FLUKA - Basics

Beginners FLUKA Course

How to download and install Fluka

Two ways of downloading the FLUKA software:

- From the FLUKA website http://www.fluka.org
- From NEA databank http://www.nea.fr through the liaison officer from your institute

It is mandatory to be registered as FLUKA user. Follow the following link:

http://www.fluka.org/download.html

After registration (or using your user-id and password) normally you can then proceed in downloading the latest official release version.

Before continuing we just need to find out one more thing, the '**shell**' you're using on your computer (mostly **bash** or **tcsh**). For this purpose please issue the following command in a terminal window:

```
echo $SHELL
```

You will then get as a results the current shell which is used on your computer, *i.e.,* one of the following:

/bin/**bash**

/bin/**tcsh**

How to download and install Fluka

First identify the location of the FLUKA distribution file: fluka2008.3b-linuxAA.tar.gz.
Depending on the OS and the method you used most probably will be located in one of the following directories:

```
        /media/FLUKA/Software          # in case you are using the USB stick
or      $HOME                          # if you downloaded from the web
        $HOME/Desktop                  # -//- depending on your browser
```

We will create a directory FLUKA under your home directory to install FLUKA.
The following commands issued from a terminal/console window will perform the entire installation.

```
        cd                             # change directory to your home
        mkdir FLUKA                    # create a directory called FLUKA
        cd FLUKA                       # change to the FLUKA directory
        tar xzf /media/disk/Software/ fluka2008.3b-linuxAA.tar.gz
                                       # expand the FLUKA package
        export FLUPRO=$HOME/FLUKA      # set FLUPRO in bash shell or similar
or      setenv FLUPRO $HOME/FLUKA      # set FLUPRO  in tcsh shell or similar
        make                           # compile FLUKA
```

Persistent settings

The only thing left to do is to make these settings persistent on your computer, *i.e.*, you don't have to set the environment variable again when you open a new terminal or log into your computer. We will thus add the following lines into your shell configuration file in your main directory.

bash users:

```
cd
emacs [or any editor] .bashrc
```

"go to the end of the document and add the following"

```
export FLUPRO=${HOME}/FLUKA
export PATH=${PATH}:$FLUPRO:$FLUPRO/flutil
```

tcsh users:

```
cd
emacs [or any editor] .tcshrc
```

"go to the end of the document and add the following"

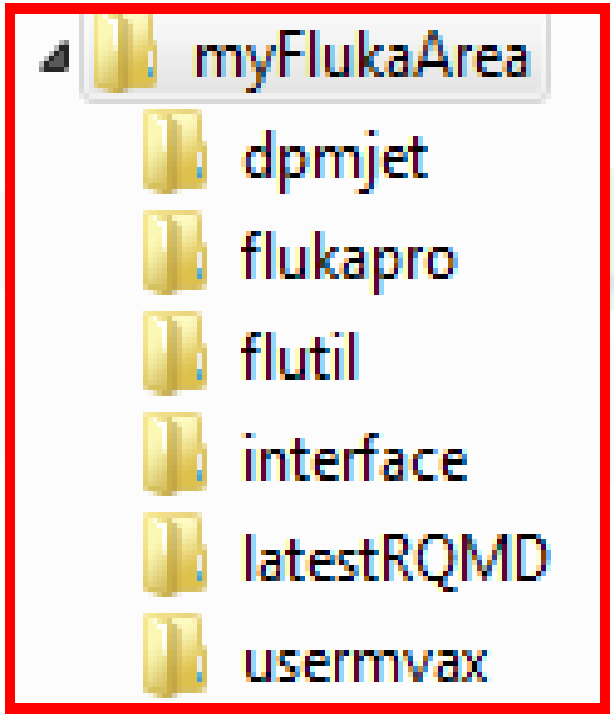
```
setenv FLUPRO ${HOME}/FLUKA
setenv PATH ${PATH}:$FLUPRO:$FLUPRO/flutil
```

The changes will be activated on the next login or if you type the command

```
source .bashrc
source .tcshrc
```

FLUKA directory structure

The tar command will create the following directory structure in your fluka installation directory: ~/work/FLUKA



Root directory of FLUKA

DPMJET data files

All FLUKA commons

Utility programs and scripts

Interface libraries to DPMJET

Interface libraries to rQMD

Fortran user routines

FLUKA release: main directory $FLUPRO

Main Library:

libflukahp.a (object collection)

Physics data files:

sigmapi.bin
elasct.bin
brems_fin.bin
cohff.bin
gxsect.bin
neuxsc-ind_260.bin
neuxsc-ind_72.bin
nuclear.bin
fluodt.dat
e6r1nds3.fyi
jef2.fyi
jendl3.fyi
xnloan.dat

Basic Scripts: (in $FLUPRO/flutil)

rfluka
lfluka
fff

Random Number seed

random.dat

Important Directories

flukapro/	all fluka commons
usermvax/	user routines
flutil/	general utilities

What's inside the physics data files:

sigmapi.bin:	pion-N double-diff. cross sections
elasct.bin:	elastic scattering cross sections
brems_fin.bin:	bremsstrahlung cross sections
cohff.bin:	atomic form factor tabulations
gxsect.bin	photon cross sections
neuxsc-ind_72.bin:	low energy neutron multi-group cross sections (72 groups)
neuxsc-ind_260.bin:	low energy neutron multi-group cross sections (260 groups)
nuclear.bin:	nuclear masses, mass excesses, levels, and many other nuclear data for evaporation, pre-equilibrium, Fermi break up and photonuclear xs gamma and beta databases
fluodt.dat:	Fluorescence data (photoelectric effect)
e6r1nds3.fyi:	Fission products (for neutrons with E<20 MeV)
jef2.fyi:	
jendl3.fyi:	
xnloan.dat:	

Available Documentation

- fluka2008.manual ASCII version of the manual (easy to edit)
- FM.pdf current version of the FLUKA manual
- CERN-2005-10.pdf official reference for FLUKA

- or navigate the manual, online version (www.fluka.org)

- or (when using FLAIR) press F1 to get an interactive manual (which can be also called on prompt level by calling '*fm.py*')

- or (at a further stage) the FAQ available at: http://www.fluka.org/Faq.html

- or (at a further stage) the archive of fluka-discuss: http://www.fluka.org/MailingList.html

FLUKA Release Notes Fluka2008.3b

The latest major release was a major step in the FLUKA development cycle with respect to the last official release version Fluka2006.3b. It added a few new features and there are a few major physics improvements.

This recent intermediate update (3b) contains a few small fixes and a significant functionality improvement with respect to Fluka2008.3.7 (the latest respin of Fluka2008.3). Most of them are likely to be irrelevant for the majority of the users, however it is safer and recommended to immediately move to Fluka2008.3b. The move should be painless since there is no change in the physics, apart the new functionality which is described below (and which is not activated by default).

Compton scattering with taking fully into account the binding and orbital electron motion: up to now FLUKA included two possibilities for the treatment of Compton scattering:

1. "naive" scattering on free electrons
2. Compton scattering corrected by an inelastic form factor, S(q,Z)

Now a third possibility has been added, where both binding effects and orbital motion of all electronic shells of all elements are accounted for. This is particularly relevant for low energy photons and/or heavy elements

FLUKA Release 2008 New Features

- New **neutron cross section library** below 20 MeV, including **260 neutron and 42 gamma groups**: 31 neutron groups are thermal (1 in the previous library).
- All neutron cross section data are freshly computed from the most recent evaluated nuclear data files.
- Please note that the new 260 group library is now the default one (even though the "old" 72 group one is still distributed).
- **New radioactive decay database**, now including also conversion electron and Auger lines- Heavy ion pair production
- New **implementation of the BME model** with vastly improved performances for peripheral collisions
- A new **neutrino-nucleus event generator**, built from scratch for FLUKA, including quasi-elastic, resonance, and deep-inelastic event generation (thanks to M. Lantz, P.R. Sala, G. Smirnov, G. Battistoni, and A. Ferrari)

Input example

- FLUKA is driven by the user almost completely by means of an input file (.inp) which contains directives issued in the form of DATA CARDS
- The standard release provides a simple case to test the installation: example.inp *(Production of particles in p-Be collisions with a 50 GeV/c proton beam.)*
- Different examples are used along this course, which will be varied in different ways for didactic reasons
- We will start with a minimum input file and after each lecture we will enhance our example with more and more functionality
- It is strongly recommended that for every exercise you create a subdirectory *i.e.,* ex1, ex2, ex3 where all the necessary input and output file will be stored
- For better clarity before starting a new exercise you will get the solution of the previous one, to be picked up at the course website: http://www.fluka.org/fluka.php?id=course&sub=intro&which=demokritos2009

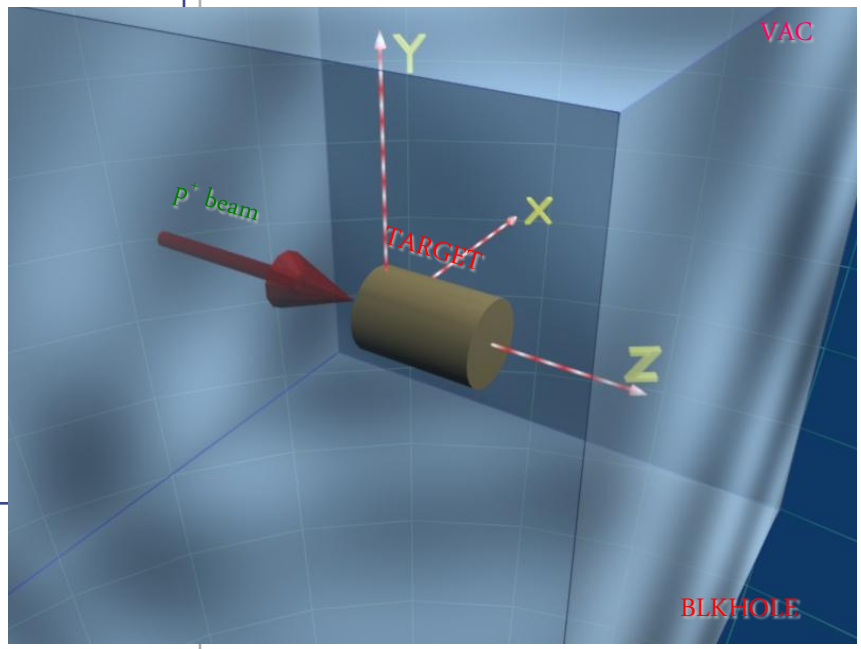
A Simple Example

```
TITLE
FLUKA Course Exercise
*23456789 123456789   123456789 123456789 123456789 123456789 123456789 123456789

DEFAULTS                                                              NEW-DEFA
BEAM              -3.5    -0.082425          -1.7          0.0          0.0          1.0PROTON
BEAMPOS           0.0               0.0           0.1          0.0          0.0          0.0

*23456789 123456789   123456789 123456789 123456789 123456789 123456789 123456789

GEOBEGIN                                                              COMBNAME
    0    0          Cylindrical Target
SPH BLK           0.0  0.0  0.0  10000.
* vacuum box
RPP VOI           -1000. 1000. -1000. 1000. -1000. 1000.
* Lead target
RCC TARG          0.0 0.0 0.0 0.0 0.0 10. 5.
  END
* Regions
* Black Hole
BLKHOLE           5    +BLK -VOI
* Void around
VAC               5    +VOI -TARG
* Target
TARGET            5    +TARG
  END
GEOEND
*23456789      123456789   123456789 123456789 123456789 123456789 123456789 123456789
ASSIGNMA BLCKHOLE   BLKHOLE
ASSIGNMA      VACUUM            VAC
ASSIGNMA           LEAD     TARGET
*
RANDOMIZ                1.0
START                   10.0              0.0
STOP
```



Prepare the working space

- We don't want to run inside the $FLUPRO directories therefore:
- Go to your home directory and create a subdirectory named Work:

```
cd
mkdir work
cd work
mkdir exercises
```

- Change to the examples subdirectory and create a new one named: ex1

```
cd exercises
mkdir ex1
cd ex1
```

- Get the source example file from the course website (copy all the ex_1*??? files to your subdirectory: ~/work/exercises/ex1)

http://www.fluka.org/fluka.php?id=course&sub=intro&which=demokritos2009

```
download ex_1*
```

Now let's test the installation

Soon after you have created your standard FLUKA we can run the first example:

No. of previous cycle (default is 0)

No. of Last cycle (default is 5)

$FLUPRO/flutil/rfluka -e $FLUPRO/flukahp -N0 -M1 ex1

Specifies the executable name: if it is flukahp in $FLUPRO (default) then it can be omitted

Name of the datacard input file. It must be a file named ****.inp and .inp has to be omitted.

What rfluka does:

It creates a temporary subdirectory: $PWD/fluka_nnnn

($PWD means the current directory)

where nnnn is the system process-id assigned to FLUKA. There all necessary assignments are defined and output files are written.

elasct.bin → $FLUPRO/ elasct.bin

fluodt.dat → $FLUPRO/ fluodt.dat

fort.1 → ../ranex_1001

fort.11 → ex_1001.out

fort.12 → libec_thihecufealw_10t.pemf

fort.15 → ex_1001.err

fort.16 → "geometry scratch"

fort.2 → ranex_1002

neuxsc.bin → $FLUPRO/ neuxsc-ind_260.bin

nuclear.bin → $FLUPRO/ nuclear.bin

sigmapi.bin → $FLUPRO/ sigmapi.bin

xnloan.dat → $FLUPRO/ xnloan.dat

At the end of the FLUKA run:

If everything is OK the temporary directory disappears

And the relevant results are copied in the start directory:

by default you have ex_100n.log, ex_100n.out, ex_100n.err and ranex_100m (seed for cycle m = n+1)

```
Removing links

Removing temporary files

Saving output and random number seed

Saving additional files generated
    Moving fort.33 to /home/battist/FlukaCourse/ex_1001_fort.33
    Moving fort.47 to /home/battist/FlukaCourse/ex_1001_fort.47
    Moving fort.48 to /home/battist/FlukaCourse/ex_1001_fort.48
    Moving fort.49 to /home/battist/FlukaCourse/ex_1001_fort.49
    Moving fort.50 to /home/battist/FlukaCourse/ex_1001_fort.50

End of FLUKA run
```

Additional files resulting from the scoring required by the user

Checking FLUKA during the run

Look in the temporary directory:

a) Initialization phase ends when the *.err file is opened.

b) Inside *.err file and (at the end of *.out file) the progress in the number of events is given in the line immediately following the one which starts by "NEXT SEEDS":

```
NEXT SEEDS:  C8888D      0      0      0      0      0 33B49B1      0      0      0
         1                  9                  9          0.0000000E+00        1.0000000E+30                  0
NEXT SEEDS:  C88894      0      0      0      0      0 33B49B1      0      0      0
         2                  8                  8          5.0010681E-03        1.0000000E+30                  0
NEXT SEEDS:  C8889A      0      0      0      0      0 33B49B1      0      0      0
         3                  7                  7          3.3340454E-03        1.0000000E+30                  0
.....
```

EVENTS ALREADY COMPLETED

EVENTS TO BE COMPLETED

AVERAGE CPU TIME CONSUMED PER EVENT

Tips & Tricks

How to make a "clean" stop of FLUKA run

- Here "clean" means performing CLOSE of all files and removing the temporary directory and files.
- In the temporary run directory:

	touch fluka.stop	To stop the present cycle
or	touch **r**fluka.stop	To stop all remaining cycles

- The clean stop will occur at the next CPU-time check, *i.e.*, at the same time when printing the random number calls : see START card instructions (5th parameter) for the frequency of these checks!!

- If the check is never performed it means that the program has entered an infinite loop (probably a fault in user code)

A New Way to "Go FLUKA" - **FLUPIX**

Besides the current FLUKA distribution you also got the current version of FLUPIX distributed on your USB stick. Some of your computers were already installed using this very flexible way to perform FLUKA calculations in a platform independent way:

FLUPIX

(FLUka in knopPIX)

© Vasilis.Vlachoudis@cern.ch 2008

FLUPIX is a **KNOPPIX** (www.knoppix.org) version of the Live CD, with pre-installed FLUKA and flair and all the necessary tools in for performing FLUKA runs.

FLUPIX can run from a CD/DVD, bootable USB or through any virtual machine from any host operating system (Ms Windows, Mac OS, Linux, Solaris etc.). FLUPIX includes all the additions of VirtualBox (www.virtualbox.org) a free and open source Virtual machine supported by Sun, that provides easy installation and high performance way.

More details will be given in a moment by Vasilis