Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

PᴇɴNᴜᴄ: Monte Carlo simulation of the decay of radionuclides☆

E. García-Toraño [a], V. Peyres [a], F. Salvat [b],*

[a] *Laboratorio de Metrología de Radiaciones Ionizantes, CIEMAT, Avda. Complutense 22, 28040 Madrid, Spain*
[b] *Facultat de Física (FQA and ICC), Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Catalonia, Spain*

A R T I C L E   I N F O

A B S T R A C T

The Fortran subroutine package PᴇɴNᴜᴄ that performs Monte Carlo simulation of the decay of radioactive nuclides is described. The adopted nuclear decay characteristics (possible disintegration modes and branching ratios, energies and half-lives of nuclear energy levels, types and energies of emitted particles) are from the ɴᴜᴄʟᴇɪᴅᴇ evaluated database (http://www.nucleide.org/DDEP_WG/ DDEPdata.htm) in the specific format PenNuc. The initial energies of electrons and positrons emitted in beta disintegrations are sampled from their continuous spectra by means of the RITA (rational inverse transform with aliasing) algorithm, which is described in the Appendix. Electronic relaxation of the residual ion after electron capture and internal conversion is simulated by using the data and sampling methods of the ᴘᴇɴᴇʟᴏᴘᴇ code system. At each call the subroutines deliver a list of state variables of the particles (photons, electrons, positrons and alphas) emitted in a random decay path of the considered radionuclide down to a metastable level or to the ground state of the daughter nucleus. The distribution package includes a main program that simulates the emission spectrum of a radionuclide and, optionally, produces a list of particles emitted in a prescribed number of decays.

Program summary
Program Title: PenNuc
Program Files doi: http://dx.doi.org/10.17632/dy28h8xmkv.1
Licensing provisions: CC by NC 3.0
Programming language: Fortran 90/95
Nature of problem: PᴇɴNᴜᴄ simulates the emission of energetic particles in random decay pathways of radionuclides. The initial state of a particle is described by the following variables: type of particle (photon, electron, positron, or alpha), energy, direction, and emission time relative to the start of the decay. For each decay the program provides a list of initial states of the particles emitted with energies higher than the adopted cutoff ECNUC, which is set to 200 eV in the original source file.
Solution method: The nuclear disintegration mode and the subsequent de-excitation of the daughter nucleus are simulated by using decay information from the ɴᴜᴄʟᴇɪᴅᴇ evaluated databases. De-excitation of ions produced in electron capture and internal conversion is simulated by using atomic transition data and sampling methods from the Monte Carlo code system ᴘᴇɴᴇʟᴏᴘᴇ [1]. The energy of beta particles is sampled using the RITA (rational inverse transform with aliasing) algorithm, which optimizes both simulation speed and accuracy.
Additional comments: Although PᴇɴNᴜᴄ can describe several radionuclides simultaneously, the selection of the active nuclide (appropriate to the characteristics of the radioactive source) must be done in the main program. The subroutines provide the complete set of particles emitted with energies higher than the cutoff ECNUC in each nuclear decay, including x rays and Auger electrons.
[1] F. Salvat, ᴘᴇɴᴇʟᴏᴘᴇ-*2014: A code System for Monte Carlo Simulation of Electron and Photon Transport*, OECD/NEA Data Bank, Issy-les-Moulineaux, France, 2015. Available from http://www.nea.fr/lists/ penelope.html.

1. Introduction

The simulation of experimental set-ups with radioisotopes that release multiple particles, usually correlated in time, requires specific procedures for sampling random decay pathways and accounting for the time evolution of particle histories. Even in

measurements of nuclides with simple decay schemes, such as ^{60}Co, the recorded energy spectra may show sum peaks and other structures that result from time-correlated emissions. In addition, time correlations may have a direct influence on coincidence measurements.

Most codes do not allow radioactive sources to be considered. Normally, to include this option the user must define the decay scheme of the active nuclide and provide a dedicated subroutine that generates random decay pathways. The Fortran subroutine package PENNUC implements the practical solution described in Ref. [1]. These subroutines simulate nuclear decay using information from the NUCLEIDE [2] database, which contains evaluated data of 219 radioactive nuclides of practical interest in nuclear safety studies, environmental surveys, nuclear medicine, *etc*. The web application http://www.nucleide.org/DDEP_WG/DDEPdata.htm allows the download of a text (ascii) file with the nuclear decay characteristics (lifetimes of nuclear levels, possible nuclear transitions, branching ratios, type and energy of emitted particles, …) of a radioisotope in the specific format "PenNuc" that is readable by the PENNUC subroutines.

Electron capture and internal conversion produce the ionization of the target atom; these processes preferentially involve electrons in inner subshells and leave the residual ion in a highly excited electronic state. Subsequently the ion de-excites through a cascade of radiative and non-radiative transitions, with the emission of x rays and Auger electrons. The PENNUC subroutines simulate the electronic de-excitation of ions by using the physical model and sampling algorithm of the Monte Carlo transport code PENELOPE [3], which provides a more detailed description of the process than the NUCLEIDE data files.

The subroutine package PENNUC is stand-alone and can be used in combination with any radiation-transport Monte Carlo code. The source file `pennuc.f` consists of a steering sampling subroutine, an initialization subroutine, a random number generator [4], and tools for the simulation of beta emission and of the electronic de-excitation of ions. The distribution bundle includes a simple main program named `decays.f` that simulates random decays of a radionuclide and optionally can produce a file with a list of types, initial energies, and emission times of the emitted particles. The emission times are relative to the start of the decay of the active nucleus; PENNUC does not account for variations of the source activity with time. The program `decays.f` generates the energy spectra of emitted gammas, electrons, and positrons. It can also produce a file with a list of particles emitted in a large number of decays, which provides a detailed description of the emitted radiation. This list can be used to model radioactive sources in a general-purpose radiation transport code, and in other calculations.

Strictly speaking, the PENNUC subroutines simulate the decay of individual nuclides, although they permit various radionuclides to be considered simultaneously. With the aid of a simple main program (to be provided by the user) the subroutines can be effectively used to describe radioactive sources with a mixture of radioisotopes, as well as decay chains (that is, sequential transformations of radioactive decay products).

The present article is organized as follows. The physics models and sampling methods implemented in the PENNUC subroutines are described briefly in Section 2. The structure and operation of the subroutines and the program `decays.f` are explained in Section 3. Sample emission spectra generated by running `decays.f` are presented in Section 4. The adaptive numerical sampling algorithm RITA (rational inverse transform with aliasing), which is used to sample the energy of beta particles from their continuous spectra, is described in the Appendix.

2. Method

The PENNUC subroutines simulate the decay of the active nuclide as a random sequence of transformations and transitions. The first stage is the disintegration of the parent nucleus either by beta or alpha emission or by electron capture; metastable nuclides may alternatively decay to their ground state. The daughter nucleus is usually in an excited level, and it de-excites through transitions by gamma emission or internal conversion. After electron capture and internal conversion, the de-excitation of the electronic cloud is simulated. The output of the sampling subroutine is a list of the particles (photons, electrons, positrons, and alphas) emitted in the random decay path, with their respective energies and emission times relative to the start of the decay.

The simulation is controlled by means of various parameters that are defined in the Fortran module PENNUC_mod, which is at the heading of the `pennuc.f` source file; the parameter values may be changed by the user by editing that source file. The parameter DRTIME is the resolution time of the detector (set to 5×10^{-6} s in the original source file). When the lifetime of a nuclear level is larger than DRTIME, the level is considered as metastable. Metastable levels are treated as effective halts of the decay process. The set of particles emitted between successive halts constitutes a "cascade" and is considered to define a single emission event. Thus, the whole decay process to the final ground level of the daughter nucleus consists of either a single cascade or a sequence of cascades. In the latter case, each cascade may produce a separate "count" in the detector. This strategy permits simulation of the response of real detectors, which may give several output counts as the result of the decay of a single nucleus through a metastable state. The parameter ECNUC is the energy cutoff for atomic transitions ($= 200$ eV in the original source file); only particles with energies higher than ECNUC are generated and delivered. PENNUC can handle up to NISM ($= 5$ in the original source file) different nuclides simultaneously.

2.1. Nuclear decay

As already mentioned, nuclear transitions are described by using the decay characteristics in the NUCLEIDE data files, which provide the following information:
● The list of possible disintegration types (α, β^-, β^+ or electron capture) of the parent nuclide with the corresponding daughter nuclides and relative disintegration probabilities.
● For each daughter nuclide:
1. The Q value, defined as the total available energy of the disintegration (*i.e.*, the difference in rest energies between the parent atom and the disintegration products).
2. The list of energies of excited levels and their half-lives, when available.
3. The list of disintegration branches, with the following information for each branch:
(a) The disintegration type (α, β^+, β^-, or electron capture).
(b) The initial level of the daughter nucleus.
(c) The branching ratio.
(d) For alpha branches, the kinetic energy of the alpha particle; for beta branches, the type of the transition (specified by the prohibition index, see Table 1) and the maximum kinetic energy E_{max} of the beta particles.
3. The list of possible de-excitation transitions of excited levels, their types (γ, internal conversion) and relative probabilities.

The type and energy of the particles released in each decay are determined randomly. The disintegration branch is sampled from the probabilities defined by the branching ratios by means of a binary search. In the cases of alpha emission and electron capture, the type of particle emitted and its energy are unique for each

Table 1
Classification of beta transitions according to the spin and parity changes. J_p and J_d denote the spins of the parent and daughter nuclei, respectively. The last column gives the corresponding prohibition index I_{pro}.

Kind of transition	$J_p - J_d$	Parity change	I_{pro}
Allowed	$0, \pm 1$	no	0
First forbidden	$0, \pm 1$	yes	0
Unique first forbidden	± 2	yes	1
Second forbidden	± 2	no	1
Unique second forbidden	± 3	no	2
Third forbidden	± 3	yes	2
Unique third forbidden	± 4	yes	3
Fourth forbidden	± 4	no	3

branch. The electrons or positrons emitted in beta disintegrations have a continuous energy spectrum, $p_\beta(E)$, that ranges from $E = 0$ to a maximum energy E_{max}. This spectrum is described by an analytical formula (see [1]) that depends on the spins and parities of the parent and daughter nuclides through a prohibition factor, which is generally determined by the prohibition index (Table 1). The PENNUC subroutines provide specific prohibition factors for a set beta emitters of metrological relevance (^{11}C, ^{13}N, ^{18}F, [22]Na, [24]Na, [36]Cl, ^{32}P, [40]K, [49]Sc, [63]Ni, ^{68}Ga, ^{89}Sr, ^{90}Sr, ^{90}Y, [89]Zr, ^{99}Tc, [129]I, [138]La, [141]Ce, [186]Re, [204]Tl, [210]Bi and [241]Pu). Sampling of the energy E of the emitted beta particle from the continuous spectrum is performed by using the RITA algorithm described in the Appendix.

De-excitation of excited levels of the daughter nuclide proceeds through a sequence of transitions, with emission of gamma rays or conversion electrons, that is determined by random sampling by the binary-search method according to the given transition probabilities. The particle emitted in each transition has an associated emission time or "age", which is the time at which it was emitted relative to the moment when the parent nucleus disintegrated. When an excited level of the daughter nucleus has a finite mean lifetime τ, we consider that the nucleus remains in that level for a time interval Δt, whose random value is sampled from the familiar exponential distribution,

$$\Delta t = -\tau \ln \xi. \tag{1}$$

Hereafter, ξ denotes a random number uniformly distributed in the interval (0, 1). The de-excitation sequence is halted when a metastable level (*i.e.*, an excited level with lifetime Δt longer than the detector resolution time DRTIME) is reached. The occurrence of such a level ends the generation of the cascade and control is returned to the calling program. De-excitation of the daughter nucleus is resumed at the next call to the PENNUC subroutines.

2.2. Atomic de-excitation

To represent atomic states and energy levels, we adopt an independent-electron model, in which electrons organize in (sub) shells characterized by the principal quantum number, n, the orbital angular momentum quantum number, ℓ, and the total angular momentum quantum number, j. A shell Si ($n_i l_i j_i$) can accommodate up to $2j_i - 1$ electrons (in orbitals with different magnetic quantum numbers) with characteristic ionization energy U_{Si}. Notice that shell ionization energies are positive; the quantity $-U_{Si}$ represents the "binding" energy of each individual electron. In the ground-state configuration of an atom, the shells with binding energies higher than about 50 eV are completely filled (closed).

As already mentioned, the processes of electron capture and internal conversion produce a vacancy in an electron shell, normally an inner shell, leaving the residual ion in a highly excited electronic state. Subsequently the ion de-excites by migration of the primary vacancy to outer shells through a cascade of radiative and non-radiative transitions.

In a radiative S0–S1 transition, a vacancy in the S0 shell is filled by an electron "falling" from the outer shell S1 (with $n_1 \geq n_0$) with emission of an x ray with characteristic energy, thus leaving a vacancy in the S1 shell. In a non-radiative S0–S1–S2 transition, a vacancy in the S0 shell is filled by an electron that falls from the outer S1 shell and an Auger electron is ejected from the S2 shell; after each non-radiative transition, the residual ion has vacancies in the S1 and S2 shells.

The PENNUC subroutines simulate the electronic de-excitation of ions by using the same method as in the Monte Carlo transport code PENELOPE [3]. The required relaxation characteristics are read from the files pdatconf.p14 (ground state configuration, shell ionization energies and atomic level widths) and pdrelax.p11 (transition probabilities and energies for singly ionized atoms) of the PENELOPE database.

Transition probabilities were extracted from the Evaluated Atomic Data Library (EADL) of Perkins et al. [5], which is the most extensive tabulation currently available. The EADL gives the probabilities of radiative transitions, P_{S0-S1}, and non-radiative transitions, $P_{S0-S1-S2}$, of singly-ionized atoms of the elements of atomic number from $Z = 1$ to $Z = 99$ with a single vacancy in a closed shell. These transition probabilities were obtained from calculations using the independent-electron model with the Dirac–Hartree–Fock–Slater (DHFS) potential, and adjusted to reproduce the Z-dependence of measured effective fluorescence yields.

The energies of x rays emitted in radiative transitions resulting from single vacancies in K or L shells were taken from the compilation by Deslattes et al. [6]. x ray energies for transitions of vacancies in M shells were taken from Bearden's [7] review and reevaluation of experimental x ray wavelengths. In the case of radiative transitions S0–S1 that are not included in Bearden's tabulation, the energy of the x ray is approximated as

$$E_x = U_{S0} - U_{S1}, \tag{2}$$

where U_{Si} is the ionization energy of an electron in the shell Si of the neutral atom given in the EADL. The EADL ionization energies are the negatives of the eigenvalues of the DHFS self-consistent equations for neutral atoms in their ground states. For tightly bound shells, with $U_{Si} \gtrsim 200$ eV, the calculated values U_{Si} agree closely with Carlson's [8] recommended shell ionization energies (see, *e.g.*, [9]). The energy of the Auger electron emitted in the non-radiative transition S0–S1–S2 is set equal to

$$E_e = U_{S0} - U_{S1} - U_{S2}. \tag{3}$$

Note that the recipes (2) and (3) correspond to assuming that the presence of the vacancy (or vacancies) does not alter the ionization energies of the active electron shells, which is an approximation.

Simulation of the electronic relaxation cascade is performed by subroutine ATREL (a duplicate of subroutine RELAX of the PENELOPE code system [3]), which starts by considering that there is a single vacancy in a shell Si. The transition that fills the initial vacancy is randomly selected according to the EADL transition probabilities by using Walker's aliasing method [10,11]. This transition leaves the ion with one or two vacancies, and the transitions that fill them are then sampled from the EADL transition probabilities. When the decaying ion has vacancies in several different shells, we assume that those in the innermost shell migrate first. That is, we give preference to those sequences of transitions that fill the innermost open shell more rapidly. This assumption determines the active shell Si of the next transition,

and the transition that effectively occurs is sampled randomly from the EADL probabilities.

Unfortunately, the EADL gives transition probabilities only for ions with a single vacancy. As calculations for multiply-ionized atoms are not generally available, we adopt the following simple strategy to partially correct for the possible occurrence of multiple vacancies during the de-excitation cascade. As each vacancy in the active shell S0 is explicitly filled by subroutine ATREL through a S0–S1 or S0–S1–S2 transition, we only need to account for vacancies existing in the intermediate and final shells, S1 and S2. We assume that transition probabilities are essentially proportional to the number of electrons in these shells. Let q_{Si} and v_{Si} denote, respectively, the degeneracy (number of orbitals) and the number of vacancies in shell Si before the transition. The probabilities of non-radiative and radiative transitions filling a S0 vacancy of an ion with vacancies in the S1 or S2 shells are approximated as

$$P_{S0\text{-}S1}^{ion} = P_{S0\text{-}S1} \left[\frac{q_{S1} - v_{S1}}{q_{S1}} \right] \qquad (4a)$$

and

$$P_{S0\text{-}S1\text{-}S2}^{ion} = P_{S0\text{-}S1\text{-}S2} \left[\frac{q_{S1} - v_{S1}}{q_{S1}} \frac{q_{S2} - v_{S2}}{q_{S2}} \right], \qquad (4b)$$

respectively. The factors in square brackets are less than unity and, hence, they can be accounted for by a rejection test: the transition is accepted only if a random number ξ is less than the corresponding factor, otherwise a new transition filling a vacancy in the S0 shell is sampled from the EADL probabilities.

The emission of x rays and Auger electrons after an ionization event is delayed because of the finite lifetime τ_{Si} of atomic energy levels (see, *e.g.*, Ref. [12]). The lifetime values used in PenNuc were obtained from the level widths $\Gamma_{Si} (= \hbar/\tau_{Si})$ in the tables of Campbell and Papp [13], which include data for elements with atomic number Z from 10 to 92, except for a few N shells. For elements with $Z > 92$, and for N shells that are missing in the tables of Campbell and Papp, the level widths were extracted from the EADL [5]. It is worth noticing that we simulate relaxation cascades using atomic lifetimes that correspond to excited states with a single vacancy. To partially account for the possible occurrence of multiple vacancies, when in the course of the de-excitation cascade we find $v_{S0} > 1$ vacancies in the active shell S0, the lifetime is set equal to τ_{S0}/v_{S0}. The program assumes that a vacancy is filled after a random time t_{S0}, which is sampled from the familiar exponential distribution as

$$t_{S0} = -(\tau_{S0}/v_{S0}) \ln \xi . \qquad (5)$$

where ξ is a random number.

3. Structure of the subroutine package

PenNuc is distributed as a zip file named `pennuc.zip` organized as a single directory with the following subdirectories:
• `pennuc/fsource`, contains the Fortran source file `pennuc.f` of the simulation subroutines, the decay-generator program `decays.f`, and the GNUPLOT script `spcnuc.gnu` for quick visualization of the particle spectra generated by `decays`.
• `pennuc/nucleide-files`, with decay data files downloaded from the NUCLEIDE web site,[1] and the files `pdatconf.p14` and `pdrelax.p11` of the PENELOPE database.

---

[1] These files were downloaded prior to submission of the present article. Since nuclear data files may change when results from new measurements are incorporated, users are advised to download updated files directly from the NUCLEIDE web site.

In the source files, all real quantities are defined as `double precision`, all integers are `integer*4`. The maximum array dimensions and other global parameters are defined within the source file `pennuc.f` in the Fortran module `PENNUC_mod` (see Table 2). At initialization, the PenNuc subroutines read information from the NUCLEIDE data files of the considered radionuclides, and from the files `pdatconf.p14` and `pdrelax.p11` of the PENELOPE database. When running the program, these files must be present in the PenNuc data directory, whose relative path (`RPATH`) is defined in the module `PENNUC_mod`. This module also conveys decay output information from the sampling subroutines. Notice that all energies are given in eV.

The subroutines to be called from the main program are the following:

• `SUBROUTINE PENNUC0(NUCFNAME,EPMAX,IWR,IER)`
reads nuclear decay and atomic relaxation data and initializes the algorithm for sampling random decay paths of a radionuclide.

Input arguments:

`NUCFNAME` . . .   filename of the NUCLEIDE data file of the considered radionuclide (up to 20 characters).

`IWR` . . .   unit number of the output file. The nuclear decay data read from the NUCLEIDE file are summarized in an output file, which must be open in the calling program as `UNIT = IWR` before the first call to `PENNUC0`.

Output values:

`EPMAX` . . .   highest energy of particles released in the decay of the radionuclide. This quantity determines the energy interval where simulated energy spectra have "counts".

`IER` . . .   error code. When `IER` ≠ 0 the initialization has not succeeded; the cause of the problem is written to the standard output unit (6).

Up to `NISM` nuclides can be considered simultaneously; the value of `NSIM` is set to 5 in the source file, but can be increased by editing the file. To simulate decays of several nuclides, call subroutine `PENNUC0` repeatedly. Each radionuclide is assigned a label `IS` defined by the order in which they are initialized. That is, `IS` = 1, 2, . . . , `NISM` for the first, second, …loaded nuclides. The sampling of the active nuclide (according to its relative activity in the radioactive source) must be performed in the calling program.

• `SUBROUTINE PENNUC(IS,METAST,IER)`
generates random decay pathways of the radioactive nuclide number `IS`, which is assumed to have been loaded previously by a call to subroutine `PENNUC0`.

Input arguments:

`IS` . . .   active radioisotope number.

`METAST` . . .   the value `METAST=0` instructs the subroutine to start the decay of a new parent nucleus.

Output arguments:

`METAST` . . .   flag for metastable levels,

=0, the de-excitation cascade has been completed;

=1, a metastable excited state has been reached.

`IER` . . .   error code; when `IER`≠0 the isotope `IS` is not defined.

Table 2
Source listing of the Fortran module PENNUC_mod, with specification of array dimensions, and output quantities from the sampling subroutines.

```
      MODULE PENNUC_mod
C ** Simulation of nuclear decay. Global quantities and state
C    variables of emitted particles. All energies in eV.
      SAVE  ! Saves all items in the module.
C -- Resolution time of the detector.
      DOUBLE PRECISION, PARAMETER :: DRTIME=5.0D-6
C -- Cutoff energy for the de-excitation of inner subshells.
      DOUBLE PRECISION, PARAMETER :: ECNUC=200.0D0
C -- Relative path to the PENNUC data directory.
      CHARACTER*18 :: RPATH='../nucleide-files/'
C -- Physical dimensions of arrays:
      INTEGER*4, PARAMETER :: NISM=5   ! Max. no. of nuclides.
      INTEGER*4, PARAMETER :: NDM=5    ! Max. no. of daughters.
      INTEGER*4, PARAMETER :: NBM=200  ! Max. no. of branches.
      INTEGER*4, PARAMETER :: NLM=200  ! Max. no. of levels.
      INTEGER*4, PARAMETER :: NTM=200  ! Max. no. of transitions.
      INTEGER*4 :: NIR=0  ! Number of loaded radionuclides.
C
C -- Daughter nucleus (IZD,IAD) and current level (LEVD).
      INTEGER*4 :: IZD,IAD,LEVD
C -- Particles emitted in the last decay cascade.
      INTEGER*4, PARAMETER :: NRM=500  ! Max. no. of particles.
      INTEGER*4 :: NR  ! No. of particles in the cascade (<= NRM).
C -- Particle types (1=electron, 2=photon, 3=positron, 4=alpha).
      INTEGER*4, DIMENSION (NRM) :: KPNR
C -- Transitions that originated the particles.
      INTEGER*4, DIMENSION (NRM) :: ITNR
C -- Particle energies, ages, and direction cosines.
      DOUBLE PRECISION, DIMENSION (NRM) :: ENR,AGENR,UNR,VNR,WNR
      END MODULE PENNUC_mod
```

When a metastable level (with $\Delta t >$ DRTIME) is reached, simulation of the nuclear de-excitation is halted, the output value of METAST is set to 1, and control is returned to the main program. At the next call to subroutine PENNUC, simulation of the decay continues down to the following metastable level or to the ground level. The output value METAST=0 indicates that the nucleus has reached its ground level.

Each call to subroutine PENNUC returns the state variables of the daughter nucleus and of the particles released in a cascade through module PENNUC_mod. The output information listed in the module is,

IZD, IAD . . . atomic number and mass number of the daughter nucleus.

LEVD . . . energy level of the daughter nucleus at the end of the simulated cascade. Nuclear energy levels are numbered in increasing order of energies (0, 1, …for the ground level, first excited level, …).

NR . . . number of particles in the cascade.

KPNR(1:NR) . . . particle types (1=, electrons; 2= photons, 3= positrons; 4= alphas).

ITNR(1:NR) . . . code numbers of the nuclear or atomic transitions that gave rise to the particles, as defined in the source file (see the heading comments of subroutine PENNUC).

ENR(1:NR) . . . energies of the particles, in eV. Notice that the list does not include particles emitted with energies lower than ECNUC.

AGENR(1:NR) . . . ages of the particles, in seconds (times elapsed from the start of the cascade to the emission time).

UNR(1:NR), VNR(1:NR), WNR(1:NR) . . . direction cosines of the particles, sampled from the isotropic distribution.

The program decays.f uses the PᴇɴNᴜᴄ subroutines to simulate random decay paths of a radionuclide. This program may be compiled and run directly from the pennuc/fsource directory. Notice that the file pennuc.f is included in decays.f and, therefore it should not be declared in the compilation list. The input decay characteristics are printed in a file named pennuc.dat. The default output of the program is a file named spcnuc.dat, which contains the energy spectra of emitted electrons, gammas, and positrons. Optionally the initial energies and ages of the emitted particles are printed in a second file named particles.dat. Be aware that the latter file may grow in size very rapidly. Notice also that these files are overwritten at each new run of the program.

The generated energy spectra can be visualized by means of any plotting program. We use ɢɴᴜᴘʟᴏᴛ, which is small in size, available for various platforms (including Windows and Linux) and free (distribution sites are listed at the Gnuplot Central site, http://www.gnuplot.info). When ɢɴᴜᴘʟᴏᴛ is installed on the computer, the provided script spcnuc.gnu plots the output spectra on the screen.
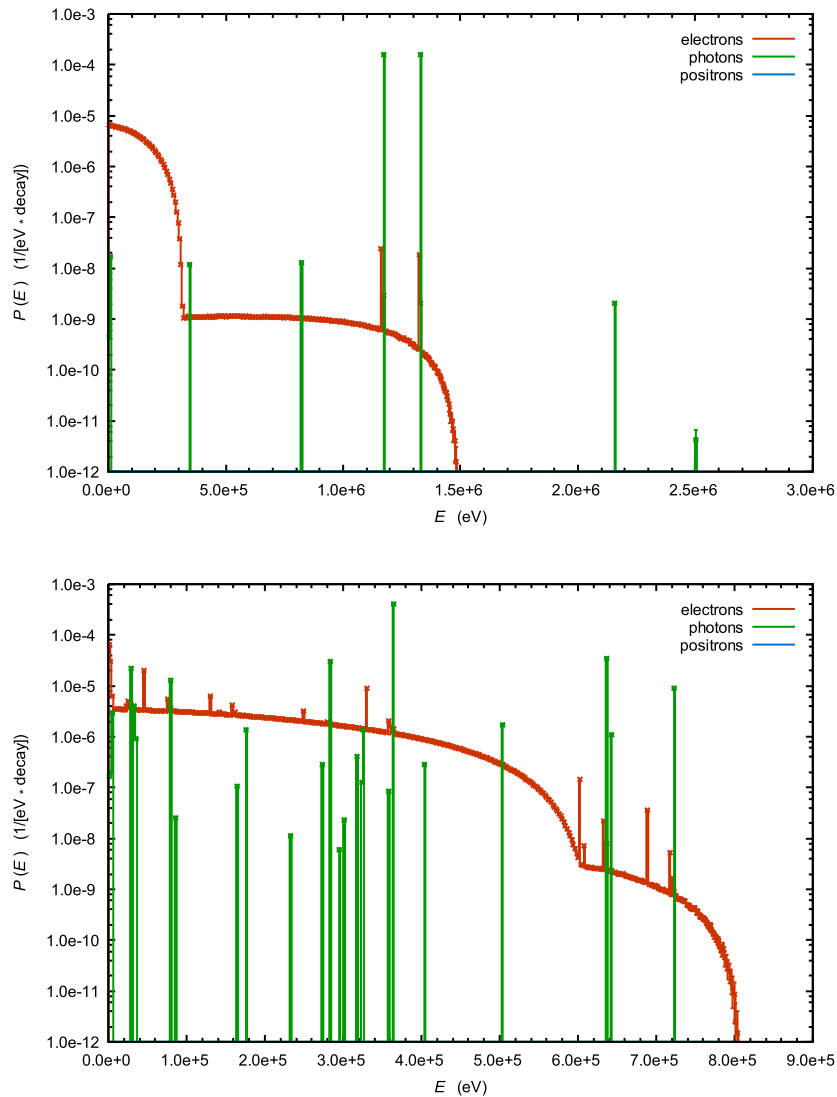
Fig. 1. Simulated energy spectra of photons and electrons emitted in 10^9 decays of ^{60}Co (top) and ^{131}I (bottom). The plots include statistical error bars, which here are smaller than the size of the symbols.

4. Sample simulation results

The results presented here were computed on an Intel Core i7-3520M CPU at 2.90 GHz with Windows 7. The program `de-cays.f` was compiled with gfortran, the Fortran compiler of the GNU Compiler collection (https://gcc.gnu.org/). Simulations of a million random decays of the nuclides ^{22}Na, ^{60}Co, and ^{214}Bi took 0.21, 0.29, and 0.33 s, respectively.

Fig. 1 shows energy spectra of particles (electrons and photons) emitted in the decay of ^{60}Co and ^{131}I, which were generated by the program `decays` after sampling 10^9 random decays of the radionuclide. The spectra were tallied as histograms with 400 bins, covering the energy interval from 0 to EPMAX (the highest energy of emitted particles), and normalized so that the integral of each histogram equals the average number of particles of the corresponding kind that are emitted in the disintegration and decay of the parent nucleus. The statistical uncertainty of each bin (3σ level) is displayed as an error bar, which is appreciable only for bins with relatively low probability. The plots in Fig. 1 were generated by GNUPLOT using the provided script `spcnuc.gnu`.

The subroutine package PENNUC was initially developed for describing radioactive sources with the general-purpose electron–photon Monte Carlo code PENELOPE. The reliability of the description of nuclear decay, and the subsequent transport of the emitted particles, has been demonstrated in various radiation metrology studies (see Ref. [1], and references therein).

Acknowledgments

Appendix. The RITA algorithm

In the PENNUC subroutines, the random sampling of the energy of beta particles is performed by using the RITA (Rational Inverse Transform with Aliasing) algorithm, an adaptive sampling procedure that is both fast and highly accurate. RITA can be applied to any continuous random variable x defined in a finite interval (x_{min}, x_{max}) with a probability density function (PDF) $p(x)$ which can be calculated accurately for any value of x in that interval.

Let us assume that the cumulative distribution function,

$$\mathcal{P}(x) = \int_{x_{min}}^{x} p(x')dx', \tag{A.1}$$

has been evaluated numerically for a certain grid of x-values that spans the interval $(x_{\min}, x_{\max})$,

$$x_1 = x_{\min} < x_2 < \cdots < x_{N-1} < x_N = x_{\max}. \tag{A.2}$$

Setting $\xi_i = \mathcal{P}(x_i)$, we get a table of the inverse cumulative distribution function $\mathcal{P}^{-1}(\xi_i) = x_i$ for a grid of ξ-values that spans the interval $(0, 1)$,

$$\xi_1 = 0 < \xi_2 < \cdots < \xi_{N-1} < \xi_N = 1. \tag{A.3}$$

Random values of x can be generated by using the conventional inverse transform method, that is, with the sampling formula $x = \mathcal{P}^{-1}(\xi)$. In principle, the value x can be obtained by interpolation in the pre-calculated table. The adopted interpolation scheme must be able to accurately reproduce the first derivative of the function $\mathcal{P}^{-1}(\xi)$,

$$\frac{d\mathcal{P}^{-1}(\xi)}{d\xi} = \left(\frac{d\mathcal{P}(x)}{dx}\right)^{-1} = \frac{1}{p(x)}. \tag{A.4}$$

Notice that this function is very steep in regions where $p(x)$ is small. Linear interpolation of $\mathcal{P}^{-1}(\xi)$ is, in general, too crude because it is equivalent to approximating $p(x)$ by a stepwise distribution. It is more expedient to use a *rational interpolation* scheme of the type[2]

$$\widetilde{\mathcal{P}}^{-1}(\xi) = x_i + \frac{(1 + a_i + b_i)\eta}{1 + a_i\eta + b_i\eta^2}(x_{i+1} - x_i) \quad \text{if } \xi_i \le \xi < \xi_{i+1}, \tag{A.5}$$

where

$$\eta \equiv (\xi - \xi_i)/(\xi_{i+1} - \xi_i), \tag{A.6}$$

and a_i and b_i are parameters. Notice that $\widetilde{\mathcal{P}}^{-1}(\xi_i) = x_i$ and $\widetilde{\mathcal{P}}^{-1}(\xi_{i+1}) = x_{i+1}$, irrespective of the values of a_i and b_i. In addition,

$$\frac{d\widetilde{\mathcal{P}}^{-1}(\xi)}{d\xi} = \frac{(1 + a_i + b_i)(1 - b_i\eta^2)}{(1 + a_i\eta + b_i\eta^2)^2}\frac{x_{i+1} - x_i}{\xi_{i+1} - \xi_i}. \tag{A.7}$$

The parameters a_i and b_i are determined by requiring that

$$\left[\frac{d\widetilde{\mathcal{P}}^{-1}(\xi)}{d\xi}\right]_{\xi = \xi_i} = \frac{1}{p(x_i)} \quad \text{and} \quad \left[\frac{d\widetilde{\mathcal{P}}^{-1}(\xi)}{d\xi}\right]_{\xi = \xi_{i+1}} = \frac{1}{p(x_{i+1})}. \tag{A.8}$$

This implies

$$1 + a_i + b_i = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i}\frac{1}{p(x_i)} \quad \text{and}$$
$$\frac{1 - b_i}{1 + a_i + b_i} = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i}\frac{1}{p(x_{i+1})},$$

and it follows that

$$b_i = 1 - \left(\frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i}\right)^2 \frac{1}{p(x_{i+1})\,p(x_i)}, \tag{A.9a}$$

$$a_i = \frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i}\frac{1}{p(x_i)} - b_i - 1. \tag{A.9b}$$

Once these parameters have been calculated, the sampling formula

$$x = x_i + \frac{(1 + a_i + b_i)\eta}{1 + a_i\eta + b_i\eta^2}(x_{i+1} - x_i) \quad \text{if } \xi_i \le \xi < \xi_{i+1} \tag{A.10}$$

gives random values of x that are *exactly* distributed according to the PDF

---
[2] We denote by $\widetilde{\mathcal{P}}^{-1}(\xi)$ the function obtained by interpolation of the tabulated values $\mathcal{P}^{-1}(\xi_i)$.

$$\widetilde{p}(x) = \left(\frac{d\widetilde{\mathcal{P}}^{-1}(\xi)}{d\xi}\right)^{-1} = \frac{(1 + a_i\eta + b_i\eta^2)^2}{(1 + a_i + b_i)(1 - b_i\eta^2)}\frac{\xi_{i+1} - \xi_i}{x_{i+1} - x_i}$$
$$\text{if } x_i \le x < x_{i+1}. \tag{A.11}$$

where η is determined by Eq. (A.10). The root that satisfies the conditions $\eta = 0$ for $x = x_i$ and $\eta = 1$ for $x = x_{i+1}$ is

$$\eta = \frac{1 + a_i + b_i - a_i\tau}{2b_i\tau}\left[1 - \sqrt{1 - \frac{4b_i\tau^2}{(1 + a_i + b_i - a_i\tau)^2}}\right],$$
$$\tau \equiv \frac{x - x_i}{x_{i+1} - x_i}. \tag{A.12}$$

It is worth noting that the distributions $p(x)$ and $\widetilde{p}(x)$ not only coincide at the grid points x_i, but also their integrals over the subinterval (x_i, x_{i+1}) are equal, i.e., $\widetilde{p}(x)$ gives the correct probability of finding x in each subinterval. The PDF (A.11) turns out to be fairly flexible and can approximate smooth PDFs over relatively wide subintervals to good accuracy. Furthermore, because formula (A.10) involves only a few arithmetic operations, random sampling will be faster than with alternative interpolation schemes that lead to sampling formulas involving transcendental functions.
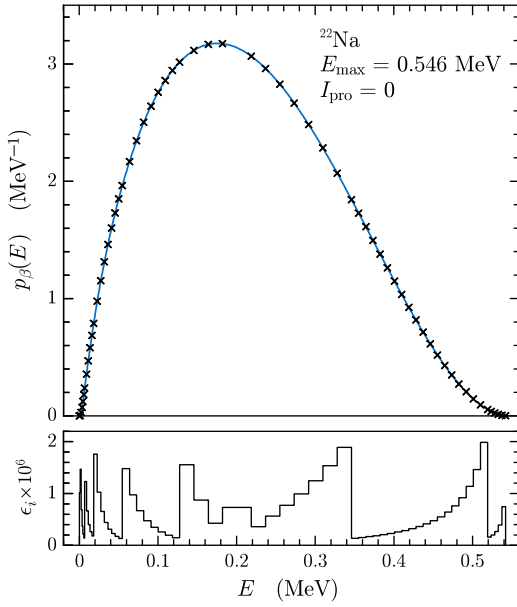
A.1. Determining the interpolation grid

The key to ensure accuracy of the sampling is to set a suitable grid of x-values, $x_i\,(i = 1, \ldots, N)$, such that errors introduced by the rational interpolation (A.11) are negligible (say, of the order of 0.01% or less). Both a simple and effective strategy for defining the x-grid is the following. We start with a uniform grid of ~ 10 equally spaced x-values. The cumulative distribution function at these grid points, $\mathcal{P}(x_i) = \xi_i$, is evaluated numerically (see below). After calculating the parameters a_i and b_i of the interpolating PDF, Eq. (A.11), the interpolation "error" in the ith subinterval (x_i, x_{i+1}) is defined as

$$\epsilon_i = \int_{x_i}^{x_{i+1}} |p(x) - \widetilde{p}(x)|\,dx, \tag{A.13}$$

where the integral is evaluated numerically using the trapezoidal rule. To reduce the interpolation error efficiently, new points x_i are added where the error is larger. The position of each new point is selected at the midpoint of the subinterval j with the largest ϵ value. After inserting each new point, the interpolation parameters a_i and b_i, for the two new subintervals (the two halves of the initial jth subinterval) are evaluated, as well as the corresponding interpolation errors ϵ_i, Eq. (A.13). The process is iterated until the last, Nth, grid point has been set. Obviously, to reduce the interpolation error we only need to increase the number N of grid points.

Using this strategy to set new grid points, the algorithm is self-adaptive: grid points automatically accumulate in subintervals where the PDF and its approximation (A.11) differ most. If the number N of grid points is large enough, the functions $p(x)$ and $\widetilde{p}(x)$ vary smoothly within each grid subinterval (x_i, x_{i+1}). Integrals of those functions over grid subintervals can then be calculated accurately by using simple quadrature formulas. In our implementation of the algorithm, we use the extended Simpson rule with 51 equally-spaced points.

Fig. A.2 displays the rational interpolation, Eq. (A.11), of the beta spectrum of the most probable decay branch of ^{22}Na ($E_{\max} = 0.546$ MeV, $I_{\text{pro}} = 0$). The crosses indicate the points of the grid with $N = 64$. The lower plot in Fig. A.2 represents the local interpolation error ϵ_i in each subinterval of the grid (as a stepwise function for visual aid); the maximum error in this case is 2×10^{-6}. For a denser grid with $N = 128$ values, the maximum error decreases to 1.4×10^{-7}.

Fig. A.2. Rational interpolation of the continuous PDF corresponding to the most probable beta branch of ^{22}Na ($I_{pro} = 0$). The crosses are grid points determined as described in the text with $N = 64$. The lower plot displays the interpolation error ϵ_i, Eq. (A.13), in each subinterval.

A.2. Sampling algorithm

After determining the interpolation grid and the parameters of the rational interpolation,

$$x_i, \ \xi_i = \mathcal{P}(x_i), \ a_i, \ b_i \qquad (i = 1, \ldots, N), \tag{A.14}$$

the sampling from the distribution (A.11) can be performed exactly by using the following algorithm:

1. Generate a random number ξ.
2. Find the subinterval i that contains ξ (*i.e.*, such that $\xi_i \leq \xi < \xi_{i+1}$) by the binary-search method.
3. Set $\nu \equiv \xi - \xi_i$, $\Delta_i \equiv \xi_{i+1} - \xi_i$, and deliver the value

$$x = x_i + \frac{(1 + a_i + b_i)\Delta_i \nu}{\Delta_i^2 + a_i \Delta_i \nu + b_i \nu^2} (x_{i+1} - x_i). \tag{A.15}$$

The sampled x values are such that $x = \widetilde{\mathcal{P}}(\xi)$ and, hence, x increases monotonically with ξ. The code PENELOPE takes advantage of this feature to simplify the sampling in restricted intervals of the variable. A drawback of the algorithm is that the sampling speed decreases (slowly) when the number N of grid points increases, due to the increasing number of comparisons needed in step 2. When a strict correspondence of x and ξ values is not relevant, this speed loss can be avoided by sampling the subinterval randomly by means of Walker's aliasing [10,11]; notice that the probability of the ith subinterval is $p_i = \mathcal{P}(x_{i+1}) - \mathcal{P}(x_i)$. Walker's method requires only a single comparison and, hence, the algorithm becomes optimal, at the expense of some additional memory storage.

The sampling speed is independent of the complexity of the PDF and of the number of grid points. With a simple main program compiled and run on an Intel Core i7-3520M CPU at 2.90 GHz with Windows 7, our RITA sampling subroutine produces about 29 million random values per second (CPU time).

References

[1] E. García-Toraño, V. Peyres, M.-M. Bé, C. Dulieu, M.-C. Lépy, F. Salvat, Nucl. Instrum. Methods B 396 (2017) 43–49.
[2] M.-M. Bé, N. Coursol, B. Duchemin, F. Lagoutine, J. Legrand, K. Debertin, E. Schönfeld, Table of Radionuclides. Introduction, Tech. Rep. Note Technique LNHB 211/53, Laboratoire National Henri Becquerel, Saclay, France, 2011.
[3] F. Salvat, PENELOPE-2014: A Code System for Monte Carlo Simulation of Electron and Photon Transport, OECD/NEA Data Bank, NEA/NSC/DOC(2015)3, 2015, Available from http://www.nea.fr/lists/penelope.html.
[4] F. James, Comput. Phys. Commun. 60 (1990) 329–344.
[5] S.T. Perkins, D.E. Cullen, M.H. Chen, J.H. Hubbell, J. Rathkopf, J. Scofield, Tables and Graphs of Atomic Subshell and Relaxation Data Derived from the LLNL Evaluated Atomic Data Library (EADL), Z = 1–100, Tech. Rep. UCRL-ID-50400, Lawrence Livermore National Laboratory, Livermore, California, 1991.
[6] R.D. Deslattes, E.G. Kessler, P. Indelicato, L. de Billy, E. Lindroth, J. Anton, Rev. Mod. Phys. 75 (2003) 36–99.
[7] J.A. Bearden, Rev. Mod. Phys. 39 (1967) 78–124.
[8] T.A. Carlson, Photoelectron and Auger Spectroscopy, Plenum Press, New York, 1975.
[9] F. Salvat, J.M. Fernández-Varea, Metrologia 46 (2009) S112–S138.
[10] A.J. Walker, ACM Trans. Math. Software 3 (1977) 253–256.
[11] F. Salvat, Comput. Phys. Commun. 46 (1987) 427–436.
[12] K.D. Sevier, Low Energy Electron Spectrometry, Wiley Interscience, New York, 1972.
[13] J.L. Campbell, T. Papp, At. Data Nucl. Data Tables 77 (2001) 1–56.