Development of the radioactivedecay Python package for radioactive decay calculations

*Alex Malins[1]

[1]JAEA

The inventories of radioactive nuclides present at nuclear power plants, research institutions, hospitals, in the environment, etc., change continuously due to radioactive decay. radioactivedecay is an easy-to-use Python package that can calculate the changes in activities of radionuclides by radioactive decay to high precision. The source code is published via GitHub and the Python Package Index (PyPI). This talk will explain the features and the development process of the code.

Keywords: radioactivity, radioactive decay, decay chain, calculation code, Python

1. Introduction

radioactivedecay is a free, open-source Python package for radioactive decay calculations. It can be used to calculate decayed activities of inventories of radionuclides for arbitrary time periods. The package supports decay chains of radionuclides, metastable nuclear isomers and branching decays. It was developed as an easy-to-access and use code for practitioners in research and industrial workplaces, and for members of the public.

2. Method and implementation details

The radioactive decay differential equations are solved analytically using a matrix exponential method [1]. The solution has the form $\mathbf{N}(t) = Ce^{\Lambda_d t}C^{-1}\mathbf{N}(0)$, where $\mathbf{N}$ is a vector with the numbers of nuclei for each radionuclide ($N_i = A_i/\lambda_i$, A_i is the activity and λ_i is the decay constant). C and C^{-1} are lower triangular matrices calculated as $C_{ij} = \begin{cases} 1 & \text{for } i = j \\ \sum_{k=j}^{i-1} b_{ki}\lambda_k C_{kj}/(\lambda_j - \lambda_i) & \text{for } i > j \end{cases}$ and $C_{ij}^{-1} = \begin{cases} 1 & \text{for } i = j \\ -\sum_{k=j}^{i-1} C_{ik}\, C_{kj}^{-1} & \text{for } i > j \end{cases}$, where b_{ki} is the branching fraction from radionuclide k to i. $e^{\Lambda_d t}$ is a diagonal matrix with $e_{ii}^{\Lambda_d t} = e^{-\lambda_i t}$, and t is the decay time.

radioactivedecay is coded in the Python programming language. Matrices C and C^{-1} are independent of time, so they are pre-calculated and imported as part of a dataset into the code. By default radioactivedecay uses the radioactive decay data from ICRP 107 [2] for calculations. However the code is written flexibly to accommodate other datasets.

As C, C^{-1} and $e^{\Lambda_d t}$ are sparse matrices, radioactivedecay uses SciPy [3] and NumPy [4] routines to compute the decay equation efficiently using double-precision floating-point operations. Undesirable round-off errors and significance issues can occur for some decay computations with double-precision floating-point operations, e.g. for decay chains containing radionuclides with orders of magnitude different in half-lives. Therefore radioactivedecay includes an alternative high numerical precision calculation mode based on SymPy [5] arbitrary precision computations.

3. Testing and distribution

The code was validated by checking against decay calculation results from PyNE [6] and Radiological Toolbox [7]. Discrepancies were investigated and accounted for whenever greater than computational uncertainties.

radioactivedecay is distributed via PyPI at https://pypi.org/project/radioactivedecay/. The source code is hosted at https://github.com/alexmalins/radioactivedecay and documentation at https://alexmalins.com/radioactivedecay/.

References

[1] M. Amaku, P.R. Pascholati and V.R. Vanin, Comp. Phys. Comm. 181, 21-23 (2010). https://doi.org/10.1016/j.cpc.2009.08.011
[2] ICRP Publication 107: Nuclear Decay Data for Dosimetric Calculations. Ann. ICRP 38 (3), 1-96 (2008).
[3] P. Virtanen et al. Nat. Methods 17, 261-272 (2020). https://doi.org/10.1038/s41592-019-0686-2
[4] C.R. Harris et al. Nat. 585, 357-362 (2020). https://doi.org/10.1038/s41586-020-2649-2
[5] A. Meurer et al. PeerJ Comp. Sci. 3, e103 (2017). https://doi.org/10.7717/peerj-cs.103
[6] N. Hertel, K. Eckerman and C. Sun, Trans. Am. Nucl. Soc. 113, 977-980 (2015). https://ans.org/pubs/transactions/a_38022
[7] A.M. Scopatz et al., Trans. Am. Nucl. Soc. 107, 985-987 (2012). https://ans.org/pubs/transactions/a_14978