SaG4n

Simulation of (α,xn) reactions with Geant4

User's Manual

Release 1.1

E. Mendoza, D. Cano-Ott, V. Pesudo, R. Santorelli

GOBIERNO DE ESPAÑA
MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES
Ciemat
Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

Contents

0 Changes from the previous version

Release 1.1 of SaG4n includes the following changes with respect to release 1.0:

1. Materials can now be defined more easily. In the previous version, the abundances of each isotope had to be explicitly defined in the input file. Now the materials can be defined from their elements, assuming natural isotopic abundances. To do this, the atomic number Z of each element has to be given in the **ZAID$_i$** entries, instead of 1000xZ+A. If the material is defined in this way, abundances can be given either as mass fractions (negative **IsoFraction$_i$**) or atomic fractions (positive **IsoFraction$_i$**). More information is provided in Section 2.1.
2. In the previous version of SaG4n all the source alpha particle energies and intensities had to be provided in the input file. Now the alpha energy spectra of the ^{232}Th, ^{235}U and ^{238}U decay series, in secular equilibrium, have been implemented inside SaG4n, and they can be added to the initial alpha energy spectrum in a very easy way (see Section 2.2). The three spectra can be mixed, each with a different weight, and also with other alpha energies.
3. New example input files are provided to show how to implement these new capabilities.
4. Minor additional changes have been implemented in the code, concerning internal checks and extra warnings.

1 Introduction

1.1 Preamble

SaG4n is a Geant4 [1] tool developed to calculate neutron production due to nuclear reactions induced by α particles in different materials. In its inception, it is thought for calculating the neutrons induced by the alphas naturally occurring in materials due to radioimpurities, but it is not restricted to that case.

Neutrons are a potential source of background for all rare-event searches, and are especially relevant for WIMP searches, where they are a source of irreducible background. The precision of the neutron yield is critical to evaluate the discovery potential of coming experiments and to reduce systematic effects in current data.

The code is available in http://win.ciemat.es/SaG4n. Once compiled with Geant4, SaG4n takes from an input file written by the user all the information necessary to define the geometry of the problem, the source, parameters of the physics, the type of output, etc... The program uses the electromagnetic models implemented in Geant4 to perform an explicit transport of the incident α particles through the geometry, and data libraries originally written in ENDF-6 format to model the (α,xn) reactions. The user is able to choose among the available data libraries (JENDL and TENDL, for instance, but also custom made ones). The computation time is reduced by the use of biasing techniques.

Its main features are that it is flexible in the selection of the data libraries to be used, it exploits the advantages of Geant4, it is transparent for the user to implement modifications as desired and it has is easily executed only by editing the mentioned input file.

In particular, the Geant4 capabilities most relevant for this use are three. First, it allows for almost arbitrarily complicated geometries. Second, it benefits from the electromagnetic processes already implemented for α transportation. Third, it allows you to bias a particular process, in this case boosting the (α,xn) cross section in order to reduce the computation time and make the simulation viable for this kind of studies. The amount of biasing and the level of detail in which the transport of the α particles are carried out (via a maximum allowed step size) are controlled by the user. The code allows for the production of γ-rays as well, in coincidence or not depending on the information provided by the data libraries.

The output of the code is editable, but in its native version includes initial position and momentum of the generated α, position and momentum of the produced neutron (and γ rays) and weight. The weight is given by Geant4 to each generated neutron to compensate the biasing that has been introduced to boost the (α,xn) process in first place. The yield of neutrons per alpha can be calculated as

$$Y[n/\alpha] = \frac{1}{N_\alpha}\sum_{i=1}^{N_n}\omega_i$$

where N_n is the number of neutrons produced in the simulation, ω_i the weight of each of them, and N_α the number of simulated α particles.

1.2 Usage

SaG4n is compiled in a similar way to any other Geant4 application. It is first necessary to have Geant4 installed. We recommend the compilation of SaG4n using CMake, with the creation of a build folder. The installation could be similar to the following:

```
$ tar –xzvf SaG4n.01.00.tar.gz
$ cd SaG4n.01.00
$ mkdir build
$ cd build
$ source /home/you/geant4-install/share/Geant4-G4VERSION/geant4make/geant4make.sh
$ cmake -DGeant4_DIR=/home/you/geant4-install/lib64/Geant4-G4VERSION   ../
$ make
```

After compiling the program, an executable named SaG4n should have been created inside the /build/ folder.

Before running the code, some environment variables must be set. They are the following:

```
$ source /home/you/geant4-install/share/Geant4-G4VERSION/geant4make/geant4make.sh
$ export G4NEUTRONHP_DO_NOT_ADJUST_FINAL_STATE=1
$ export G4PHP_DO_NOT_ADJUST_FINAL_STATE=1
$ export G4NEUTRONHP_SKIP_MISSING_ISOTOPES=1
$ export G4PARTICLEHPDATA=/pathtoyourdatalibraries/ENDFLIBING4FORMAT
```

The environmental variable G4PARTICLEHPDATA must be pointing to the full path where is placed the (α,n) data library (JENDL/AN-2005, for example).

To run SaG4n, the name of the executable must be followed by the name of the corresponding input file. For example:

```
$ ./build/SaG4n inputs/examples/geometry/geom01.inp
```

The description of the input file is described in Section 0 and some examples distributed together with the code in Section 3.

2 Input description

The input is a text file containing all the information needed for the simulation.

The input file consists of a series of keywords followed by a variable amount of entries.

The "#" symbol is used to comment. Anything on a line that follows the "#" symbol is not read by the program.

The input file ends with the "**END**" keyword. Anything below this keyword is not read by the program.

Units: distances are in cm, densities in g/cm^3, energies are in MeV.

The entries described below can be written in three types of formats, which are indicated in square brackets. These formats are integer numbers, real numbers, or character strings.

The list of keywords is the following:

Keyword	Description	Optional	Default value
VOLUME	To define the volumes of the geometry	no	
MATERIAL	To define the materials of the geometry	no	
WORLDSIZE	To define the total size of the geometry	yes	1000
SOURCE	To define the source	no	
OUTPUTTYPE	Controls the type of information appearing in the output file(s)	yes	1 0 2
OUTPUTFILE	Name of the output file	yes	./output
MAXSTEPSIZE	Maximum step size for the transport of the alpha particles	yes	0.001
BIASFACTOR	Biasing factor for the cross sections	yes	-1
NEVENTS	Number of histories to be run	yes	100 000
SEED	To set the seed for the random number generator of the Monte Carlo	yes	1234567
INTERACTIVE	If defined, Geant4 runs in interactive mode	yes	not in input
DONOTKILLSECONDARIES	If defined, secondary particles are transported	yes	not in input
END	End of the input	no	

2.1 Geometry

The geometry of the problem is defined with the keywords: **VOLUME**, **MATERIAL** and **WORLDSIZE**.

The keyword **VOLUME** is used to define the volumes of the geometry. Up to 100 different volumes can be defined in the input. For the moment, three types of volumes can be

defined: spheres, parallelepipeds (with the sides parallel to the XYZ axis) and cylinders (with the symmetry axis parallel to the Z axis). Each volume can be inserted either in the *world* volume or inside any of the other defined volumes. The world volume is the largest one, and it contains all the other volumes. In this program the world volume is defined by default and does not have to be explicitly defined in the input file. It consists of a 1000 cm side cube made of vacuum. The size of the cube side can be changed with the **WORLDSIZE** keyword with the following entry:

WORLSIZE WSize

Where:

WSize [real] is the new size of the world cube side, in cm.

Each of the volumes should appear in the input with the following entries:

VOLUME VolID VolName MatID VolType MotherID DummyWord λ_1 λ_2 … $\lambda_{N\lambda}$

Where each of the entries is:

VolID [integer] – Is the volume identification number. Any non-zero integer can be used for this entry. Each volume should have a different **VolID** number.

VolName [string] – Is the name of the volume. The output may use this field to give the results.

MatID [integer] – The identification number of the material from which the volume is made. Materials are defined with the **MATERIAL** keyword. Each material has a different identification number.

VolType [integer] – Defines the type of volume: sphere (**VolType** = 1), parallelepiped (**VolType** = 2) or cylinder (**VolType** = 3).

MotherID [integer] – Defines where the volume is inserted (the mother volume). To insert the volume into the world volume: **MotherID** = 0. To insert the volume inside any other volume, set the **MotherID** number to the **VolID** number of the desired mother volume.

DummyWord [string] – not used at this moment. It is used only to separate the λ_i entries from the rest.

λ_1 λ_2 … $\lambda_{N\lambda}$ [real] – parameters used to define the dimensions and position of the volume. The position of the volume is defined with a translation vector (vx,vy,vz) which corresponds to the translation of the volume with respect to its mother volume. The meaning of the λ-parameters depends on the volume type (all of them in cm):

VolType = 1 (Sphere - **Nλ**=4): **λ_1** is the radius of the sphere (in cm); and vx=**λ_2**, vy=**λ_3**, vz= **λ_4**. The center of the sphere is placed is (vx,vy,vz) with respect to its mother volume.

VolType = 2 (Parallelepiped parallel to the X, Y and Z axis - **Nλ**=6): **λ_1** is the length of the side parallel to the X axis; **λ_2** is the size of the side parallel to the Y axis; **λ_3** is the

size of the side parallel to the Z axis; and vx=**λ4**, vy=**λ5**, vz= **λ6**. The center of the parallelepiped is (vx,vy,vz) with respect to its mother volume.

VolType = 3 (Cylinder parallel to the Z axis - **Nλ**=5): **λ1** is the radius of the cylinder; **λ2** is the length of the cylinder; and vx=**λ3**, vy=**λ4**, vz= **λ5**. The center of the cylinder in (vx,vy,vz) with respect to its mother volume.

General note on the **λ** parameters: Note that full dimensions are used, instead of half-dimensions as usually done by Geant4.

Example:

```
VOLUME 5 Sphere01     10 1 0 --- 20 3 5 7
VOLUME 6 Box01        13 2 5 --- 3 3 3 0 0 0
```

These two lines in the input define two volumes: a sphere made of a material with MatID=10 (has to be defined in the input) and with 20 cm radius, centered in x=3 cm, y=5 cm and z=7 cm, and a 3x3x3 cm^3 parallelepiped centered in the center of the sphere, made of a material with MatID=6 (has to be defined in the input).

The materials are defined with the **MATERIAL** keyword. Up to 100 different materials can be defined in the input file. Each of them has an identification number, which is used to assign them to the volumes. The format to define the materials is the following:

MATERIAL MatID MatName Density NIso
$ZAID_1$ $IsoFraction_1$
$ZAID_2$ $IsoFraction_2$
…
$ZAID_{NIso}$ $IsoFraction_{NIso}$
ENDMATERIAL

Where each of the entries is:

MatID [integer] – Material identification number. Any non-zero integer can be used for this entry.

MatName [string] – Name of the material.

Density [real] – Density of the material, in g/cm^3.

NIso [integer] – Number of isotopes or elements from which the material is made.

$ZAID_i$ [integer] – Atomic number *Z* and mass number *A* of the isotope *i*, written as ZAID = 1000xZ+A. For example, ^{56}Fe has ZAID=26056; and ^{235}U has ZAID=92235. Materials can also be defined from their elements. In this case **$ZAID_i$**=Z, i.e. the atomic number of the element has to be provided. Each material is defined either from its elements, or from its isotopic abundances. Elements and isotopes cannot appear both in the same material.

$IsoFraction_i$ [real] – If **$ZAID_i$** is an isotope (1000xZ+A), then **$IsoFraction_i$** is the atomic fraction of the isotope *i*. If **$ZAID_i$** is an element, then **$IsoFraction_i$** can be either positive or

negative. If positive, **IsoFraction$_i$** is the atomic fraction of element *i*. If negative, **-IsoFraction$_i$** is the mass fraction of element *i*. Negative and positive values cannot appear both in the same material. In any case, the values of **IsoFraction$_i$** don't need to be normalized.

ENDMATERIAL – At the end of each material description, with **NIso** pairs of ZAID and IsoFraction values, the word **ENDMATERIAL** should appear.

Example:

```
MATERIAL 9 H2O 1.00 5
1001 1.99977
1002 0.00023
8016 0.99757
8017 0.00038
8018 0.00205
ENDMATERIAL
```

These lines define a material, which is water with density 1.00 g/cm^3. The identification number is MatID=9. An equivalent definition is:

```
MATERIAL 9 H2O 1.00 2
1 2
8 1
ENDMATERIAL
```

And also:

```
MATERIAL 9 H2O 1.00 2
1 -11.19
8 -88.81
ENDMATERIAL
```

It is the responsibility of the user to make geometries that do not contain errors. In particular, to make geometries in which volumes do not overlap.

2.2 Source

The source is defined with the **SOURCE** keyword. The source are alpha particles emitted uniformly distributed inside a volume. There is one alpha particle per history in the run. They are emitted isotropically. The energies of the alpha particles are sampled according to a discrete probability distribution, given in the input. The following entries have to be provided in the input file:

SOURCE SPosType λ_1 λ_2 … $\lambda_{N\lambda}$
SNorm NS
$DummyWord_1$ $AlphaEnergy_1$ $AlphaIntensity_1$
$DummyWord_2$ $AlphaEnergy_2$ $AlphaIntensity_2$
...
$DummyWord_{NS}$ $AlphaEnergy_{NS}$ $AlphaIntensity_{NS}$
ENDSOURCE

Where each of the entries is:

SPosType [integer] –The source alpha particles are emitted uniformly distributed inside a volume. **SPosType**, together with the λ_i values that follows it, defines the type of volume from which the source is emitted. There are various possibilities:

SPosType = 0: then the source will be emitted from one of the volumes defined in the **VOLUME** field. In this case **Nλ**=1 and λ_1 has to be set to the desired **VolID**. For example, **SPosType** = 0 and λ_1=14 means that the source will be emitted uniformly distributed from volume with **VolID** =14.

SPosType ≠ 0: More general sources can be defined. There are three types of volumes can be defined for the moment, the same as in the **VOLUME** field: a sphere (**SPosType** = 1), a parallelepiped (**SPosType** = 2) or a cylinder (**SPosType** = 3).

λ_1 λ_2 … $\lambda_{N\lambda}$ [real] – parameters used to define the dimensions and position of the source volume. If **SPosType** = 0, then **Nλ**=1 and λ_1 has to be set to the desired **VolID** . Otherwise they have **almost** the same meaning as in the **VOLUME** field. The only difference is that the **translation vector** (vx,vy,vz) is **respect to the world volume** for the source.

SNorm [real] – If positive, this is the normalization factor for the histograms in the output. If not positive, then the normalization factor is the sum of all the **$AlphaIntensity_i$** values. The histograms in the output are normalized to be per source α particle multiplied by the normalization factor.

NS [integer] – Number of discrete alpha particle energies to consider in the source.

$DummyWord_i$ [string] – It is not used by the code at this moment. It is used only to note, if desired, where the alpha line comes from. There are some exceptions (see **Note** below).

$AlphaEnergy_i$ [real] – Energy (in MeV) of the i^{th}-alpha particle.

$AlphaIntensity_i$ [real] – Intensity of the i^{th}-alpha particle. The amount of source alpha particles generated by the program is proportional to this number. If **SNorm** is negative or cero, then this value is also used for normalizing the output histograms.

Note: If **$DummyWord_i$** is set to “Chain_Th232”, “Chain_U235” or “Chain_U238” then the entire alpha particle energies and intensities of the ^{232}Th, ^{235}U or ^{238}U decay series in secular equilibrium, respectively, will be added to the initial alpha energy spectrum. In this case, the entry **$AlphaEnergy_i$** will not be used by the code and **$AlphaIntensity_i$** will be the total intensity of the given decay series.

Example 1:

```
SOURCE 3 10 20 +15 0 0
1.00 3
ThreeMeVAlpha 3.00 1.00
FiveMeVAlpha  5.00 1.00
EightMeVAlpha 8.00 1.00
ENDSOURCE
```

These lines define a source where the α particles are emitted isotropically and uniformly distributed inside a cylindrical volume of 10 cm radius and 20 cm length, with the center in x=15 cm, y=z=0 cm, with its symmetry axis parallel to the Z axis. The α particles are emitted with 3, 5 or 8 MeV, with the same probability.

Example 2:

```
VOLUME 15 Cylinder01 9 3 0 - 10 20 +15 0 0
SOURCE 0 15
1.00 3
ThreeMeVAlpha 3.00 1.00
FiveMeVAlpha  5.00 1.00
EightMeVAlpha 8.00 1.00
ENDSOURCE
```

These lines define a source which is the same as the one in the previous example.

Example 3:

```
VOLUME 15 Cylinder01 9 3 0 - 10 20 +15 0 0
SOURCE 0 15
1.00 3
Chain_Th232 0 30.00
Chain_U235  0 50.00
Chain_U238  0 20.00
ENDSOURCE
```

The spatial distribution of the source is the same as before, but now the energies of the alpha particles are sampled from the ^{232}Th decay series 30% of the times, from the ^{235}U decay series 50% of the times and from the ^{238}U decay series 20% of the times. In all the cases the decay series are in secular equilibrium.

2.3 Output

The information in the output is controlled by two keywords: **OUTPUTTYPE** and **OUTPUTFILE**. The **OUTPUTTYPE** keyword controls the type of information to be

retrieved by the program; and the **OUTPUTFILE** keyword is used to provide the name of the output file(s).

The program can generate two types of outputs. The first is a set of histograms. These histograms are of two types: energy distributions of the neutrons generated in each volume; and alpha particle flux in each volume. Histograms are written to a file either in ROOT format [2] (rootfile) or in text (ASCII) format. There, a neutron energy spectrum histogram and an alpha particle flux histogram will appear for each volume defined in the geometry, with the exception of the world volume. The names of these histograms will be NSpec_XXX and AFlux_XXX, where XXX is the **VolName** of the corresponding volume.

The second type of output is a text file containing a list of the neutrons and gamma rays generated in the calculation. For each neutron and gamma ray, a new line will appear in the output file with its energy, momentum, position …

The **OUTPUTFILE** keyword has to be followed by the following entry:

OUTPUTFILE FName

Where:

FName [string] – Is the base name for the output file. The base name should be provided without extension. The program will create eventually two output files: one root or text file with the histograms, whose name will be the one provided here plus the extension ".root" or ".txt"; and other text file with the list of generated neutrons and gamma rays, whose name will be the one provided here plus the extension ".out".

The **OUTPUTTYPE** keyword has to be followed by the following entries:

OUTPUTTYPE OutType1 OutType2 OutputFormat

Where each of the entries is:

OutType1 [integer] – This entry controls whether or not to write the file with the neutron energy spectra and the alpha particle fluxes. Two possibilities are allowed:

OutType1 = 0 – The file with the histograms is not generated.
OutType1 = 1 – The file with the histograms is generated.

OutType2 [integer] – This entry controls how the text file with the list of generated neutrons and gamma rays is generated. Three possibilities are allowed:

OutType2 = 0 – No file is generated.
OutType2 = 1 – The text file with the list of generated neutrons and gamma rays is generated. Each line in the output file corresponds to a secondary particle. There are 10 columns, with the following information and in this order: event number (history in the run), particle name (neutron or gamma), energy (MeV), weight, position (three entries, corresponding to x, y and z coordinates, in cm), and momentum direction (three entries,

corresponding to the projections of the momentum direction, which is a unitary vector, in the x, y and z axis, respectively) of the secondary particle.

OutType2 = 2 – Same as **OutType2** = 1, but 10 more columns are added, in the following order: Z of the nucleus involved in the reaction, A of the nucleus involved in the reaction, energy of the α particle (in MeV) at the time of the interaction, energy of the α particle (in MeV) at the beginning of the history run, position (three entries, corresponding to x, y and z coordinates, in cm), and momentum direction (three entries, corresponding to the projections of the momentum direction, which is a unitary vector, in the x, y and z axis, respectively) of the α particle at the beginning of the history run.

OutputFormat [integer] – This entry controls the format in which the output files are written. Two options are allowed:

OutputFormat = 1 – The file with the histograms is written in text (ASCII) format (*).
OutputFormat = 2 – The file with the histograms is written in root format.

By default **OutType1** = 1, **OutType2** = 0 and **OutputFormat** = 2.

(*) The format for the histograms written in ASCII is the following. For each histogram, the following lines appear in the file. First, a line with three entries, in this order: the name of the histogram, the number of bins (NBins) and the upper limit of the histogram (in MeV). The lower limit of the histogram is always 0 MeV. This line is followed by NBins+2 lines, each with three entries, in this order: lower limit of the bin (in MeV), bin content, and uncertainty of the bin due to counting statistics.

2.4 Rest of the input

The rest of the input file is given by the following keywords.

BIASFACTOR BFactor

Where:

BFactor [real] – The nuclear cross section in all the volumes defined in the geometry is biased by this factor. If negative or cero, no biasing is applied. This keyword is optional, and the default value of **BFactor** is -1 (no biasing). Given the usual yields of 10^{-9} – 10^{-5} neutrons / α, bias factors in the order of 10^{4} are needed to make the estimation of the yields viable with this approach.

NOTE: The optimal point depends on the case and the user should explore convergence of the results before relying on the results

MAXSTEPSIZE StepMax

Where:

StepMax [real] – Is the maximum allowed step size (in cm) for the transport of the alpha particles. The smaller this value is, the more detailed the transport of alpha particles is made, but the slower the execution of the program. This keyword is optional, and the default value of **StepMax** is 0.001.

NEVENTS NEvents

Where:

NEvents [integer] – Is the number of histories to be run (or number of *events*). This keyword is optional, and the default value of **NEvents** is 100000.

SEED Seed

Where:

Seed [integer] – Is the seed for the random number generator. This keyword is optional, and the default value of **Seed** is 1234567.

INTERACTIVE

No entries are provided after the **INTERACTIVE** keyword. If this keyword is in the input file, then Geant4 runs in interactive mode.

DONOTKILLSECONDARIES

No entries are provided after the **DONOTKILLSECONDARIES** keyword. By default in this program all the secondary particles are not transported but killed when created. If this keyword is in the input file, then secondary particles are transported.

3 Examples

There are four sets of examples in four folders:

- geometry: Special attention is paid to show what can be done in terms of generating the volumes.
- yields: examples for the yield of a ^{232}Th source in different elements are provided.
- components: examples of usage for calculating the yields of the different chains and subchains on simple detector components.

- basic01: additional basic examples, focused in the definition of the materials and the sources.

Here we will only describe in detail the pieces of the code that are relevant for each specific example. The definitions of the volumes are explained in most cases because is probably the part that needs more care.

3.1 geometry

3.1.1 geom01

A box, a sphere and a cylinder centered along the x axis. The source is defined independently but placed coincident with the cylinder.

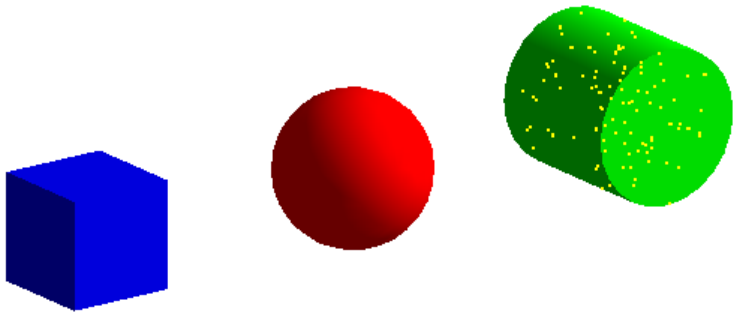

Figure 1: Geometry of the geom01 example visualized with vis01.mac

The first relevant line is:

```
# A sphere made of CH2 placed in the center:
VOLUME 77 Sphere01    7 1 0 - 5 0 0 0
```

A volume is defined with VolID 77 and name Sphere01. It is made of material 7 (CH2, defined later), it is of type 1 (sphere) and its mother volume is 0 (the world). The dash is a dummy word to separate the definition from the dimensions. It is irrelevant and can be whatever word. The sphere has radius of 5 cm and is placed in the (0,0,0) coordinate of the world (the position is with respect to the mother volume).

```
# A parallelepiped made of C placed at -20 cm in the X axis:
VOLUME 49 Box01       8 2 0 - 7 7 7 -20 0 0
```

A volume is defined with VolID 49 and name Box01. It is made of material 8 (C, defined later), it is of type 2 (parallelepiped) and its mother volume is 0 (the world). Its dimensions are L_x = 7 cm, L_y = 7 cm, L_z = 7 cm, and it is placed in the coordinate (-20,0,0).

```
# A cylinder made of H2O placed at +20 cm in the X axis:
VOLUME 54 Cylinder01  9 3 0 - 5 10  +20 0 0
```

A volume is defined with VolID 54 and name Cylinder01. It is made of material 9 (H2O, defined later), it is of type 3 (cylinder) and its mother volume is 0 (the world). Its dimensions are R = 5 cm, L_z = 10 cm, and it is placed in the coordinate (20,0,0). In this first version of the code, cylinders are always defined along the z axis.

```
# MATERIALS:
MATERIAL 7 CH2 0.93 4
1001 1.99977
1002 0.00023
6012 0.9893
6013 0.0107
ENDMATERIAL
```

Material number 7 is named CH2, given a density of 0,93 g/cm^3 and it is stated that will be made of 4 isotopes. First isotope is ^{1}H, which contributes to the compound with an atomic fraction of 1,99977. Then ^{2}H, ^{12}C and ^{13}C atomic contributions are written. Isotopes are defined with the code Z*1000 + A. One can see that in this case the sum of the atomic fractions is 3, which corresponds to the CH_2 molecule that wants to be defined. This is not necessary and, as it is evident by this and the following definitions, it does not need to be normalized either. The number of atoms per unit volume is completely defined by the density and the ratio between the atomic fractions.

```
MATERIAL 8 C 2.266 2
6012 0.9893
6013 0.0107
ENDMATERIAL

MATERIAL 9 H2O 1.00 5
1001 1.99977
1002 0.00023
8016 0.99757
8017 0.00038
8018 0.00205
ENDMATERIAL
```

The rest of the materials are defined following the same principle and do not need specific explanation.

```
SOURCE 3 5 10 +20 0 0
1.00 1
Dummy 8.00 1.00
ENDSOURCE
```

The source is a cylinder (3) of R = 5 cm and L_z = 10 cm placed at (20,0,0), so alphas are emitted from Volume 54, Cylinder01.

3.1.2 geom02

A four-wheeled van, with a central body made of several boxes is defined.

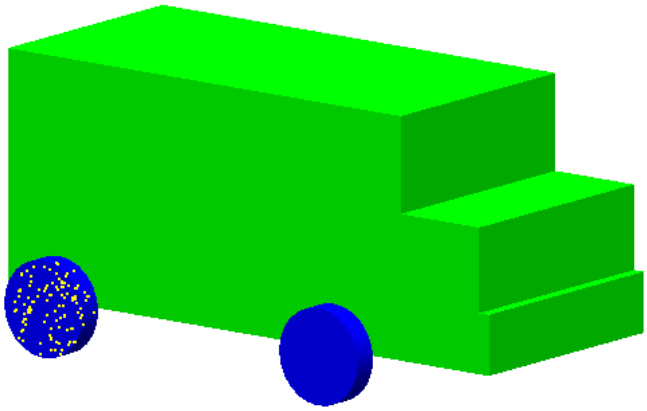

Figure 2: Geometry of geom02 example.

```
VOLUME 1 BigBox   7 2 0 - 20 10 10 0 0 0
VOLUME 2 MedBox01 7 2 0 - 4 6 10 12 -2 0
VOLUME 3 MedBox02 7 2 0 - 0.5  2.5 10 14.25  -3.75 0
VOLUME 4 Wheels01 8 3 0 - 2 1 -7 -5 6
VOLUME 5 Wheels02 8 3 0 - 2 1 +7 -5 6
VOLUME 6 Wheels03 8 3 0 - 2 1 -7 -5 -6
VOLUME 7 Wheels04 8 3 0 - 2 1 +7 -5 -6
```

The main body of the van is defined in the first line, with VolID 1 and named BigBox. It made of material 7 and it's a volume of type 2 (parallelepiped) defined with the world as mother volume (0). Its dimensions are L_x = 20 cm, L_y = 10 cm, L_z = 10 cm, and it is placed in the coordinate (0,0,0).

The car nose is numbered with VolId 2 and defined in the second volume as MedBox01, made of material 7, being a parallelepiped (2) hanging from the mothe volume (0). Its dimensions are L_x = 4 cm, L_y = 6 cm, L_z = 10 cm, and it is placed at (12,-2,0). Coordinate Z is across the paper in Illustration 1. This volume is shifted 12 cm along the X axis (10 cm from the half length of the BigBox + 2 cm from it's own length) to avoid overlaps. The shifting along the Y axis is to maxe its bottom plane coincident with the bottom plane of BigBox.

The car bumper is defined as MedBox02 following the same concept as the nose, so won't be explained here.

The wheels are defined as made of material 8, being cylinders in the mother volume with R = 2 cm and L_z = 1 cm. They are placed at (±7,-5,±6). The "+" sign in X corresponds to

the front wheels and the "–" side to the back wheels. The "+" sign in Z corresponds to the right hand side and the "–" sign to the left hand one. All wheels are in the negative side of y (because they are in the bottom of the van).

3.1.3 geom03

The volume consists of a box, embedded in a cylinder, embedded in a box.

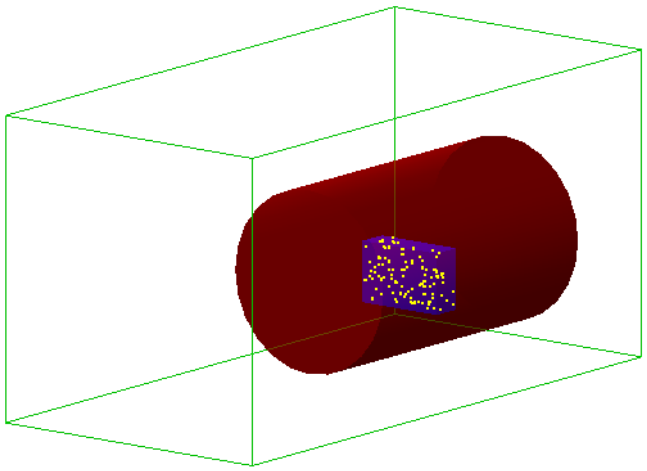

Figure 3: Set of embedded geometries as defined in example geom03. The overlaps are dealt with by Geant4 by defining the volumes hierarchically as mother volumes.

```
VOLUME 12 Box01    7 2 0 - 10 10 20 0 0 0
```

The outer box, Box01, has VolID12, is made of material 7 (CH2) is a parallelepiped (2) and is embedded in the volume 0. Its size is L_x = 20 cm, L_y = 10 cm, L_z = 10 cm and is placed in the center of the world (0,0,0).

```
VOLUME 14 Cylinder01    8 3 12 -  3 10 1 -1 -3
```

The cylinder, with VolId 14 and named Cylinder01, is made of material 8 (carbon) and is indeed a cylinder (3) which is embedded in VolID 12 (which is defined as the mother volume). This is a simple way within Geant4 to define volumes that are completely inside other volumes and not worrying about overlaps. If two volumes are occupying the same volume in Geant4 and don't have a hierarchy, errors are expected.

```
VOLUME 25 Box02    9 2 14 - 3 2 1 0.5 -0.5 0.5
```

The inner box is given VolID 25, named Box02, made of material 9 (H20). It is indeed a parallelepiped (2) with VolID 14 as its mother volume.

The innermost volume, VolID 25, is defined as the alpha source after the definition of the materials:

```
SOURCE 0 25
1.00 1
Dummy 8.00 1.00
ENDSOURCE
```

The first “0” states that a volume already defined will be used as source, and 25 states which volume will be (VolID) . The normalization factor is 1.00 and alphas of one energy will be emitted. This energy is set in the following line, with a dummy introduction word, and the energy (8.00) and its relative intensity. In this case the relative intensity does not make much sense because there aren’t more alphas to weight against, but a number needs to be written in any case.

3.1.4 visualizing macros

If the inputs in the geom folder are run in interactive mode, the respective macros in the vis/ folder can be used to visualize the geometry. For the last case, after typing

$ ./build/SaG4n inputs/examples/geometry/geom03.inp

One can visualize it typing

$ /control/execute inputs/examples/geometry/vis01.mac

The macros might need to be adapted to the Geant4 visualization options of the user.

3.2 yields

The yields for different materials exposed to ^{232}Th decay chain in secular equilibrium. These are some of the results reported in [3]. Only the lines in the MATERIALS and OUTPUTFILE fields change from one example to the other and will be commented here.

```
SOURCE 1 0 0 0 0
1.00 26
90232  4.0123  78.2
90232  3.9472  21.7
90232  3.8111  0.069
90228  5.42315  73.4
…
ENDSOURCE
```

This is a point source defined as sphere (volume of type 1) of radius 0 and placed in the (0,0,0) coordinate. The number to normalize to is 1.00 and 26 different alphas will be emitted by the source, corresponding to the alphas that can be produced in a ^{232}Th decay. In this case the dummy word defines the parent nucleus of the alpha in (Z*1000 + A) format, then the energy (4.0123 MeV in the first case) and it’s relative intensity. Only the first 4 alphas are listed here, because they are enough to clarify the example. The SOURCE field needs to be ended with the ENDSOURCE word.

There are 15 different materials, each of them defined in the input files yield01.inp, yield02.inp,…, yield15.inp. The same inputs but with a different way of defining the materials and the source, are provided in yield01_v2.inp, yield02_v2.inp,…,

yield15_v2.inp. There the materials are defined using elements instead of isotopes, and the ^{232}Th decay chain implemented inside SaG4n is used.

3.3 components

We present the result of evaluating the neutron yield for two relatively simple geometries that are paradigmatic cases of the importance of the interfaces and, hence, full simulations:

1) a copper cable with a fluorine-based insulator,

2) a PCB substrate with copper and liquid argon on each side, and

For each of the examples, the ^{232}Th chain and the different parts of the ^{238}U chain are considered. Here we will only explain one example of each geometry.

One can comment away the different volumes and change the materials to study the effect in the total yield. The simulation shows the sizable effect of interfaces between materials with high (α,n) cross section and materials with small mass which contribution to the neutron budget is often considered negligible.

3.3.1 cable_U238_mid

A cylinder embedded in another cylinder are defined in the VOLUME section.

```
VOLUME 11 CuCable     21 3 12 DummyWord 0.025 2 0 0 0
VOLUME 12 fepInsul    22 3 0  DummyWord 0.1   2 0 0 0
```

The inner one is defined as VolID 11 and called CuCable. It's made of material 21, is indeed a cylinder (3) and is embedded in the volume with VolID 12. Its radius is 0.025 cm and it's length 2 cm. The outer one is defined as VolID 12 and called fepInsul. It's made of material 22, is indeed a cylinder (3) and is embedded in the "world" volume (0). Its radius is 0.1 and its length 2 cm, too. Both of them are placed in the origin of coordinates.

Then in the MATERIAL section,

```
MATERIAL 21 Cu 8.96 2
29063 0.6915
29065 0.3085
ENDMATERIAL
```

the material of the inner cylinder (21) is defined as "Cu", with a density of 8,96 g/cm^3 and having two isotopes, ^{63}Cu and ^{65}Cu -specified in Z*1000 + A format- with 0,6915 and 0,3085 atomic fraction, respectively. Each material needs to be finished with the ENDMATERIAL word.

```
MATERIAL 22 FEP 2.15 3
6012 0.9893
6013 0.0107
9019 3.0
ENDMATERIAL
```

Then material 22, the one corresponding to the outer cylinder is defined as “FEP”. It has density of 2.15 g/cm^3 and is made of 3 isotopes: ^{12}C, ^{13}C and ^{19}F. The atomic fractions state that there is the natural abundance of ^{12}C and ^{13}C and that there are 3 fluorine atoms per C atom.

```
MATERIAL 23 LAr 1.4 1
18040 1.
ENDMATERIAL
```

A third material is defined in case the user wants to study the effect of having this geometry in a LAr bath. The definition of the volume is left as an exercise. In any case, adding this material should only have an effect in the case of very thin insulators.

The source of alphas

```
SOURCE 0 11
1.0 24
```

is defined as one of the existing volumes (by starting with a 0) and in particular VolID 11, the inner cylinder. The yield will be normalized to 1.0 and alphas with 24 different energies will be shooted. Then a dummy word, in this case specifying the parent nuclei, is followed by the energy and the relative intensity of this alpha. It is clear that relative intensities do not need to be normalized.

```
88226  4.78434  93.84
88226  4.601  6.16
88226  4.34  0.0065
88226  4.191  0.001
88226  4.16  0.00027
86222  5.48948  99.92
86222  4.986  0.078
86222  4.826  0.0005
84218  6.00235  99.97890022
84218  5.181  0.00109978
85218  6.756  0.00071928
85218  6.693  0.017982
85218  6.653  0.00127872
86218  7.1292  1.9974e-05
86218  6.5311  2.54e-08
83214  5.516  0.009408
83214  5.452  0.012936
83214  5.273  0.001392
83214  5.184  0.0001464
83214  5.023  5.04e-05
83214  4.941  6e-05
84214  7.68682  99.96550252
84214  6.9022  0.010397504
84214  6.6098  5.99856e-05
```

```
ENDSOURCE
```

The source definition needs to end with the ENDSOURCE word. As the name of the input states, in this case we are simulating what we call the ^{238}U middle chain, with all the alpha emitters between ^{226}Ra and up to the ^{210}Pb "waiting point", with the latter excluded.

3.3.2 pcb_Th232

In this example we define two planar bodies with an interface in Z=0 and submerged in Liquid Argon.

```
VOLUME 11 CuPCB          21 2 13 Dummy  5  5 0.05 0 0  0.025
VOLUME 12 SubstratePCB   22 2 13 Dummy  5  5 0.01 0 0 -0.005
VOLUME 13 LAr            23 2 0  Dummy 10 10   10 0 0  0
```

First, a volume with VolID 11, called "CuPCB" is defined. It is made of material 21, is a parallelepiped embedded inside the volume with VolID 13. Its dimensions are 5 x 5 x 0,05 cm^3 and is shifted by the vector (0, 0, 0.025). This shifting in Z is introduced to place the interface between materials at Z=0.

The second volume, with VolID 12 and called "SubstratePCB" is made of material 22, is a parallelepiped and is also embedded in VolID13. Its dimensions are 5 x 5 x 0,01 cm^3 and is shifted towards negative Z by the vector (0, 0, -0.005) for the already explained reason. Note that there are no overlaps in the geometry at this point.

The third volume, with VolID 13 and called "LAr", is made of material 23, is a parallelepiped and hangs directly from the WORLD volume (its mother volume is 0). Its dimensions are 10 x 10 x 10 cm^3 and sits in the origin of coordinates. The other two geometries that were formerly defined have overlaps with this volume, but since the latter is actually defined as their mother volumes this induces no conflict.

Note that the volume section does **not** need any specific ending word.

Only MatID 22 will be described in this section (see the explanation of the other examples for the other ones).

```
MATERIAL 22 kevlar 1.44 9
1001 9.998
1002 0.002
6012 13.8502
6013 0.1498
7014 1.992
7015 0.008
8016 1.9952
8017 0.0008
8018 0.004
ENDMATERIAL
```

Material with ID 22 is named "kevlar", has a density of 1,44 g/cm^3 and is made of 9 isotopes, ^{1}H, ^{2}H, ^{12}C, ^{13}C, ^{14}N, ^{15}N, ^{16}O, ^{17}O and ^{18}O. The atomic fractions are calculated to account for the isotopic abundances and for the average formula of kevlar ($C_{14}H_{10}N_2O_2$).

The source is placed in one of the already defined volumes (by starting with a 0), and in particular VolID 11 is chosen, the CuPCB. The yield will be normalized to 1.0, meaning that the results will be in neutrons per alpha in the chain, and 26 alphas are considered. Each of them is defined starting with a dummy word, which in this case states the parent nucleus, and followed by its energy and relative intensity. Relative intensities do not need to be normalized.

```
SOURCE 0 11
1.0 26
90232  4.0123  78.2
90232  3.9472  21.7
90232  3.8111  0.069
90228  5.42315  73.4
90228  5.34036  26.0
90228  5.211  0.408
90228  5.173  0.218
90228  5.138  0.036
88224  5.68537  94.92
88224  5.4486  5.06
88224  5.161  0.0071
88224  5.051  0.0076
88224  5.034  0.003
86220  6.28808  99.886
86220  5.747  0.114
84216  6.7783  99.9981
84216  5.985  0.0019
83212  6.08988  9.746928
83212  6.05078  25.125654
83212  5.768  0.61098
83212  5.626  0.0564258
83212  5.607  0.406122
83212  5.481  0.0046722
83212  5.345  0.0003594
83212  5.302  3.9534e-05
84212  8.78486  64.06
ENDSOURCE
```

In this case the source is the ^{232}Th chain. The source definition must end with the ENDSOURCE word.

3.4 basic01

There are two types of input examples in this folder.

In concrete01.inp the alpha particles of the ^{232}Th decay series in secular equilibrium are emitted inside a concrete block. The output is a root file with the generated neutron energy spectrum and the list-mode file with the generated neutrons and γ-rays. The input file concrete02.inp is exactly the same as concrete01.inp, but the concrete material has been defined using mass fraction abundances instead of atomic fractions. These two examples can be used to check that the same result is obtained in both cases.

In source01.inp alpha particles are generated inside a block of CF_2, and the output is a root file with the generated neutron energy spectrum. The energy distribution of the source alpha particles is a mix of the ^{232}Th (30%), ^{235}U (50%) and ^{238}U (20%) decay series in secular equilibrium. The normalization factor of the source is 1. On the other hand, source02.inp, source03.inp and source04.inp are the same as source01.inp, but the source contains only the ^{232}Th (source02.inp), ^{235}U (source03.inp) or ^{238}U (source04.inp) spectra. The normalization factors are in these cases 0.30, 0.50 and 0.20, respectively. These four inputs can be used to check that the results obtained from source01.inp are the same as the sum of the results from source02.inp, source03.inp and source04.inp.

4 References

[1] S. Agostinelli et al., Nucl. Instrum. Methods A 506, 250 (2003).

[2] Rene Brun and Fons Rademakers, Nucl. Inst. and Meth. A 389, 81 (1996). See also http://root.cern.ch/

[3] E. Mendoza et al., Nucl. Instrum. Methods A 960, 163659 (2020).