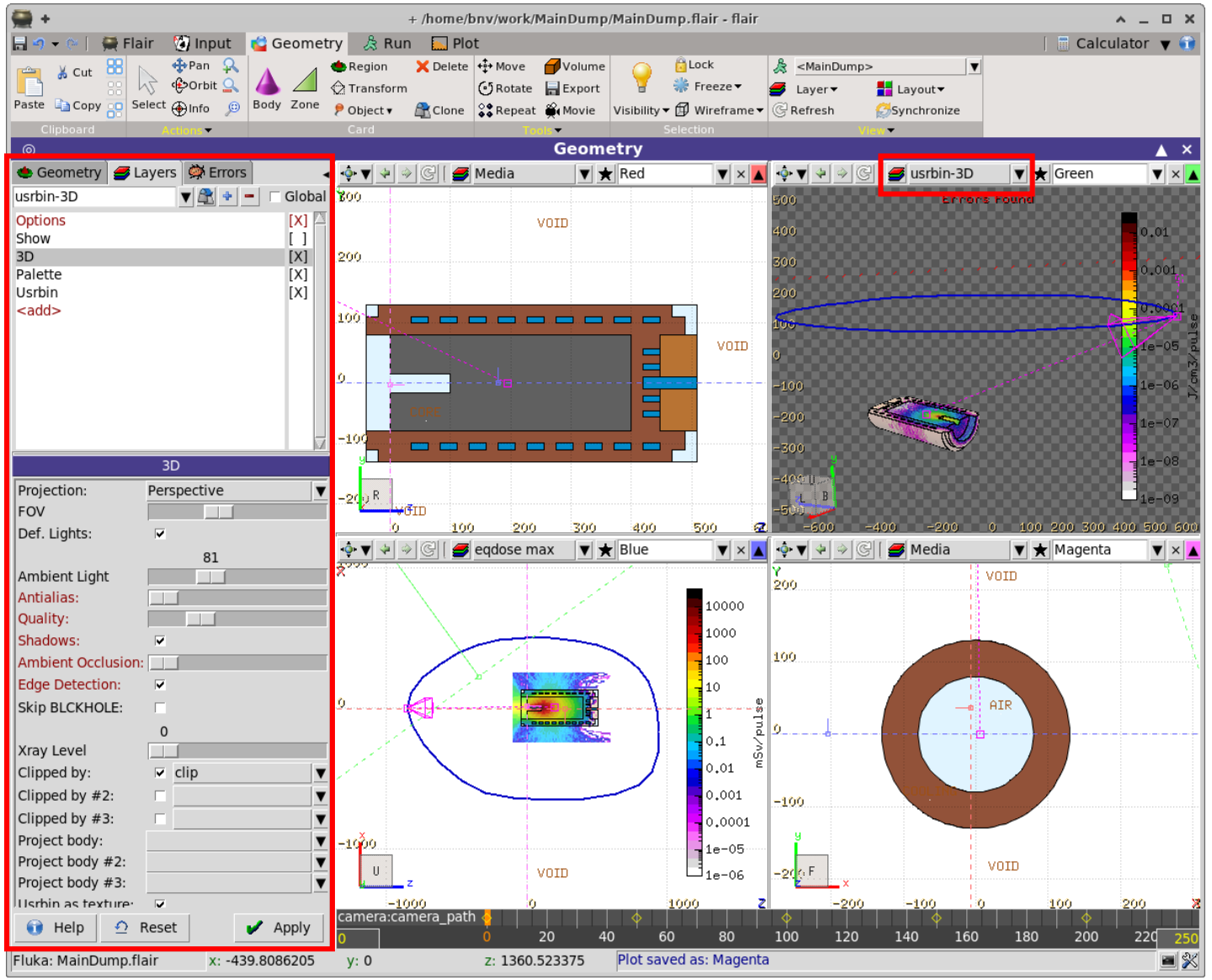
1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time
   (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



Geometry Editor: movie creation

Steps

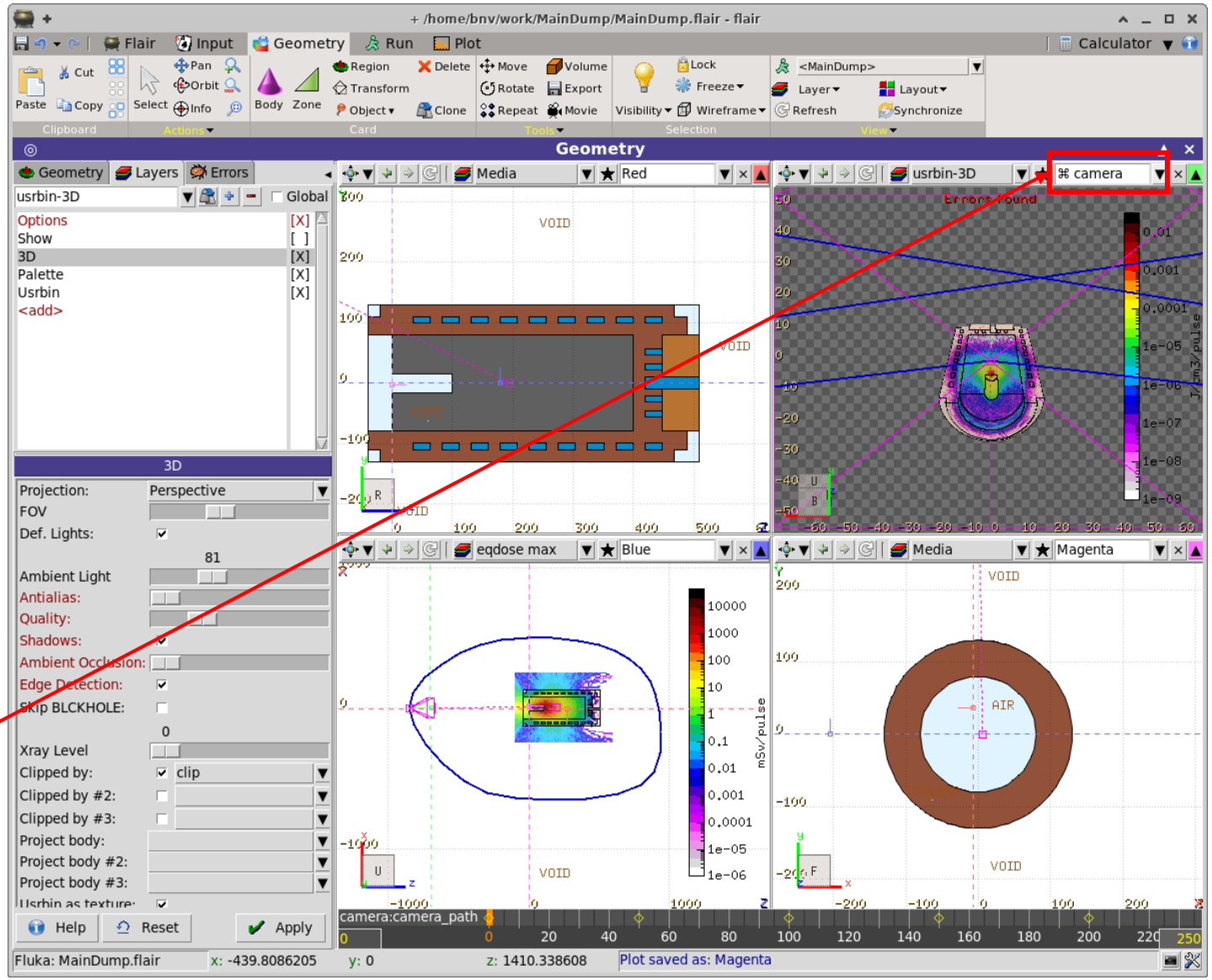
1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time
   (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



Geometry Editor: movie creation

Steps

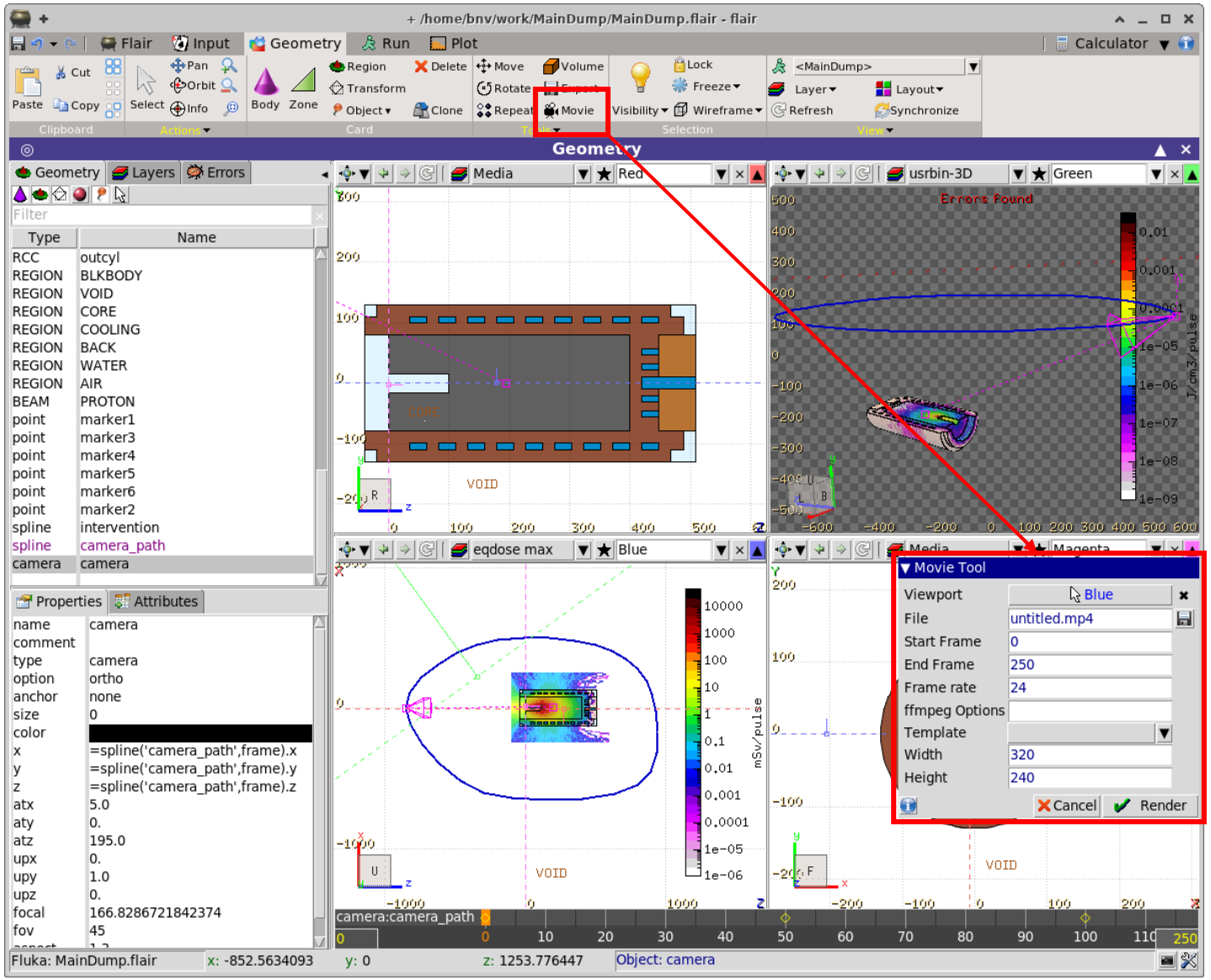
1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time
   (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



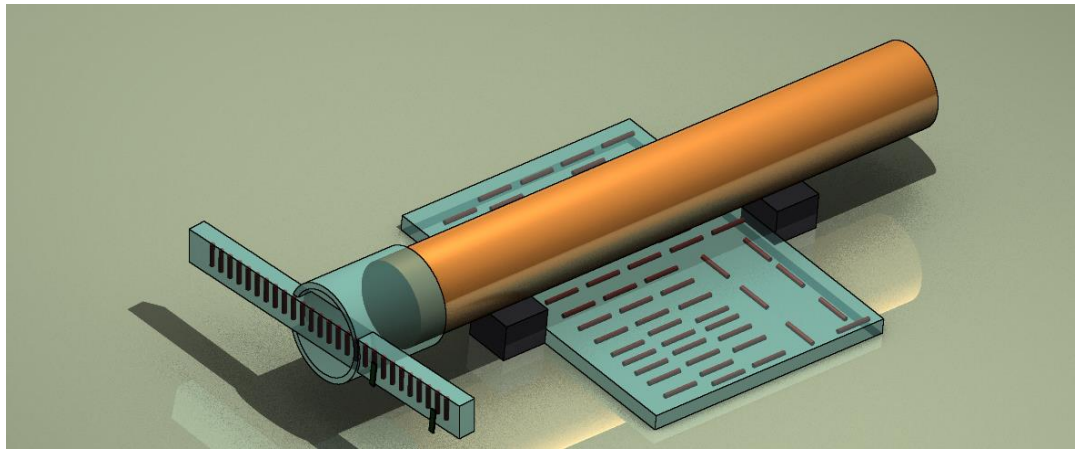
Geometry Editor: movie creation

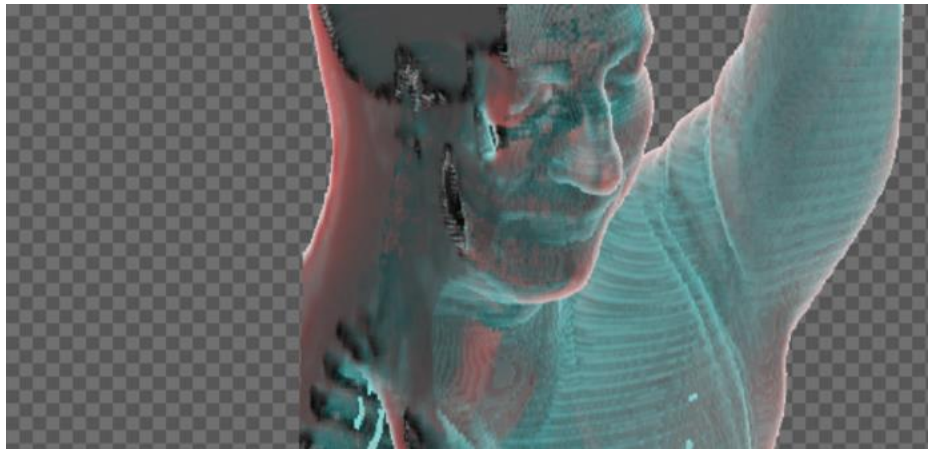
Steps

1. Create a spline path
   Object → Spline
2. Create a camera
   Object → Camera
3. Link the Camera with the Spline
4. Adjust the spline keyframes in time
   (EU: 1s=24 frames USA: 29.97)
   - Either manually in the Input editor
   - or with the use of the keyframe
5. Adjust the # of frames in the keyframe
6. Customize a 3D layer for the rendering
7. View the camera from on viewport
8. Open the Movie tool



More advanced topics: FARM





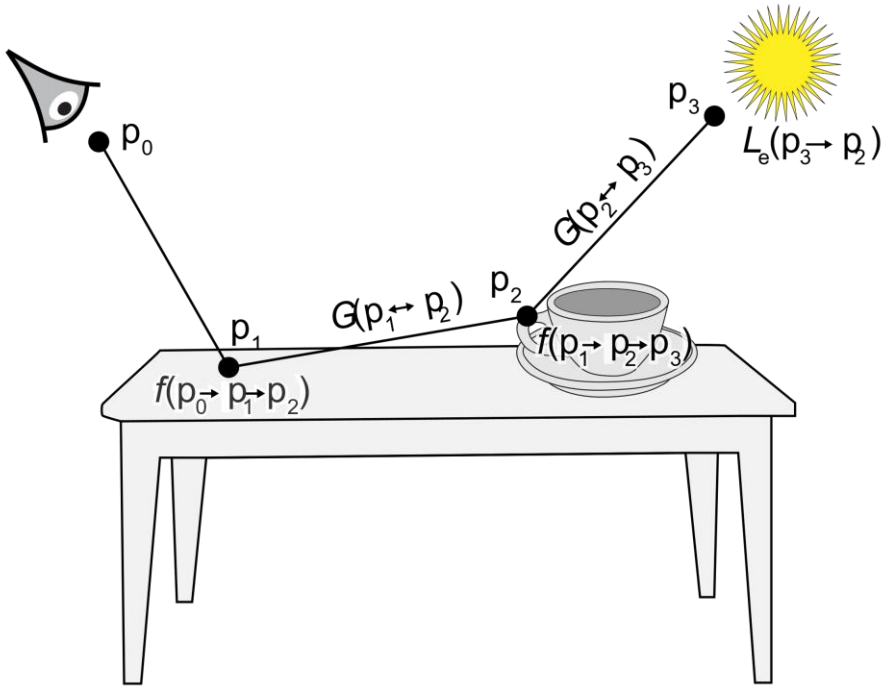
FARM – Flair Advanced Render Module

- Physics-based ray-tracer which solves light transport equations

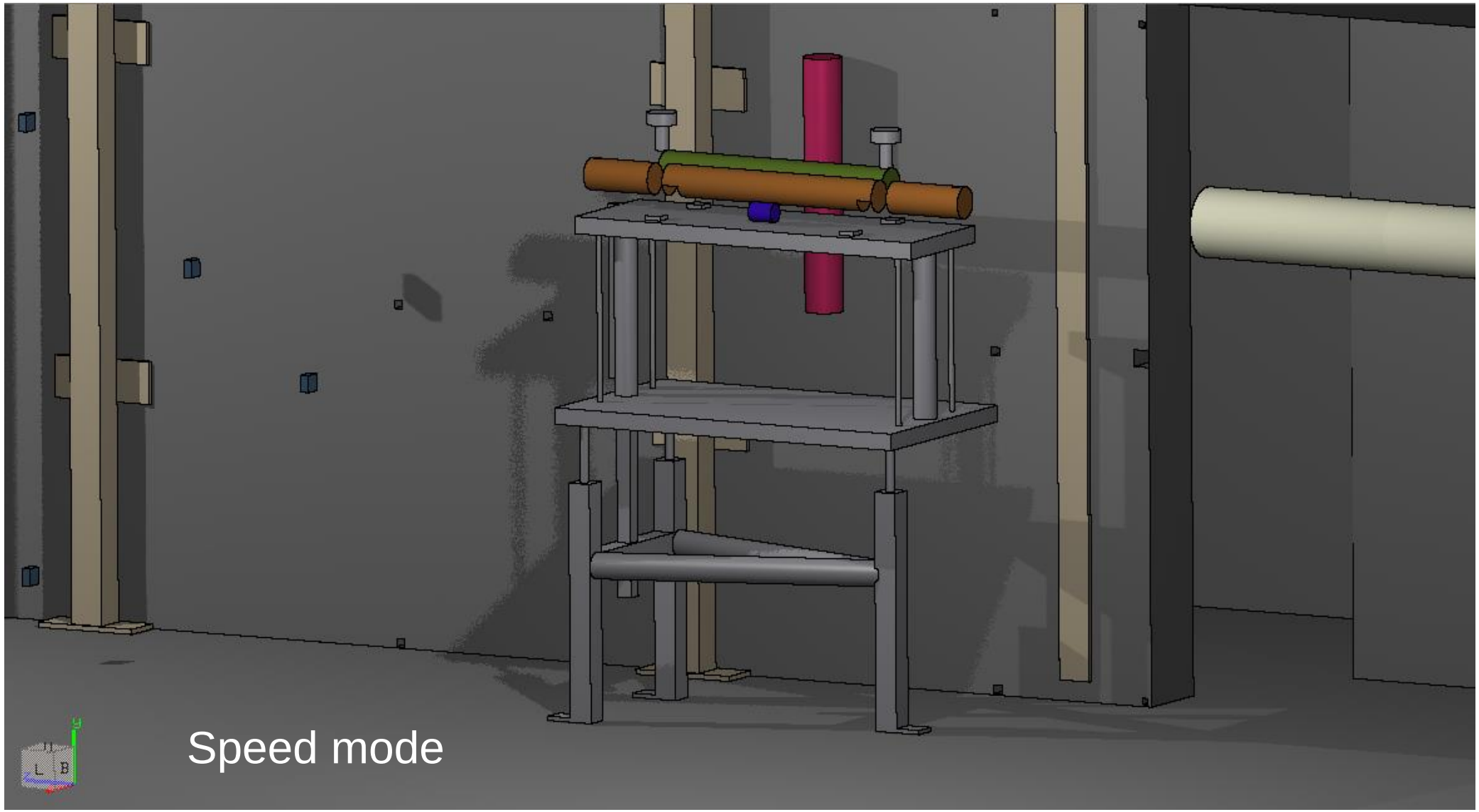
- Integrates so-called “bi-direction scattering function”, which model the interaction of photons with materials using Monte Carlo or analytical methods

- In “Speed” mode (default), it falls back to a relatively simplified unphysical lighting model

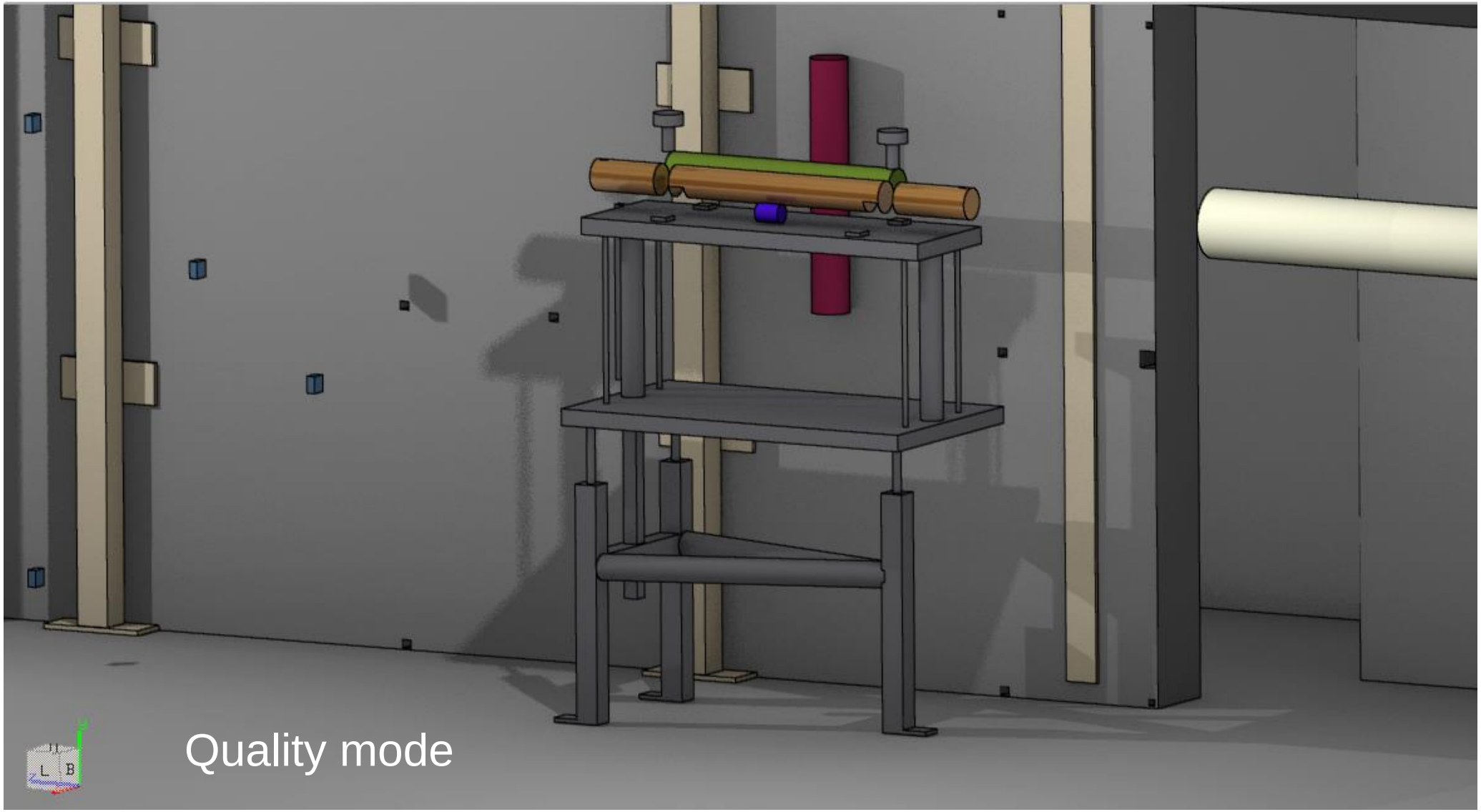
- In “Quality mode, numerous different light interaction models (partially based on measured scattering data) can be selected



FARM – Flair Advanced Render Module



FARM – Flair Advanced Render Module




FLUKA