FLUKA

in pills

S.M. VALLE, G. BATTISTONI

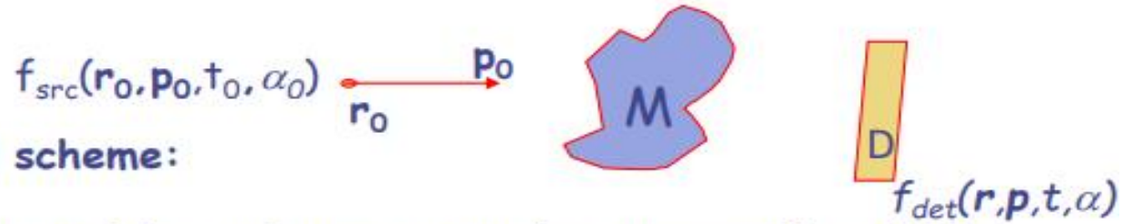
SOFTWARE TUTORIAL, 14-15 JUNE 2018

Outline

- FLUKA: a Monte Carlo code
- Basic input
  - Physics & beam settings
  - Geometry
  - Materials
  - Scoring
- Flair: the user interface
- Running FLUKA

FLUKA: a Monte Carlo code

Particle transport MC in 7 steps



General (simplified) scheme:

1. Initialize particle position and momentum (or energy+direction).
   1.1 If particle is in vacuum, bring it to next material boundary.
2. Determine total cross section at current energy, material, etc: σ
3. *Sample step length to next interaction from exponential distribution.*
4. *Decide nature of interaction:* $P_i = \sigma_i / \sigma, \quad i=1,2,\ldots,n$
5. *Sample energy loss (or change of direction) from differential cross section for interaction mechanism i.*
6. *Add generated secondary particles to the stack* if any.*
7. *Go to 2 unless*
   1. *Particle energy drops below user preset threshold (lecture Thu)*
   2. *Particle exits the geometry*

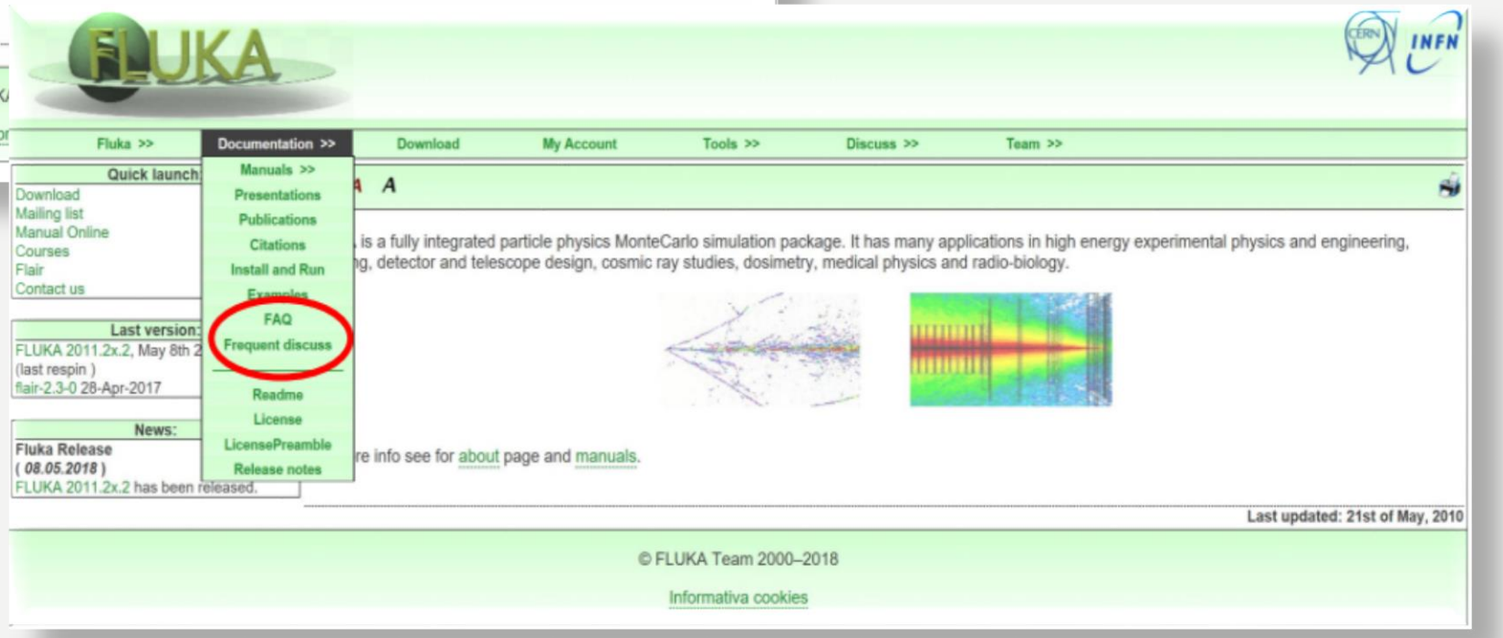
What is FLUKA?

FLUKA is a general purpose tool for calculations of particle transport and interactions with matter, covering an extended range of applications :

- proton and electron accelerator shielding to target design
- calorimetry
- activation
- dosimetry
- detector design
- Accelerator Driven Systems
- osmic rays,
- neutrino physics,
- radiotherapy
- etc.

- All Hadrons (p, n, π, K,pbar, nbar, (anti)hyperons…) [0-10000 TeV]
- Electromagnetic (γ, $e^{+/-}$) and μ and ν [1 keV - 10000 TeV]
- Nucleus-nucleus [0-10000 TeV/n]
- Low energy neutrons (0-20 MeV, multigroup, ENDF…)
- Transport in magnetic field
- Combinatorial (boolean) and Voxel geometries
- Double capability to run either fully analogue and/or biased calculations
- On-line evolution of induced radioactivity and dose
- Radiation damage predictions (NIEL, DPA)
- User-friendly GUI interface thanks to the Flair interface

Finding FLUKA infos



www.fluka.org

In the webpage of the last FLUKA course you can find the lessons' material (click here)

Some FLUKA defaults

Default units of measurement

- time → s
- length → cm
- energy → GeV,
- masses → GeV/c^2
- B → Tesla

Reference frame (cartesian, right-handed)

- z is primary beam direction
- y is pointing upwards

Particles

- each particle is identified by a number (see next slide)

Particles transported

Fluka name	Fluka number	Common name	Standard PDG number (Particle Data Group)
4-HELIUM (1)	-6	Alpha	---
3-HELIUM (1)	-5	Helium-3	
TRITON (1)	-4	Triton	---
DEUTERON (1)	-3	Deuteron	---
HEAVYION (1)	-2	Generic heavy ion (see command HI-PROPE)	
OPTIPHOT	-1	Optical Photon	---
RAY (2)	0	Pseudoparticle	---
PROTON	1	Proton	2212
APROTON	2	Antiproton	-2212
ELECTRON	3	Electron	11
POSITRON	4	Positron	-11
NEUTRIE	5	Electron Neutrino	12
ANEUTRIE	6	Electron Antineutrino	-12
PHOTON	7	Photon	22
NEUTRON	8	Neutron	2112
ANEUTRON	9	Antineutron	-2112
MUON+	10	Positive Muon	-13
MUON-	11	Negative Muon	13
KAONLONG	12	Kaon-zero long	130
PION+	13	Positive Pion	211
PION-	14	Negative Pion	-211
KAON+	15	Positive Kaon	321
KAON-	16	Negative Kaon	-321
LAMBDA	17	Lambda	3122
ALAMBDA	18	Antilambda	-3122
KAONSHRT	19	Kaon zero short	310
SIGMA-	20	Negative Sigma	3112
SIGMA+	21	Positive Sigma	3222
SIGMAZER	22	Sigma-zero	3212
PIZERO	23	Pion-zero	111
KAONZERO	24	Kaon-zero	311
AKAONZER	25	Antikaon-zero	-311
Reserved	26	---	---
NEUTRIM	27	Muon neutrino	14
ANEUTRIM	28	Muon antineutrino	-14
Blank	29	---	---

Fluka name	Fluka number	Common name	Standard PDG number (Particle Data Group)
Reserved	30	---	---
ASIGMA-	31	Antisigma-minus	-3222
ASIGMAZE	32	Antisigma-zero	-3212
ASIGMA+	33	Antisigma-plus	-3112
XSIZERO	34	Xi-zero	3322
AXSIZERO	35	Antixi-zero	-3322
XSI-	36	Negative Xi	3312
AXSI+	37	Positive Xi	-3312
OMEGA-	38	Omega-minus	3334
AOMEGA+	39	Antiomega	-3334
Reserved	40	---	---
TAU+	41	Positive Tau	-15
TAU-	42	Negative Tau	15
NEUTRIT	43	Tau neutrino	16
ANEUTRIT	44	Tau antineutrino	-16
D+	45	D-plus	411
D-	46	D-minus	-411
D0	47	D-zero	421
D0BAR	48	AntiD-zero	-421
DS+	49	D_s-plus	431
DS-	50	D_s-minus	-431
LAMBDAC+	51	Lambda_c-plus	4122
XSIC+	52	Xi_c-plus	4232
XSIC0	53	Xi_c-zero	4112
XSIPC+	54	Xi'_c-plus	4322
XSIPC0	55	Xi'_c-zero	4312
OMEGAC0	56	Omega_c-zero	4332
ALAMBDC-	57	Antilambda_c-minus	-4122
AXSIC-	58	AntiXi_c-minus	-4232
AXSIC0	59	AntiXi_c-zero	-4132
AXSIPC-	60	AntiXi'_c-minus	-4322
AXSIPC0	61	AntiXi'_c-zero	-4312
AOMEGAC0	62	AntiOmega_c-zero	-4332
Reserved	63	---	---
Reserved	64	---	---

Fragments and nucleons originating in the "nuclear evaporation" phase are identified with particle number in the range from -39 to -7. Can be identified by Z and A.

Basic input

Structure of the input file

General definitions

Beam definition
Materials: definition and assignment
Random number initialization
Start/Stop of simulation

Physics settings

Defaults
Physical processes
Transport thresholds
Low energy neutrons
Induced radioactivity

Biasing

Geometry related biasing
Interaction/decay biasing

Geometry

Setup description
Voxel phantoms

Output settings

Estimators / scoring cards

FLUKA commands

Commands aka cards, aka options, aka directives, aka definitions

One keyword (command), 6 floating point numbers (WHATs), one string (SDUM)

Example of a FLUKA command (text editor style)

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM          1.E+04        0.0        0.0        0.0        0.0        0.0PROTON
*keyword    momentum mom.spread  diverg.   X-width   Y-width  ignored particle
*             WHAT(1)    WHAT(2)    WHAT(3)    WHAT(4)    WHAT(5)    WHAT(6)   SDUM
```

- Command keywords MUST be uppercase, numbers MUST have the decimal point
- Some commands require more than one "card"
- Some special commands (like `TITLE` and `OPEN`) are/may be followed by a text line
- With few exceptions, the order of commands is irrelevant
- Most commands can be repeated several times
- Repeated commands can add themselves or override previous commands
- A line with a * character in column 1 is a comment
- Text after an exclamation mark (!) is ignored (does not work within the geometry)
- Almost all the WHAT() have a default value
- Commands can be issued in fixed or free format
- Special commands, called #directives, allow input parametrization

Fixed and free format

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM          1.E+04   0.0D+00       0.0       0.0       0.0       0.0PROTON
*keyword    momentum mom.spread  diverg.   X-width   Y-width  ignored particle
*            WHAT(1)   WHAT(2)   WHAT(3)   WHAT(4)   WHAT(5)   WHAT(6)   SDUM
```

- The "traditional" FLUKA format is (A8, 2X, 6E10.0, A8)
  Numbers: 9 digits at most can be used
- All WHAT fields are in floating point format, *even integers*
  They must always be written with the decimal point
- Exponential notation numbers (e.g. 1.234E+5), must be right aligned
- Double precision format (e.g. 1.234D+5) is allowed
- Blank numerical fields are read as 0.0
  In most cases (*not all!*) such values are ignored and the corresponding default values are used
- Blank lines NOT ALLOWED in geometry declaration (tolerated elsewhere)

- Both lines are correct

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM          1.E+04       0.0       0.0       0.0       0.0       0.0PROTON
BEAM          1.E+04     0.0         0.0       0.0       0.0       0.0  PROTON
*keyword    momentum mom.spread  diverg.   X-width   Y-width  ignored particle
*            WHAT(1)   WHAT(2)   WHAT(3)   WHAT(4)   WHAT(5)   WHAT(6)   SDUM
```

- Incorrect: decimal point is missing

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM          1.E+04       1         0         0         0         0    PROTON
```

WHAT(2) would be interpreted as 1000!

- Incorrect: exponential number not correctly alligned

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM       1.E+04       0         0         0         0         0         PROTON
```

WHAT(1) might be interpreted as 1.E+4000!

- Free format can be made *locally* available issuing option FREE (without any parameter), until the option FIXED restores the fixed format; the opposite can be done either
- Option GLOBAL provides free format also for the geometry input
- In free format input, the different fields are separated by blanks and/or separators (usually commas). *All fields must be present* or at least represented by two successive separators
- Character fields (command name, SDUM) must be input without quotes

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
BEAM , 1.234567890E+04 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , PROTON
*
*keyword    momentum mom.spread  diverg.   X-width   Y-width   weight particle
*            WHAT(1)   WHAT(2)   WHAT(3)   WHAT(4)   WHAT(5)   WHAT(6)   SDUM
```

A basic input

A basic input is a text file with `.inp` extension. The geometry can be stored in another text file with `.geo` extension.

They can be opened with a text editor.

Comment lines begins with *.

```
TITLE
My Basic Input example
* Set the defaults for precision simulations
DEFAULTS                                                              PRECISIO
* Define the beam characteristics
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAM             3.5 -0.082425      -1.7       0.0       0.0          PROTON
* Define the beam position
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAMPOS          0.0       0.0      -0.1       0.0       0.0
*
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 target1    0.0 0.0  0.0 0.0 0.0 10.0 5.0
RCC target2    0.0 0.0 20.0 0.0 0.0 10.0 5.0
RCC target3    0.0 0.0 40.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY      5 +blkbody -void
* Void around
VOID         5 +void -target1 -target2 -target3
* Target
TARGET1      5 +target1
TARGET2      5 +target2
TARGET3      5 +target3
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
MATERIAL        24.0               7.18                               CHROMIUM
MATERIAL         0.0            0.73E-3                               AMMONIA
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
COMPOUND         1.0  NITROGEN       3.0  HYDROGEN                    AMMONIA
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA    BLCKHOLE   BLKBODY
ASSIGNMA      VACUUM      VOID
ASSIGNMA     AMMONIA   TARGET3
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..*
ASSIGNMA    CHROMIUM   TARGET1   TARGET2       1.0

* Set the random number seed
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
RANDOMIZ         1.0 54217137.

* Set the number of primary histories to be simulated in the run
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
START          1000.
STOP
```

```
TITLE
My Basic Input example
* Set the defaults for precision simulations
DEFAULTS                                                              PRECISIO
* Define the beam characteristics
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAM             3.5 -0.082425      -1.7       0.0       0.0          PROTON
* Define the beam position
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAMPOS          0.0       0.0      -0.1       0.0       0.0
*
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 target1    0.0 0.0  0.0 0.0 0.0 10.0 5.0
RCC target2    0.0 0.0 20.0 0.0 0.0 10.0 5.0
RCC target3    0.0 0.0 40.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY      5 +blkbody -void
* Void around
VOID         5 +void -target1 -target2 -target3
* Target
TARGET1      5 +target1
TARGET2      5 +target2
TARGET3      5 +target3
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
MATERIAL        24.0                7.18                              CHROMIUM
MATERIAL         0.0             0.73E-3                              AMMONIA
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
COMPOUND         1.0  NITROGEN       3.0  HYDROGEN                    AMMONIA
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA    BLCKHOLE   BLKBODY
ASSIGNMA      VACUUM      VOID
ASSIGNMA     AMMONIA   TARGET3
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..*
ASSIGNMA    CHROMIUM   TARGET1   TARGET2       1.0

* Set the random number seed
RANDOMIZ         1.0 54217137.
*
* Set the number of primary histories to be simulated in the run
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
START          1000.
STOP
```

Physics & beam settings

Physics settings

Select predefined physics settings (e.g. transport thresholds) for a specific kind of simulation:

SDUM =

- **CALORIMEtry** : calorimeter simulations
- **EET/TRANsmut** : Energy Transformer or transmutation calculations
- **EM-CASCAde** : pure EM cascades
- **HADROTHErapy** : hadrotherapy calculations
- **ICARUS** : studies related to the ICARUS experiment
- **NEUTRONS** : pure low-energy neutron runs
- **NEW-DEFAults** : minimal set of generic defaults – set by default
- **PRECISIOn** : precision simulations (**recommended**)
- **SHIELDINg** : hadron shielding calculations without gammas

DEFAULTS

- EM transport on (**EMF** on), production/transport thresholds should always be set by the **EMFCUT!**
- Inelastic form factor correction to Compton scattering on (**EMFRAY** on)
- Detailed photoelectric edge treatment and fluorescence photons activated
- Low energy neutron transport on (**LOW-NEUT** on), threshold 20 MeV, with fully analogue absorption
- All transport threshold = 100keV, but neutrons (10^{-5} eV) and neutrinos (0, but they are discarded)
- Multiple Scattering threshold at minimum allowed energy, for both primary and secondary charged particles
- Delta rays production on, threshold 100keV (**DELTARAY**)
- Restricted ionization energy loss fluctuations for all particles (**IONFLUCT**)
- Tabulation ratio for hadron/muon dp/dx set at 1.04, fraction of the kinetic energy to be lost in a step set at 0.05, number of dp/dx tabulation points set at 80 (**DELTARAY, EMFFIX, FLUKAFIX**)
- e+e- pair production and bremsstrahlung by heavy particles on (**PAIRBREM**)

  Pair threshold =2 m_e , bremsstrahlung threshold = 300keV
- Muon photonuclear interactions on (**MUPHOTON**)

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+
DEFAULTS                                                        PRECISIO
```

Physics settings & transport

PHYSICS

Allows to override standard FLUKA defaults for some processes:
- activates coalescence (critical for calculation of residual nuclei)
- activates the new fragmentation model ("evaporation" of fragments up to A=24, critical for calculation of residual nuclei)
- activates PEANUT above 5 GeV
- activates electromagnetic dissociation of heavy ions
- activates charmed particle transport

PART-THR

- Defines transport cut-offs for hadrons, muons, and neutrinos
- Setting done by particle type, overriding the selected **DEFAULTS**
- For **neutrons**, a <20.0 MeV cut-off is internally translated into the corresponding group energy; On a region basis, the neutron cut-off can be *increased* by the **LOW-BIAS** card (see Neutrons lecture)
- Charged particles (but electrons) are not stopped, but ranged out to rest in an approximate way (if the threshold is < 100 MeV)

EMF-CUT

For electron, positron, and photon, sets:
- Energy thresholds for production in the selected materials
- Transport cut-offs in the selected regions.
- Use **STRONGLY** recommended

DELTARAY

- Activates delta ray production by muons and charged hadrons
- Sets energy threshold for their production

PAIRBREM

controls simulation of pair production and bremsstrahlung by high-energy heavy charged particles

Beam settings

The card BEAM defines the particle type and energy (or momentum).

The card BEAMPOS controls particle starting position and direction.

BEAMPOS

WHAT(1) = x-coordinate of the spot centre.

WHAT(2) = y-coordinate of the spot centre.

WHAT(3) = z-coordinate of the spot centre.

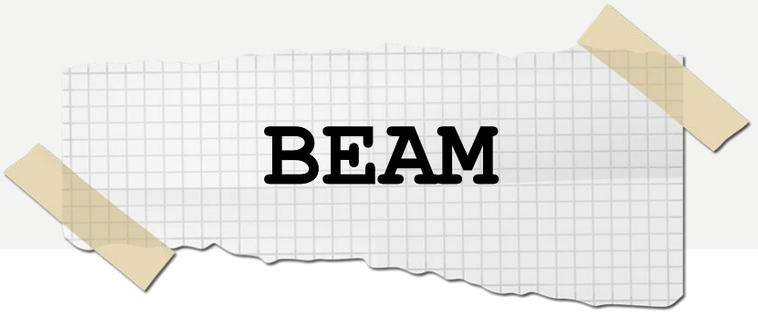
WHAT(4) = direction cosine of the beam with respect to the x-axis of the beam reference frame.

WHAT(5) = direction cosine of the beam with respect to the y-axis of the beam reference

WHAT(6) : not used

SDUM = NEGATIVE means that the direction cosine with respect to z-axis is negative.

Beam settings



WHAT(1) > 0.0 : average beam momentum in GeV/c
< 0.0 : average beam kinetic energy in GeV

WHAT(2) > 0.0 : beam momentum spread in GeV/c. The momentum distribution is assumed to be rectangular
< 0.0 : FWHM of a Gaussian momentum distribution

WHAT(3) specifies the beam divergence (in mrad):
> 0.0 : width of a rectangular angular distribution for a beam directed along the positive z-axis
< 0.0 : FWHM of a Gaussian angular distribution for a beam directed along the positive z-axis

WHAT(4) >= 0.0: If **WHAT(6)**> 0.0, beam width in x-direction in cm for a beam directed along the positive z-axis. The beam profile is assumed to be rectangular. If **WHAT(6)** < 0.0, **WHAT(4)** is the maximum radius of an annular beam spot.
< 0.0 : FWHM of a Gaussian profile in x-direction (whatever the value of **WHAT(6)**) for a beam directed along the positive z-axis

WHAT(5) >= 0.0: If **WHAT(6)**> 0.0, beam width in y-direction in cm for a beam directed along the positive z-axis. The beam profile is assumed to be rectangular. If **WHAT(6)**< 0.0, **WHAT(5)** is the minimum radius of an annular beam spot.
< 0.0 : FWHM of a Gaussian profile in y-direction (whatever the value of **WHAT(6)**) for a beam directed along the positive z-axis

WHAT(6) < 0.0: if positive, **WHAT(4)** and **WHAT(5)** are interpreted as the maximum and minimum radii of an annular beam spot. If negative, they are interpreted as FWHMs of Gaussian profiles as explained above, independent of the value of **WHAT(6)**.
>= 0.0: ignored

SDUM = beam particle name. For heavy ions, use the name HEAVYION and specify further the ion properties by means of option HI-PROPE. In this case **WHAT(1)** will mean the energy (or momentum) PER NUCLEAR MASS UNIT, and not the total energy or momentum. The light nuclei 4He, 3He, triton and deuteron are defined with their own names (4-HELIUM, 3-HELIUM, TRITON and DEUTERON) and **WHAT(1)** will be the total kinetic energy or momentum.

Heavy ions transport

HI-PROPE

- When **BEAM**'s SDUM is **ISOTOPE**...
  - ...specifies the isotope of a radioactive source
  - ...requires a **RADDECAY** card
- When **BEAM**'s SDUM is **HEAVYION**...
  - ...specifies the properties of an ion beam:
  - in the **BEAM** card, the beam energy is given in GeV/nmu
  - (**n**uclear **m**ass **u**nit, i.e. 1/12 of the ^{12}C nucleus mass)
  - ^{2}H, ^{3}H, ^{3}He, and ^{4}He beams have dedicated SDUM in the **BEAM** card

RADDECAY

- Activates the simulation of the decay of generated radioactive nuclides
- Allows to set biasing for radioactive decay products

IONTRANS

- Is not required when using an heavy ion beam **HEAVYION**
- Activates the ions transport
- Allows to limit it to a subset of light ions (A < 5)
- Switches between approximate and full transport
  - ...(including nuclear interactions)
- Nucleus-nucleus interactions above 125 MeV/n
  - ...can be performed only if the event generators
    - DPMJET and RQMD are linked to the FLUKA executable
- Below 125 MeV/n...
  - ...the BME event generator is already linked in the standard executable

Magnetic field

MGNFIELD

Sets the tracking conditions for transport in magnetic fields and possibly defines a homogeneous magnetic field

WHAT(1) = largest angle in degrees that a charged particle is allowed to travel in a single step

WHAT(2) = upper limit to error of the boundary iteration in cm (minimum accuracy accepted in determining a boundary intersection). It also sets the minimum radius of curvature for stepping according to WHAT(1)

WHAT(3) = minimum step length if the step is forced to be smaller because the angle is larger than WHAT(1).

WHAT(4) - WHAT(6): = Bx, By, Bz components of magnetic field on the coordinate axes (in tesla). If Bx = By = Bz = 0.0: a user-supplied subroutine MAGFLD (see routines lecture) is assumed to provide the actual values

SDUM : not used

Magnetic field tracking is performed only in regions defined as magnetic field regions by command **ASSIGNMA** (see later). It is strongly recommended to define as such only regions where a non-zero magnetic field effectively exists, due to the less efficient and accurate tracking algorithm used in magnetic fields.

Seed, Start & Stop

RANDOMIZ

The random number generator is initialized to read a vector of 97 seeds from an external file. Different numbers input will initialize different and independent random number sequences.

[WHAT(1)] : logical file unit from which to read the seeds. Must be 1.0!!
[WHAT(2)] : any number < 9.E8, initialization of the random seed sequences. Different WHAT(2) lead to different sequences allowing to run parallel jobs
[WHAT(3-6), SDUM] : not used

START
STOP

A START card at the end of the input file is mandatory. It defines the number of particle histories required.
[WHAT(1)] = maximum number of primary histories simulated in the run
[WHAT(2)] = not used
[WHAT(3-6), SDUM] = see manual

The START card is optionally followed by a STOP card, which stops the execution of the program.

```
TITLE
My Basic Input example
* Set the defaults for precision simulations
DEFAULTS                                                              PRECISIO
* Define the beam characteristics
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAM             3.5 -0.082425      -1.7        0.0        0.0           PROTON
* Define the beam position
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAMPOS          0.0        0.0       -0.1        0.0        0.0
*
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 target1    0.0 0.0  0.0 0.0 0.0 10.0 5.0
RCC target2    0.0 0.0 20.0 0.0 0.0 10.0 5.0
RCC target3    0.0 0.0 40.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY      5 +blkbody -void
* Void around
VOID         5 +void -target1 -target2 -target3
* Target
TARGET1      5 +target1
TARGET2      5 +target2
TARGET3      5 +target3
END
GEOEND
*
MATERIAL         24.0                 7.18                             CHROMIUM
MATERIAL          0.0              0.73E-3                             AMMONIA
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
COMPOUND          1.0  NITROGEN       3.0  HYDROGEN                    AMMONIA
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA    BLCKHOLE   BLKBODY
ASSIGNMA      VACUUM      VOID
ASSIGNMA     AMMONIA   TARGET3
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..*
ASSIGNMA    CHROMIUM   TARGET1   TARGET2       1.0

* Set the random number seed
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
RANDOMIZ         1.0 54217137.
*
* Set the number of primary histories to be simulated in the run
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
START          1000.
STOP
```

Geometry

Combinatorial geometry

Basic objects called bodies (such as cylinders, spheres, parallelepipeds, etc.) are combined to form more complex objects called regions

This combination is done using Boolean operations:

Math	Operation	FLUKA
∪	Union	|
∩	Intersection	+
-	Subtraction	-

- **Bodies**: basic convex objects, plus infinite planes (half-spaces), infinite cylinders (circular and elliptical), and generic quadric surfaces (surfaces described by 2nd degree equations)
- **Regions**: defined as Boolean operations on bodies (union of zones)
- **Lattices**: duplication of existing objects (translated & rotated), will be explained in a separate lecture

GEOBEGIN card
- VOXELS card (optional, see advanced geometry lecture)
- Geometry title (and reading format options)
  - Body data
- END card
  - Region data
- END card
- LATTICE cards (optional, see advanced geometry lecture)
- Region volumes (optionally requested by a flag in the Geometry title, used together with the SCORE command)

GEOEND card

Bodies

- Each body divides the space into two domains: inside and outside.
- 3-character code of available bodies:
  - RPP: **R**ectangular **P**arallele**P**iped
  - SPH: **SPH**ere
  - XYP, XZP, YZP: Infinite half space delimited by a coordinate plane
  - PLA: Generic infinite half-space, delimited by a **PLA**ne
  - XCC, YCC, ZCC: Infinite Circular Cylinder, parallel to coordinate axis

- The input for each body consists of:
  - the 3-letter code indicating the body type (RPP, ZCC...)
  - a unique "body name" (alphanumeric identifier, 8 character maximum, **case sensitive**)
  - a set of geometrical quantities defining the body (the number depends on the body type, see next slides)

Bodies: planes

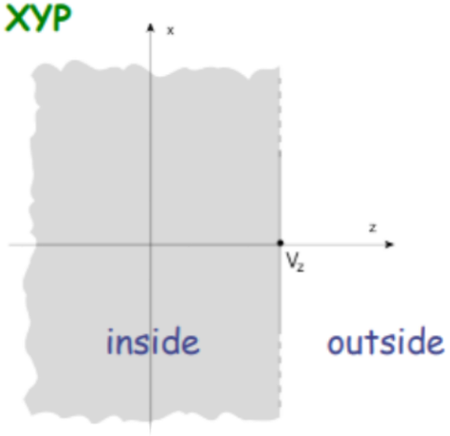
YZP
XZP
XYP

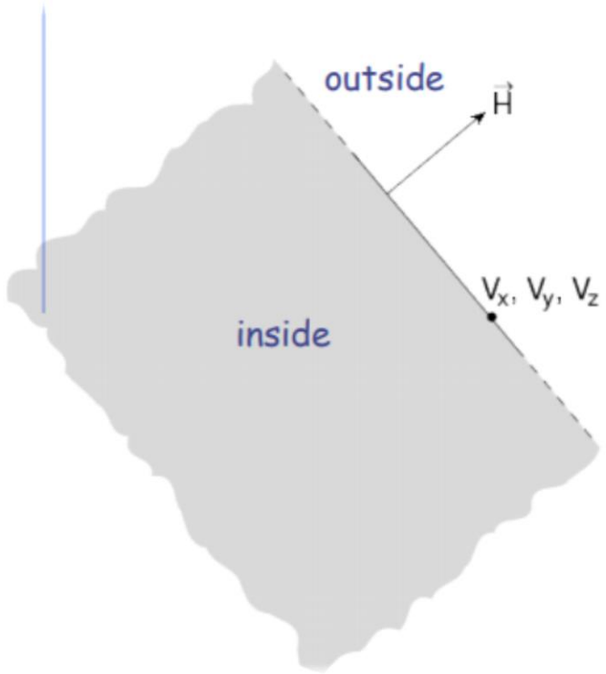
Three are delimited by planes perpendicular to the coordinate axes:

1. Delimited by a plane ⊥ to the x-axis. **Code:** YZP
2. Delimited by a plane ⊥ to the y-axis. **Code:** XZP
3. Delimited by a plane ⊥ to the z-axis. **Code:** XYP

All defined by a single number:
Vx (resp. Vy, or Vz),
coordinate of the plane on the corresponding perpendicular axis

Points for which:
x < Vx (resp. y < Vy, or z < Vz)
are "inside the body"





A PLA defines the infinite half space delimited by a generic plane

A PLA is defined by 6 numbers:
Hx, Hy, Hz (vector ⊥ to the plane, arbitrary length);
Vx, Vy, Vz (any point lying on the plane)

The half-space "inside the body" is that from which the vector is pointing (i.e. the vector points "outside").

PLA

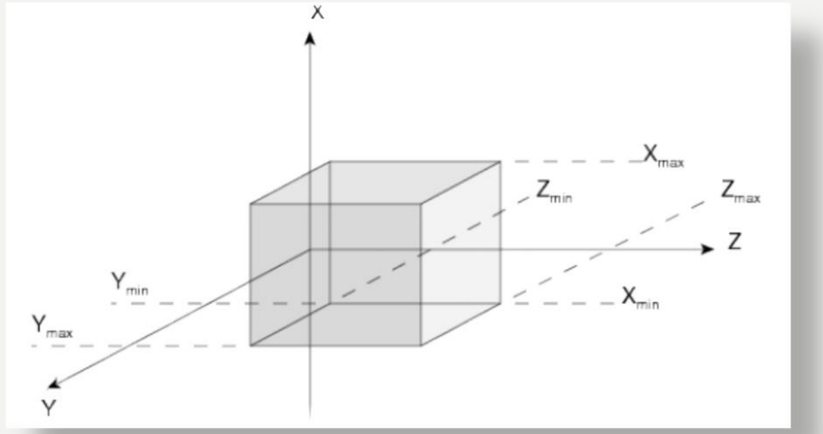
Bodies: parallelepipeds & spheres

RPP

An RPP has its edges parallel to the coordinate axes
It is defined by 6 numbers in the following order:

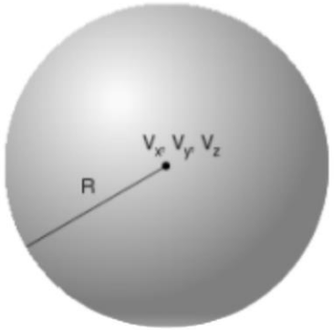
X_{min} , X_{max} , Y_{min} , Y_{max} , Z_{min} , Z_{max}

(min and max coordinates delimiting the parallelepiped)



SPH

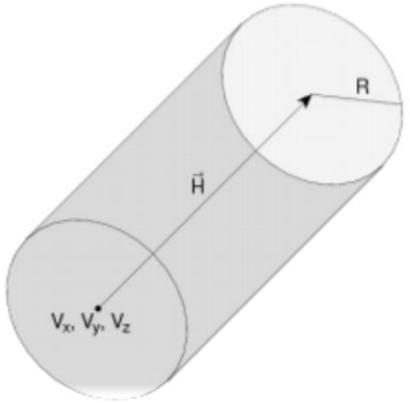
A SPH is defined by 4 numbers:
Vx, Vy, Vz (coordinates of the centre), R (radius)



Bodies: cylinders

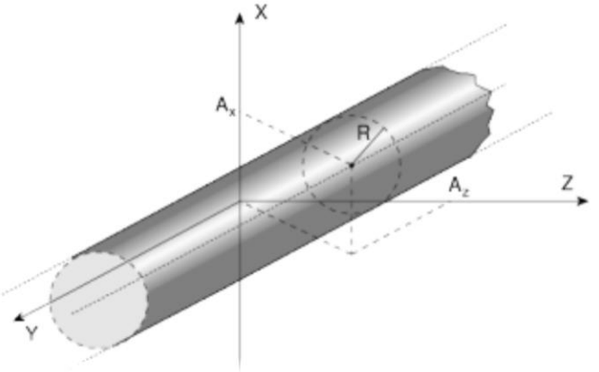
RCC

An RCC can have any orientation in space
Limited by a cylindrical surface and two plane faces ⊥ to its axis.
Each RCC is defined by 7 numbers:
Vx, Vy, Vz (centre of one face);
Hx, Hy, Hz (vector corresponding to the cylinder height, pointing toward the other face);
R (cylinder radius).



XCC
YCC
ZCC

Infinite Circular Cylinder parallel to coordinate axis:
XCC, YCC, ZCC
Each XCC (YCC, ZCC) is defined by 3 numbers:
Ay, Az for XCC
(Az, Ax for YCC, Ax, Ay for ZCC) (coordinates of the cylinder axis),
R (cylinder radius)



Regions

Regions are defined as combinations of bodies obtained by boolean operations:

	Union	Subtraction	Intersection
Name based format	|	-	+
Fixed format	OR	-	+
Mathematically	∪	−	∩

Regions but must be of homogeneous material composition.

Each point of space must belong to one and only one region!

Input for each region starts on a new line and extends on as many continuation lines as are needed. It is of the form:

REGNAME NAZ boolean-zone-expression

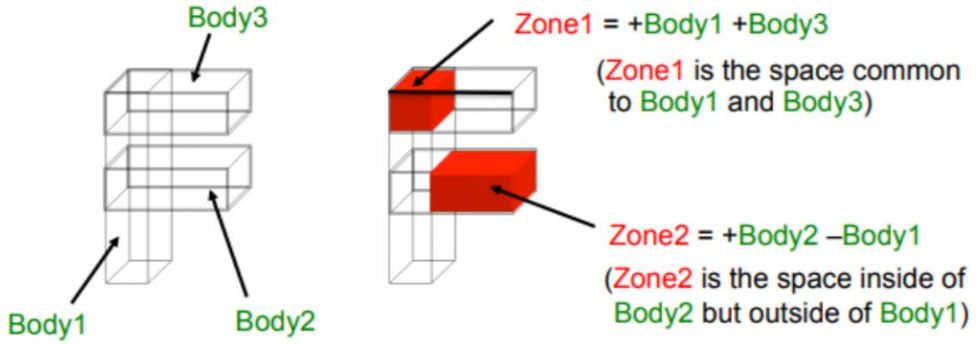
or

REGNAME NAZ boolean-zone-expression | boolean-zone-expression | ...

- REGNAME is the region "name"
  alphanumeric identifier, 8 character maximum, **case sensitive**
  Must start by an alphabetical character
- **NAZ** (Number of Adjacent Zones) is a rough estimate of the number of zones a particle can enter when leaving the current region zones (5 by default). What actually matters is the NAZ sum over all regions, defining the size of the *contiguity list*

Regions: boolean zone expressions

- Zones are defined by intersections and/or subtraction of bodies
  - Zones are a described by a sequence of one or more bodies each being preceded by its + or – sign
  - **+body**: only the inner part of the body can belong to the zone (means that the zone being described is fully contained inside this body)
  - **–body**: only the outside of the body can belong to the zone (means that the zone being described is fully outside this body)



Zones must be finite: normally in the description of each zone

Zones can overlap ➔ Points in space could belong to more than one zone BUT of the **SAME region**!

- The | (or OR) operator is used as a Boolean union operator in order to combine zones
- Such combination of zones forms a region
  - In its simplest form a region just consist of one zone
  - Regions are not necessarily simply connected, i.e. they can consist of zones which are not contiguous
  - On the other hand, zones belonging to the same region can be partially overlapping

Example:

```
Ground 5 | +Body9 | +Body15 | +Body1 | +Body8 -Body2 | +Body8 -Body3
*          <- 1st -><-  2nd -><- 3rd -><----  4th ----><----  5th --->
         | +Body8 +Body18  | +Body12 -Body10 -Body11 -Body13 -Body14
*           <-----  6th -----><------- 7th and last zone       -------->
```

In name based format one can also use parenthesis to form more complex Boolean operations

In evaluating the expressions, the highest operator precedence is given to **parentheses**, followed by **+**, **-** and the **|** operator

Regions: the black hole

All particles entering a black-hole are absorbed (they vanish)

FLUKA geometry MUST be embedded into a BLCKHOLE region
(to avoid tracking particles to infinity)

The outer surface of the BLCKHOLE region must be a single closed body (e.g. a sphere).

Further black-hole regions can be defined by the user if desired

BLCKHOLE region: has material BLCKHOLE assign to it

```
TITLE
My Basic Input example
* Set the defaults for precision simulations
DEFAULTS                                                              PRECISIO
* Define the beam characteristics
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAM              3.5 -0.082425      -1.7        0.0        0.0           PROTON
* Define the beam position
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAMPOS           0.0        0.0      -0.1        0.0        0.0
*
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 target1     0.0 0.0  0.0 0.0 0.0 10.0 5.0
RCC target2     0.0 0.0 20.0 0.0 0.0 10.0 5.0
RCC target3     0.0 0.0 40.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY       5 +blkbody -void
* Void around
VOID          5 +void -target1 -target2 -target3
* Target
TARGET1       5 +target1
TARGET2       5 +target2
TARGET3       5 +target3
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
MATERIAL         24.0                7.18                               CHROMIUM
MATERIAL          0.0             0.73E-3                               AMMONIA
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
COMPOUND          1.0  NITROGEN       3.0  HYDROGEN                      AMMONIA
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA     BLCKHOLE   BLKBODY
ASSIGNMA       VACUUM      VOID
ASSIGNMA      AMMONIA   TARGET3
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..*
ASSIGNMA     CHROMIUM   TARGET1   TARGET2       1.0

* Set the random number seed
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
RANDOMIZ          1.0 54217137.
*
* Set the number of primary histories to be simulated in the run
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
START          1000.
STOP
```

Materials

Materials

FLUKA handles:

- elemental materials (by default natural composition, the user can set a specific isotope, being aware of low energy neutron cross sections availability)
- compounds (chemical molecules, alloys, mixtures…)

FLUKA has a set of predefined materials (see FLUKA manual)

Users can both use/modify these and define their own ones

Basic cards:

MATERIAL	material declaration
COMPOUND	compound definition (a **MATERIAL** card is mandatory for a compound declaration)
ASSIGNMA	material assignment to regions of geometry

In FLUKA 2 special materials +23 natural elements of most common use, e.g. Oxygen, Carbon, Iron… (check them out in the manual, Chap. 5), are predefined

The first two are very important:

- **BLCKHOLE** (mat #1): material with infinite absorbance;
- **VACUUM** (mat #2)

12 compound materials with the composition suggested by ICRU are predefined as well (again, check the manual!)

All predefined materials can be used WITHOUT explicit **MATERIAL** / **COMPOUND** cards

WARNING: user defined **MATERIAL** cards OVERRIDE PREDEFINED materials having the same name

MATERIAL
COMPOUND
ASSIGNMA

Materials

COMPOUND

Each COMPOUND card must be associated to a MATERIAL card
More COMPOUND card can be used to define a compound

- [SDUM] compound name
- [WHAT(1)] amount of the first component ←
- [WHAT(2)] first component material
- [WHAT(3)] amount of the second component ←
- [WHAT(4)] second component material
- [WHAT(5)] amount of the third component ←
- [WHAT(6)] third component material

How to define the "amount"?

content > 0	component material > 0	**ATOM content**
content < 0	component material > 0	**MASS content**
content < 0	component material < 0	**VOLUME content**

MATERIAL

- [SDUM] CHROMIUM, material name
- [WHAT(1)] 24.0, atomic number Z
- [WHAT(2)] atomic weight (**leave it empty**)
- [WHAT(3)] 7.18 g/cm^3, density
- [WHAT(4)] material number (**leave it empty**)
- [WHAT(5)] alternate material to be used for dE/dx (**normally empty**)
- [WHAT(6)] mass number A (**leave it empty unless you want a specific isotope**)

ASSIGNMA

Assign a material to one (or more) region in the geometry
(for the region definition see the geometry lecture or the manual)
A material must be associated to each of the geometry regions, except to those defined as blackhole.

The assigned material could be either a single element material or a compound

- [WHAT(1)] = material index, or material name
- [WHAT(2)] = first region to be "filled" with the material (Default = 2.0)
- [WHAT(3)] = last region to be "filled" with the material (Default = WHAT(2))
- [WHAT(4)] = step length in assigning indices
- [WHAT(5)] = to activate magnetic and electric fields (see manual)
- [WHAT(6)] = assign another material for radioactive decay products transport. As of now, only vacuum and blackhole are allowed.

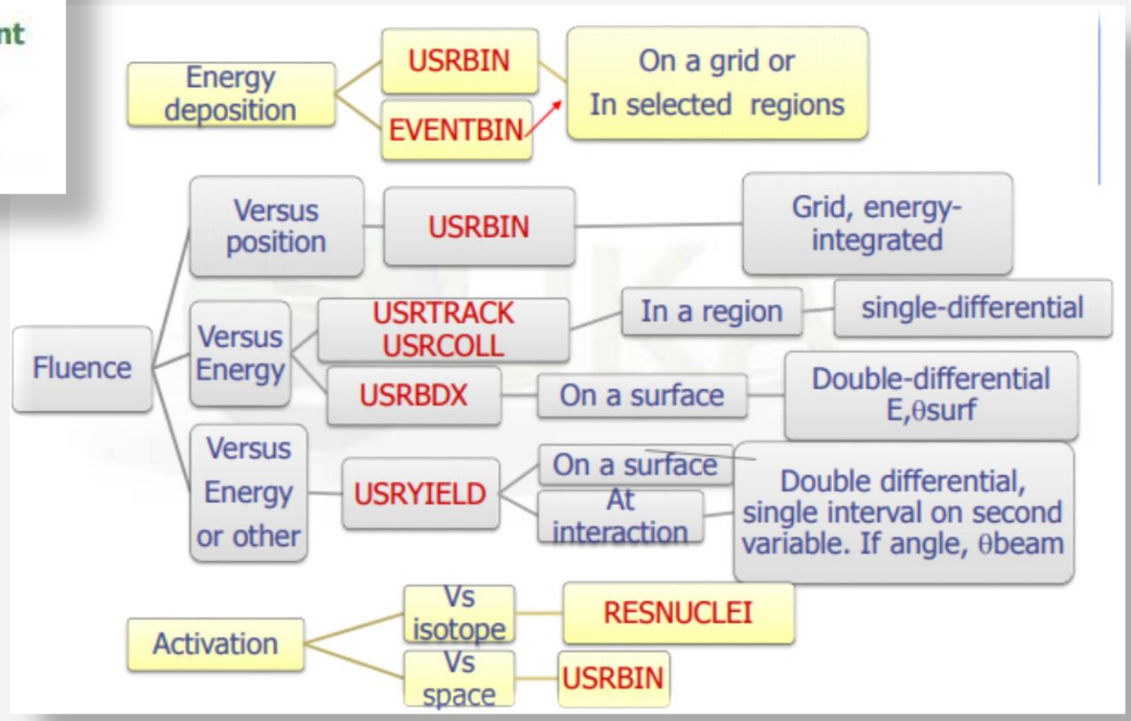
```
TITLE
My Basic Input example
* Set the defaults for precision simulations
DEFAULTS                                                              PRECISIO
* Define the beam characteristics
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAM              3.5 -0.082425      -1.7        0.0        0.0           PROTON
* Define the beam position
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
BEAMPOS           0.0        0.0      -0.1        0.0        0.0
*
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 target1     0.0 0.0  0.0 0.0 0.0 10.0 5.0
RCC target2     0.0 0.0 20.0 0.0 0.0 10.0 5.0
RCC target3     0.0 0.0 40.0 0.0 0.0 10.0 5.0
END
* Black hole
BLKBODY       5 +blkbody -void
* Void around
VOID          5 +void -target1 -target2 -target3
* Target
TARGET1       5 +target1
TARGET2       5 +target2
TARGET3       5 +target3
END
GEOEND
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
MATERIAL         24.0                7.18                               CHROMIUM
MATERIAL          0.0             0.73E-3                               AMMONIA
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
COMPOUND          1.0  NITROGEN       3.0  HYDROGEN                     AMMONIA
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
ASSIGNMA     BLCKHOLE   BLKBODY
ASSIGNMA       VACUUM      VOID
ASSIGNMA      AMMONIA   TARGET3
*
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..*
ASSIGNMA     CHROMIUM   TARGET1   TARGET2        1.0

* Set the random number seed
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
RANDOMIZ          1.0 54217137.
*
* Set the number of primary histories to be simulated in the run
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7..
START          1000.
STOP
```

Scoring

FLUKA estimators

- FLUKA offers numerous different estimators, *i.e.*, directly from the input file the users can request scoring the respective quantities they are interested in.
- Several pre-defined estimators can be activated in FLUKA.
- One usually refers to these estimators as "scoring" capabilities
- Users have also the possibility to build their own scoring through **user routines**, HOWEVER:
  - Built-in scoring covers most of the **common needs**
  - Built-in scoring has been **extensively tested**
  - Built-in scoring takes BIASING **weights automatically into account**
  - Built-in scoring has **refined algorithms** for track subdivision
  - Built-in scoring comes with **utility programs** that allow to evaluate statistical errors



Flair: the user interface

Flair

FLUKA Advanced Interface [http://www.fluka.org/flair]

- **All-in-one** User friendly graphical Interface
- Minimum requirements on additional software
- Working in an intermediate level

Not hiding the inner functionality of FLUKA

Front-End interface:

- Fully featured **Input file Editor**
  - Mini-dialogs for each card, allows easy and almost error free editing
  - Uniform treatment of all FLUKA cards
  - Card grouping in categories and card filtering
  - Error checking and validation of the input file during editing
- **Geometry:** interactive visualization editing, transformation, optimizations and debugging
- **Compilation** of the FLUKA Executable
- **Running** and **monitoring** of the status of a/many run(s)

Flair makes an association of the following extensions:

 ***.flair**

 ***.fluka *.inp**

Store in a single file all relevant information:

- Project notes
- Links to needed files: input file, source routines, output files …
- Multiple runs from the same input file, as well as running status
- Procedures on how to run the code
- Rules on how to perform data merging
- Information on how to post process and create plots of the results

- To open Flair in a command line:

```
flair
```

- To open a specific input:

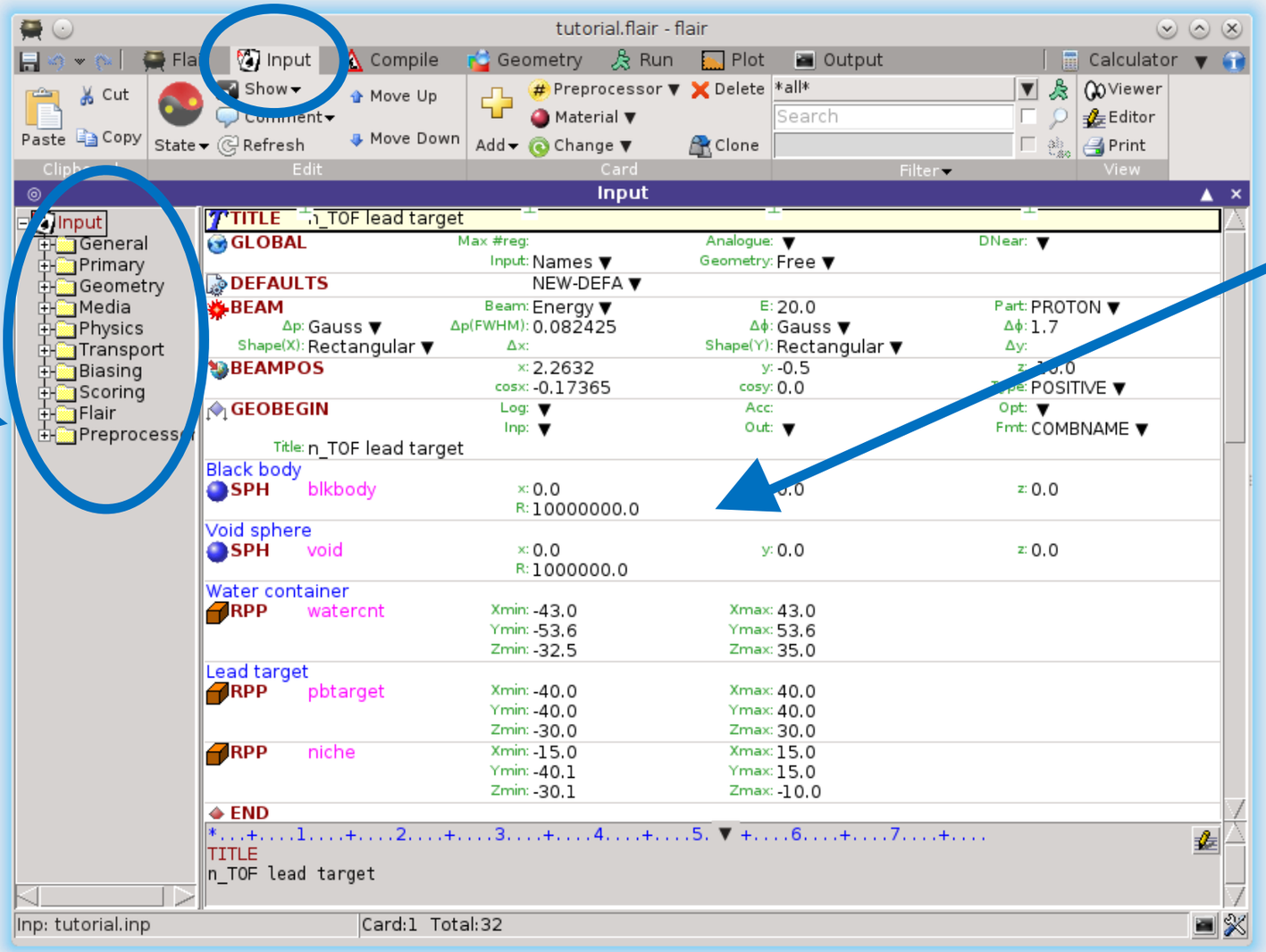
```
flair inputname.inp
```

- To open a specific project:

```
flair projectname.flair
```

Check & modify the input

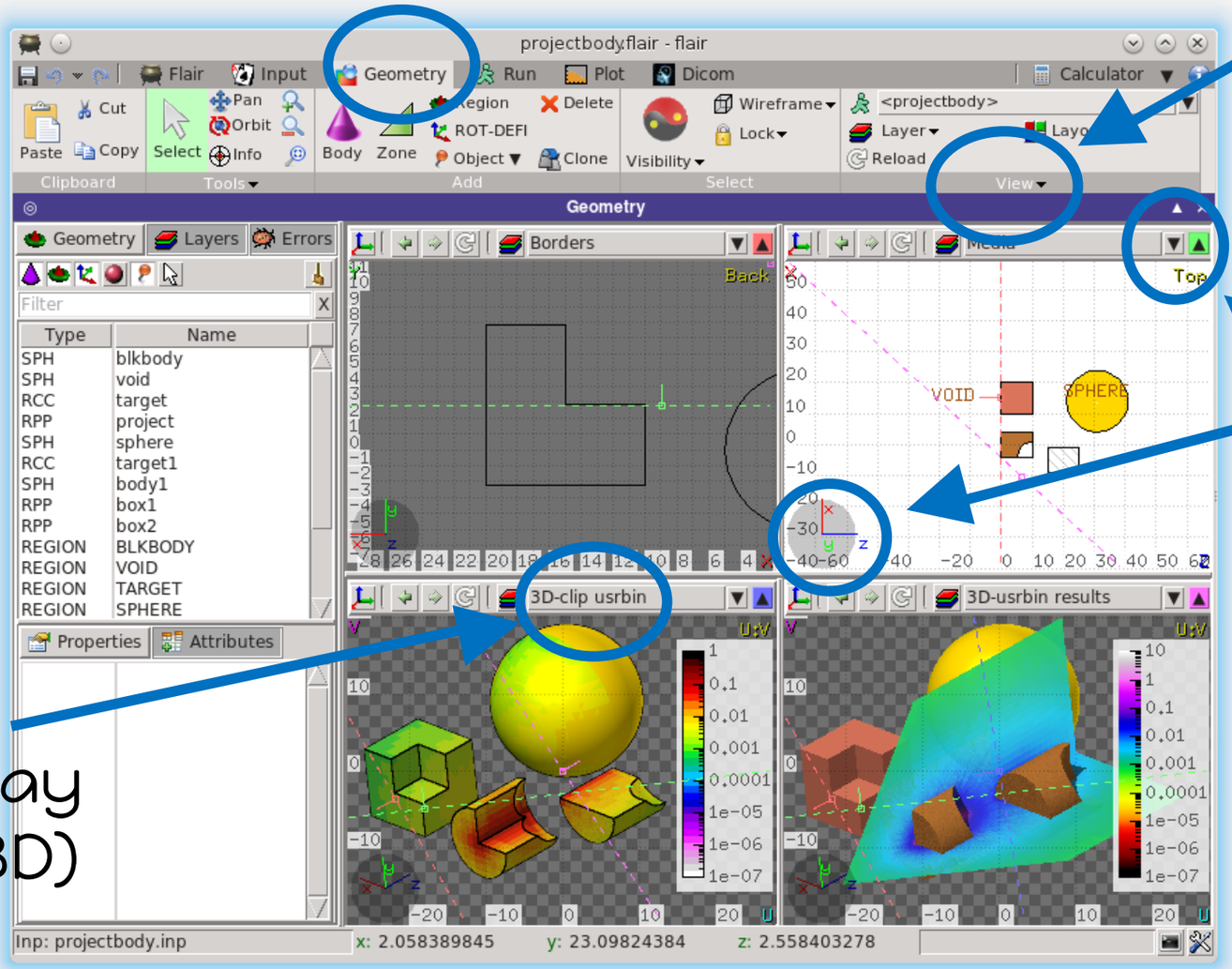
Running throug the input



Input

Access to manual: select the card of interest ad press F1

Have a look at the geo



Save images

View (ex, x-z, y-z...)

Change the display (ex. 2D, 3D)

Running FLUKA

How to run FLUKA: command line

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 input file. It must be a file named ****.inp and .inp has to be omitted.

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.

If everything is OK the temporary directory disappears
And the relevant results are copied in the start directory:

Removing links

by default you have ex100n.log, ex100n.out, ex100n.err and ranex100m (seed for cycle m = n+1)

Removing temporary files

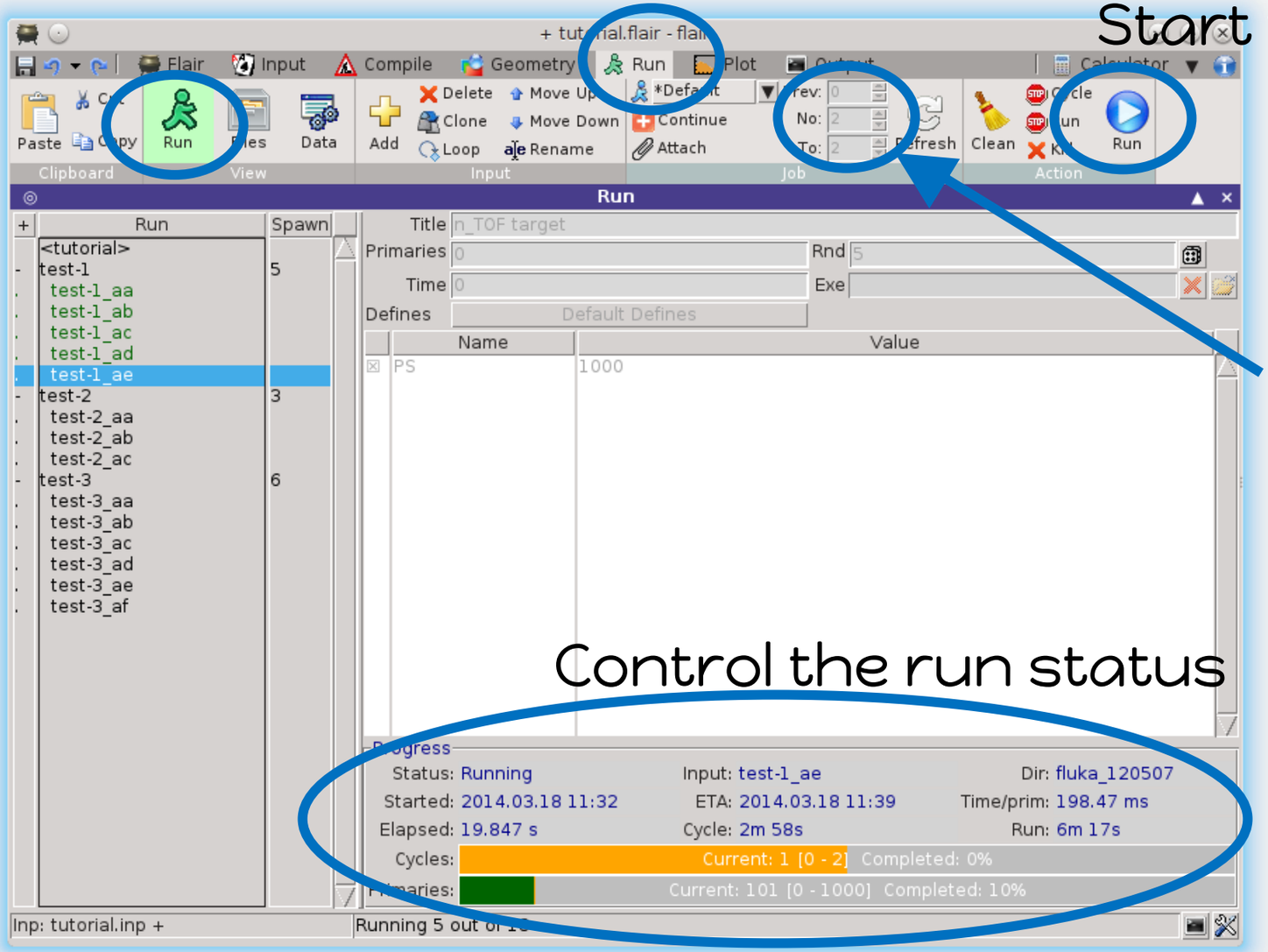
Saving output and random number seed

Saving additional files from scoring requested by the user

```
Moving fort.33 to /home/username/work/ex1/ex1001_fort.33
Moving fort.47 to /home/username/work/ex1/ex1001_fort.47
Moving fort.48 to /home/username/work/ex1/ex1001_fort.48
Moving fort.49 to /home/username/work/ex1/ex1001_fort.49
Moving fort.50 to /home/username/work/ex1/ex1001_fort.50
```

End of FLUKA run

How to run FLUKA: Flair



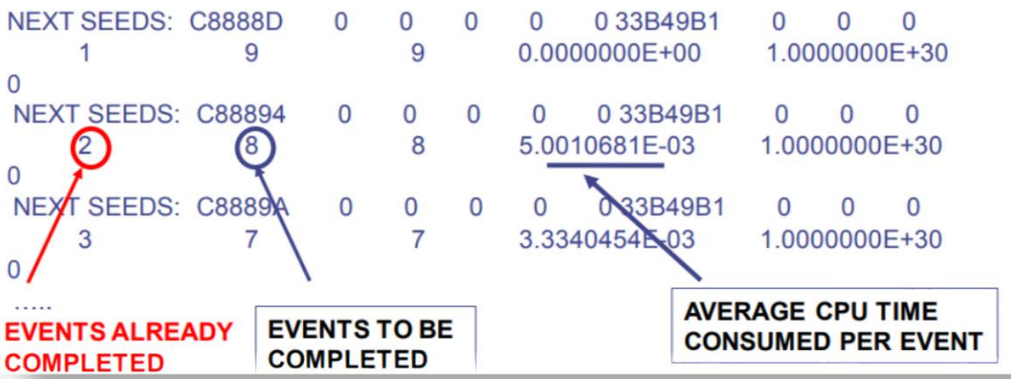
Tips & tricks

CHECK FLUKA DURING THE RUN

Look in the temporary directory:

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

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":



MAKE A CLEAN STOP OF THE 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	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 (5th parameter) for the frequency of these checks!!

FLUKA output files

The Fluka output consists of:

- A main (standard) output, written on logical output unit LUNOUT (predefined as 11 by default) [.out]
- A file with the last random number seeds, unit LUNRAN (2 by default) [ran*]
- A file of error messages, unit LUNERR (15 by default) [.err]
- Any number (including zero) of estimator output files. Their logical unit number is defined by the user [*fort_xx*]
- The available range of logical output numbers is: 21-99
- Generally, the user can choose between formatted (ASCII) and unformatted (binary) scoring (negative or positive sign in the logical unit number). Unformatted scoring is mandatory for the use of provided post-processing utilities.
- Several estimators can be output on the same file (same logical unit) provided they are of the same type
- Possible additional output generated by the user in any user routine;

For example, in F00T simulation the main output file is a txt file [.dat] containing all the quantities of interest