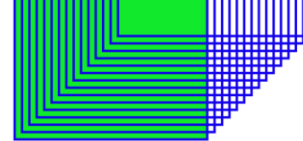


CURSO INTRODUTÓRIO



23 DE JANEIRO
A 8 DE MARÇO
DE 2023

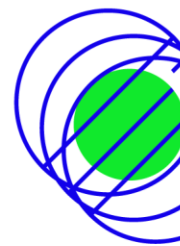
Tópicos essenciais



AULA 04 Materiais

Iniciaremos em breve

Código Monte Carlo de interação e transporte de partículas



01
02
03
04
05
06

01
02
03
04
05
06

2. The FLUKA preprocessor

The FLUKA preprocessor

- A limited, “C”-like preprocessor
- Manipulates the input before execution using directives

- 3 type of directives (starting with: #):
 - Definition:
`#define, #undef`
 - Conditional:
`#if, #elif, #else, #endif`
 - Include:
`#include`

FLUKA preprocessor - *Definition*

Directive: `#define`

- Identifiers without numerical or character value:

④ `#define identifier_name :`

- Used in conjunction with conditional directives (`#if identifier_name ... #endif`)
- *identifier_name* can be up to 40 character long

- Identifiers with numerical or character value:

④ `#define identifier_name : value`

- The *value* can be used in any other input card by referencing `$identifier_name` and can be up to 40 characters long
- Can also be used in conjunction with conditional directives

Directive: `#undef`

④ `#undef identifier_name ▼`

- Deletes a previously defined identifier

FLUKA preprocessor - *Definition*

- Identifiers can be *defined* and *referenced* anywhere in the input file
- Example:

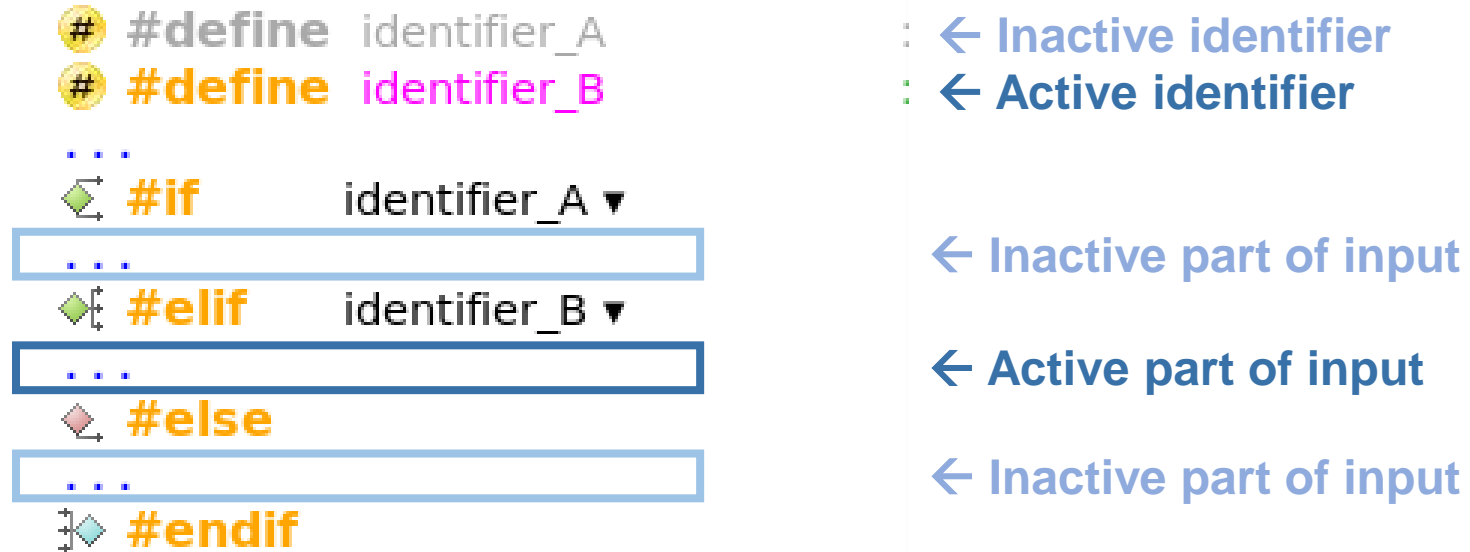
```
# #define Momentum           : 0.1
# #define Particle           : PROTON
* BEAM                       Beam: Momentum ▼
    Δp: Flat ▼                Δp:
Shape(X): Rectangular ▼      Δx:
                               p: $Momentum      Part: $Particle ▼
                               Δφ: Flat ▼        Δφ:
                               Shape(Y): Rectangular ▼ Δy:
```

- **Note:** It is possible to redefine an identifier with a second `#define` directive
 - This is **NOT RECOMMENDED**
 - If an identifier is redefined, the new value is only applied to cards below
 - The output and error files will contain warning messages

FLUKA preprocessor - *Conditional*

Directives: `#if`, `#elif`, `#else`, `#endif`

- To include or exclude parts of the input in conjunction with the `#define` directives



- They work similarly to any programming language
 - Limitation:** Cannot compare values, only test if an identifier is defined or not
- Can be nested
 - Limitation:** Maximum of 10 nesting levels can be used

FLUKA preprocessor - *Conditional*

```
# #define NeutronBeam      :  
...  
# #if NeutronBeam ▼  
  * BEAM Beam: Energy ▼ E: 0.1 Part: NEUTRON ▼  
    Δp: Flat ▼ Δp: Δφ: Flat ▼ Δφ:  
    Shape(X): Rectangular ▼ Δx: Shape(Y): Rectangular ▼ Δy:  
# #else  
  * BEAM Beam: Energy ▼ E: 0.05 Part: PHOTON ▼  
    Δp: Flat ▼ Δp: Δφ: Flat ▼ Δφ:  
    Shape(X): Rectangular ▼ Δx: Shape(Y): Rectangular ▼ Δy:  
# #endif  
...  
# #if NeutronBeam ▼  
  * ASSIGNMA Mat: CONCRETE ▼ Reg: SHIELD ▼ to Reg: ▼  
    Mat(Decay): ▼ Step: Field: ▼  
# #else  
  * ASSIGNMA Mat: LEAD ▼ Reg: SHIELD ▼ to Reg: ▼  
    Mat(Decay): ▼ Step: Field: ▼  
# #endif
```


FLUKA preprocessor - *Conditional*

```
# #define NeutronBeam      :  
...  
#if NeutronBeam ▾  
  # #define Energy        : -0.1  
  # #define Particle      : NEUTRON  
  # #define ShieldMa     : CONCRETE  
#else  
  # #define Energy        : -0.05  
  # #define Particle      : PHOTON  
  # #define ShieldMa     : LEAD  
#endif
```

```
...  
* BEAM          Beam: Energy ▾          E: $Energy          Part: $Particle ▾  
  Δp: Flat ▾      Δp:                    Δφ: Flat ▾          Δφ:  
  Shape(X): Rectangular ▾  Δx:          Shape(Y): Rectangular ▾  Δy:  
...  
* ASSIGNMA     Mat: $ShieldMa ▾        Reg: SHIELD ▾        to Reg: ▾  
  Mat(Decay): ▾      Step:                    Field: ▾
```

FLUKA preprocessor - *Include*

Directive: `#include`

 **#include** <path>/filename.inp ▼

- Includes the specified file to the input
- Can be nested at multiple levels

- The path can be:
 - Relative to the **main input** file
 - Absolute

- Use cases:
 - Split large input files into multiple smaller ones
 - Reuse same input section (beam definition, scoring, etc.) in multiple input files



Materials in FLUKA

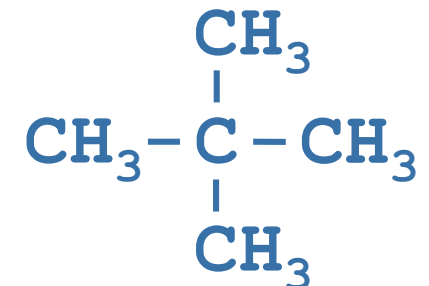
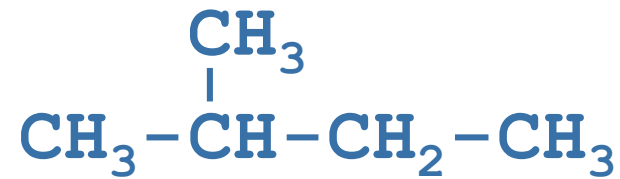
How to define and use (complex) materials in your calculations

Lecture overview

- Introduction
- Pre-defined materials
- Declaring & assigning materials
- Compound materials
- Flair material database
- Special cards

Introduction

- FLUKA assumes that radiation propagates through **homogeneous media**
 - E.g., within a volume occupied by CO₂, the probability of encountering an O atom is the same at any location within the volume (and double that of encountering a C atom)
 - **Exception:** crystals! (particle propagation depends on the direction with respect to the lattice)
- Therefore, **only the stoichiometry and the density matter**, as they define the **atom density per unit volume** for each atomic species (C and O in the above example)
- **Example:** as long as the density is the same, chemical isomers are identical for FLUKA, despite their different molecular structure and other physical properties
 - Pentane, 2-methylbutane, 2,2-dimethylpropane: C₅H₁₂
 - $\rho \approx 6.2 \text{ g/cm}^3$, but different structure, boiling points etc.



Pre-defined materials

- A number of common materials (23 elements and 12 compounds) are pre-defined and **can be assigned** to a region **without the corresponding material declaration**
 - Hydrogen, carbon, oxygen, iron, lead, argon etc. – natural isotopic composition
 - Water, dry air, polyethylene, various biological tissues (bone, muscle etc.)
 - Full lists available in the manual (section 5.2)
- 2 special materials are also included:
 - **VACUUM**: obvious definition. Static electrical fields can be defined only in vacuum.
 - **BLCKHOLE**: Ideal absorber, must be assigned to the “black body” region surrounding your geometry (see Geometry lecture), but can also be used elsewhere in the geometry, e.g. for perfect shielding/collimation, to reduce CPU-time by killing tracking in certain regions etc. – Beware of potential consequences!
- Users may modify the pre-defined materials (except for **BLCKHOLE** and **VACUUM**)
 - **Note**: Using the name of a predefined material for a user-defined material will override the FLUKA definition.

Declaring materials

- A non-predefined (single element) material is declared with a **MATERIAL** card

● **MATERIAL** **FLUORINE** Name
Z: 9 Am: Leave empty ρ: 0.001696 g/cm³
#:
A:

- Each material must have a different name/identifier (8 character limit!)
 - **Note:** this is the case even when e.g. defining different isotopes of an element as materials, such as **Boron-10** and **Boron-11** (as opposed to natural boron)

● **MATERIAL** Boron-10
Z: 5 Am: Use to define specific isotope ρ: 2.34
#:
A: 10 dE/dx: ▼

● **MATERIAL** Boron-11
Z: 5 Am: Use to define specific isotope ρ: 2.34
#:
A: 11 dE/dx: ▼

- **Note:** The choice of material name is important for low-energy neutron transport: a **LOW-MAT** card may be required to assign correct neutron cross-sections to the material. Conversely, using the FLUKA name will automatically call the corresponding cross-sections (if they exist).

Assigning materials

- The **ASSIGNMA** (material assignment) card assigns a material to one or more **regions**
 - **Region**: an enclosed finite volume in our geometry
 - Example: here material **COPPER** is assigned to region **TARGET**
 - **Note**: the assignment can be made to more than one region with the **Reg / to Reg / Step** options

 **ASSIGNMA**

Mat: COPPER ▼ Reg: TARGET ▼ to Reg: ▼
Mat(Decay): ▼ Step: Field: ▼

- It can also activate the **magnetic** or **electric field** (the latter only in vacuum) in the same region(s)
 - To be used in conjunction with a magnetic/electric field definition (see relevant lecture)
 - **Note**: Only one of the two types of field may be defined in a given (vacuum) region
- **Note**: a declared material does not have to be assigned to any region (but it will not be initialised), or it may be assigned only under certain conditions (see lecture on preprocessor commands)

Compound materials

- Composite materials can be defined via the **COMPOUND** card
- Each **COMPOUND** card **must** be associated to a **MATERIAL** declaration through a matching name/identifier

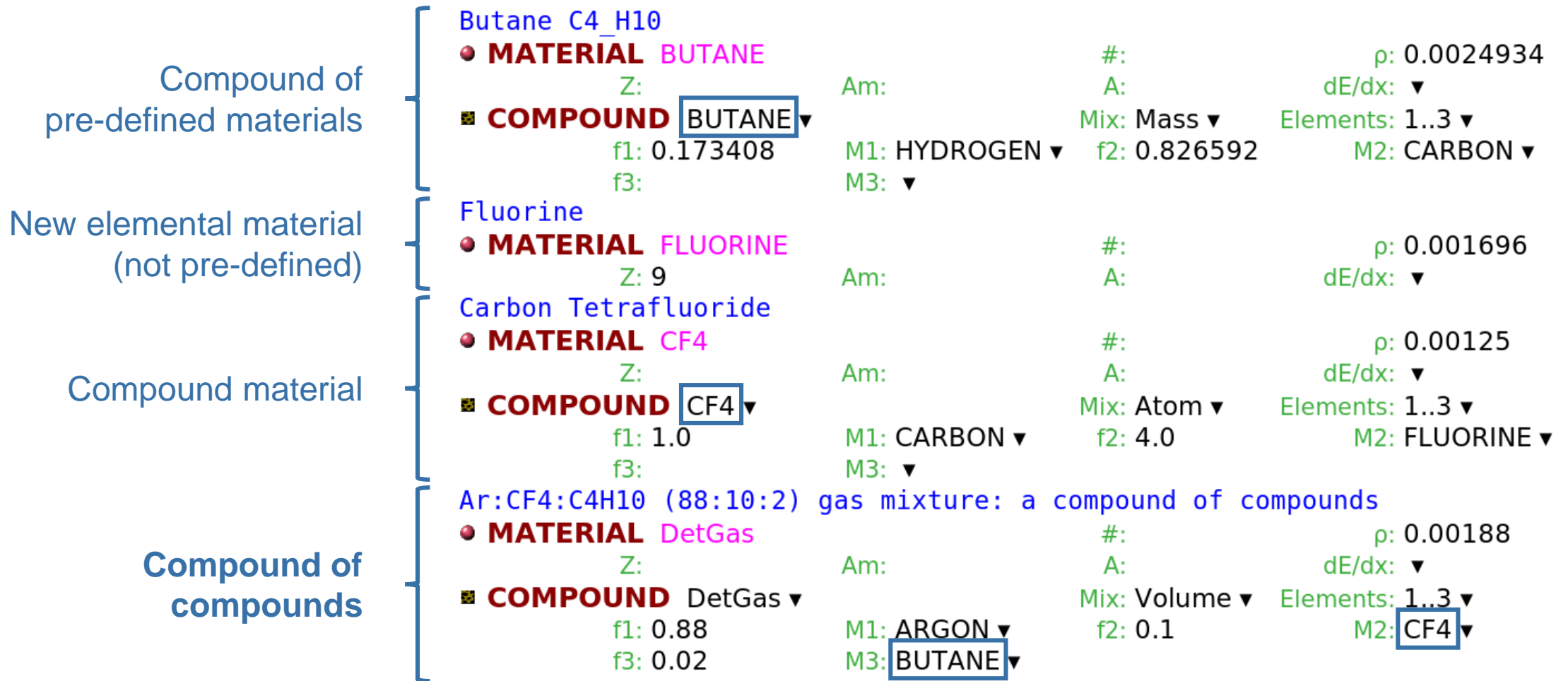
Components and abundances {

- **MATERIAL** CO2
 - Z:
 - Am:
 - #:
 - A:
 - ρ: 0.001842
- **COMPOUND** CO2
 - f1: 1.0
 - M1: CARBON ▼
 - M3: ▼
 - f2: 2.0
 - Mix: Atom ▼
 - f2: 2.0
 - Elements: 1..3 ▼
 - M2: OXYGEN ▼
 - dE/dx: ▼

- The number of components can be increased via the **Elements** option
- The **abundances** can be expressed in terms of:
 - Atom content (as in the above example)
 - Mass fractions
 - Volume fractions
- **Note:** Mass and volume fractions do not need to add up exactly to 1 (or 100), they are automatically scaled

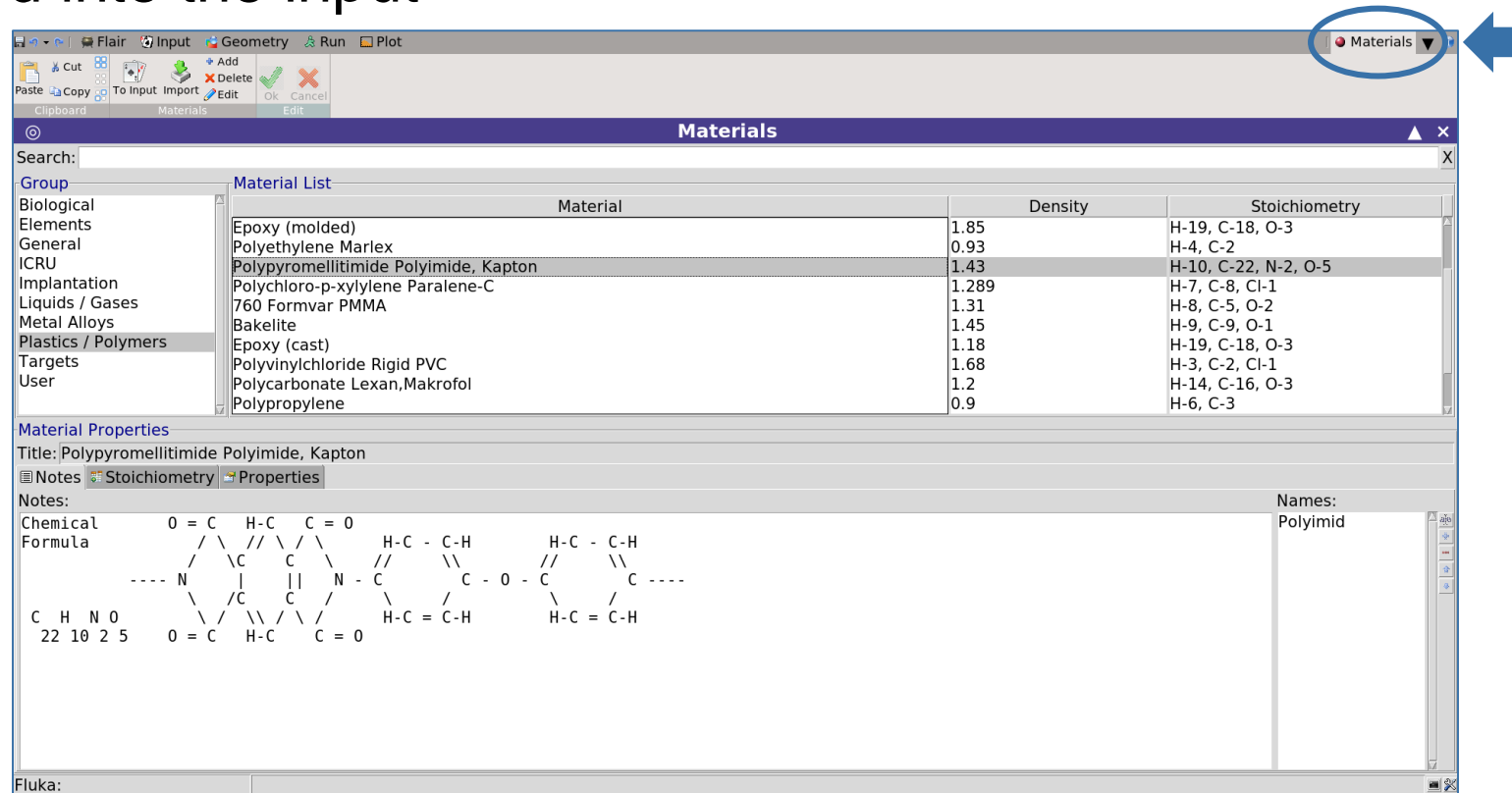
Compound of compounds: an example

- Compound materials can be used as components of other compound materials:



The material database

- Flair comes with an extensive library of materials (elemental and compounds) that can be imported into the input



The screenshot shows the FLUKA Materials database interface. The 'Materials' menu is highlighted with a blue circle and an arrow. The 'Material List' table shows various materials with their densities and stoichiometries. The 'Material Properties' section for Polypyromellitimide Polyimide, Kapton shows its chemical structure and stoichiometry.

Material	Density	Stoichiometry
Epoxy (molded)	1.85	H-19, C-18, O-3
Polyethylene Marlex	0.93	H-4, C-2
Polypyromellitimide Polyimide, Kapton	1.43	H-10, C-22, N-2, O-5
Polychloro-p-xylylene Paralene-C	1.289	H-7, C-8, Cl-1
760 Formvar PMMA	1.31	H-8, C-5, O-2
Bakelite	1.45	H-9, C-9, O-1
Epoxy (cast)	1.18	H-19, C-18, O-3
Polyvinylchloride Rigid PVC	1.68	H-3, C-2, Cl-1
Polycarbonate Lexan, Makrofol	1.2	H-14, C-16, O-3
Polypropylene	0.9	H-6, C-3

Material Properties for Polypyromellitimide Polyimide, Kapton:

Chemical Formula: $C_{22}H_{10}N_2O_5$

Names: Polyimid

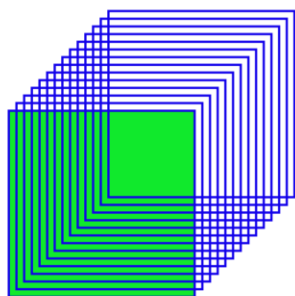
- **Caution:** verify material properties before importing
 - E.g. multiple (similar) versions of the material with different properties may exist

Special material cards (less frequently required)

- The **MAT-PROP** card
 - Modification of density and average ionisation potential, setting threshold for DPAs (Displacements Per Atom), request call to `usrmed.f` routine when particles are transported in specific material
- The **CORRFAC** card
 - Modification of material density for energy loss calculation purposes and nuclear processes on a region-by-region basis
 - Relevant for voxel geometries obtained from CT scans
- The **STERNHE** card
 - Allows to modify the recommended **Sternheimer** density effect parameters used for the stopping power calculations (Sternheimer *et al.*, DOI: [10.1016/0092-640X\(84\)90002-0](https://doi.org/10.1016/0092-640X(84)90002-0)) for a material

Summary

- An introduction to materials in FLUKA, including declaration and definition of (compound) materials, the FLUKA pre-defined materials and the Flair material database
- Essential cards to remember:
 - **MATERIAL**: material declaration
 - **ASSIGNMA**: material assignment to region(s)
 - **COMPOUND**: definition of compound material
- Special cards (used more rarely)
 - **MAT-PROP**, **CORRFACT**, **STERNHEI**



FLUKA

CURSO INTRODUTÓRIO



23 DE JANEIRO
A 8 DE MARÇO
DE 2023

Código Monte Carlo de interação e transporte de partículas

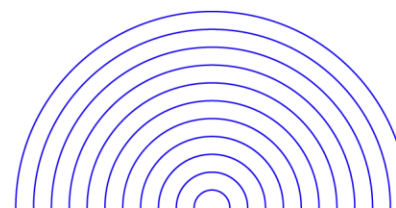
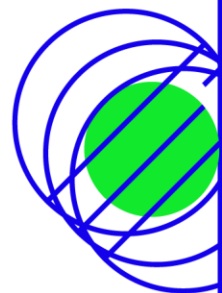
Pausa

Voltamos em 15 minutos

FLUKA

FLUKA

FLUKA



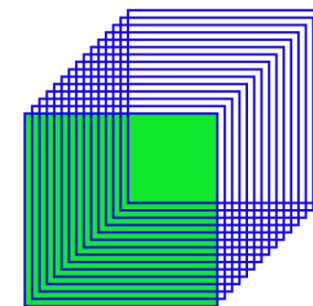
MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO



Tarefa – Aula 4

Iremos testar os conceitos apresentados em aula para montar um composto muito apreciado no Brasil, a caipirinha!

Caipirinha é o coquetel oficial do Brasil feito com **limão, açúcar, gelo e cachaça**. Mas iremos fazer algumas aproximações.

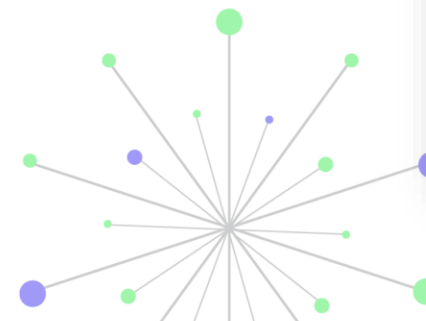


O **gelo** será considerado um item a parte:

- Cubos de água (H_2O) com densidade de $0,92 \text{ g/cm}^3$

A composição do **líquido** deve seguir a proporção abaixo:

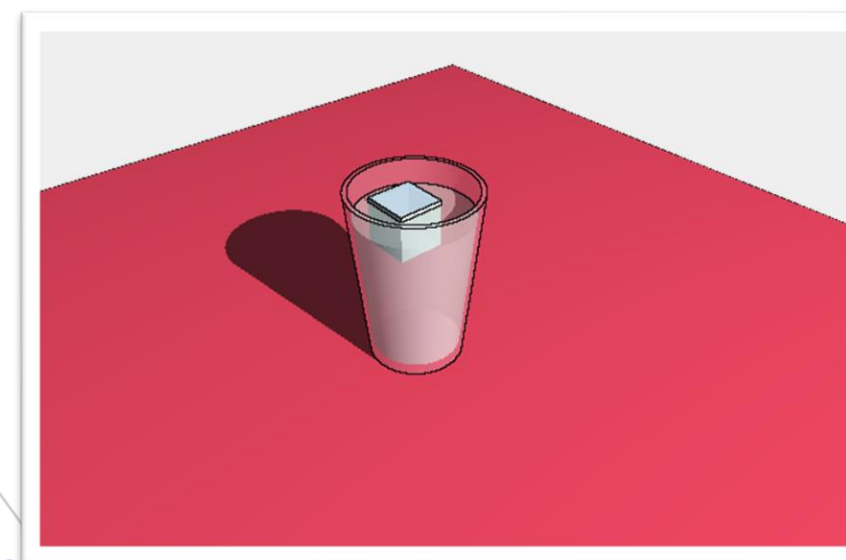
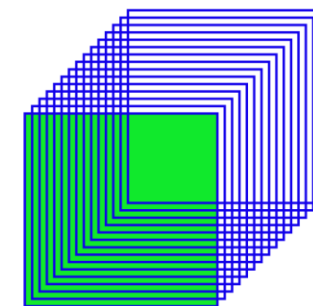
- 60 ml de álcool ($\text{C}_2\text{H}_6\text{O}$ - $0,789 \text{ g/cm}^3$)
- 80 ml de água (H_2O - 1 g/cm^3)
- 2 g ácido cítrico ($\text{C}_6\text{H}_8\text{O}_7$ - $1,66 \text{ g/cm}^3$)
- 24 g açúcar ($\text{C}_{12}\text{H}_{22}\text{O}_{11}$ - $1,5805 \text{ g/cm}^3$)



Tarefa – Aula 4

SOLUÇÃO:

NOVOS MATERIAIS			
ICE - 0.92 g/cm3			
MATERIAL ICE	#:		ρ : 0.92
Z:	Am:	A:	dE/dx: ▼
COMPOUND ICE ▼	Mix: Atom ▼	Elements: 1..3 ▼	
f1: 2.0	M1: HYDROGEN ▼	f2: 1.0	M2: OXYGEN ▼
f3:	M3: ▼		
Ethyl alcohol C2_H5_0_H			
MATERIAL ETHANOL	#:		ρ : 0.7893
Z:	Am:	A:	dE/dx: ▼
COMPOUND ETHANOL ▼	Mix: Mass ▼	Elements: 1..3 ▼	
f1: 0.131269	M1: HYDROGEN ▼	f2: 0.521438	M2: CARBON ▼
f3: 0.347294	M3: OXYGEN ▼		
Citric Acid - 1.66 g/cm3			
MATERIAL ACIDCITR	#:		ρ : 1.66
Z:	Am:	A:	dE/dx: ▼
COMPOUND ACIDCITR ▼	Mix: Atom ▼	Elements: 1..3 ▼	
f1: 4.0	M1: HYDROGEN ▼	f2: 2.0	M2: CARBON ▼
f3: 2.0	M3: OXYGEN ▼		
Sucrose C12_H22_011			
MATERIAL SUCROSE	#:		ρ : 1.5805
Z:	Am:	A:	dE/dx: ▼
COMPOUND SUCROSE ▼	Mix: Mass ▼	Elements: 1..3 ▼	
f1: 0.064779	M1: HYDROGEN ▼	f2: 0.42107	M2: CARBON ▼
f3: 0.514151	M3: OXYGEN ▼		
Capirinha liquid compound			
MATERIAL DRINK	#:		ρ : 0.979
Z:	Am:	A:	dE/dx: ▼
COMPOUND DRINK ▼	Mix: Volume ▼	Elements: 1..3 ▼	
f1: 60.0	M1: ETHANOL ▼	f2: 80.0	M2: WATER ▼
f3: 1.2	M3: ACIDCITR ▼	f4: 15.0	M4: SUCROSE ▼

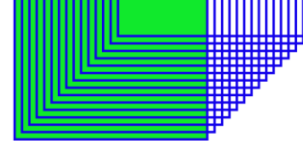


CURSO INTRODUTÓRIO



23 DE JANEIRO
A 8 DE MARÇO
DE 2023

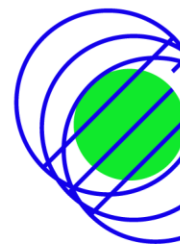
Tópicos essenciais



AULA 04 Materiais

Obrigado pela participação!

Código Monte Carlo de interação e transporte de partículas



01
02
03
04
05
06

23
FLUKA

2023

01
02
03
04
05
06