

CRADLE++ Documentation

Customisable RAdioactive Decay for Low Energy Particle Physics:
A C++ event generator

leendert.hayen@fys.kuleuven.be

February 15, 2016

1 Introduction

This C++ package attempts to provide a general event generator for low energy particle physics, specifically nuclear spectroscopy studies. Its main focus lies on weak interaction studies perpetuated by nuclear beta decay. Its goals are accessibility and extendibility.

The general structure of the cradle CRADLE++ program is as follows

DecayManager: Performs the main loop, keeps the particle stack, particle factory and the registered DecayModes.

↳ Particle: Contains physical information (mass, charge, ...), kinematical information, and array of DecayChannels.

↳ DecayChannel: Contains intensity, parent and daughter excitation energy, Q value, lifetime and DecayMode.

↳ DecayMode: Dynamics of the specific decay. Inherited by daughter classes BetaMinus, Alpha, etc. Parent class performs TwoBodyDecay() and ThreeBodyDecay(), daughter classes perform particle creation and initialisation.

2 Prerequisites

We attempted to use as few external libraries as possible without slowing down the coding progress. Two libraries were used. Both are extensive and widespread C++ libraries, typically installed by default on Linux systems.

1. BOOST: The linear algebra routines of uBLAS, and the program_options library are extensively used. (<http://www.boost.org>)
2. Gnu Scientific Library (GSL): Used for the calculation of the calculation of the complex Gamma function in the Fermi function. (<http://www.gnu.org/software/gsl/>)

When using the precompiled binary, the libraries are statically included, and require only a Linux 64-bit system to run. Building requires the boost libraries to be installed, if not done already. This can be done easily using the package manager `sudo apt-get`

`install libboost-dev` (replace `apt-get` for any other package manager on your system).

CRADLE++ uses the data files from NNDC and ENSDF databases, as compiled for the Geant4 distribution. These can be downloaded at <http://geant4.web.cern.ch/geant4/support/download.shtml>. CRADLE++ uses the G4RadioactiveDecayX.X and G4PhotonEvaporationX.X data files.

CRADLE++ was tested on Ubuntu 14.04 LTS, and Ubuntu 15.10 in a fresh virtual machine install.

3 Configuration

All options can be seen by running the program with the `-h` [`-help`] flags.

CRADLE++ requires a configuration file on start-up to set several parameters, such as weak interaction coupling constants, general verbosity or lifetime cuts. Options are written in typical INI format. By default `config.txt` in the local folder is loaded. This behaviour can be overridden using command line options.

The location of the data files can be set using the environment variables `CRADLE.GammaData` and `CRADLE.RadiationData`. In case these are not defined, CRADLE++ looks for them in the `GammaData` and `RadiationData` local folders.

4 Usage

CRADLE++ performs the full decay chain until it arrives at a state with a lifetime longer than the value `Cut.Lifetime` in seconds specified in the configuration file. It starts from an initial state with a name, Z value, A value and an excitation energy. The initial state is at rest. Beta decay is currently the only `DecayMode` where angular correlations have been implemented. Each decay is performed in the rest frame of the mother and afterwards Lorentz boosted into the lab frame. In the calculation natural units are used, i.e. $\hbar = c = m_e = 1$. The standard energy scale is keV. Output is by default written to `output.txt`. This behaviour can be overridden with the command line options.

The structure of the output file is as follows:

- Column 1: Event ID
- Column 2: Time of particle creation in seconds. The decay of the parent is included.
- Column 3: Particle name
- Column 4: Excitation energy
- Column 5: Kinetic energy (keV)
- Column 6-9: Four momentum of emitted particle in lab frame

Particles are emitted in a 4π solid angle and momentum conservation is guaranteed.

5 Example

We study the proton spectrum after emission from $^{32}\text{Cl}(E_X = 5046.3 \text{ keV})$ after β^+ decay from ^{32}Ar for both scalar and vector interactions. Setting `Couplings.CS` and `Couplings.CV` to 0 and 1 in `config.txt`, respectively, we run CRADLE++ as follows

```
./Cradle++ -n 32Ar -z 18 -a 32 -l 500000 -f 32Ar_vector.txt
```

Doing the same thing, but reversing the two coupling constants we get the scalar recoil spectrum. Both are plotted in figure 1. The gamma rays from the decay are shown in figure 2.

Note: This is a special case, as nucleon emission is not included in the Geant4 code-base, and so it is not present in its data files. It can however be added manually without any issue. See the README file of the G4RadioactiveDecayX.X for the structure of the data files. Proton decay can then be trivially included by adding

```
# Denotes metastable state; Excitation Energy; Lifetime
P      5046.3      0.0000e+00
# Decay modes from this state; Name; Daughter Excitation; Intensity; Q
      Proton  0.0000e+00      1.0000e+02      3.4580e+03
```

to the data file `z17.a32`. The intensity (here set to $J_p = 100.0$) is used to choose the decay mode via the probability

$$P(p) = \frac{J_p}{\sum_i J_i} \quad (1)$$

where J_i denotes the intensity of the i -th decay channel from the current state. This means it competes with γ decay from the same level. Isolating the proton emission can be done simply by increasing J_p .

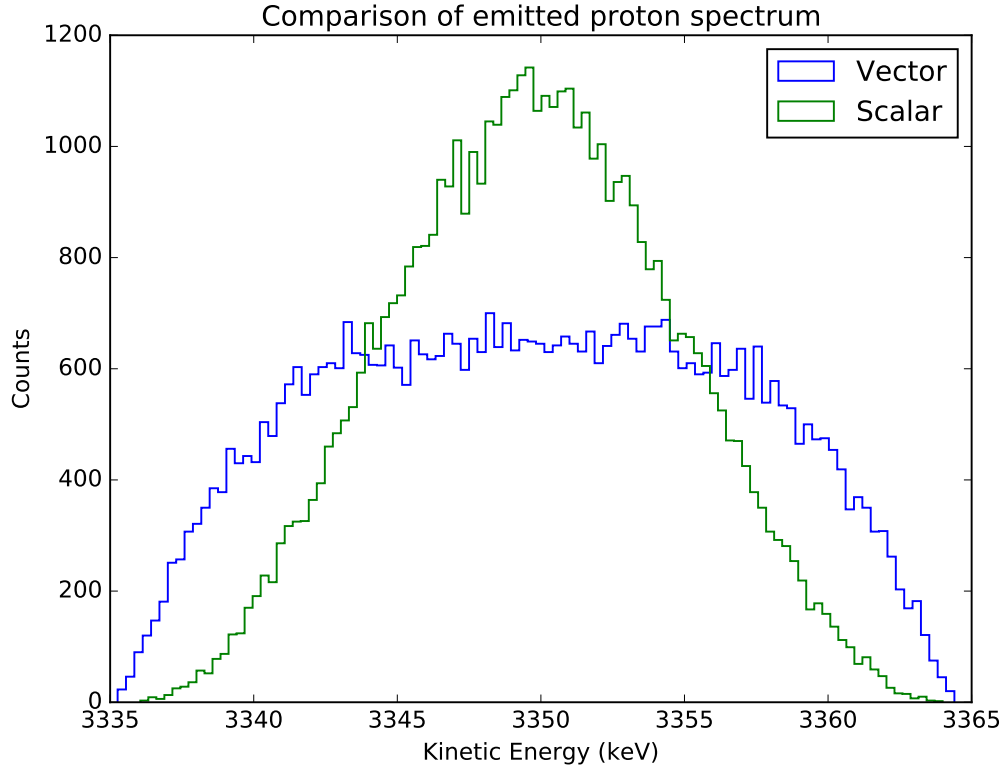


Figure 1: Comparison of the proton peak after emission from ^{32}Cl for a vector and scalar interaction.

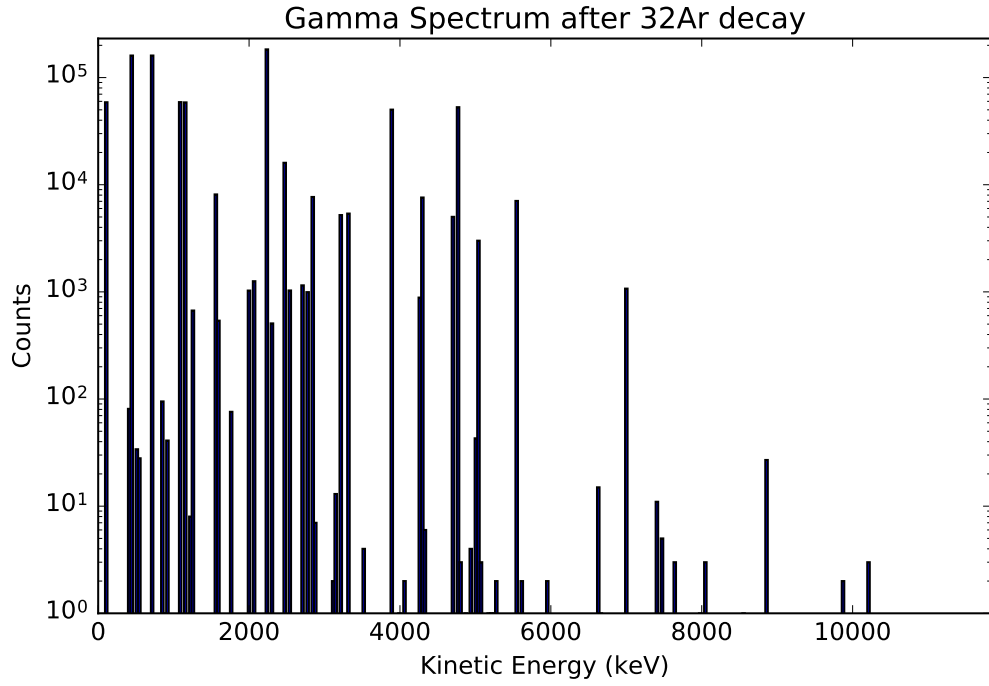


Figure 2: Gamma spectrum after β^+ decay of ^{32}Ar . The highest γ energies above 10 MeV come from very weakly populated levels in ^{32}P after β^+ from ^{32}Cl .