



兰州大学
LANZHOU UNIVERSITY

Introduction to Flair

23rd FLUKA Beginner's Course
Lanzhou University
Lanzhou, China
June 2-7, 2024



V. Vlachoudis
"FLAIR: A Powerful But User
Friendly Graphical
Interface For FLUKA"
Proc. Int. Conf. on
Mathematics,
Computational Methods &
Reactor Physics (M&C
2009), Saratoga Springs,
New York, 2009

flea(r)

n [U,C] natural or instinctive ability (to do something well, to select or recognize what is best, more useful, etc.
[Oxford Advanced Dictionary of Current English]



FLUKA Advanced Interface [<http://www.fluka.org/flair>]

- All-in-one User friendly graphical Interface
- Minimum requirements on additional software
- Working in an intermediate level
Not hiding the inner functionality of FLUKA, exploiting FLUKA utilities

Front-End interface:

- Fully featured Input file Editor Writes a standard .inp file
 - Mini-dialogs for each card, allows easy editing
 - Error checking and validation of the input file during editing
- Geometry: interactive visualization, editing, and debugging
- Generation of the FLUKA Executable if needed Through *lfluka* and *ldpmqmd*
- Running and monitoring of the status of a/many run(s) Through *rfluka*



Back-End interface:

- Inspection of the fluka_nnn subdirectories
- Fluka output file **viewer** divided in sections
- **Post processing** (merging) the output data files
- **Plot** generation through an interface with gnuplot

Look at it!!

Utilities in \$FLUPRO/flutil

Other Goodies:

- Access to FLUKA **manual** as hyper text
- Checking for release updates of FLUKA and flair
- Nuclear wallet cards
- Library of materials

Look at it!!



Flair website <http://www.fluka.org/flair> (download and documentation)

Two packages:

- **flair** : input / run and data handling / plotting
- **flair-geoviewer**: geometry handling and visualization

the python3 version is now recommended!

Installation using RPM/DEB packages (strongly recommended!). As super-user:

- `rpm -ivh flair-X-XX.noarch.rpm flair-geoviewer-X-XX.X86-64rpm`
or
`dpkg -i flair_X.XX-X_all.deb flair-geoviewer_X.XX-XX_amd64.deb`
- The package will create all file associations, menu items, and keep track of updates.
- The package will install the program to: `/usr/local/flair`

Installation using tar files when necessary (non-compatible operating system, no admin): see the website for instructions



Programs Menu (Linux)

- Click the icon of Flair from the programs menu

Usually, Flair is in the Science/Physics sub-menu but this can change depending on the Linux distribution and window manager (look also in Applications, Education, Science, or Others sub-menus)

Window Manager (Linux, only via RPM or DEB installation)

- Flair makes an association of the following extensions:



.flair



.inp

Console

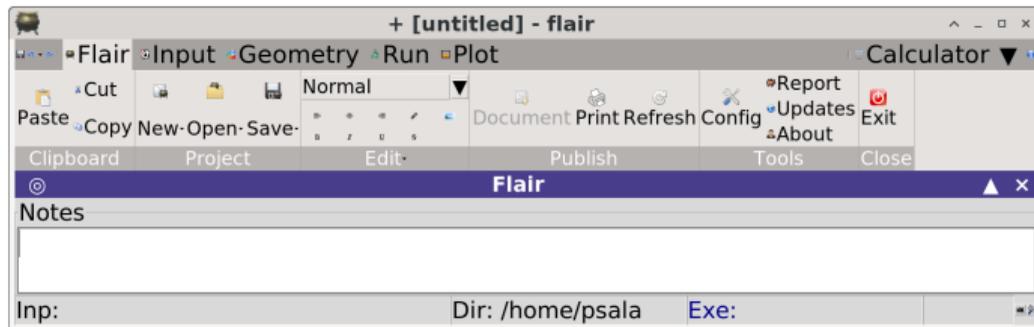
- Type the command **flair**

Check that your \$PATH includes the directory where flair is installed



At startup, flair

- Opens its window at the “flair” page
- Checks for the existence of a FLUKA installation (looking for the **FLUPRO** env. var.)
- Opens the “Check for Updates” dialog box (every 30 days interval)
- Reads the initialization files: `/usr/local/flair/flair.ini` and `$HOME/flair/flair.ini`
these contain standards and personal preferences about fonts, colors etc.
Can be changed from the Config menu, not described in this lecture



From here, one can open the various subpages (Input, Geometry, Run, Plot) and have fun

As soon as something is added, flair creates and fills a **project file** with extension **.flair**



Store in a single file all relevant information:

- Project notes
- Links to needed files: input file, user routines, output files ...
- Definition of Multiple runs from the same input file
- Procedures on how to run the code
- Information on how to post process and create plots of the results

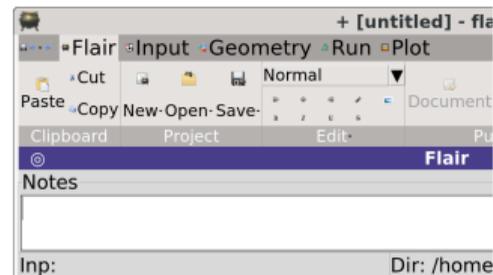
The format is plain ASCII with extension: .flair

To save it and give it a name:

- the **Save** button in the flair main page, or at exit

To read it back:

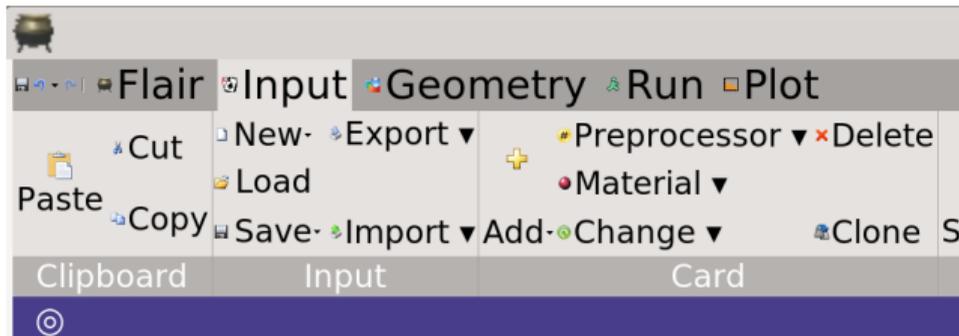
- the **Open** button, or
- `flair myproject.flair`



The Input page



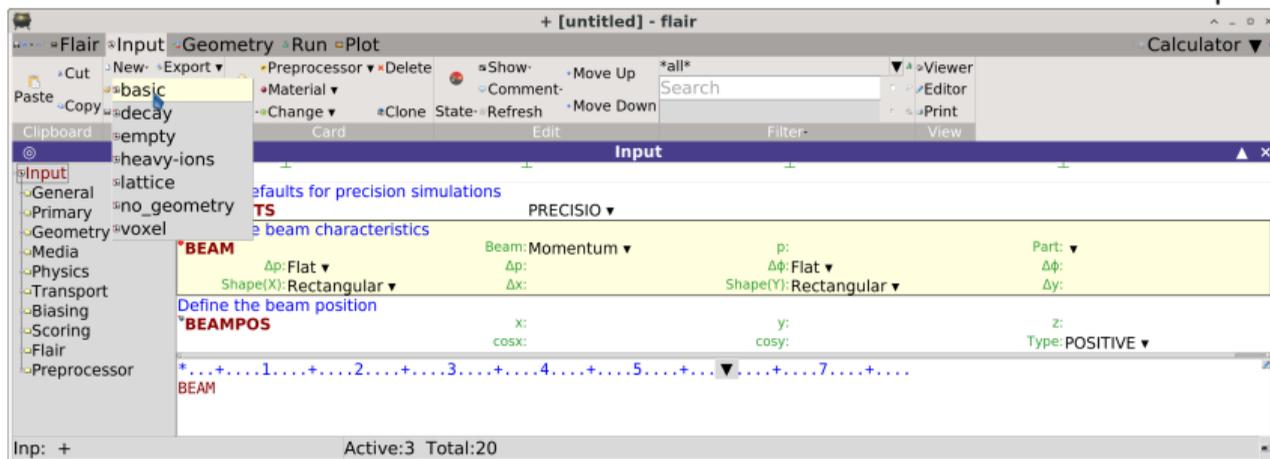
9



From the **input** page one can

- **Load** an existing input file
- Create **New** from template and modify it
- just start adding cards
- and of course **Save** the inp

Here we start a **New** from template **Basic**





Flair cards are extensions of the input cards, containing command, whats, and more:

- Comments
- Assemble continuation cards
- Fillable fields for whats with numbers
- Extra fields for multiple-meaning whats
- Drop-down menu for whats with choices



Flair cards are extensions of the input cards, containing command, whats, and more:

Always check the real input card

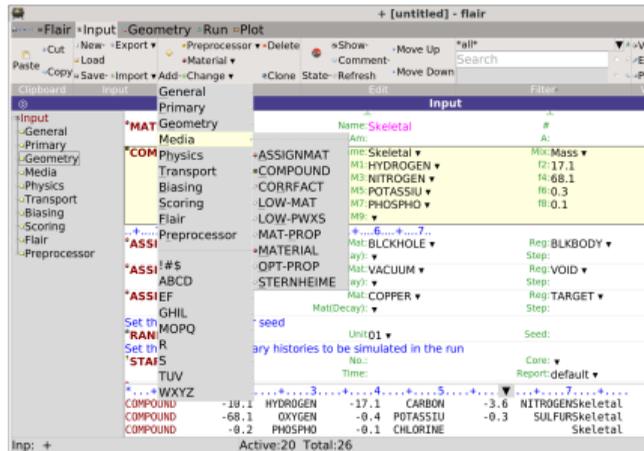
The card as written in the input file is shown at the bottom of the page

Check that it is what you wanted! [Using the manual](#)

You can open the manual from flair: the blue icon at the top-right or F1

MATERIAL	Name: Skeletal	#	ρ : 1.05
Z:	Am:	A:	dE/dx: ▼
COMPOUND	Name: Skeletal ▼	Mix: Mass ▼	Elements: 7..9 ▼
f1: 10.1	M1: HYDROGEN ▼	f2: 17.1	M2: CARBON ▼
f3: 3.6	M3: NITROGEN ▼	f4: 68.1	M4: OXYGEN ▼
f5: 0.4	M5: POTASSIU ▼	f6: 0.3	M6: SULFUR ▼
f7: 0.2	M7: PHOSPHO ▼	f8: 0.1	M8: CHLORINE ▼
f9:	M9: ▼		
*...+...1...+...2...+...3...+...4...+...5...+...▼...+...7...+...			
COMPOUND	-10.1 HYDROGEN	-17.1 CARBON	-3.6 NITROGENSkeletal
COMPOUND	-68.1 OXYGEN	-0.4 POTASSIU	-0.3 SULFURSkeletal
COMPOUND	-0.2 PHOSPHO	-0.1 CHLORINE	Skeletal

Another example: Compound card, all continuation cards together, a field to choose the type of composition (mass/volume/atoms)



Cards can be

- Edited (click on it)
- Added
 - they are grouped in categories
- Cloned
- Enabled / Disabled
- deleted

Saving

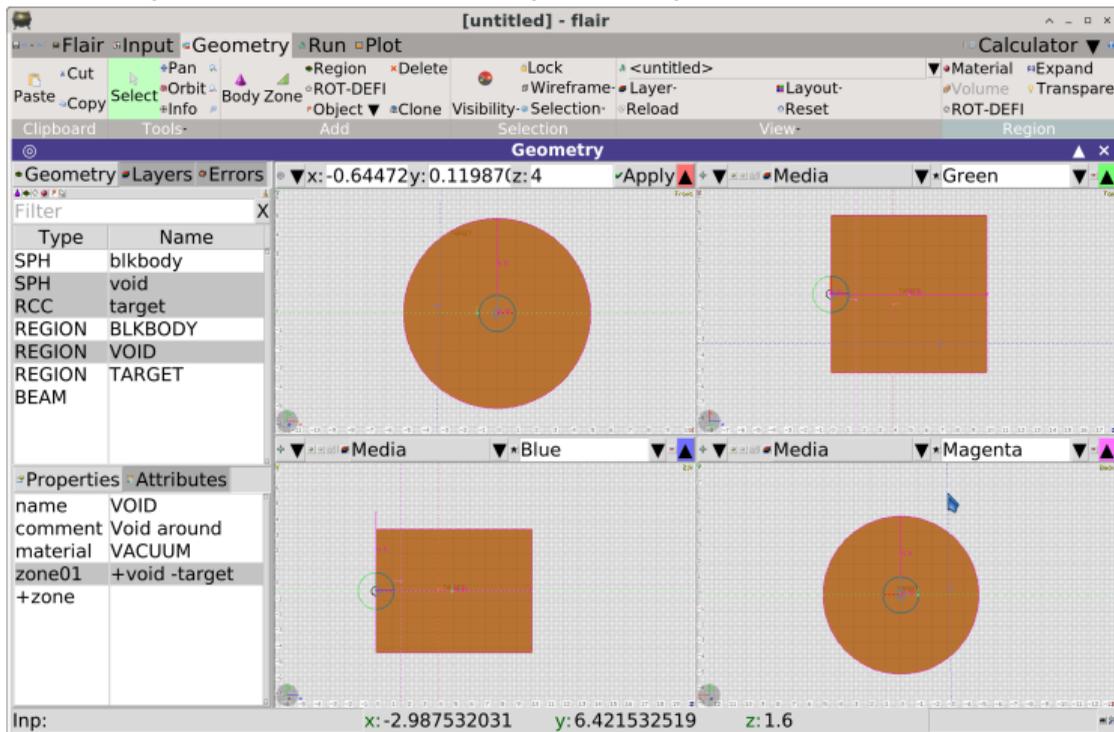
The input can be saved at any moment ("Save")

Or when Saving the project (from the flair main page)

Or when exiting flair (it asks)

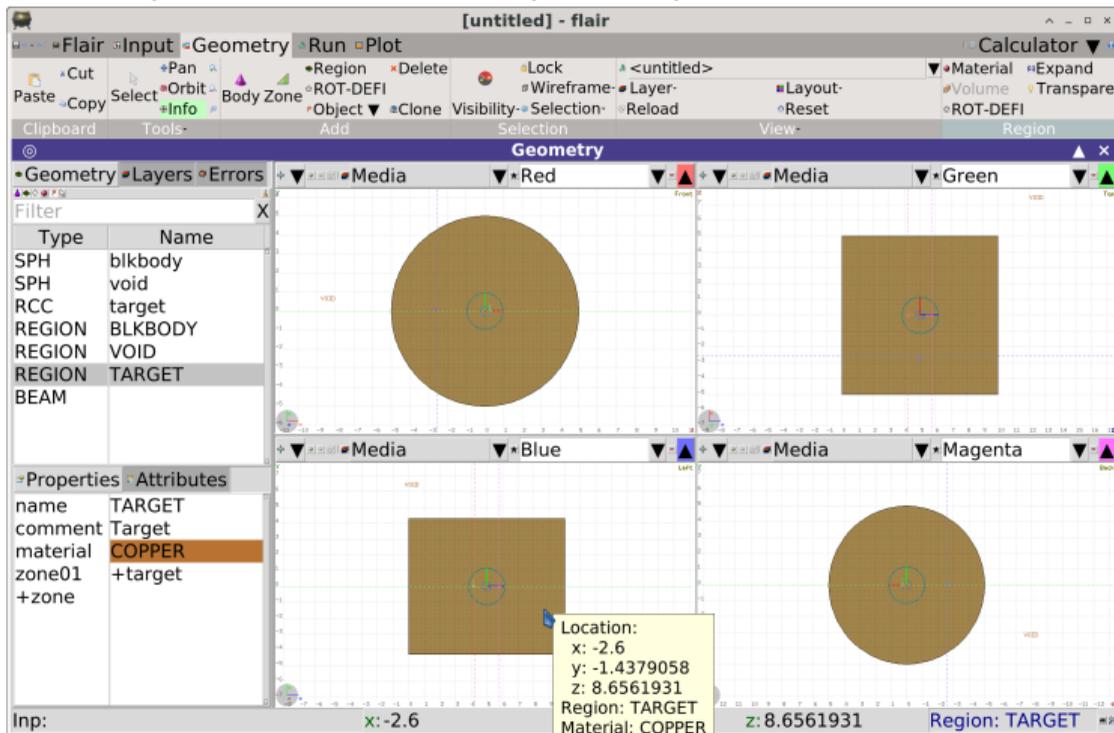
remember: the "master" for Fluka is the input file, not the flair file

Example, with the basic input template



- Four projections: Blue, Green, Red, Magenta
- can move "Pan", zoom in, zoom out..
- projection planes moved with hatched lines

Example, with the basic input template



The screenshot shows the Flair Geometry viewer interface. The main window displays four projections of a 3D model: Front, Top, Left, and Right. The model consists of a central cylindrical target (COPPER) surrounded by a larger cylindrical body (BLKBODY). The properties panel on the left shows the selected object's details:

Type	Name
SPH	blkbody
SPH	void
RCC	target
REGION	BLKBODY
REGION	VOID
REGION	TARGET
BEAM	

The Properties panel shows the following attributes for the selected object:

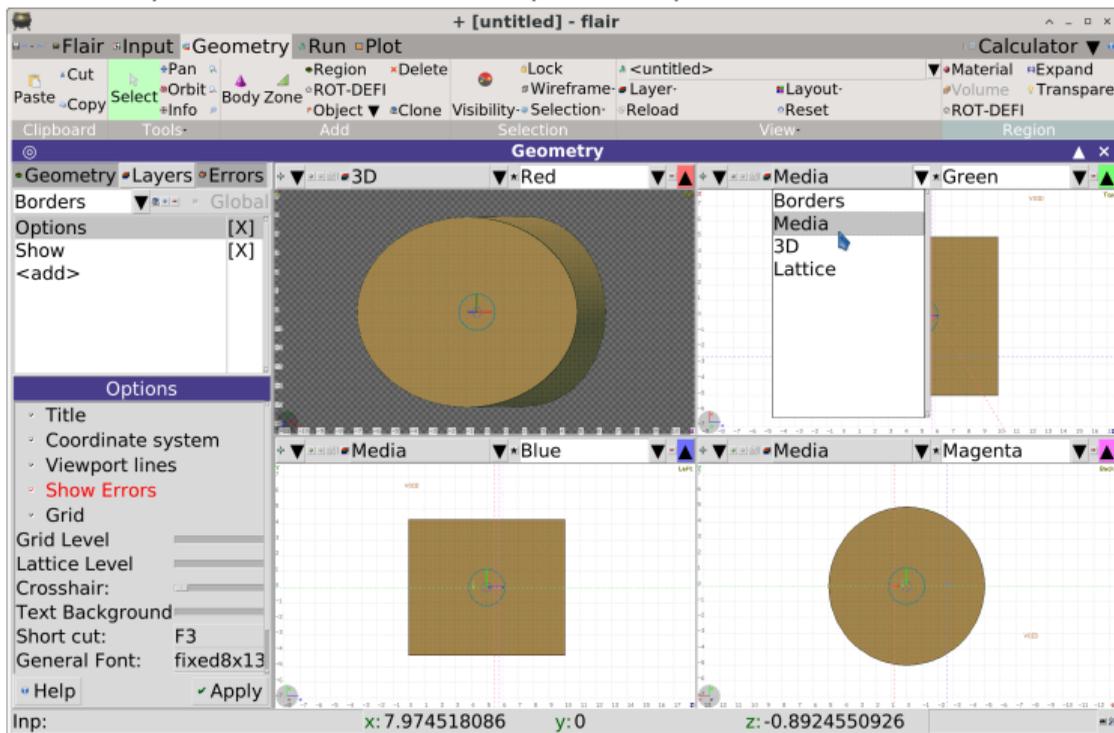
Attributes	Value
name	TARGET
comment	Target
material	COPPER
zone01	+target
+zone	

The status bar at the bottom shows the current location and region/material information:

Location: X: -2.6, y: -1.4379058, z: 8.6561931
Region: TARGET
Material: COPPER

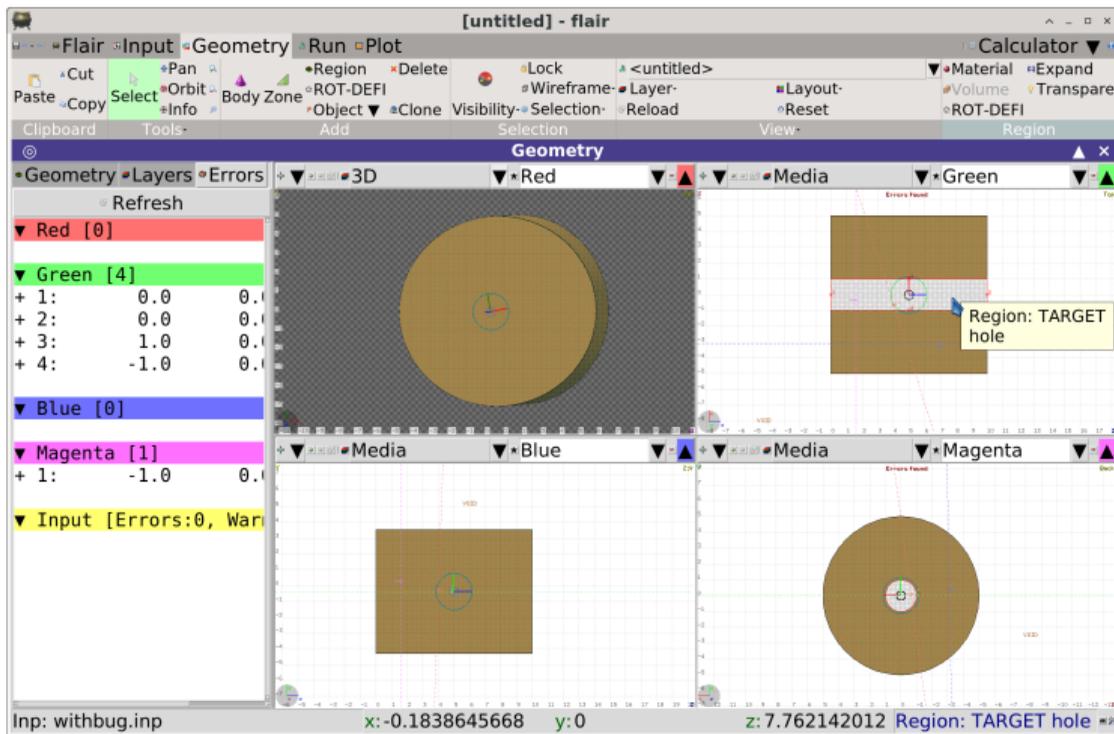
- Four projections: Blue, Green, Red, Magenta
- can move "Pan", zoom in, zoom out.
- Mouse click with "info" or "select" active: information on region/material/position

Example, with the basic input template



- Layers define what is plotted
- Default is Media (colors==materials)
- Powerful 3D Layer
- Layers can be customized

Example, with a geometry error added on purpose



- Defined a hole in the target
- Forgot to take away from the target
- The message **Error found** appears
- The zone is hatched
- Mouse click shows two regions in the same place
- Lateral panel gives error points

Run

Run settings



The screenshot shows the FLAIR software interface. The main window is titled '+ [untitled] - flair'. The 'Run' window is open, showing a table with columns 'Name' and 'Value'. The 'Progress' section shows 'Status: Finished OK', 'Input: example', 'Dir:', 'Started:', 'ETA:', 'Time/prim:', 'Elapsed:', 'Cycle:', 'Run:', 'Cycles:', and 'Primaries:'. The status bar at the bottom indicates 'Inp: example.inp' and 'Running 0 out of 1'.

- Basic use: run fluka
- Choose number of cycles and previous cycle
- Advanced use will be explained later
- → start
- launches the **\$FLUPRO/flutil/rfluka** script

Run

Run settings II



Here with example.inp from the standard distribution

The screenshot shows the 'Run' dialog box in the Flair software. The dialog has a title bar '+ [untitled] - flair' and a menu bar with 'Clipboard', 'View', 'Input', 'Job', and 'Action'. The main area contains a table with columns 'Name' and 'Value'. Below the table is a 'Progress' section with the following information:

Name	Value
Primaries	0
Rnd	0
Time	0
Exe	

Progress
Status: Running Input: example Dir: fluka_199:
Started: 2024.04.24 10:27 ETA: 2024.04.24 10:28 Time/prim: 0.102406
Elapsed: 9.42136 s Cycle: 819.249 ms Run: 36.8511 s
Cycles: Current: 2 [5] Completed: 20%
Primaries: Current: 92001 [100000] Completed: 92%

Running 0 out of 1

- launches the `$FLUPRO/flutil/rfluka` script
- and keeps track of the status

Run Output files



Here with example.inp from the standard distribution

Run	Spawn	Cycles	File	Type	Size	Date
<untitled>		001	example001_fort.50		24002	2024.04.24 1
		002	example001_fort.51		8238	2024.04.24 1
		003	example001.out	Output	21561	2024.04.24 1
		004	example001_fort.47		4324	2024.04.24 1
		005	ranexample001	-file-	1651	2024.04.24 1
		006	example001_fort.49		282	2024.04.24 1
		compile	example001_fort.48		282	2024.04.24 1
		data	example001.err	Error	29650	2024.04.24 1
		input	example001.log	Log	12984	2024.04.24 1
		plot	example001_fort.77		51957	2024.04.24 1
		temporary	example.out	Output	5158	2024.04.24 1

- Warning:: **flair** output is not **fluka** output
- example.out contains what would be on the screen when you run from the command line
- example001.out is the real fluka output

Run Output files



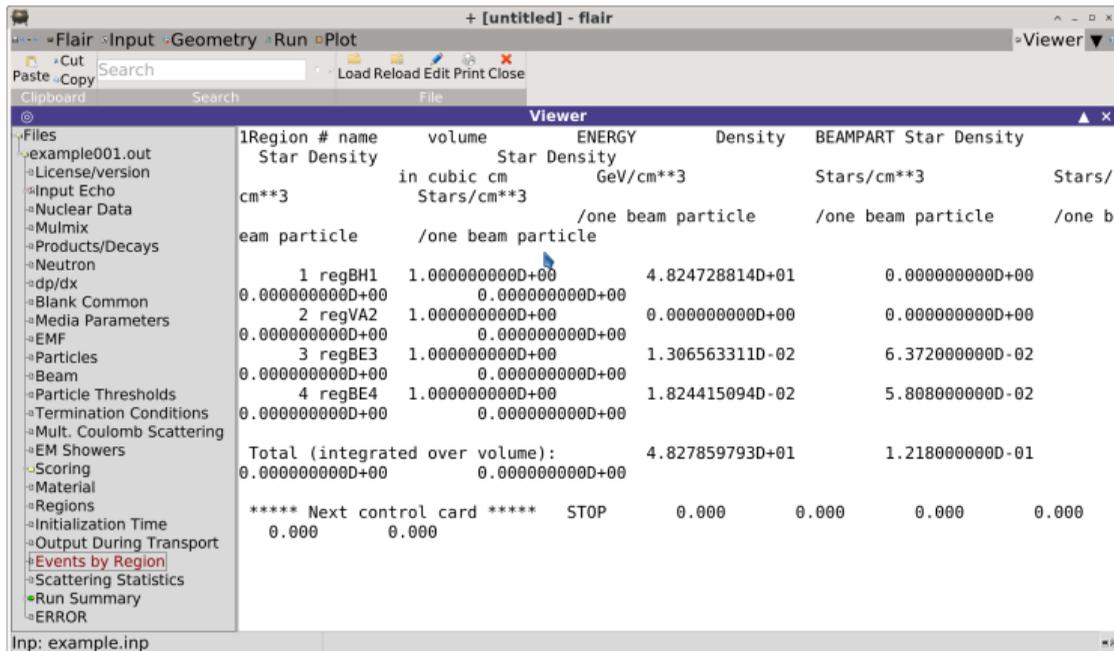
Here with example.inp from the standard distribution

The screenshot shows the Flair software interface. The 'Run' window is open, displaying a table of files generated during a simulation. The table has columns for 'Run', 'Spawn', 'Cycles', 'File', 'Type', 'Size', and 'Date'. The 'Run' column contains values from 001 to 007, and 'spawn' is set to 'temporary'. The 'File' column lists various files including 'example001_fort.50' through 'example001_fort.77', 'example001.out', 'example001.err', and 'example001.log'. The 'Type' column indicates the file type, such as 'Output', 'Error', and 'Log'. The 'Size' and 'Date' columns show the file size and the date of creation (2024.04.24).

Run	Spawn	Cycles	File	Type	Size	Date
<untitled>		001	example001_fort.50		24002	2024.04.24 1
		002	example001_fort.51		8238	2024.04.24 1
		003	example001.out	Output	21561	2024.04.24 1
		004	example001_fort.47		4324	2024.04.24 1
		005	ranexample001	-file-	1651	2024.04.24 1
		006	example001_fort.49		282	2024.04.24 1
		compile	example001_fort.48		282	2024.04.24 1
		data	example001.err	Error	29650	2024.04.24 1
		input	example001.log	Log	12984	2024.04.24 1
		plot	example001_fort.77		51957	2024.04.24 1
		temporary	example.out	Output	5158	2024.04.24 1

- double click on example001.out

Here with example.inp from the standard distribution

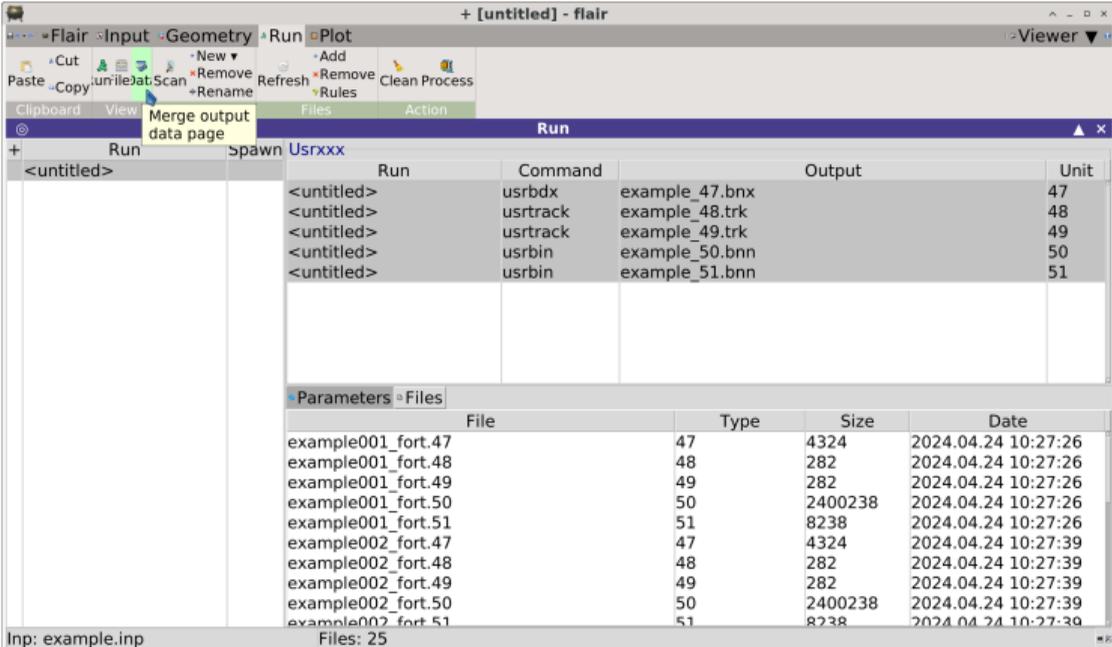


The screenshot shows the Flair software interface with the 'Viewer' window open. The output is a table of simulation results for 'example.inp'. The table has columns for Region #, name, volume, ENERGY, Density, BEAMPART, and Star Density. The output is divided into sections for different regions (regBH1, regVA2, regBE3, regBE4) and a total integrated over volume. The 'Total' section shows a density of 4.827859793D+01 and a star density of 1.218000000D-01. The 'Next control card' section shows 'STOP' with various zero values.

Region #	name	volume	ENERGY	Density	BEAMPART	Star Density
1	regBH1	1.000000000D+00	4.824728814D+01	0.000000000D+00		
2	regVA2	1.000000000D+00	0.000000000D+00	0.000000000D+00		
3	regBE3	1.000000000D+00	1.306563311D-02	0.000000000D+00		
4	regBE4	1.000000000D+00	1.824415094D-02	0.000000000D+00		
Total (integrated over volume):			4.827859793D+01			1.218000000D-01
**** Next control card ****			STOP	0.000	0.000	0.000

- The fluka output divided in sections
- **Always have a look!!!**
- especially when setting up a simulation
- or in case of crashes

Here with example.inp from the standard distribution



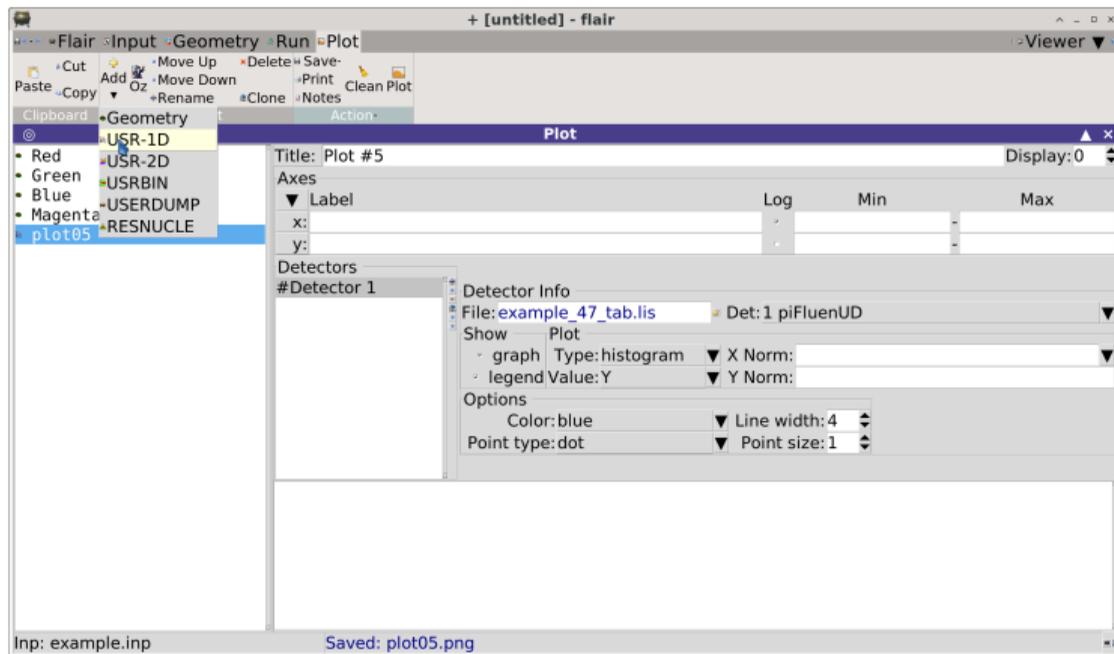
The screenshot shows the Flair software interface with the 'Run' menu open. The 'Merge output data page' option is highlighted. Below the menu, a table displays the output of the 'Run' command, listing files, commands, and units. At the bottom, a 'Parameters' table lists files, types, sizes, and dates.

Run	Command	Output	Unit
<untitled>	usrbdx	example_47.bnx	47
<untitled>	usrtrack	example_48.trk	48
<untitled>	usrtrack	example_49.trk	49
<untitled>	usrbin	example_50.bnn	50
<untitled>	usrbin	example_51.bnn	51

File	Type	Size	Date
example001_fort.47	47	4324	2024.04.24 10:27:26
example001_fort.48	48	282	2024.04.24 10:27:26
example001_fort.49	49	282	2024.04.24 10:27:26
example001_fort.50	50	2400238	2024.04.24 10:27:26
example001_fort.51	51	8238	2024.04.24 10:27:26
example002_fort.47	47	4324	2024.04.24 10:27:39
example002_fort.48	48	282	2024.04.24 10:27:39
example002_fort.49	49	282	2024.04.24 10:27:39
example002_fort.50	50	2400238	2024.04.24 10:27:39
example002_fort.51	51	8238	2024.04.24 10:27:39

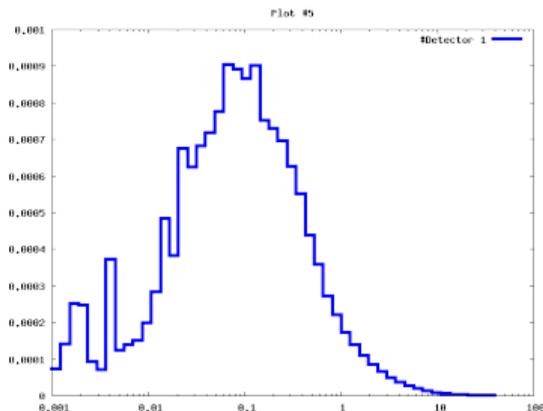
- “Process” merge data from different cycles
- launches the utility programs in **\$FLUPRO/flutil/**
- Wait for the scoring lecture for details

Here with example.inp from the standard distribution



- Plot results with gnuplot
- Using files produced by merging utilities
- Wait for the scoring lecture for details

Here with example.inp from the standard distribution



- Plot results with gnuplot
- Using files produced by merging utilities
- Wait for the scoring lecture for details



- **flair** is a powerful graphical interface for FLUKA
- here only a glimpse was given
- more all along this course
- best learning method is practising
- important: The master file, the one used by fluka, is always the **.inp** file
- Flair exploits the **standard** fluka utilities (scripts and auxiliary codes)
- Thus, in case of problems, crashes, etc, look at the fluka **.out** and **.err** files, sometimes also the output from auxiliary programs can be instructive (see scoring lecture)