Package 'radsafer'

December 3, 2025

Type Package

Title Radiation Safety

Version 2.4.0

Date 2025-11-05

Description Provides functions for radiation safety, also known as ``radiation protection'' and ``radiological control''. The science of radiation protection is called ``health physics'' and its engineering functions are called ``radiological engineering''. Functions in this package cover many of the computations needed by radiation safety professionals. Examples include: obtaining updated calibration and source check values for radiation monitors to account for radioactive decay in a reference source, simulating instrument readings to better understand measurement uncertainty, correcting instrument readings for geometry and ambient atmospheric conditions. Many of these functions are described in Johnson and Kirby (2011, ISBN-13: 978-1609134198). Utilities are also included for developing inputs and processing outputs with radiation transport codes, such as MCNP, a general-purpose Monte Carlo N-Particle code that can be used for neutron, photon, electron, or coupled neutron/photon/electron transport (Werner et. al. (2018) <doi:10.2172/1419730>).

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests testthat, beepr, knitr, rmarkdown

Imports ggplot2, readr, stats, graphics, RadData, stringr, magrittr, dplyr, rlang, scatterplot3d, ggthemes, lifecycle

Depends R (>= 3.5)

URL https://github.com/markhogue/radsafer

BugReports https://github.com/markhogue/radsafer/issues

VignetteBuilder knitr

NeedsCompilation no

Author Mark Hogue [aut, cre]

Maintainer Mark Hogue <mark.hogue.chp@gmail.com>

Repository CRAN

Date/Publication 2025-12-03 21:30:10 UTC

Contents

air_dens_cf 3
disk_to_disk_solid_angle 3
dk_correct 5
dk_pct_to_num_half_life 7
dk_time 7
half_life_2pt 8
hvl 9
mcnp_cone_angle 9
mcnp_est_nps 10
mcnp_matrix_rotations 11
mcnp_mesh_bins 12
mcnp_plot_out_spec 13
mcnp_scan2plot 14
mcnp_scan_save 15
mcnp_sdef_erg_hist 16
mcnp_sdef_erg_line 17
neutron_geom_cf 18
photons_cs137_hist 19
rate_meter_sim 19
RN_bin_screen_phot 20
RN_find_parent 21
RN_index_screen 22
RN_info 23
RN_plot_df 24
RN_plot_search_results 25
RN_plot_spectrum 26
RN_save_spectrum 27
RN_search_alpha_by_E 28
RN_search_beta_by_E 30
RN_search_phot_by_E 31
RN_Spec_Act 32
scaler_sim 32
stay_time 33
tau_estimate 34

Index **35**

air_dens_cf — *Correct for air density - useful for vented ion chambers*

Description

Obtain a correction factor for ion chamber temperature and pressure vs reference calibration values.

Usage

```
air_dens_cf(T.actual, P.actual, T.ref = 20, P.ref = 760)
```

Arguments

T.actual	The actual air temperature, in Celsius
P.actual	The actual air pressure, in mm Hg
T.ref	The reference air temperature - default is 20C
P.ref	The reference air pressure - default is 760 mm Hg

Value

The ratio of actual to reference air density.

See Also

Other rad measurements: disk_to_disk_solid_angle(), neutron_geom_cf(), scaler_sim(), tau_estimate()

Examples

```
air_dens_cf(T.actual = 20, P.actual = 760, T.ref = 20, P.ref = 760)
air_dens_cf(30, 750)
```

disk_to_disk_solid_angle — *Calculate fractional solid angle for disk to disk*

Description

Returns fractional solid angle for a geometry frequently encountered in health physics analysis of air samples or disk smears. This is useful in correcting configurations that do not exactly match calibration (by ratioing the respective fractional solid angles). While units of steridian are used for solid angle, this function only uses a fraction of the total field of view.

Usage

```
disk_to_disk_solid_angle(
  r.source,
  gap,
  r.detector,
  plot.opt = "n",
  runs = 10000,
  off_center = 0,
  beep = "off"
)
```

Arguments

`r.source`	source radius (all units must be consistent)
`gap`	distance between source and detector
`r.detector`	detector radius
`plot.opt`	plot options - "2d", "3d" or "n".
`runs`	Number of particles to simulate. Running more particles improves accuracy. Default = 1e4.
`off_center`	measure of eccentricity between the center of the source and the center of the disk. This is applied to the x-dimension of the source.
`beep`	Set to "on" if desired. Default is "off". Alerts to end of run if runs is set to a high number.

Value

Fractional solid angle and plot of simulation.

References

https://karthikkaranth.me/blog/generating-random-points-in-a-sphere/ https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance

See Also

Other rad measurements: `air_dens_cf()`, `neutron_geom_cf()`, `scaler_sim()`, `tau_estimate()`

Examples

```
disk_to_disk_solid_angle(r.source = 15, gap = 20, r.detector = 10, plot.opt = "n", runs = 1e3)
```

`dk_correct` *Correct activity-dependent value based on radioactive decay.*

Description

Decay-corrected values are provided. Either a single or multiple values are computed. The computation is made either based on a single radionuclide, or based on user-provided half-life, with time unit. The differential time is either computed based on dates entered or time lapsed based on the time unit. Time units must be consistent. Decay-correct a source to today's date by assigning a reference `date1` and allowing default `date2`, the system date.

Usage

```
dk_correct(
  RN_select = NULL,
  half_life = NULL,
  time_unit = NULL,
  time_lapse = NULL,
  date1 = NULL,
  date2 = Sys.Date(),
  A1 = 1,
  num = FALSE
)
```

Arguments

`RN_select`	identify the radionuclide of interest in the format, *"Es-254m"* Required unless `half_life` is entered.
`half_life`	Required if `RN_select` is not provided.
`time_unit`	acceptable values are *"years"*, *"days"*, *"hours"*, *"minutes"*, and *"seconds"*. May be shortened to *"y"*, *"d"*, *"h"*, *"m"*, and *"s"*. Required if `half_life` or `time_lapse` are to be entered.
`time_lapse`	a single value or vector of values representing time lapsed since `date1`, with units identified in `time_unit`. Positive values represent time past `date1`. Negative values represent time before `date1`. Required unless `date1` is entered.
`date1`	Reference date - Required unless using `time_lapse`. Format is required to be date-only: *"YYYY-MM-DD"* (e.g. *"1999-12-31"*). If `half_life` is short relative to calendar dates, use `time_lapse` instead.
`date2`	Date or dates of interest. Default is today's date, obtained from the computer operating system.
`A1`	The reference activity or related parameter, such as count rate or dose rate. Default value is 1, resulting in a returned value that may be used as a correction factor.
`num`	Set for TRUE to facilitate as.numeric results. Default = FALSE.

Value

Decay adjusted activity or related parameter. See A1.

See Also

Other decay corrections: dk_pct_to_num_half_life(), dk_time()

Examples

```
# RN_select and date1 (saving numerical data)
my_dks <- dk_correct(
  RN_select = "Sr-90",
  date1 = "2009-01-01",
  date2 = "2019-01-01",
  num = TRUE
)

#   RN_select and time_lapse (random sample)
dk_correct(
  RN_select = base::sample(RadData::ICRP_07.NDX$RN, 1),
  time_lapse = 1:10,
  time_unit = base::sample(c("y", "d", "h", "m", "s"), 1)
)

#   half_life and date1
dk_correct(
  half_life = 10,
  time_unit = "y",
  date1 = "2009-01-01",
  date2 = c(
    "2015-01-01",
    "2016-01-01",
    "2017-01-01"
  )
)

#   half_life and time_lapse
dk_correct(
  half_life = 10,
  time_lapse = 10,
  time_unit = "y"
)

# decay to today
dk_correct(RN_select = "Sr-90", date1 = "2009-01-01")

# reverse decay - find out what readings should have been in the past given today's reading of 3000
dk_correct(
  RN_select = "Sr-90",
  date1 = "2019-01-01",
  date2 = c("2009-01-01", "1999-01-01"),
  A1 = 3000
```

```
)
```

`dk_pct_to_num_half_life` *Number of half-lives past*

Description

Given a percentage reduction in activity, calculate how many half-lives have passed.

Usage

```
dk_pct_to_num_half_life(pct_lost)
```

Arguments

`pct_lost`	Percentage of activity lost since reference time.

Value

Number of half-lives passed.

See Also

Other decay corrections: `dk_correct()`, `dk_time()`

Examples

```
dk_pct_to_num_half_life(pct_lost = 93.75)
```

`dk_time` *Time to decay to target radioactivity.*

Description

Calculate time for a radionuclide to decay to a target activity.

Usage

```
dk_time(half_life, A0, A1)
```

Arguments

`half_life`	Half-life. Units are arbitrary, but must match time past.
`A0`	The original activity, or related parameter.
`A1`	The target activity.

Value

Time, in same units as half-life, to decay to target activity.

See Also

Other decay corrections: dk_correct(), dk_pct_to_num_half_life()

Examples

```
# A carbonaceous artifact has a C-14 measurement of 1 dpm per g pure carbon.
# The reference activity is 14 dpm per g pure carbon. How old is our sample?
dk_time(half_life = 5730, A0 = 14, A1 = 1)
```

---

half_life_2pt *Calculate half-life based on two data points*

Description

Estimate half-life from two data points. Half-life units are consistent with time units of input. @family rad measurements

Usage

```
half_life_2pt(time1, time2, N1, N2)
```

Arguments

time1	First time: Must be numeric with no formatting.
time2	Second time: Must be numeric with no formatting.
N1	First measurement - can be count rate, dose rate, etc.
N2	Second measurement in units consistent with first measurement.

Value

The calculated half-life in units of time input.

Examples

```
# Between the first two data points in a series of counts
half_life_2pt(time1 = 0, time2 = 1, N1 = 45, N2 = 30)
#
# Between the second and third in the series (same intervals)
half_life_2pt(time1 = 1, time2 = 2, N1 = 30, N2 = 21)
#
# Use on a series
count_times <- 1:5
acts <- 10000 * 2^(-count_times / 10) # activities
acts <- rpois(5, acts) # activities with counting variability applied
```

```
half_life_2pt(
  time1 = count_times[1:4], time2 = count_times[2:5],
  N1 = acts[1:4], N2 = acts[2:5]
)
```

---

hvl *half-value layer and tenth-value layer computations*

Description

Derive hvl and tvl from radiation values through a material thickness.

Usage

```
hvl(x, y)
```

Arguments

x	material thickness
y	radiation measure through the material

Value

a data frame with the inputs, followed by the computed values for attenuation coefficient (listed as "mu"), half-value layer (hvl), tenth-value layer (tvl), and the homogeneity coefficient (hc) which is the ratio of a half-value layer to the following half-value layer.

Examples

```
H50_ex <- data.frame("mm_Al" = 0:5, "mR_h" = c(7.428, 6.272, 5.325,4.535, 3.878, 3.317))
hvl(x = H50_ex$mm_Al, y = H50_ex$mR_h)
```

---

mcnp_cone_angle *MCNP Cone Opening Parameter*

Description

MCNP cone surface requires a term, t^2, which is the tangent of the cone angle, in radians, squared. This function takes an input in degrees and provides the parameter needed by MCNP.

Usage

```
mcnp_cone_angle(d)
```

Arguments

d	The cone angle in degrees.

Value

tangent of cone angle squared

See Also

Other mcnp tools: mcnp_est_nps(), mcnp_matrix_rotations(), mcnp_mesh_bins(), mcnp_plot_out_spec(), mcnp_scan2plot(), mcnp_scan_save(), mcnp_sdef_erg_hist(), mcnp_sdef_erg_line()

Examples

```
mcnp_cone_angle(45)
```

---

mcnp_est_nps — *Copy and paste MCNP tally fluctuation charts*

Description

Provides quick estimate of number of particles histories, (nps) to obtain target MCNP 'error'. Paste may include up to three tallies side by side in the default MCNP order. For example, the headers of a three tally report includes column names: nps, mean, error, vov, slope, fom, mean, error, vov, slope, fom, mean, error, vov, slope, fom. The structure of the tfc has been the same for versions 4 through 6, including MCNPX.

Usage

```
mcnp_est_nps(err_target)
```

Arguments

err_target	The target Monte Carlo uncertainty

Value

estimate of number of particle histories needed

See Also

Other mcnp tools: mcnp_cone_angle(), mcnp_matrix_rotations(), mcnp_mesh_bins(), mcnp_plot_out_spec(), mcnp_scan2plot(), mcnp_scan_save(), mcnp_sdef_erg_hist(), mcnp_sdef_erg_line()

Examples

```
# Since this function requires the user
# to copy and paste input, this example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# Enter '1' for number of tallies.
# mcnp_est_nps(0.01)
#      32768000   4.5039E+00 0.2263 0.0969  0.0 5.0E-02
#      65536000   3.9877E+00 0.1561 0.0553  0.0 5.1E-02
#      98304000   3.4661E+00 0.1329 0.0413  0.0 4.7E-02
#     131072000   3.5087E+00 0.1132 0.0305  0.0 5.0E-02
#     163840000   3.5568E+00 0.0995 0.0228  0.0 5.2E-02
#     196608000   3.8508E+00 0.0875 0.0164  0.0 5.5E-02
#     229376000   3.8564E+00 0.0810 0.0135  0.0 5.5E-02
#     262144000   3.9299E+00 0.0760 0.0118  0.0 5.5E-02
#     294912000   4.0549E+00 0.0716 0.0100  0.0 5.6E-02
#     327680000   4.0665E+00 0.0686 0.0090  0.0 5.4E-02
#     360448000   4.1841E+00 0.0641 0.0079  0.0 5.7E-02
```

---

mcnp_matrix_rotations *Rotation matrices for transformations in MCNP*

Description

Create 3 x 3 rotation matrix in cosines of the angles between the main and auxiliary coordinate systems in the form: xx’ yx’ zx’ xy’ yy’ zy’ xz’ yz’ zz’

Usage

```
mcnp_matrix_rotations(rot.axis, angle_degrees)
```

Arguments

rot.axis	axis of rotation
angle_degrees	degree of rotation

Value

rotational matrix for copy and paste to MCNP input

See Also

Other mcnp tools: mcnp_cone_angle(), mcnp_est_nps(), mcnp_mesh_bins(), mcnp_plot_out_spec(), mcnp_scan2plot(), mcnp_scan_save(), mcnp_sdef_erg_hist(), mcnp_sdef_erg_line()

Examples

```
mcnp_matrix_rotations("x", 30)
mcnp_matrix_rotations("y", 7)
mcnp_matrix_rotations("z", 15)
# For combined rotations, use matrix multiplication (%*%)
# rotate 45 degrees on x-axis and 45 degrees on y-axis
mcnp_matrix_rotations("x", 45) %*% mcnp_matrix_rotations("y", 45)
```

---

mcnp_mesh_bins *Make mesh tally size settings for MCNP*

Description

#' **[Experimental]**: Find the parameters needed for a rectilinear "superimposed mesh tally b" in MCNP. It can be a challenge to center mesh tally bins at a desired value of x, y, or z. This function looks at a single dimension, – in units of cm – at a time. This is a new function and hasn't been tested thoroughly. The idea is to identify a single setting in the MCNP mesh tally for imesh and iints (or jmesh and jints or kmesh and kints). It is designed only for uniform mesh bin sizes.

Usage

```
mcnp_mesh_bins(
  target,
  width,
  lowest_less,
  lowest_high,
  highest_high,
  highest_less
)
```

Arguments

target	the desired center a single mesh
width	the individual mesh
lowest_less	in the direction of a decreasing dimension, what is the lowest that it can go and still be acceptable?
lowest_high	in the direction of an increasing dimension, what is the lowest that it can go and still be acceptable?
highest_high	in the direction of an increasing dimension, what is the highest that it can go and still be acceptable?
highest_less	in the direction of a decreasing dimension, what is the highest that it can go and still be acceptable?

Value

a data frame providing:

low_set, the minimum dimension. This is probably best used in the origin parameter in the MCNP mesh tally. high_set, the maximum dimension for the bin. This can be identified in the MCNP mesh tally setting of imesh, jmesh, or kmesh. width, this is just a return of the parameter supplied to the function. numblocks, the number of fine meshes. This can be used in the MCNP mesh tally setting of iints, jints, or kints.

See Also

Other mcnp tools: `mcnp_cone_angle()`, `mcnp_est_nps()`, `mcnp_matrix_rotations()`, `mcnp_plot_out_spec()`, `mcnp_scan2plot()`, `mcnp_scan_save()`, `mcnp_sdef_erg_hist()`, `mcnp_sdef_erg_line()`

Examples

```
mcnp_mesh_bins(target = 30, width = 10, lowest_less = 0,
 highest_less = 15, highest_high = 304.8, lowest_high = 250)
#'
```

`mcnp_plot_out_spec` *Convert histogram data to average points and plot as spectrum.*

Description

Model results or input source histograms from MCNP and perhaps other sources typically provide binned tally results with columns representing maximum energy in MeV, a column with the mean tally result or bin probability and an uncertainty column (not used). Once the data is scanned in, or otherwise entered into the R global environment, they can be plotted with this function.

Usage

```
mcnp_plot_out_spec(spec.df, title = deparse(substitute(spec.df)), log_plot = 0)
```

Arguments

`spec.df`	A data frame with no header. Maximum energy in MeV should be in the first column, (named E_MeV), and binned results in the second column, (named prob).
`title`	Title for chart (default = name of spec.df)
`log_plot`	0 = no log axes (default), 1 = log y-axis, 2 = log both axes.

See Also

`mcnp_scan_save()` to copy and paste output spectrum.

Other mcnp tools: `mcnp_cone_angle()`, `mcnp_est_nps()`, `mcnp_matrix_rotations()`, `mcnp_mesh_bins()`, `mcnp_scan2plot()`, `mcnp_scan_save()`, `mcnp_sdef_erg_hist()`, `mcnp_sdef_erg_line()`

Examples

```
mcnp_plot_out_spec(photons_cs137_hist, "example Cs-137 well irradiator")
```

---

mcnp_scan2plot *Copy and paste MCNP output spectral data to directly plot*

Description

Provides quick copy-and-paste to plot. Paste either a source histogram distribution or tally spectrum from MCNP outputs. Three-column output tally spectra have columns of maximum energy, bin tally, and relative Monte Carlo uncertainty for the bin tally value. Four-column source histogram distributions have columns of entry number, maximum energy, cumulative probability, and bin probability. In either case, only the maximum energy and bin probability or result values are used.

Usage

```
mcnp_scan2plot(title = "", log_plot = FALSE)
```

Arguments

title	Title for chart (default = name of spec.df)
log_plot	0 = no log axes (default), 1 = log y-axis, 2 = log both axes.

Value

spectrum file with maximum energy and MCNP bin value

See Also

Other mcnp tools: mcnp_cone_angle(), mcnp_est_nps(), mcnp_matrix_rotations(), mcnp_mesh_bins(), mcnp_plot_out_spec(), mcnp_scan_save(), mcnp_sdef_erg_hist(), mcnp_sdef_erg_line()

Examples

```
# Since this function requires the user
# to copy and paste input, this three column example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# mcnp_scan2plot(title = "example1")
# 0.1000000 3.133122e-05 0.3348260
# 0.4222222 6.731257e-05 0.2017546
# 0.7444444 5.249198e-05 0.4524577
# 1.0666667 2.046046e-04 0.4201954
# 1.3888889 1.525125e-03 0.8049388
# 1.7111111 2.922743e-05 0.7985399
# 2.0333333 5.162954e-03 0.1974694
# 2.3555556 2.048186e-05 0.5011170
# 2.6777778 1.468040e-04 0.7248116
# 3.0000000 1.037092e-04 0.7659850
```

`mcnp_scan_save` *Copy and paste MCNP output spectral data for use with* `mcnp_plot_out_spec()`

Description

Provides quick copy-and-paste conversion to data frame. Paste either a source histogram distribution or tally spectrum from MCNP outputs. Three-column output tally spectra have columns of maximum energy, bin tally, and relative Monte Carlo uncertainty for the bin tally value. Four-column source histogram distributions have columns of entry number, maximum energy, cumulative probability, and bin probability. Seven-column biased histogram distributions have columns of entry number, maximum energy, cumulative probability, biased cumulative probability, probability of bin, biased probability, and weight multiplier. In all cases, only the maximum energy and bin probability or result values are used.

Usage

```
mcnp_scan_save()
```

Value

spectrum file with maximum energy and MCNP bin value

See Also

Other mcnp tools: `mcnp_cone_angle()`, `mcnp_est_nps()`, `mcnp_matrix_rotations()`, `mcnp_mesh_bins()`, `mcnp_plot_out_spec()`, `mcnp_scan2plot()`, `mcnp_sdef_erg_hist()`, `mcnp_sdef_erg_line()`

Examples

```
# Since this function requires the user
# to copy and paste input, this example
# is set up to provide data for this purpose.
# To run the example, copy and paste the following
# into an input file and delete the hash tags to run.
# my_hist_data <- mcnp_scan_save()
# 0.1000000 3.133122e-05 0.3348260
# 0.4222222 6.731257e-05 0.2017546
# 0.7444444 5.249198e-05 0.4524577
# 1.0666667 2.046046e-04 0.4201954
# 1.3888889 1.525125e-03 0.8049388
# 1.7111111 2.922743e-05 0.7985399
# 2.0333333 5.162954e-03 0.1974694
# 2.3555556 2.048186e-05 0.5011170
# 2.6777778 1.468040e-04 0.7248116
# 3.0000000 1.037092e-04 0.7659850
```

mcnp_sdef_erg_hist — *energy distribution histogram from pasted data*

Description

energy distribution histogram from pasted data

Usage

```
mcnp_sdef_erg_hist(
  entry_mode = "scan",
  my_dir = NULL,
  E_MeV = NULL,
  bin_prob = NULL,
  write_permit = "n",
  log_plot = 0
)
```

Arguments

`entry_mode`	How do you want to enter the data? Default is "scan", allowing you to copy and paste data in at prompts. The other option is "read". In read mode, you identify the data that is already loaded in R.
`my_dir`	Optional directory. The function will write to the working directory by default.
`E_MeV`	Energy bin levels in MeV.
`bin_prob`	Relative probability of bin energy.
`write_permit`	Set this to 'y' to allow writing output to your directory.
`log_plot`	0 = no log axes (default), 1 = log y-axis, 2 = log both axes.

Details

The output includes si# and sp#. The # should be changed to the appropriate distribution number. The data is saved in the global environment and appended to a file in the user's working directory, si_sp.txt. Two plots of the data are provided to the plot window, one with two linear axes and one with two log axes.

Value

A vector of energy bins and probabilities for an energy distribution, formatted as needed for MCNP input. It is designed for copying and pasting into an MCNP input.

See Also

`mcnp_sdef_erg_line()` for data from `RadData`

Other mcnp tools: `mcnp_cone_angle()`, `mcnp_est_nps()`, `mcnp_matrix_rotations()`, `mcnp_mesh_bins()`, `mcnp_plot_out_spec()`, `mcnp_scan2plot()`, `mcnp_scan_save()`, `mcnp_sdef_erg_line()`

Examples

```
## Not run:
mcnp_sdef_erg_hist()

## End(Not run)
```

`mcnp_sdef_erg_line` *Produce MCNP source terms from ICRP 107 data*

Description

Obtain emission data from the RadData package and write to a file for use with the radiation transport code, MCNP.

Usage

```
mcnp_sdef_erg_line(
  desired_RN,
  rad_type = NULL,
  photon = FALSE,
  cut = 0.001,
  erg.dist = 1,
  my_dir = NULL,
  write_permit = "n"
)
```

Arguments

`desired_RN`	Radionuclide in form "Ba-137m"
`rad_type`	Radiation type, leave NULL if selecting photons or select from: 'X' for X-Ray 'G' for Gamma 'AE' for Auger Electron 'IE' for Internal Conversion Electron 'A' for Alpha 'AR' for Alpha Recoil 'B-' for Beta Negative 'AQ' for Annihilation Quanta 'B+' for Beta Positive 'PG' for Prompt Gamma 'DG' for Delayed Gamma 'DB' for Delayed Beta 'FF' for Fission Fragment 'N' for Neutron
`photon`	'Y' to select all rad_types that are photons
`cut`	minimum energy, defaults to 1e-3 MeV
`erg.dist`	energy distribution number for MCNP input
`my_dir`	Optional directory. The function will write an output text file, si_sp.txt to the working directory by default.
`write_permit`	Set this to 'y' to allow writing output to your directory.

Value

a data frame can be saved to memory if desired (i.e. by my_file <- mcnp_sdef_erg_line(...)). For use with MCNP, a text file, 'si_sp.txt' is written to working directory. If file already exists, it is appended. The file contains all emission energies in the si 'card' and the Line indicator, L is included, e.g. si1 L 0.01 (showing a first energy of 0.01 MeV). This is followed by the emission probability of each si entry. An additional text entry is made summing up the probabilities.

See Also

mcnp_sdef_erg_hist() if radioactive emission data is available in histogram form and needs formatting for MCNP input.

Other mcnp tools: mcnp_cone_angle(), mcnp_est_nps(), mcnp_matrix_rotations(), mcnp_mesh_bins(), mcnp_plot_out_spec(), mcnp_scan2plot(), mcnp_scan_save(), mcnp_sdef_erg_hist()

Examples

```
## Not run:
mcnp_sdef_erg_line("Co-60", photon = TRUE, cut = 0.01, erg.dist = 13)
mcnp_sdef_erg_line("Sr-90", rad_type = "B-", cut = 0.01, erg.dist = 15)
mcnp_sdef_erg_line("Am-241", rad_type = "A", cut = 0.01, erg.dist = 23)

## End(Not run)
```

---

neutron_geom_cf — *Solid Angle Correction for Neutron Detectors with Point Source*

Description

Correction factors are needed when an Neutron Rem Detector (NRD) aka "Remball" is used in close proximity to a points source. This formula is per ISO ISO 8529-2-2000 section 6.2. Note, however, that the ISO formula predicts the response. The formula used here takes the inverse to correct for the over-response.

Usage

```
neutron_geom_cf(l, r.d, del = 0.5)
```

Arguments

l	The distance from the center of the detector to the center of the source. Units of l and r.d must be consistent.
r.d	The detector radius. Value for typical NRD is 11 cm. An example is also provided with a Rem 500 detector with a radius of 4.5 cm.
del	The neutron effectiveness factor, default per ISO.

Value

The correction factor for solid angle.

See Also

Other rad measurements: air_dens_cf(), disk_to_disk_solid_angle(), scaler_sim(), tau_estimate()

Examples

```
neutron_geom_cf(l = 11.1, r.d = 11)
neutron_geom_cf(30, 11)
neutron_geom_cf(5, 4.5)
```

---

photons_cs137_hist *File Description:*

Description

This data file was generated in MCNP from a model of Gamma Well Irradiator with no attenuator in place. MCNP will include in the output a histogram of tally results when there is an E Tally Energy card. Results in the output up to MCNP version 6 have no headers, but the columns are:

Usage

```
photons_cs137_hist
```

Format

A `data.frame`

E_max Maximum Energy in MeV

bin_tally Tally result for this bin

R Monte Carlo uncertainty for this bin

---

rate_meter_sim *Ratemeter Simulation*

Description

Plot simulated ratemeter readings once per second for 600 seconds. The meter starts with a reading of zero and builds up based on the time constant. Resolution uncertainty is established to express the uncertainty from reading an analog scale, including the instability of its readings. Many standard references identify the precision or resolution uncertainty of analog readings as half of the smallest increment. This should be considered the single coverage uncertainty for a very stable reading. When a reading is not very stable, evaluation of the reading fluctuation is evaluated in terms of numbers of scale increments covered by meter indication over a reasonable evaluation period.

Usage

```
rate_meter_sim(
  cpm_equilibrium,
  meter_scale_increments,
  trials = 600,
  tau = 9.5,
  log_opt = ""
)
```

Arguments

`cpm_equilibrium`
: The expected count rate.

`meter_scale_increments`
: The meter scale increments.

`trials`
: Number of seconds to run simulation. Default = 600.

`tau`
: equal to the Resistance * Capacitance of the counting circuit. Units = seconds. Default set to 9.5, which provides 90% equilibrium in 22 seconds. If the user does not know the time constant, but has an estimate of equilibrium in some time, use tau.estimate.

`log_opt`
: If logarithmic scale is needed, set to "y". If set to anything but blank (default), scale will be logarithmic.

Value

Plot of simulated meter reading every second..

Examples

```
rate_meter_sim(cpm_equilibrium = 270, meter_scale_increments = seq(100, 1000, 20))
rate_meter_sim(cpm_equilibrium = 2.7e5, meter_scale_increments = seq(2e5, 1e6, 2e4))
rate_meter_sim(450, seq(20, 1000, 20), trials = 1200, tau = 24.8534)
```

RN_bin_screen_phot — *Search for radioisotopes that dominate a specified energy bin*

Description

Identify photon emitters that represent a target range of energies, while screening out other selected energy ranges. This may be helpful for identifying radionuclides in low-definition spectroscopy or in selecting representative spectra for modeling shielding.

Usage

```
RN_bin_screen_phot(
  E_min = 0,
  E_max = 10,
  min_prob = 0,
  min_half_life_seconds = NULL,
  max_half_life_seconds = NULL,
  no_E_min = 0,
  no_E_max = 10,
  no_min_prob = 100,
  no_E_min2 = 0,
  no_E_max2 = 10,
  no_min_prob2 = 100
)
```

Arguments

E_min	target energy range minimum in MeV, default = 0
E_max	target energy range maximum in MeV, default = 10
min_prob	minimum probability of selected range with default = 0.
min_half_life_seconds	minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,
max_half_life_seconds	maximum half-life. See min_half_life_seconds.
no_E_min, no_E_min2	minimum energies in ranges to minimize in MeV, default = 0
no_E_max, no_E_max2	maximum energies in bins to minimize in MeV, default = 10
no_min_prob, no_min_prob2	minimum probability to minimize with default = 100 (no minimum).

Value

radionuclides that match selection criteria

See Also

RN_plot_spectrum()

Other radionuclides: RN_Spec_Act(), RN_index_screen(), RN_info(), RN_plot_df(), RN_plot_search_results(), RN_plot_spectrum(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
RN_bin_screen_phot(
  E_min = 0.1, E_max = 0.3,
  min_prob = 0.4, min_half_life_seconds = 30 * 24 * 3600,
  max_half_life_seconds = 3.153e7, no_E_min = 0.015,
 no_E_max = 0.0999, no_min_prob = 0.05, no_E_min2 = 0.301, no_E_max2 = 10, no_min_prob2 = 0.01
)
```

RN_find_parent — *Find a potential precursor of a radionuclide @description Find a potential parent radionuclide by searching the progeny fields in RadData ICRP_07.NDX*

Description

Find a potential precursor of a radionuclide @description Find a potential parent radionuclide by searching the progeny fields in RadData ICRP_07.NDX

Usage

```
RN_find_parent(RN_select)
```

Arguments

RN_select	identify the radionuclide of interest in the format "Es-254m"

Value

a subset of the data frame RadData::ICRP_07.NDX

Examples

```
Th_230_df <- RN_find_parent("Th-230")

Tl_208_df <- RN_find_parent("Tl-208")
```

RN_index_screen — *Screen radionuclide data to find matches to decay mode, half-life, and total emission energy*

Description

Provides a set of radionuclides matching screening criteria. This is a limited screening based on average energy per transformation. Consider RN_search_phot_by_E, RN_search_alpha_by_E, and RN_search_beta_by_E for spectroscopic measurement matching.

Usage

```
RN_index_screen(
  dk_mode = NULL,
  min_half_life_seconds = NULL,
  max_half_life_seconds = NULL,
  min_E_alpha = NULL,
  min_E_electron = NULL,
  min_E_photon = NULL
)
```

Arguments

dk_mode	default = NULL #' select from: 'A' for Alpha 'B-' for Beta Negative 'B+' for Beta Positive 'EC' for Electron Capture 'IT' for Isomeric Transition 'SF' for Spontaneous Fission
min_half_life_seconds	default = NULL. If half-life is known in units other than seconds, enter with conversion factor, e.g. for 15 minutes, enter min_half_life_seconds = 15 * 60.

max_half_life_seconds
: default = NULL. If half-life is known in units other than seconds, enter with conversion factor, e.g. for 30 minutes, enter max_half_life_seconds = 30 * 60.

min_E_alpha
: default = NULL. This will be used to screen the index for average alpha energy per decay, including all decay branches.

min_E_electron
: default = NULL. This will be used to screen the index for average electron energy per decay, including all decay branches.

min_E_photon
: default = NULL. This will be used to screen the index for average photon energy per decay, including all decay branches.

Value

data frame of radionuclide data from the RadData package index data (RadData::ICRP_07.NDX), matching search criteria.

See Also

Other radionuclides: RN_Spec_Act(), RN_bin_screen_phot(), RN_info(), RN_plot_df(), RN_plot_search_results(), RN_plot_spectrum(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
RN_index_screen(dk_mode = "SF")
RN_index_screen(dk_mode = "IT", max_half_life_seconds = 433 * 3.15e7)
```

RN_info *Quick table of Radionuclide Data from the RadData package*

Description

Access a quick summary of radionuclide data. This is for convenience only and does not replace a more comprehensive view as is available in the Radiological Toolbox doi:10.2172/1201298

Usage

```
RN_info(RN_select)
```

Arguments

RN_select
: identify the radionuclide of interest in the format "Es-254m"

Value

a table including half-life, decay modes, decay progeny, and branch fractions

See Also

Other radionuclides: RN_Spec_Act(), RN_bin_screen_phot(), RN_index_screen(), RN_plot_df(), RN_plot_search_results(), RN_plot_spectrum(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
Es_254m <- RN_info("Es-254m") #saves output to global environment
RN_info("Cf-252")
RN_info("Cs-137")
RN_info("Am-241")
```

RN_plot_df — *Plot results of RN_search functions*

Description

Plots results by radionuclide with E_MeV on x-axis and prob on y-axis.

Usage

```
RN_plot_df(discrete_df, title = deparse(substitute(discrete_df)), log_plot = 0)
```

Arguments

discrete_df	A data frame results from a radsafer search function. Columns must include RN, E_MeV, and prob, and code_AN.
title	Title for chart (default = name of search_results)
log_plot	0 = no log axes (default), 1 = log y-axis, 2 = log both axes.

See Also

Use RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), or RN_search_phot_by_E() and save the results, e.g. save_results <- RN_search_phot_by_E(0.99, 1.01, 13 * 60, 15 * 60, 1e-4)

Other radionuclides: RN_Spec_Act(), RN_bin_screen_phot(), RN_index_screen(), RN_info(), RN_plot_search_results(), RN_plot_spectrum(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
search_results <- RN_search_phot_by_E(0.99, 1.01, 13 * 60, 15 * 60, 1e-4)
RN_plot_df(search_results, title = "example1", log_plot = 0)
search_results <- RN_save_spectrum("In-115m", photon = TRUE)
RN_plot_df(search_results, title = "In-115m", log_plot = 0)
```

RN_plot_search_results

Plot results of RN_search functions

Description

Plots results by radionuclide with E_MeV on x-axis and prob on y-axis. **[Deprecated]**: This function is deprecated and will be removed in a future package revision. For now, it is still usable. The replacement, `RN_plot_df` plots saved data frames including those saved with search functions.

Usage

```
RN_plot_search_results(
  discrete_df,
  title = deparse(substitute(discrete_df)),
  log_plot = 0
)
```

Arguments

`discrete_df`	A data frame results from a `radsafer` search function. Columns must include RN, E_MeV, and prob, and code_AN.
`title`	Title for chart (default = name of search_results)
`log_plot`	0 = no log axes (default), 1 = log y-axis, 2 = log both axes.

See Also

Use `RN_search_alpha_by_E()`, `RN_search_beta_by_E()`, or `RN_search_phot_by_E()` and save the results, e.g. save_results <- RN_search_phot_by_E(0.99, 1.01, 13 * 60, 15 * 60, 1e-4)

Other radionuclides: `RN_Spec_Act()`, `RN_bin_screen_phot()`, `RN_index_screen()`, `RN_info()`, `RN_plot_df()`, `RN_plot_spectrum()`, `RN_save_spectrum()`, `RN_search_alpha_by_E()`, `RN_search_beta_by_E()`, `RN_search_phot_by_E()`

Examples

```
search_results <- RN_search_phot_by_E(0.99, 1.01, 13 * 60, 15 * 60, 1e-4)
RN_plot_search_results(search_results, title = "example1", log_plot = 0)
```

RN_plot_spectrum *Plot radionuclide emission spectra directly from ICRP-107 data.*

Description

Plot emission spectra based on radionuclide and desired radiation type. Plot on log axes if desired. Select cutoff value for probability optional, included at 1% by default. Plot includes energy times probability for dosimetric importance comparisons.

Usage

```
RN_plot_spectrum(
  desired_RN,
  rad_type = NULL,
  photon = FALSE,
  log_plot = 0,
  prob_cut = 0.01
)
```

Arguments

`desired_RN`	Radionuclide in form "Ba-137m"
`rad_type`	Radiation type, leave NULL if selecting photons or select from: 'X' for X-Ray 'G' for Gamma 'AE' for Auger Electron 'IE' for Internal Conversion Electron 'A' for Alpha 'AR' for Alpha Recoil 'B-' for Beta Negative 'AQ' for Annihilation Quanta 'B+' for Beta Positive 'PG' for Prompt Gamma 'DG' for Delayed Gamma 'DB' for Delayed Beta 'FF' for Fission Fragment 'N' for Neutron
`photon`	Use only if you do not specify `rad_type`. TRUE will select all rad_types that are photons. Note that if you select `rad_type = "G"`, for example, you will not get X-rays or Annihilation Quanta (the 0.511 MeV photon from pair annihilation).
`log_plot`	0 = no log axes, 1 (default) = log y-axis, 2 = log both axes. This argument is ignored for B- plots.
`prob_cut`	minimum probability defaults to 0.01

Value

plot of spectrum

See Also

Other radionuclides: RN_Spec_Act(), RN_bin_screen_phot(), RN_index_screen(), RN_info(), RN_plot_df(), RN_plot_search_results(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
RN_plot_spectrum(
  desired_RN = c("Sr-90", "Y-90"), rad_type = "B-",
  photon = FALSE, prob_cut = 0.01
)
RN_plot_spectrum(
  desired_RN = c("Co-60", "Ba-137m"), rad_type = NULL,
  photon = TRUE, prob_cut = 0.015
)
RN_plot_spectrum(
  desired_RN = c("Co-60", "Ba-137m"), rad_type = NULL,
  photon = TRUE, log_plot = 0
)
RN_plot_spectrum(desired_RN = c("Co-60", "Ba-137m"), rad_type = "G")
RN_plot_spectrum(
  desired_RN = c("Pu-238", "Pu-239", "Am-241"), rad_type = "A",
  photon = FALSE, prob_cut = 0.01, log_plot = 0
)
```

RN_save_spectrum   *Save radionuclide emission spectra.*

Description

Save emission spectra based on radionuclide and desired radiation type. Select cutoff value for probability optional, included at 1% by default.

Usage

```
RN_save_spectrum(desired_RN, rad_type = NULL, photon = FALSE, prob_cut = 0)
```

Arguments

desired_RN	Radionuclide in form "Ba-137m"
rad_type	Radiation type, leave NULL if selecting photons or select from: 'X' for X-Ray 'G' for Gamma 'AE' for Auger Electron

'IE' for Internal Conversion Electron
'A' for Alpha
'AR' for Alpha Recoil
'B-' for Beta Negative
'AQ' for Annihilation Quanta
'B+' for Beta Positive
'PG' for Prompt Gamma
'DG' for Delayed Gamma
'DB' for Delayed Beta
'FF' for Fission Fragment
'N' for Neutron

photon: Use only if you do not specify `rad_type`. TRUE will select all rad_types that are photons. Note that if you select `rad_type = "G"`, for example, you will not get X-rays or Annihilation Quanta (the 0.511 MeV photon from pair annihilation).

prob_cut: minimum probability defaults to 0.01

Value

Dataframe with energy spectra - including probability of emission quantum, or, for beta, the probability density.

See Also

Other radionuclides: RN_Spec_Act(), RN_bin_screen_phot(), RN_index_screen(), RN_info(), RN_plot_df(), RN_plot_search_results(), RN_plot_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
Sr_Y_90_df <- RN_save_spectrum(desired_RN = c("Sr-90", "Y-90"), rad_type = "B-",
photon = FALSE, prob_cut = 0.01)
Co_60_Ba_137m_p_df <- RN_save_spectrum(desired_RN = c("Co-60", "Ba-137m"), rad_type = NULL,
photon = TRUE, prob_cut = 0.015)
Co_60_Ba_137m_g_df <- RN_save_spectrum(desired_RN = c("Co-60", "Ba-137m"), rad_type = "G")
actinide_a_df <- RN_save_spectrum(desired_RN = c("Pu-238", "Pu-239", "Am-241"), rad_type = "A",
photon = FALSE, prob_cut = 0.01)
```

RN_search_alpha_by_E *Search for alpha*

Description

Search for alpha emission based on energy, half-life and minimum probability.

Usage

```
RN_search_alpha_by_E(
  E_min = 0,
  E_max = 10,
  min_half_life_seconds = NULL,
  max_half_life_seconds = NULL,
  min_prob = 0
)
```

Arguments

`E_min`	minimum energy in MeV, default = 0
`E_max`	maximum energy in MeV, default = 10
`min_half_life_seconds`	minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,
`max_half_life_seconds`	maximum half-life. See min_half_life_seconds.
`min_prob`	minimum probability with default = 0.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

Other radionuclides: `RN_Spec_Act()`, `RN_bin_screen_phot()`, `RN_index_screen()`, `RN_info()`, `RN_plot_df()`, `RN_plot_search_results()`, `RN_plot_spectrum()`, `RN_save_spectrum()`, `RN_search_beta_by_E()`, `RN_search_phot_by_E()`

Examples

```
# between 7 and 8 MeV
search_results <- RN_search_alpha_by_E(7, 8)

# 1-4 MeV; half-life between 1 and 4 hours
search_results <- RN_search_alpha_by_E(1, 4, 1 * 3600, 4 * 3600)

# between 7 and 10 MeV with at least 1e-3 probability
search_results <- RN_search_alpha_by_E(7, 10, min_prob = 1e-3)
```

RN_search_beta_by_E *Search for beta*

Description

Search for beta emission based on maximum energy and half-life.

Usage

```
RN_search_beta_by_E(
  E_max,
  min_half_life_seconds = NULL,
  max_half_life_seconds = NULL
)
```

Arguments

- `E_max` maximum energy in MeV, default = 10
- `min_half_life_seconds` minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,
- `max_half_life_seconds` maximum half-life. See min_half_life_seconds.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

Other radionuclides: `RN_Spec_Act()`, `RN_bin_screen_phot()`, `RN_index_screen()`, `RN_info()`, `RN_plot_df()`, `RN_plot_search_results()`, `RN_plot_spectrum()`, `RN_save_spectrum()`, `RN_search_alpha_by_E()`, `RN_search_phot_by_E()`

Examples

```
# Max beta at least 2 MeV
search_results <- RN_search_beta_by_E(2)
# Max beta at least 2 MeV and half-life between 1 s and 1 h
search_results <- RN_search_beta_by_E(2, 1, 3600)

# Max beta at least 1 MeV and half-life between 1 d and 2 d
search_results <- RN_search_beta_by_E(1, 3600 * 24, 2 * 3600 * 24)
```

`RN_search_phot_by_E` *Search for photon*

Description

Search for photon emission based on energy, half-life and minimum probability.

Usage

```
RN_search_phot_by_E(
  E_min = 0,
  E_max = 10,
  min_half_life_seconds = NULL,
  max_half_life_seconds = NULL,
  min_prob = 0
)
```

Arguments

`E_min` minimum energy in MeV, default = 0

`E_max` maximum energy in MeV, default = 10

`min_half_life_seconds` minimum half-life in seconds. Use multiplier as needed, e.g. 3 * 3600 for 3 hours. Default = NULL,

`max_half_life_seconds` maximum half-life. See min_half_life_seconds.

`min_prob` minimum probability with default = 0.

Value

search results in order of half-life. Recommend assigning results to a viewable object, such as 'search_results'

See Also

`RN_plot_spectrum()`

Other radionuclides: `RN_Spec_Act()`, `RN_bin_screen_phot()`, `RN_index_screen()`, `RN_info()`, `RN_plot_df()`, `RN_plot_search_results()`, `RN_plot_spectrum()`, `RN_save_spectrum()`, `RN_search_alpha_by_E()`, `RN_search_beta_by_E()`

Examples

```
# between 1 and 1.2 MeV, between 6 and 6.2 hours half-life,
# ... probability at least 1e-4
search_results <- RN_search_phot_by_E(1, 1.2, 6 * 3600, 6.2 * 3600, 1e-4)

# between 0.1 and 0.15 MeV, between 1 and 3 million years half-life
search_results <- RN_search_phot_by_E(0.1, 0.15, 1e6 * 3.153e7, 3e6 * 3.153e7)
```

RN_Spec_Act *Specific Activity*

Description

Provides specific activity of a radionuclide in Bq/g.

Usage

```
RN_Spec_Act(RN_select, numeric = "n")
```

Arguments

RN_select	identify the radionuclide of interest in the format "Es-254m". For multiple specific activities, combine the radionuclides of interest in the form, c("At-219", "Es-251").
numeric	default is "n" and a data frame is returned showing the radionuclide, its relative specific activity, and the units. If "y", or any other option is selected for the numeric parameter, specific activity results are delivered as numeric.

Value

specific activity in Bq / g

See Also

Other radionuclides: RN_bin_screen_phot(), RN_index_screen(), RN_info(), RN_plot_df(), RN_plot_search_results(), RN_plot_spectrum(), RN_save_spectrum(), RN_search_alpha_by_E(), RN_search_beta_by_E(), RN_search_phot_by_E()

Examples

```
RN_Spec_Act("Ac-230")
RN_Spec_Act(c("At-219", "Es-251"))
RN_Spec_Act("Pd-96", numeric = "y")
RN_Spec_Act(c("Cs-137", "Ba-137m"), numeric = "y")
```

scaler_sim *Count Room Scaler Simulation*

Description

Returns a plotted distribution of results for a scaler model based on the Poisson distribution. Inputs and outputs in counts per minute.

Usage

```
scaler_sim(true_bkg, true_samp, ct_time, trials = 1e+05)
```

Arguments

`true_bkg`	True background count rate in counts per minute.
`true_samp`	True sample count rate in counts per minute.
`ct_time`	Count time in minutes.
`trials`	Number of sample values, default = 1e5.

Value

A histogram of all trial results including limits for +/- 1 standard deviation.

See Also

Other rad measurements: `air_dens_cf()`, `disk_to_disk_solid_angle()`, `neutron_geom_cf()`, `tau_estimate()`

Examples

```
scaler_sim(true_bkg = 5, true_samp = 10, ct_time = 1, trials = 1e5)
scaler_sim(true_bkg = 50, true_samp = 30, ct_time = 1, trials = 1e5)
```

`stay_time` *Stay time for radiation work.*

Description

Calculate stay time for radiation work.

Usage

```
stay_time(dose_rate, dose_allowed, margin = 20)
```

Arguments

`dose_rate`	Dose rate per hour for the work - units consistent with dose allowance, e.g. mRem/h, microSv/h.
`dose_allowed`	Dose that can not be exceeded for this job.
`margin`	Percent margin to protect limit, default = 20 percent.

Value

Time in minutes allowed for the work.

Examples

```
stay_time(dose_rate = 100, dose_allowed = 50, margin = 20)
```

`tau_estimate` *Estimate tau parameter for* `ratemeter_sim`

Description

If the time constant is not known, but the vendor specifies that the ratemeter will reach some percentage of equilibrium in some number of seconds, use this function to estimate tau.

Usage

```
tau_estimate(pct_eq, t_eq)
```

Arguments

`pct_eq`	Percent equilibrium
`t_eq`	Time, in seconds, to the given percent equilibrium is achieved.

Value

tau, the time constant, in seconds.

See Also

Other rad measurements: `air_dens_cf()`, `disk_to_disk_solid_angle()`, `neutron_geom_cf()`, `scaler_sim()`

Examples

```
tau_estimate(pct_eq = 90, t_eq = 22)
```

Index

∗ **datasets**
photons_cs137_hist, 19
∗ **decay corrections**
dk_correct, 5
dk_pct_to_num_half_life, 7
dk_time, 7
∗ **mcnp tools**
mcnp_cone_angle, 9
mcnp_est_nps, 10
mcnp_matrix_rotations, 11
mcnp_mesh_bins, 12
mcnp_plot_out_spec, 13
mcnp_scan2plot, 14
mcnp_scan_save, 15
mcnp_sdef_erg_hist, 16
mcnp_sdef_erg_line, 17
∗ **rad measurements**
air_dens_cf, 3
disk_to_disk_solid_angle, 3
neutron_geom_cf, 18
scaler_sim, 32
tau_estimate, 34
∗ **rad_measurements**
rate_meter_sim, 19
∗ **radionuclides**
RN_bin_screen_phot, 20
RN_index_screen, 22
RN_info, 23
RN_plot_df, 24
RN_plot_search_results, 25
RN_plot_spectrum, 26
RN_save_spectrum, 27
RN_search_alpha_by_E, 28
RN_search_beta_by_E, 30
RN_search_phot_by_E, 31
RN_Spec_Act, 32

air_dens_cf, 3, *4*, *18*, *33, 34*

disk_to_disk_solid_angle, *3*, 3, *18*, *33, 34*

dk_correct, 5, *7, 8*
dk_pct_to_num_half_life, *6*, 7, *8*
dk_time, *6, 7*, 7

half_life_2pt, 8
hvl, 9

mcnp_cone_angle, 9, *10, 11*, *13–16*, *18*
mcnp_est_nps, *10*, 10, *11*, *13–16*, *18*
mcnp_matrix_rotations, *10*, 11, *13–16*, *18*
mcnp_mesh_bins, *10, 11*, 12, *13–16*, *18*
mcnp_plot_out_spec, *10, 11*, *13*, 13, *14–16*, *18*
mcnp_scan2plot, *10, 11*, *13*, 14, *15, 16*, *18*
mcnp_scan_save, *10, 11*, *13, 14*, 15, *16*, *18*
mcnp_scan_save(), *13*
mcnp_sdef_erg_hist, *10, 11*, *13–15*, 16, *18*
mcnp_sdef_erg_hist(), *18*
mcnp_sdef_erg_line, *10, 11*, *13–16*, 17
mcnp_sdef_erg_line(), *16*

neutron_geom_cf, *3, 4*, 18, *33, 34*

photons_cs137_hist, 19

rate_meter_sim, 19
RN_bin_screen_phot, 20, *23–25*, *27–32*
RN_find_parent, 21
RN_index_screen, *21*, 22, *24, 25*, *27–32*
RN_info, *21*, *23*, 23, *24, 25*, *27–32*
RN_plot_df, *21*, *23, 24*, 24, *25*, *27–32*
RN_plot_search_results, *21*, *23, 24*, 25, *27–32*
RN_plot_spectrum, *21*, *23–25*, 26, *28–32*
RN_plot_spectrum(), *21*, *31*
RN_save_spectrum, *21*, *23–25*, *27*, 27, *29–32*
RN_save_spectrum(), *24*
RN_search_alpha_by_E, *21–25*, *27, 28*, 28, *30–32*
RN_search_alpha_by_E(), *24, 25*

RN_search_beta_by_E, *21–25*, *27–29*, 30, *31, 32*
RN_search_beta_by_E(), *24, 25*
RN_search_phot_by_E, *21–25*, *27–30*, 31, *32*
RN_search_phot_by_E(), *24, 25*
RN_Spec_Act, *21*, *23–25*, *27–31*, 32

scaler_sim, *3, 4*, *18*, 32, *34*
stay_time, 33

tau_estimate, *3, 4*, *18*, *33*, 34