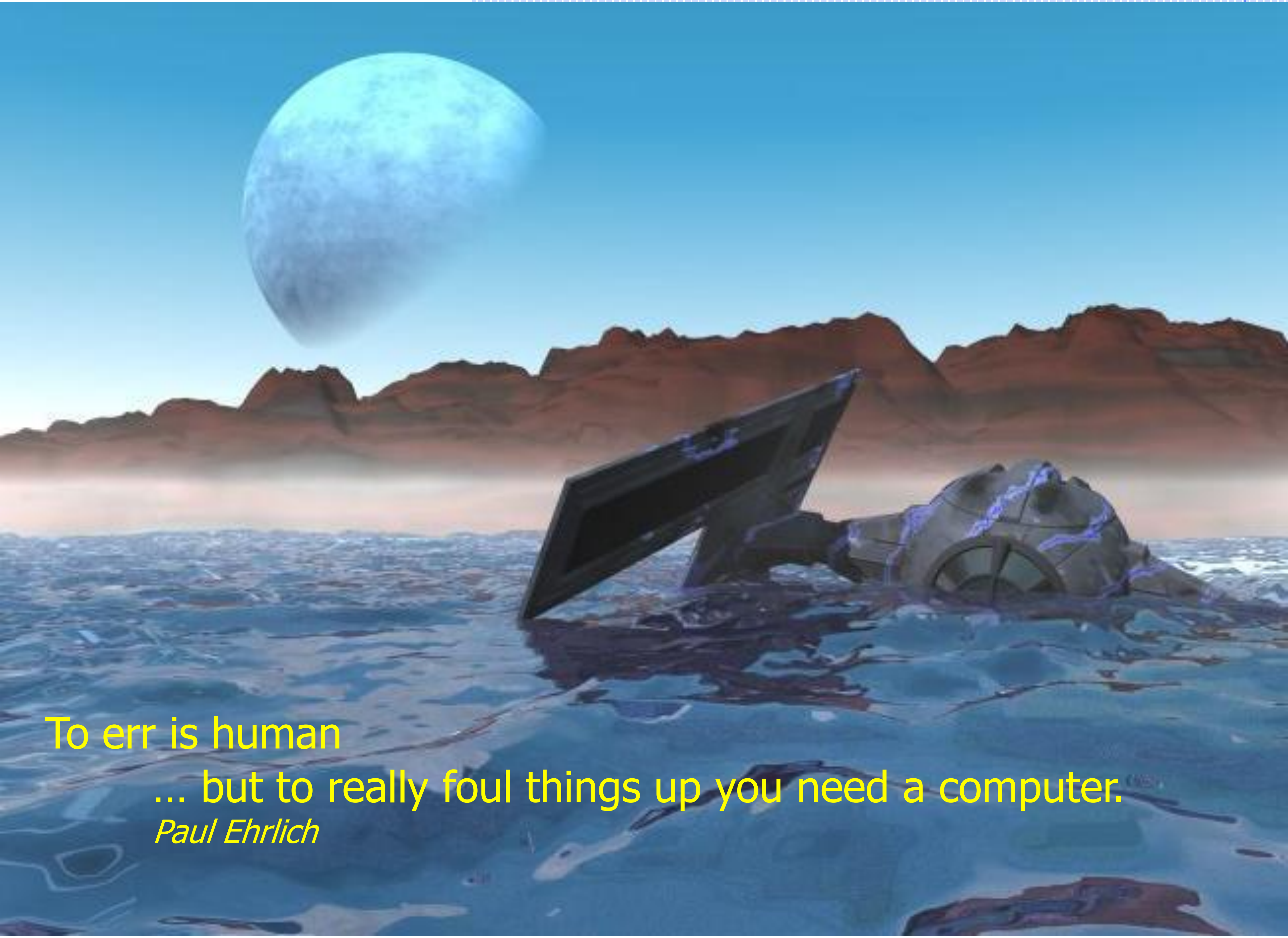
# Handling of errors and crashes

Beginners’ FLUKA Course



To err is human

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

*Paul Ehrlich*

# DON’T PANIC

# Overview [1/2]

Sometimes the users get crashes…

Sometimes results look weird…

- These are often simple problems that the user can address and solve on their own: the first purpose of this lecture is to help you to spot this type of error.
- In a few other cases, a real FLUKA problem is found, and the second purpose of this lecture is to help you to spot this type of error.
- The other cases deal mainly with the sensibleness of results and the best approach to simulations: this has nothing to do with the present lecture...

# Overview [2/2]

Typical (trivial?) errors:

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

NB: (NOT trivial) crashes at runtime NOT explicitly related to geometry issues usually need the use of gdb…

# Installation problems

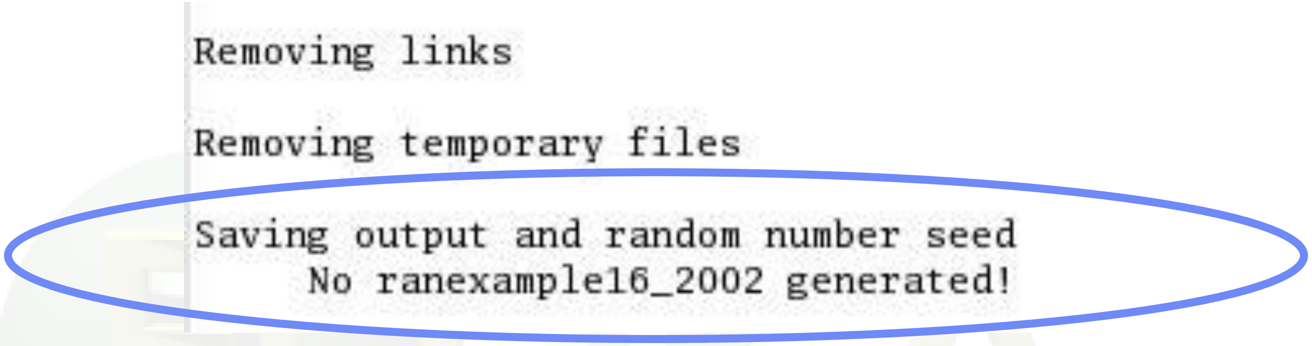
- Nothing happens, or “executable not found” message in FLAIR
  - Check that you have the correct version of FLUKA for your platform and you have all packages needed for compiling and executing fortran programs. Generally, for LINUX distributions, you need:
    - g77 or compat-gcc-34-g77 **64-bit machines: check FAQ**
    - libg2c
  - Verify your $FLUPRO environment variable, then issue a make command in the $FLUPRO directory: all the FLUKA tools and the default executable $FLUPRO/flukahp will be compiled;
  - The $FLUPRO variable must always be set (in .bashrc, .tcshrc, .cshrc…);
- You get the following message (.out file):

```
**** This version is obsolete and/or  ****
**** you are not allowed to use FLUKA ****
**** Please contact Alfredo Ferrari   ****
**** CERN-AB/ATB, tel.+41-22-76-76119 ****
**** or look for an updated version   ****
**** at http://www.fluka.org          ****
```

Update your version of FLUKA and “make” again; re-link in case of user routines.

# Crashes at run initialization: general

- The following message appears on terminal (or in the nohup.out file when running with 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 file and/or at the end of the .out file:
  - either in your working dir;
  - or in the temporary subdir fluka_####;
  - or in the Output Files window in FLAIR;

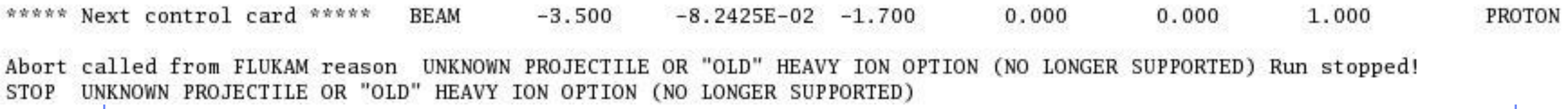
  and check the error message;

# Wrong formatting in the .inp file

- Quite easy case: FLUKA echoes all the parsed lines of the *.inp file in the *.out file. In case of problems, the echo stops at the last card correctly interpreted;
- FLUKA reads the input file in many passages. Therefore, the order of the echoed cards does not necessarily correspond to the order in the .inp file!
- In case the problem is located in the geometry declaration, please have a look also at the fort.16 file in the fluka_#### temp subdir;
- WARNING sometimes non visible control characters may appear in a file sent via e-mail (NOT for ALL mail clients): in this case, the "dos2unix" command (specific rpm in LINUX) can remove them (or use simple perl or editor commands);
- Remember to *always* check the first cycle you requested (e.g. *_001.out);

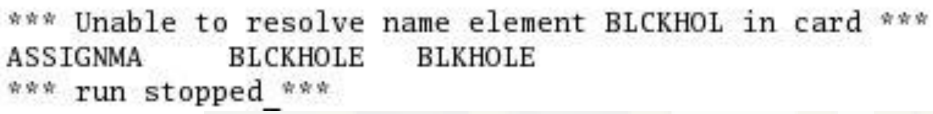
- This type of errors were very common before the advent of FLAIR;
- Keep in mind that the total length of a line in FREE format is 132 characters;

# Examples [1/2]



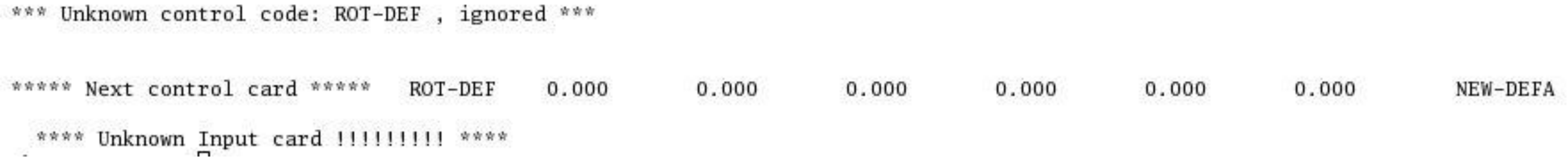
```
***** Next control card *****   BEAM      -3.500     -8.2425E-02  -1.700       0.000       0.000       1.000         PROTON

Abort called from FLUKAM reason  UNKNOWN PROJECTILE OR "OLD" HEAVY ION OPTION (NO LONGER SUPPORTED) Run stopped!
STOP  UNKNOWN PROJECTILE OR "OLD" HEAVY ION OPTION (NO LONGER SUPPORTED)
```

.inp file in fixed format: the SDUM identifying the particle is not properly aligned



```
*** Unable to resolve name element BLCKHOL in card ***
ASSIGNMA      BLCKHOLE    BLKHOLE
*** run stopped ***
```

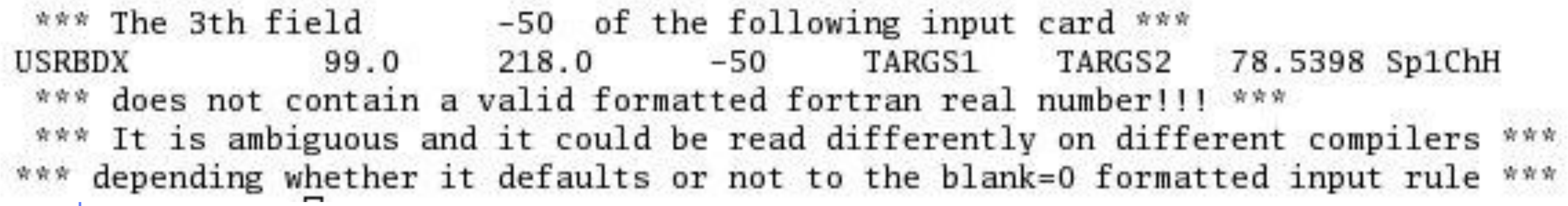
.inp file in fixed format: misalignment in the BLCKHOLE word (the final “E” is lost)



```
*** Unknown control code: ROT-DEF , ignored ***

***** Next control card *****   ROT-DEF    0.000       0.000       0.000       0.000       0.000       0.000       NEW-DEFA

 **** Unknown Input card !!!!!!!!! ****
```

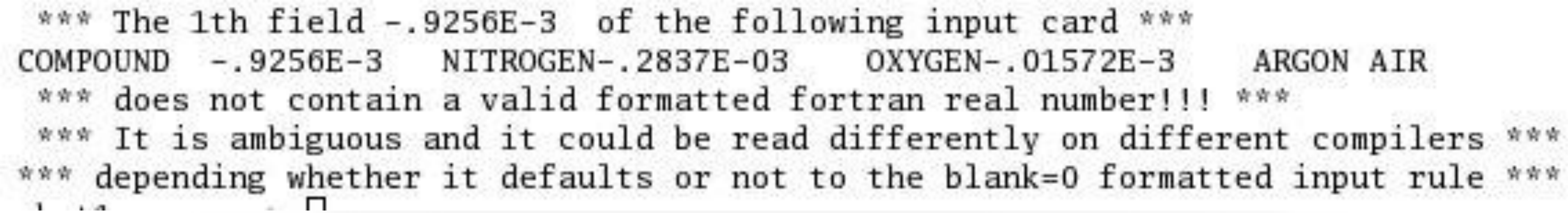
Mistyped ROT-DEFI card: ROT-DEF

# Examples [2/2]



```
 *** The 3th field       -50   of the following input card ***
USRBDX          99.0      218.0       -50      TARGS1     TARGS2   78.5398 Sp1ChH
 *** does not contain a valid formatted fortran real number!!! ***
 *** It is ambiguous and it could be read differently on different compilers ***
*** depending whether it defaults or not to the blank=0 formatted input rule ***
```

The third WHAT of the card has no decimal point: remember that FLUKA expects ALL numbers with a floating point representation, integers included (except only if they are aligned to the right)



```
 *** The 1th field -.9256E-3   of the following input card ***
COMPOUND   -.9256E-3    NITROGEN-.2837E-03     OXYGEN-.01572E-3     ARGON AIR
 *** does not contain a valid formatted fortran real number!!! ***
 *** It is ambiguous and it could be read differently on different compilers ***
*** depending whether it defaults or not to the blank=0 formatted input rule ***
```

The first WHAT of the card must be aligned to the right, being in exponential format

# Errors in USRBIN/EVENTBIN scoring definitions

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

- USRBIN scoring method:
  - 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…): **WHAT(1)<10**

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

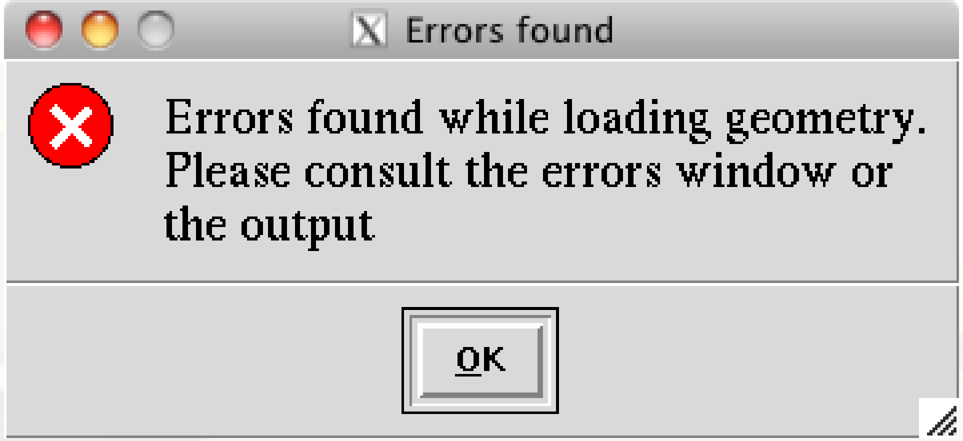
- Badly defined USRBIN scoring;

- Output units:
  - Never use unit numbers <20 (reserved for internal use of FLUKA) or >99 (request by fortran);
  - Never mix the output of different scoring cards in the same unit;

These requests are automatically fulfilled by FLAIR;

# Debugging Geometry Errors [1/3]

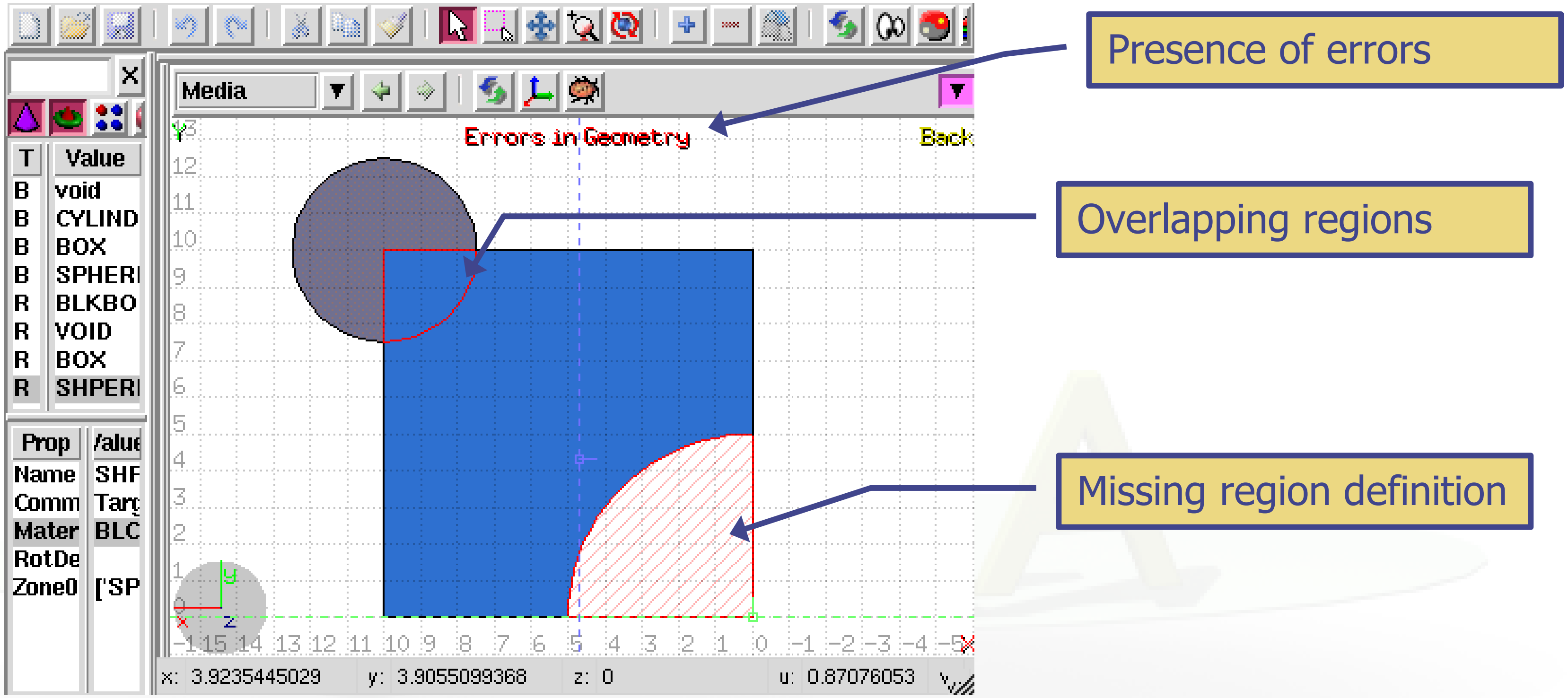
- The geometry editor allows to spot geometry errors when the .inp file is parsed/updated (see dedicated lecture!)



Errors found while loading geometry. Please consult the errors window or the output

- A warning window notifies the user about the presence of errors in the geometry;
- Non-strictly geometrical errors (i.e. missing material assignment to a region, non recognized cards…) are also notified;

# Debugging Geometry Errors… [2/3]



| T | Value |
|---|---|
| B | void |
| B | CYLIND |
| B | BOX |
| B | SPHERI |
| R | BLKBO |
| R | VOID |
| R | BOX |
| R | SHPERI |

| Prop | /alue |
|---|---|
| Name | SHF |
| Comm | Tarç |
| Mater | BLC |
| RotDe | |
| Zone0 | ['SP |

x: 3.9235445029 y: 3.9055099368 z: 0 u: 0.87076053 v:

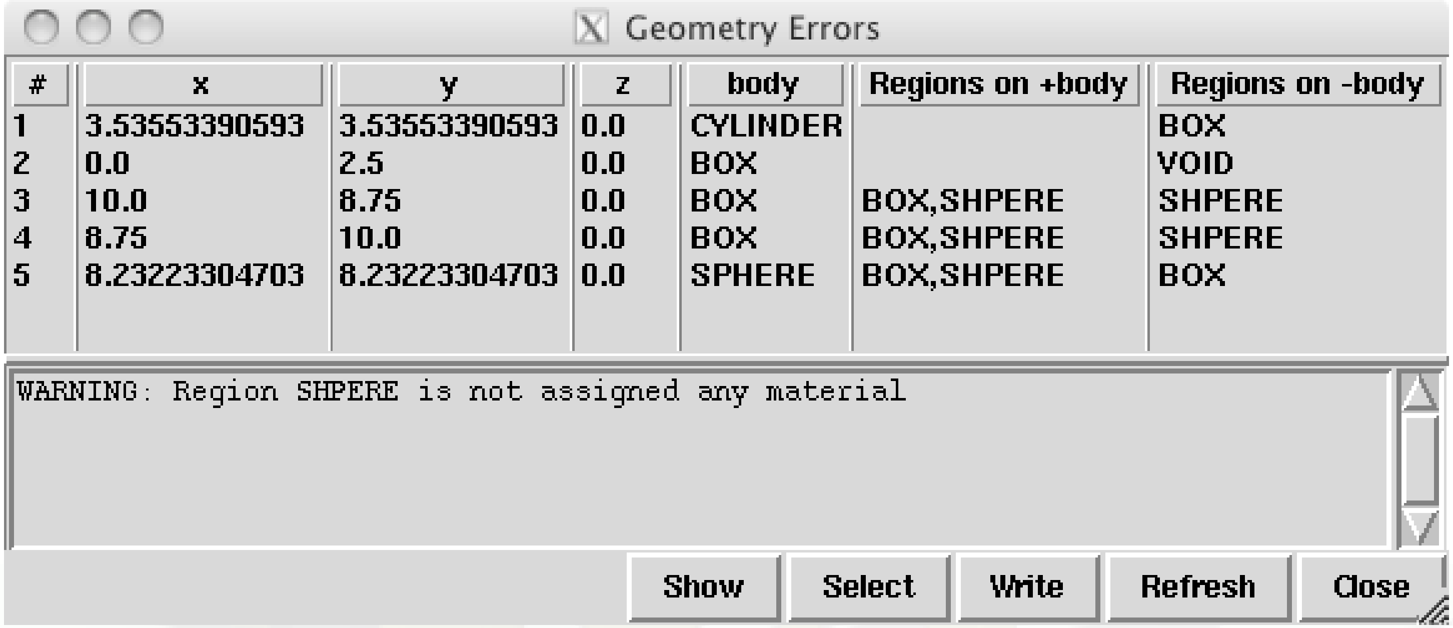
Presence of errors

Overlapping regions

Missing region definition

- The 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 🐞 icon displays the dialog box with the errors;
- Touching surfaces are checked against 10 significant digits;

# Debugging Geometry Errors… [3/3]



| # | x | y | z | body | Regions on +body | Regions on -body |
|---|---|---|---|---|---|---|
| 1 | 3.53553390593 | 3.53553390593 | 0.0 | CYLINDER | | BOX |
| 2 | 0.0 | 2.5 | 0.0 | BOX | | VOID |
| 3 | 10.0 | 8.75 | 0.0 | BOX | BOX,SHPERE | SHPERE |
| 4 | 8.75 | 10.0 | 0.0 | BOX | BOX,SHPERE | SHPERE |
| 5 | 8.23223304703 | 8.23223304703 | 0.0 | SPHERE | BOX,SHPERE | BOX |

```
WARNING: Region SHPERE is not assigned any material
```

**x, y, z** Coordinates of the error (on the surface of the body);

**body** Body to which the error belongs;

**+body** Regions that are on the + side of the body;
Regions where the body should be subtracted to remove the error;

**-body** Regions that are on the – side of the body;
Regions that the body should be intersected to remove the error;

**+/-** are defined according to the normal to the surface, **+** refers to inside, **-** to outside;

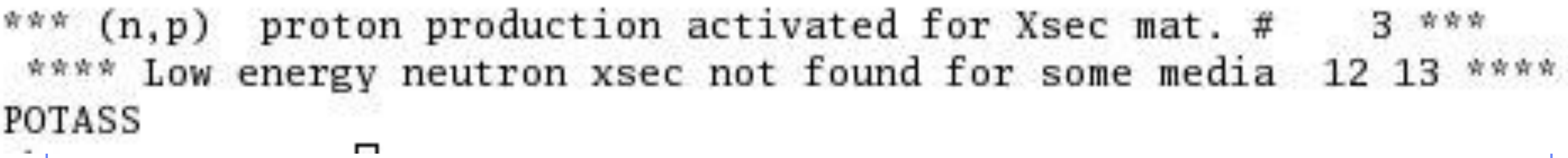
# Geometry: parenthesis expansion

- At the initialization of the geometry, you get the following message in the first line of xxx.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 100’000.
- WARNING even with a simple region expression this limit can be reached, especially if you are using infinite bodies!

# LOW-MAT errors [1/2]



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

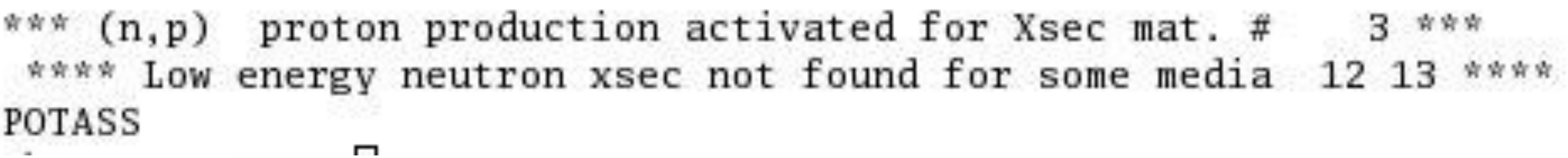
The name of a material breaks the correspondence with the low energy neutron cross section library available in FLUKA (see chap. 10 of the manual). In the present example: POTASS is specified in the SDUM of the LOW-MAT card concerned, instead of POTASSIU

Check the use of LOW-MAT cards in the manual, in particular material names and updated parameters.

NB: Do you really need LOW-MAT?
This card is necessary only in a limited number of cases, and it can be easily misinterpreted if you are not an expert user.
*[Read carefully the manual]*

# LOW-MAT errors [2/2]

- WARNING: the default temperature has changed with respect to older FLUKA distributions:
  - New library with 260 groups: **296 K**
  - Old library with 72 groups: **293 K**
- When an error is printed, FLUKA reports the SDUM of the LOW-MAT card concerned. *If the SDUM is empty, you will see nothing in the output*



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

# Crashes during tracking: geometry [1/2]

- 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;

# Crashes during tracking: geometry [2/2]

```
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 file) during tracking point to:
  - errors in the geometry;
  - numerical precision errors;
  - perpendicularity of the REC and BOX axes (use maximum precision);
- FLUKA will always try to “save” the problematic particles: it will stop the run in case of too many issues.
- The problematic point is given by position + step * direction
- LATTICE 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.

# It runs, but…

something seems wrong…

you do not understand the results…

LOOK in the output file!

what?

everything!

a few examples follow, you’ll find out more…

# 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 is on the previous cycle which didn't generate a random number seed

# Cards defining the primary particle

The total/partial energy balance does not meet expectations

- Remember that for HEAVY IONS the Energy or Momentum in the BEAM card is PER NUCLEON, while in all other cards the energy is the total kinetic one. Verify 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 it is always inside a region;
- If you define a SOURCE routine, the standard FLUKA source is NOT CALLED.
  - You have to define all the concerned distributions (position, momentum etc...);
  - You need a BEAM card to define the maximum energy of your problem;

# MATERIAL / COMPOUND cards

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

please 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;
- recursion in COMPOUND cards;
- Formatting (the NAME-based format is preferred…);

# Geometry induced weird results

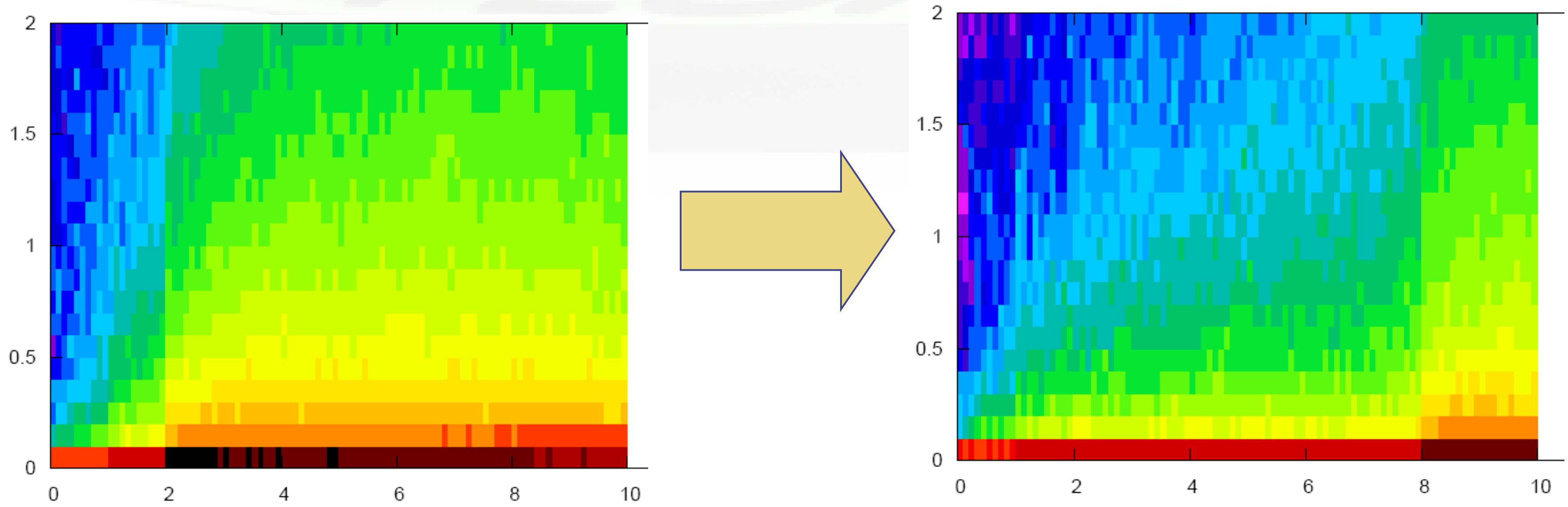
possible symptoms:

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

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

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

Example: in "our" target, make targs2 overlap with targs3



# Preprocessor

Preprocessor directives are very powerful, but pay attention with nesting:

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

# Heavy Ions

You have a heavy ion beam and no reaction products

Did you link the interaction generators with ldpmqmd?

Check, looking for initialization message like “RQMD initialized” and “DPMJET-3 initialized” in the .out file

# Merging cycles from different jobs

- Verify that you merge cycles the .inp files of which are different only by the RANDOMIZ card and/or number of primaries;
- It’s a good habit to clean the files before starting a run:
  - Flair offers this possibility from the “Output Files” frame;
  - It’s good to develop the habit to clean the output files from test runs.

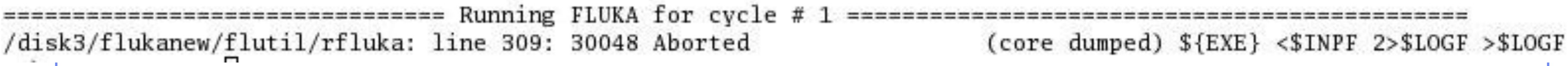
# FLUKA users and the manual

All the reported problems so far implies that a FLUKA user lives symbiotically with the Manual....

# NOW PANIC

# Run time errors with exceptions [1/2]

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 remains there, 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 before the error!

The GNU debugger (gdb) can help in spotting the problem.

# Run time errors with exceptions [2/2]

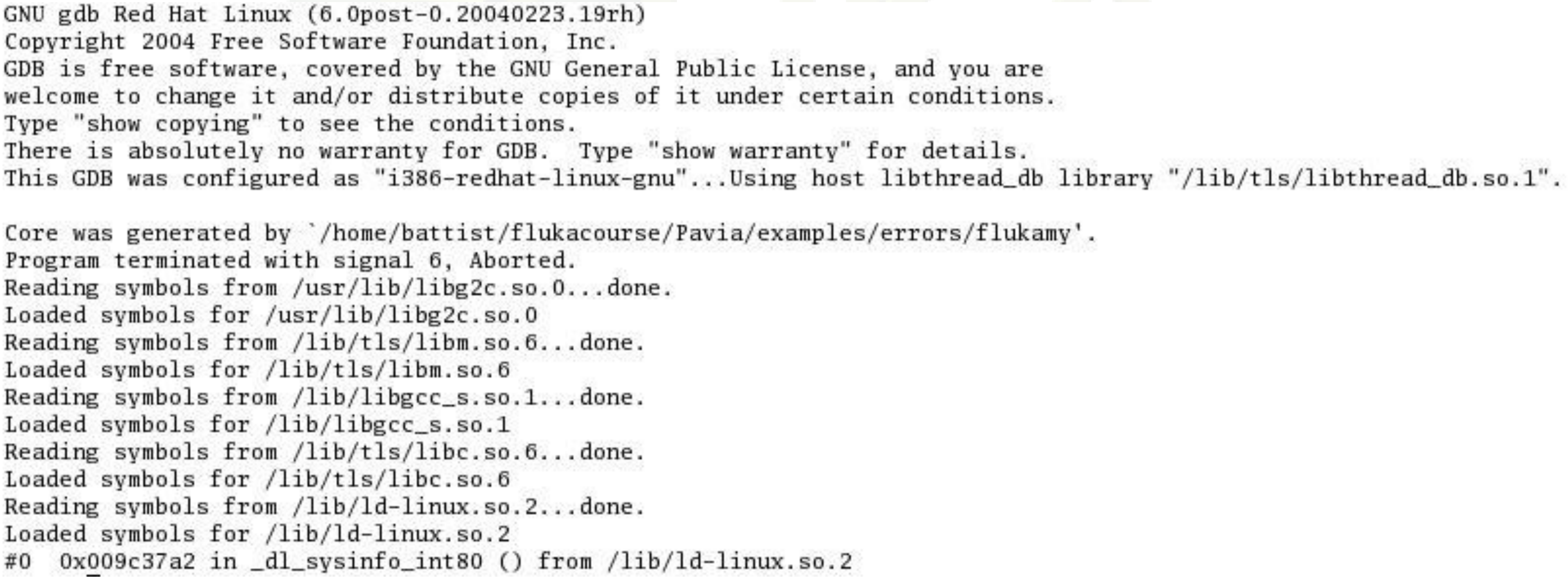
Suggestions:

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. If you use your own user routines, please check them:
   - with the hints found at the beginning of the *.log file;
   - with the help of the gdb debugger;
3. Maybe there is a very well hidden geometry problem;

# gdb [1/2]

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;

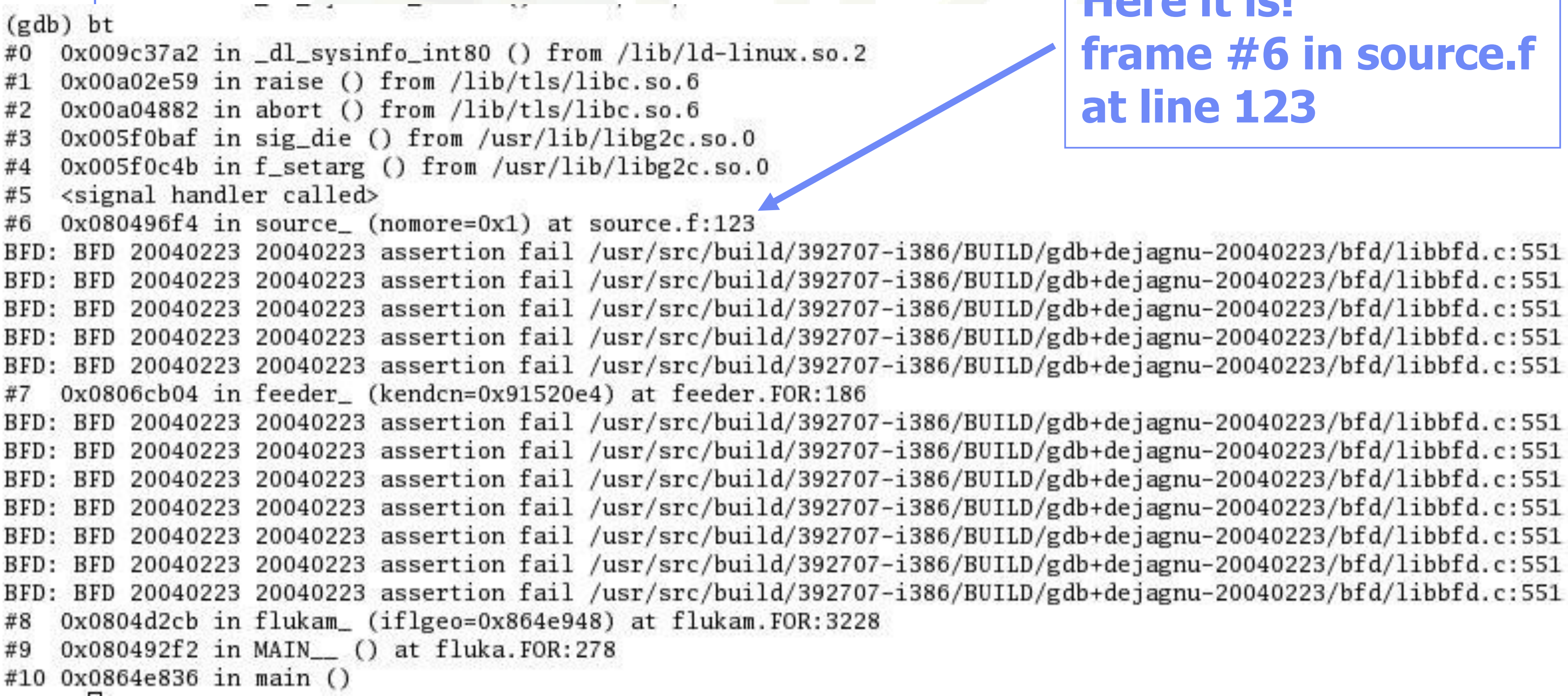


```
GNU gdb Red Hat Linux (6.0post-0.20040223.19rh)
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There 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 [2/2]

Basic commands:

- **`bt`**, for listing the functions which led to the current one and the crash;
- **`f`**, for switching to a given frame (i.e. interrupted function);
- **`p`**, for printing 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**

# 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: a problem in 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!

# FLUKA bug?

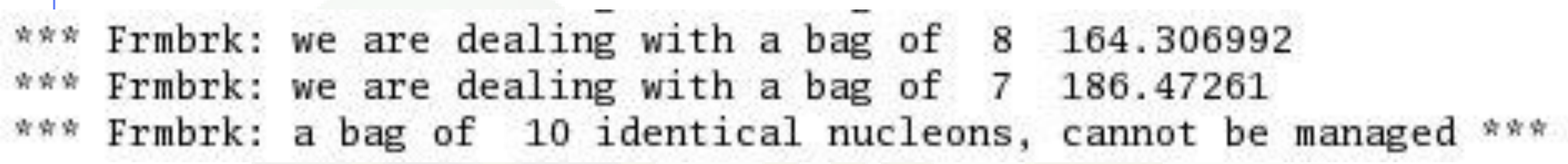
If your crash does not fall in any of the previous cases, and you do not understand what happens, prepare a report with ( possibly ) the gdb results and send it to fluka-discuss@fluka.org together with a tar file containing:

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

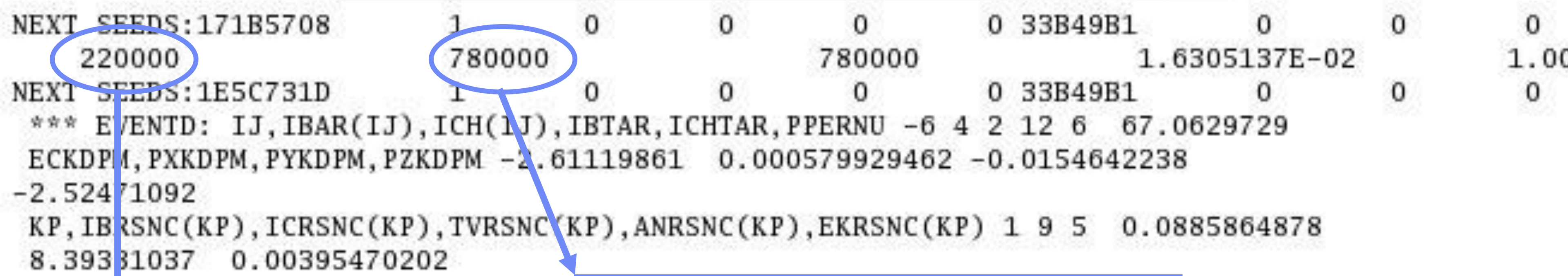
# The *.err file

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 ***
```



```
NEXT SEEDS:171B5708       1       0       0       0       0 33B49B1       0       0       0
   220000                 780000                  780000                1.6305137E-02        1.00
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
 8.39381037  0.00395470202
```

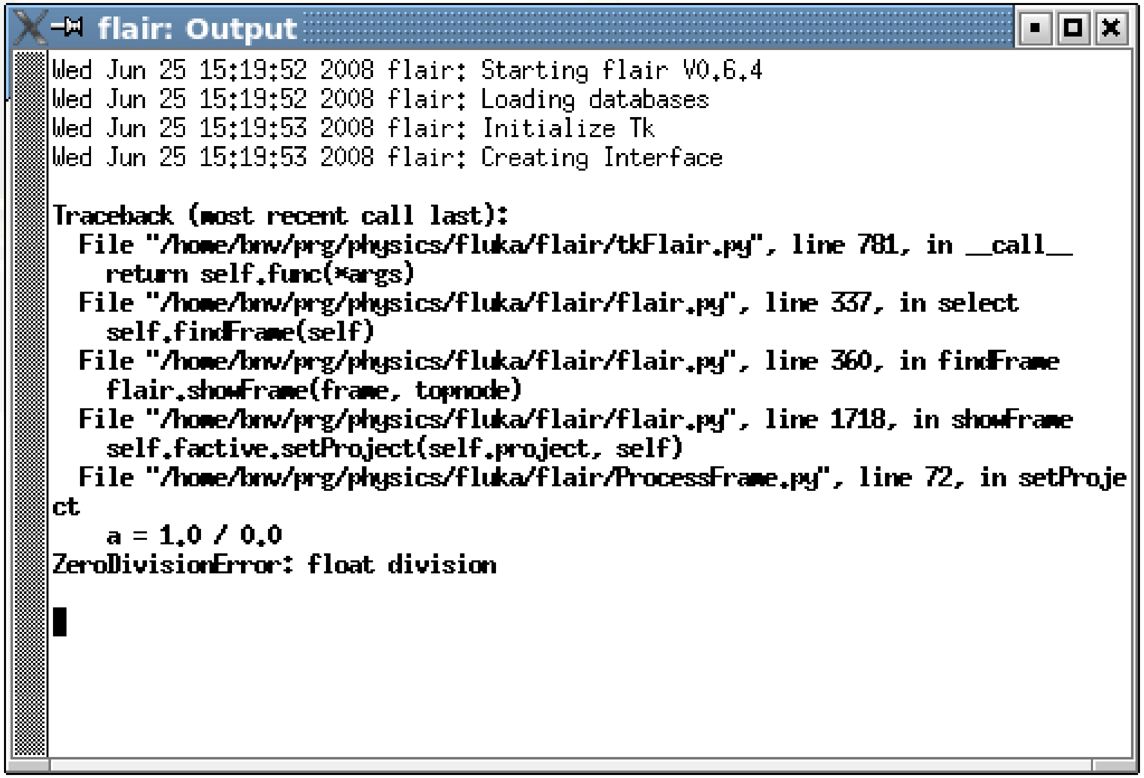
**No. of events simulated so far**

**No. of events remaining to be simulated**

# 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

A Traceback message means that something went wrong with FLAIR



```
Wed Jun 25 15:19:52 2008 flair: Starting flair V0.6.4
Wed Jun 25 15:19:52 2008 flair: Loading databases
Wed Jun 25 15:19:53 2008 flair: Initialize Tk
Wed Jun 25 15:19:53 2008 flair: Creating Interface

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 showFrame
    self.factive.setProject(self.project, self)
  File "/home/bnv/prg/physics/fluka/flair/ProcessFrame.py", line 72, in setProje
ct
    a = 1.0 / 0.0
ZeroDivisionError: float division
```

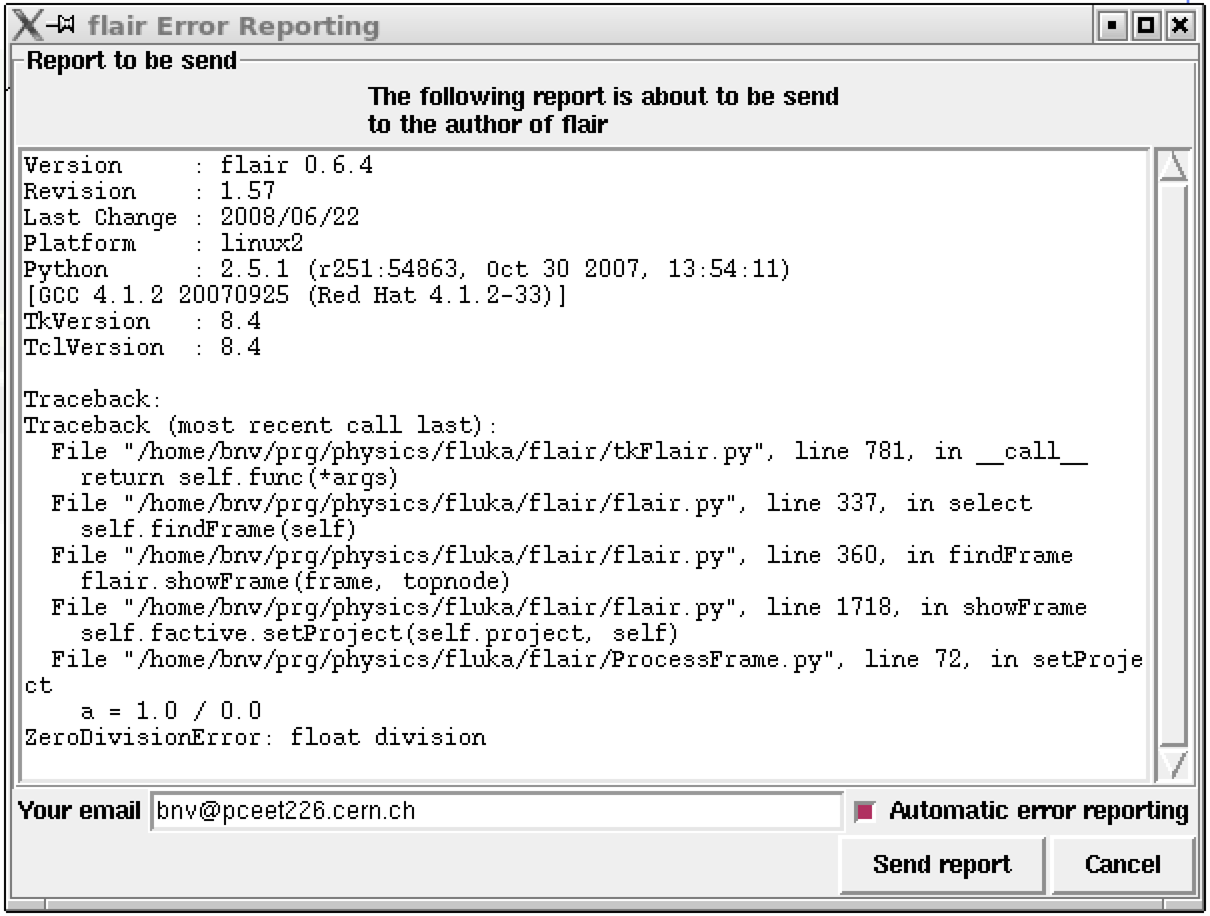
# Flair Error Report

- In case of Traceback, when closing the program or after too many messages, FLAIR will propose to send the Traceback message to the author for further investigation

Please send this report unless you know the reason for the problem, e.g. opening a problematic .inp, non existing directory etc…

It will be good to provide also a small description on how to reproduce the problem.

The text box can be edited



The following report is about to be send
to the author of flair

```
Version      : flair 0.6.4
Revision     : 1.57
Last Change  : 2008/06/22
Platform     : linux2
Python       : 2.5.1 (r251:54863, Oct 30 2007, 13:54:11)
[GCC 4.1.2 20070925 (Red Hat 4.1.2-33)]
TkVersion    : 8.4
TclVersion   : 8.4

Traceback:
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 showFrame
    self.factive.setProject(self.project, self)
  File "/home/bnv/prg/physics/fluka/flair/ProcessFrame.py", line 72, in setProje
ct
    a = 1.0 / 0.0
ZeroDivisionError: float division
```

Your email: bnv@pceet226.cern.ch

☑ Automatic error reporting

# Remember

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

- In case of doubts on the results:
    1. read the FAQ of FLUKA;
    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