



Introduction to Flair and basic input

A very basic introduction to perform your first simulation

A very short introduction

- Fluka's story begun a long time ago (1970s)...
...no graphical interfaces, input and output via text file
- Inputfile can be very long > 50k lines
- Inputfile based on “cards”: `.inp` file
- Each card has 1 name, 6 values (called WHATs), 1 string (called SDUM)
- Two examples of cards (the actual meaning is not relevant here):



BEAMPOS	4750.5	130.0	4866.5				NEGATIVE
BEAM	-0.4	0.2	5.0	1.E-4	1.E-4		ELECTRON

↑ ↑ ↑ ↑ ↑ ↑ ↑ ↑

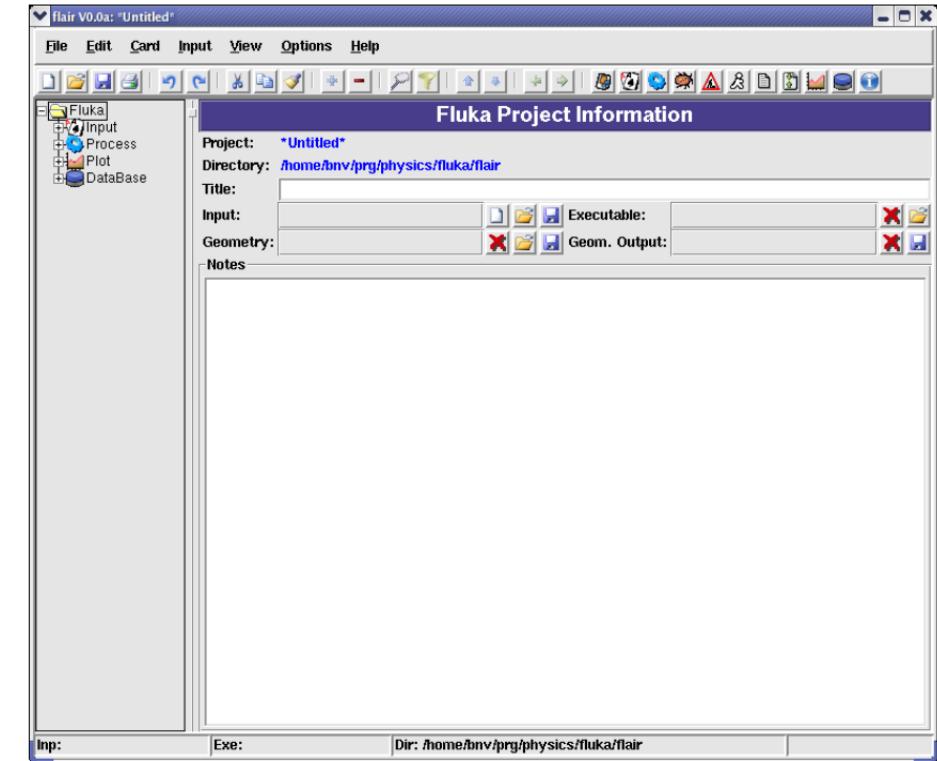
Card name WHAT(1) WHAT(2) WHAT(3) WHAT(4) WHAT(5) WHAT(6) SDUM

A very short introduction

- In 2006, Flair was born!



- Fluka advanced interface
- Inputfile creation
- Geometry visualization and construction
- Simulation execution
- Results visualization



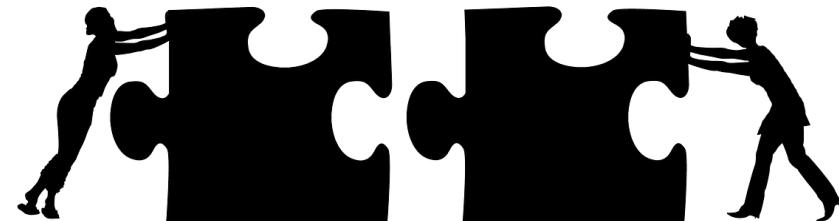
- Flair acts as an intermediate layer between the user and the inputfile
- It allows a user friendly editing of the Fluka input
- Based on a `.flair` file and generates the `.inp` file that is run by Fluka

Flair ≠ Fluka

Fluka & Flair

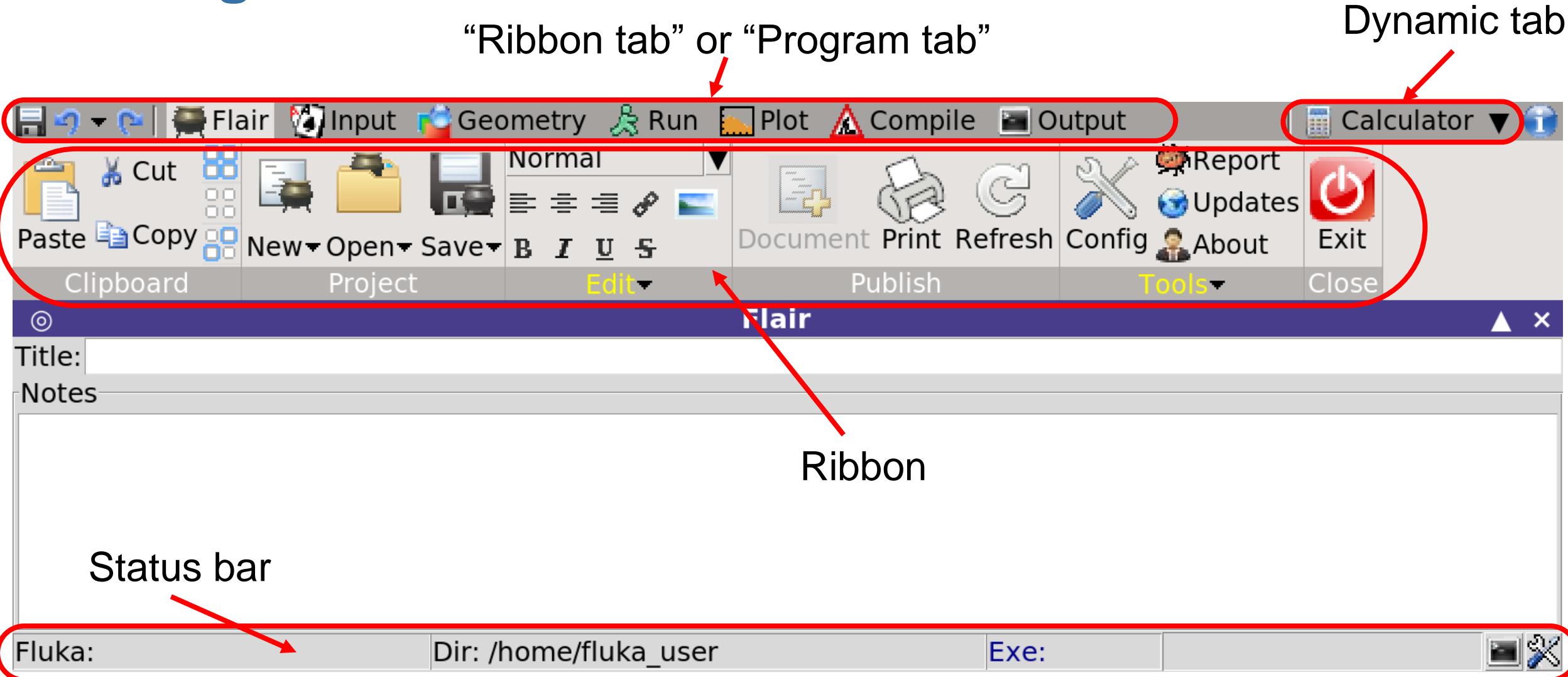


- Although strongly linked, they are two different things (`.inp ≠ .flair`)
- Fluka is a Monte Carlo transport code based on text files
- Flair is a graphical interface to Fluka
- They work together but are different

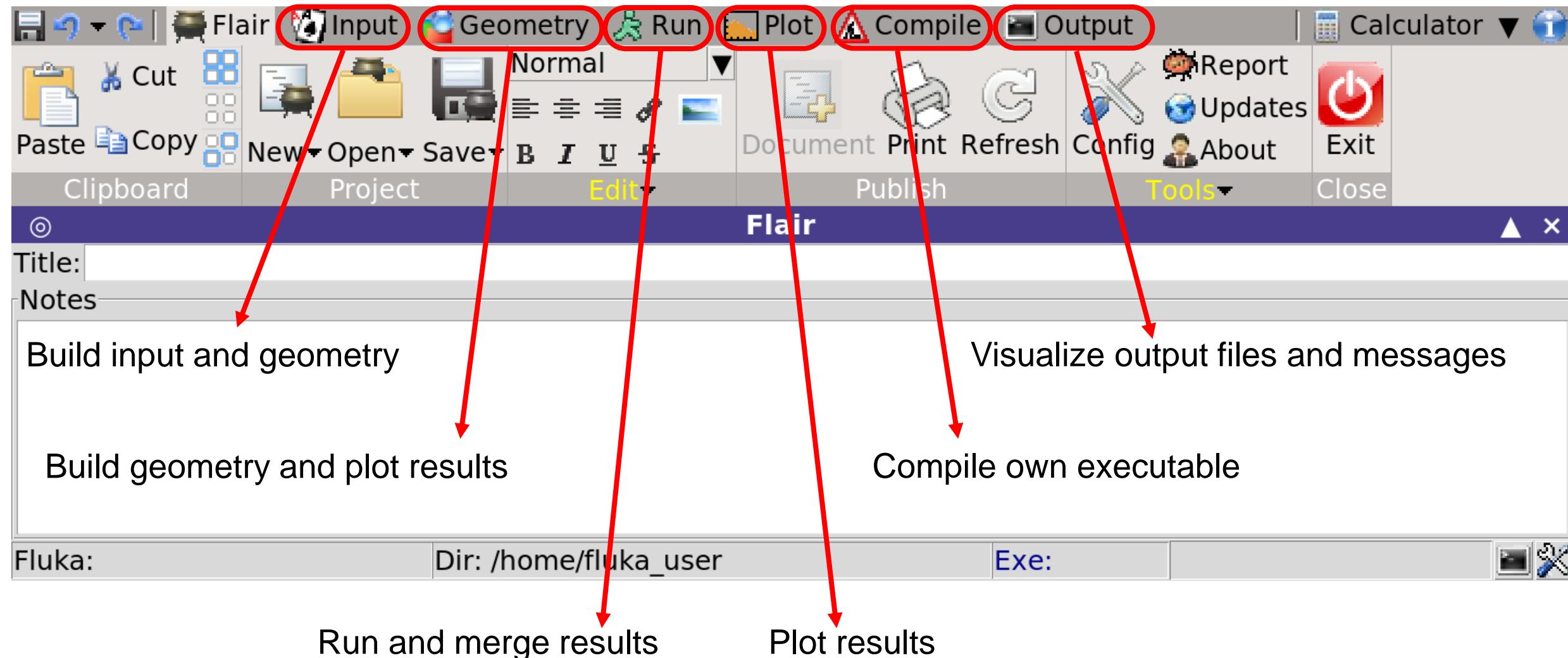


- It is possible to work with Fluka only using text editors (for expert or old users)
- Flair is not just a graphical interface for text editing
- Flair has a lot of features very useful for expert users
- This entire course will be based on Flair

Starting Flair and basic nomenclature



What's each tab for?



The input as a text file

- Mentioned here just for completeness

```
TITLE  
basic template  
* Set the defaults for precision simulations  
DEFAULTS  
* Define the beam characteristics  
BEAM  
* Define the beam position  
BEAMPOS  
GEOBEGIN  
    0 0  
* Black body  
SPH blkbody 0.0 0.0 0.0 1000000.0  
* Void sphere  
SPH void 0.0 0.0 0.0 10000.0  
* Cylindrical target  
RCC target 0.0 0.0 0.0 0.0 0.0 10.0 5.0  
END  
* Black hole  
BLKBODY 5 +blkbody -void  
* Void around  
VOID 5 +void -target  
* Target  
TARGET 5 +target  
END  
GEOEND  
* ...+....1.....+....2.....+....3.....+....4.....+....5.....+....6.....+....7...  
ASSIGNMA BLCKHOLE BLKBODY  
ASSIGNMA VACUUM VOID  
ASSIGNMA COPPER TARGET  
* Set the random number seed  
RANDOMIZ 1.0  
* Set the number of primary histories to be simulated in the run  
START  
STOP  
-:--- basic.inp All (26,69) (Fluka)
```

.inp

PRECISIO
COMBNNAME

.flair file includes
info & instructions
for the flair project

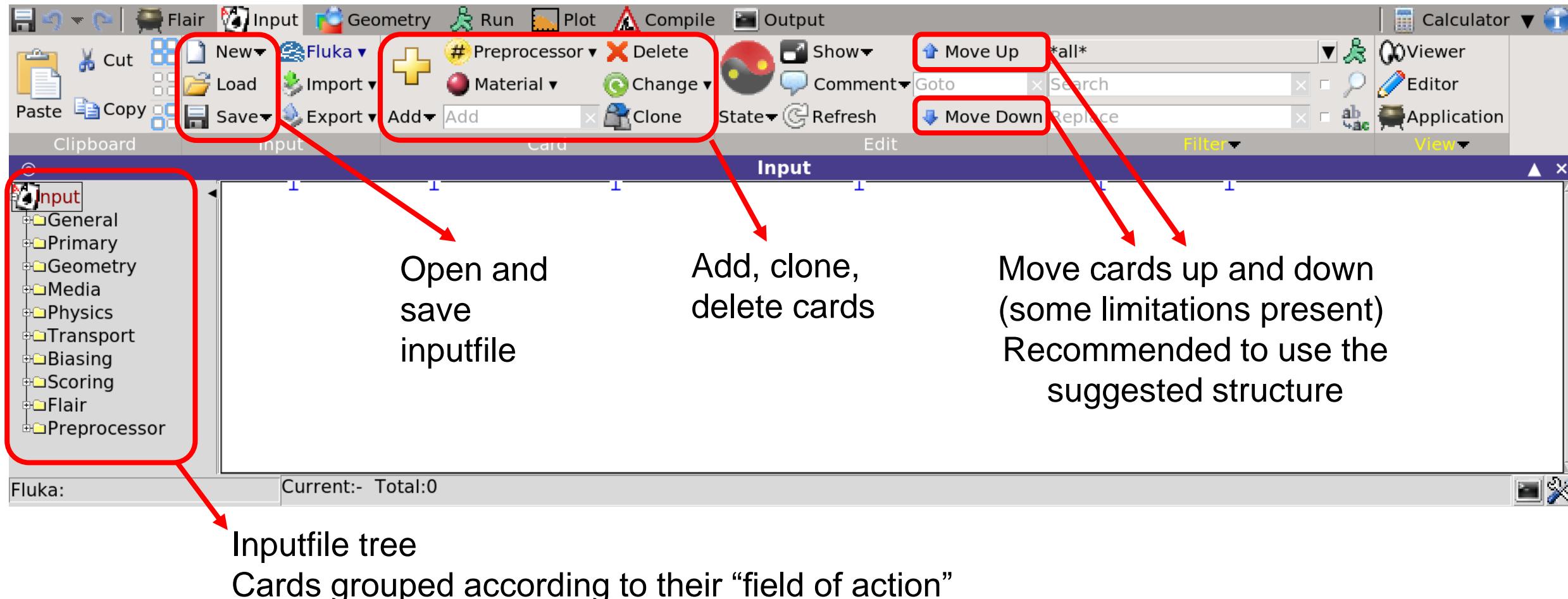
This course is based
on the use of flair,
no further mention
of these text files

```
# flair project file  
Version: 300  
Mode: fluka  
md5: c8e26fe184526e9282e8555b8fab2455  
Input:  
TITLE  
    fully-working template  
#define pointless_define_1 10  
#define pointless_define_2  
*Set the defaults for precision simulations  
DEFAULTS PRECISIO  
*Define the beam characteristics  
BEAM PROTON 0.8  
*Define the beam position  
BEAMPOS , 0. 0. -1.  
GEOBEGIN COMBNNAME  
*Black body  
SPH blkbody 0.0 0.0 0.0 1000000.0  
*Void sphere  
SPH void 0.0 0.0 0.0 10000.0  
*Cylindrical target  
RCC target 0.0 0.0 0.0 0.0 0.0 10.0 5.0  
END  
*Black hole  
REGION BLKBODY 5  
    +blkbody -void  
*Void around  
REGION VOID 5  
    +void -target  
*Target  
REGION TARGET 5  
    +target  
END  
GEOEND  
*...+....1.....+....2.....+....3.....+....4.....+....5.....+....6.....+....7...  
ASSIGNMA , BLCKHOLE BLKBODY  
ASSIGNMA , VACUUM VOID  
ASSIGNMA , COPPER TARGET  
USRBIN allpart 10 ALL-PART -21 6. 6. 11. -6. -6. -2. 120. 120. 130.  
USRBIN edep 10 ENERGY -22 6. 6. 11. -6. -6. -2. 120. 120. 130.  
*Set the random number seed  
RANDOMIZ , 1.0  
*Set the number of primary histories to be simulated in the run  
START , 10000.  
STOP  
EndInput  
Page: Plot  
# Run information  
Run: <default>  
End  
Run: test/test  
    Define:  pointless_define_2=10  
    Start: 1000  
    StartRun: 1598620157  
End  
Run: small_prod/small  
    Define:  pointless_define_2=10  
    Start: 1000  
    Last: 1
```

.flair

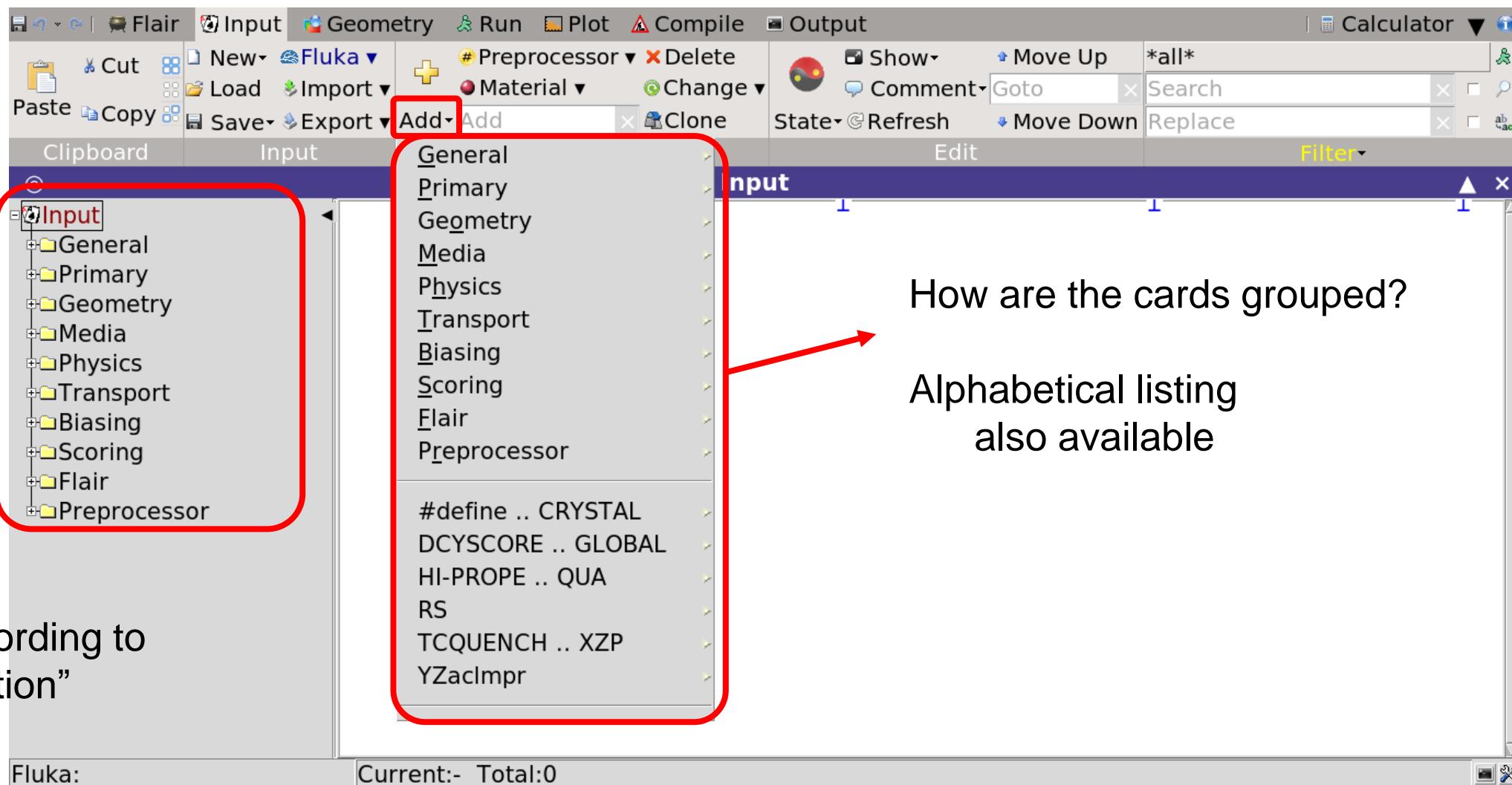
Input tab – 1: general info

- Standard looking “Windows” tab

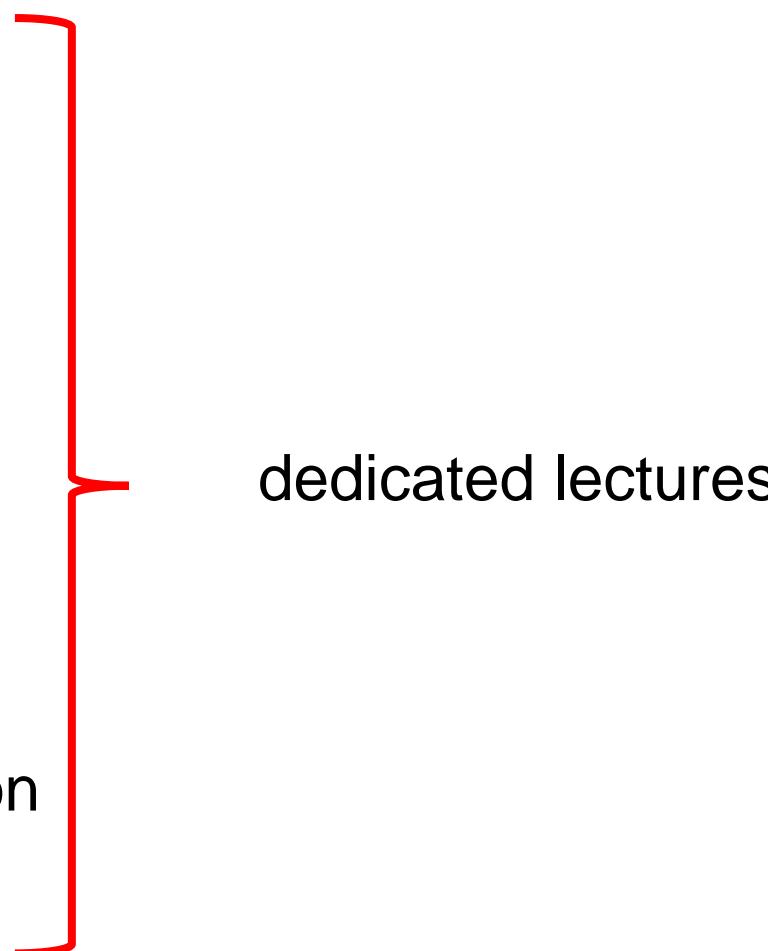


Input tab – 2: inputFile tree and card grouping

- Inputfile tree and card grouping



Input tab – 3: inputFile tree and card grouping

- General: defaults selection and other general cards → this lecture
 - Primary: definition of the particle source
 - Geometry: definition of the geometry
 - Media: definition and assignment of “materials”
 - Physics: control specific physics processes
 - Transport: control specific transport details
 - Biasing: definition of biasing
 - Scoring: definition of estimators
 - Flair: definition of flair add-ons for visualization
 - Preprocessor: definition of preprocessor instructions
- 
- 

Input tab – 4: General cards

TITLE	START	STOP	RANDOMIZE	DEFAULTS
-------	-------	------	-----------	----------

TITLE

- Not a mandatory card
- Allows to assign a title to the simulations
- The title is printed in the output files



Input tab – 5: General cards

TITLE	START	STOP	RANDOMIZE	DEFAULTS
-------	-------	------	-----------	----------

START

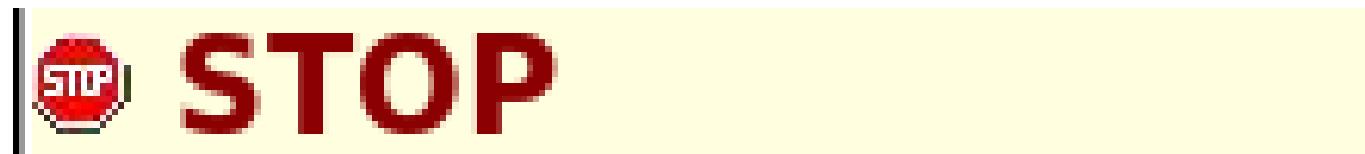
- Actually listed among the “Primary” cards
- Allows to set the number of primary particles to be simulated
- Allows to set other parameters for advanced use

Set the number of primary histories to be simulated in the run

 START	No.: 10000.	Core: ▼
	Time:	Report: default ▼

Input tab – 6: General cards

TITLE	START	STOP	RANDOMIZE	DEFAULTS
<hr/>				
STOP				
• Stop the execution of the program				
• Not really mandatory (program stops at the end of the input)				
• Can become handy for debugging purposes				



Input tab – 7: General cards

TITLE	START	STOP	RANDOMIZE	DEFAULTS
-------	-------	------	-----------	----------

RANDOMIZ

- Allows to initialize different random sequences
- For debugging purposes, the “random seed” must be the same
- Different “random seeds” are required in order to differentiate histories
- Flair takes care of the “random seeds” when spawning runs (see later)

Set the random number seed

 **RANDOMIZ**

Unit: 01 ▾

Seed: 123

Input tab – 8: General cards

TITLE	START	STOP	RANDOMIZE	DEFAULTS
-------	-------	------	-----------	----------

DEFAULTS

- Allows to select the physics defaults (list of predefined defaults available)
- Physics defaults can be overridden with specific cards
- Can be preceded only by the **TITLE** and **GLOBAL** cards
- Given the progress over time in computer power, it is a reasonable approach to:
 - always select the most detailed physics defaults: **PRECISIO**
 - depending on the needs of the problem, override specific defaults

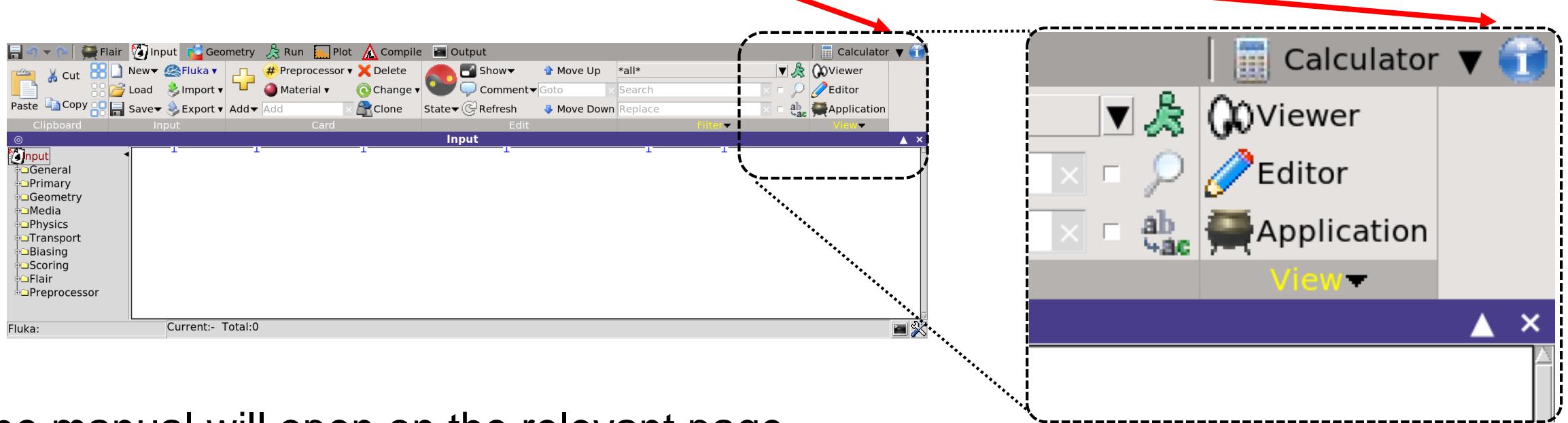
Set the defaults for precision simulations

 **DEFAULTS**

: PRECISIO ▾

The manual

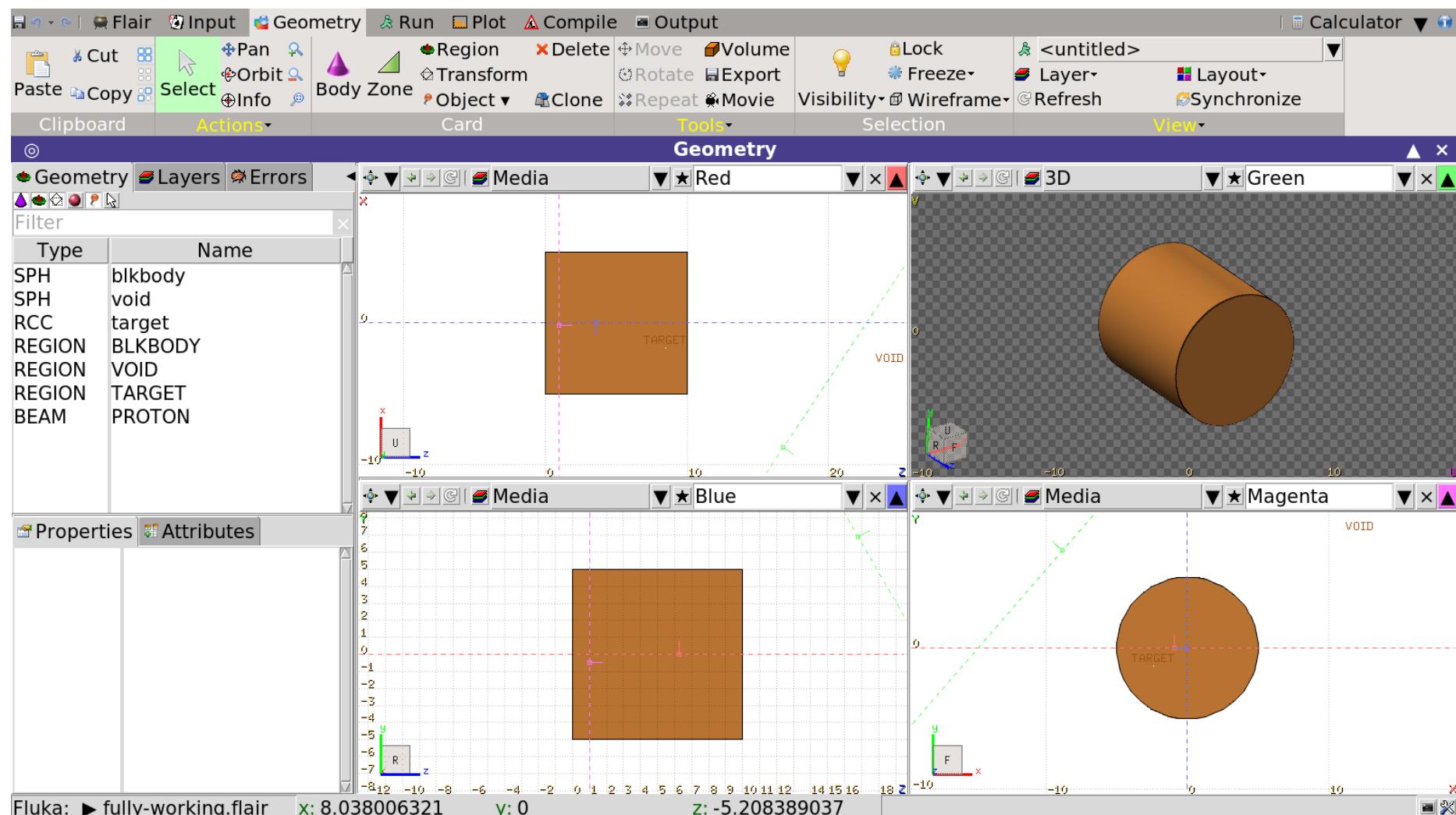
- Can be accessed using F1 button
- Can be accessed clicking on the “info” button



- The manual will open on the relevant page
- The manual is also available on the Fluka web page www.fluka.cern

Geometry tab – 1

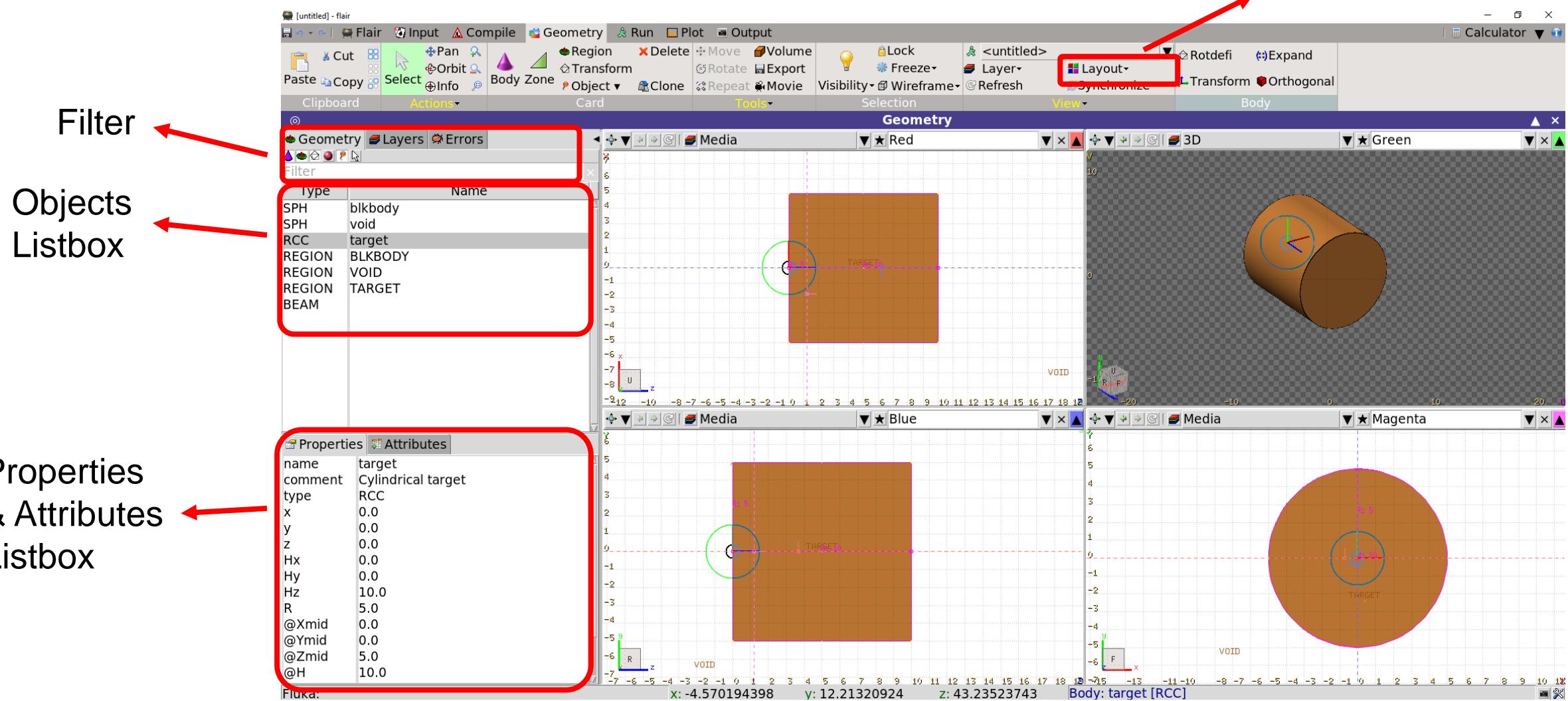
- Visualize and edit geometry
- Plot results
- Dedicated lectures



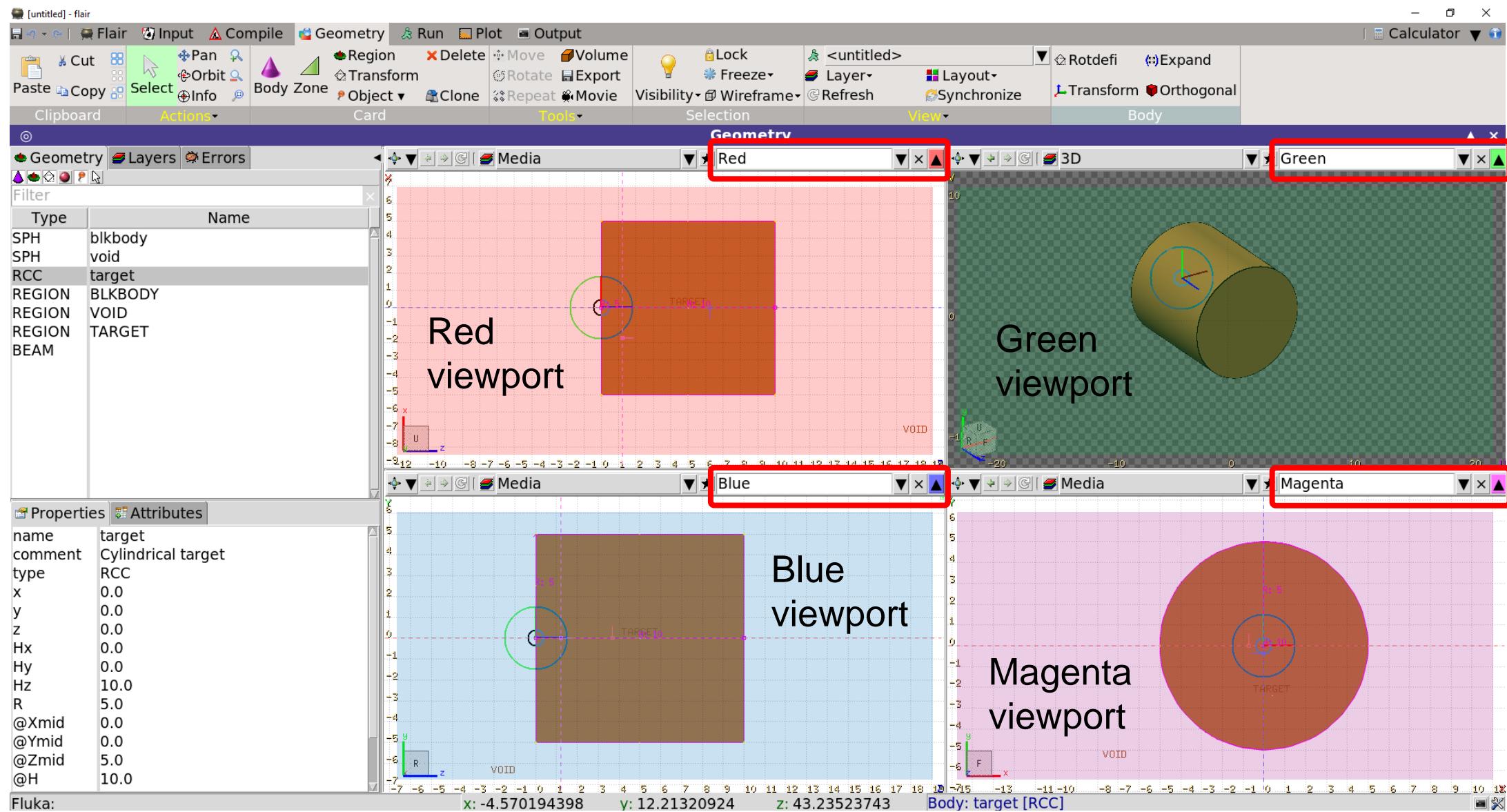
Geometry tab – 2

- Viewports automatically refreshed when input is changed

Layout management

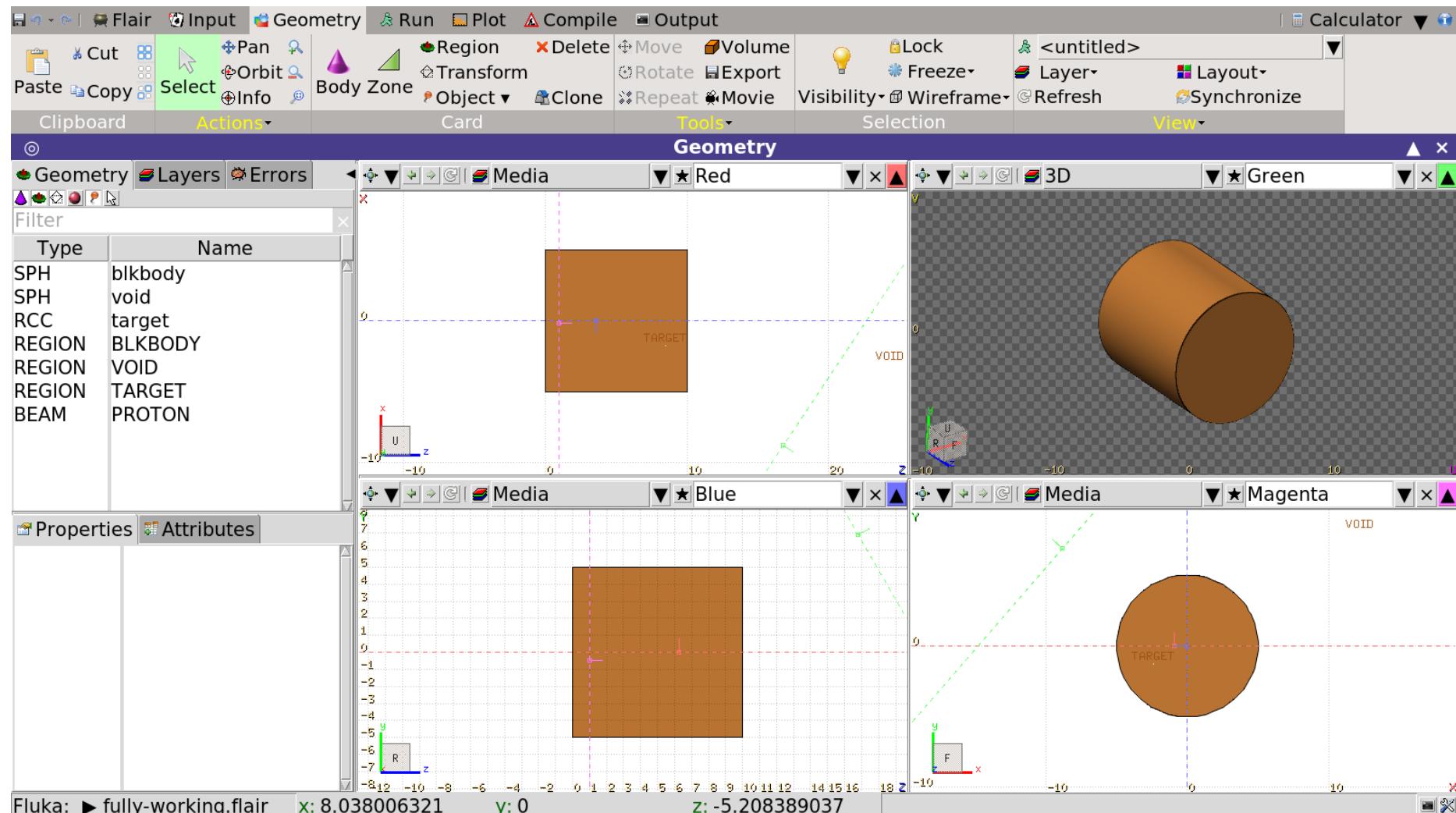


Geometry tab – 3



Geometry tab – 4

- Possible to navigate with mouse and keyboard (see dedicated lecture)



Run tab

- 3 views: “Runs”, “Files”, and “Data”

The screenshot shows the Flair application window with the 'Run' tab selected. The toolbar at the top includes icons for Flair, Input, Geometry, Run, Plot, Compile, Output, Cut, Copy, Paste, and various actions like Remove, Move Up, Move Down, Clone, Loop, Rename, Attach, Clean, Cycle, Run, Kill, Refresh, and Start. Below the toolbar is a menu bar with 'Clipboard' and 'View'. The main area is divided into sections: 'Run' (containing a tree view of simulations), 'Override' (containing simulation details), and 'Progress' (containing status information). A red box highlights the 'Runs' icon in the toolbar, and a red arrow points from it to the 'Run' section of the interface. Another red arrow points from the 'Override' section to the 'Run' section, indicating they share the same listbox.

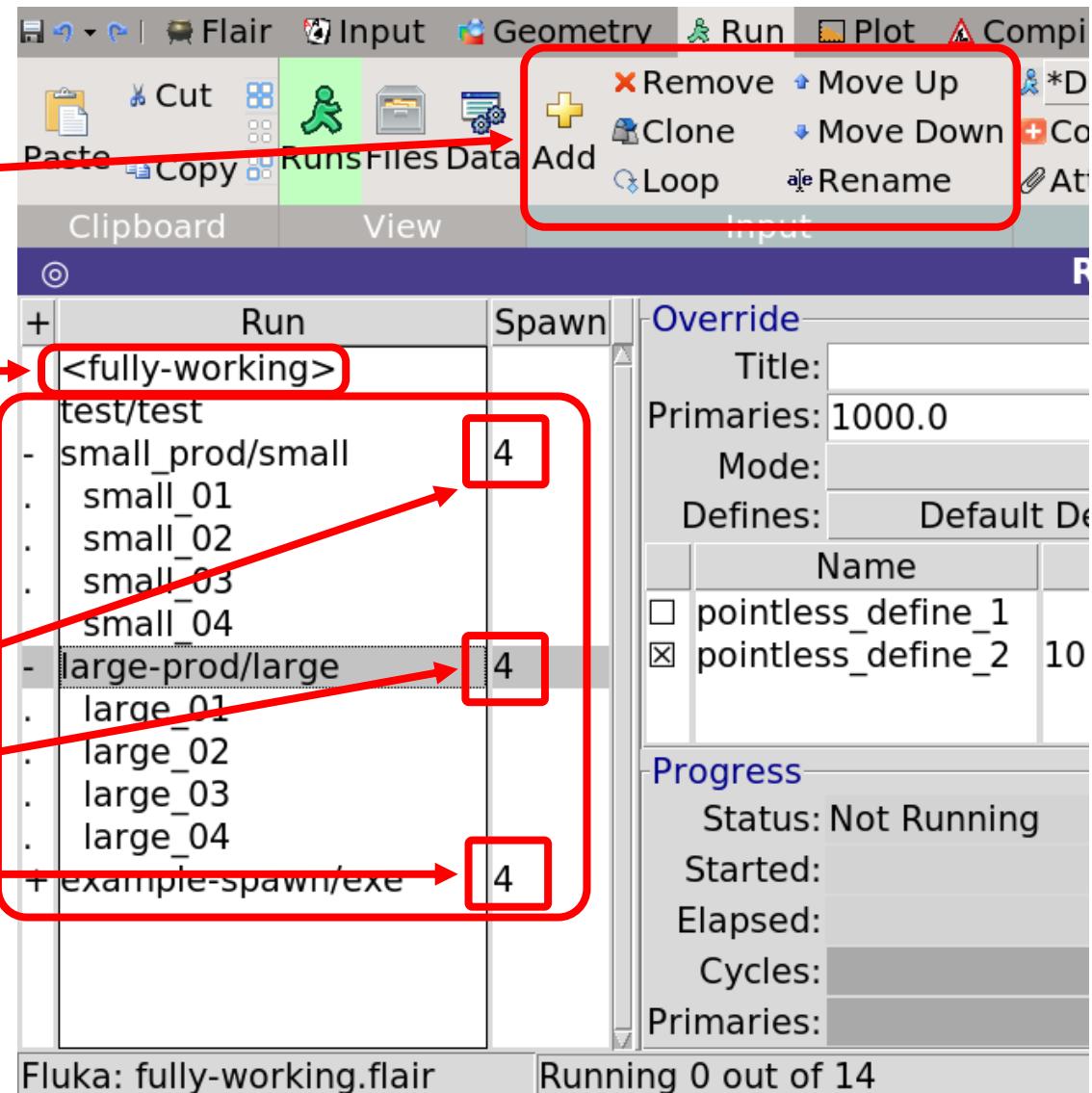
• Shared listbox:
List of simulations
associated with the project

Run	Spawn
<fully-working>	
test/test	
- small_prod/small	4
. small_01	
. small_02	
. small_03	
. small_04	
- large-prod/large	4
. large_01	
. large_02	
. large_03	
. large_04	
+ example-spawn/exe	4

Fluka: fully-working.flair Running 0 out of 14

Run tab – Runs view – 1

- Management of the various simulations



- Basic inputfile of the Flair project

- Different simulations associated with the Flair project

- Number of spawns

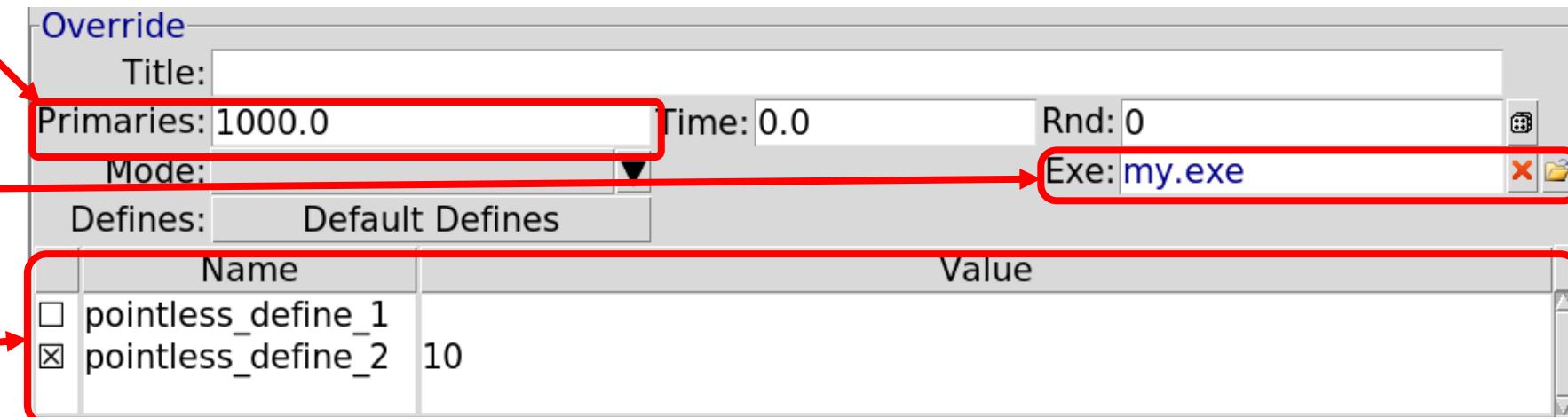
Run tab – Runs view – 2

- Override of inputs

- Number of primaries

- Executable

- #define



Dedicated lecture

Run tab – Runs view – 3

- Cycles control: how many cycles to run, starting from which cycle

- Cleanly stop the cycles/runs currently running



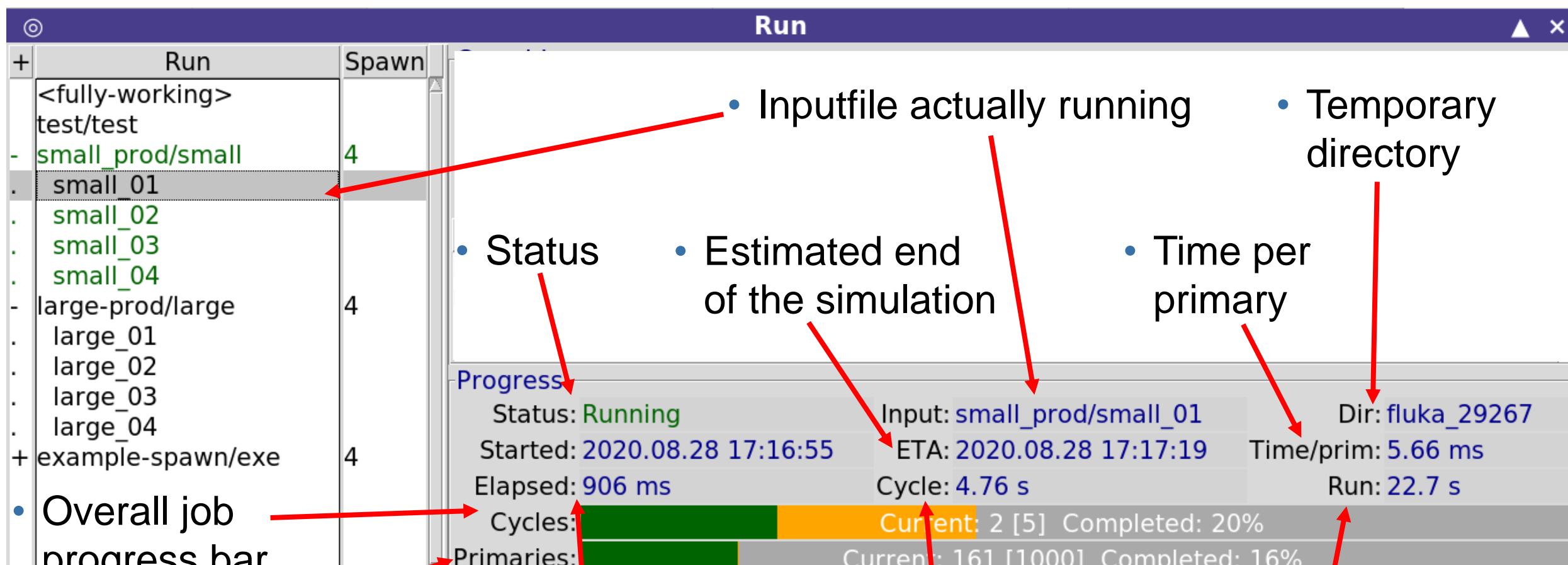
- Remove files from a previous simulation

- Start a simulation

- Refresh the progress field

- Kill the current simulations

Run tab – Runs view – 4



Run tab – Runs view – 5

- At the end of the simulations...

The screenshot shows the FLAIR interface with the 'Run' tab selected. On the left, there's a tree view of runs:

- <fully-working>
- test/test
- small_prod/small
 - . small_01
 - . small_02
 - . small_03
 - . small_04
- large-prod/large
 - . large_01
 - . large_02
 - . large_03
 - . large_04
- + example-spawn/exe

In the center, the 'Runs view' window shows a table for the 'small_01' run. The table has two tabs: 'Run' (selected) and 'Override'. The 'Run' tab displays the following information:

Progress	Input	Dir:
Status: Finished OK	Input: small_prod/small_01	Dir:
Started:	ETA:	Time/prim:
Elapsed:	Cycle:	Run:
Cycles:		
Primaries:		

A red arrow points from the 'Status' bullet point in the list below to the 'Status: Finished OK' entry in the table.

- Status
- WARNING:
“Finished OK” means OK from the computing point of view, there is no guarantee that the output of the simulation is physically meaningful!

After running – 1

- Content of the working directory

```
fluka_user:/home/fluka_user$  
fluka_user:/home/fluka_user$ ls  
fully-working.flair  fully-working.inp  my.exe  small_prod  tutorial.flair  
fluka_user:/home/fluka_user$  
fluka_user:/home/fluka_user$ -
```

- Content of the working sub-directory

```
fluka_user:/home/fluka_user$ cd small_prod/  
fluka_user:/home/fluka_user/small_prod$ ls  
ransmall_01001  ransmall_04005      small_01004_fort.21  small_02003_fort.21  small_03002_fort.21  small_04001_fort.21  
ransmall_01002  ransmall_04006      small_01004_fort.22  small_02003_fort.22  small_03002_fort.22  small_04001_fort.22  
ransmall_01003  small_01.inp       small_01005.err       small_02004.err       small_03003.err       small_04002.err  
ransmall_01004  small_01.out       small_01005.log       small_02004.log       small_03003.log       small_04002.log  
ransmall_01005  small_01001.err    small_01005.out      small_02004.out      small_03003.out      small_04002.out  
ransmall_01006  small_01001.log    small_01005_fort.21  small_02004_fort.21  small_03003_fort.21  small_04002_fort.21  
ransmall_02001  small_01001.out    small_01005_fort.22  small_02004_fort.22  small_03003_fort.22  small_04002_fort.22  
ransmall_02002  small_01001_fort.21 small_02.inp        small_02005.err       small_03004.err       small_04003.err  
ransmall_02003  small_01001_fort.22 small_02.out        small_02005.log       small_03004.log       small_04003.log  
ransmall_02004  small_01002.err    small_02001.err       small_02005.out      small_03004.out      small_04003.out  
ransmall_02005  small_01002.log    small_02001.log       small_02005_fort.21  small_03004_fort.21  small_04003_fort.21  
ransmall_02006  small_01002.out    small_02001.out      small_02005_fort.22  small_03004_fort.22  small_04003_fort.22  
ransmall_03001  small_01002_fort.21 small_02001_fort.21  small_03.inp        small_03005.err       small_04004.err  
ransmall_03002  small_01002_fort.22 small_02001_fort.22  small_03.out        small_03005.log       small_04004.log  
ransmall_03003  small_01003.err    small_02002.err       small_03001.err       small_03005.out       small_04004.out  
ransmall_03004  small_01003.log    small_02002.log       small_03001.log       small_03005_fort.21  small_04004_fort.21  
ransmall_03005  small_01003.out    small_02002.out      small_03001.out      small_03005_fort.22  small_04004_fort.22  
ransmall_03006  small_01003_fort.21 small_02002_fort.21  small_03001_fort.21  small_04.inp        small_04005.err  
ransmall_04001  small_01003_fort.22 small_02002_fort.22  small_03001_fort.22  small_04.out        small_04005.log  
ransmall_04002  small_01004.err    small_02003.err       small_03002.err       small_04001.err       small_04005.out  
ransmall_04003  small_01004.log    small_02003.log       small_03002.log       small_04001.log       small_04005_fort.21  
ransmall_04004  small_01004.out    small_02003.out      small_03002.out      small_04001.out       small_04005_fort.22  
fluka_user:/home/fluka_user/small_prod$ -
```

After running – 2

- Content of the working directory

```
fluka_user:/home/fluka_user$  
fluka_user:/home/fluka_user$ ls  
fully-working.flair  fully-working.inp  my.exe  small_prod  tutorial.flair  
fluka_user:/home/fluka_user$  
fluka_user:/home/fluka_user$ -
```

```
fluka_user:/home/fluka_user$ cd small_prod/  
fluka_user:/home/fluka_user/small_prod$ ls  
ransmall_01001  ransmall_04005      small_01004_fort.21  small_02003_fort.21  small_03002_fort.21  small_04001_fort.21  
ransmall_01002  ransmall_04006      small_01004_fort.22  small_02003_fort.22  small_03002_fort.22  small_04001_fort.22  
ransmall_01003  small_01.inp       small_01005.err      small_02004.err      small_03004.err      small_04004.err  
ransmall_01004  small_01.out       small_01005.log     small_02004.log     small_03004.log     small_04004.log  
ransmall_01005  small_01001.err    small_01005.out     small_02004.out     small_03004.out     small_04004.out  
ransmall_01006  small_01001.log    small_01005_fort.21 small_02004_fort.21 small_03004_fort.21 small_04004_fort.21  
ransmall_02001  small_01001.out    small_01005_fort.22 small_02004_fort.22 small_03004_fort.22 small_04004_fort.22  
ransmall_02002  small_01001_fort.21 small_02005.err      small_03004_err      small_04004_err      small_04005_err  
ransmall_02003  small_01001_fort.22 small_02005.log     small_03004.log     small_04004.log     small_04005.log  
ransmall_02004  small_01002.err    small_02005.out     small_03004.out     small_04004.out     small_04005.out  
ransmall_02005  small_01002.log    small_02005_fort.21 small_03004_fort.21 small_04004_fort.21 small_04005_fort.21  
ransmall_02006  small_01002.out    small_02005_fort.22 small_03004_fort.22 small_04004_fort.22 small_04005_fort.22  
ransmall_03001  small_01002_fort.21 small_02001_fort.21  small_03001_fort.21  small_04001_fort.21  small_04004_fort.21  
ransmall_03002  small_01002_fort.22 small_02001_fort.22  small_03001_fort.22  small_04001_fort.22  small_04004_fort.22  
ransmall_03003  small_01003.err    small_02002.err      small_03001_err      small_04001_err      small_04004_err  
ransmall_03004  small_01003.log    small_02002.log     small_03001.log     small_04001.log     small_04004.log  
ransmall_03005  small_01003.out    small_02002.out     small_03001.out     small_04001.out     small_04004.out  
ransmall_03006  small_01003_fort.21 small_02002_fort.21 small_03001_fort.21 small_04001_fort.21 small_04004_fort.21  
ransmall_04001  small_01003_fort.22 small_02002_fort.22 small_03001_fort.22 small_04001_fort.22 small_04004_fort.22  
ransmall_04002  small_01004.err    small_02003.err      small_03002_err      small_04002_err      small_04005_err  
ransmall_04003  small_01004.log    small_02003.log     small_03002.log     small_04002.log     small_04005.log  
ransmall_04004  small_01004.out    small_02003.out     small_03002.out     small_04002.out     small_04005.out  
fluka_user:/home/fluka_user/small_prod$ -
```

.inp and .out files specific of each spawn

- Content of the working sub-directory

Run tab – Files view – 1

- Generated files accessible via the Files view

Run	Spawn	Cycles	File	Type	Size	Date
<fully-working>		001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
test/test		002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
- small_prod/small	4	003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
. small_01		004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
. small_02		005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
. small_03		006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
. small_04		compile				
- large-prod/large	4	data				
. large_01		input				
. large_02		plot				
. large_03		temporary				
. large_04						
+ example-spawn/exe	4					

Run tab – Files view – 2

- File per each cycle:
 - one (1) .out file
 - one (1) .log file
 - one (1) .err file
 - one (1) random seed file
 - one (1) scoring file per each logical unit scoring used

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

Run tab – Files view – 3

- Naming convention for file names; the filename contains:
 - the name of the run, e.g.: **small**
 - The spawn identifier, e.g.: **01**
 - The cycle identifier, e.g.: **001**
 - The file type identifier, e.g.: **.err , fort.21 , ran**

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

Run tab – Files view – 4

- Naming convention for file names; the filename contains:
 - the name of the run, e.g.: **small**
 - The spawn identifier, e.g.: **01**
 - The cycle identifier, e.g.: **001**
 - The file type identifier, e.g.: **.err , fort.21 , ran**
- In this example 6 files were generated:

small_01001.err

small_01001.log

small_01001.out

ransmall_01001

small_01001_fort.21

small_01001_fort.22

Run tab – Files view – 4

- Naming convention for file names; the filename contains:
 - the name of the run, e.g.: **small**
 - The spawn identifier, e.g.: **01**
 - The cycle identifier, e.g.: **001**
 - The file type identifier, e.g.: **.err , fort.21 , ran**
- In this example 6 files were generated:

small_01001.err

small_01001.log

small_01001.out

ransmall_01001

small_01001_fort.21

small_01001_fort.22

renaming is planned

Run tab – Files view – 5

- Spawn 1 Cycle 1

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

- Spawn 1 Cycle 5

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01005	-file-	1651	2020.08.28 17:17
002	small_prod/small_01005.err	Error	22470	2020.08.28 17:17
003	small_prod/small_01005.log	Log	0	2020.08.28 17:17
004	small_prod/small_01005.out	Output	104249	2020.08.28 17:17
005	small_prod/small_01005_fort.21	21	7488238	2020.08.28 17:17
006	small_prod/small_01005_fort.22	22	7488238	2020.08.28 17:17
compile				
data				
input				
plot				
temporary				

Run tab – Files view – 6

- Spawn 1 Cycle 1

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

- Spawn 2 Cycle 1

Cycles	File	Type	Size	Date
001	small_prod/ransmall_02001	-file-	1651	2020.08.28 17:16
002	small_prod/small_02001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_02001.log	Log	0	2020.08.28 17:16
004	small_prod/small_02001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_02001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_02001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

Run tab – Files view – 7

- Spawn 1 Cycle 1

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01001	-file-	1651	2020.08.28 17:16
002	small_prod/small_01001.err	Error	22470	2020.08.28 17:16
003	small_prod/small_01001.log	Log	0	2020.08.28 17:16
004	small_prod/small_01001.out	Output	104372	2020.08.28 17:16
005	small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16
006	small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16
compile				
data				
input				
plot				
temporary				

- Spawn 1 Cycle 6

Cycles	File	Type	Size	Date
001	small_prod/ransmall_01006	-file-	1651	2020.08.28 17:17
002				
003				
004				
005				
006				
compile				
data				
input				
plot				
temporary				

- Random file for the next cycle is generated

Run tab – Data view – 1

- All the generated files need to be merged to be analyzed

`small_01001_fort.21`

`small_01002_fort.21`

`small_01003_fort.21`

`small_01004_fort.21``

`small_01005_fort.21`

`small_02001_fort.21`

`small_02002_fort.21`

`small_02003_fort.21`

`small_02004_fort.21`

`small_02005_fort.21`

`small_03001_fort.21`

`small_03002_fort.21`

`small_03003_fort.21`

`small_03004_fort.21`

`small_03005_fort.21`

`small_04001_fort.21`

`small_04002_fort.21`

`small_04003_fort.21`

`small_04004_fort.21`

`small_04005_fort.21`

Run tab – Data view – 2

- Flair automatically identifies the logical units used from the inputfile

The screenshot shows the Flair application interface with the 'Run' tab selected. The 'Data' icon in the toolbar is highlighted. The 'Run' tab displays a hierarchical list of logical units:

- <fully-working>
- test/test
- small_prod/small
 - . small_01
 - . small_02
 - . small_03
 - . small_04
- large-prod/large
 - . large_01
 - . large_02
 - . large_03
 - . large_04
- + example-spawn/exe

The 'Spawn' tab shows the following table:

Run	Type	Output	Name/Unit
small_prod/small	usrbin	small_prod/small_21.bnn	21
small_prod/small	usrbin	small_prod/small_22.bnn	22

The 'Detectors' tab shows the following table:

Run	Type	Output	Name/Unit
small_prod/small	usrbin	small_prod/small_21.bnn	21
small_prod/small	usrbin	small_prod/small_22.bnn	22

The status bar at the bottom left shows 'Fluka: fully-working.flair' and 'Files: 40'.

Run tab – Data view – 3

- Flair finds all the corresponding file (per spawn and per cycle)

The screenshot shows the Flair application interface with the 'Run' tab selected. The 'Data' tab is active in the top navigation bar. The main workspace displays two tables: 'Detectors' and 'Files'.

Detectors Table:

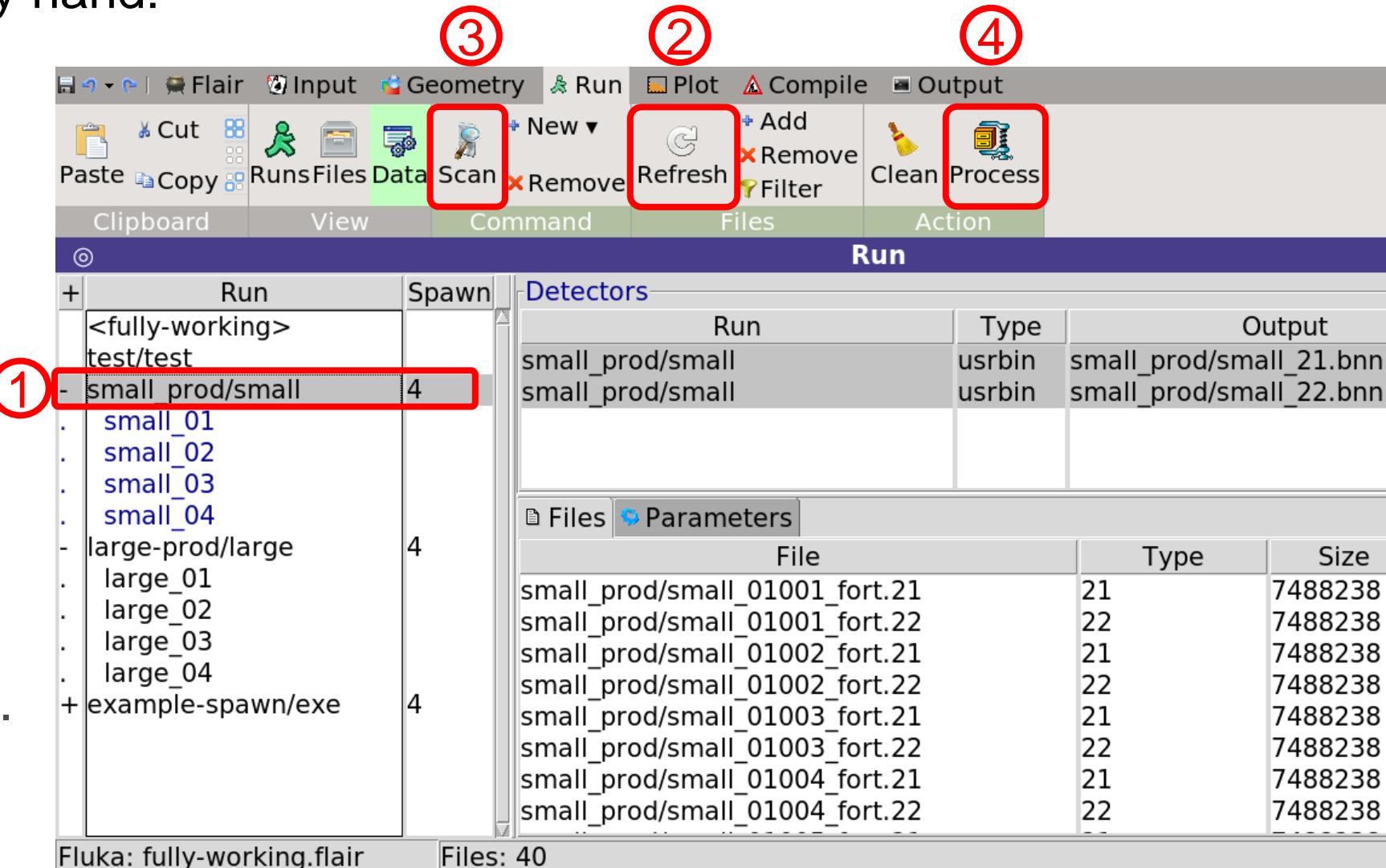
Run	Spawn	Detectors			
		Run	Type	Output	Name/Unit
<fully-working>		small_prod/small	usrbin	small_prod/small_21.bnn	21
test/test		small_prod/small	usrbin	small_prod/small_22.bnn	22
- small_prod/small	4				
. small_01					
. small_02					
. small_03					
. small_04					
- large-prod/large	4				
. large_01		small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16:54
. large_02		small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16:54
. large_03		small_prod/small_01002_fort.21	21	7488238	2020.08.28 17:17:02
. large_04		small_prod/small_01002_fort.22	22	7488238	2020.08.28 17:17:02
+ example-spawn/exe	4	small_prod/small_01003_fort.21	21	7488238	2020.08.28 17:17:10
		small_prod/small_01003_fort.22	22	7488238	2020.08.28 17:17:10
		small_prod/small_01004_fort.21	21	7488238	2020.08.28 17:17:18
		small_prod/small_01004_fort.22	22	7488238	2020.08.28 17:17:18

Files Table:

File	Type	Size	Date
small_prod/small_01001_fort.21	21	7488238	2020.08.28 17:16:54
small_prod/small_01001_fort.22	22	7488238	2020.08.28 17:16:54
small_prod/small_01002_fort.21	21	7488238	2020.08.28 17:17:02
small_prod/small_01002_fort.22	22	7488238	2020.08.28 17:17:02
small_prod/small_01003_fort.21	21	7488238	2020.08.28 17:17:10
small_prod/small_01003_fort.22	22	7488238	2020.08.28 17:17:10
small_prod/small_01004_fort.21	21	7488238	2020.08.28 17:17:18
small_prod/small_01004_fort.22	22	7488238	2020.08.28 17:17:18

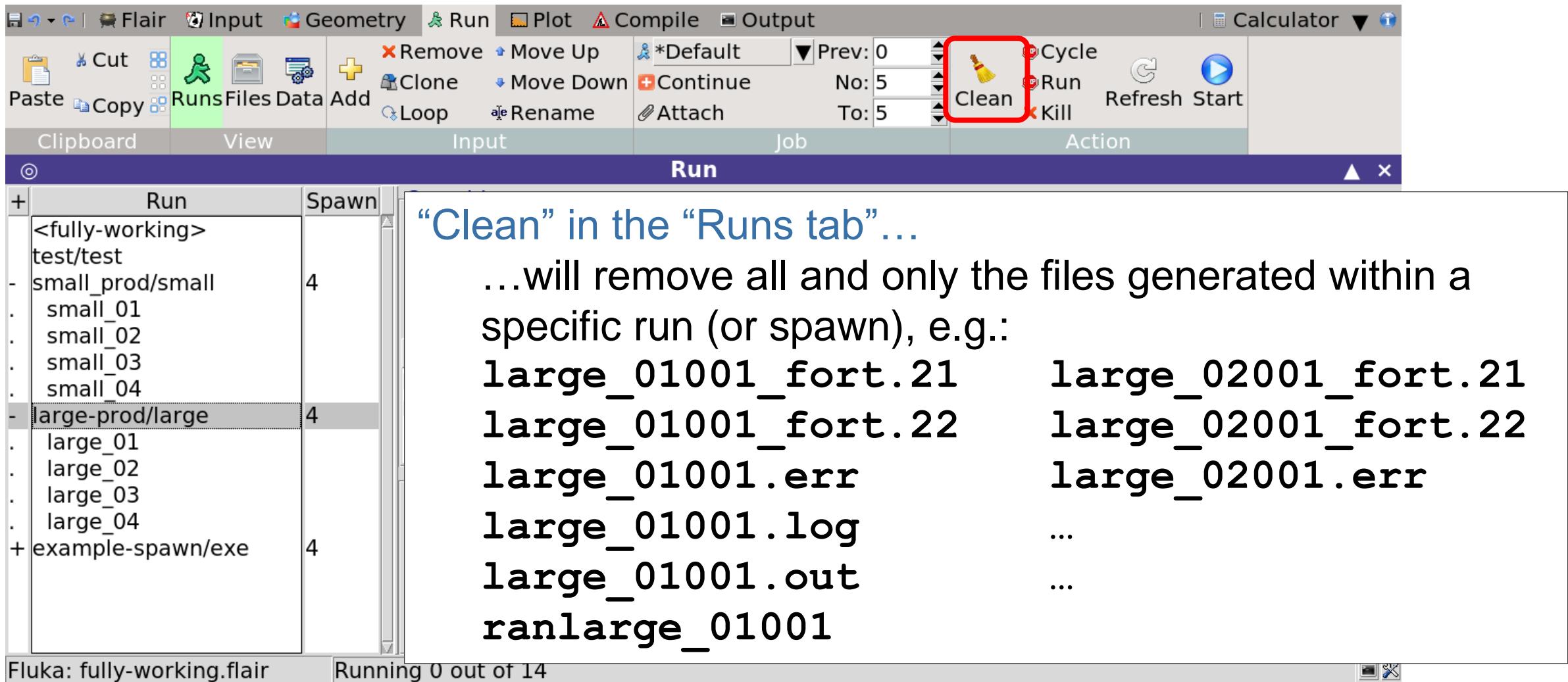
Run tab – Data view – 4

- Process can be forced by hand:
- 1-Select the run
- 2-Refresh
- 3-Scan
- 4-Process (merge)
- Processed binary results
files are generated
(specific extensions:
.bnn, .bnx, .rnc, etc.
more in other lectures)



Run tab – Cleaning – 1

- Removing files generated for the cycles and merged files are different actions!



The screenshot shows the Flair application window with the "Run" tab selected. In the top menu bar, the "Run" tab is highlighted. Below the menu, there is a toolbar with various icons for file operations like Cut, Copy, Paste, and a "Clean" icon, which is highlighted with a red box. The main area displays a tree view of runs and spawns. A tooltip box is overlaid on the right side of the screen, containing the text: "'Clean' in the 'Runs tab'... ...will remove all and only the files generated within a specific run (or spawn), e.g.:". Below this text, a list of file names is shown, all starting with "large_01001_":

- large_01001_fort.21
- large_01001_fort.22
- large_01001.err
- large_01001.log
- large_01001.out
- ranlarge_01001

At the bottom of the window, status bars indicate "Fluka: fully-working.flair" and "Running 0 out of 14".

Run tab – Cleaning – 2

- Removing files generated for the cycles and merged files are different actions!

The screenshot shows the Flair application window with the 'Run' tab selected. In the toolbar under the 'Action' section, the 'Clean' button (represented by a broom icon) is highlighted with a red box. Below the toolbar, the 'Run' tab is active, displaying a list of runs and their details. A callout box with a blue border and white background contains the text: "‘Clean’ in the ‘Data tab’... ...will remove only the merged results, e.g.: **small_21.bnn** **small_22.bnn**".

Run	Spawn	Detectors
<fully-working>		
test/test		
- small_prod/small	4	Run Type Output Name/Unit
small_01		small_prod/small usrbin small_prod/small_21.bnn 21
small_02		small_prod/small usrbin small_prod/small_22.bnn 22
small_03		
small_04		
- large-prod/large	4	
large_01		
large_02		
large_03		
large_04		
+ example-spawn/exe	4	

Fluka: fully-working.flair Files: 40

Compile tab

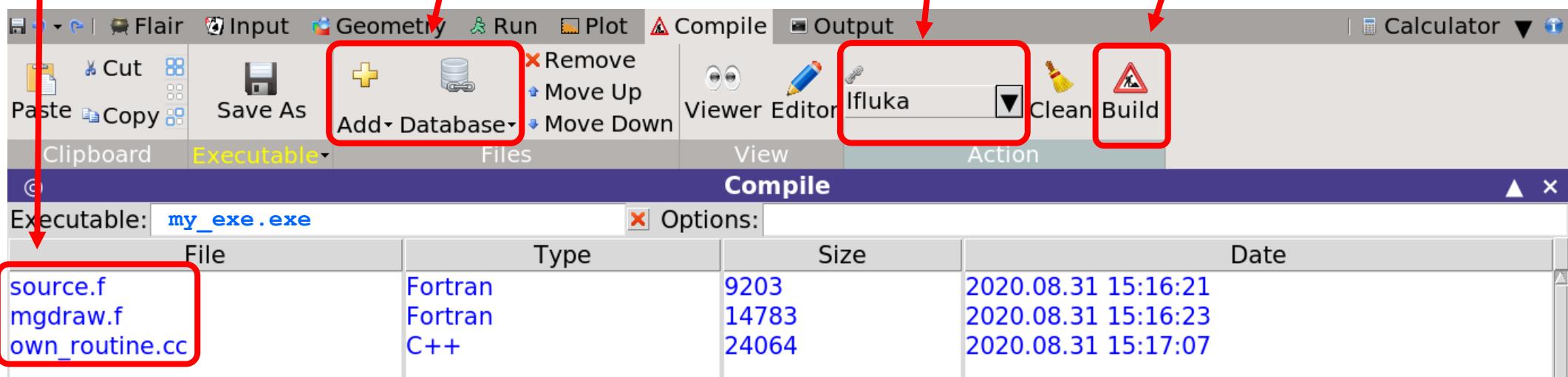
- Only very basic information is given here

- Routine to be compiled

- Add routines

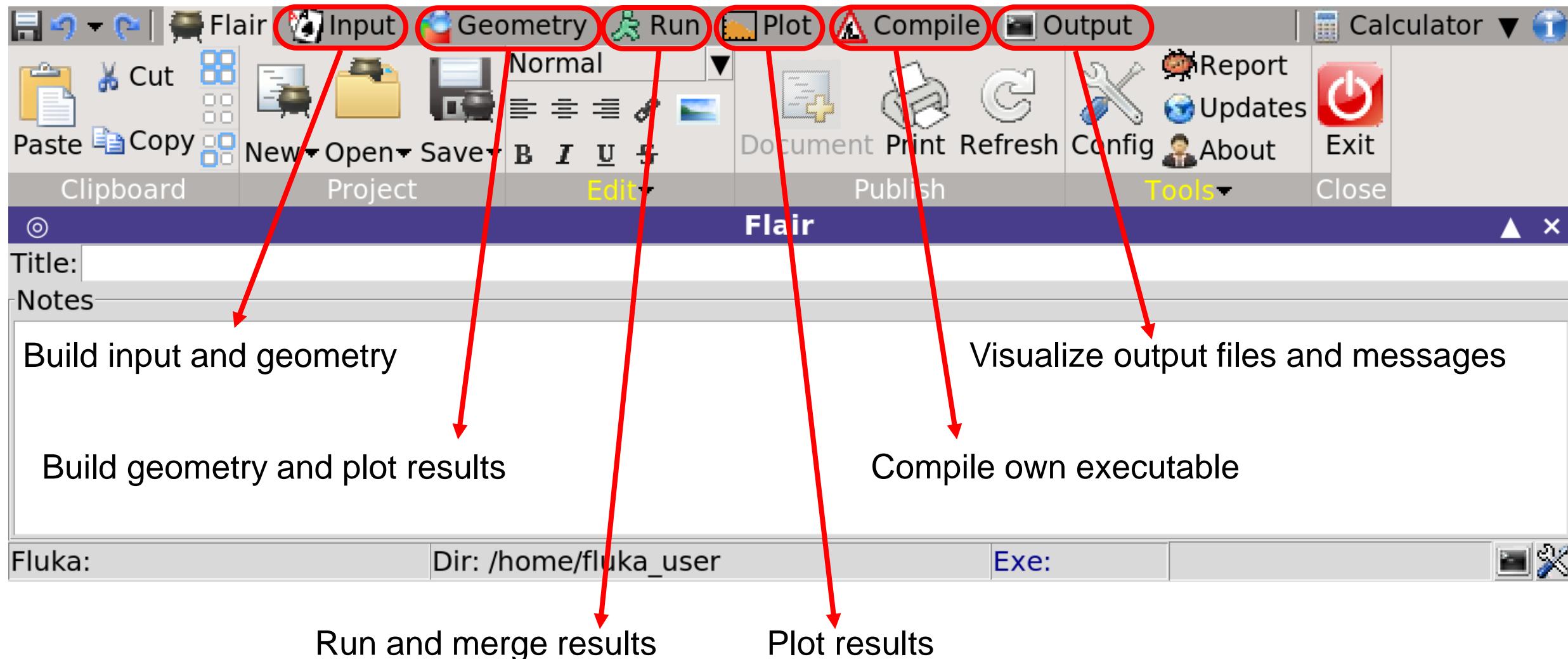
- Select linker

- Build executable

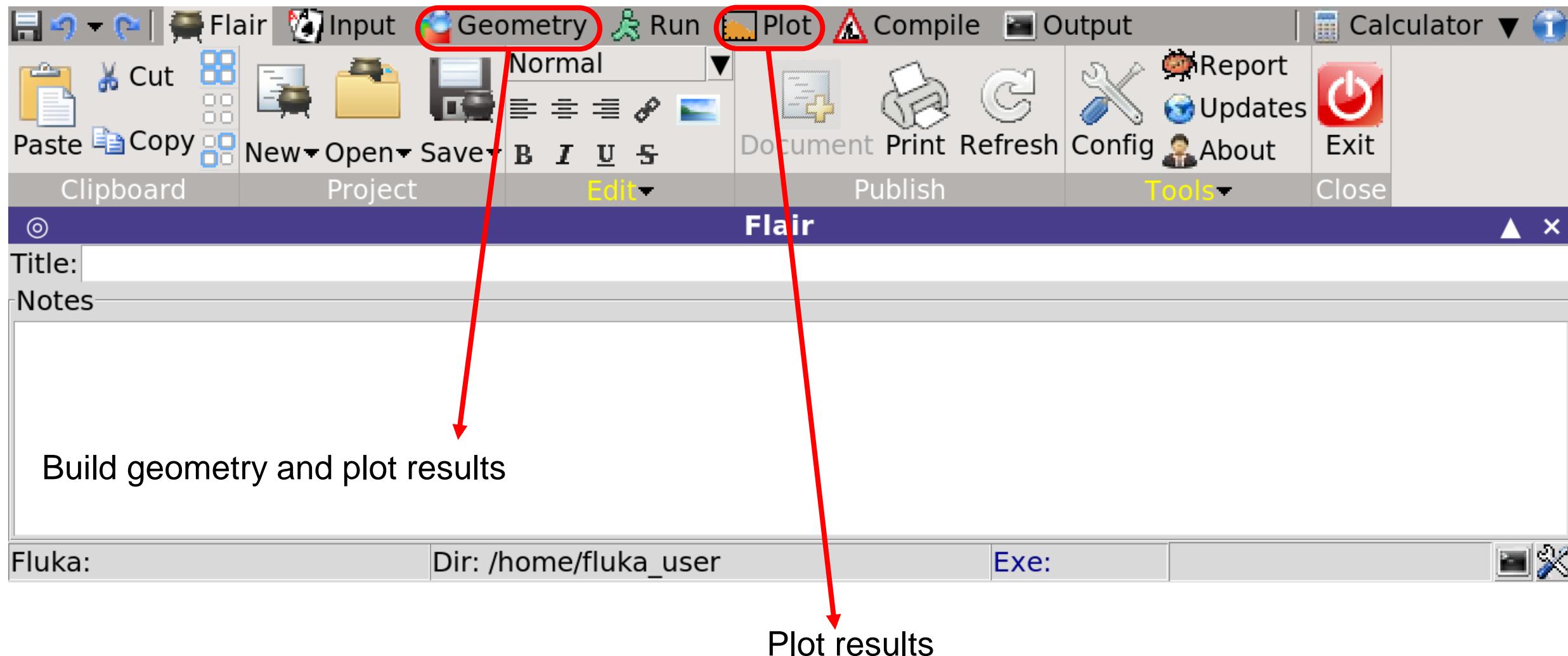


- User routine are discussed in advanced courses

Do you remember slide 6?

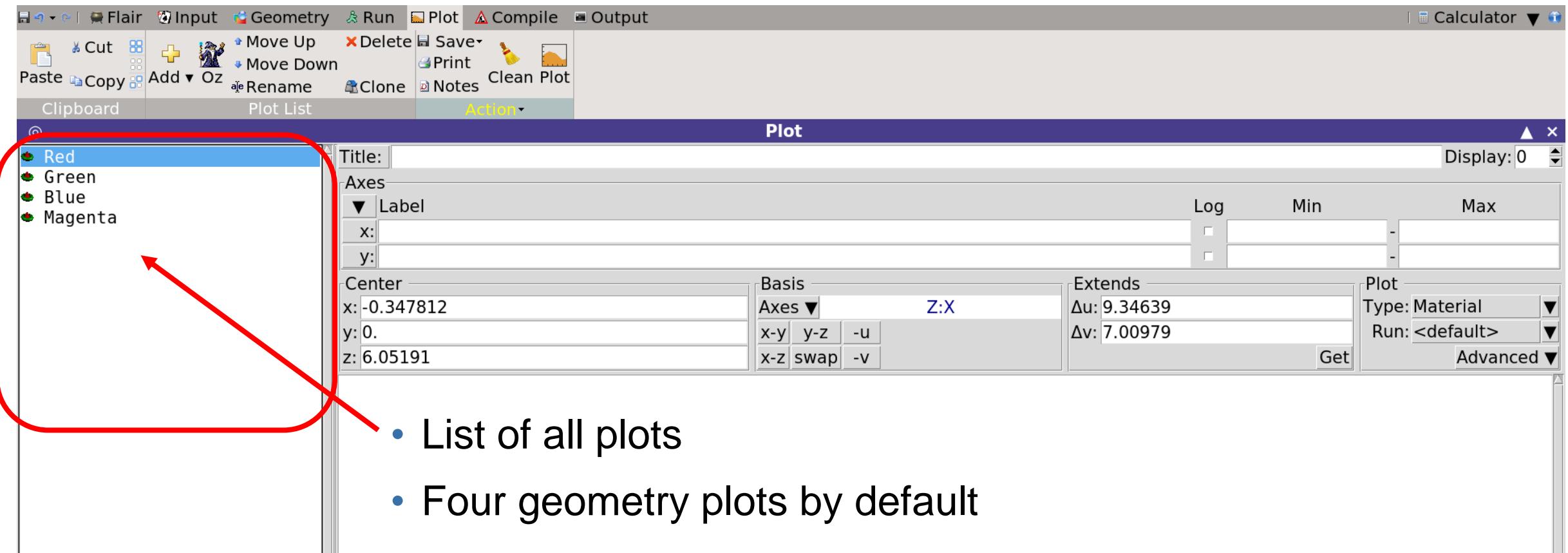


Do you remember slide 6?



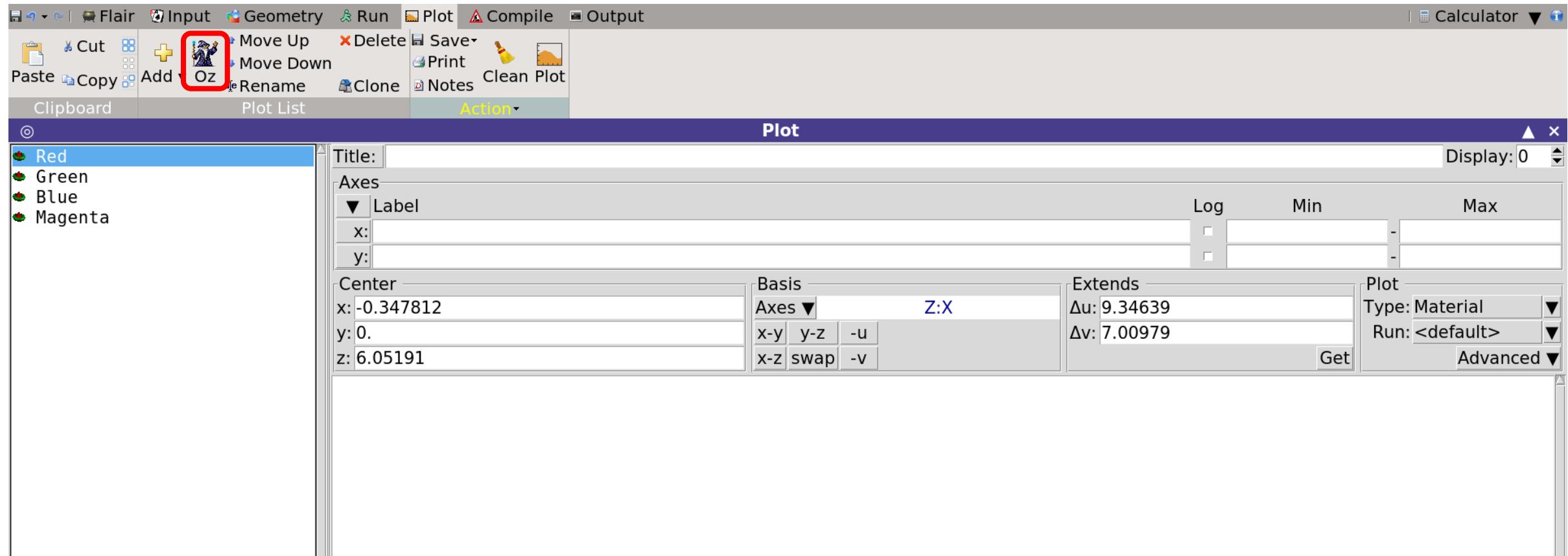
Plotting result in the Plot tab – 1

- Possible to plot geometry and all built-in scorings results



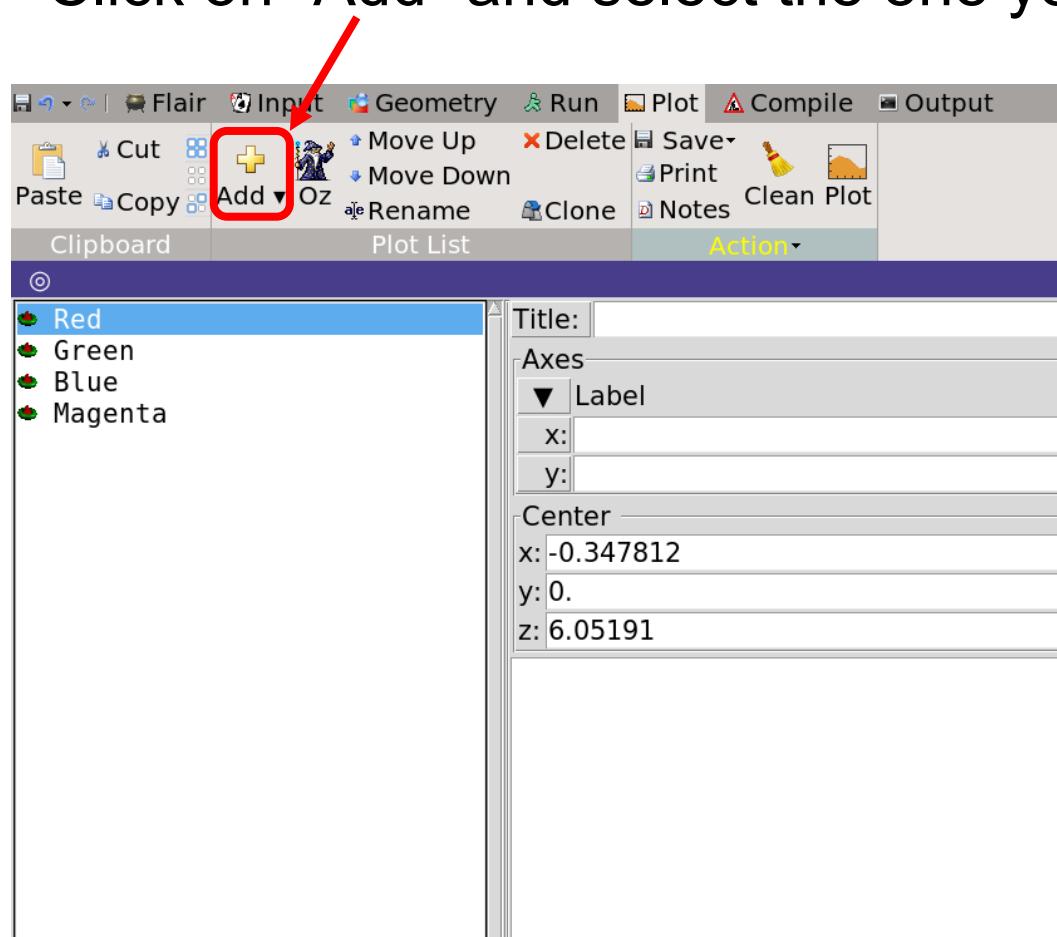
Plotting result in the Plot tab – 2

- It is possible to automatically generate the plots for all scorings in the input
- The program scans the input when “Oz” is invoked

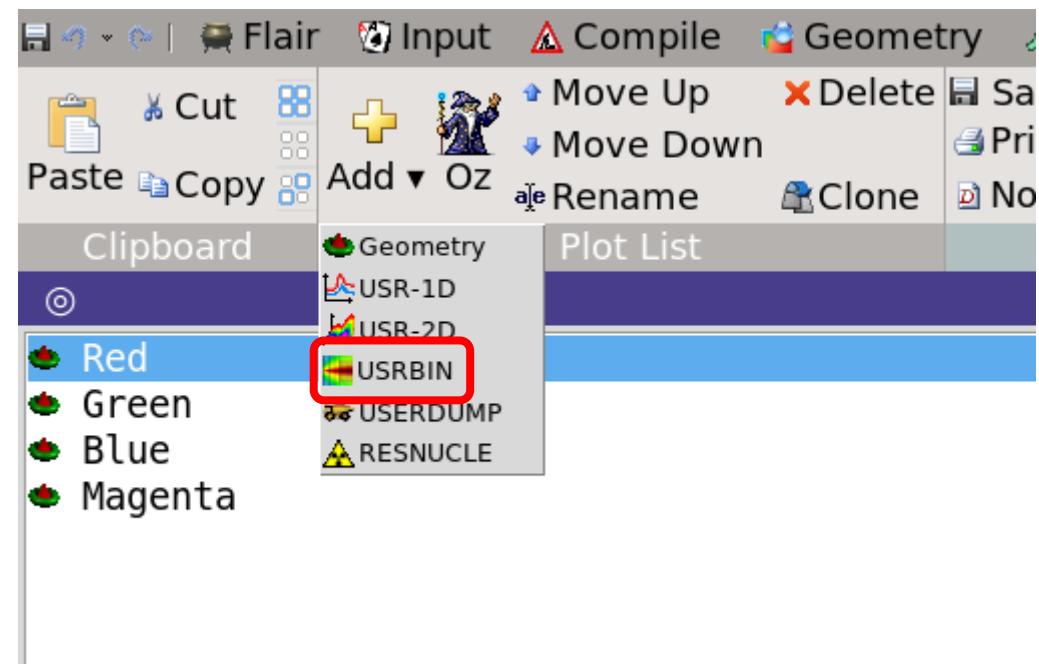


Plotting result in the Plot tab – 3

- It is possible to add plots by hand, one by one
- Click on “Add” and select the one you like from the pull down menu

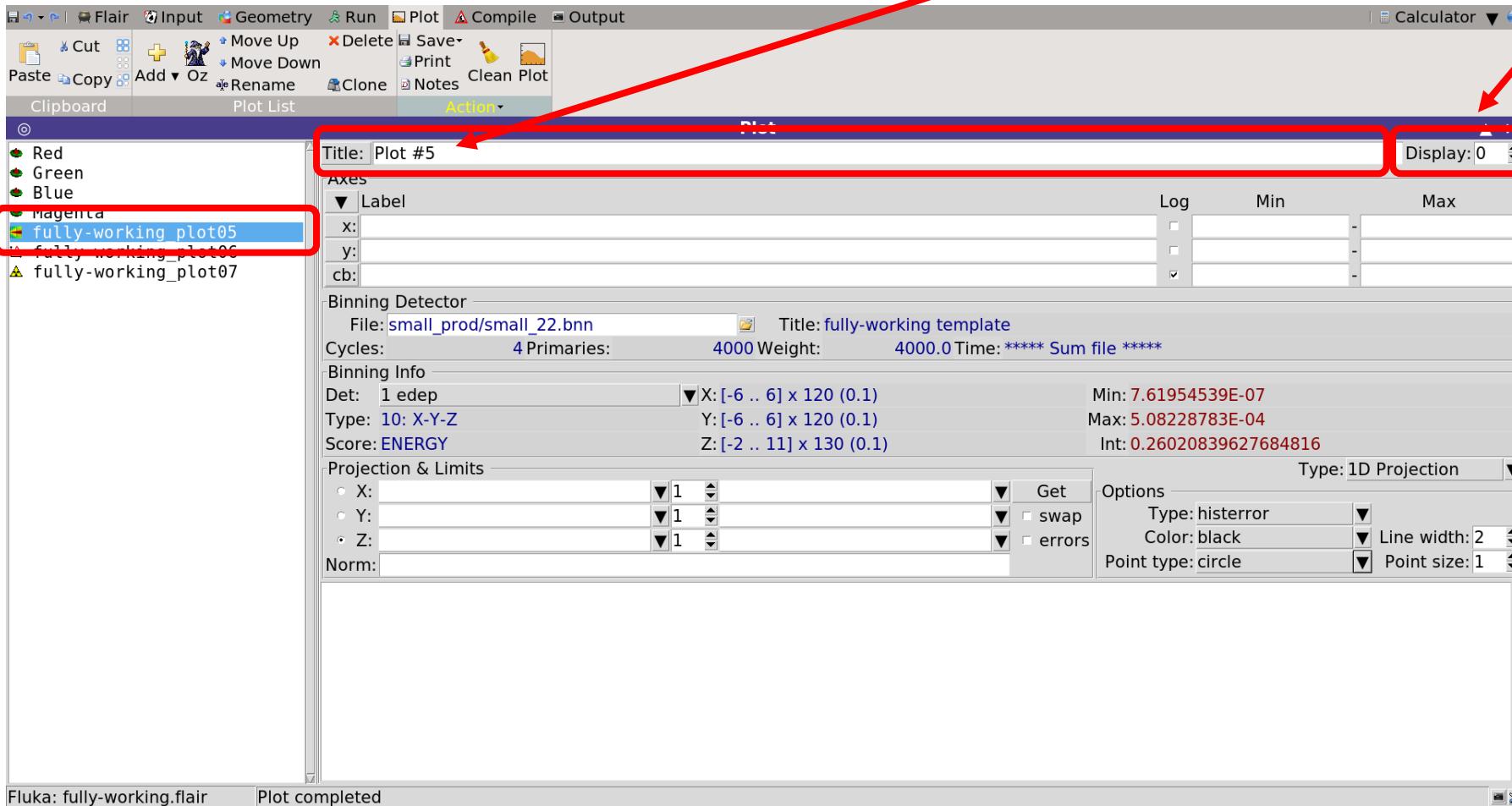


(here, we'll see only USRBIN)



Plotting result in the Plot tab – 4

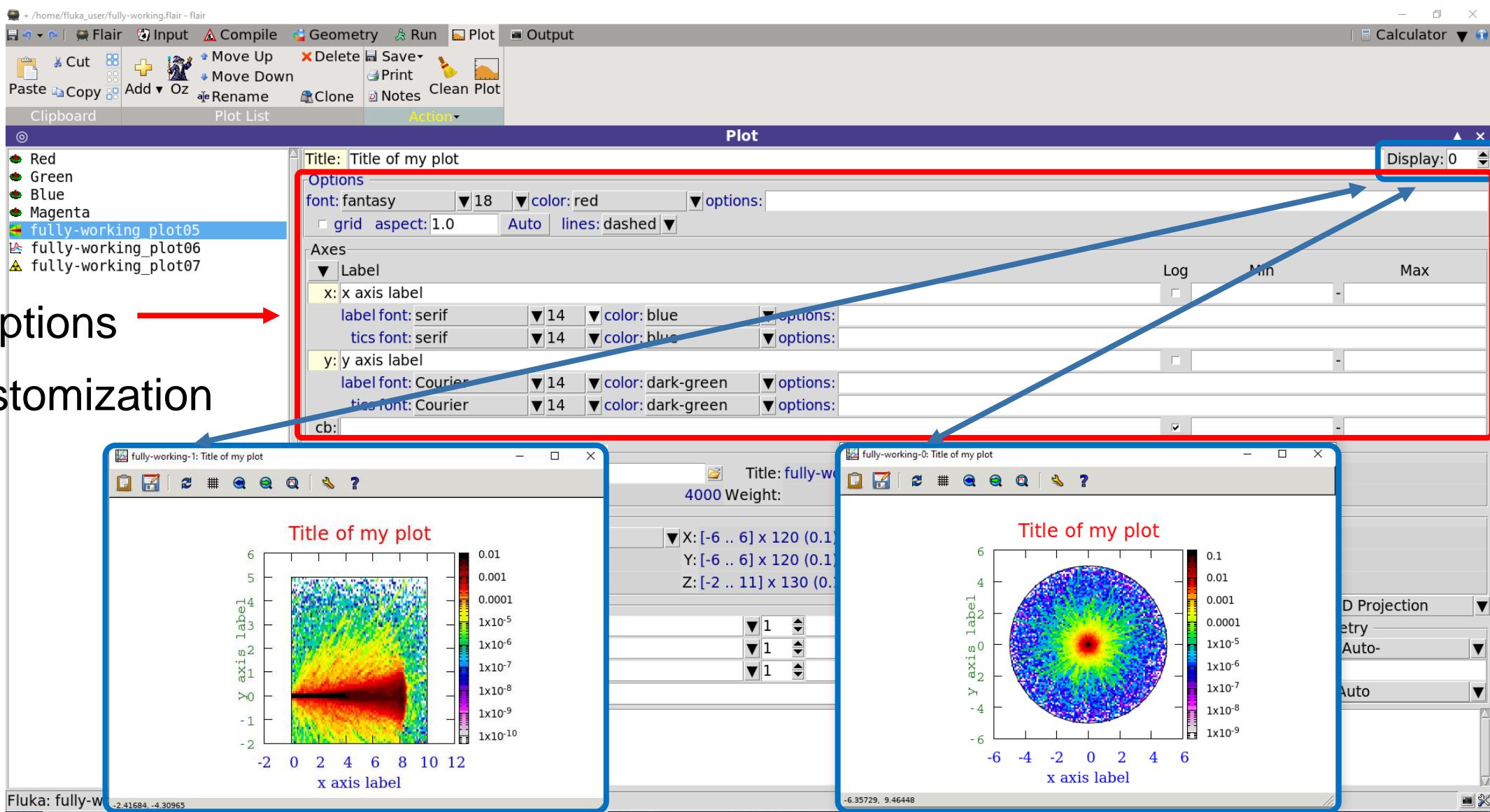
- Name of the file that will be saved



- Title of the plot

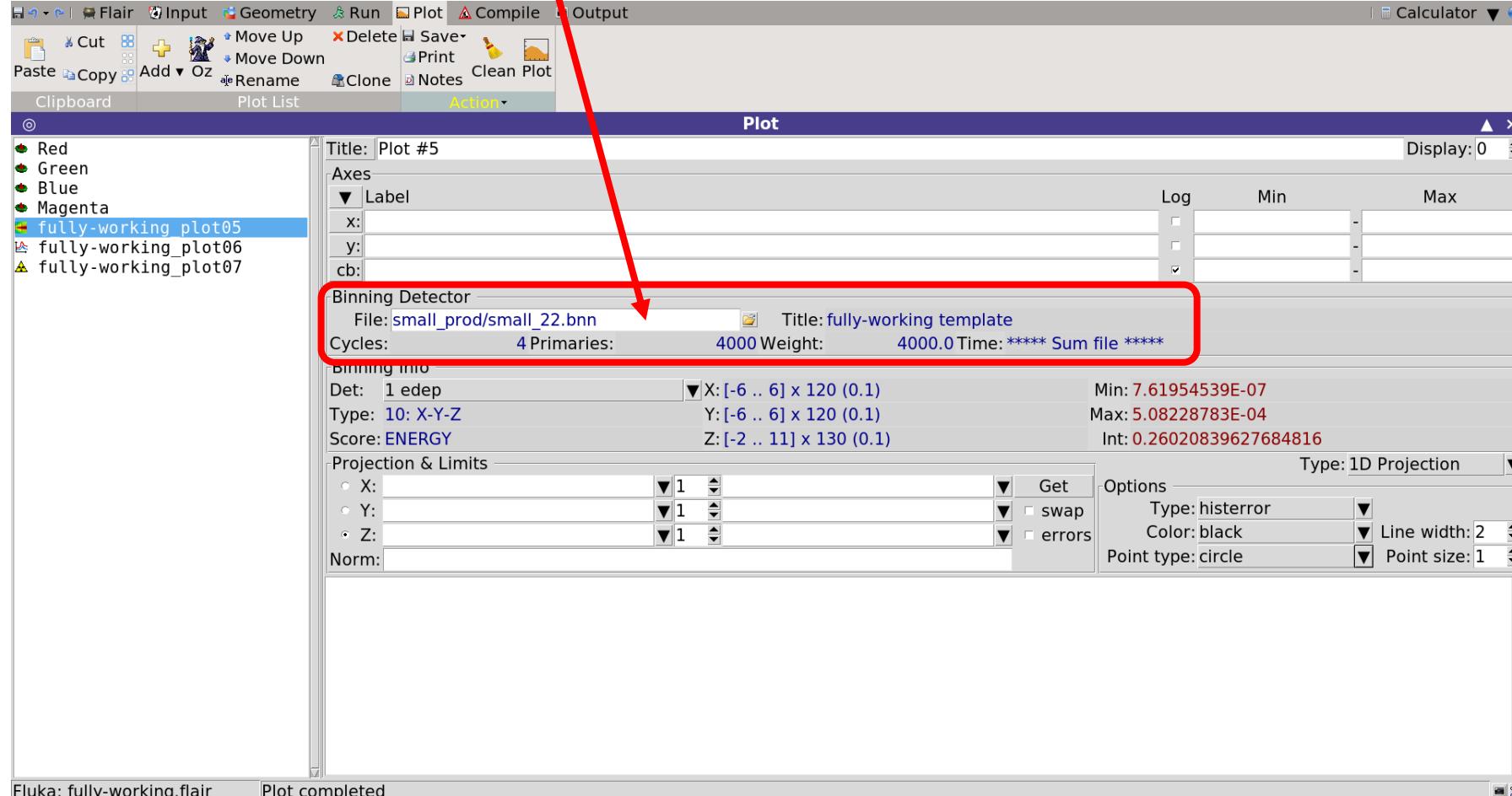
- Display ID

Plotting result in the Plot tab – 5



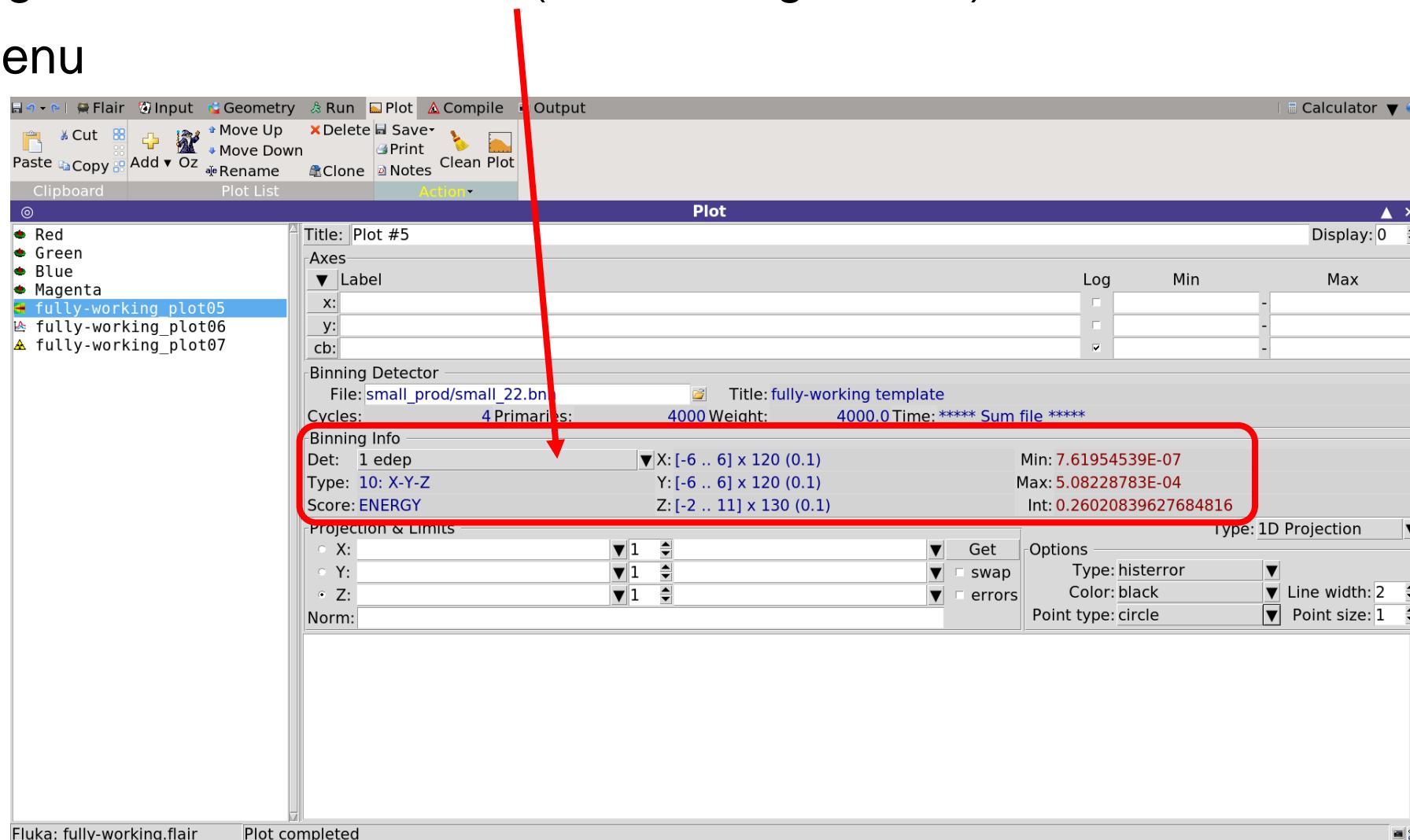
Plotting result in the Plot tab – 6

- Selection of the file containing the results of the simulations
- Opens standard pop-up for file selection
- Extra info available
 - #primaries
 - #cycles



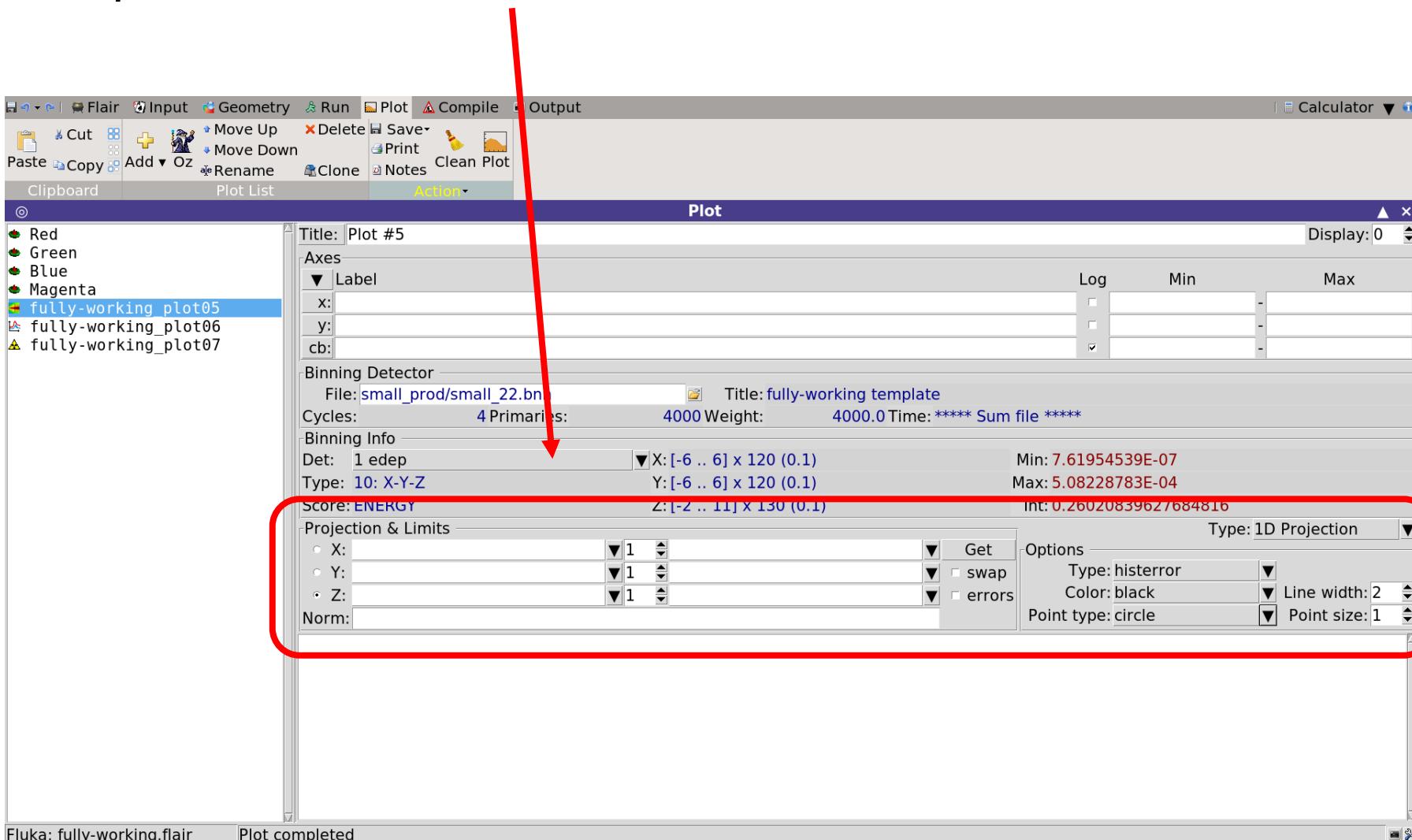
Plotting result in the Plot tab – 7

- Selection of the scoring within the chosen file (see scoring lecture)
- Standard pull-down menu
- Extra info available
 - Quantity scored
 - Type of mesh
 - Mesh details
 - Min & max values



Plotting result in the Plot tab – 8

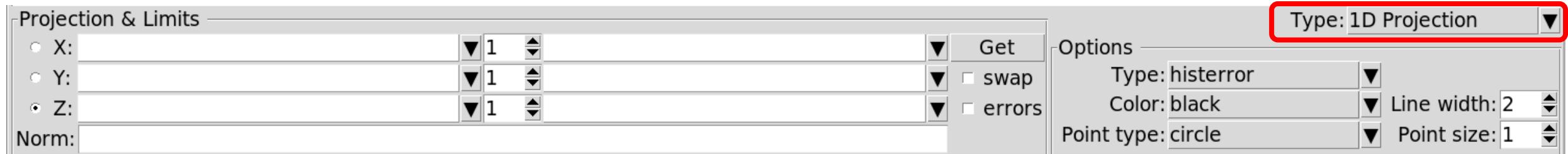
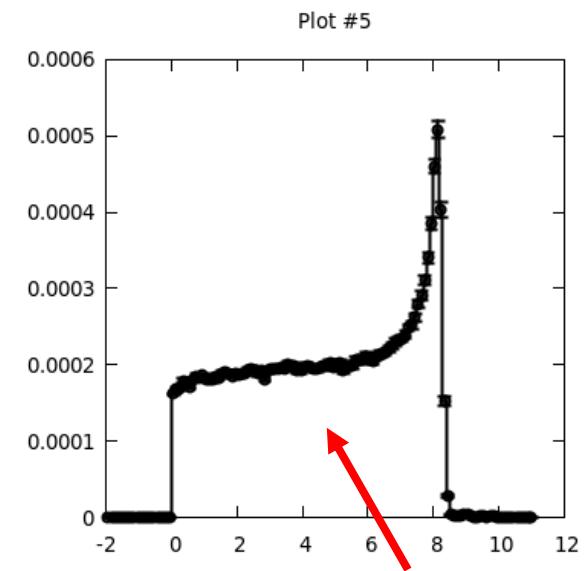
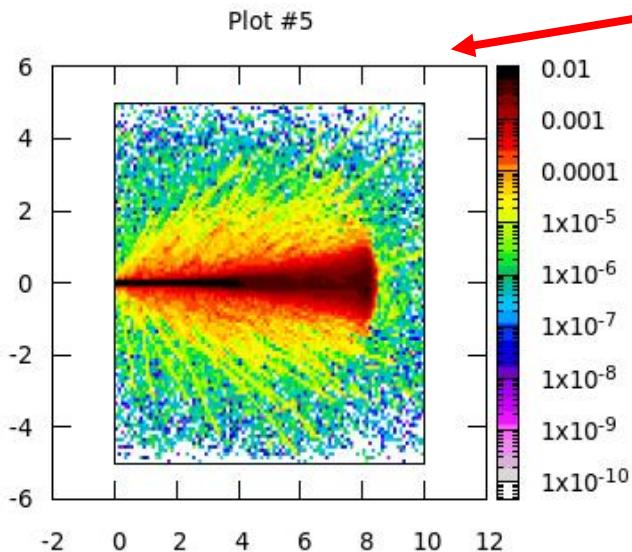
- Selection of plot type and options
 - 2D vs 1D projections
 - Plot extension
 - Uncertainty
 - Graphical options
 - Normalization



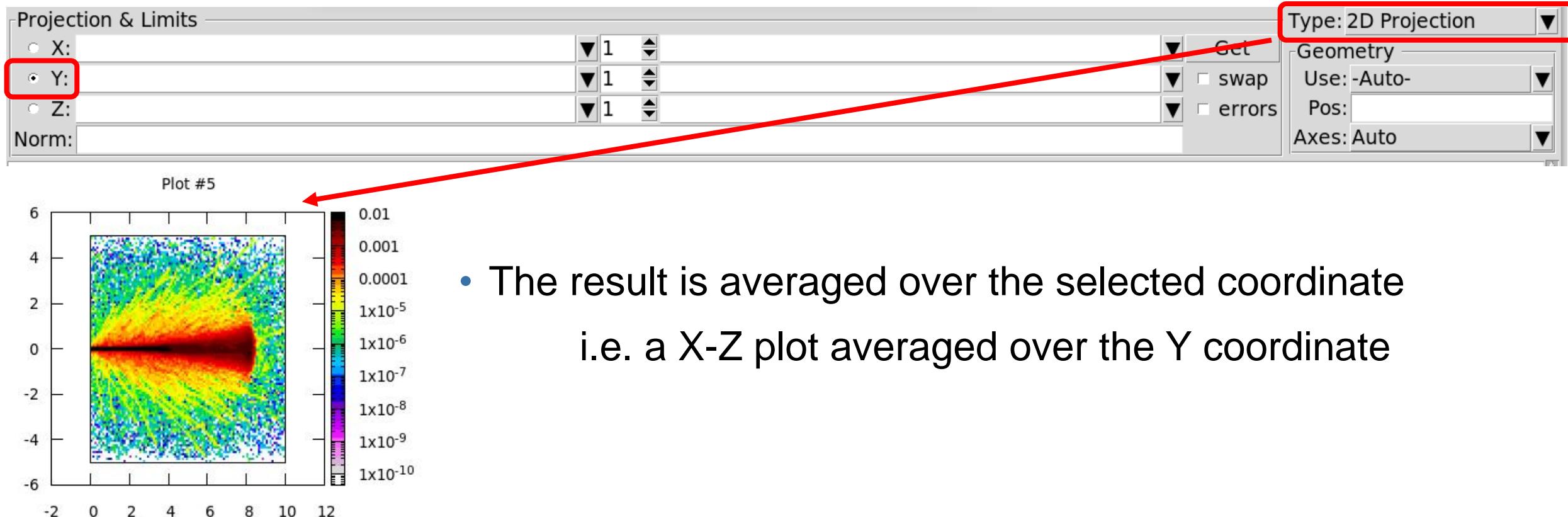
Plotting result in the Plot tab – 9



- 2D vs 1D projections

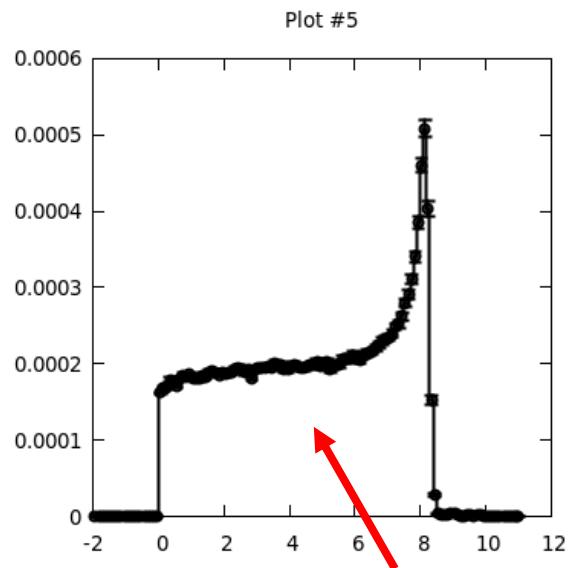


Plotting result in the Plot tab – 10



Plotting result in the Plot tab – 11

- The result is projected along the selected coordinate and averaged over the non-selected coordinates i.e. a projection along the Z axis



Projection & Limits

Z:

Type: 1D Projection

Options

Type: histerror

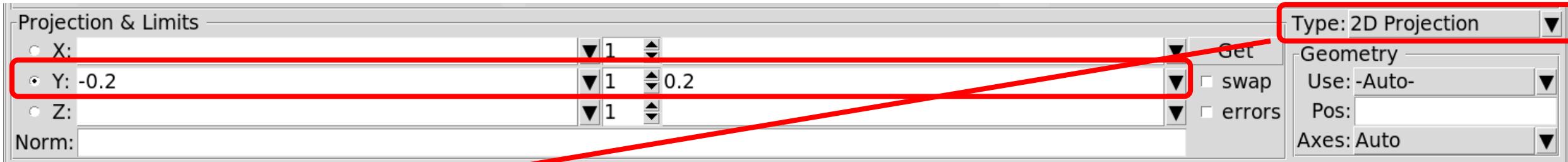
Color: black

Line width: 2

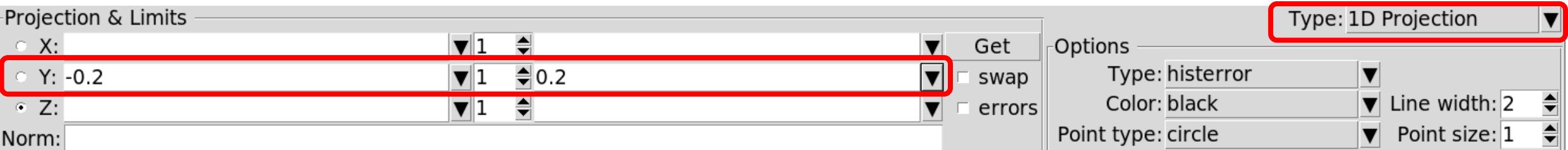
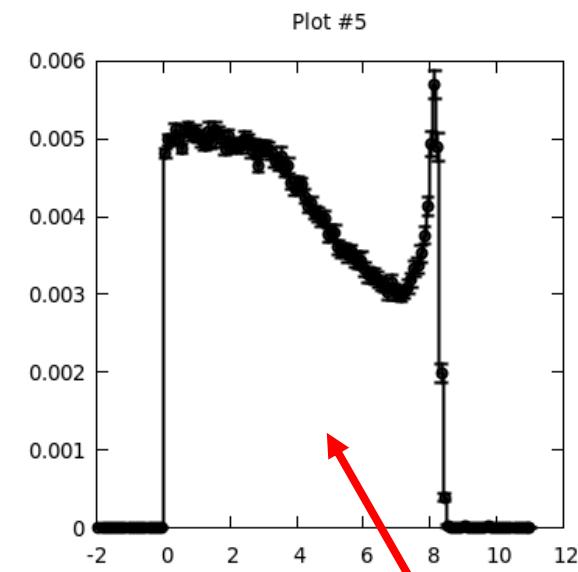
Point type: circle

Point size: 1

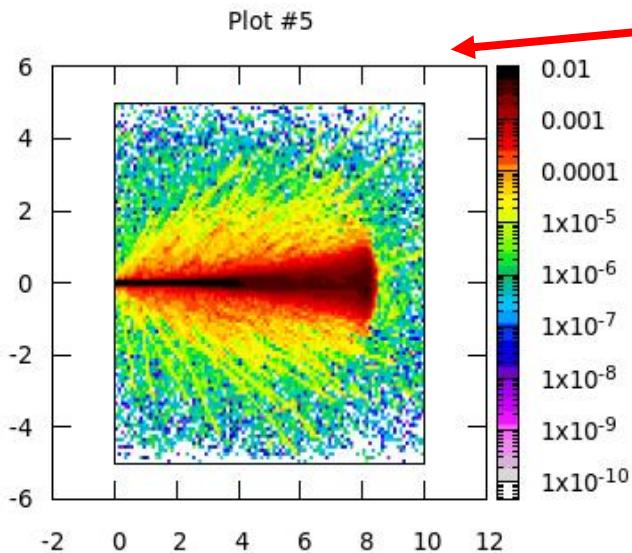
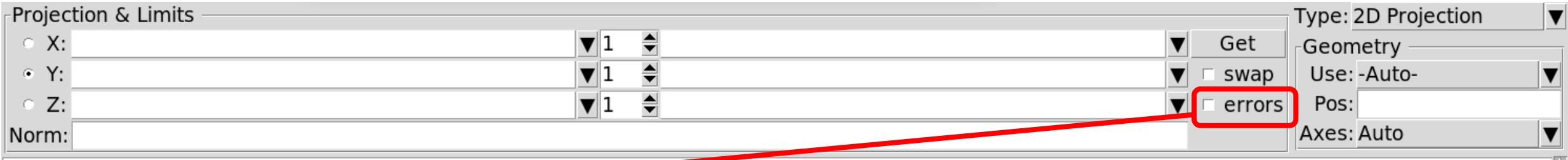
Plotting result in the Plot tab – 12



- Plot extension
- The results is averaged only within the specified limits

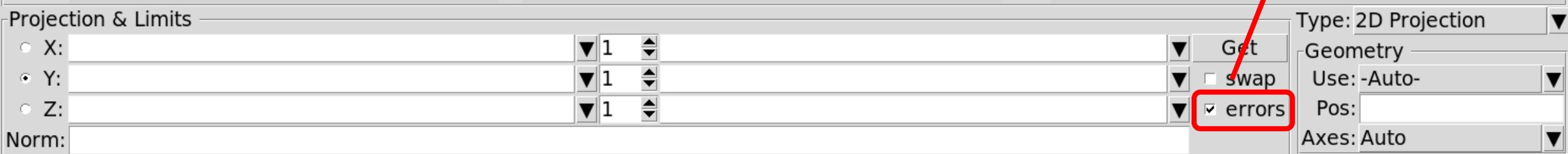
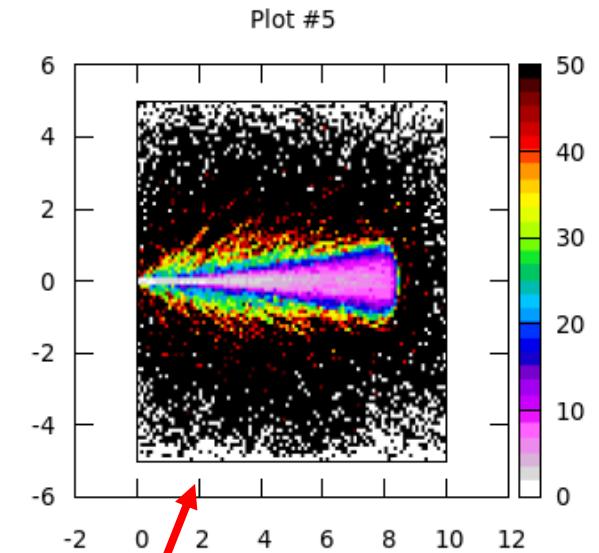


Plotting result in the Plot tab – 13



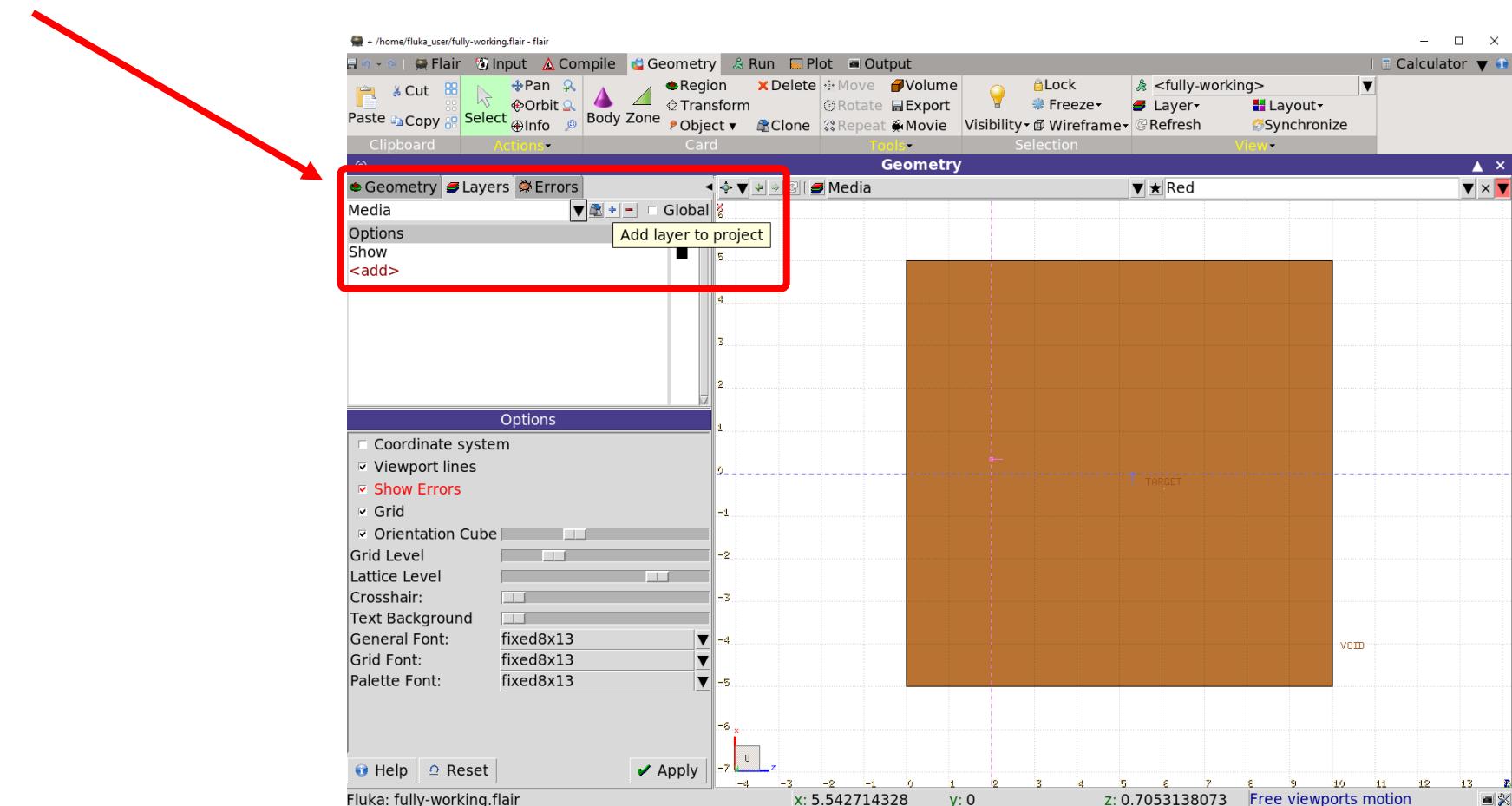
- Uncertainty

By ticking the “errors” box, it is possible to plot the statistical uncertainty in percent



Plotting result in the Geometry tab – 1

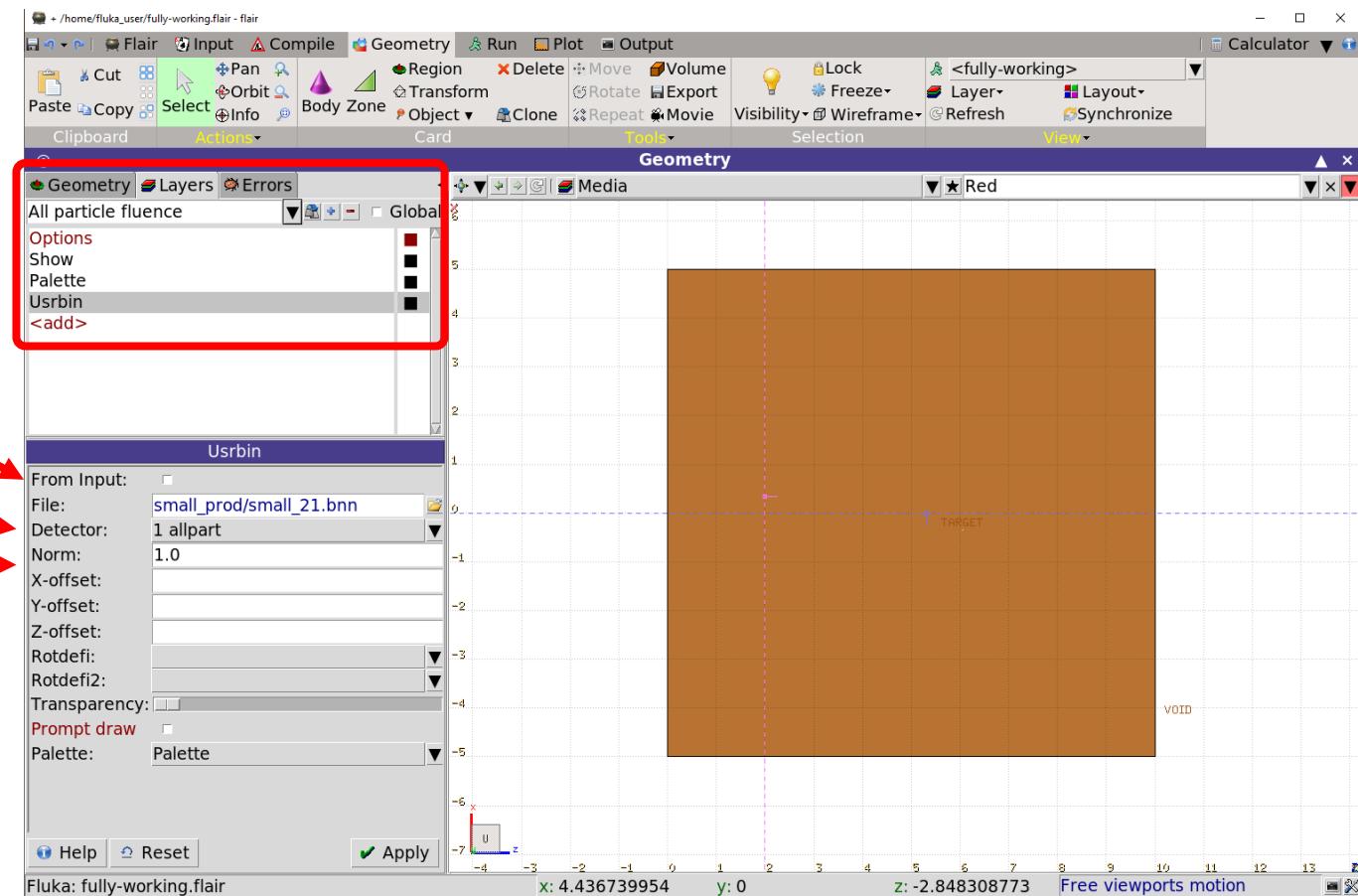
- It is possible to superimpose USRBIN results on the geometry
- A new layer has to be created or cloned from an existing one



Plotting result in the Geometry tab – 2

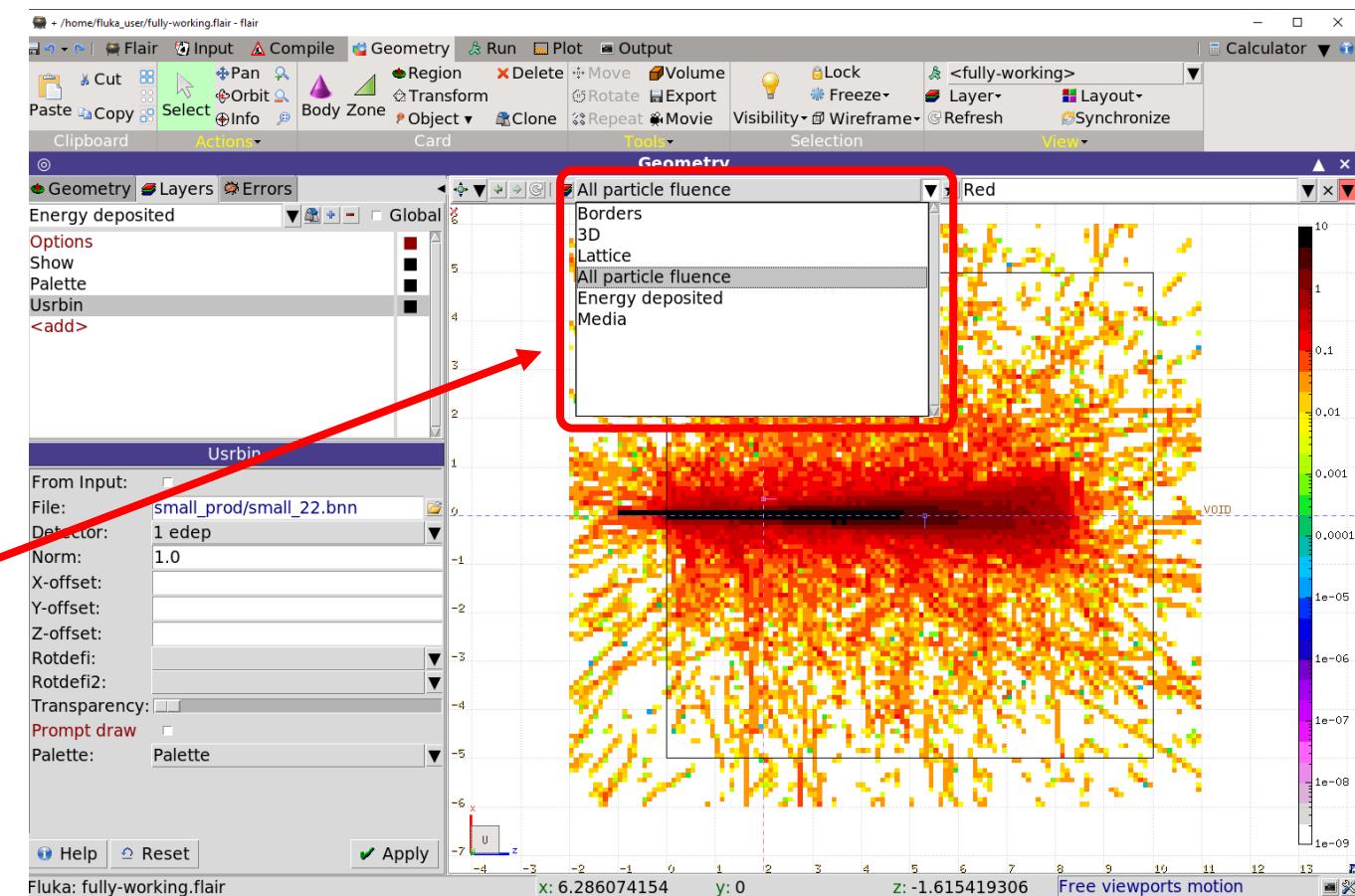
- It is possible to superimpose USRBIN results on the geometry
- A new layer has to be created or cloned from an existing one
- <add> “Usrbin”

- Select the file with the results
- Select the detector
- Play with normalization, palette and other options



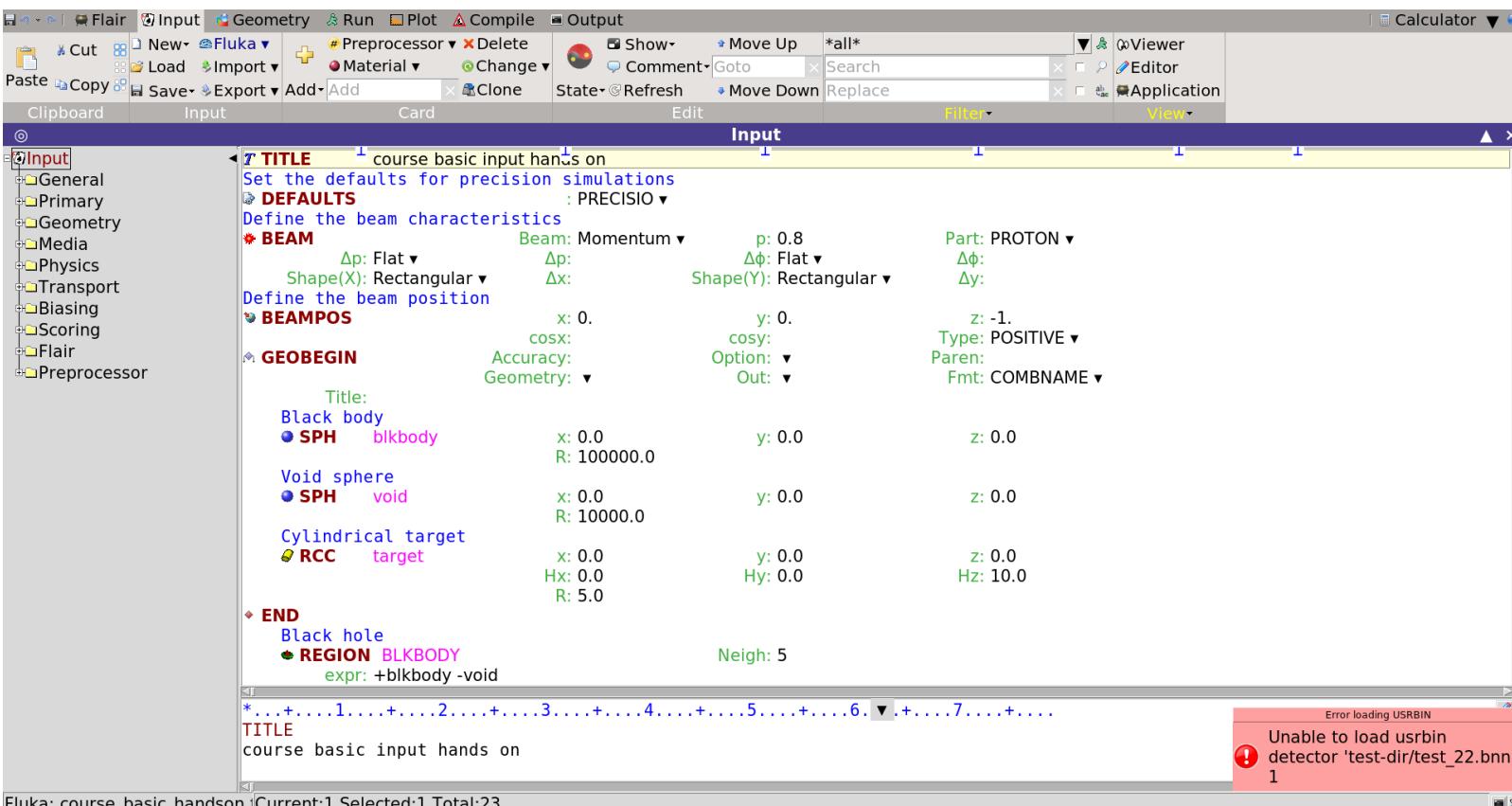
Plotting result in the Geometry tab – 3

- It is possible to superimpose USRBIN results on the geometry
- A new layer has to be created or cloned from an existing one
- <add> “Usrbin”
- Select the file with the results
- Select the detector
- Play with normalization, palette and other options
- Select the layer to visualize



Plotting result in the Geometry tab – 4

- WARNING: if the USRBIN used in a layer is missing, an error message is issued
- Not necessarily something to be worried about
- This will happen in the hands on that follows this lecture! Don't worry!



Summary of the work flow

- Create your **input** in the Input tab and Geometry tab (see future lectures)
- Verify your geometry in the Geometry tab
- **Run** the simulations and **merge** the output files in the Run tab
- **Plot** your results in the Plot tab and Geometry tab (see future lectures)

Time to do some practice!

- Let's start from the example file
and run a simulation step by step





FLUKA