FLUKA

蘭州大學
LANZHOU UNIVERSITY

Errors and Crashes

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

To err is human

… but to really foul things up you need a computer.
Paul Ehrlich

DON’T PANIC

Overview

Sometimes the users get crashes…

Sometimes results look weird…

- Often, these are simple problems the users can address and solve on their own: the first purpose of this lecture is to help you to identify this type of errors.
- In few other cases, a real FLUKA problem might be found: the second purpose of this lecture is to help you identify this kind of situation
- Other error cases mainly deal with the sensibleness of the results and the best approach to simulations

Overview

Typical errors:

- Installation problems
- Crashes at run initialization, usually due to mistyping / wrong setting of cards in the input file
- Crashes while tracking particles (typically due to geometry)
- Problems found after the end of the simulation, usually due to subtle errors in the input file

NB: non-trivial crashes at runtime which are not explicitly related to geometry issues typically require the use of a debugger (gdb)…

Installation problems

Installation problems

- Nothing happens, or “executable not found” message in FLAIR or
- It starts, and crashes with weird errors like undefined reference to `_ZGVbN2v_atan'
  - Check you have the proper version of FLUKA for your platform and you have all packages needed for compiling and executing Fortran programs

    For LINUX or MAC distributions, you generally need:

    gfortran >8.0 (for 32 bits machines : g77 , compat-gcc-34-g77)

    check the supported versions of gfortran (gfortran –version) **and** glibc (on LINUX, **ldd –version)** **in the RELEASE-NOTES file included in the distribution**

    gfortran version and glibc version appear in the name of the distribution file: chose the right one

  - Verify your $FLUPRO environment variable is properly set (in .bashrc, .tcshrc, .cshrc…). Check also the preferences panel (config) in FLAIR.
  - If you are using gfortran, chack that you’ve either set the FLUFOR env or have the string “gfor” in the FLUPRO directory name

Installation problems

- In the .out file you get a message like this:

```
**** This version is obsolete and is  ****
**** going expire in few days.        ****
**** Please contact the FLUKA list    ****
**** or look for an updated version   ****
**** at http://www.fluka.org          ****
```

- update your version of FLUKA
- “make” again
- re-link your executable

Starting a run

If you see a message a la:

ERROR: Running in the FLUKA folder XXXX is a very bad idea

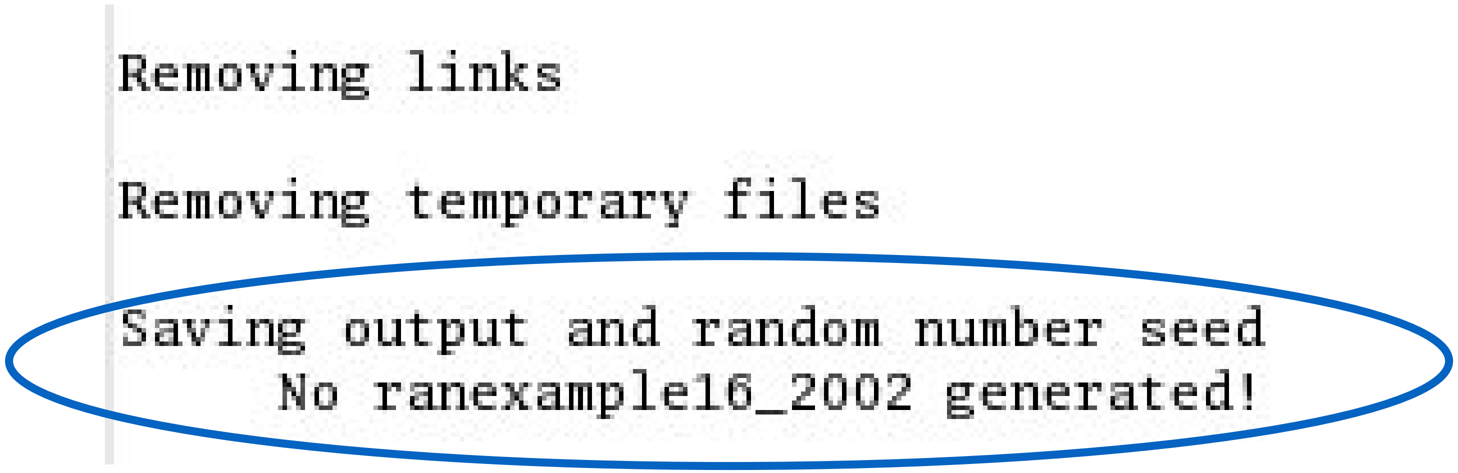
.... it means you are running a FLUKA run in $FLUPRO, something which must be avoided

DO NOT RUN inside the FLUKA folder!!

Crashes at run initialization

Crashes at run initialization: general

The following message appears on the terminal (or in the *inpname*.out file then using Flair) and no results are produced:



```
Removing links

Removing temporary files

Saving output and random number seed
     No ranexample16_2002 generated!
```

Look at the BEGINNING of the .log and at the END of the .out/.err files and check for Messages. These files can be located in:

- your working directory
- temporary subdir *fluka_#####*
- Output Files window in FLAIR

Crashes at initialization General:

- Always launch only 1 cycle when setting up a run. If not..at least:
- Always check the output from the **FIRST** cycle of a run.
- If not, you could be misled by a message like this:

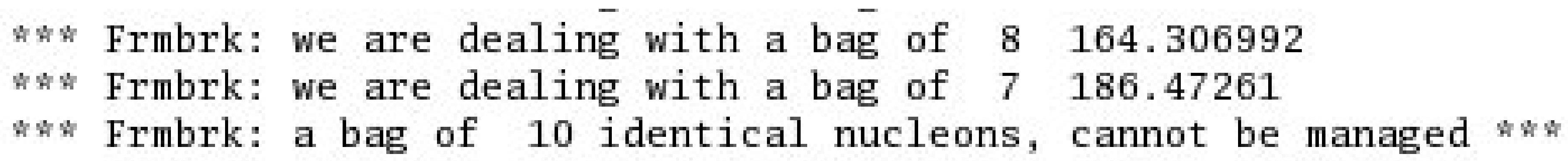
```
**** No Random file available !!!!!! ****
   Abort called from FLRM64 reason NO
RANDOM FILE Run stopped!
   STOP NO RANDOM FILE
```

- If the first cycle crashes, it will not write the random seed fior the second one, and so on: all the others will crash with the message above
- The REAL error message will be in the first output

The *.err file: not only errors

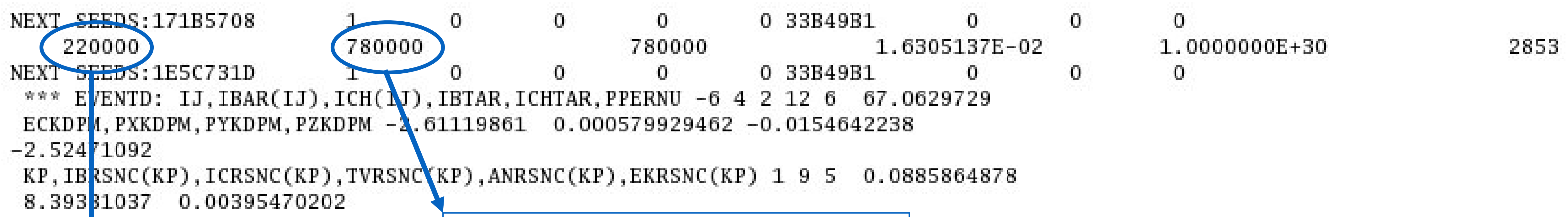
The *.err file will report errors, but also a lot of warning messages which have a meaning mostly for the developers.

For instance: the following messages are not errors!



```
*** Frmbrk: we are dealing with a bag of  8  164.306992
*** Frmbrk: we are dealing with a bag of  7  186.47261
*** Frmbrk: a bag of  10 identical nucleons, cannot be managed ***
```

Additional useful information is written to the .err file:



```
NEXT SEEDS:171B5708        1        0        0        0        0 33B49B1        0        0        0
     220000                780000                    780000                  1.6305137E-02              1.0000000E+30                    2853
NEXT SEEDS:1E5C731D        1        0        0        0        0 33B49B1        0        0        0
  *** EVENTD: IJ,IBAR(IJ),ICH(IJ),IBTAR,ICHTAR,PPERNU -6 4 2 12 6  67.0629729
 ECKDPM,PXKDPM,PYKDPM,PZKDPM -2.61119861  0.000579929462 -0.0154642238
-2.52471092
 KP,IBRSNC(KP),ICRSNC(KP),TVRSNC(KP),ANRSNC(KP),EKRSNC(KP) 1 9 5  0.0885864878
 8.39331037  0.00395470202
```

No. of events simulated so far

No. of events remaining to be simulated

Formatting errors in the .inp file

This type of error was very common before the development of FLAIR

- The offending card is reported at the end of the <myinput>001.out: for instance, if the name “BERYLLIUM” is not fully contained in its 10-character field:

```
*** Unable to resolve name element 0  BERYLLI in card ***
EMFCUT       -0.010     0.010       1.0  BERYLLIU                     PROD-CUT
```

- In case the problem is located in the geometry declaration, please have a look also at the fort.16 file in the fluka_##### temporary subdirectory
- Some non visible control characters (e.g. tab, \t, \n, ^M) may inadvertently be present in the input file. You can clean your input using text editors, perl or “dos2unix” Linux command.
- Remember that tabs and empty lines are not admitted in the geometry:, and flair does not give any error message, fluka does.

```
Abort called from FLKCGI reason NULL BODY NAME Run stopped!
 STOP NULL BODY NAME
```

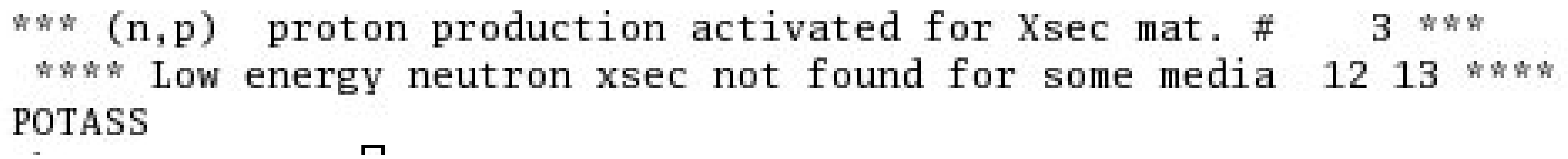
- Remember that the total length of a line in FREE format is 132 characters

Low-energy neutron cross section not found

- FLUKA associates low-energy neutron cross section on a MATERIAL name basis.

→ Open manual, table 10.4.1.2

- Should your material name be different from the material name in the table, the association fails and an error is issued:



```
*** (n,p)  proton production activated for Xsec mat. #    3 ***
 **** Low energy neutron xsec not found for some media  12 13 ****
POTASS
```

Either change the name, or issue a LOW-MAT card!

Heavy Ions

- **The binary (flukahp) obtained with the default linking script (lfluka) does not include the interaction generators for heavy ions above 100-150 MeV/n.**

- To include them, a dedicated binary has to be linked with ldpmqmd

- You can verify whether the relevant event generators are linked by looking for initialization message like “initializing RQMD” and “initializing DPMJET” in the .out file.

Heavy Ions -II

- The code stops if rqmd/DPMJET are not linked and
  - The beam is an Heavy Ion with energy larget than the BME upper limit (remember what?)
  - AND/OR   Coalescence is activated and the beam energy is large (> 5 GeV/c) WHY?
    - Because coalescence can produce heavy ions. They will have to be transported in a physically reasonable way

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!          ATTENTION PLEASE            !!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 Coalescence activated and beam above BME  limit with no
rQMD
 and no or incompatible IONSPLIT option!
 Execution terminated.
```

- If you really do not want to link rqmd/dpmjet, (why??) IONSPLIT can help you

Residual Nuclei

- Residual nuclei scoring and/or radioactive decays require the best low energy nuclear physics: activate
- PHYSICS with SDUM= EVAPORAT and WHAT(1)=3 , otherwise:

```
' *** Predictions for residual nuclei production and    ***',
   ' *** decays require the activation of heavy fragment   ***',
  ' *** evaporation by means of the PHYSICS/EVAPORAT card ***',
  ' *** look at the manual and release notes for further  ***',
  ' *** details                                           ***'
     WRITE ( LUNERR, '(/,A,/,A,/,A,/,A,/,A)' )
  ' *** Predictions for residual nuclei production and    ***',
  ' *** decays require the activation of heavy fragment   ***',
  ' *** evaporation by means of the PHYSICS/EVAPORAT card ***',
  ' *** look at the manual and release notes for further  ***',
  ' *** details                                           ***'
' *** Execution terminated ***'
```

USRBIN/EVENTBIN scoring definitions errors

- USRBIN scoring method:
  - **WHAT(1)>=10** Track-length quantities, i.e. they can be distributed along a track (fluence, energy deposition…)
  - **WHAT(1)<10** Point-wise quantities, i.e. they have to be scored on a point, or in the middle of the step (activity, fission, neutron balance…)

  *** Activity/fission/neutron balance binnings cannot be track-length!!!

- Badly defined USRBIN scoring:

```
******* Fluka stopped in Usrbin: "usr/eventbin" n.    1 *******
******* with zero width   0.000     for axis  R  ******
```

- Output units:
  - Never use unit# <20 (reserved for FLUKA internal use) or >99 (Fortran77 limitation)
  - Never mix the output of different scoring cards in the same unit

Flair checks all these requirements!

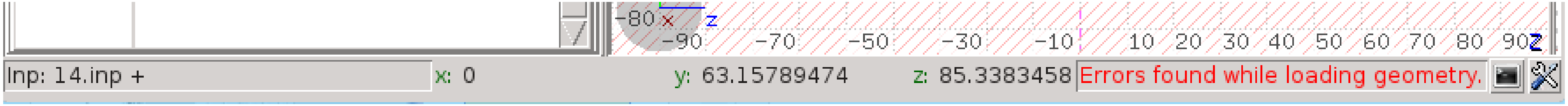
Geometry Errors

- The Geometry Editor allows to spot geometry errors when the .inp file is parsed/updated (recall dedicated lecture!)

- A warning message in the Geometry status bar notifies the user about the presence of geometry errors

- A warning is also issued for not strictly geometrical errors

  (e.g.: missing material assignment to a region, unknown card…)



Inp: 14.inp +

x: 0

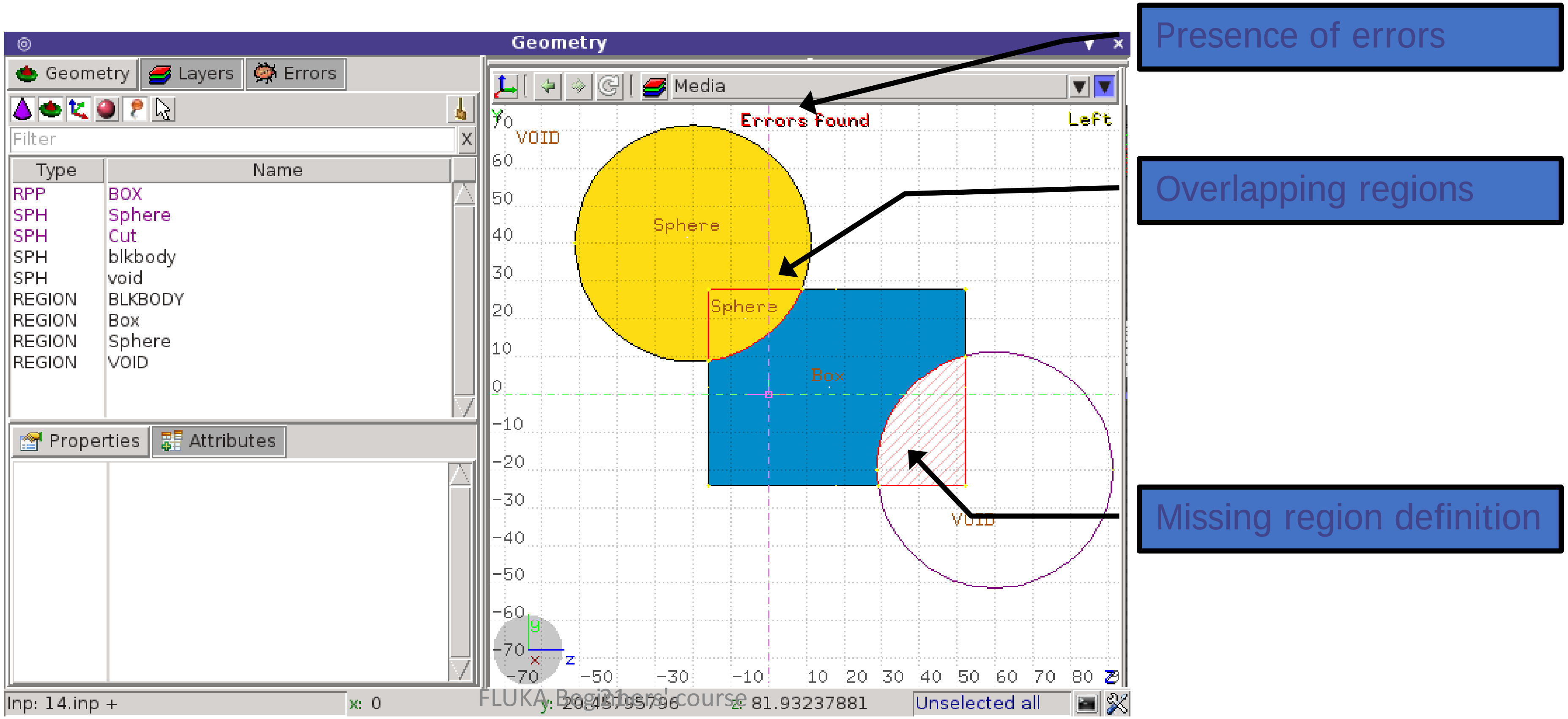
y: 63.15789474

z: 85.3383458

Errors found while loading geometry.

Geometry Errors

- Affected areas are surrounded by red lines:
  - Areas filled with a full color correspond to overlapping regions
  - Areas filled with red lines correspond to a missing region definition
- Clicking the 🐞rrors icon displays the dialog box with the errors
- Touching surfaces are checked against 10 significant digits



Type	Name
RPP	BOX
SPH	Sphere
SPH	Cut
SPH	blkbody
SPH	void
REGION	BLKBODY
REGION	Box
REGION	Sphere
REGION	VOID

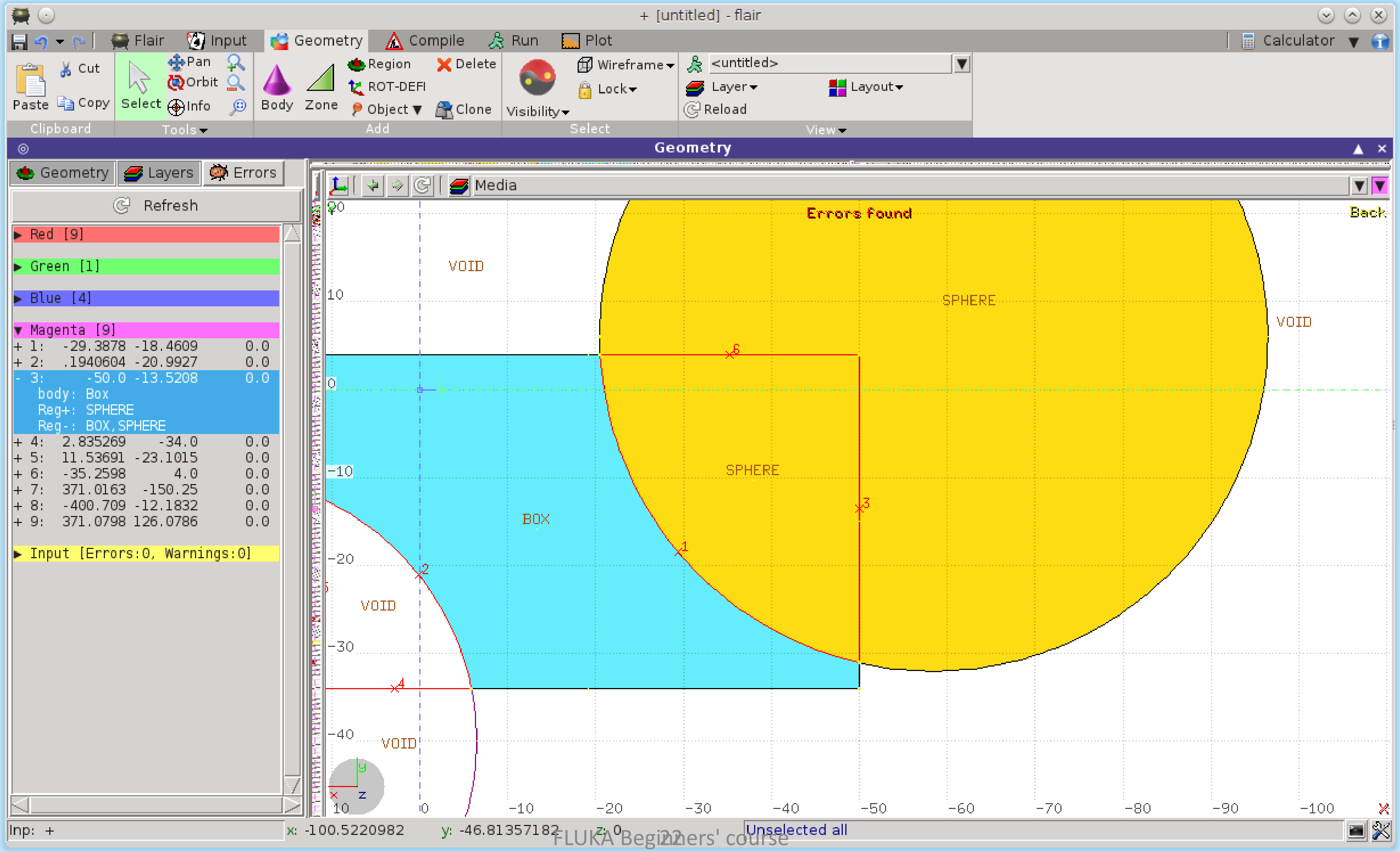
Inp: 14.inp + x: 0 y: 20.45795796 z: 81.93237881 Unselected all

Presence of errors

Overlapping regions

Missing region definition

Geometry Errors



```
▶ Red [9]

▶ Green [1]

▶ Blue [4]

▼ Magenta [9]
+ 1:  -29.3878 -18.4609      0.0
+ 2:  .1940604 -20.9927      0.0
- 3:     -50.0 -13.5208      0.0
    body: Box
    Reg+: SPHERE
    Reg-: BOX,SPHERE
+ 4:  2.835269    -34.0      0.0
+ 5:  11.53691 -23.1015      0.0
+ 6:  -35.2598      4.0      0.0
+ 7:  371.0163  -150.25      0.0
+ 8:  -400.709 -12.1832      0.0
+ 9:  371.0798 126.0786      0.0

▶ Input [Errors:0, Warnings:0]
```

```
Inp: +      x: -100.5220982     y: -46.81357182     z: 0     Unselected all
```

Crashes during tracking

Crashes during tracking: geometry

- Typical message due to errors in geometry

```
 Abort called from FLKAG1 reason TOO MANY ERRORS IN GEOMETRY Run stopped!
  STOP TOO MANY ERRORS IN GEOMETRY
```

Did you debug your geometry????????

- Look in the .err/.out files to better direct your debugging

- Consider also the use of the RAY particle to trace and analyze your geometry. WARNING: this is an advanced topic not covered in this course (see the Manual)

Crashes during tracking: geometry

```
Geofar: Particle in region 3 (cell # 0) in position
1.000000000E+00 0.000000000E+00 1.000000000E+00 is now causing
trouble, requesting a step of 6.258867675E-07 cm to direction -
2.285059979E-01 -9.412338141E-01 2.487245789E-01, error count:
0 [...skipped...] Particle index 3 total energy 5.189748600E-04
GeV Nsurf 0 We succeeded in saving the particle: current region
is n. 2 (cell # 0)
```

- GEOFAR errors (.out/.err file) during tracking, point to:
  - errors in the geometry
  - numerical precision errors
- FLUKA tries to save problematic particles, but will stop after too many errors of this kind.
- The problematic point is given by position + step * direction
  (LATTICEs are VERY sensitive to numerical precision: use as many digits as possible to ensure a proper description of the lattice, the related transformation and the prototype)

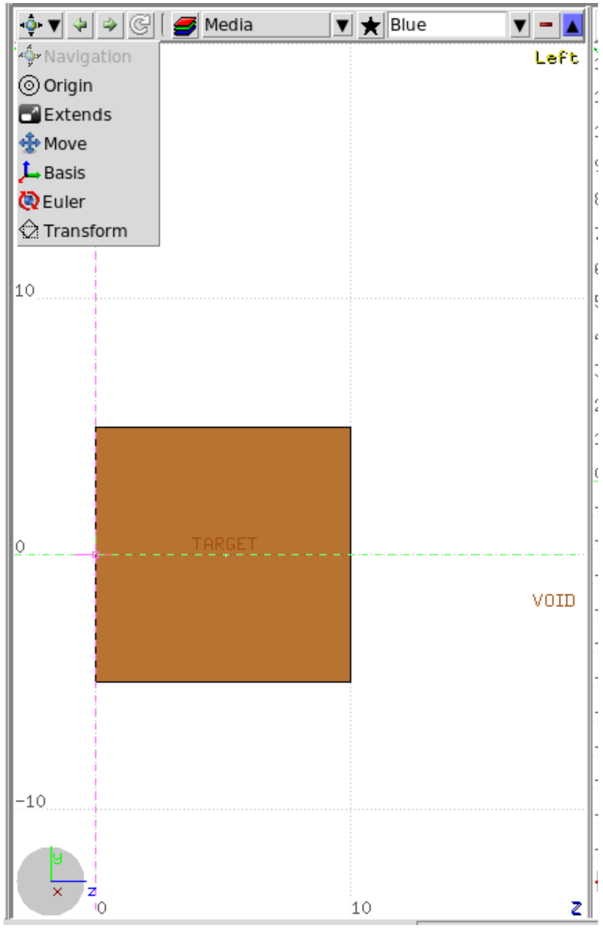
Tip: To find the position in the geometry viewer

- Either you do the math: position + step * direction

.... and navigate to that position in Flair

- Alternatively, in Flair's geometry editor:
  - Open the projection dialog [o]
  - Origin: Enter the X,Y,Z coordinates of the position
  - Basis: Enter as U vector the direction
  - Move: enter as +U: the step

  And you will end up in the error location



It runs, but…

…something seems wrong…

…you do not understand the results

LOOK in the output file!

what?

everything, of course!

a few examples follow, you’ll find out more on your own

RANDOMIZ card

- You run several cycles and the statistical errors appears to be ZERO

Most probably you are starting the same histories on every cycle

This is a typical error if the RANDOMIZe card does not exist!

Verify the existence of RANDOMIZe card and that WHAT(1)=1.0

- You get the following message in the *.out file

```
**** No Random file available !!!!!! ****
Abort called from FLRM64 reason NO RANDOM FILE Run stopped!
STOP NO RANDOM FILE
```

Most probably the error occurred during the previous cycle which didn’t generate random number seed for the current cycle, whereupon the program halts.

Cards defining the beam

The total/partial energy balance does not meet expectations:

- Remember that for HEAVY IONS the kinetic Energy or Momentum in the BEAM card is given PER NUCLEON, while in all other cards the energy is the total kinetic one: check the numbers in the beam section of the .out file
- Check if you selected Energy or Momentum in BEAM
- Verify that your primary source BEAMPOS is **NOT defined on a surface boundary**, but always inside a region
- Check at the end of the .out file the energy balance per region and per particle.

MATERIAL / COMPOUND cards

The total/partial energy balance does not meet your expectations or particle fluences show discontinuities

Check:

- the density is defined for all MATERIALs
- Z and A are given in case of a specific isotope
- mixing fractions (and signs!) in COMPOUND cards
- Check syntax using Flair
- Look in the Fluka output file, where the properties of all defined material/compounds are written

Geometry induced weird results

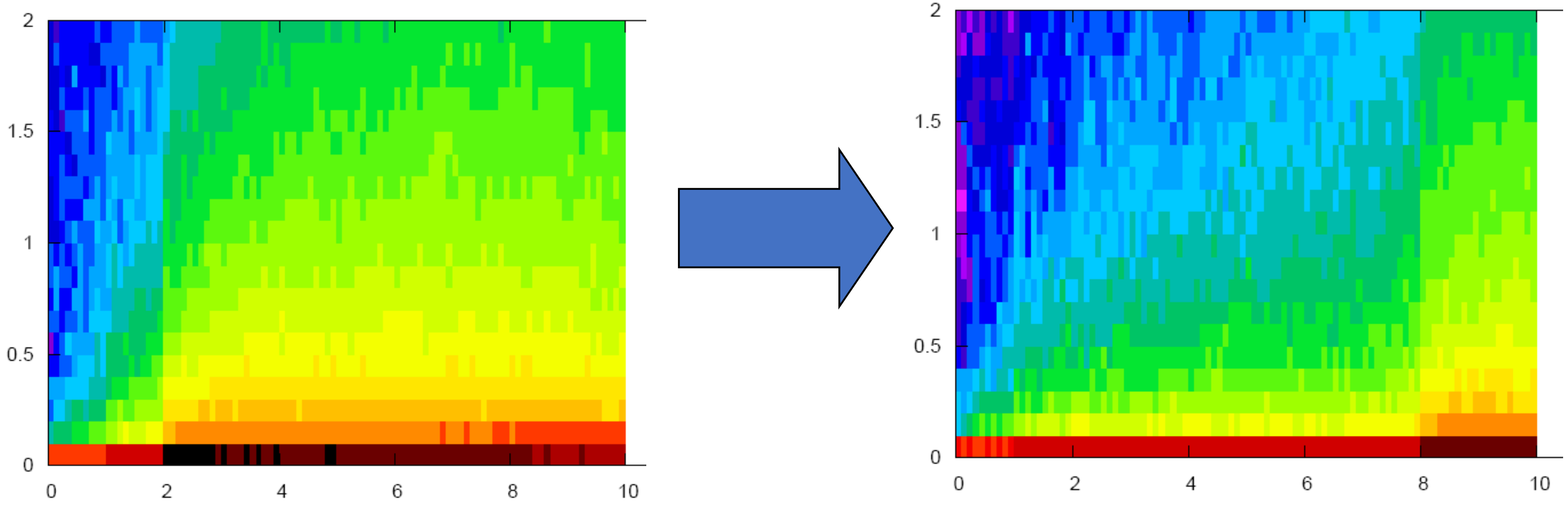
Possible symptoms:

- the energy/fluence among different regions is not as expected
- problems with boundary-crossing estimators (low fluences, empty data…)

Did you debug your geometry????????

Overlapping regions are NOT detected by FLUKA at initialization, and do NOT cause run-time errors.

Example: overlapping cylinders



ZERO result??

I have ZERO Energy deposition/ Fluence/ Dose/ Activity in some portion of space / energy range/ particle type

This is probably not true: check:

- Thresholds in the fluka .out file
  - EMF production thresholds are listed by MATERIAL and called Ae (e^{+-}) and Ap (γ)
  - EMF transport thresholds by REGION Ecut and Pcut,
  - other particles under "Particle transport thresholds"
- Materials and their assignment to regions in the fluka .out file
- Do I have enough primaries? Look at neighboring areas/energies, look at statistical errors (in the _tab.lis files or with flair)
- Do I need biasing? As above
- Did I switch on the relevant physics process?
  - No neutrons from high energy electrons/photons→ PHOTONUC
  - No γ lines or α particles in neutron transport → LOW-PWXS

  …

Preprocessor

Preprocessor directives (see basic input lecture) are very powerful, but pay attention to nesting:

- every #if directive must have its closing #endif directive;
- Up to 10 nested levels are allowed;
- Up to 40 variables can be defined;

Merging cycles from different jobs

- Verify that you merge cycles for which the .inp files differ only by the RANDOMIZ card.

- It is a good habit to remove files from previous runs before starting a new one:

    - Flair offers this possibility to “Clean” the generated files from each process: Hit the button “Clean” before a run
    - It’s good to develop the habit to remove the output files **from test runs**

- If you are running on many CPU’s in parallel, check that the initial random seeds are different

FLUKA users and the manual

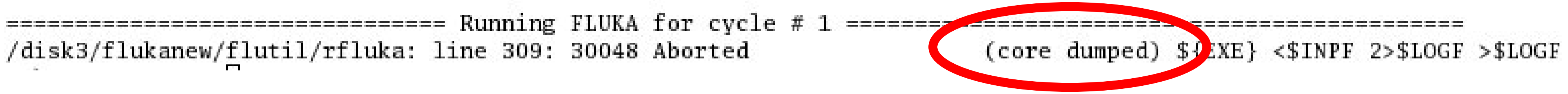
The identification of the solution to all the problems shown so far, imply that a FLUKA user lives symbiotically with the Manual

And that he always checks the fluka output file!!

Now
PANIC

Run time errors with exceptions

You get a message on terminal like:



```
================================ Running FLUKA for cycle # 1 =============================================
/disk3/flukanew/flutil/rfluka: line 309: 30048 Aborted                  (core dumped) ${EXE} <$INPF 2>$LOGF >$LOGF
```

The temporary fluka_##### subdir remain, with the *.log, *.out, *.err files, a core.* file and the last random seed

The last random seed allows to restart the run from the configuration occurring at maximum 5 minutes of CPU-time before the error!

Run time errors with exceptions

Suggestions:

1. Check the end of *.out / *.err files or the beginning of the *.log file:
   - they may contain important information for you or for the experts
   - the code has many internal checks, and some error conditions are recorded

2. Maybe there is a very well hidden geometry problem

Reading errors

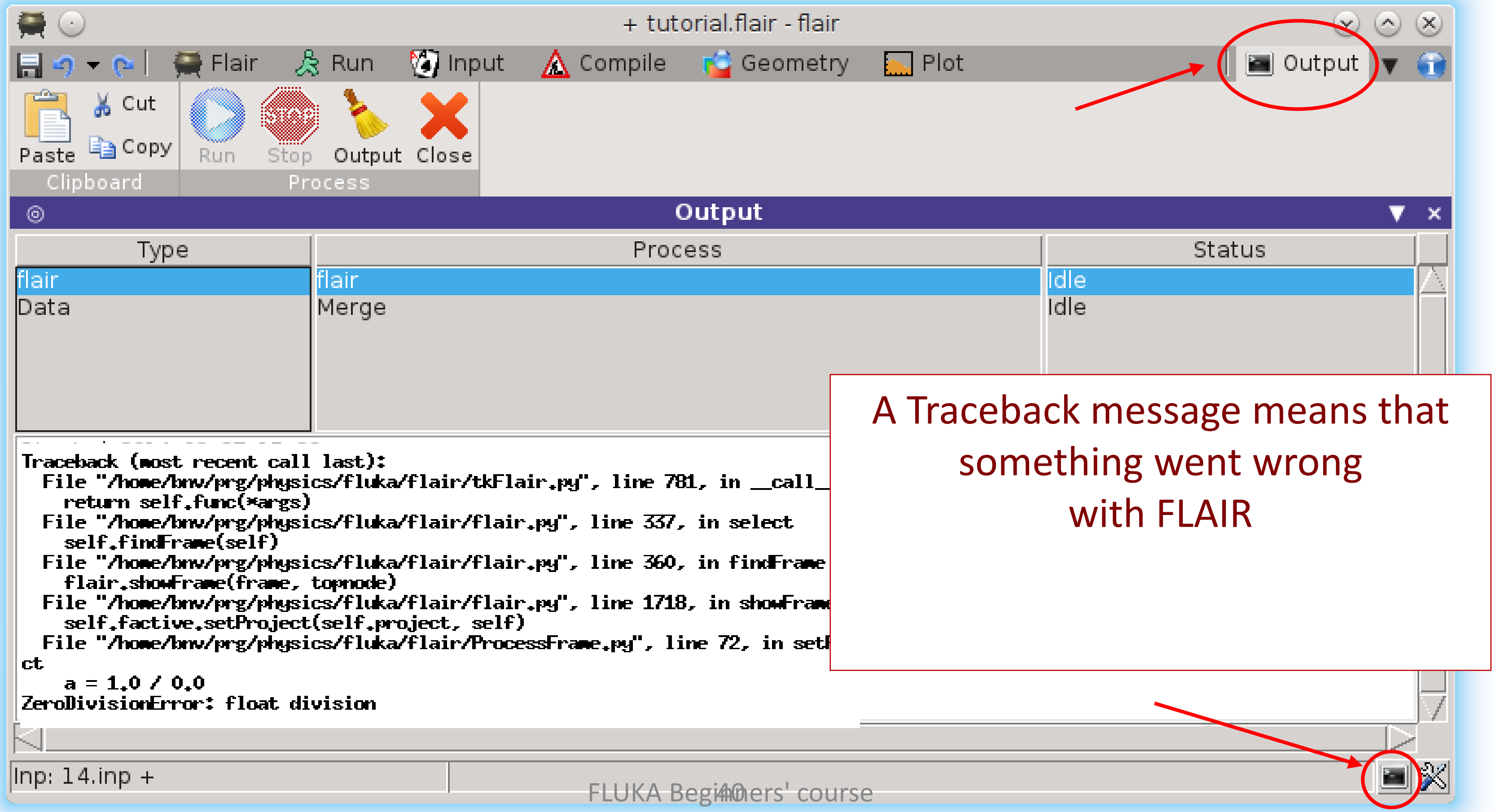
- On Unit 14: your nuclear.bin file might be corrupted or missing (check all your binary files in $FLUPRO) or doesn’t match the FLUKA distribution

- On Unit 1: leads to a problem with the random seed.

  It typically happens when you start the simulation with a rfluka command with option –N n with n>0: check for the existence of ran***n in the directory where you issue rfluka!

Flair output window

- The output window of FLAIR contains all messages from FLAIR, FLUKA, processing, and plotting tools
- Always consult it in case of problem or doubts



Type	Process	Status
flair	flair	Idle
Data	Merge	Idle

```
Traceback (most recent call last):
  File "/home/bnv/prg/physics/fluka/flair/tkFlair.py", line 781, in __call_
    return self.func(*args)
  File "/home/bnv/prg/physics/fluka/flair/flair.py", line 337, in select
    self.findFrame(self)
  File "/home/bnv/prg/physics/fluka/flair/flair.py", line 360, in findFrame
    flair.showFrame(frame, topnode)
  File "/home/bnv/prg/physics/fluka/flair/flair.py", line 1718, in showFram
    self.factive.setProject(self.project, self)
  File "/home/bnv/prg/physics/fluka/flair/ProcessFrame.py", line 72, in setP
ct
    a = 1.0 / 0.0
ZeroDivisionError: float division
```

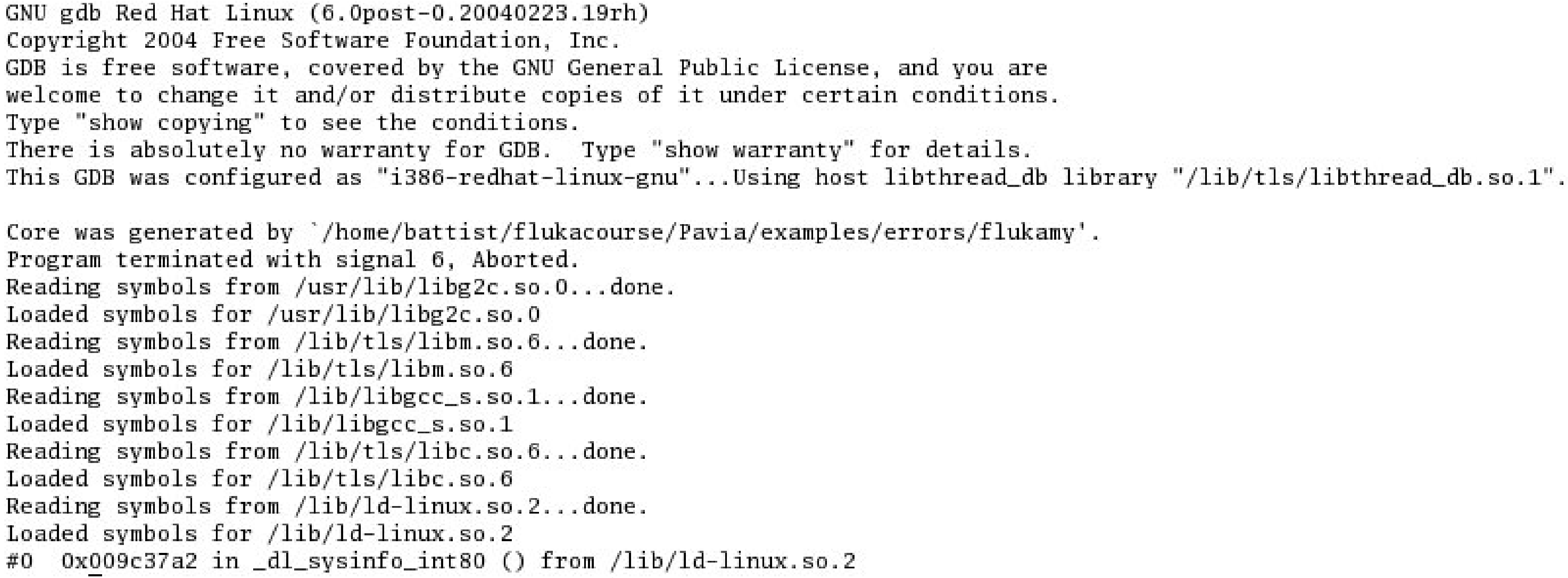
A Traceback message means that something went wrong with FLAIR

gdb: How To

Advanced Topic:
The GNU debugger (gdb) can help in spotting the problem

How to run it:

- from terminal:
  ```
  cd fluka_####
  gdb $FLUPRO/flukahp core.*
  ```
- from FLAIR:
  double click on the core file from the "Output Files" Frame;



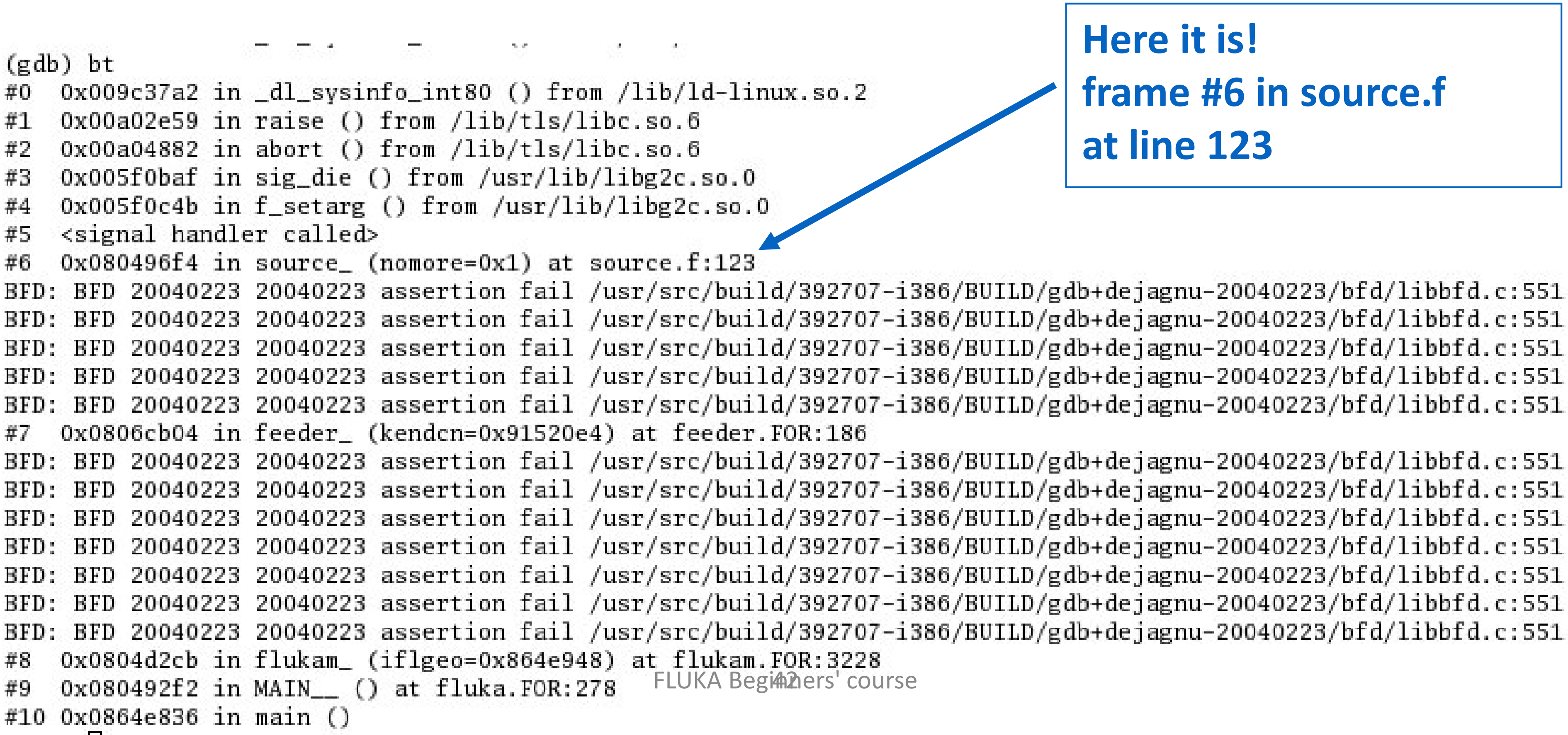
```
GNU gdb Red Hat Linux (6.0post-0.20040223.19rh)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux-gnu"...Using host libthread_db library "/lib/tls/libthread_db.so.1".

Core was generated by `/home/battist/flukacourse/Pavia/examples/errors/flukamy'.
Program terminated with signal 6, Aborted.
Reading symbols from /usr/lib/libg2c.so.0...done.
Loaded symbols for /usr/lib/libg2c.so.0
Reading symbols from /lib/tls/libm.so.6...done.
Loaded symbols for /lib/tls/libm.so.6
Reading symbols from /lib/libgcc_s.so.1...done.
Loaded symbols for /lib/libgcc_s.so.1
Reading symbols from /lib/tls/libc.so.6...done.
Loaded symbols for /lib/tls/libc.so.6
Reading symbols from /lib/ld-linux.so.2...done.
Loaded symbols for /lib/ld-linux.so.2
#0  0x009c37a2 in _dl_sysinfo_int80 () from /lib/ld-linux.so.2
```

gdb: How To

Basic commands:

- **bt**, to list the functions which led to the current one and the crash
- **f**, to switch to a given frame (i.e. interrupted function)
- **p**, to print the value of a given variable



```
(gdb) bt
#0  0x009c37a2 in _dl_sysinfo_int80 () from /lib/ld-linux.so.2
#1  0x00a02e59 in raise () from /lib/tls/libc.so.6
#2  0x00a04882 in abort () from /lib/tls/libc.so.6
#3  0x005f0baf in sig_die () from /usr/lib/libg2c.so.0
#4  0x005f0c4b in f_setarg () from /usr/lib/libg2c.so.0
#5  <signal handler called>
#6  0x080496f4 in source_ (nomore=0x1) at source.f:123
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
#7  0x0806cb04 in feeder_ (kendcn=0x91520e4) at feeder.FOR:186
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
BFD: BFD 20040223 20040223 assertion fail /usr/src/build/392707-i386/BUILD/gdb+dejagnu-20040223/bfd/libbfd.c:551
#8  0x0804d2cb in flukam_ (iflgeo=0x864e948) at flukam.FOR:3228
#9  0x080492f2 in MAIN__ () at fluka.FOR:278
#10 0x0864e836 in main ()
```

Here it is!
frame #6 in source.f
at line 123

FLUKA bug?

If your crash does not fall in any of the previous categories, and you do not understand what happens, prepare a report and send it to

fluka-discuss@fluka.org together with a tar file containing:

- .inp, .out, .log, .err files
- additional auxiliary files (if any)
- last random seed (ran* file in fluka_####)
- and any other possible useful information

Advanced users may also add the gdb results

Summary

- In case of technical problems, your best allies are:
  1. the FLUKA manual
  2. error messages at the end of .out/.err files, and beginning of .log file

- In case of doubts on the results:
  1. read the FAQ of FLUKA: http://www.fluka.org/fluka.php?id=faq&mm2=3
  2. search for a similar problem in the FLUKA discussion list
  3. if you really cannot understand the issue, or if you need to ask about physics related problems, write to:

     fluka-discuss@fluka.org

SPARES

Heavy Ions

- **The binary (flukahp) obtained with the default linking script (lfluka) does not include the interaction generators for heavy ions above 100-150 MeV/n.**

- To include them, a dedicated binary has to be linked with ldpmqmd

- You can verify whether the relevant event generators are linked by looking for initialization message like “initializing RQMD” and “initializing DPMJET” in the .out file.

The FUDGEM parameter

In EMFCUT with SDUM=PROD-CUT, WHAT(3) accounts for the fraction of atomic electrons contributing to MCS.

In Rutherford xs: $Z^2 \rightarrow Z(Z+f)$ f=WHAT(3)

A value of 10^{-5} is fine for low delta-ray production thresholds: the contribution of atomic electrons is accounted for via ionization losses. Low means much lower than typical atomic shell binding energies (several 10s of keV).

For higher delta-ray production thresholds, if WHAT(3)<<1, there would be a fraction of atomic electrons not contributing to scattering. WHAT(3) should be set to 1.

```
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
EMFCUT     ElePosiTh   WHAT(2)   WHAT(3)      Mat1      Mat2      StepPROD-CUT
```

A finite value should be entered for WHAT(3), otherwise:
*** Atomic electron contribution to MCS for material …. Set to zero.
Are you sure?****

USRBIN/EVENTBIN scoring definitions errors

- USRBIN scoring method:
  - **WHAT(1)>=10** Track-length quantities, i.e. they can be distributed along a track (fluence, energy deposition…)
  - **WHAT(1)<10** Point-wise quantities, i.e. they have to be scored on a point, or in the middle of the step (activity, fission, neutron balance…)

  *** Activity/fission/neutron balance binnings cannot be track-length!!!

- Badly defined USRBIN scoring:

```
******* Fluka stopped in Usrbin: "usr/eventbin" n.    1 *******
******* with zero width   0.000     for axis  R  ******
```

- Output units:
  - Never use unit# <20 (reserved for FLUKA internal use) or >99 (Fortran77 limitation)
  - Never mix the output of different scoring cards in the same unit

Flair checks all these requirements!

Geometry: parentheses expansion

At the initialization of the geometry, you may get this kind of message in the first line of .log file:

```
Subscript out of range on file line 56, procedure rpnorm.f/rpnorm
Attempt to access the 114705-th element of variable tx
```

You have exceeded the maximum expansion limit for parenthesis, set to 100000

WARNING even with a simple region expression this limit can be reached, especially if you are using infinite bodies!

Re-write the geometry in such a way to diminish the
Number of parenthesis e.g. using the OR (pipe) to make the union of different
zones