FLUKA

蘭州大學
LANZHOU UNIVERSITY

Installing and Running

23rd FLUKA Beginner's Course
Lanzhou University
Lanzhou, China

June 2–7, 2024

Installing and Running
How to download and install FLUKA

Download the FLUKA software from either the:

- FLUKA website http://www.fluka.org

It is mandatory to be registered as FLUKA user.

Follow the link:

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

After registration, using your **FLUKA user-id (fuid)** and **password**, you can proceed to download the latest official release version. FLUKA releases have a numbering scheme like:

Fluka<major revision>.<minor revision><patch level>(.respin)

Installing and Running

The currently available distribution files are:

- Linux

fluka2024.1-linux-gfor64bit-9.4-glibc2.17-AA.tar.gz	64 bit, gfortran 9.4, glibc-2.17
fluka2024.1-linux-gfor64bit-10.3-glibc2.32-AA.tar.gz	64 bit, gfortran 10.3, glibc-2.32
fluka2024.1-linux-gfor64bit-10.5-glibc2.17-AA.tar.gz	64 bit, gfortran 10.5, glibc-2.17
fluka2024.1-linux-gfor64bit-11.4-glibc2.35-AA.tar.gz	64 bits, gfortran 11.4, glibc 2.35
fluka2024.1-linux-gfor64bit-12.2-glibc2.35-AA.tar.gz	64 bit, gfortran 12.2, glibc 2.35
fluka2024.1-linux-gfor64bit-13.2-glibc2.38-AA.tar.gz	64 bit, gfortran 13.2, glibc-2.38
fluka2024.1-0.x86_64.rpm	rpm 64 bit, gfortran-13.2
fluka2024.1-0.i686.rpm	rpm 32/64 bit G77
fluka2024.1-linuxAA.tar.gz	32/64bit G77

- MAC Apple Silicon (M1/2/3)

fluka2024.1-macm123-gfor64bit-12.3-AA.tar.gz	64 bit, gfortran 12.3

- MAC intel

fluka2024.1-mac-gfor64bit-12.2-AA.tar.gz	64 bit, gfortran 12.2
fluka2024.1-mac-gfor64bit-13.2-AA.tar.gz	64 bit, gfortran 13.2

Installing and Running

How to download and install FLUKA from the tar file

Choose the file compatible with your operating system/compiler version and download it.

Important!

Data files for Fluka2024.1.0 – `fluka2024.1-data.tar.gz` need to be downloaded as well, except if you are installing via *rpm.

Installing and Running

How to download and install FLUKA – g77/gfortran

The installation for g77 and gfortran versions follow the same procedure.
Attention! For gfortran, you must ensure that you have the right gfortran and glibc versions:

```
ldd --version # check version of glibc
gfortran --version # check version of gcc-gfortran
```

Also, it is important to tell the system that we are using gfortran, either by setting another environment variable FLUFOR with

```
export FLUFOR=gfortran # sets FLUFOR in bash shell or similar
or setenv FLUFOR gfortran # sets FLUFOR in tcsh shell or similar
```

or, by choosing a name for the installation directory containing “gfor”, as in this course.

Installing and Running

In the following instructions we assume you are using gfortran, having gfortran version $\gamma\gamma.\gamma$ and glibc version $\eta\eta.\eta$, thus the tar file will be `fluka2024.1-linux-gfor64bit-γγ.γ-glibcηη.η.tar.gz`:

From a terminal/console window, create a directory fluprogfor under your home directory and install FLUKA.

```
cd # changes directory to your home
mkdir fluprogfor # creates a directory called fluprogfor
cd fluprogfor # changes to the fluprogfor directory
tar zxvf path-to-download/fluka2024.1-linux-gfor64bit-γγ.γ-glibcηη.η.tar.gz #
expands the FLUKA package
tar zxvf path-to-download/fluka2024.1-linux-gfor64bit-γγ.γ-glibcηη.η.tar.gz #
expands the data package
```

set the FLUPRO environment variable

```
export FLUPRO=$HOME/fluprogfor # sets FLUPRO in bash shell or similar
or setenv FLUPRO $HOME/fluprogfor # sets FLUPRO in tcsh shell or similar
make # builds a FLUKA executable and auxiliary programs
```

Installing and Running
How to download and install FLUKA from rpm

On systems supporting rpms you can install FLUKA via the rpm distribution file, depending on your architecture choose either fluka2024.1-0.x86_64.rpm (gfortran) or fluka2024.1-0.i686.rpm (g77)
Note: The gfortran rpm needs the most recent versions of compiler and glibc

Some Linux distributions offer graphical rpm installers; alternatively, you can install the rpm directly from the command line, for instance using:

```
rpm -ivh path-to/fluka2024.1-0.x86_64.rpm
or
dnf install path-to/fluka2024.1-0.x86_64.rpm
```

Note: In this way FLUKA will be installed in the system directory tree (`/usr/local`) and hence one needs root privileges (or according permissions via sudo) for the installation.

Installing and Running
$FLUPRO!

FLUPRO must be set each time you compile or run FLUKA

To make environment variable settings persistent on your computer, you can add the following lines in your shell configuration file:

- bash

```
cd emacs .bashrc #feel free to use other text editor and add
export FLUPRO=$HOME/fluprogfor
export FLUFOR=gfortran # if gfortran is required
export PATH=$PATH:$FLUPRO:$FLUPRO/flutil
```

- tcsh

```
cd emacs .tcshrc #feel free to use other text editor and add
setenv FLUPRO $HOME/fluprogfor
setenv FLUFOR gfortran # if gfortran is required
setenv PATH $PATH:$FLUPRO:$FLUPRO/flutil
```

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

```
source $HOME/.bashrc
source $HOME/.tcshrc
```

Installing and Running

FLUKA release: main directory $FLUPRO

Main library

`libflukahp.a` (object collection)

Physics data files:

```
sigmapi.bin            jendl40.fyi
elasct.bin             xnloan.dat
neuxsc-ind_260.bin     nunstab.data
nuclear.bin            sid*.dat
fluodt.dat             grv*.grid
brems_fin.bin          CT14LL.pds
gxsect.bin             dpmjpar.dat
cohff.bin              cx*.bin
endf8r0.fyi            pwxs/*.pwx
incohff.bin            Fad/*
jef33.fyi              DDS/*
```

Basic Scripts (in $FLUPRO/flutil):

```
rfluka
lfluka
ldpmqmd
fff
```

Random Number seed

`random.dat`

Important Directories

`flukapro/` *all FLUKA commons*
`usermvax/` *user routines*
`flutil/` *general utilities*

Installing and Running

Working directory

- Reserve the $FLUPRO directory for the FLUKA installation only.
- Simulations shall be run within separate working directories.
- The FLUKA code and scripts take care of retrieving all information, provided the environmental variable $FLUPRO is set!
- you can check with:

```
env | grep FLUPRO
```

Installing and Running

Available documentation in the installation folder

- fluka2024.manual ASCII version of the manual
- FM.pdf current version of the FLUKA manual
- CERN-2005-10.pdf historic FLUKA reference (not up to date)

You can always navigate the manuals available online at `www.fluka.org` or, when using Flair, press F1 to get an interactive manual.

It is important to keep in mind that the fluka-discussion list archive contains extensive information which can be relevant for new users; regarding new features you can always consult the Release Notes included in the FLUKA installation folder.

Installing and Running
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 - `example.inp` - to test the installation, in three different formats (free, fixed and mixed)
- Different examples are used along this course, which will be varied in different ways for didactic reasons, generally increasing in complexity as we progress throughout the course.
- A solution of each exercise is also included so you can compare the results at the end.

Installing and Running
Prepare the working space

- For convenience of access, place the exercise materials (i.e., Exercises) in a folder of your choice and create a directory for each exercise e.g., new_running where all the necessary input and output file will be stored.

Remember

We don't want to run inside the $FLUPRO directory, so you can always use your home folder:

```
cd # changes directory to your home
mkdir new_running # same pattern for other exercises
cd new_running
cp path-to-download/Exercises/example_running/example_running.inp .
mv example_running.inp your_running.inp
```

Installing and Running

Units and Coordinates

- FLUKA units:
  - Length [cm]
  - Mass [g]
  - Energy [GeV]
  - Time [s]

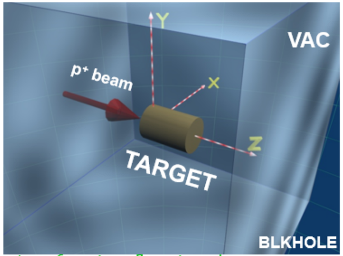
- FLUKA coordinate system:
  - Right-handed Cartesian system
  - By default, the primary beam is directed along the z axis, positive direction
    – *Obviously this can be changed by the user.*

Installing and Running

A simple example

```
TITLE
FLUKA Course Exercise
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
DEFAULTS                                                              NEW-DEFA
BEAM            -3.5      -0.8      -1.7       0.0       0.0       1.0PROTON
BEAMPOS          0.0       0.0      -0.1       0.0       0.0       0.0
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
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
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
ASSIGNMA    BLCKHOLE   BLKHOLE
ASSIGNMA      VACUUM       VAC
ASSIGNMA        LEAD    TARGET
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....*
RANDOMIZ         1.0
START           10.0       0.0
STOP
```

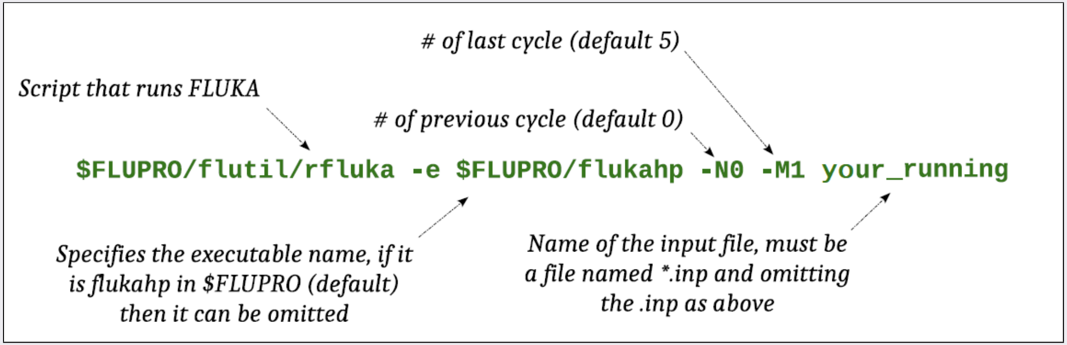
Geometry



Installing and Running

Testing the installation

After creating your standard FLUKA input we can run the first example:



```
$FLUPRO/flutil/rfluka -e $FLUPRO/flukahp -N0 -M1 your_running
```

Script that runs FLUKA
# of last cycle (default 5)
# of previous cycle (default 0)
Specifies the executable name, if it is flukahp in $FLUPRO (default) then it can be omitted
Name of the input file, must be a file named *.inp and omitting the .inp as above

Installing and Running
What rfluka does 1/2

- It creates a temporary subdirectory: $PWD/fluka_nnnn – **$PWD** stands for the current directory and **nnnn** is the system process-id assigned to FLUKA – There all necessary logical links are established and output files are written.

```
elasct.bin     →  $FLUPRO/elasct.bin
fluodt.dat     →  $FLUPRO/fluodt.dat
fort.1         →  ../ranexample_running001
fort.11        →  example_running001.out
fort.15        →  example_running001.err
fort.16        →  "geometry scratch"
fort.2         →  ranexample_running002
neuxsc.bin     →  $FLUPRO/neuxsc-ind_260.bin
nuclear.bin    →  $FLUPRO/nuclear.bin
sigmapi.bin    →  $FLUPRO/sigmapi.bin
xnloan.dat     →  $FLUPRO/xnloan.dat
```

For non-experts in Fortran

`fort.xx` is the default file name for writing/reading in Fortran, xx being a logical unit number. Can be substituted of course with a real name.

Installing and Running

What `rfluka` does 2/2

- As described in the introduction to Monte Carlo:
  - FLUKA uses **pseudo-random** numbers to simulate physics processes
  - Many “**histories**”, or **“primary particles”** are needed to achieve statistical convergence
  - Statistical errors can be derived as RMS from “batches” of primaries
- `rfluka` takes care of running several “batches” or cycles
  - numbering them for convenience and further use and giving appropriate names to the output files: i.e., your_running002.out is the output from input your_running.inp’s 2nd cycle.
  - How many cycles? Defined by the `-M` and `-N` parameters, from cycle `N+1` to cycle `M` – The collection of these cycles is called a **“run”**
  - The pseudo-random sequence is preserved by FLUKA + `rfluka`:
    1. Initial random copied from $FLUPRO or generated (see lecture) as **ran**_yourinp_**001**
    2. At the end of the N^{th} cycle, random written to **ran**_yourinp_**### , ###=N+1**
    3. To be used as starting point for the next cycle

Installing and Running
At the end of the FLUKA run

Assuming everything went O.K. the temporary directory disappears and the relevant results are copied in the start directory after:

- Removing links
- Removing temporary files
- Saving output and random number seed*
- Saving additional files from scoring requested by the user (see scoring lecture):

```
Moving fort.33 to /home/username/new_running/your_running001_fort.33
Moving fort.47 to /home/username/new_running/your_running001_fort.47
Moving fort.48 to /home/username/new_running/your_running001_fort.48
Moving fort.49 to /home/username/new_running/your_running001_fort.49
Moving fort.50 to /home/username/new_running/your_running001_fort.50
```

- End of FLUKA run

*by default you have `your_running00n.log`, `your_running00n.out`, `your_running00n.err` (**n**=cycle) and `ranyour_running00m` (seed for cycle **m** = n+1)

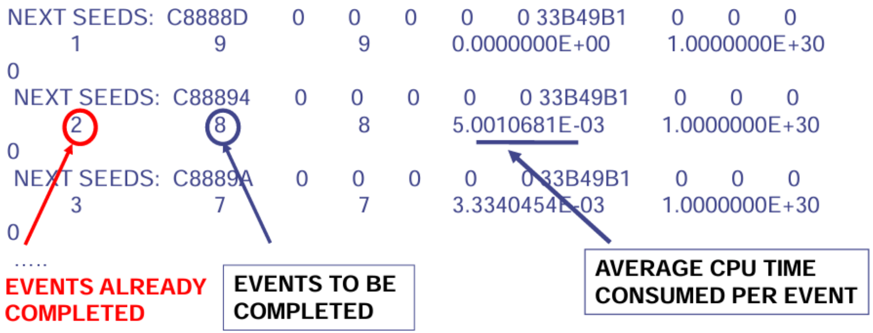
Installing and Running

Checking FLUKA during the run

Look in the temporary directory:

a Initialization phase ends when the *.err file is created.

b Inside the *.err file (and at the end of *.out file) the progress in the number of events is listed immediately after the line with “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
.....
```

Installing and Running

Always open the output file

- The standard inp###.out file contains plenty of information
- If FLUKA crashes, it gives hints on the reason
- It tells you how FLUKA interpreted your input cards $\rightarrow$ spot subtle errors
- It lists the physics data used by FLUKA
- It provides a summary of the cycle: energy deposited, CPU time, particles produced...
- When setting up a simulation, it is a good practice to always run a short test and check the output file
- If something in the results puzzles you, always check in the output file that the settings are what you meant to have.
- We will show you examples all along the course

Installing and Running

Output–Timing of the run- number of primaries

Use it to choose the number of primaries/cycle

Q: how many primaries?

A: as many as needed to reach a good statistical convergence

Q: what is a "reasonable" CPU time for a long cycle ?

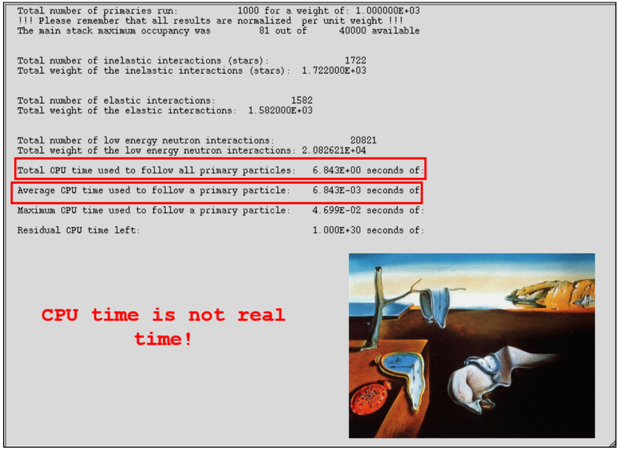
A: less than one day, to be on the safe side for crashes

Q: in this example, how many primaries can be run in a 10h cycle?

A: $3600/6.8 \cdot 10^{-3} \approx 5 \cdot 10^{5}$

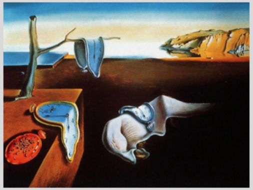
Q: how many cycles?

A: minimum 5 to be able to calculate statistics



```
Total number of primaries run:          1000 for a weight of: 1.000000E+03
!!! Please remember that all results are normalized per unit weight !!!
The main stack maximum occupancy was          81 out of      40000 available


Total number of inelastic interactions (stars):              1722
Total weight of the inelastic interactions (stars):  1.722000E+03


Total number of elastic interactions:              1582
Total weight of the elastic interactions:  1.582000E+03


Total number of low energy neutron interactions:              20821
Total weight of the low energy neutron interactions: 2.082621E+04

Total CPU time used to follow all primary particles:    6.843E+00 seconds of

Average CPU time used to follow a primary particle:    6.843E-03 seconds of

Maximum CPU time used to follow a primary particle:    4.699E-02 seconds of

Residual CPU time left:                                1.000E+30 seconds of
```

CPU time is not real time!

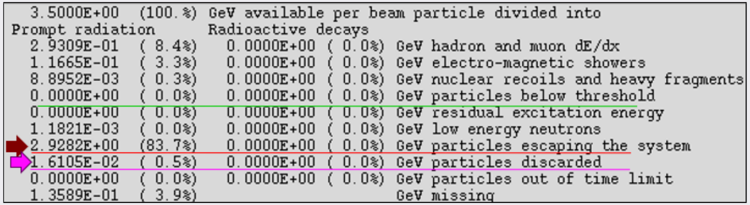


Installing and Running

Complete the run

- add statistics by running more cycles:
- `$FLUPRO/flutil/rfluka -N1 -M5 your_running`
- While it runs, have a look

Output: Energy Balance



```
   3.5000E+00   (100.%)  GeV available per beam particle divided into
 Prompt radiation         Radioactive decays
   2.9309E-01   ( 8.4%)    0.0000E+00 ( 0.0%)  GeV hadron and muon dE/dx
   1.1665E-01   ( 3.3%)    0.0000E+00 ( 0.0%)  GeV electro-magnetic showers
   8.8952E-03   ( 0.3%)    0.0000E+00 ( 0.0%)  GeV nuclear recoils and heavy fragments
   0.0000E+00   ( 0.0%)    0.0000E+00 ( 0.0%)  GeV particles below threshold
   0.0000E+00   ( 0.0%)    0.0000E+00 ( 0.0%)  GeV residual excitation energy
   1.1821E-03   ( 0.0%)    0.0000E+00 ( 0.0%)  GeV low energy neutrons
   2.9282E+00   (83.7%)    0.0000E+00 ( 0.0%)  GeV particles escaping the system
   1.6105E-02   ( 0.5%)    0.0000E+00 ( 0.0%)  GeV particles discarded
   0.0000E+00   ( 0.0%)    0.0000E+00 ( 0.0%)  GeV particles out of time limit
   1.3589E-01   ( 3.9%)                        GeV missing
```

Escaping the system: out of the geometry and going to other blackholes (see lecture on geometry). If you find 100%…maybe something is wrong…

Discarded particles (i.e., neutrinos).

Missing Energy, calculated by difference:

- pure EM problems it should be 0 within the rounding accuracy;
- in hadronic problems it is the energy spent in endothermic nuclear reactions ($\approx$ 8 MeV/n), or gained in exothermic (i.e., mostly neutron capture): it is –total Q.

How to make a "clean" stop of FLUKA run

- Here "clean" means closing all files, writing scoring output and removing the temporary directory and files.
- In the temporary run directory:

```
touch fluka.stop # to stop the present cycle, or
kill -SIGTERM <process_id> # the same id as in the fluka_xxxx, or
touch rfluka.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 (5^{th} 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)

Installing and Running

Mac users

- A Mac version is available
  - Apple Silicon (M1/2/3)
  - Apple Intel
- Users shall have **gfortran** installed.
- For the installation of the flair graphical interface, instructions will be provided briefly in the webpage.

Installing and Running

MS Windows users

- A VM distribution based on Docker is available: https://flukadocker.github.io/F4D/
  - The instructions provided allow you to install Docker, generate your personal Docker image with FLUKA and create your first FLUKA container
  - There is also a list of known issues and instructions to update the FLUKA Docker image
- Scripts to install FLUKA on Windows 10/11 using WSL: https://github.com/flukadocker/fluka_wsl
  - These scripts will set up and install FLUKA on Windows 10/11 using the Windows Subsystem for Linux (WSL).
  - It lets users run GNU/Linux environment - including most command-line tools, utilities, and applications - directly on Windows, unmodified, without the overhead of a virtual machine.

www.fluka.org