CURSO INTRODUTÓRIO FLUKA

23 DE JANEIRO A 8 DE MARÇO DE 2023

Tópicos essenciais

FLUKA

FLUKA

FLUKA

FLUKA

FLUKA

AULA 01 Introdução ao FLUKA

Iniciaremos em breve

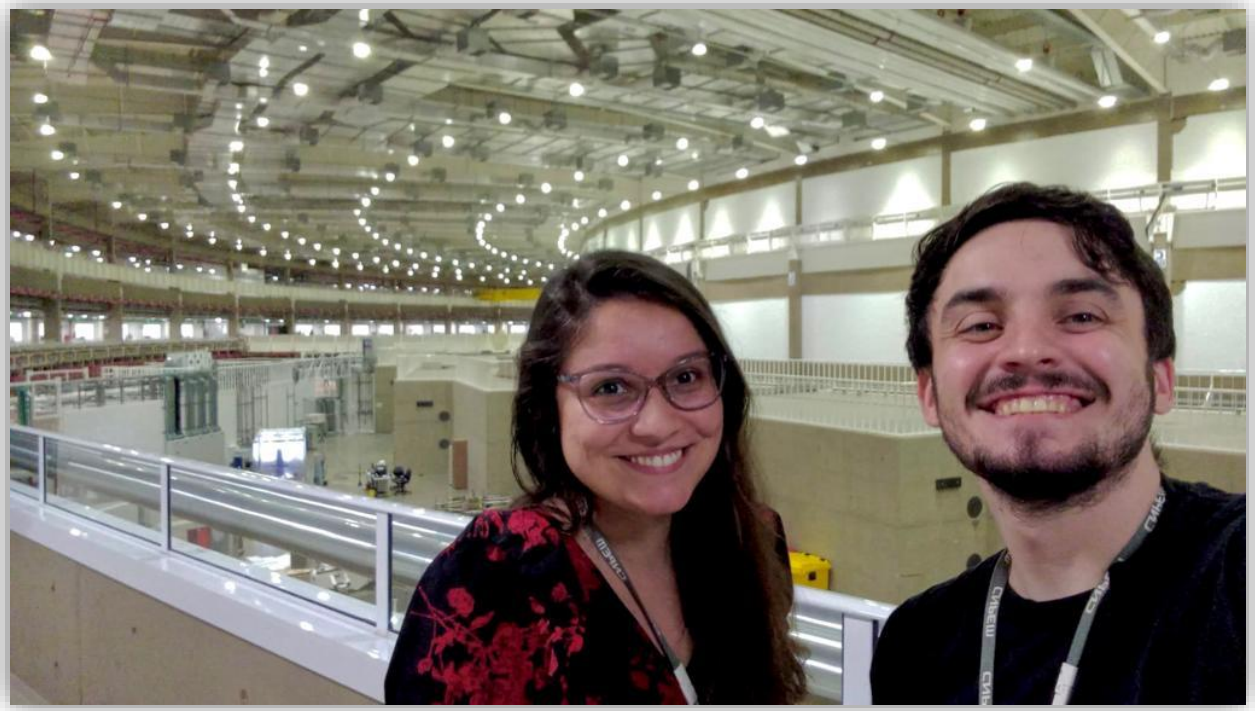
Código Monte Carlo de interação e transporte de partículas

23 FLUKA 2023

08

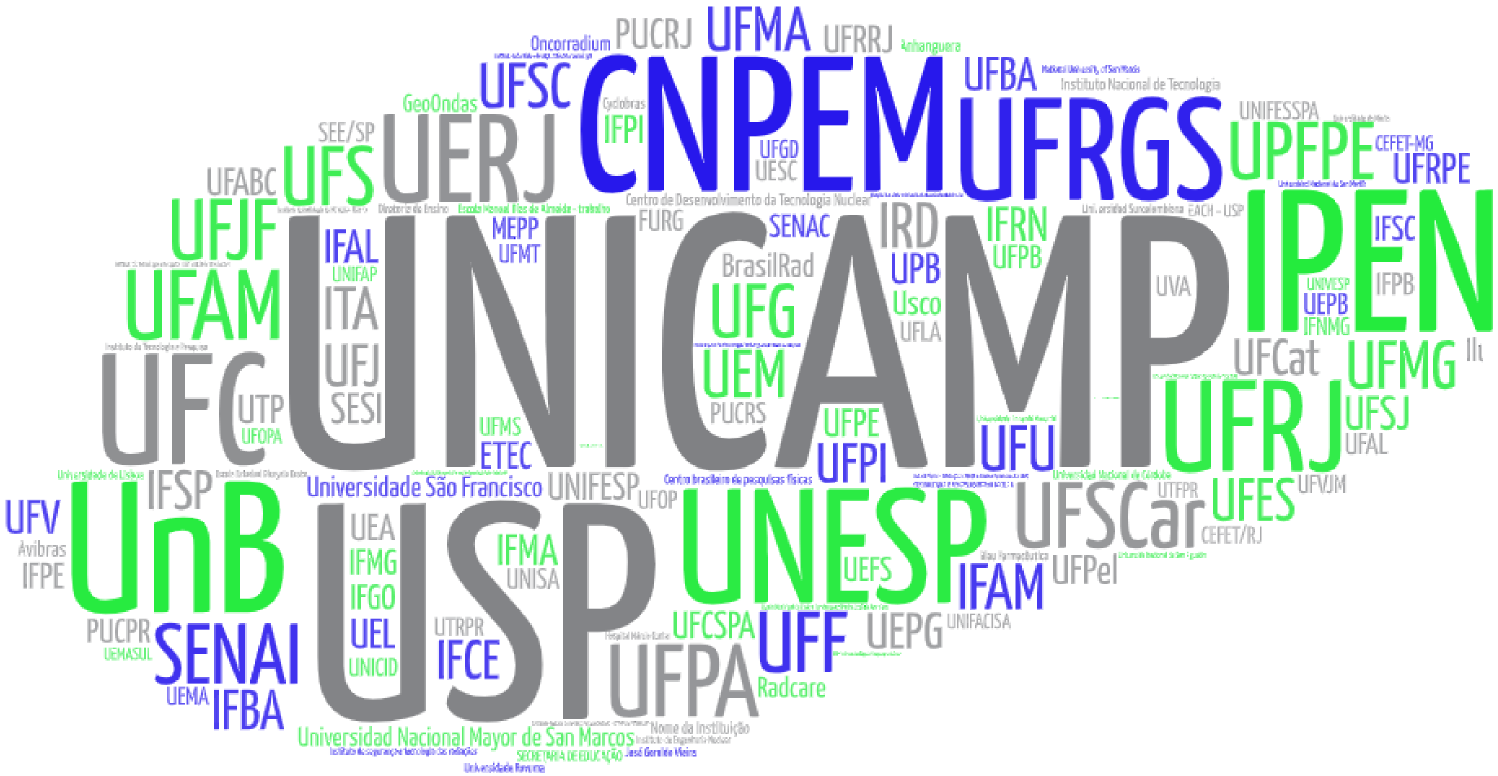
CNPEM
ORGANIZAÇÃO SOCIAL DO MCTI

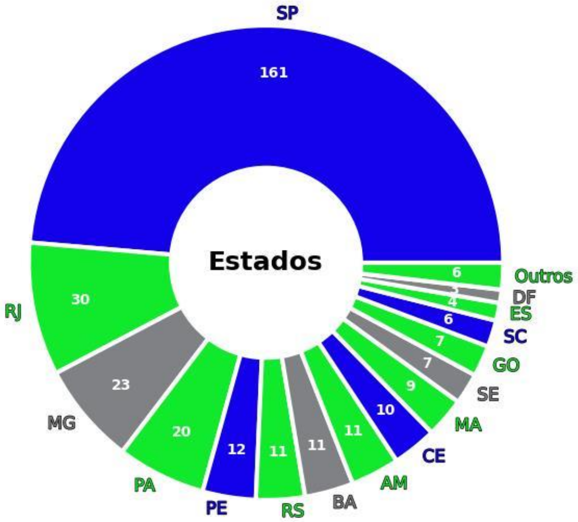
Sobre nós

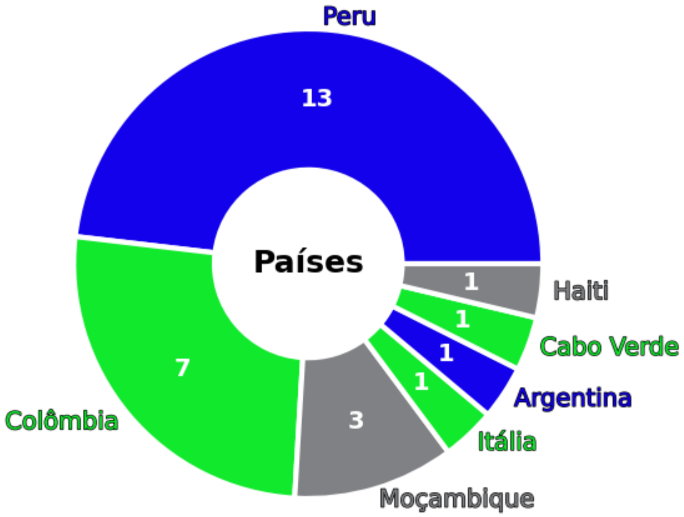


UNICAMP → 2018 / 2020 → Proteção Radiológica → 2022 → Física Aplicada

Sobre vocês


UNICAMP
USP
CNPEM
IPEN
UFRGS
UNESP
UnB
UERJ
UFRJ
UFC
UFSCar
UFPA
SENAI
UFMA
UFSC
UFPE
UFPI
UFU
UFF
UFAM
UFJF
UFS
UFG
UEM
IFAM
UFMG
UFES
IFBA
IFCE
IFSP
UFV
ITA
IFAL
UFRN
UPB
UFBA
UFRPE
IRD
UFABC
UFRRJ
PUCRJ
Oncorradium
Anhanguera
GeoOndas
SEE/SP
UFPB
SENAC
BrasilRad
Usco
UFLA
UFJ
SESI
UTP
UFOPA
IFPI
UFGD
UESC
FURG
MEPP
UFMT
UNIFAP
UFMS
ETEC
PUCRS
UFCat
UFSJ
UFAL
UVA
IFPB
IFSC
UEPB
IFNMG
UNIFESSPA
CEFET-MG
UFVJM
UTFPR
CEFET/RJ
UFPel
UEFS
UEA
IFMG
IFGO
UEL
UNICID
UTRPR
IFMA
UNISA
UFCSPA
UEPG
UNIFACISA
Radcare
PUCPR
UEMASUL
UEMA
Avibras
IFPE
UNIFESP
UFOP
Universidade São Francisco
Universidad Nacional Mayor de San Marcos
Centro de Desenvolvimento da Tecnologia Nuclear
Instituto Nacional de Tecnologia
Centro brasileiro de pesquisas físicas
Nome da Instituição
EACH - USP


Estados
SP
161
RJ
30
MG
23
PA
20
PE
12
RS
11
BA
11
AM
11
CE
10
MA
9
SE
7
GO
7
SC
6
ES
DF
Outros
6


Países
Peru
13
Colômbia
7
Moçambique
3
Haiti
1
Cabo Verde
1
Argentina
1
Itália
1

Cronograma

Tópicos Essenciais

01. Introdução
02. Monte Carlo
03. Geometria
04. Materiais
05. Feixes
06. USRBDX/USRTRACK
07. USRBIN I
08. USRBIN II

23/01 a 08/02

Tópicos Avançado

09. Transporte e física
10. Biasing
11. Ferramentas avançadas

10/02 a 15/02

Sem aulas - Carnaval

Aplicações Específicas

12. Source routines
13. Ativação
14. Campos eletromagnéticos
15. DICOM
16. Clusters
17. Sirius

24/02 a 08/03

Cronograma

CURSO INTRODUTÓRIO FLUKA

---------- Tópicos Essenciais ----------

1 – Introdução ao FLUKA: Aplicações do FLUKA. Apresentação da interface Flair. Simulação básica.

Data: 23/01/2023

Apresentadora: Isabela

2 – Introdução ao Método Monte Carlo: Funcionamento geral de um código Monte Carlo.

Data: 25/01/2023

Apresentadora: Tanus

3 - Geometria: como construir geometrias no FLUKA com geometria booleana, cartões, criando corpos e regiões. Ferramentas básicas de geometria da Flair. Cartão ruler: a régua do Flair.

Data: 27/01/2023

Apresentador: Tanus

4 - Materiais: construção de materiais e compostos com o FLUKA. Preprocessor (if, else, elif).

Data: 30/01/2023

Apresentadora: Isabela

5 – Feixes: parâmetros gerais do feixe. Cartões BEAM, BEAMPOS, BEAMAXES.

Data: 01/02/2023

Apresentadora: Isabela

6 - Detectores – USRBDX/USRTRACK: visualização de espectros e partícula por partícula.

Data: 03/02/2023

Apresentador: Tanus

7 - Detectores – USRBIN (parte I): dos plots de resultados do 1D ao 3D. Montagem de cartões. Parâmetros. Normalizações. Realizando plots com a interface.

Data: 06/02/2023

Apresentadora: Isabela

8 - Detectores – USRBIN (parte II): dos plots de resultados do 1D ao 3D. Plots sobrepostos. Arquivos de output. Parâmetros gnuplot.

Data: 08/02/2023

Apresentador: Tanus

---------- Tópicos Avançado----------

9 - Cartões de transporte e física: apresentação dos cartões EMFCUT (transport, PROD-CUT, ELPO-THR), PART-THR, DISCARD, MULSOPT, LOW-NEUT e PHYSICS (EVAPORAT, PEATHRES, COALESCE), PHOTONUC

Data: 10/02/2023

Apresentador: Tanus.

10 - Técnicas de Bias: métodos para atingir resultados com melhores estatísticas em menor tempo.

Data: 13/02/2023

Apresentadora: Isabela

11 – Ferramentas avançadas: Rotação e translação. Flair "avançado". Gerando vídeos com resultados.

Data: 15/02/2023

Apresentador: Tanus

Data: 17/02/2023 – Não haverá aula

Data: 20/02/2023 – Não haverá aula

Data: 22/02/2023 – Não haverá aula

---------- Aplicações Específicas ----------

12 - Uso de espectros com source routines: Importando espectros externos no FLUKA.

Data: 24/02/2023

Apresentador: Tanus

13 - Ativação e produção de radioisótopos: apresentação dos principais cartões para o estudo da ativação induzida por feixe de partículas

Data: 27/02/2023

Apresentador: Tanus

14 - Campos eletromagnéticos: implementação de campos eletromagnéticos nas simulações.

Data: 01/03/2023

Apresentadora: Isabela

15 - Imagens DICOM: demonstração do carregamento de imagens DICOM na geometria no FLUKA.

Data: 03/03/2023

Apresentadora: Isabela

16 - Simulando em clusters: demonstração de exemplos de adaptações para simulações em clusters. Enviando e recebendo arquivos das simulações.

Data: 06/03/2023

Apresentadora: Isabela

17 – FLUKA no Sirius – Apresentação das principais definições realizadas com o FLUKA em um acelerador tipo síncrotron.

Data: 08/03/2023

Apresentadores: Isabela e Tanus

CNPEM
ORGANIZAÇÃO SOCIAL DO MCTI

Introduction to FLUKA

FLUKA.CERN

Código Monte Carlo de transporte e interação da radiação com a matéria.

- TIPOS DE RADIAÇÃO:
- Fótons (>100 eV)
- Nêutrons
- Prótons
- Elétrons (> 1keV)
- Pósitrons
- Neutrinos
- ...

FLUKA.CERN

Código Monte Carlo de transporte e interação da radiação com a matéria.

TIPOS DE RADIAÇÃO:

- Fótons (>100 eV)
- Nêutrons
- Prótons
- Elétrons (> 1keV)
- Pósitrons
- Neutrinos
- ...

GRANDEZAS MEDIDAS:

- Energia depositada
- Dose absorvida
- Dose efetiva/equivalente
- Fluência/Fluxo
- Atividade
- DPA
- ...

FLUKA.CERN

Código Monte Carlo de transporte e interação da radiação com a matéria.

TIPOS DE RADIAÇÃO:

- Fótons (>100 eV)
- Nêutrons
- Prótons
- Elétrons (> 1keV)
- Pósitrons
- Neutrinos
- ...

GRANDEZAS MEDIDAS:

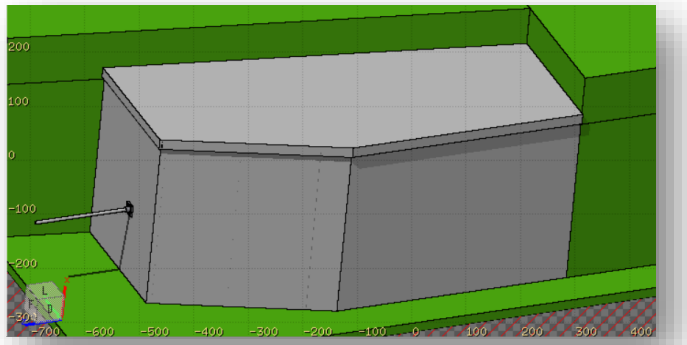
- Energia depositada
- Dose absorvida
- Dose efetiva/equivalente
- Fluência/Fluxo
- Atividade
- DPA
- ...

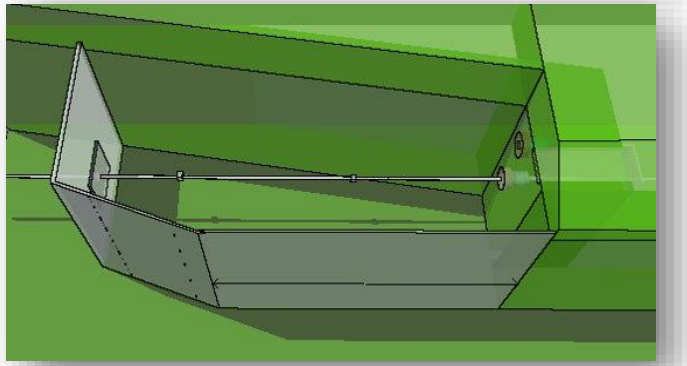
APLICAÇÕES:

- Cálculo de blindagem
- Design de detectores
- Física de partículas
- Dosimetria
- Aplicações médicas
- Design de aceleradores
- ...

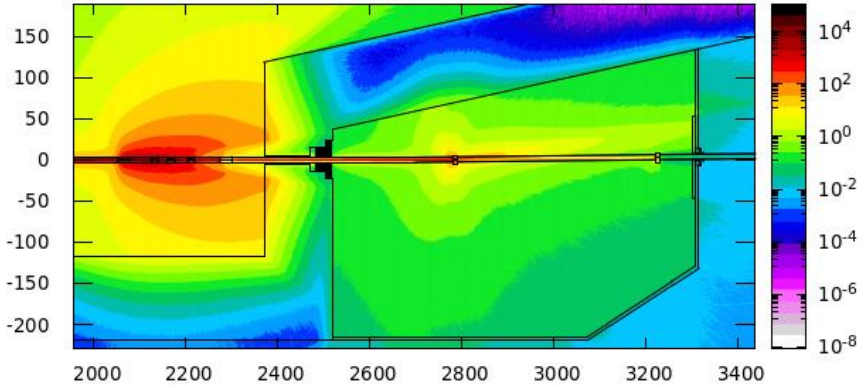
Capacidades do FLUKA

Transporte e deposição de partículas e suas secundárias

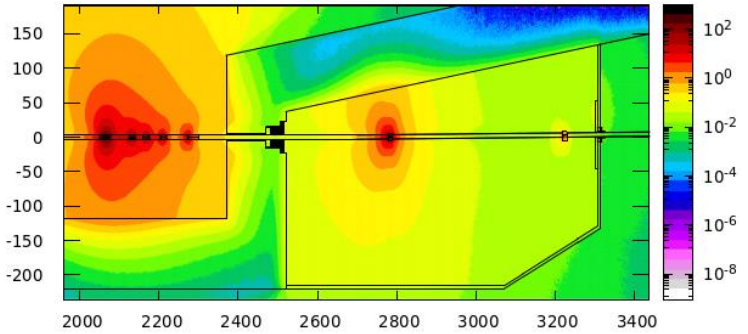




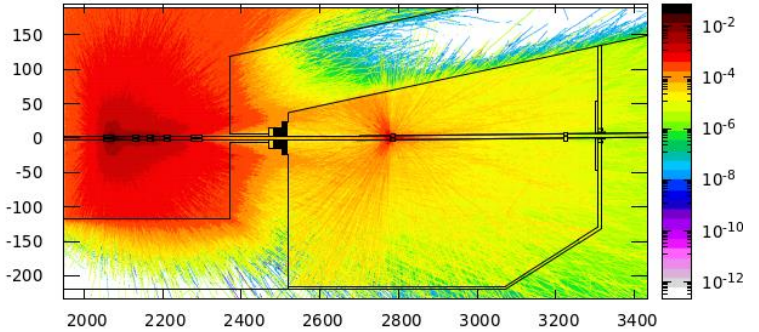
Todas as partículas transportadas pelo código



Nêutrons

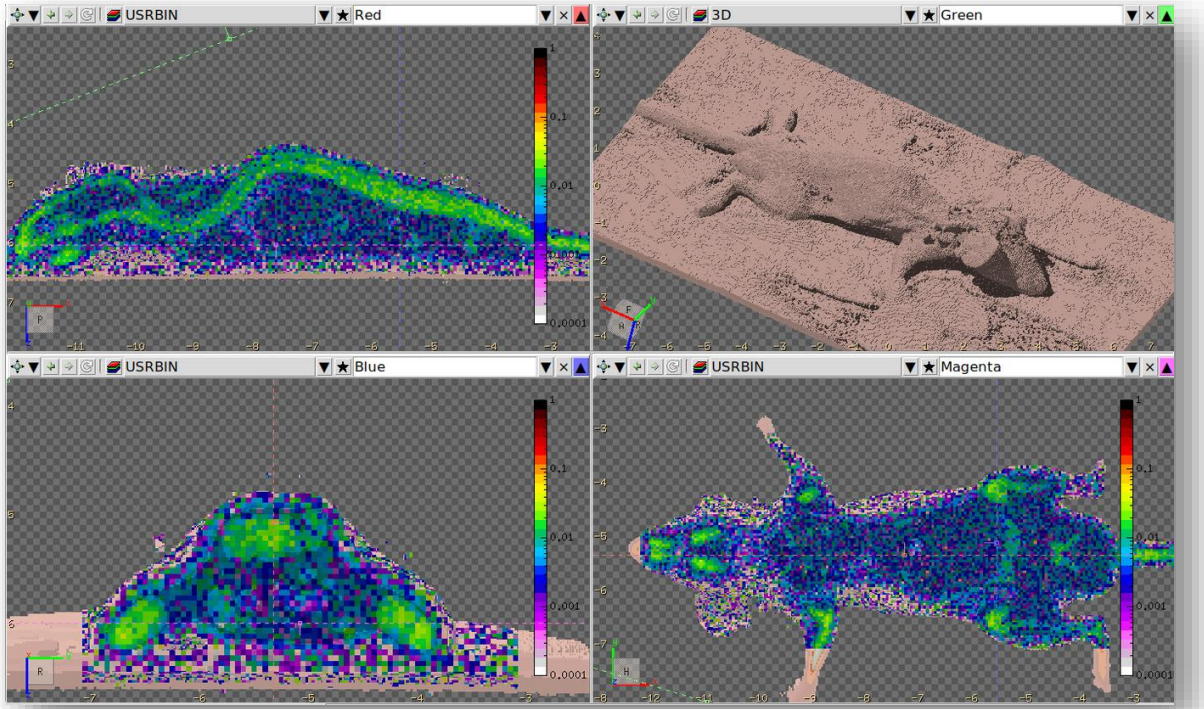


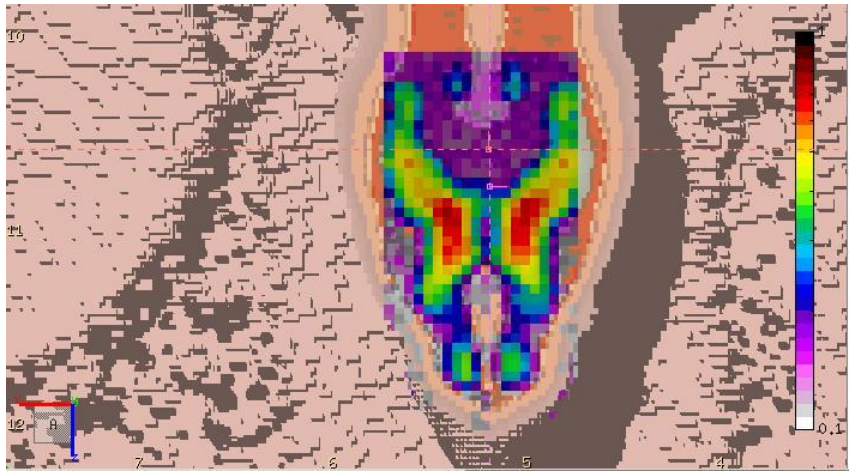
Múons



Capacidades do FLUKA

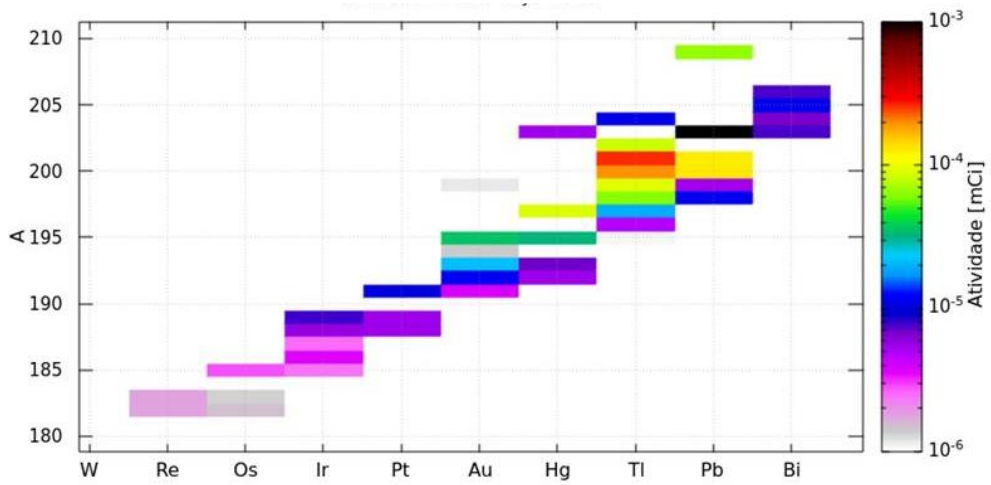
Geometrias complexas



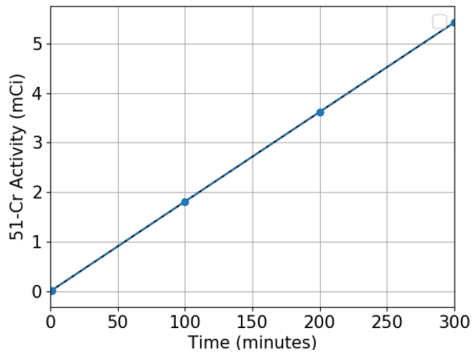


Capacidades do FLUKA

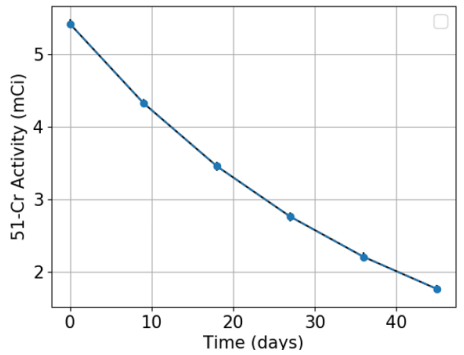
Avaliação da quantidade de material radioativo produzido em função do tempo

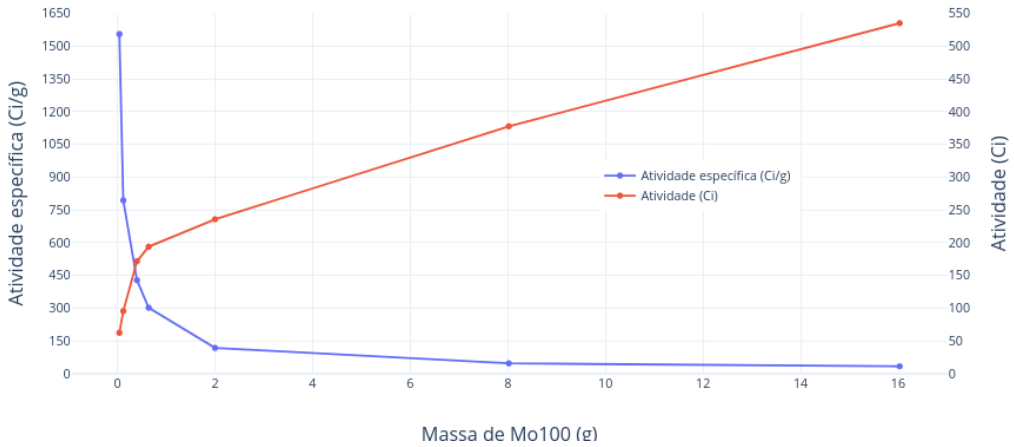


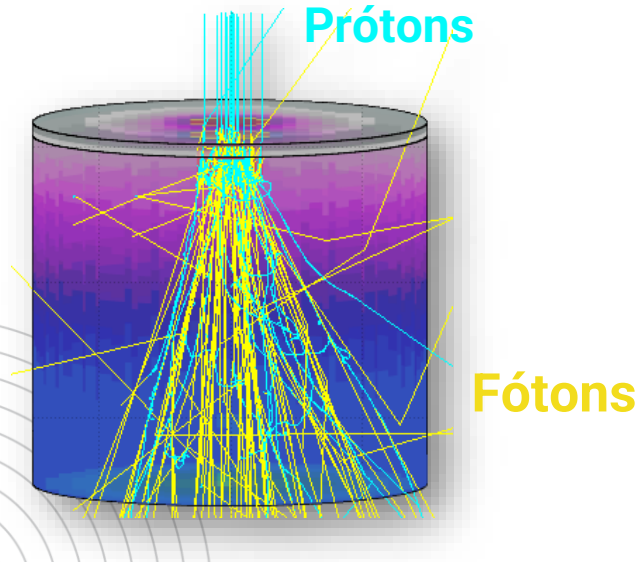
Atividade durante irradiação



Atividade após irradiação







Suporte aos usuários

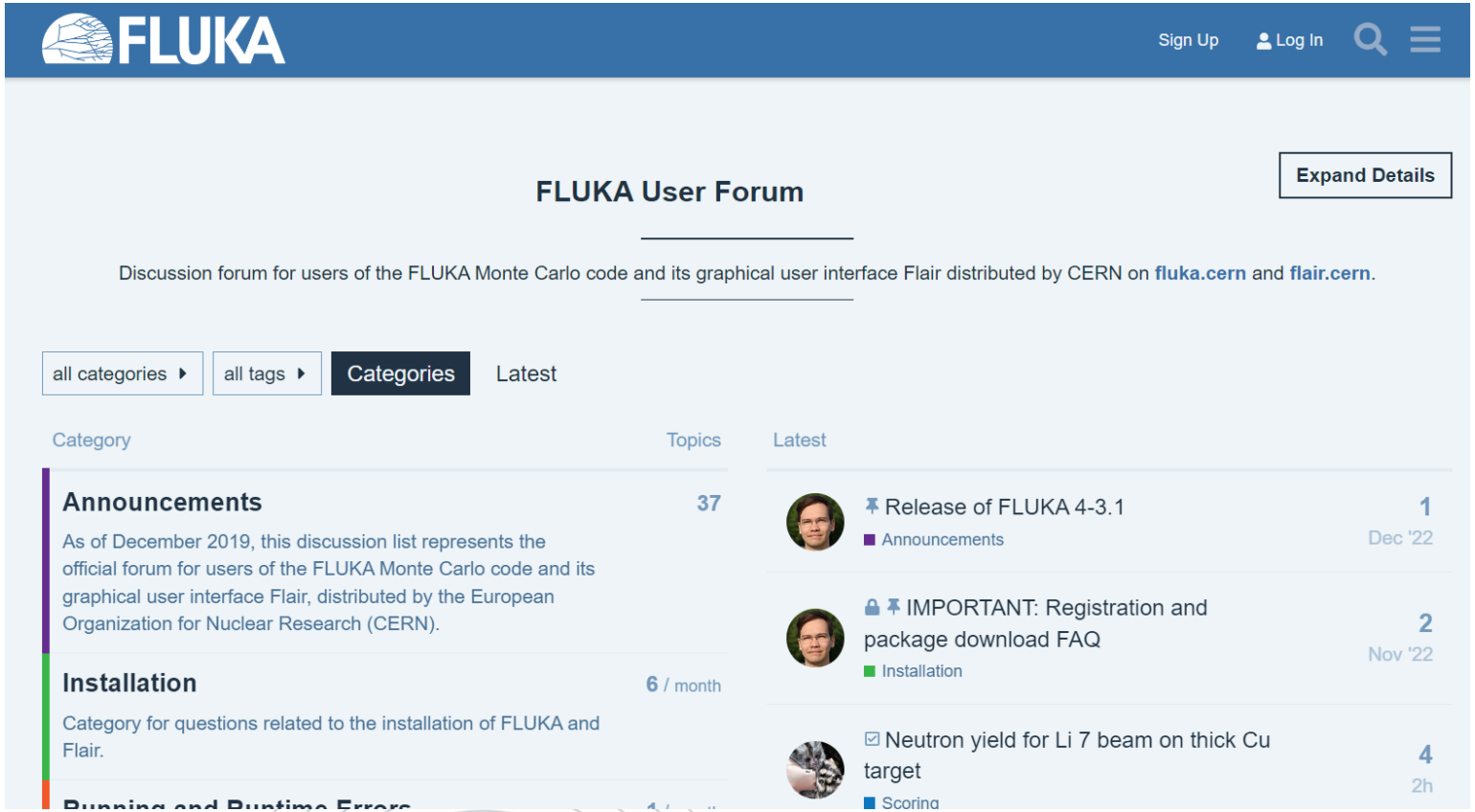
Cursos e fórum de discussão para os usuários do FLUKA

Curso Iniciante (Beginners' Training):

- Ocorre anualmente.

Curso Avançado (Advanced Course):

- Última edição ocorreu em 2019;
- Planejado para ocorrer entre 5 e 9 de junho nos EUA em 2023.



Flair

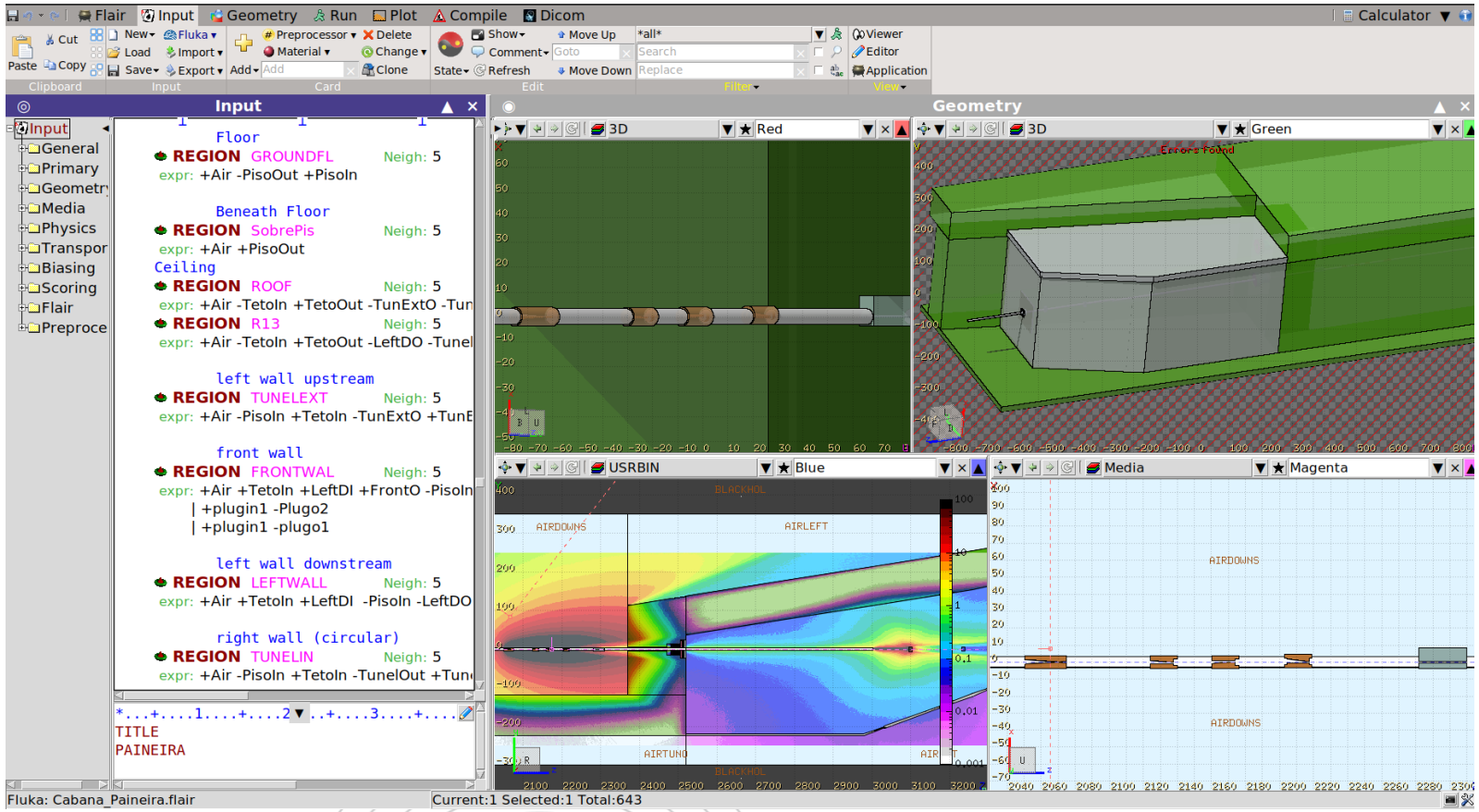
A Interface Gráfica do FLUKA.CERN (e futuramente de outros códigos)

Authors

authors: Vasilis Vlachoudis *(lead author)*
Christian Theis
Wioletta Kozlowska

Features

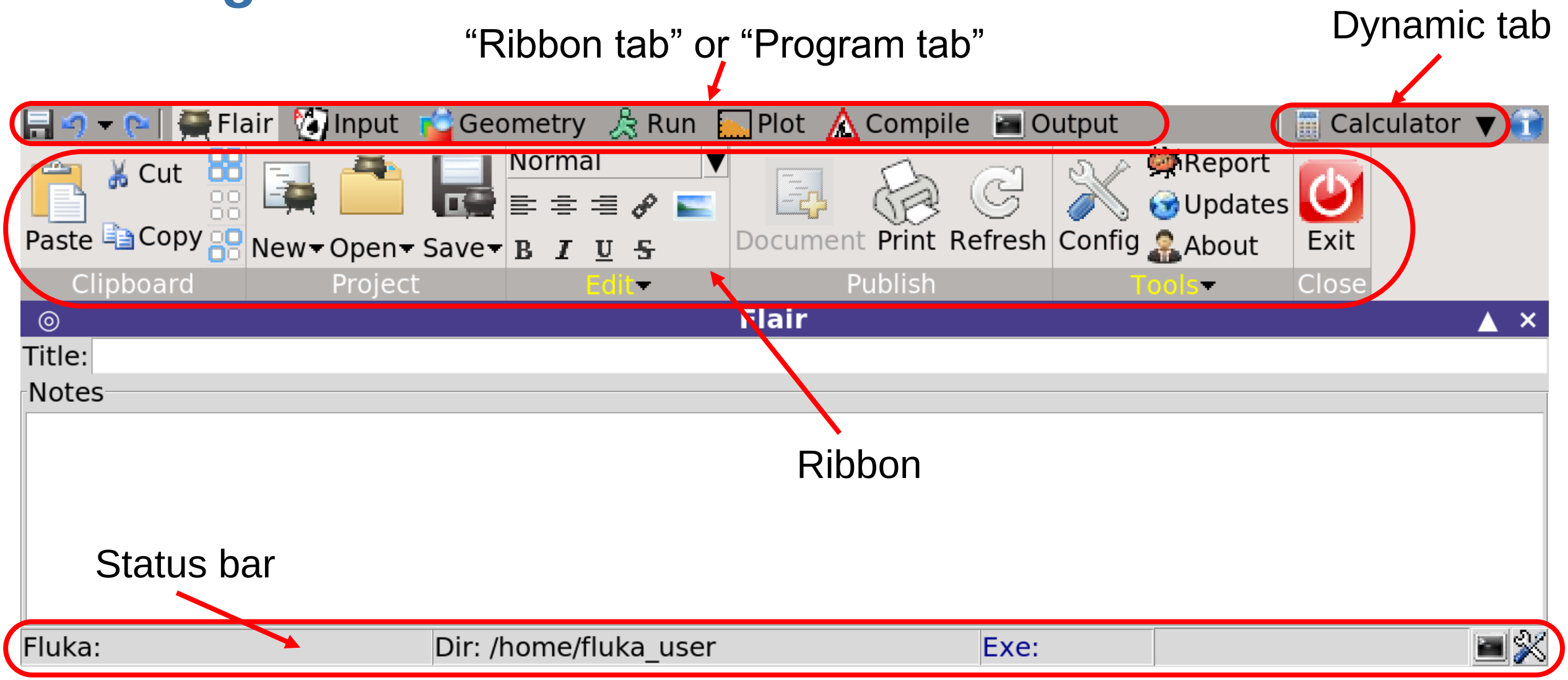
- modern and intuitive design
- Input editor for error free inputs
- Interactive geometry editor, photorealistic ray tracer and debugger
- run and monitor the simulation
- back-end for post-processing of results
- I/O of other simulation formats (MCNPX,GDML,...)
- Medical file importing, DICOM, RT-PLAN,DOSE,...
- extended material library



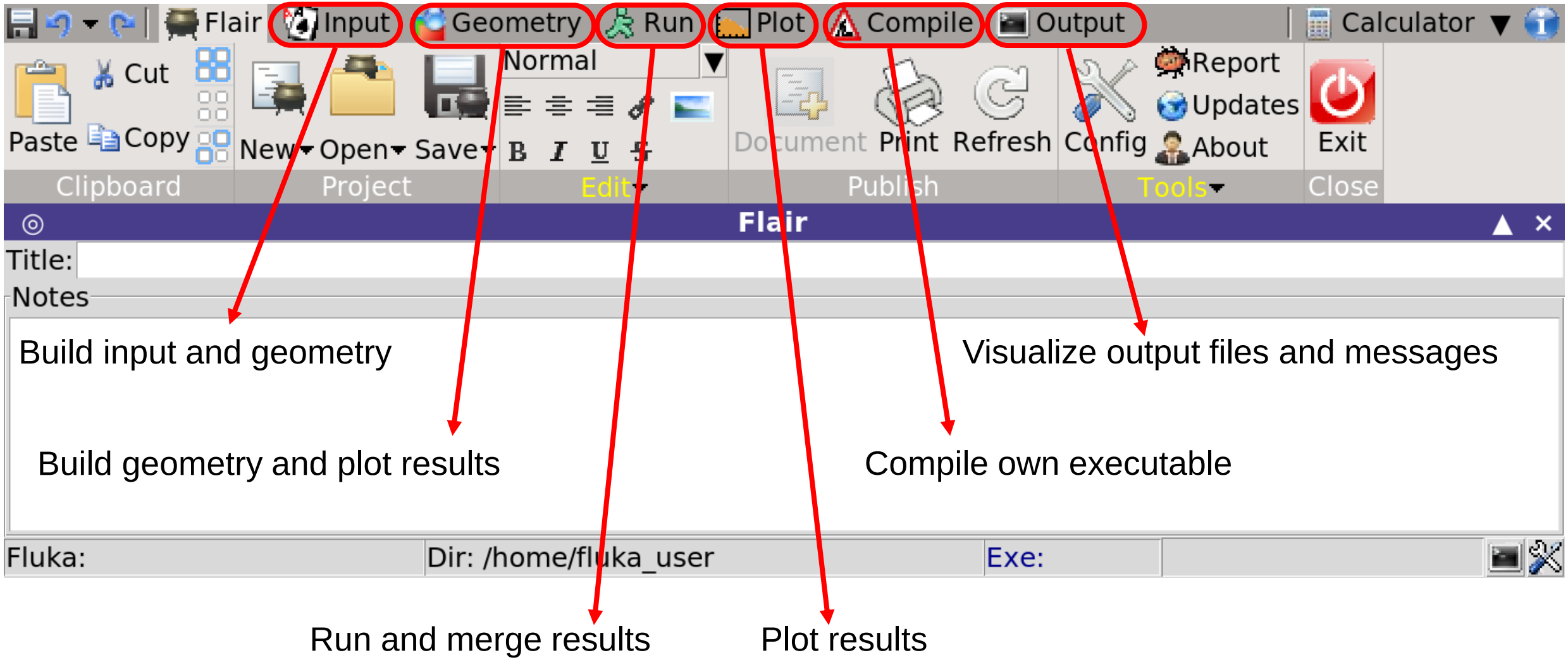
Introduction to Flair and basic input

A very basic introduction to perform your first simulation

Starting Flair and basic nomenclature



What's each tab for?



The input as a text file

- Mentioned here just for completeness

.inp

```
TITLE
basic template
* Set the defaults for precision simulations
DEFAULTS                                              PRECISIO
* Define the beam characteristics
BEAM
* Define the beam position
BEAMPOS
GEOBEGIN                                              COMBNAME
    0    0
* Black body
SPH blkbody    0.0 0.0 0.0 100000.0
* 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)
```

.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

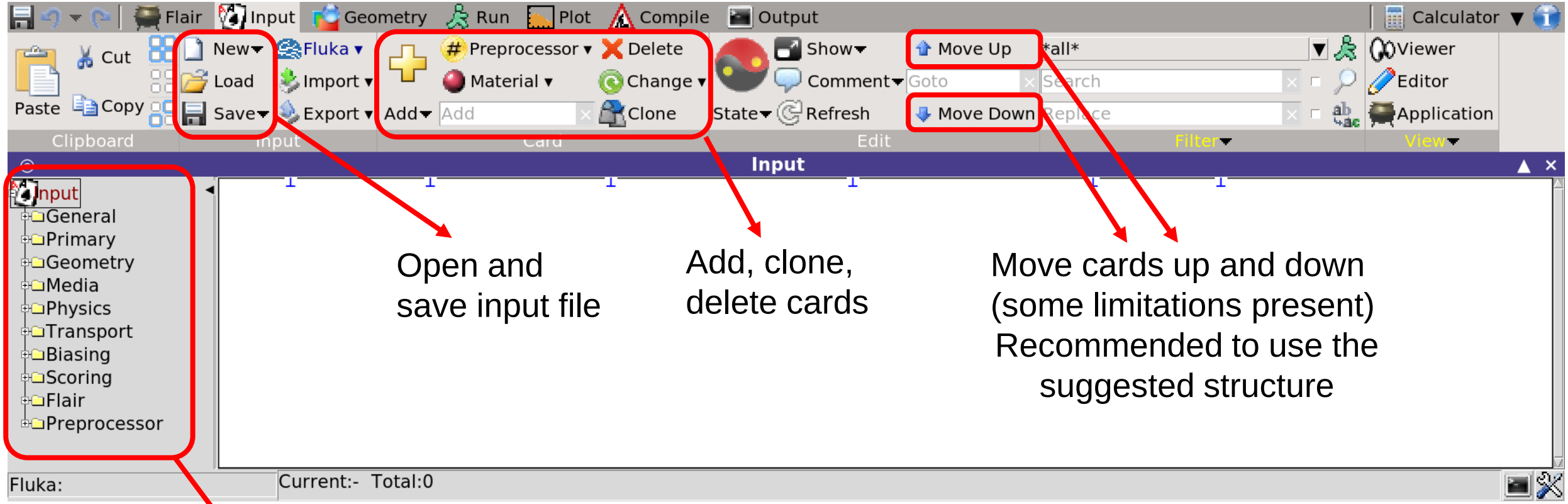
```
# flair project file
Version: 300
Mode: fluka
md5: c8e26fe184526e9282e8555b8fab2455
Input:
TITLE
        fully-working template
#define pointless_define_1 10
#define pointless_define_2
*Set the defaults for precision simulations
DEFAULTS PRECISIO
*Define the beam characteristics
BEAM PROTON 0.8
*Define the beam position
BEAMPOS , 0. 0. -1.
GEOBEGIN COMBNAME
*Black body
SPH blkbody 0.0 0.0 0.0 100000.0
*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
```

Input tab – 1: general info

- Standard looking “Windows” tab



Open and save input file

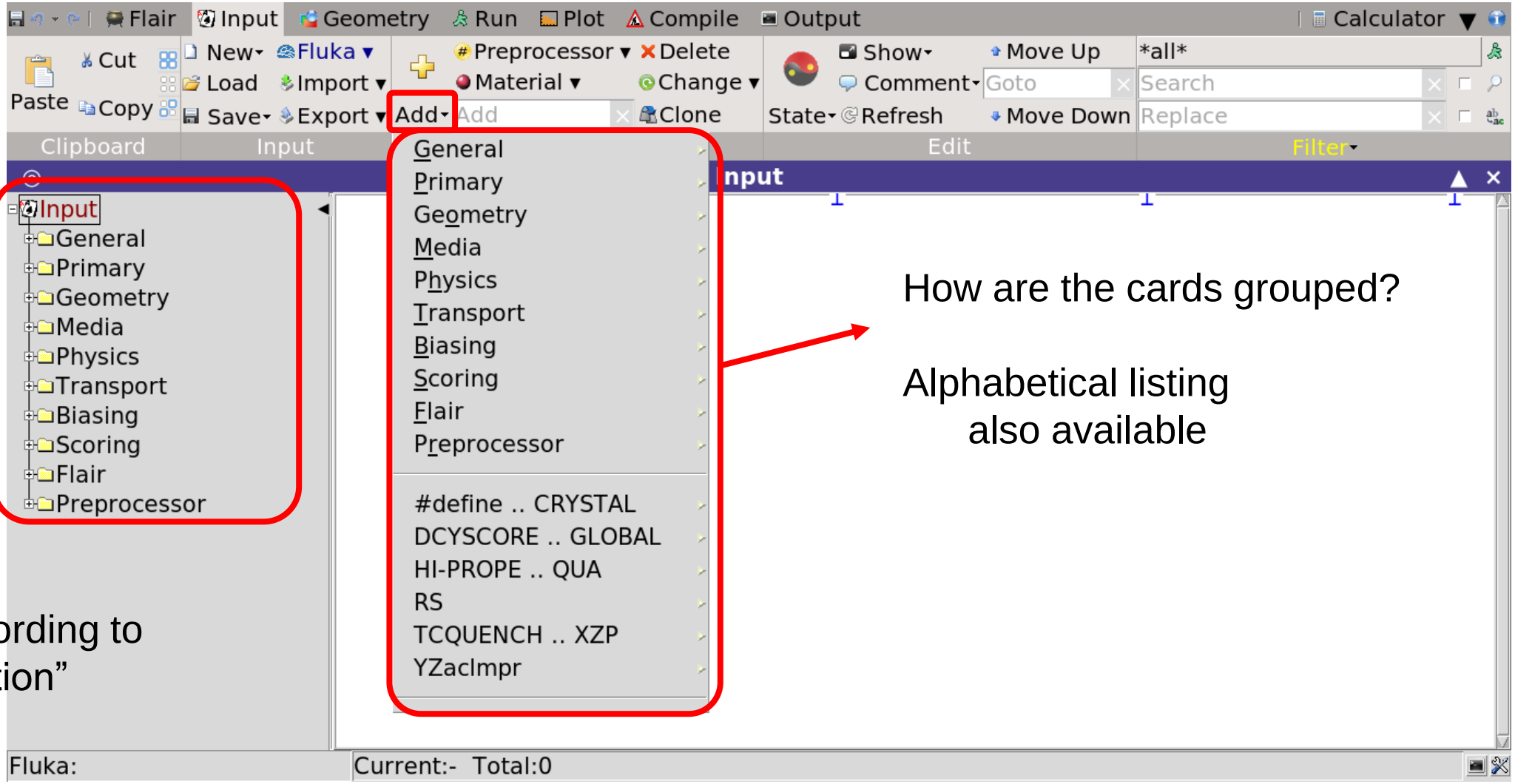
Add, clone, delete cards

Move cards up and down (some limitations present) Recommended to use the suggested structure

Input file tree
Cards grouped according to their “field of action”

Input tab – 2: input file tree and card grouping

- Input file tree and card grouping



How are the cards grouped?

Alphabetical listing also available

Input file tree
Cards grouped according to their "field of action"

Input tab – 3: input file 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

} dedicated lectures

CURSO INTRODUTÓRIO

FLUKA

23 DE JANEIRO
A 8 DE MARÇO
DE 2023

Código Monte Carlo de interação e transporte de partículas

Pausa

Voltamos em 15 minutos

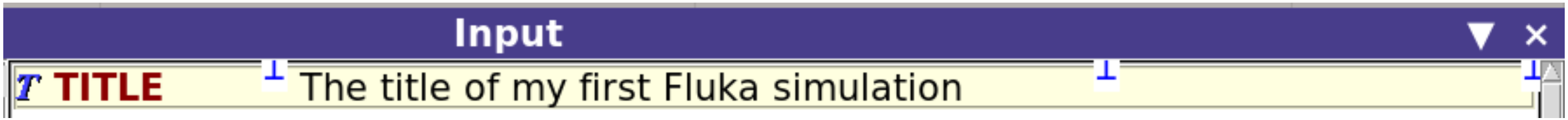
CNPEM
ORGANIZAÇÃO SOCIAL DO MCTI

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

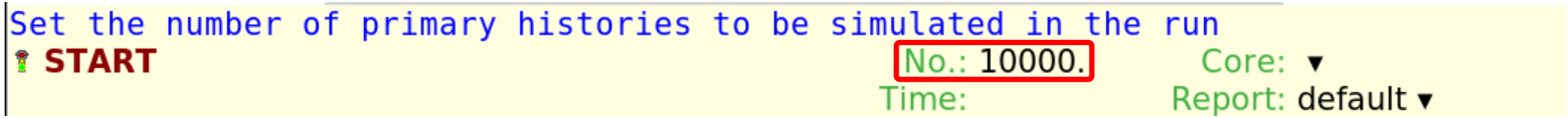


Input tab – 5: General cards

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



Input tab – 6: General cards

TITLE START STOP RANDOMIZe DEFAULTS

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

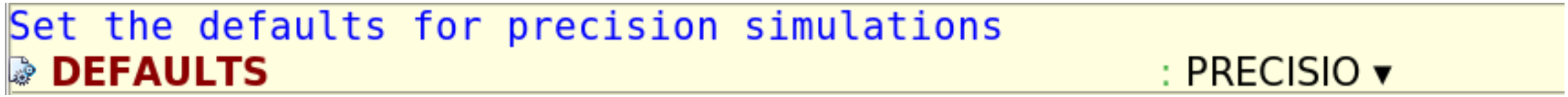
- Allows to initialize different random sequences
- For debugging purposes, the “random seed” must be the same
- Different “random seeds” are required in order to differentiate histories
- Flair takes care of the “random seeds” when spawning runs (see later)

Input tab – 8: General cards

TITLE **START** **STOP** **RANDOMIZe** **DEFAULTS**

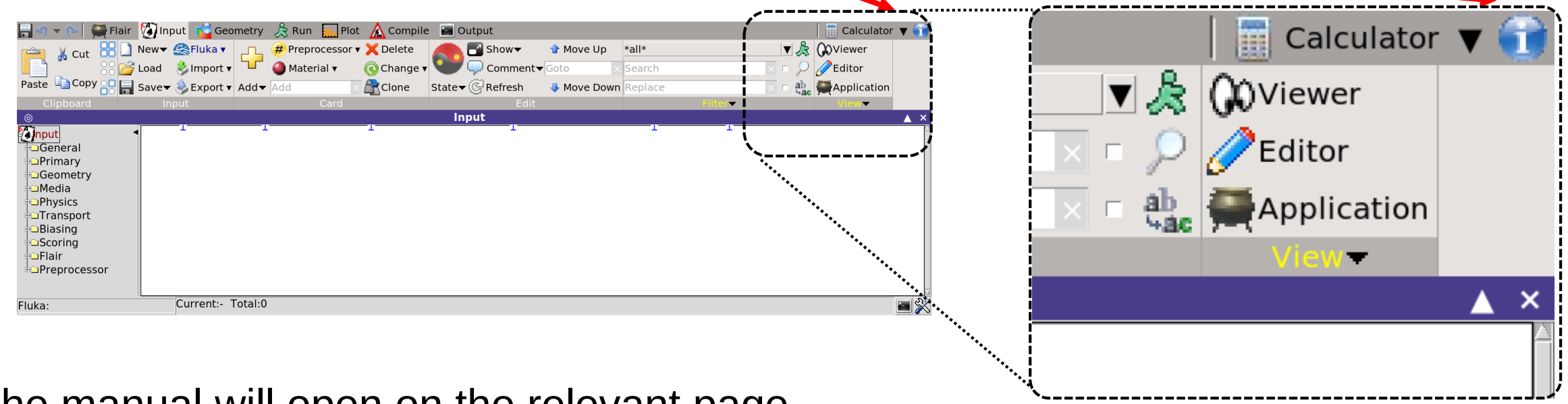
DEFAULTS

- 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



The manual

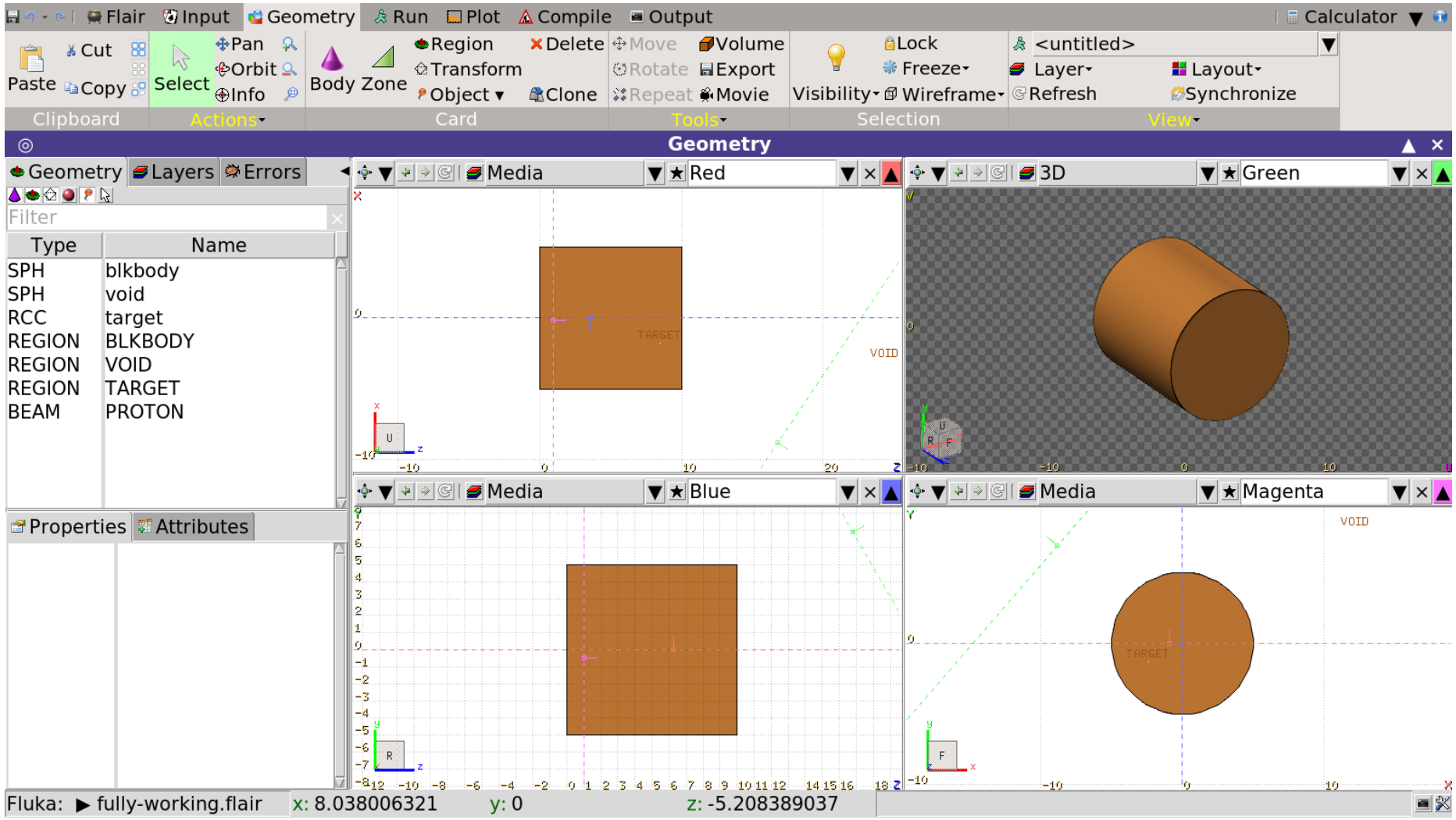
- Can be accessed using F1 button
- Can be accessed clicking on the “info” button



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

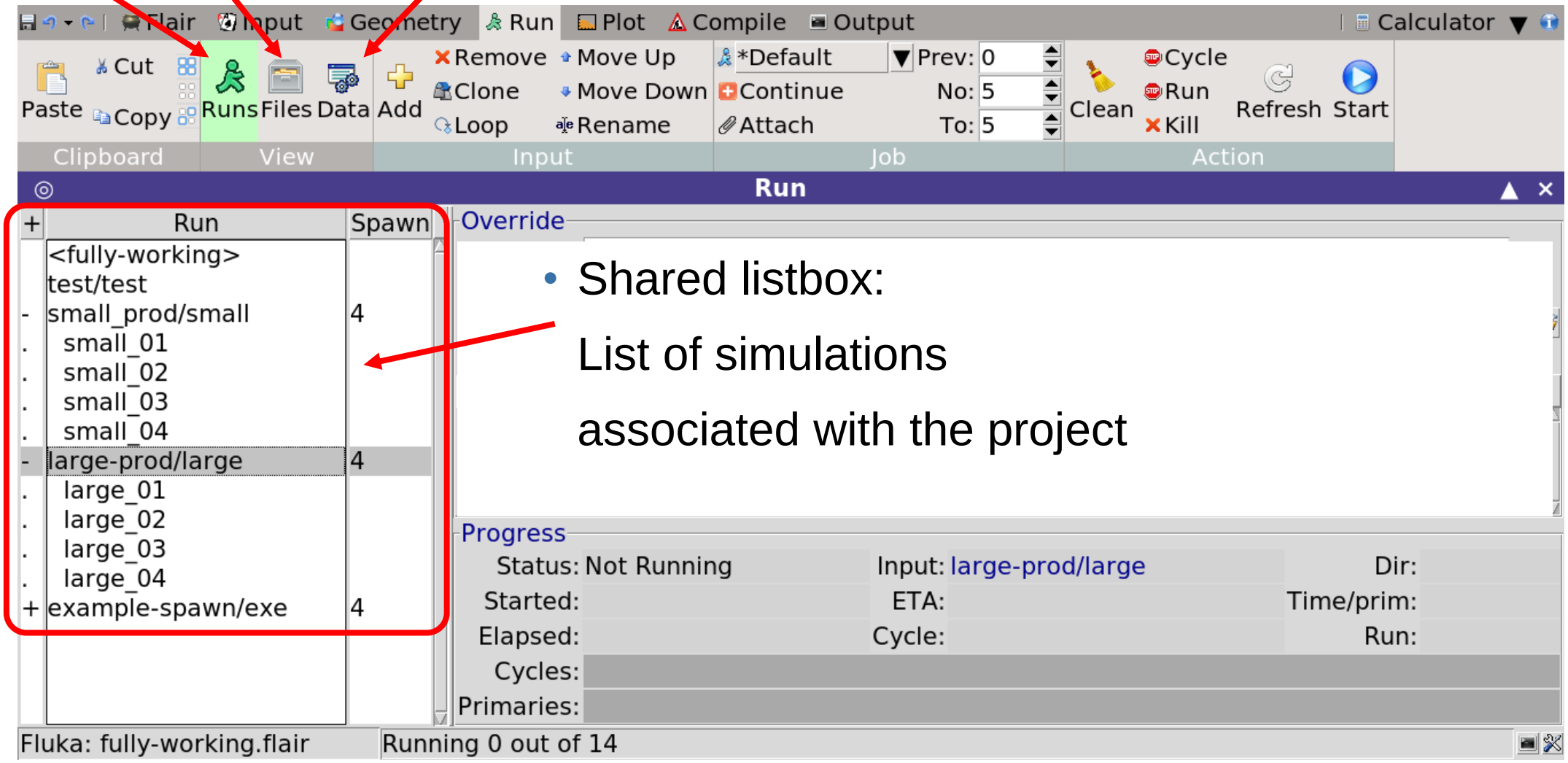
Geometry tab – 1

- Visualise and edit geometry
- Plot results
- Dedicated lectures



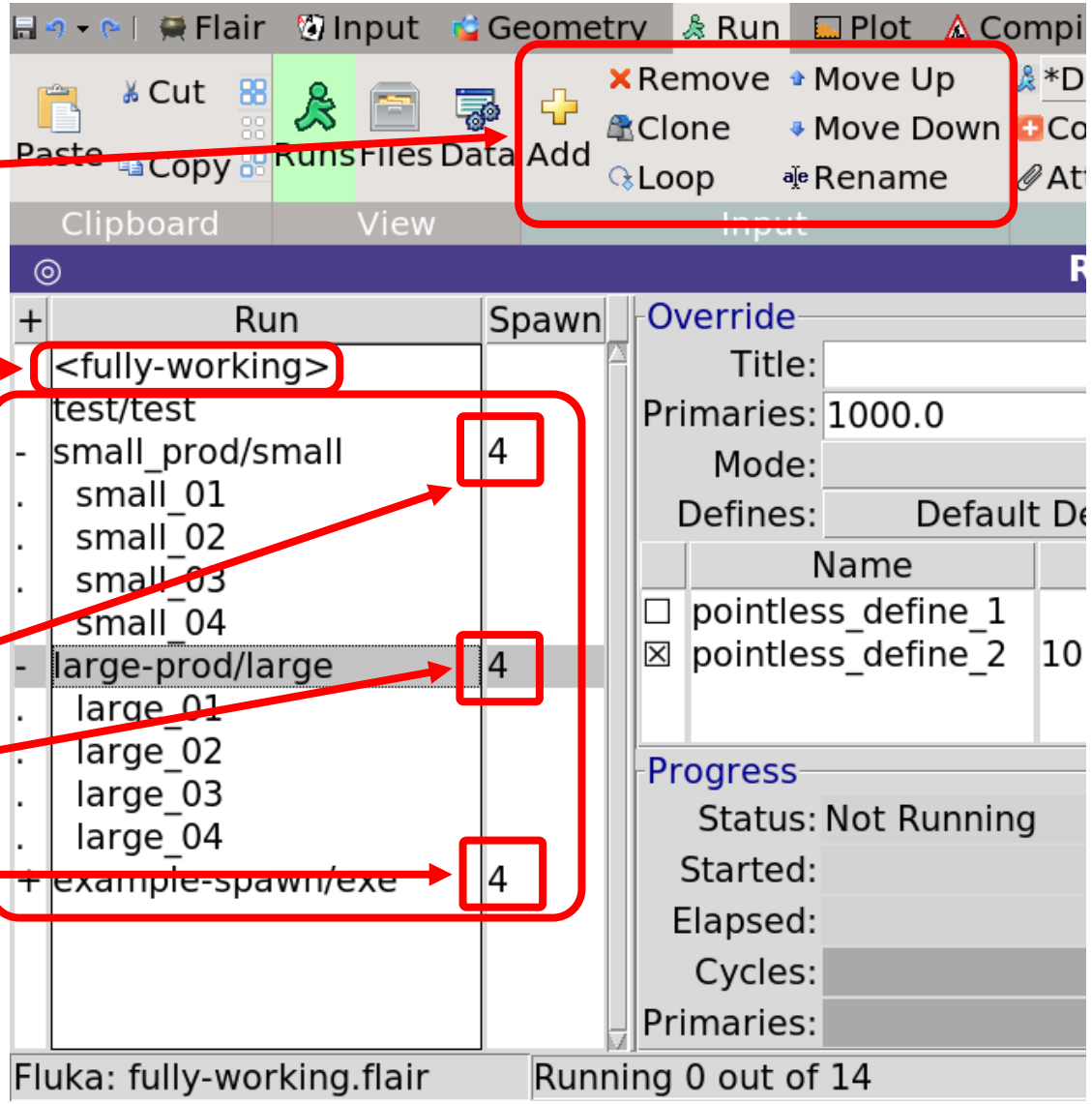
Run tab

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



Run tab – Runs view – 1

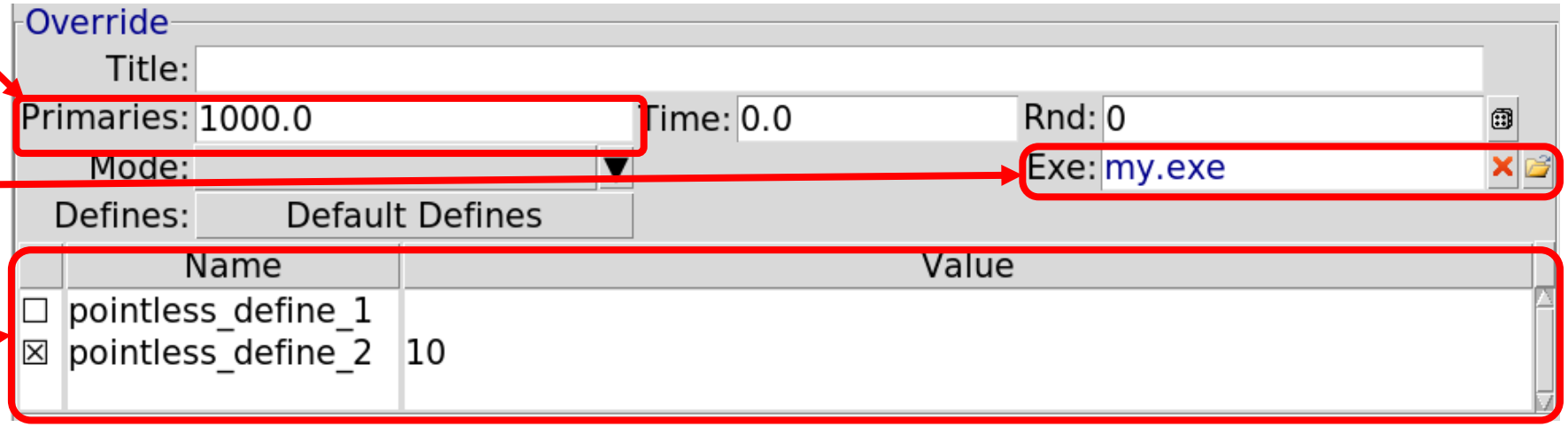
- Management of the various simulations
- Basic inputfile of the Flair project
- Different simulations associated with the Flair project
- Number of spawns



Run tab – Runs view – 2

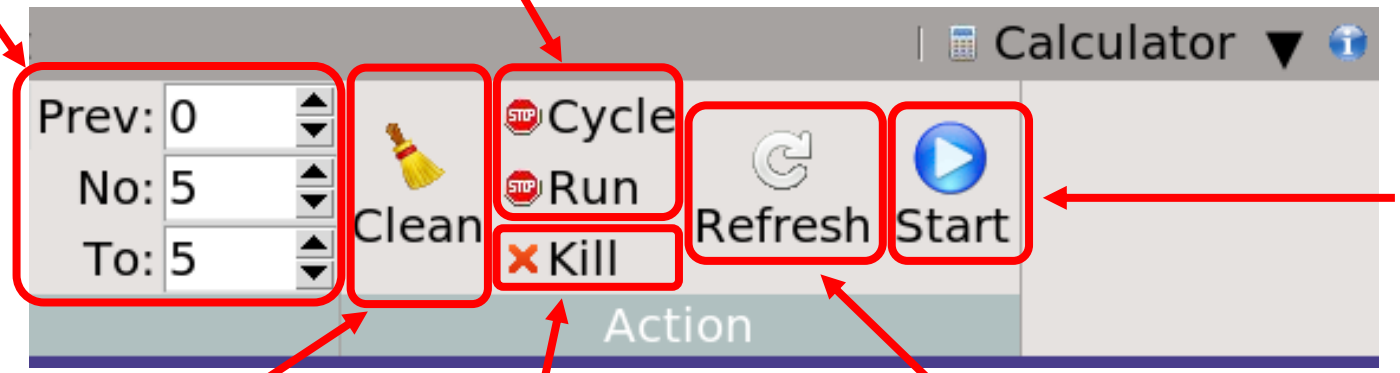
- Override of inputs
- Number of primaries
- Executable
- #define

Dedicated lecture



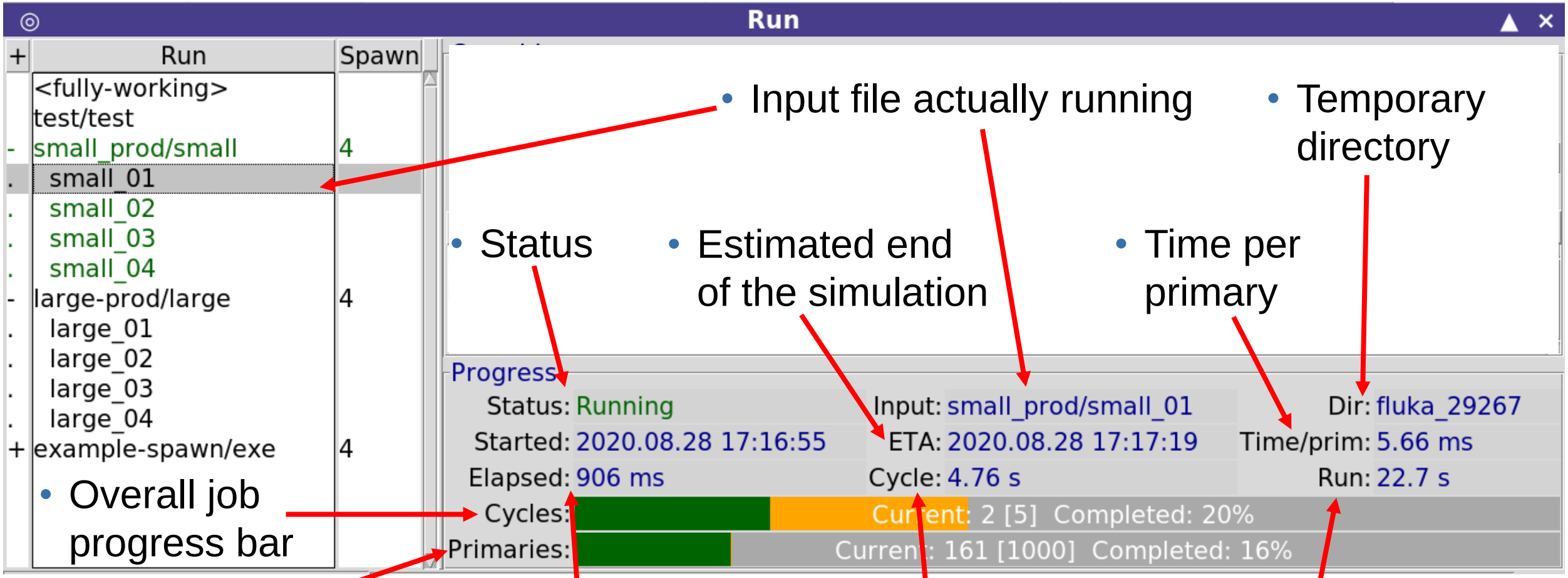
Run tab – Runs view – 3

- Cycles control: how many cycles to run, starting from which cycle
- Cleanly stop the cycles/runs currently running



- Start a simulation
- Remove files from a previous simulation
- Refresh the progress field
- Kill the current simulations

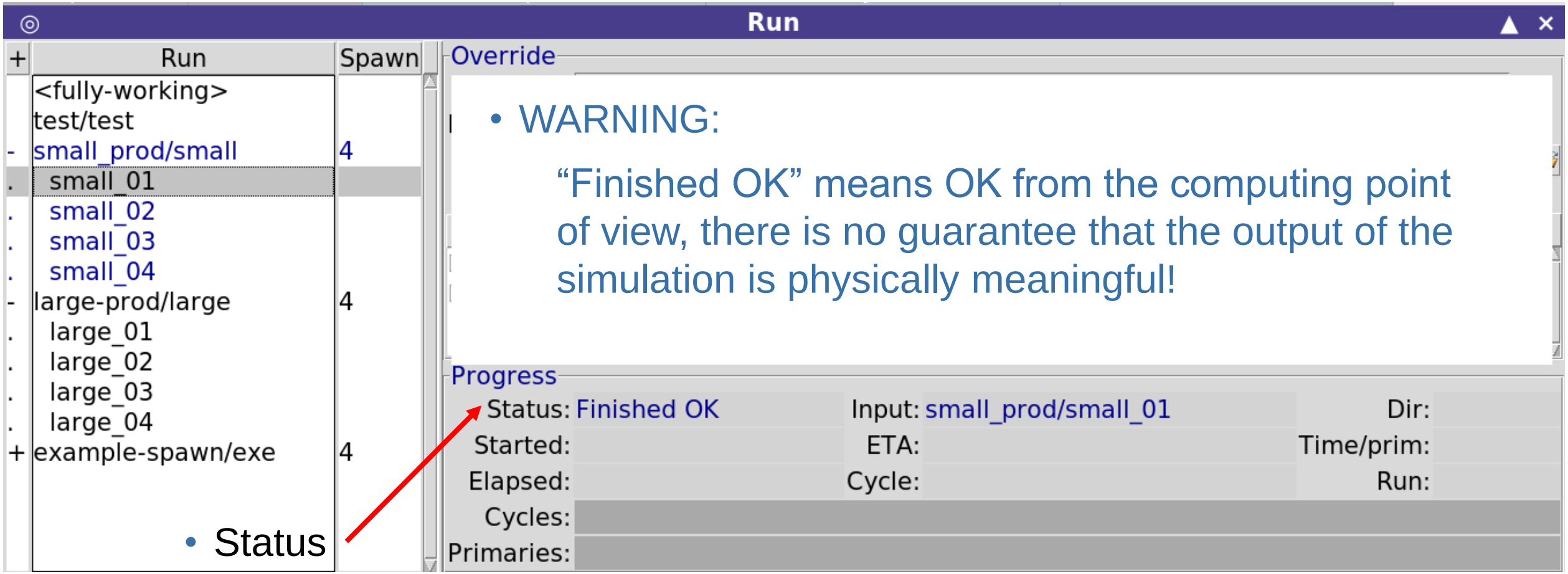
Run tab – Runs view – 4



- Input file actually running
- Temporary directory
- Status
- Estimated end of the simulation
- Time per primary
- Overall job progress bar
- Cycle progress bar
- Time since the start of the cycle
- Time until the end of the cycle
- Time until the end of the simulation

Run tab – Runs view – 5

- At the end of the simulations…



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$
```

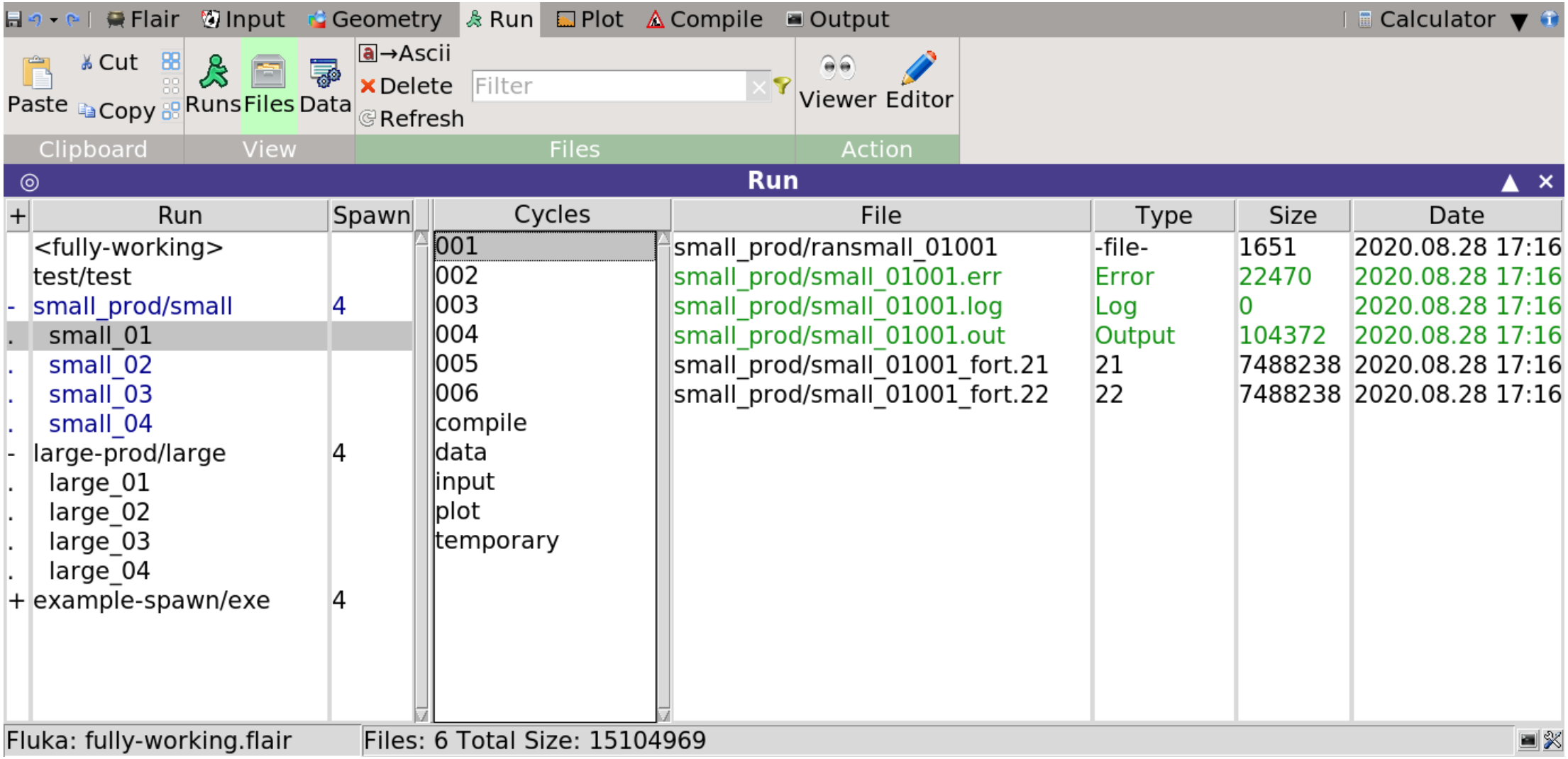
- Content of the working sub-directory

```
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.
ransmall_01004  small_01.out         small_01005.log      small_02004.
ransmall_01005  small_01001.err      small_01005.out      small_02004.
ransmall_01006  small_01001.log      small_01005_fort.21  small_02004
ransmall_02001  small_01001.out      small_01005_fort.22  small_02004
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$
```

`.inp` and `.out` files specific of each spawn

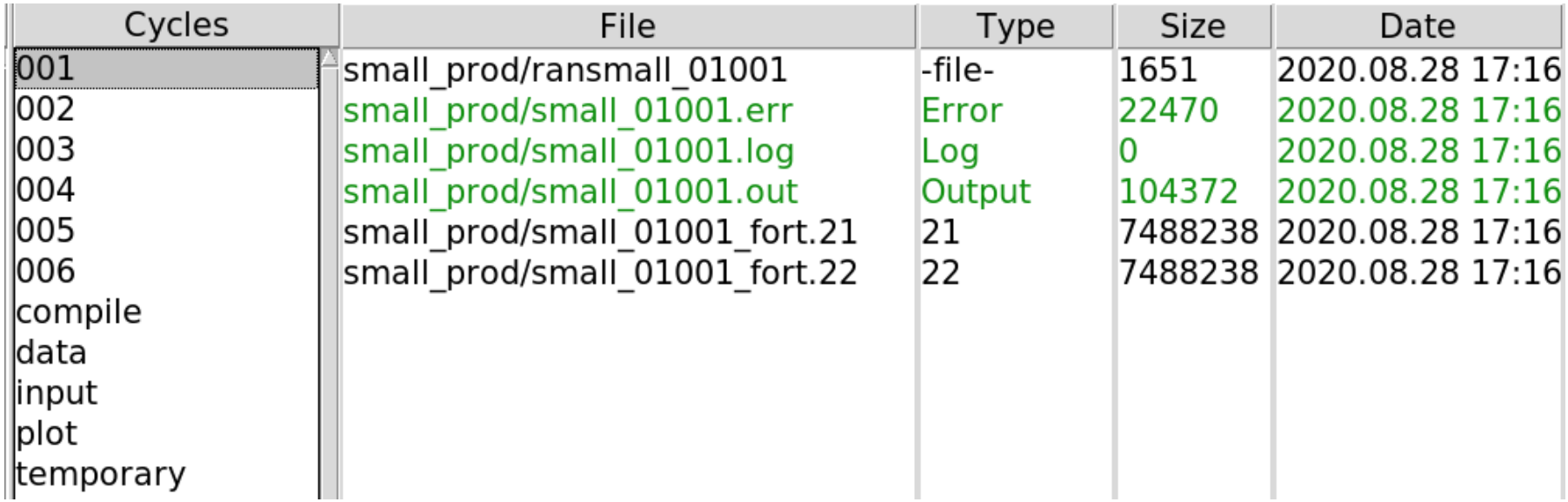
Run tab – Files view – 1

- Generated files accessible via the Files view



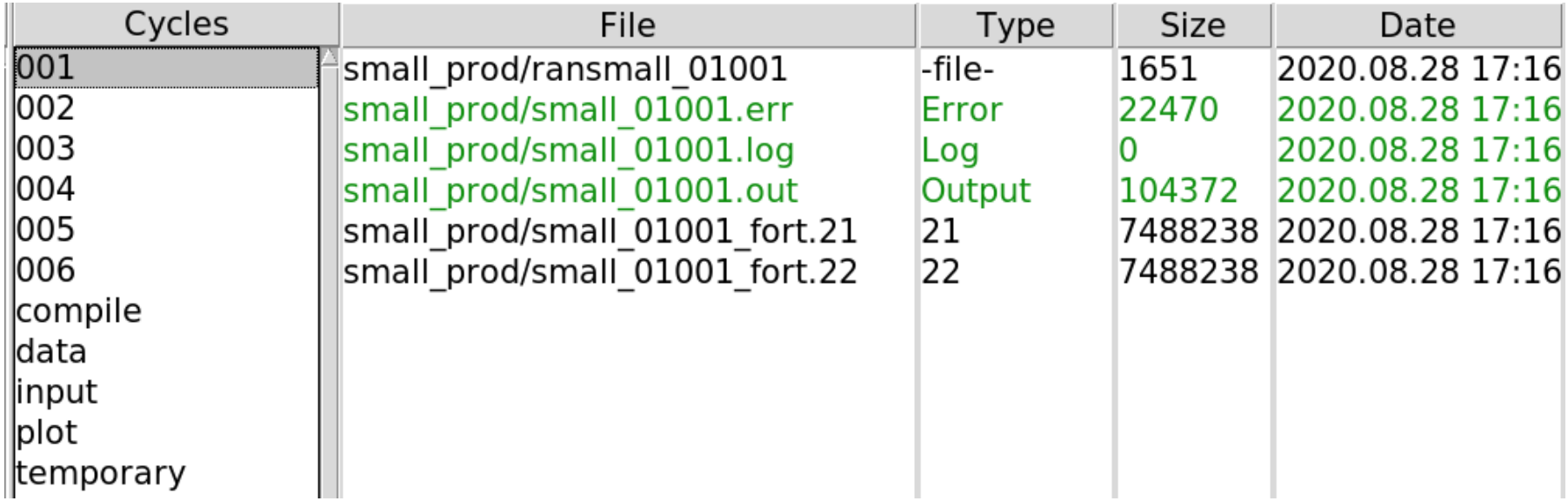
Run tab – Files view – 2

- 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



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`**



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`

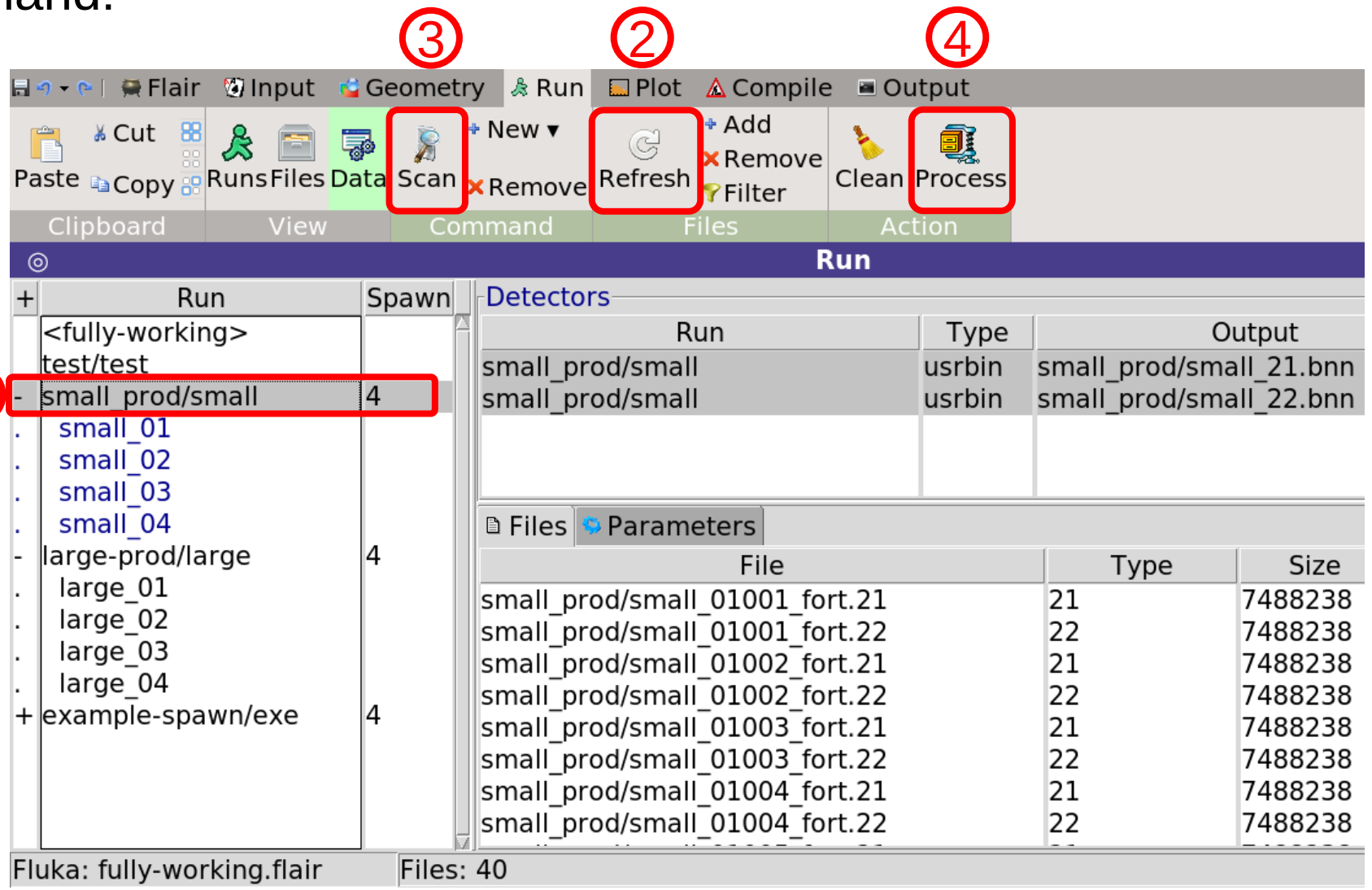
`small_01001_fort.21`

`small_01001_fort.22`

Run tab – Data view – 4

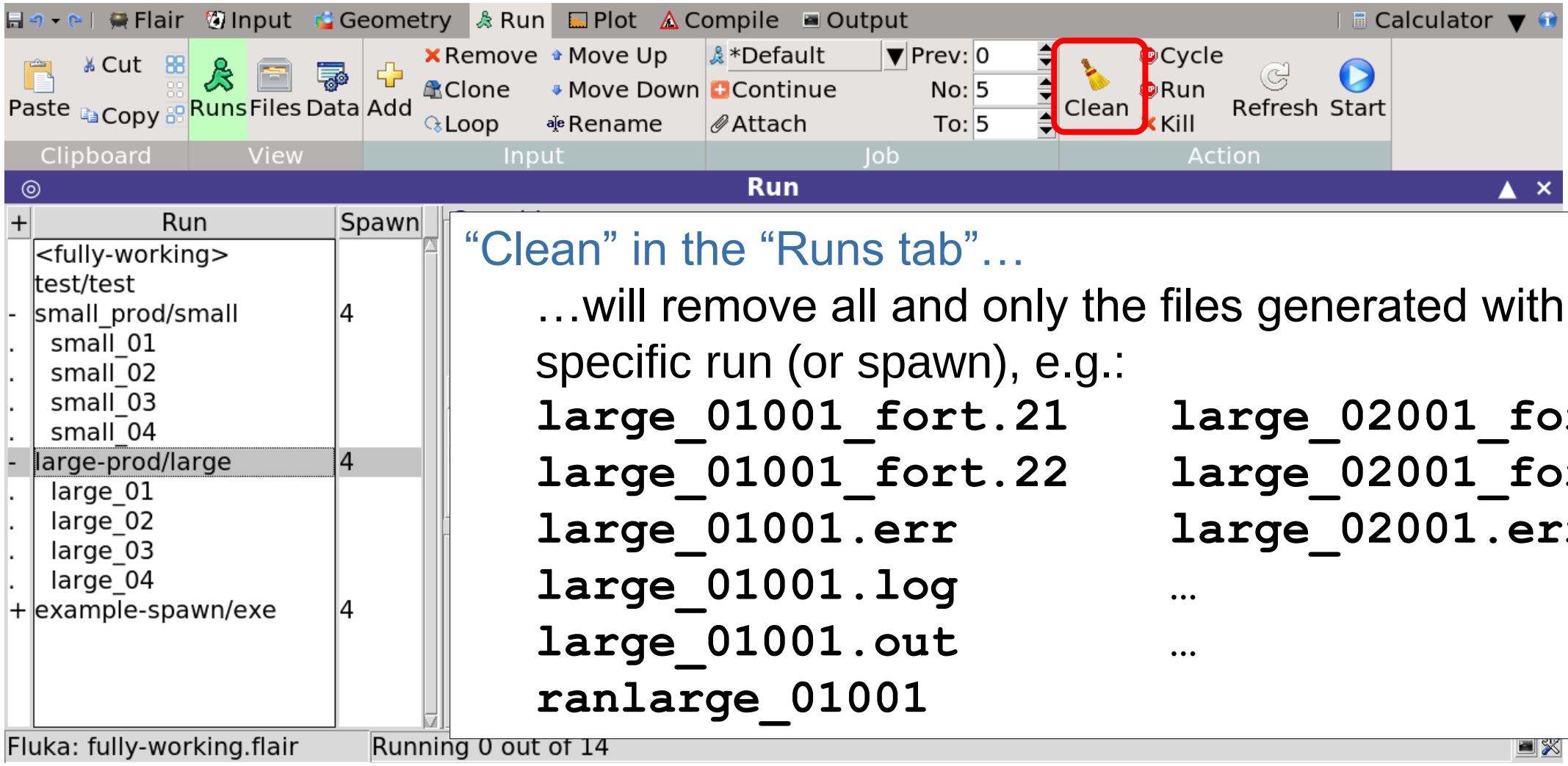
- Process can be forced by hand:
- 1-Select the run
- 2-Refresh
- 3-Scan
- 4-Process (merge)

- 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!



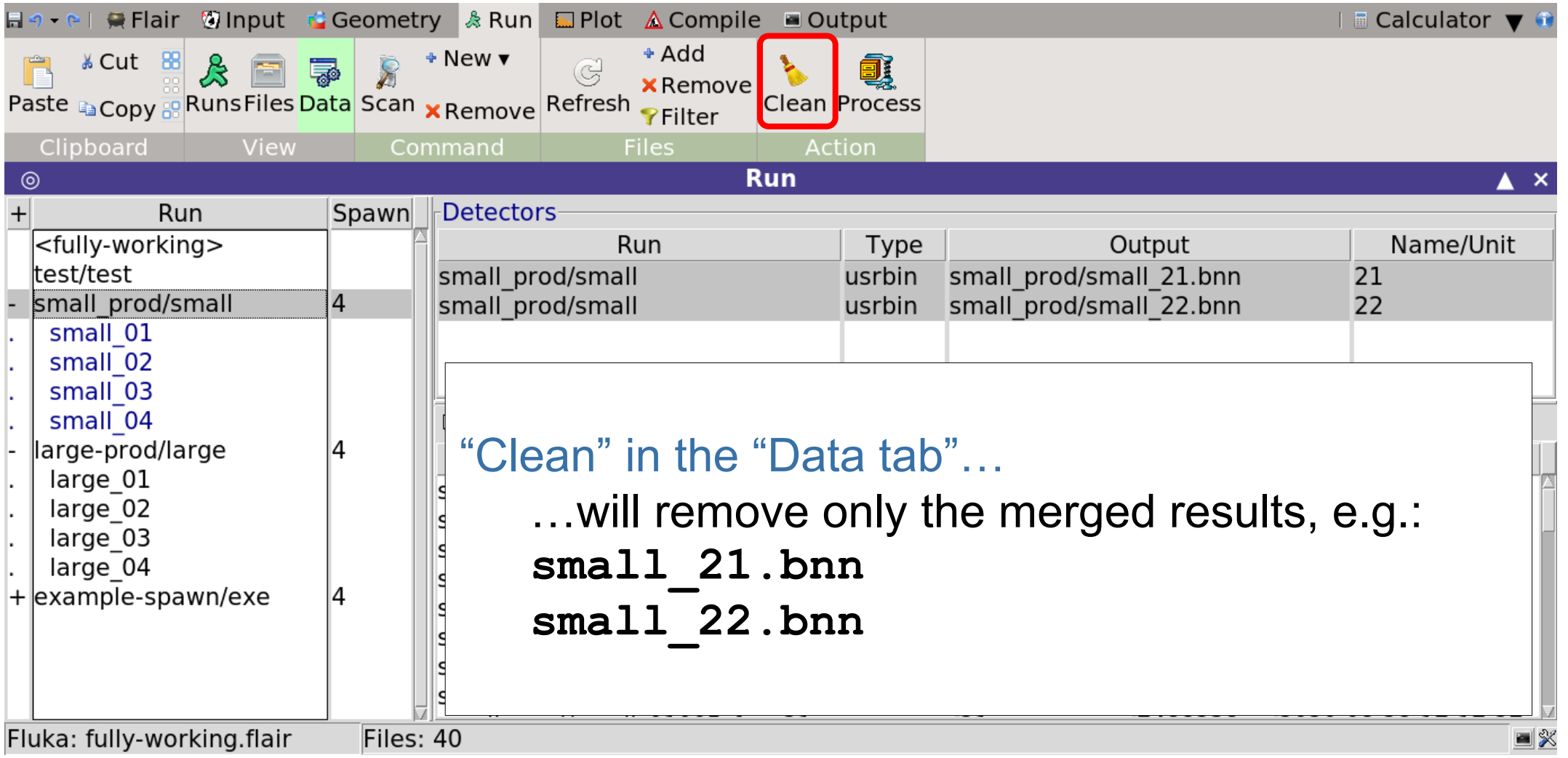
"Clean" in the "Runs tab"…

…will remove all and only the files generated within a specific run (or spawn), e.g.:

```
large_01001_fort.21      large_02001_fort.21
large_01001_fort.22      large_02001_fort.22
large_01001.err          large_02001.err
large_01001.log          …
large_01001.out          …
ranlarge_01001
```

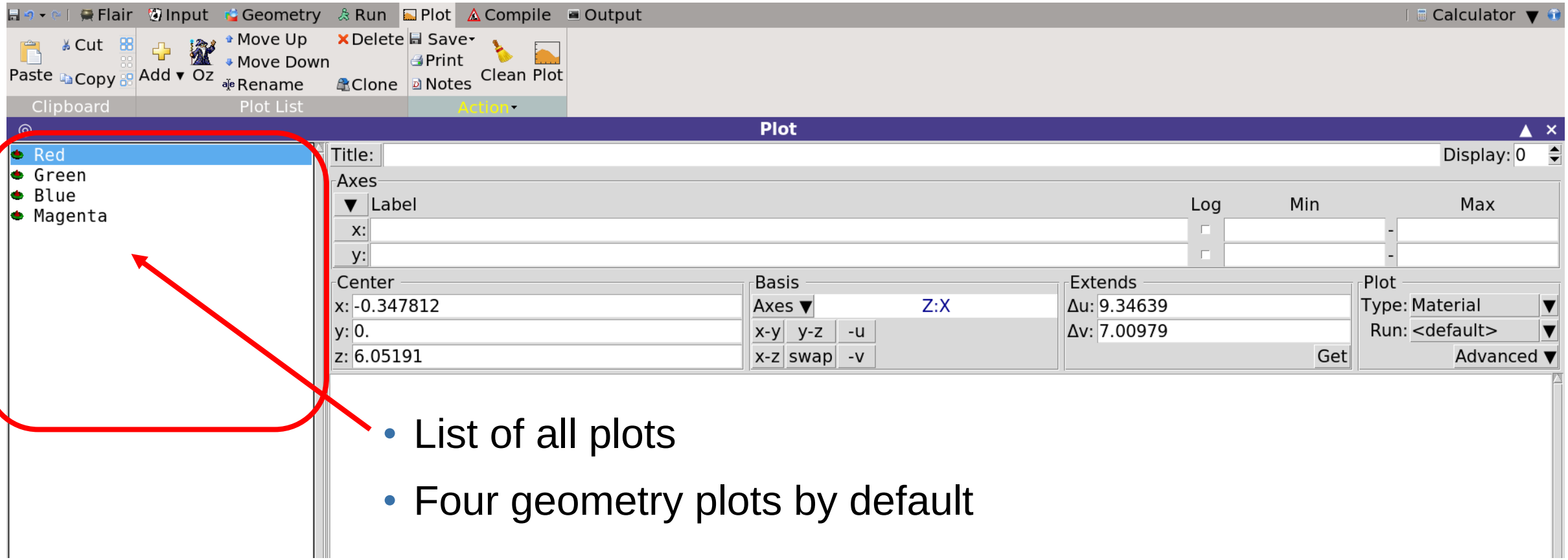
Run tab – Cleaning – 2

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



Plotting results in the Plot tab – 1

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



- List of all plots
- Four geometry plots by default