ORNL/TM-2020/1780

Bateman Equation Adaptation for Solving and Integrating Peak Activity into EPA ELCR and Dose Models

Leslie D. Galloway
Michael B. Bellamy
Fredrick G. Dolislager
Hayden J. Ringer
Ethan A. Asano
Debra J. Stewart
Kathryn A. Noto
Karessa L. Manning
Louis J. Gross
Anthony Q. Armstrong

Approved for public release.
Distribution is unlimited.

October 2020

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2020/1780

Environmental Sciences Division
Center for Radiation Protection Knowledge

BATEMAN EQUATION ADAPTATION FOR SOLVING AND INTEGRATING PEAK ACTIVITY INTO EPA ELCR AND DOSE MODELS

Leslie D. Galloway, Michael B. Bellamy, Fredrick G. Dolislager,
Hayden J. Ringer, Ethan A. Asano, Debra J. Stewart, Kathryn A. Noto,
Karessa L. Manning, Louis J. Gross, Anthony Q. Armstrong

Date Published: October 2020

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

CONTENTS iii
LIST OF FIGURES v
ACRONYMS vii
ABSTRACT ix
1. INTRODUCTION 1
2. PARENT-ONLY PRG WITH DECAY 2
3. SECULAR EQUILIBRIUM PRG 2
4. PEAK PRG 3
5. THE BATEMAN SOLVER 5
6. MATHEMATICS OF CRAM 6
7. GENERATION OF THE UNIT ACTIVITY FILE 7
8. CONVERSION OF THE UAF TO ELCR RATES 10
9. HOW PERIOD OF PEAK ELCR IS FOUND 13
10. HOW A PEAK PRG IS CALCULATED 13
11. EXAMPLE CALCULATION OF A PEAK PRG FOR RA-226 14
12. COMPARISON OF CURRENT AND PAST EPA PRG OPTIONS to the PEAK PRG 15
13. PEAK ELCR CALCULATIONS USING UAF AND AUC 16
14. CONCLUSION 17
15. REFERENCES 17
APPENDIX A. BATEMAN SOLVER (C++ ODE METHOD) A-3
APPENDIX B. CRAM MATRIX (PER PARENT AND CHAIN METHOD) B-3
APPENDIX C. CRAM BATEMAN SOLVER C-3
APPENDIX D. CRAM BATEMAN OUTPUT D-3
APPENDIX E. PEAK PRG SOLVER E-3
APPENDIX F. PEAK RISK (ELCR) SOLVER F-3
APPENDIX G. BATEMAN SOLVER (C++ ODE METHOD) UAF FOR RA-226 .... G-3
APPENDIX H. BATEMAN SOLVER (CRAM METHOD) UAF FOR RA-226 .... H-3

LIST OF FIGURES

Figure 1. Ra-226 decay chain for all progeny from RAIS decay chain tool. 8
Figure 2. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Ra-226 at 1000 years with initial activity of 1 (Bq, pCi, etc.). 9
Figure 3. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Am-241 at 100 years with initial activity of 1 (Bq, pCi, etc.). 9
Figure 4. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Th-228 at 10 years with initial activity of 1 (Bq, pCi, etc.). 10
Figure 5. Ra-226 relative risk for the soil ingestion route for the composite worker, 25-year duration. 11
Figure 6. Ra-226 relative risk for the soil inhalation route for the composite worker, 25-year duration. 11
Figure 7. Ra-226 relative risk for the soil external exposure route for the composite worker, 25-year duration. 12
Figure 8. Ra-226 relative risk for total soil exposure for the composite worker, 25-year duration. 12
Figure 9. Peak PRG calculator output for Ra-226 chain without decay 14
Figure 10. Peak PRG calculator output for Ra-226 chain accounting for peak ELCR. 15
Figure 11. Comparison of current (and one discontinued) PRG calculator output options for Ra-226. 15
Figure 12. Comparison of current PRG calculator output options for Th-228. 16
Figure 13. Ra-226 peak risk for composite worker soil external exposure over 25 years. 17

ACRONYMS

AUC	area under the curve
CRAM	Chebyshev Rational Approximation Method; introduced for solving decay/burnup equations; shown to be capable of providing accurate and efficient solutions without the need to exclude the short-lived radionuclides.
DCC	dose compliance concentration; the activity of a radionuclide in a media that is derived from an acceptable target dose limit
ED	exposure duration (in years)
ELCR	excess lifetime cancer risk or the probability of getting cancer from a measured activity of a radionuclide
EP	exposure parameter
EPA	US Environmental Protection Agency
IP	isotope parameter
NORM	naturally occurring radioactive material
ODE	ordinary differential equations
OS	operating system
PRG	preliminary remediation goal, the activity of a radionuclide in a media that is derived from an acceptable target ELCR
RAIS	Risk Assessment Information System
ROPC	radionuclide of potential concern or any radionuclide detected at a Superfund site above background
ROC	radionuclides of concern or any radionuclide whose activity in a media is greater than a PRG or DCC calculated for the same media
SE	secular equilibrium
TENORM	technologically enhanced naturally occurring radioactive material
TR	target ELCR
UAF	unit activity file; the time-series of activities across an entire decay chain starting with an initial activity of 1 (pCi or Bq) and decayed across time until activity is depleted or 1E+16 years, whichever comes first

ABSTRACT

This report presents a new preliminary remediation goal (PRG) methodology, called a peak PRG, which calculates the activity of the parent radionuclide to be protective of the peak excess lifetime cancer risk (ELCR) for the entire decay chain. With this important addition to the current set of PRG models offered by the U.S. Environmental Protection Agency (EPA), all waste profiles can now be addressed at Superfund sites. For waste profiles that contain a refined radionuclide product with a relatively long half-life, the progeny may present more ELCR in the distant future than the parent in the present. By modeling the decay of the parent, with the ingrowth of all progeny, a protective peak PRG can be calculated and used to compare against current monitoring or sample data for the protection of future receptors.

The mathematical model used to calculate abundance and activity as a function of time is called the Bateman equation. A unique C++ Bateman solver was written to calculate the relative activity of the complete decay chain for all 1252 radionuclides identified in ICRP Publication 107 (ICRP 2007). In addition, a Python solution was developed using the Chebyshev Rational Approximation Method (CRAM). The results are stored as unit activity files (UAFs) for access by the EPA PRG calculators. As the UAF is a time-series of activity points per year, a conversion to ELCR is necessary. Risk rates are calculated for each member in the chain; these rates are used to create curves in units of ELCR/year for each exposure route. The period of peak ELCR is then found using numerical methods with peak search algorithms. The ELCR rate as a function of time is calculated for each chain member and summed to determine the total ELCR rate function for the chain. Using Simpson's Rule for numerical integration (unequal intervals) with the points provided from the UAF, the area under the curve, or peak ELCR, is calculated for the exposure duration interval of peak risk.

Benefits of the peak PRG are that site risk is not over or underestimated, as can be the case using current EPA PRG models, and the future dates of peak ELCR are known. This knowledge helps make remedial decisions that are protective of future populations while reducing remediation costs, by not remediating a site where decay will reduce the ELCR naturally in a reasonable amount of time. Peak PRGs can also help design waste criteria and landfill caps protective of future ELCRs. This new peak PRG model will also be applicable to the EPA's dose compliance concentrations (DCCs) calculation websites.

1. INTRODUCTION

The U.S. Environmental Protection Agency (EPA) presents a new radionuclide preliminary remediation goal (PRG) model that, when combined with the existing PRG models, will address all Superfund site conditions. In the EPA Superfund program, PRGs are used in the data screening step to reduce the number of radioactive contaminants that are carried through a quantitative risk assessment. If the activity for a radionuclide of potential concern (ROPC) is below its PRG, then no further analysis of that ROPC needs to occur. ROPCs that exceed their PRGs are called radionuclides of concern (ROC) and have their excess lifetime cancer risk (ELCR) calculated based on their measured activity at the site. The objective of this report is to provide the technical basis for the calculation of PRG values that consider the decay and ingrowth of radionuclide progeny.

Currently, the default PRG models used in the EPA Superfund calculators (U.S. EPA 2020a, b, and c) are based on secular equilibrium (SE), where the parent radionuclide is continually renewed and all progeny are present at the same activity of the parent, in the case of a straight decay chain. In the case of a branched decay chain, progeny are present at their fractional activity to the parent. A second PRG model is also provided in the EPA calculators based solely on the decay of the parent without concern for progeny ingrowth; this PRG is always underprotective in the case of a radionuclide with progeny, where the progeny are not considered individually by the risk assessor. The SE scenario is not representative of all Superfund sites and is often overly protective, particularly when the ROPC has a relatively short half-life. This report presents a new PRG model, proposed by the EPA, that remains protective but is based on the period of peak ELCR throughout the state of the decay chain (called a "peak PRG"). These peak PRG calculations assume that a refined radionuclide product is generated, released into the environment, and decays without regeneration of the parent. For some ROPCs with relatively long half-lives, the progeny may present more ELCR in the distant future than the parent in the present. By modeling the decay of the parent with the ingrowth of progeny, a protective peak PRG can be predicted and used to compare against current monitoring or sample data for the protection of future receptors. Additionally, the models used to predict peak PRGs can be used to calculate the period of greatest ELCR in the future. This new peak PRG model will also be applicable to the EPA's dose compliance concentration (DCC) calculation websites (U.S. EPA 2020d, e, and f).

Superfund sites can have different radionuclide waste profiles that require matching ROPCs to specific PRG models. A large number of Superfund sites are abandoned uranium mines, where SE PRGs are likely appropriate due to the presence of wastes containing naturally occurring radioactive materials (NORM). Other Superfund sites contain a refined product (e.g., discarded smoke detectors containing Am-241), and a peak PRG would be appropriate. Still, many sites are a mixture of ROPCs, where parent only PRGs and peak PRGs can be appropriate. Finally, more complicated Superfund sites exist where the decay chain had a desirable progeny removed (i.e., radium for luminescent dials and thorium for lantern mantels) for industrial use, and multiple PRG models are required to handle the disequilibrium. These technologically enhanced naturally occurring radioactive material (TENORM) sites require careful analysis to determine what part of the chain is still growing in and what part has achieved equilibrium.

The methods for calculating PRGs are based on Risk Assessment Guidance for Superfund: Volume I, Human Health Evaluation Manual (Part B, Development of Risk-Based Preliminary Remediation Goals) (U.S. EPA 1991). PRGs are determined for a defined usage scenario (Residential, Indoor Worker, Outdoor Worker, etc.) with appropriate routes of exposure (ingestion, inhalation, external exposure, etc.) for defined media (soil, water, air, produce, biota, etc.) per isotope. Additionally, exposure parameters (EP), such as exposure time, duration, and frequency, and isotope-specific parameters (IP), such as risk coefficients (ORNL 2014a), biota uptake factors (ORNL 2016), gamma shielding factors (ORNL 2014b), and area correction factors (ORNL 2014c), are applied. This combination of EPs and IPs for route-specific equations results in a unique PRG for each ROPC.

2. PARENT-ONLY PRG WITH DECAY

The EPA currently offers two PRG models in their online calculators. The first PRG model is the simplest and based solely on decay of the parent without consideration of progeny ingrowth or parent regeneration. This PRG model is used in two PRG output options (currently options 2 and 3) in the calculators. Equation 1 represents a total PRG for a single land use, single media, multiple exposure routes, and a parent radionuclide that is decaying without accounting for the ELCR contribution from progeny.

$$\text{Parent only PRG}$$

$$PRG_{tot} = \frac{1}{\left(\sum_{i=1}^{n}\left(\frac{1}{PRG_{route-i}}\right)\right)\left(\frac{t(yr)\times\lambda\left(\frac{1}{yr}\right)}{(1-e^{-\lambda t})}\right)} \quad \text{Equation 1}$$

$where$:
$n \text{ is the total number of exposure routes}$

PRG output option 3 only presents the PRGs for selected radionuclides using Equation 1. PRG output option 2 also uses equation 1 but provides the PRGs for all members of the decay chain. This PRG model is only directly applicable when the parent decays directly to a stable isotope. It is most useful for radionuclides that decay faster than the exposure duration. For long-lived radionuclides, the decay term has no impact on the PRG. In the case of multiple progeny from the parent, only using the parent PRG to screen a site is not protective, as the total ELCR in the future may be greater than the current parent ELCR. To account for the ELCR of progeny, the activity of each chain member needs to be measured to calculate a corresponding PRG.

PRG output option 2 automatically generates these progeny PRGs in addition to the parent. Also, the future activity of each chain member should be determined and compared to PRGs. This approach is costly, as many of the progeny in a chain can be very short-lived and difficult to measure. It would be convenient to roll up the progeny ELCR contribution into the PRG of the parent and account for future ELCR; this is the intent of the peak PRG outlined in section 4.

In the past, PRG output option 2 could be manipulated to simulate a PRG that was inclusive of progeny ELCR. The calculator could be run in user-provided mode, which gave the user the ability to change the half-lives of each progeny to match the parent. This results in simulating equilibrium of progeny with the parents. The PRG for each chain member is then compared to environmental data. This approach is inferior to using the peak PRG and generally should no longer be used.

3. SECULAR EQUILIBRIUM PRG

The second PRG model, currently PRG output option 1 (default) in the EPA Superfund calculators, is based on SE, where the parent radionuclide is continually renewed and all progeny are present at the same activity of the parent, in the case of a straight decay chain. In the case of a branched decay chain, progeny are present at their fractional activity to the parent. The SE scenario is the most protective possible PRG and can therefore be overprotective for short-lived chains. Equation 2 accounts for the fractional ELCR contribution from all progeny in the chain for each route and is used in Equation 3, which presents a total parent SE PRG for a single land use, single media, and multiple exposure routes.

$Total\ secular\ equilibrium\ PRG\ for\ parent\ radionuclide$

$$PRG_{SE-tot} = \frac{1}{\left(\sum_{i=1}^{n}\left(\frac{1}{PRG_{SE-route-i}}\right)\right)} \qquad \text{Equation 2}$$

$where$:
$n = total\ number\ of\ exposure\ routes$

$Route\ secular\ eqilibrium\ PRG\ for\ parent\ radionuclide$

$$PRG_{SE-route} = \frac{1}{\sum_{i=1}^{n}\left(\frac{PRG}{FC}\right)_i} \qquad \text{Equation 3}$$

$where$:
$n\ is\ the\ total\ number\ of\ radionuclides\ in\ the\ decay\ chain$
$FC\ is\ the\ fractional\ contribution\ of\ the\ i^{th}\ radionuclide\ in\ the\ decay\ chain$
$PRG\ is\ the\ PRG\ for\ the\ i^{th}\ radionuclide\ in\ the\ decay\ chain\ without\ decay$

Although this PRG model assures users that a site with ROPC activities below the SE PRGs is always going to be acceptable at any point in the future, SE is overly conservative for many radionuclides encountered at Superfund sites, and remediation to SE PRG standards could be unnecessarily costly. The SE PRG model assumes continual renewal of the parent, which may occur at an active ore processing facility; however, these types of active facilities are not expected to be part of a Superfund site remediation.

4. PEAK PRG

A PRG that is applicable to many Superfund sites, not overprotective, and accounts for the ingrowth of progeny over the entire decay chain is called a peak PRG. These peak PRG calculations assume that a refined radionuclide product is generated, released into the environment, and decays without source renewal from the parent. As a refined parent radionuclide decays, progeny that grow in the future may present a greater ELCR than the parent in the present. By modeling the decay of the parent with the ingrowth of progeny, a protective peak PRG for future receptors can be calculated and used to compare against current monitoring data.

Inherent in the two PRG models currently presented in EPA calculators is an ELCR rate over time. The parent-only PRG with decay assumes that the activity, and hence the ELCR rate, goes down exponentially over time. The decay curve for any radionuclide can be solved by equation 4.

$$N(t) = N_0 e^{-\lambda t} \qquad \text{Equation 4}$$

$where$:
$t\ is\ the\ time\ in\ years$
$\lambda\ is\ the\ decay\ constant\ (^{0.693}/_{half-life}\ (year^{-1})where\ 0.693 = \ln 2$
$N(t) is\ the\ size\ of\ a\ population\ of\ radioactive\ atoms\ at\ a\ given\ time\ t$
$N_0\ is\ the\ size\ of\ a\ population\ of\ radioactive\ atoms\ at\ time\ t = 0$

While this information is accurate, the PRG, and thus the decay curve, needs be solved for every chain member that may be present now or in the future. The SE PRG equation conveniently includes the fractional contribution of all progeny and assumes that the parent is continually renewed. This effectively creates a steady state ELCR rate over time, which is unrealistic for most Superfund sites.

For the situation where a non-renewing radionuclide is released into environmental media, a new PRG model is needed for the calculators that will eliminate continual regeneration of the parent, account for progeny ingrowth, and be protective throughout the decay cycle. The ELCR rate for many long-lived parents increases over time, as progeny with greater ELCR potential grow in. The intention is to calculate and plot the ELCR of each member of the chain for the entire decay cycle and determine the total ELCR during an exposure duration (ED) by using the time-points in the unit activity file (UAF) to determine the maximum risk ED interval.

The first step is to determine the decay curves. For any nuclide, the ingrowth-decay equation can be solved by integrating equation 5 with respect to t.

$$\frac{dN_2}{dt} = \lambda_1 N_1 - \lambda_2 N_2 \qquad \text{Equation 5}$$

The equations include both ingrowth and decay. Looking at the base equation, $\lambda_1 N_1$ represents the ingrowth of the daughter, and $\lambda_2 N_2$ represents the decay of the daughter product. For the first daughter product, where there is no ingrowth of the parent, the integration resolves to equation 6.

$$N_2 = (N_1)_0 \frac{\lambda_{1A}}{\lambda_2 - \lambda_1}\left(e^{-\lambda_1 t} - e^{-\lambda_2 t}\right) \qquad \text{Equation 6}$$

For the second daughter product, the base equation is rearranged, as in equation 7.

$$\frac{dN_3}{dt} = \lambda_2 N_2 - \lambda_3 N_3 \qquad \text{Equation 7}$$

Including the ingrowth of the first daughter product (i.e., N_2 is a function of time) and integration over time yields equation 8.

$$N_3 = \lambda_1 \lambda_2 (N_1)_0 \left(\frac{e^{-\lambda_1 t}}{(\lambda_2 - \lambda_1)(\lambda_3 - \lambda_2)} + \frac{e^{-\lambda_2 t}}{(\lambda_1 - \lambda_2)(\lambda_3 - \lambda_2)} + \frac{e^{-\lambda_3 t}}{(\lambda_1 - \lambda_3)(\lambda_2 - \lambda_3)}\right) \qquad \text{Equation 8}$$

This solution method is complicated for several reasons. Accounting for chain branching and rejoining is computationally intense and is addressed by using partial decay constants for the ingrowth in the above equations. However, as the chains get longer, these equations become unwieldly and progressively more difficult to solve. Finally, one must be very careful when dealing with these equations because it is very easy for numerical noise to creep into the result, giving non-positive results. A unique C++/g++ activity solver was written to address these issues and numerically approximate these equations utilizing a Quadmath library to increase accuracy. This solver generated the entire suite of UAF for various decay chain and PRG tools. Appendix A presents the code used to approximate the decay cycle. In addition, a python solution was developed using the Chebyshev Rational Approximation Method (CRAM) (Pusa 2016), which can be run dynamically within the calculator (section 6).

The peak PRG will account for decay of the parent and ingrowth of all progeny and determine the future time period when the peak ELCR occurs. One of the EPs discussed previously is the ED. Each land use scenario for a PRG has a default ED that will be set equal to the peak ELCR period. The ED variable (and many others) may be changed by the calculator user. This level of user control over EPs and IPs requires new computationally advanced procedures to be implemented in addition to the traditional algebraic PRG calculations as follows.

- The peak PRG requires the fractional decay of every member of the chain at every point in time. This can be considered the ingrowth-decay equation curve. The fractional decay needs to be addressed by use of a Bateman equation solver (Conner 2019).
- A time series of these relative fractional activities needs to be generated in a UAF.
- An ELCR component, relative to the land use scenario selected by the user using the EPs and IPs, needs to be applied to the UAF.
- The concentration of the parent radionuclide to be protective while accounting for progeny ingrowth needs to be calculated.

5. THE BATEMAN SOLVER

Radionuclides decay and may produce a linear or branched chain of progeny. The decay and ingrowth of radionuclides can be accurately modeled with a system of linear, first order differential equations. Bateman's early work in this area provides an analytical solution to a non-branching chain of radionuclides. In this approach, branching chains of radionuclides can be decomposed into a set of linear chains using a mathematical technique called the linear chain method. While the Bateman solution is valid for most chains, there are degenerate cases that do not produce accurate results. These degeneracies occur when two radionuclides in the chain have identical, or close to identical, rate coefficients. In all cases, however, slight modifications to the rate coefficient of progeny removes any degeneracy in the calculations while maintaining accurate results.

The solved radionuclide decay chains in the peak PRGs were obtained from a pre-existing database created for dosimetry. The ICRP Publication 107 (Eckerman and Endo 2007; ICRP 2007) database contains the half-lives, decay chains, and branching fractions used in the present work. This database supersedes the data of Publication 38 (Endo et al. 2005) and provides nuclear decay data for 1252 radionuclides.

The method to determine the activity for each chain member of a non-branching chain was obtained from the Bateman solutions. The activity of any arbitrary chain member is a function of the initial parent activity and the decay constants of the specified chain member and all its ancestor radionuclides. Equation 9 presents the traditional form of the solution.

$$N_n(t) = \frac{N_1(0)}{\lambda_n} \sum_{i=1}^{n} \lambda_i \alpha_i e^{(-\lambda_i t)} \qquad \text{Equation 9}$$

$where:$

$$\alpha_i = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{\lambda_j}{(\lambda_j - \lambda_i)}$$

λ_i $is\ the\ decay\ chain\ constant\ for\ the\ \ i^{th}\ radionculide\ in\ the\ decay\ chain$
$N_n(t)\ is\ the\ mass\ concentration\ of\ the\ n^{th}\ radionuclide\ in\ the\ decay\ chain$
$\lambda_n N_n(t)\ is\ the\ activity\ of\ the\ n^{th}\ radionuclide\ in\ the\ decay\ chain$

Equation 9 is only valid for non-branching chains. Two modifications are necessary to correctly estimate the activity of chain members of branched chains. The first modification is that each branched chain needs to be decomposed into its component linear sub-chains. This modification is necessary because while the Bateman solutions do not consider branching, if all sub-chains that contain a particular chain member are analyzed, then the actual activity of the chain member is equal to the sum of contributions from all possible sub-chains. A generalized procedure for decomposing a branched radionuclide chain can be obtained from standard tree traversal algorithms. The second modification is that the solution needs to include a multiplicative correction factor, which considers the product of all previous branching ratios in the sub-chain. Further details on the branching factor have been discussed by Cetnar (2006).

The linearization of the chain is accomplished by constructing a complete series of chains where each radionuclide has zero or one parent radionuclide. Therefore, each row of the f matrix in equation 10, below, has only a single positive term. Such a system of differential equations has a straightforward analytic solution. In some rare cases, the rate constants of chain members may be equal, which causes degeneracies in the solution due to a potential division by zero. In these cases, adding an arbitrarily small delta to the rate constant of the later chain member resolves this difficulty.

$$\begin{bmatrix} A' \\ B' \\ C' \\ \vdots \end{bmatrix} = \begin{bmatrix} -f_{AA} & f_{BA} & f_{CA} & \cdots \\ f_{AB} & -f_{BB} & f_{CB} & \cdots \\ f_{AC} & f_{BC} & -f_{CC} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} -\lambda_A & 0 & 0 & \cdots \\ 0 & -\lambda_B & 0 & \cdots \\ 0 & 0 & -\lambda_C & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} A \\ B \\ C \\ \vdots \end{bmatrix} \qquad \text{Equation 10}$$

6. MATHEMATICS OF CRAM

The Chebyshev Rational Approximation Method (CRAM) was used to develop additional UAFs. The objective of the chain decay solver is to determine the time-course of the decay products for a unit of a radionuclide. The underlying mathematical problem is to solve a system of ordinary differential equations in which each variable, $x_i(t)$, represents the amount of a particular radionuclide and its decay products at time t, with $X(t)$ being the vector of radionuclide amounts at time t, and $X(0)$ giving the initial concentrations or amounts of the radionuclide and its products. For this initial value differential equation problem, the associated matrix equation (called the burnup equation) is linear, with $X'(t) = AX(t)$, where A is the burnup matrix giving the decay and transmutation coefficients of the radionuclide and its decay products.

An underlying mathematical and computational challenge arises, because the decay and transmutation rates associated with various decay products from any particular radionuclide typically vary tremendously in magnitude, with some products decaying much more rapidly than others. Effectively, this means in the system of differential equations that some variables are changing on time scales much more rapidly than others. Although the solution to the linear differential equation system is simply $X(t) = e^{At}$, the eigenvalues of A are of greatly different magnitudes, and a numerical solution method has difficulty, since one component has rapid changes over short time intervals while others have extremely small changes over this time interval. The underlying system is called "stiff", and there are a variety of numerical methods that have been developed for these systems, since standard methods are numerically unstable (e.g., the numerical solutions do not converge to the true solution).

Alternative to using numerical methods designed specifically for stiff systems to solve the underlying linear differential equations, a solution can be directly computed by finding the matrix exponential e^{At}. A property of the burn equations that is helpful here is that, in general, the eigenvalues of A are close to the negative real axis. This has led to the development of the Chebyshev Rational Approximation Method (CRAM), in which the exponential function e^x is approximated on the negative real axis by the best function that is a ratio of polynomials $r_{k,k}(x) = p_k(x)/q_k(x)$, where $p_k(x)$ and $q^k(x)$ are both kth order polynomials.

7. GENERATION OF THE UNIT ACTIVITY FILE

After the Bateman solver calculates activities across time for the decay cycle, static UAFs are generated for all 1252 radionuclides in the ICRP 107 database. These UAFs are used to create plots in PRG and risk calculators and provide the following benefits: a) faster access for the EPA web-based calculators, b) same file is applicable for multiple calculators, and c) less compile time across server. Perl is used to create a run batch via access to the database table containing parent, progeny, half-life, branching fractions, and lambda for every parent. Assembled elements are then sent to the C++ Bateman solver by parent.

The same Perl batch also sends the information required for CRAM (Pusa 2016). These CRAM activity files are run at the same time, so the result is two sets of activity files that can then be compared for verification. In addition, the CRAM method can be used dynamically to insert additional time points, where necessary, to account for the ELCR rates in ED. Appendices B and D present the CRAM code.

Using the CRAM UAF, Figure 1 presents the state of the chain for Ra-226 over time for all progeny, assuming a starting activity of 1 pCi, from the Risk Assessment Information System (RAIS) decay chain tool (RAIS 2020).

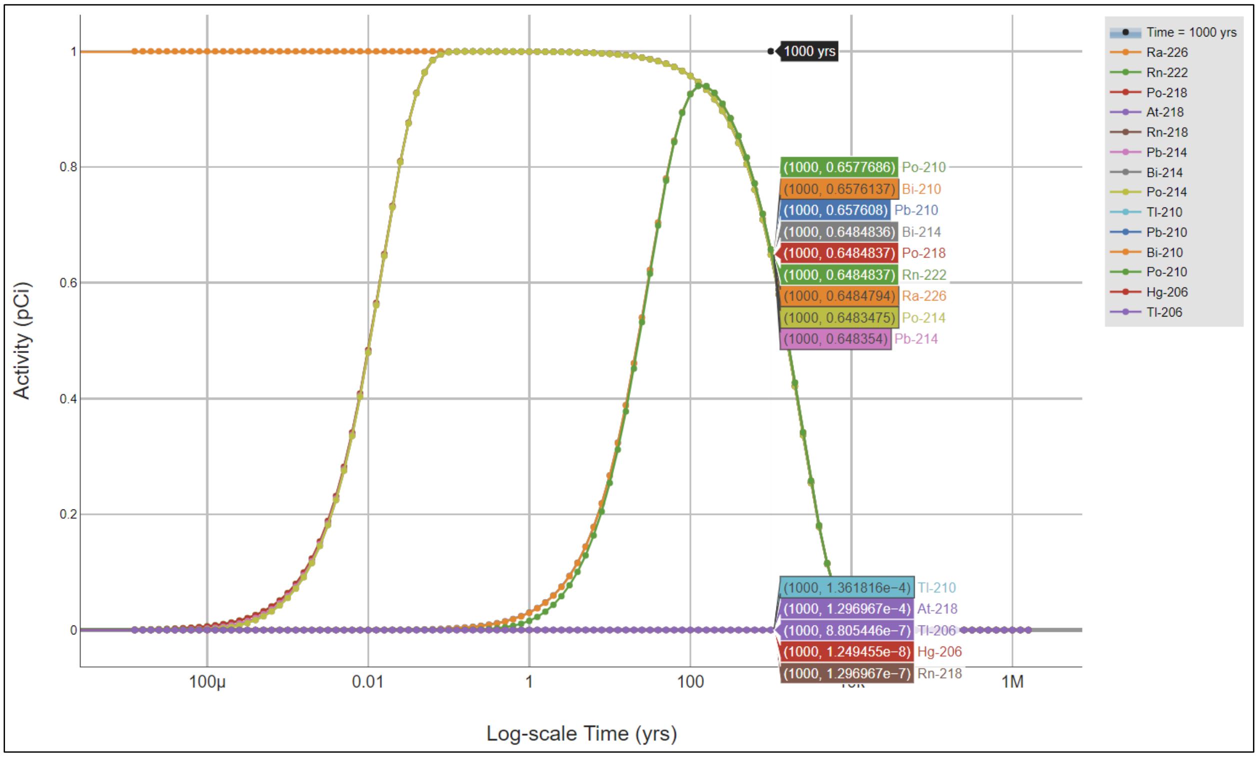

Figure 1. Ra-226 decay chain for all progeny from RAIS decay chain tool.

As confirmatory evidence for several chains at several time points, the following tables have been created with an additional reference, the World Information Service on Energy Uranium Project (Wise 2018) universal decay calculator, for validation. Figure 2 refers to Ra-226 at 1000 years for A_i=1, Figure 3 refers to Am-241 at 100 years for A_i=1, and Figure 4 refers to Th-228 at 10 years for A_i=1.

Activities for Ra-226 and daughters where initial activity = 1

Time	Isotope	CRAM Calculated Activity	Bateman C++ Calculated Activity	Wise-Uranium Calculated Activity*
1000	Ra-226	6.4848e-01	6.4842e-01	6.4840e-01
1000	Rn-222	6.4848e-01	6.4842e-01	6.4840e-01
1000	Po-218	6.4848e-01	6.4842e-01	6.4840e-01
1000	At-218	1.2970e-04	1.2968e-04	1.2960e-04
1000	Rn-218	1.2970e-07	1.2968e-07	
1000	Pb-214	6.4835e-01	6.4829e-01	6.4830e-01
1000	Bi-214	6.4848e-01	6.4842e-01	6.4840e-01
1000	Po-214	6.4835e-01	6.4829e-01	6.4830e-01
1000	Tl-210	1.3618e-04	1.3617e-04	
1000	Pb-210	6.5761e-01	6.5755e-01	6.5740e-01
1000	Bi-210	6.5761e-01	6.5755e-01	6.5740e-01
1000	Po-210	6.5777e-01	6.5771e-01	6.5760e-01
1000	Hg-206	1.2495e-08	1.2493e-08	
1000	Tl-206	8.8054e-07	8.8046e-07	

* Some progeny were not reported in this solver.

Figure 2. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Ra-226 at 1000 years with initial activity of 1 (Bq, pCi, etc.).

Activities for Am-241 and daughters where initial activity = 1

Time	Isotope	CRAM Calculated Activity	Bateman C++ Calculated Activity	Wise-Uranium Calculated Activity*
100	Am-241	8.5185e-01	8.5182e-01	8.5180e-01
100	Np-237	2.9864e-05	2.9870e-05	2.9920e-05
100	Pa-233	2.9835e-05	2.9841e-05	2.9890e-05
100	U-233	6.6588e-09	6.6615e-09	6.7030e-09
100	Th-229	2.1161e-11	2.1174e-11	2.1300e-11
100	Ra-225	2.1124e-11	2.1137e-11	2.1270e-11
100	Ac-225	2.1099e-11	2.1112e-11	2.1240e-11
100	Fr-221	2.1099e-11	2.1112e-11	2.1240e-11
100	At-217	2.1099e-11	2.1112e-11	2.1240e-11
100	Bi-213	2.1096e-11	2.1110e-11	2.1240e-11
100	Po-213	2.0656e-11	2.0669e-11	2.0780e-11
100	Tl-209	4.4092e-13	4.4120e-13	4.5890e-13
100	Pb-209	2.1096e-11	2.1110e-11	2.1240e-11

Figure 3. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Am-241 at 100 years with initial activity of 1 (Bq, pCi, etc.).

Activities for Th-228 and daughters where initial activity = 1

Time	Isotope	CRAM Calculated Activity	Bateman C++ Calculated Activity	Wise-Uranium Calculated Activity*
10	Th-228	2.6643e-02	2.6622e-02	2.6690e-02
10	Ra-224	2.6783e-02	2.6763e-02	2.6840e-02
10	Rn-220	2.6783e-02	2.6763e-02	2.6840e-02
10	Po-216	2.6783e-02	2.6763e-02	2.6840e-02
10	Pb-212	2.6800e-02	2.6780e-02	2.6850e-02
10	Bi-212	2.6802e-02	2.6781e-02	2.6850e-02
10	Po-212	1.7169e-02	1.7156e-02	1.7200e-02
10	Tl-208	9.6327e-03	9.6252e-03	9.6500e-03

Figure 4. Comparison of CRAM, Bateman, and Wise-Uranium calculated activities for Th-228 at 10 years with initial activity of 1 (Bq, pCi, etc.).

8. CONVERSION OF THE UAF TO ELCR RATES

As the UAF is a time-series of activity points per year, ELCR rates have been calculated in units of ELCR/year by route. For example, the ELCR rate for soil ingestion would be the denominator of a generic ingestion PRG multiplied by activity (A) at time t_i / ED. Equation 11 presents the modification of the PRG equation to yield an ELCR rate. PRG equations for other exposure routes would be similarly modified.

$$Risk\ Rate\left(\frac{risk}{yr}\right) = A_t\left(\frac{pCi}{g}\right) \times \left(\frac{\left(SF\left(\frac{risk}{pCi}\right) \times EF\left(\frac{d}{yr}\right) \times ED(yr) \times IRS\left(\frac{mg}{d}\right) \times \left(\frac{g}{1000\ mg}\right)\right)}{ED(yr)}\right) \quad \text{Equation 11}$$

$where:$
$A_t\ is\ the\ activity\ at\ time\ t$
$SF\ is\ the\ slope\ factor$
$EF\ is\ the\ exposure\ frequency$
$ED\ is\ the\ exposure\ duration$
$IRS\ is\ the\ soil\ ingestion\ rate$

ELCR rates can be calculated for every route by yearly increments and then summed for a total ELCR/yr at time T. The totals at every timepoint are evaluated in the solver to determine the maximum ELCR across ED. Appendix E presents the code used to solve this process. This maximum is the “Peak ELCR Interval”.

Figures 5-8 present the relative ELCR over time for the composite worker, soil exposure to Ra-226, for each exposure route, after applying equation 11 to the UAF curves presented in Figure 1. Note that the individual exposure route peak ELCRs are at different times. The progeny contributing to the peak ELCR change between exposure routes, as not all radionuclides have all types of IPs.

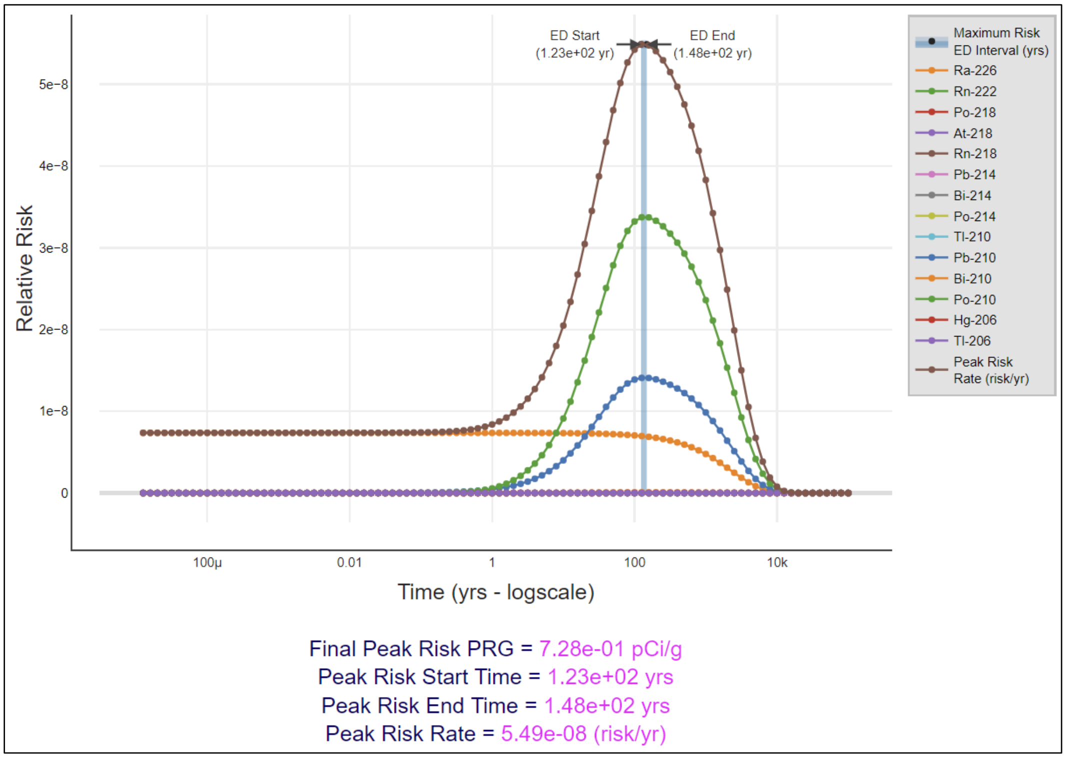

Figure 5. Ra-226 relative risk for the soil ingestion route for the composite worker, 25-year duration.

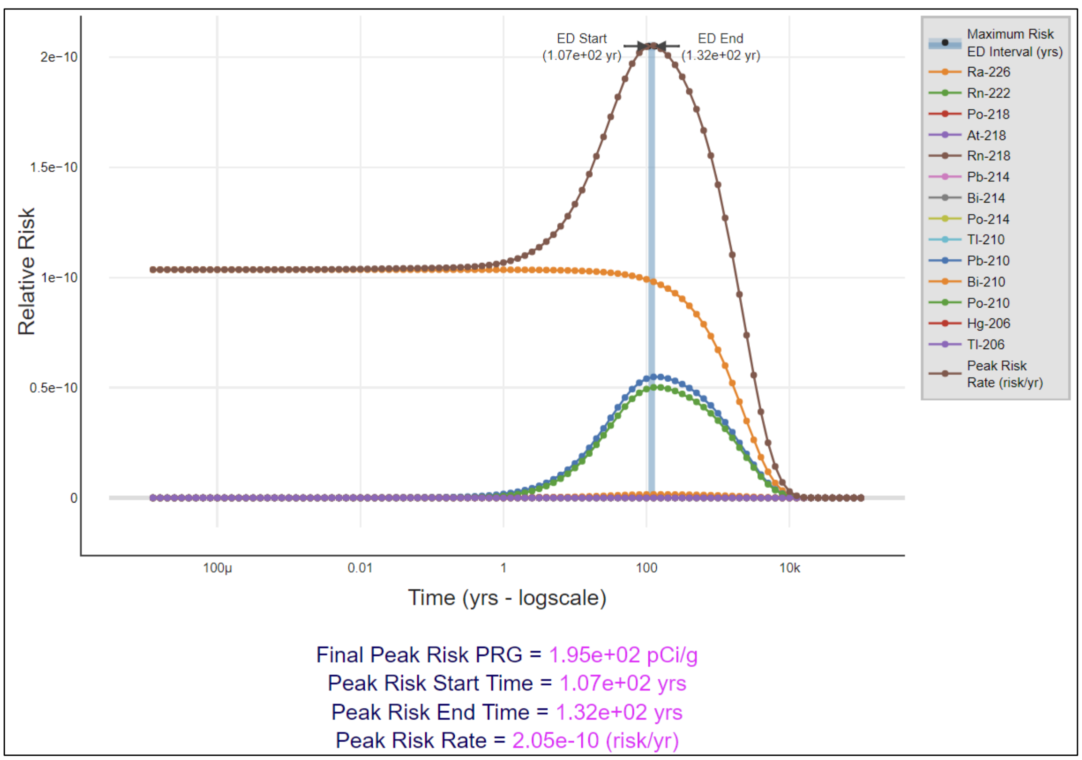

Figure 6. Ra-226 relative risk for the soil inhalation route for the composite worker, 25-year duration.

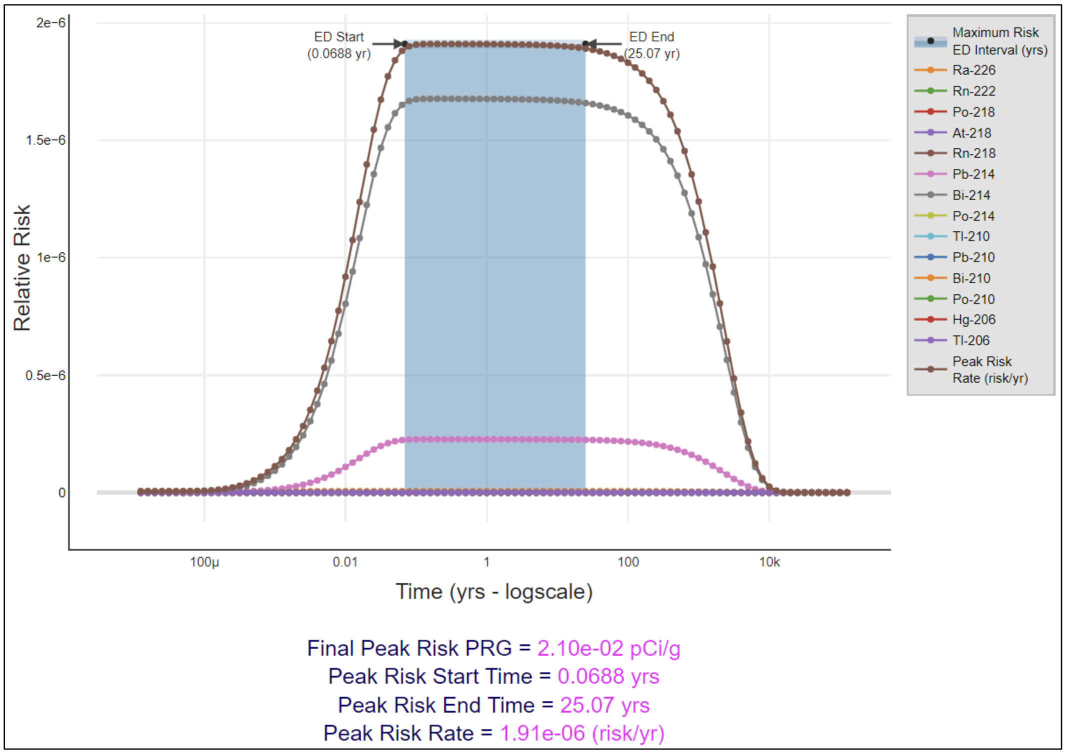

Figure 7. Ra-226 relative risk for the soil external exposure route for the composite worker, 25-year duration.

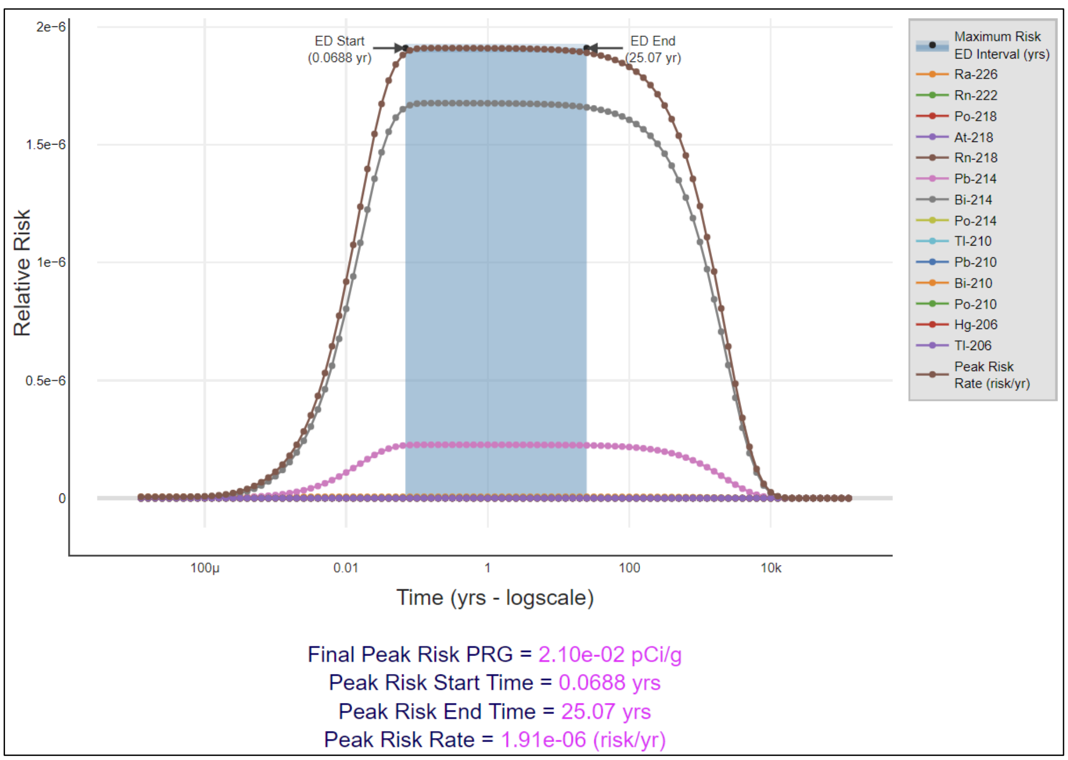

Figure 8. Ra-226 relative risk for total soil exposure for the composite worker, 25-year duration.

9. HOW PERIOD OF PEAK ELCR IS FOUND

The ELCR from exposure to a radionuclide depends on several factors, such as the IPs (e.g., risk coefficients), the concentration of the radionuclide in the environment, and the EPs. The ELCR associated with prolonged exposure (during ED) to a single radionuclide can be solved by equation 12.

$$\overline{AUC} = \int_{\mathrm{t}_{pst}}^{\mathrm{t}_{pet}} ELCR(\mathrm{t})\mathrm{dt} \qquad \text{Equation 12}$$

where:
$\overline{AUC}$ *is the radiogenic cancer total ELCR (area under the curve) associated with living in a contaminated environment during the peak time interval*
t_{pst} *is the peak start time*
t_{pet} *is the peak end time (pst + ED)*
$ELCR(t)$ *is the risk rate at time t*

Given a particular radionuclide, exposure duration, and initial radionuclide concentration, the ELCR depends on initial time of exposure, t_{pst}. For a single radionuclide, the largest ELCR will always occur when $\mathrm{t}_1 = 0$. For a radionuclide decay chain where only the parent is initially present; however, the largest ELCR could occur at future times. This is likely to occur if a progeny has a large risk coefficient and can be present at large activities at later times during the radionuclide chain's decay and ingrowth. For a radionuclide chain, the period of peak ELCR starts at the value of t_{pst}, when ELCR of exposure to all radionuclides in the chain is maximized (Equation 13).

$$Start\ of\ peak\ risk\ period = \max_{t_i}\left\{\sum_i \int_{t_{pst}}^{t_{pst+ED}} ELCR(t)dt\right\} \qquad \text{Equation 13}$$

where:
ED *is the exposure duration*
i *is the index for each member of the radionuclide chain*

The period of peak ELCR is found using numerical methods with peak search algorithms. The ELCR rate as a function of time is calculated for each chain member and summed to determine the total ELCR rate function for the chain. This step employs the activities estimated using the Bateman solver (see section 5) as well as scenario-dependent, time-invariant, radionuclide-specific risk coefficients. Appendix F presents the Python code used to solve for the peak ELCR period.

10. HOW A PEAK PRG IS CALCULATED

Equation 14 presents the fractional summation of individual PRGs in a chain at the peak across all exposure routes for the selected media and land use.

$$PRG_{peak} = \frac{1}{\frac{F_A}{PRG_A} + \frac{F_B}{PRG_B} + \frac{F_C}{PRG_C} + \cdots + \frac{F_n}{PRG_n}} \qquad \text{Equation 14}$$

where:
n *is the total number of radionuclides in the decay chain*

The average fractional activity is defined, as follows, in equation 15.

$$F_i = \frac{\bar{a}_i}{A_0} \quad \text{Equation 15}$$

where:
F_i *is the average fractional activity of the* i^{th} *chain*
$\bar{a}_i$ *is the average activity of the* i^{th} *chain member during the scenario exposure duration*
A_0 *is the initial activity of the parent at disposal or release*

After determining the peak start time, the Peak PRG is then calculated by normalizing the ELCR rates over the peak interval to the target ELCR (TR). Peak PRGs are calculated per route and for the total of all routes.

11. EXAMPLE CALCULATION OF A PEAK PRG FOR RA-226

The EPA PRG calculator offers many land use scenarios, but for simplicity sake, this example calculation is based on the composite worker exposed to soil. This land use was developed with the most protective aspects of the indoor and outdoor worker and is the default industrial/commercial screening land use. This receptor has three exposure routes: ingestion, inhalation, and external exposure. To calculate the Peak PRG, first retrieve the PRGs without decay for each member of the chain. Figure 9 presents the RAIS peak PRG calculator output of the IPs and PRGs for each member of the Ra-226 chain. The PRGs are unaltered and not corrected for decay or peak ELCR period.

Isotope	ICRP Lung Absorption Type	Inhalation Slope Factor (risk/pCi)	External Exposure Slope Factor (risk/yr per pCi/g)	Adult Soil Ingestion Slope Factor (risk/pCi)	Lambda (1/yr)	Halflife (yr)	Default Soil Volume Area Correction Factor	Default Soil Volume Gamma Shielding Factor	Particulate Emission Factor (m³/kg)	Ingestion PRG TR=1.0E-6 (pCi/g)	Inhalation PRG TR=1.0E-6 (pCi/g)	External Exposure PRG TR=1.0E-6 (pCi/g)	Total PRG TR=1.0E-6 (pCi/g)
Ra-226	S	2.82E-08	2.50E-08	2.95E-10	4.33E-04	1.60E+03	1.00E+00	1.00E+00	1.36E+09	5.43E+00	3.86E+02	7.01E+00	**3.04E+00**
Rn-222	-	2.28E-12	1.69E-09	0.00E+00	6.62E+01	1.05E-02	1.00E+00	1.00E+00	1.36E+09	-	4.77E+06	1.03E+02	**1.03E+02**
Po-218	-	1.39E-11	6.84E-15	0.00E+00	1.17E+05	5.90E-06	9.00E-01	1.00E+00	1.36E+09	-	7.82E+05	2.85E+07	**7.61E+05**
At-218	-	0.00E+00	2.74E-11	0.00E+00	1.46E+07	4.76E-08	9.00E-01	1.00E+00	1.36E+09	-	-	7.09E+03	**7.09E+03**
Rn-218	-	0.00E+00	3.39E-09	0.00E+00	6.24E+08	1.11E-09	1.00E+00	1.00E+00	1.36E+09	-	-	5.17E+01	**5.17E+01**
Pb-214	S	7.77E-11	9.94E-07	2.21E-13	1.36E+04	5.10E-05	1.00E+00	1.00E+00	1.36E+09	7.26E+03	1.40E+05	1.76E-01	**1.76E-01**
Bi-214	S	6.18E-11	7.34E-06	1.47E-13	1.83E+04	3.79E-05	1.00E+00	1.00E+00	1.36E+09	1.09E+04	1.76E+05	2.39E-02	**2.39E-02**
Po-214	-	0.00E+00	3.85E-10	0.00E+00	1.33E+11	5.21E-12	1.00E+00	1.00E+00	1.36E+09	-	-	4.55E+02	**4.55E+02**
Tl-210	-	0.00E+00	1.34E-05	0.00E+00	2.80E+05	2.47E-06	1.00E+00	1.00E+00	1.36E+09	-	-	1.30E-02	**1.30E-02**
Pb-210	S	1.59E-08	1.48E-09	5.99E-10	3.12E-02	2.22E+01	1.00E+00	1.00E+00	1.36E+09	2.67E+00	6.85E+02	1.18E+02	**2.60E+00**
Bi-210	S	4.55E-10	2.77E-09	3.74E-12	5.05E+01	1.37E-02	1.00E+00	1.00E+00	1.36E+09	4.28E+02	2.39E+04	6.33E+01	**5.50E+01**
Po-210	S	1.45E-08	4.51E-11	1.44E-09	1.83E+00	3.79E-01	1.00E+00	1.00E+00	1.36E+09	1.11E+00	7.50E+02	3.89E+03	**1.11E+00**
Hg-206	-	0.00E+00	4.83E-07	0.00E+00	4.47E+04	1.55E-05	1.00E+00	1.00E+00	1.36E+09	-	-	3.62E-01	**3.62E-01**
Tl-206	-	0.00E+00	6.11E-09	0.00E+00	8.67E+04	7.99E-06	1.00E+00	1.00E+00	1.36E+09	-	-	2.87E+01	**2.87E+01**

Figure 9. Peak PRG calculator output for Ra-226 chain without decay

Figure 10 shows the PRGs after they have been decayed, summed over period of peak ELCR, and all progeny ELCR contribution rolled into the parent, Ra-226. The table indicates that each exposure route has a different period of peak ELCR. The Ra-226 PRG without decay and progeny ingrowth, in Figure 9, is 3.04 pCi/g. Accounting for the period of peak ELCR, Figure 10 provides a PRG of 0.0208 pCi/g, beginning at 10.1 years from sample collection. Not accounting for the ingrowth of progeny would underestimate the actual ELCR.

Peak PRG Results	Ingestion PRG TR=1.0E-06 (pCi/g)	Inhalation PRG TR=1.0E-06 (pCi/g)	External Exposure PRG TR=1.0E-06 (pCi/g)	Total PRG TR=1.0E-06 (pCi/g)
Peak PRG for Ra-226 @ PRG units	*7.28E-01*	*1.95E+02*	*2.10E-02*	*2.08E-02*
Peak start time for maximum risk (yrs)	*1.23E+02*	*1.07E+02*	*6.88E-02*	*1.01E+01*
Maximum risk during peak interval (unitless)	*1.37E-06*	*5.13E-09*	*4.75E-05*	*4.81E-05*
Maximum risk-rate during peak interval (risk/yr)	*3.37E-08*	*1.04E-10*	*1.68E-06*	*1.68E-06*

Figure 10. Peak PRG calculator output for Ra-226 chain accounting for peak ELCR.

12. COMPARISON OF CURRENT AND PAST EPA PRG OPTIONS TO THE PEAK PRG

Figure 11 shows the EPA PRG calculator output for the three PRG models for a relatively long-lived (in comparison to the ED) radionuclide, Ra-226. Several land use scenarios are plotted, and all default EPs and IPs were used. This chart shows that the SE and Peak PRGs are very similar, and the PRG with decay is not protective. Additionally, this chart includes a discontinued PRG model (U.S. EPA 2001), where slope factors for the parent incorporate short-lived progeny up to 100 years (+D PRG). In the case of Ra-226, the progeny included are Rn-222, Po-218, Pb-214, At-218, Bi-214, Rn-218, Po-214, and Tl-210. Pb-210 is next in the chain, with a half-life of 22 years, so it and subsequent progeny are not included. As expected, the +D PRG is larger than the peak PRG and SE PRG, because it does not include the full chain. The resident scenario is not included in the comparison due to changes in the produce exposure route.

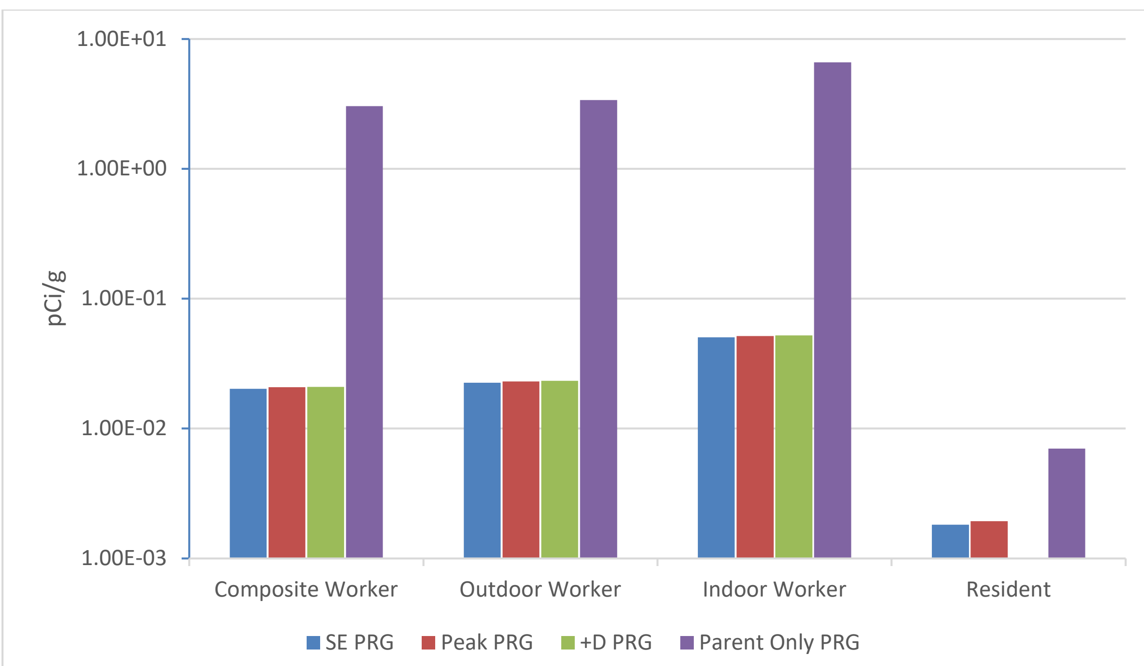

Figure 11. Comparison of current (and one discontinued) PRG calculator output options for Ra-226.

Figure 12 shows the EPA PRG calculator output for the three PRG models for a relatively short-lived (in comparison to the ED) radionuclide, Th-228. Several land use scenarios are plotted, and all default EPs and IPs were used. This chart shows that the SE and Peak PRGs are significantly different, and the SE PRG was overly protective. The +D PRG is not included, because Th-228 does not have a +D slope factor.

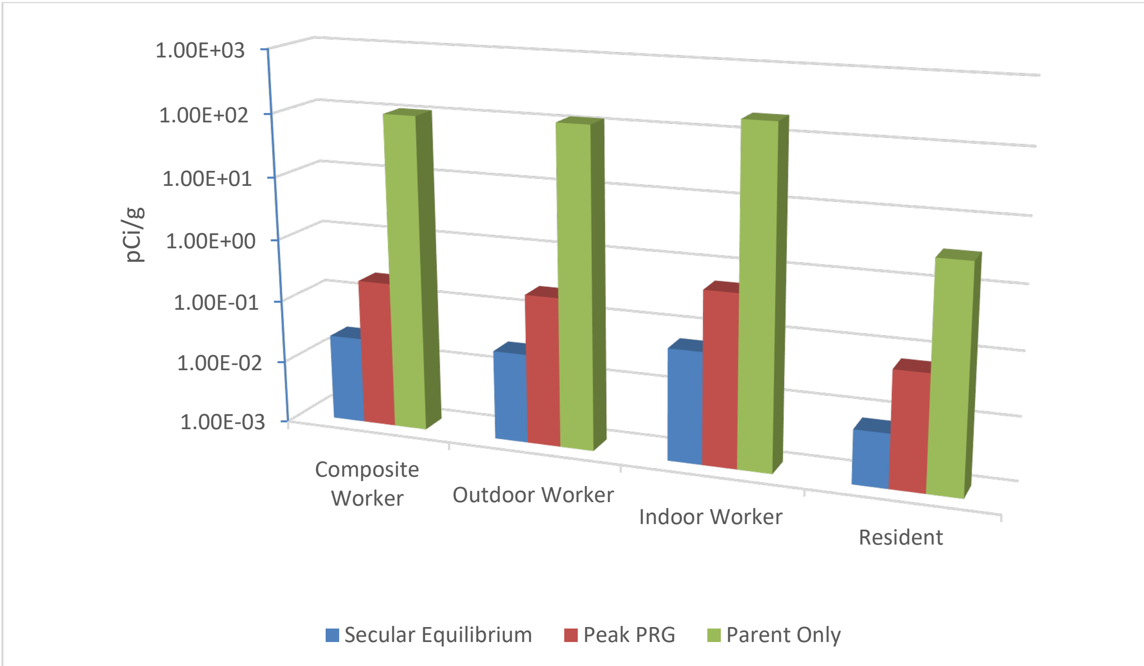

Figure 12. Comparison of current PRG calculator output options for Th-228.

13. PEAK ELCR CALCULATIONS USING UAF AND AUC

The peak PRG calculator allows the user to enter a site media concentration for the parent and calculate the period of peak ELCR for the complete chain. The calculator uses the methods previously described for UAF and area under the curve (AUC) to determine accurate ELCR over time. By utilizing the peak start time calculated from the Peak PRG calculation, the user-entered media activities are multiplied through the UAF matrix from peak start time to peak end time. If the timepoints do not currently exist in the time array, they are added using the CRAM routine described previously. Using Simpson's Rule for numerical integration (unequal intervals) with the points provided from the UAF, the AUC, or peak ELCR, is calculated for the ED interval of peak risk (methodology in python, Zaib 2019). Then, the ELCR rates are summed and totaled across the ED. For each chain member, ELCRs are summed in that interval to calculate a correct peak ELCR.

These ELCRs are calculated per route and for the total across all routes during the ED interval. Figure 13 presents the external exposure route ELCRs for the Ra-226 chain members across a 25-year ED. Ra-226 and the relatively short-lived progeny quickly achieve equilibrium, as evidenced by the flat curves. The relatively longer-lived progeny achieve equilibrium much later, as evidenced by the curves showing increased risk over time.

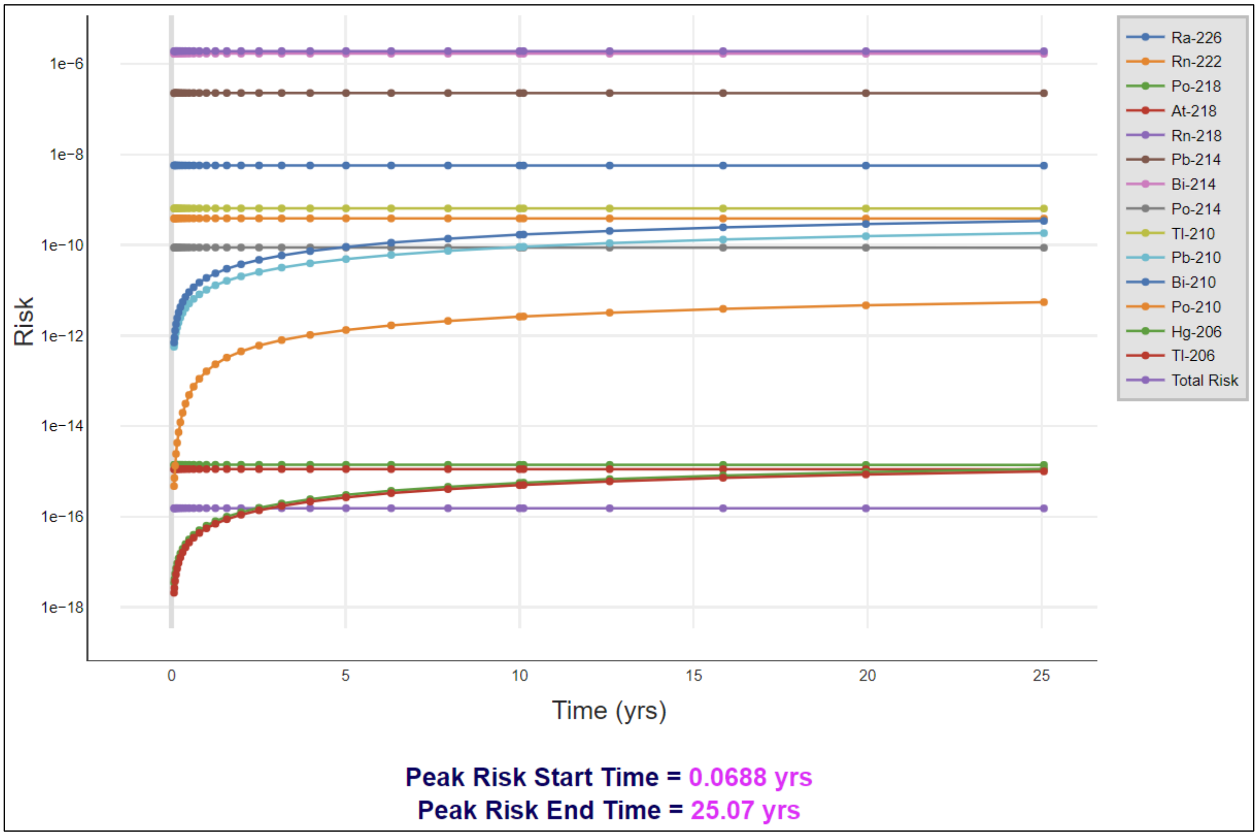

Figure 13. Ra-226 peak risk for composite worker soil external exposure over 25 years.

14. CONCLUSION

Creating a computational solution to the Bateman decay equations has always been challenging; however, two robust solutions have been adequately demonstrated in this work. Selecting the CRAM solver to create the UAFs and converting these plots from activity to a risk-rate has allowed for the creation of a new PRG model based on the period of peak risk. The CRAM solver runs dynamically on the web server and allows the recreation of UAFs with peak start and end times for the peak PRG tool.

The new peak PRGs developed in this report, combined with the two existing PRG models, allow all Superfund site conditions to be addressed more accurately than the current default SE PRGs and provide PRGs that are not overprotective. The peak PRGs also provide the added benefit of calculating when the period of peak ELCR begins and ends. This knowledge is important in making remedial decisions that are protective of future populations while reducing remediation costs by not remediating a site where the decay will reduce the ELCR naturally in a reasonable amount of time. Peak PRGs can also help design waste criteria and landfill caps protective of future ELCRs. Additionally, the techniques used to calculate the peak PRG can solve for peak ELCR and apply to dose-based models.

15. REFERENCES

Cetnar, J. 2008. General solution of Bateman equations for nuclear transmutations. Annals of Nuclear Energy. 33(7), 640-645.

Conner, N. 2019. What is Bateman Equation – Radioactive Decay – Definition. https://www.radiation-dosimetry.org/what-is-bateman-equation-radioactive-decay-definition/. December 14.

Eckerman, K. and Endo, A. 2007. ICRP Publication 107. Nuclear decay data for dosimetric calculations. Annals of the ICRP. 38(3), 7-96.

Endo, A., Yamaguchi, Y., & Eckerman, K. 2005. Nuclear decay data for dosimetry calculation. Revised data of ICRP Publication 38 (JAERI--1347). Japan.

ICRP. 2007. Nuclear decay data for dosimetric calculations. ICRP Publication 107. Annals of ICRP. 38(3).

ORNL (Oak Ridge National Laboratory). 2014a. Area Correction Factors for Contaminated Soil for Use in Risk and Dose Assessment Models and appendix. Center for Radiation Protection Knowledge. September.

ORNL. 2014b. Gamma Shielding Factors for Soil Covered Contamination for Use in Risk and Dose Assessment Models and appendix. Center for Radiation Protection Knowledge. September.

ORNL. 2014c. Calculation of Slope Factors and Dose Coefficients and appendix. Center for Radiation Protection Knowledge. September.

ORNL. 2016. Biota Modeling in EPA's Preliminary Remediation Goal and Dose Compliance Concentration Calculators for Use in EPA Superfund Risk Assessment: Explanation of Intake Rate Derivation, Transfer Factor Compilation, and Mass Loading Factor Sources Center for Radiation Protection Knowledge. November.

Pusa, M. 2016. Higher-Order Chebyshev Rational Approximation Method and Application to Burnup Equations, Nuclear Science and Engineering, 182:3, 297-318, DOI:10.13182/NSE15-26.

RAIS (Risk Assessment Information System). 2020. Radionuclide Decay Chain Tool. https://rais.ornl.gov/cgi-bin/chain/chain.pl.

U.S. EPA (Environmental Protection Agency). 1991. Risk Assessment Guidance for Superfund, Volume I: Human Health Evaluation Manual (Part B, Development of Risk-Based Preliminary Remediation Goals). Office of Emergency and Remedial Response. EPA/540/R-92/003. December.

U.S. EPA. 2001. HEAST Radionuclide Table: User's Guide. Office of Air and Radiation. https://www.epa.gov/radiation/radionuclide-table-radionuclide-carcinogenicity-slope-factors. April.

U.S. EPA. 2020a. Preliminary Remediation Goals for Radionuclide Contaminants at Superfund Sites (PRG). https://epa-prgs.ornl.gov/radionuclides/.

U.S. EPA. 2020b. Preliminary Remediation Goals for Radionuclides in Buildings at Superfund Sites (BPRG). https://epa-bprg.ornl.gov/.

U.S. EPA. 2020c. Preliminary Remediation Goals for Radionuclides on Outdoor Surfaces at Superfund Sites (SPRG). https://epa-sprg.ornl.gov/.

U.S. EPA. 2020d. Dose Compliance Concentrations for Radionuclides at Superfund Sites (DCC). https://epa-dccs.ornl.gov/.

U.S. EPA. 2020e. Dose Compliance Concentrations for Radionuclides in Buildings at Superfund Sites (BDCC). https://epa-bdcc.ornl.gov/.

U.S. EPA. 2020f. Dose Compliance Concentrations for Radionuclides on Outdoor Surfaces at Superfund Sites (SDCC). https://epa-sdcc.ornl.gov/.

Wise (World Information Service on Energy Uranium Project). Universal Decay Calculator. https://www.wise-uranium.org/rcc.html. October 2018.

Zaib, A. File Parsing and Data Analysis in Python Part II (area Under Curve and Engine Performance). https://skill-lync.com/ Post 82072. January 8, 2019.

APPENDIX A. BATEMAN SOLVER (C++ ODE METHOD)

APPENDIX A. BATEMAN SOLVER (C++ ODE METHOD)

```
#include <stdio.h>
#include <iostream>
#include <string>
#include <math.h>
#include <fstream>
#include <iomanip>
#include <vector>
#include <quadmath.h>
#include <algorithm>

using namespace std;

class chain {
	struct chainMember {
		int ID;
		double branchFromParent;
		__float128 decayRate;
	};

	double branchingFraction[100][100];
	__float128 decayRate[100];
	vector<vector<chainMember> > ListOfChains;

	public:
		chain(double bf[][100], __float128 dr[]) {
			for (size_t i = 0; i < 100; i++) {
				decayRate[i] = dr[i];
				for (size_t j = 0; j < 100; j++) {
					branchingFraction[i][j] = bf[i][j];
				};
			};
		}

		vector<vector<chainMember> > getSubChains(int endMemberID) {
			ListOfChains.clear();
			vector<chainMember> subChain;

			chainMember p;
			p.ID = 1;
			p.branchFromParent = 1;
			p.decayRate= decayRate[1];

			subChain.push_back(p);

			traverseChain(1, endMemberID, subChain);
```

```
        return ListOfChains;
}

void traverseChain(int current, int finalMember, vector<chainMember> subChain) {
        chainMember p;
        __float128 sum = 0;

        vector<chainMember> subChain2;

        for (size_t j = 0; j < 100; j++) {  //loop through all potential progeny
                if (branchingFraction[current][j] > 0) {
                        if (j == finalMember) {
                                // chain has ended with success. Append results and exit.
                                p.ID = j;
                                p.branchFromParent = branchingFraction[current][j];
                                p.decayRate = decayRate[j];
                                subChain2 = subChain;
                                subChain2.push_back(p);

                                ListOfChains.push_back(subChain2);
                        }
                        else {
                                //chain not ended, go deeper with recursive call, append
potential chain
                                p.ID = j;
                                p.branchFromParent = branchingFraction[current][j];
                                p.decayRate = decayRate[j];
                                subChain2 = subChain;
                                subChain2.push_back(p);

                                traverseChain(j, finalMember, subChain2);
                        }
                }
        }
}

__float128 solveActivity(int chainMember, long double time) {
        getSubChains(chainMember); //populates ListOfChains

        __float128 activity = 0;
        double initialA = 1;

        if (chainMember == 1) {
                activity = expq(-decayRate[1] * time);
        }
        else {
                for (size_t i = 0; i < ListOfChains.size(); i++) {
                        activity = activity + chainSolver(ListOfChains[i], time);
                }
        }
```

```
            return activity * initialA;
        }

        // solves activity of last element in chain at a time
        __float128 chainSolver(vector<chainMember> thisChain, long double time) {
            __float128 sum = 0;
            __float128 product = 1;
            __float128 innerProduct = 1;
            int currentElement = 0;
            long double t = time;

            __float128 dr[100];
            double br[100];

            for (size_t i = 0; i < thisChain.size(); i++) {
                br[i+1] = thisChain[i].branchFromParent;
                dr[i+1] = thisChain[i].decayRate;
            }

            currentElement = thisChain.size();

            product = 1;

            for (size_t i = 2; i <= currentElement; i++) {
                product = product * dr[i] * br[i];
            }

            sum = 0;

            for (size_t j = 1; j <= currentElement; j++) {
                innerProduct = 1;
                for (size_t k = 1; k <= currentElement; k++) {
                    //InnerProduct Loop
                    if (k != j) {
                        innerProduct = innerProduct * (dr[k] - dr[j]);
                    }
                }

                //Sum loop
                sum = sum + expq(-dr[j] * t) / innerProduct;
            }

            return (product * sum);
        }
};

int main() {
    ofstream outfile("/home/i65/activity_files/Ra-226.csv");
      string list = "Time,Ra-226,Rn-222,Po-218,At-218,Rn-218,Pb-214,Bi-214,Po-214,Tl-210,Pb-210,Bi-
210,Po-210,Hg-206,Tl-206" ;
      string alist = "0,1,0,0,0,0,0,0,0,0,0,0,0,0,0" ;
```

```
outfile << list << endl ;
size_t numElements = 14;

    char buf[128];
    int width = 20;

    double branchingFraction[100][100];
    __float128 decayRate[100];

    for (size_t i = 0; i < 100; i++) {
            decayRate[i] = 0;
            for (size_t j = 0; j < 100; j++) {
                    branchingFraction[i][j] = 0;
            };
    };
    decayRate[1] = 0.00043321698784999998 ;
    decayRate[2] = 66.21472674238778211020 ;
    decayRate[3] = 117602.48099609327618964016 ;
    decayRate[4] = 14582707.64994840323925018311 ;
    decayRate[5] = 624973169.78317034244537353516 ;
    decayRate[6] = 13603.27205548552774416748 ;
    decayRate[7] = 18319.98447672524343943223 ;
    decayRate[8] = 133133677888.74174499511718750000 ;
    decayRate[9] = 280436.68545112438732758164 ;
    decayRate[10] = 0.03122284597116870009 ;
    decayRate[11] = 50.50309349681236170682 ;
    decayRate[12] = 1.82959478305139389143 ;
    decayRate[13] = 44732.23203513757471228018 ;
    decayRate[14] = 86801.83121164762997068465 ;

    branchingFraction[1][2] = 1;
    branchingFraction[2][3] = 1;
    branchingFraction[3][6] = .9998;
    branchingFraction[3][4] = .0002;
    branchingFraction[4][7] = .999;
    branchingFraction[4][5] = .001;
    branchingFraction[5][8] = 1;
    branchingFraction[6][7] = 1;
    branchingFraction[7][9] = .00021;
    branchingFraction[7][8] = .99979;
    branchingFraction[8][10] = 1;
    branchingFraction[9][10] = 1;
    branchingFraction[10][13] = .000000019;
    branchingFraction[10][11] = .999999981;
    branchingFraction[11][12] = .99999868;
    branchingFraction[11][14] = .00000132;
    branchingFraction[13][14] = 1;



    chain C(branchingFraction, decayRate);
```

```
	long double time;
	int switcher[100];
	__float128 activity=0;

	for( int i = 0; i < 100; i++) {
		switcher[i] = 1;
	}

	//cout << list << endl ;
	for (int j = 100; j > -50; j--) {
		time = pow(10.0, j / 10.0);
		outfile << setprecision(20) << time ;

		for (size_t i = 1; i < numElements+1 ; i++) {
			activity = C.solveActivity(i,time);
			if (activity < 0) {
				switcher[i] = 0;
			}
			if (switcher[i] == 0) {
				activity = 0;
			}
			int n = quadmath_snprintf(buf, sizeof buf, "%#*.20Qe", width, activity);
			outfile << "," << setprecision(20) << buf ;
			//if (time == 1000) {
			//	cout << time << "," << setprecision(20) << buf << endl ;
			//}
		}
		outfile << endl;
	}

    outfile << alist << endl ;
	outfile.close();
}
```

APPENDIX B. CRAM MATRIX (PER PARENT AND CHAIN MEMBER)

APPENDIX B. CRAM MATRIX (PER PARENT AND CHAIN METHOD)

```
import numpy as np
import cx_Oracle
from sys import argv

if __name__ == "__main__":
    # Connect to database, or read in file
    connection = cx_Oracle.connect(dbuser,dbpwd,db)
    cursor = connection.cursor()
    N = len(argv)-1
    for i in range(1,N+1):
        parent = argv[i]

        # Select relevant items per parent in query
        query = f"""select * from
            (select distinct parent, child, lambda, fraction, c.mass
            from your_istotope_table
            where child <> 'SF'
            start with (parent='{parent}' and child not like '%SF%')
            connect by prior child_id=parent_id)
            order by mass desc"""

        # Execute and fetch
        cursor.execute(query)
        res = cursor.fetchall()

        indices = {}
        lambdas = []
        i = 0

        # Add isotopes to indices dictionary and lambdas list
        for isotope,child,lambda_,fraction,mass in res:
            if isotope not in indices.keys():
                indices[isotope] = i
                lambdas.append(lambda_)
                i += 1

        n = len(indices.keys())
        F = -np.eye(n) # Branching fraction matrix

        # Fill in branching fraction matrix
        for isotope,child,lambda_,fraction,mass in res:
            if child in indices.keys():
                F[indices[child],indices[isotope]] = fraction

        # Multiply branching matrix by decay constants to get activity-form Bateman matrix A
        A = (lambdas*(F.T)).T
```

```
# Save the matrix
np.save(f'{parent}_activity_matrix.npy',A)
with open(f'{parent}_labels.txt','w+') as f:
    f.write(str(list(indices.keys())))
```

APPENDIX C. CRAM BATEMAN SOLVER

APPENDIX C. CRAM BATEMAN SOLVER

```
import numpy as np
from numba import jit
from ast import literal_eval

# hardcoded path to file
path = '/opt/cram/cram/'

@jit(nopython=True,cache=True)
def cram16(A, y0, T):
    """ Chebyshev Rational Approximation Method, order 16
    Algorithm is the 16th order Chebyshev Rational Approximation Method,
    implemented in the more stable incomplete partial fraction (IPF) form
    [cram16]_.
    .. [cram16]
        Pusa, Maria. "Higher-Order Chebyshev Rational Approximation Method and
        Application to Burnup Equations." Nuclear Science and Engineering 182.3
        (2016).
    Parameters
    ----------
    A : ndarray of shape (N,N)
        Matrix to take exponential of.
    y0 : ndarray of shape (N,)
        Vector to operate a matrix exponential on.
    T : ndarray of shape (M,)
        Times to evaluate matrix exponential at.
    Returns
    -------
    ndarray of shape (M,N)
        Results of the matrix exponential.
    """

    # Incomplete partial fraction decomposition poles and residues
    theta = np.array([+3.509103608414918e+0 + 8.436198985884374e+0j,
                  +5.948152268951177e+0 + 3.587457362018322e+0j,
                  -5.264971343442647e+0 + 1.622022147316793e+1j,
                  +1.419375897185666e+0 + 1.092536348449672e+1j,
                  +6.416177699099435e+0 + 1.194122393370139e+0j,
                  +4.993174737717997e+0 + 5.996881713603942e+0j,
                  -1.413928462488886e+0 + 1.349772569889275e+1j,
                  -1.084391707869699e+1 + 1.927744616718165e+1j])

    alpha = np.array([+5.464930576870210e+3 - 3.797983575308356e+4j,
                  +9.045112476907548e+1 - 1.115537522430261e+3j,
                  +2.344818070467641e+2 - 4.228020157070496e+2j,
                  +9.453304067358312e+1 - 2.951294291446048e+2j,
                  +7.283792954673409e+2 - 1.205646080220011e+5j,
```

```
                  +3.648229059594851e+1 - 1.155509621409682e+2j,
                  +2.547321630156819e+1 - 2.639500283021502e+1j,
                  +2.394538338734709e+1 - 5.650522971778156e+0j])

    alpha0 = 2.124853710495224e-16

    # Set up output array
    m = A.shape[0]
    Y = np.empty((len(T),m))
    Y[0] = y0

    # Set up time delta array
    Dt = T[1:]-T[:-1]

    # Identity matrix
    I = np.eye(m)

    # Step forward in time
    for i,dt in enumerate(Dt):
        y = Y[i].copy() + 0*1j
        At = A*dt
        # Loop over poles
        for l in range(8):
            y = 2.0*np.real(alpha[l]*np.linalg.solve(At - theta[l]*I, y)) + y

        # Add to array
        Y[i+1] = alpha0*np.real(y)

    return Y

@jit(nopython=True,cache=True)
def cram48(A, y0, T):
    """ Chebyshev Rational Approximation Method, order 48
    Algorithm is the 48th order Chebyshev Rational Approximation Method,
    implemented in the more stable incomplete partial fraction (IPF) form
    [cram48]_.
    .. [cram48]
        Pusa, Maria. "Higher-Order Chebyshev Rational Approximation Method and
        Application to Burnup Equations." Nuclear Science and Engineering 182.3
        (2016).
    Parameters
    ----------
    A : ndarray of shape (N,N)
        Matrix to take exponential of.
    y0 : ndarray of shape (N,)
        Vector to operate a matrix exponential on.
    T : ndarray of shape (M,)
        Times to evaluate matrix exponential at.
    Returns
    -------
    ndarray of shape (M,N)
```

```
    Results of the matrix exponential.
"""

# Incomplete partial fraction decomposition poles and residues
theta_r = np.array([-4.465731934165702e+1, -5.284616241568964e+0,
                    -8.867715667624458e+0, +3.493013124279215e+0,
                    +1.564102508858634e+1, +1.742097597385893e+1,
                    -2.834466755180654e+1, +1.661569367939544e+1,
                    +8.011836167974721e+0, -2.056267541998229e+0,
                    +1.449208170441839e+1, +1.853807176907916e+1,
                    +9.932562704505182e+0, -2.244223871767187e+1,
                    +8.590014121680897e-1, -1.286192925744479e+1,
                    +1.164596909542055e+1, +1.806076684783089e+1,
                    +5.870672154659249e+0, -3.542938819659747e+1,
                    +1.901323489060250e+1, +1.885508331552577e+1,
                    -1.734689708174982e+1, +1.316284237125190e+1])
theta_i = np.array([+6.233225190695437e+1, +4.057499381311059e+1,
                    +4.325515754166724e+1, +3.281615453173585e+1,
                    +1.558061616372237e+1, +1.076629305714420e+1,
                    +5.492841024648724e+1, +1.316994930024688e+1,
                    +2.780232111309410e+1, +3.794824788914354e+1,
                    +1.799988210051809e+1, +5.974332563100539e+0,
                    +2.532823409972962e+1, +5.179633600312162e+1,
                    +3.536456194294350e+1, +4.600304902833652e+1,
                    +2.287153304140217e+1, +8.368200580099821e+0,
                    +3.029700159040121e+1, +5.834381701800013e+1,
                    +1.194282058271408e+0, +3.583428564427879e+0,
                    +4.883941101108207e+1, +2.042951874827759e+1])
theta = theta_r + theta_i * 1j

alpha_r = np.array([+6.387380733878774e+2, +1.909896179065730e+2,
                    +4.236195226571914e+2, +4.645770595258726e+2,
                    +7.765163276752433e+2, +1.907115136768522e+3,
                    +2.909892685603256e+3, +1.944772206620450e+2,
                    +1.382799786972332e+5, +5.628442079602433e+3,
                    +2.151681283794220e+2, +1.324720240514420e+3,
                    +1.617548476343347e+4, +1.112729040439685e+2,
                    +1.074624783191125e+2, +8.835727765158191e+1,
                    +9.354078136054179e+1, +9.418142823531573e+1,
                    +1.040012390717851e+2, +6.861882624343235e+1,
                    +8.766654491283722e+1, +1.056007619389650e+2,
                    +7.738987569039419e+1, +1.041366366475571e+2])
alpha_i = np.array([-6.743912502859256e+2, -3.973203432721332e+2,
                    -2.041233768918671e+3, -1.652917287299683e+3,
                    -1.783617639907328e+4, -5.887068595142284e+4,
                    -9.953255345514560e+3, -1.427131226068449e+3,
                    -3.256885197214938e+6, -2.924284515884309e+4,
                    -1.121774011188224e+3, -6.370088443140973e+4,
                    -1.008798413156542e+6, -8.837109731680418e+1,
                    -1.457246116408180e+2, -6.388286188419360e+1,
                    -2.195424319460237e+2, -6.719055740098035e+2,
```

```
                    -1.693747595553868e+2, -1.177598523430493e+1,
                    -4.596464999363902e+3, -1.738294585524067e+3,
                    -4.311715386228984e+1, -2.777743732451969e+2])
    alpha = alpha_r + alpha_i * 1j

    alpha0 = 2.258038182743983e-47

    # Set up output array
    m = A.shape[0]
    Y = np.empty((len(T),m))
    Y[0] = y0

    # Set up time delta array
    Dt = T[1:]-T[:-1]

    # Identity matrix
    I = np.eye(m)

    # Step forward in time
    for i,dt in enumerate(Dt):
        y = Y[i].copy() + 0*1j
        At = A*dt
        # Loop over poles
        for l in range(24):
            y = 2.0*np.real(alpha[l]*np.linalg.solve(At - theta[l]*I, y)) + y

        # Add to array
        Y[i+1] = alpha0*np.real(y)

    return Y

def cram(parent,initialA=1,order=16,n=1000,t0=1e-16,tf=None):
    """Calls CRAM of the correct order for a given parent and initial activity"""

    # load Bateman matrix
    A = np.load(path+f'matrices/{parent}_activity_matrix.npy')

    # build time array
    if tf is None:
        halflives = -np.log(2)/np.diag(A)
        tf = 7*np.sum(halflives)
    T = np.concatenate(([0],np.geomspace(t0,tf,n+1)))

    # initial activity
    a0 = np.zeros(A.shape[0])
    a0[0] = initialA

    # call CRAM
    if order == 16:
        activities = cram16(A,a0,T)
    elif order == 48:
```

```python
        activities = cram48(A,a0,T)

    return T,activities

def savecram(parent,outfile,initialA=1,order=16,n=1000,t0=1e-16,tf=None):
    """Runs CRAM and writes the activity data to outfile"""

    if outfile[-4:] != '.csv':
        raise ValueError('outfile must be .csv')

    # call CRAM
    T,activities = cram(parent,initialA,order,n,t0,tf)

    # load labels
    labels = ['Time']
    with open(path+f'labels/{parent}_labels.txt') as f:
        labels += literal_eval(f.readline())

    # set up data array
    header = ','.join(labels)
    data = np.insert(activities,0,T,axis=1)

    # save data to outfile as csv
    np.savetxt(outfile,data,delimiter=',',header=header,comments='')
```

APPENDIX D. CRAM BATEMAN OUTPUT

APPENDIX D. CRAM BATEMAN OUTPUT

```
import numpy as np
import pandas as pd
import csv
from sys import argv
from ast import literal_eval

# CRAM method has 16 or 48 precision
# use the method that is preferable
#from cram import cram16
from cram import cram48

if __name__ == "__main__":
    N = len(argv)-1

    for i in range(1,N+1):
        parent = argv[i]
        labels = ['Time']

        # Use the matrix file with the parent/child fractions, half-lives and lambdas
        A = np.load(f'{parent}_activity_matrix.npy')
        Afile = parent + '.csv' ;

        with open(f'{parent}_labels.txt') as f:
            labels += literal_eval(f.readline())

        header = ','.join(labels)

        # Unpack data from files
        activity_data = pd.read_csv(Afile)
        activity_data.set_index("Time", inplace=True)
        T = np.array(activity_data.index)

        a0 = np.zeros(A.shape[0])
        a0[0] = 1

        # Solve using 16 or 48
        #sol = cram16(A,a0,T)
        sol = cram48(A,a0,T)
        data = np.insert(sol,0,T,axis=1)

        np.savetxt(f'{parent}_cram.csv',data,delimiter=',',header=header,comments='')
```

APPENDIX E. PEAK PRG SOLVER

APPENDIX E. PEAK PRG SOLVER

```
# Peak PRG Solver
import numpy as np
import pandas as pd
from scipy import integrate
from scipy.interpolate import interp1d
from scipy.integrate import quad
from scipy.optimize import root_scalar
from sys import argv
import csv

if len(argv) < 2:
    # Test case for simple command line run
    # Resident Defaults for Soil
    # Am-241
    scenario = 'Resident'
    media = 'soil'
    parent = 'Am-241'
    numElements = int(13)
    exposure_duration = float(26)
    target_risk = float(1.0E-06)
    filenum = '22898'
    units = 'pCi'
    withrisk = 'no'

else:
    scenario = argv[1]
    media = argv[2]
    parent = argv[3]
    numElements = int(argv[4])
    exposure_duration = float(argv[5])
    target_risk = float(argv[6])
    filenum = argv[7]
    units = argv[8]
    withrisk = argv[9]
    chaindir = argv[10]
    actdir = argv[11]

nodash = parent.replace('-', '')

# Inputs
activity_file = actdir + parent + '_cram.csv'
prg_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_prgs.csv'
rate_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_rates.csv'

# Outputs
results_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_results.csv'
```

```
res = open(results_file, 'w')

# Rates
ratedata = pd.read_csv(rate_file, na_values='inf')
ratedata = ratedata.fillna(0)

# Turn rate data into numpy array
ratenp = ratedata.set_index("route", drop = True)
ratenp = ratenp.rename_axis('ID').values

# PRGs
prgdata = pd.read_csv(prg_file)

# Turn prg data into numpy array
prgnp = prgdata.set_index("route", drop = True)
prgnp = prgnp.rename_axis('ID').values

# Activities
actdata = pd.read_csv(activity_file,dtype='float')
actdata.drop(actdata[actdata.Time < 1e-16].index, inplace=True)
actdata.set_index("Time", inplace=True)

# Get Times and put in array
T = np.array(actdata.index).astype('float')
labels = list(actdata.columns.values.tolist())

# Decide which source of data to use - rates or PRGs
if media == 'comb':
    word = 'prg'
else:
    word = 'rate'

if word == 'prg':
    whichdata = prgdata
    whichnp = prgnp.astype('float')
else:
    whichdata = ratedata
    whichnp = ratenp.astype('float')

# Loop over routes starting here
# Each route has different toxicity and ELCR time range
for x in range(len(whichdata.index)):
    route = whichdata['route'][x]
    SOURCE = whichnp[x].astype('float')

    output_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + route + '_' + filenum +
'_output.csv' ;
    open(output_file, 'w')

    # Calculate risk sum and max
    if word == 'prg':
```

```
    SOURCE_actdata = (target_risk * actdata / SOURCE.astype('float')) / exposure_duration
else:
    SOURCE_actdata = actdata * SOURCE.astype('float')

SOURCE_risk_sum = np.array(SOURCE_actdata.sum(axis=1))
SOURCE_risk_max = np.array(SOURCE_actdata.max(axis=1))
SOURCE_actdata["SUM"] = SOURCE_risk_sum
SOURCE_actdata = SOURCE_actdata.loc[SOURCE_actdata["SUM"] > 0]
maxRate = SOURCE_risk_max.max()

# calculate when risk is zero
tol = 1e-30
idx = np.nonzero(SOURCE_risk_sum < tol)[0][0]

# write output file
SOURCE_actdata.iloc[:idx].to_csv(output_file)

# interpolate SOURCE_risk_sum data with a quadratic spline interpolation
f = interp1d(T,SOURCE_risk_sum,kind="quadratic")

# define F as above
F = lambda t: f(t+exposure_duration)-f(t)

# Use product F(t_i)F(t_i+1) (during duration of nonzero risk) to find root intervals
Ft = F(T[:idx])
indices = np.concatenate(np.where(0 > Ft[:-1]*Ft[1:]))
interval_start = np.array(T[indices])
interval_end = np.array(T[indices+1])

# Use rootfinding to get actual time of root
peak_times = [1.000000e-16] # smallest timepoint used in AUF runs
for t0,t1 in zip(interval_start,interval_end):
    peak_times.append(root_scalar(F,bracket=(t0,t1)).root)

# Evaluate to find maximum risk and interval
peak_risk = 0
for time in peak_times:
    risk = quad(f, time, time + exposure_duration)[0]
    if risk > peak_risk:
        peak_risk = risk
        peak_start = time

# Compute average activity
average_activities = []
for isotope in actdata:
    g = interp1d(T,actdata[isotope],kind="quadratic")
    average_activities.append(quad(g, peak_start, peak_start + exposure_duration)[0])
average_activities = np.array(average_activities) / exposure_duration

# Calculate final PRG and write important numbers to stdout
if word == 'prg':
```

```
        finalPRG = 1 / (average_activities / SOURCE.astype('float')).sum()
    else:
        finalPRG = target_risk / (average_activities * SOURCE.astype('float') * exposure_duration).sum()

    outdata = [route, peak_start, finalPRG, peak_risk, maxRate]
    outdata = [str(item) for item in outdata]
    print(','.join(outdata), file = res)

res.close()
```

APPENDIX F. PEAK RISK (ELCR) SOLVER

APPENDIX F. PEAK RISK (ELCR) SOLVER

```
# Peak Risk Solver
import numpy as np
from scipy import integrate
import pandas as pd
from scipy.interpolate import interp1d
from scipy.integrate import quad
from scipy.optimize import root_scalar
from cram import cram48
from sys import argv
import csv

if len(argv) < 2:
    # Resident Defaults for Soil
    # Am-241
    scenario = 'Resident'
    media = 'soil'
    parent = 'Am-241'
    numElements = int(13)
    exposure_duration = float(26)
    target_risk = float(1.0E-06)
    filenum = '22898'
    units = 'pCi'
    withrisk = 'no'

else:
    scenario = argv[1]
    media = argv[2]
    parent = argv[3]
    numElements = int(argv[4])
    exposure_duration = float(argv[5])
    target_risk = float(argv[6])
    filenum = argv[7]
    units = argv[8]
    withrisk = argv[9]
    chaindir = argv[10]
    actdir = argv[11]
    chaindir = '/data/tmp/chain/'

nodash = parent.replace('-', '')

# Inputs
activity_file = actdir + parent + '_cram.csv'
matrix_file = actdir + parent + '_activity_matrix.npy'
results_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_results.csv'
rate_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_rates.csv'
conc_file = chaindir + scenario + '_' + media + '_' + filenum + '_concs.csv'
```

```
# Outputs
risk_results_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_risk_results.csv'
aucrisk_results_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum +
'_aucrisk_results.csv'
risk_concs_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + filenum + '_risk_concs.csv'

# Activities
actdata = pd.read_csv(activity_file,dtype='float')
actdata.drop(actdata[actdata.Time < 1e-16].index, inplace=True)
actdata.set_index("Time", inplace=True)

# Get Times and put in array
T = np.array(actdata.index).astype('float')
labels = list(actdata.columns.values.tolist())

# Prepare output
totals = actdata.copy()
totals.drop(totals.tail(len(totals)).index,inplace=True)
iconcs = actdata.copy()
iconcs.drop(iconcs.tail(len(iconcs)).index,inplace=True)

# PRG Output Results
header_list = ["route", "peak_start", "peak_prg", "peak_risk", "peak_max"]
results = pd.read_csv(results_file, names=header_list)
if media == 'ext2d':
    results['route'] = results['route'].str.replace('prg','ext')

# Risk Rates
ratedata = pd.read_csv(rate_file, na_values='inf')
ratedata = ratedata.fillna(0)

# Media Concentrations
concs = pd.read_csv(conc_file)
concs.columns = map(str.lower, concs.columns)
pconcs = concs[concs['analysis'] == parent]

# AUC begin using finalPRG from risk rates
# slice out time series with peak start and peak end times
# add all the pst and pet to activity file then integrate by route
resulttimes = results["peak_start"]
addtimesflag = 0

for pt in range(len(resulttimes)):
    pst = float(resulttimes[pt])
    pet = float(pst + exposure_duration)

    # Let's add those points if not already there
    if pst not in T and pst > 0:
        addtimesflag = 1
        T = np.insert(T,T.searchsorted(pst),pst)
```

```
    if pet not in T and pet > 0:
        addtimesflag = 1
        T = np.insert(T,T.searchsorted(pet),pet)

if addtimesflag == 1:
    # Make a new unit activity file with peak start points added
    A = np.load(matrix_file)
    a0 = np.zeros(A.shape[0])
    a0[0] = float(1.000000e+00)
    sol = cram48(A,a0,T)

    cramdf = pd.DataFrame(sol,columns=labels)
    cramdf.loc[:,'Time'] = T
    cramdf = cramdf.set_index('Time')
else:
    # Use existing activity file
    cramdf = actdata

# New df for AUC results
routes = ratedata['route']
aucresults = pd.DataFrame(index=routes,columns=labels)

# Loop over routes starting here
# Each route has different toxicity and ELCR time range
for x in range(len(ratedata.index)):
    route = ratedata['route'][x]
    riskname = route + 'risk'
    ratedf = ratedata.loc[ratedata['route'] == route]
    ratedf = ratedf.set_index("route", drop = True)
    ratedf = ratedf.rename_axis('ID').values

    # Recreate risk rates with new activity including the PST and PET
    risk_rates = cramdf * ratedf
    routeres = results.loc[results['route'] == route]
    peak_start = routeres.iloc[0]['peak_start']

    pst = float(peak_start)
    pet = float(pst + exposure_duration)

    # Starting Concentrations at PST
    sa = cramdf[(cramdf.index >= pst) & (cramdf.index <= pet)]
    if media == 'ext2d' or media == 'biota' or media == 'game':
        curconc = route + 'c'
        pconc = pconcs.iloc[0][curconc]

        # Multiply Parent concentration by starting activity
        sa = sa.iloc[0]*pconc
        sa['route'] = curconc
        iconcs = iconcs.append(sa)
    else:
        curconc = media + 'c'
```

```
        pconc = pconcs.iloc[0][curconc]

        # Multiply Parent concentration by starting activity
        sa = sa.iloc[0]*pconc
        sa['route'] = route + 'c'
        iconcs = iconcs.append(sa)

    # Calculate risk for peak time interval
    riskdf = risk_rates[(risk_rates.index >= pst) & (risk_rates.index <= pet)] * pconc

    # Set absolute floor value
    riskdf = riskdf.mask(riskdf < 1e-50, 0)

    # Get sum of risk and add to dataframe
    risk_sum = np.array(riskdf.sum(axis=1))
    riskdf["Total Risk"] = risk_sum
    riskdf = riskdf.loc[riskdf['Total Risk'] > 0]

    # Calculating Area Under Curve
    # With Samples, Unequal Interval Sizes, Simpson's Rule
    # Need to calculate for each isotope
    Time = np.array(riskdf.index).astype("float")
    for column in riskdf:
        Risk = riskdf.loc[:,column].values.astype("float")
        spy_simps_sample = integrate.simps(Risk, Time)
        aucresults.loc[route,column] = spy_simps_sample

    # This file is for plotting by route, if needed
    cdi_output_file = chaindir + nodash + '_' + scenario + '_' + media + '_' + route + '_' + filenum +
'_risk_output.csv' ;
    open(cdi_output_file, 'w')

    # Write output file
    riskdf.to_csv(cdi_output_file)

    CDI_data = riskdf.copy()
    CDI_data.drop('Total Risk', axis=1, inplace=True)
    CDI_data.loc[riskname] = CDI_data.sum(numeric_only=True, axis=0)

    # Create total risk row
    totals = totals.append(CDI_data.iloc[len(CDI_data)-1])

# Set up output files with totals for reports and plotting
newconcs = iconcs.reset_index()
newconcs.set_index('Time').T

totals = totals.transpose().reset_index()
result = totals.rename(columns={"index":"Analysis"})

total = result.apply(np.sum)
total['Analysis'] = 'Total'
```

```
result = result.append(pd.DataFrame(total.values, index=total.keys()).T, ignore_index=True)

aucresults = aucresults.T.reset_index()
aucresults = aucresults.add_suffix('risk')
aucresults = aucresults.rename(columns={"indexrisk":"Analysis"})

# Write to output files
aucresults.to_csv(aucrisk_results_file)
result.to_csv(risk_results_file)
newconcs.to_csv(risk_concs_file)
```

APPENDIX G. BATEMAN SOLVER (C++ ODE METHOD) UAF FOR RA-226

APPENDIX G. BATEMAN SOLVER (C++ ODE METHOD) UAF FOR RA-226

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E-05	1.00000E+00	8.33247E-04	3.98534E-04	7.90073E-08	7.89910E-11	2.41429E-05	1.43382E-06	1.43360E-06	1.36032E-10	1.19490E-13	1.32052E-17	4.48635E-23	2.05041E-22	3.16660E-23
1.58489E-05	1.00000E+00	1.04888E-03	5.73477E-04	1.13930E-07	1.13912E-10	4.41431E-05	3.30809E-06	3.30751E-06	3.59290E-10	3.51584E-13	4.93668E-17	2.12613E-22	7.50792E-22	1.42715E-22
1.99526E-05	1.00000E+00	1.32029E-03	8.11590E-04	1.61498E-07	1.61479E-10	7.93271E-05	7.49709E-06	7.49567E-06	9.18233E-10	1.01884E-12	1.82111E-16	9.95726E-22	2.69845E-21	6.27870E-22
2.51189E-05	1.00000E+00	1.66186E-03	1.12880E-03	2.24901E-07	2.24881E-10	1.39816E-04	1.66396E-05	1.66363E-05	2.26286E-09	2.89997E-12	6.61252E-16	4.59768E-21	9.48401E-21	2.68298E-21
3.16228E-05	1.00000E+00	2.09170E-03	1.54319E-03	3.07754E-07	3.07733E-10	2.41254E-04	3.60567E-05	3.60494E-05	5.36119E-09	8.08454E-12	2.35710E-15	2.08787E-20	3.24596E-20	1.10734E-20
3.98107E-05	1.00000E+00	2.63258E-03	2.07593E-03	4.14288E-07	4.14267E-10	4.06975E-04	7.60463E-05	7.60307E-05	1.21821E-08	2.20109E-11	8.22580E-15	9.30024E-20	1.07698E-19	4.38759E-20
5.01187E-05	1.00000E+00	3.31310E-03	2.75316E-03	5.49729E-07	5.49708E-10	6.70566E-04	1.55639E-04	1.55607E-04	2.65016E-08	5.83593E-11	2.80268E-14	4.05280E-19	3.44767E-19	1.65851E-19
6.30957E-05	1.00000E+00	4.16915E-03	3.60848E-03	7.20792E-07	7.20771E-10	1.07871E-03	3.08229E-04	3.08165E-04	5.51260E-08	1.50276E-10	9.29830E-14	1.72323E-18	1.05976E-18	5.94376E-19
7.94328E-05	1.00000E+00	5.24582E-03	4.68546E-03	9.36189E-07	9.36168E-10	1.69395E-03	5.89065E-04	5.88942E-04	1.09536E-07	3.74859E-10	2.99622E-13	7.13126E-18	3.11334E-18	2.00820E-18
1.00000E-04	1.00000E+00	6.59960E-03	6.03997E-03	1.20709E-06	1.20707E-09	2.59679E-03	1.08358E-03	1.08336E-03	2.07725E-07	9.03660E-10	9.35533E-13	2.86549E-17	8.70347E-18	6.36724E-18
1.25893E-04	1.00000E+00	8.30129E-03	7.74261E-03	1.54762E-06	1.54760E-09	3.88619E-03	1.91381E-03	1.91341E-03	3.75639E-07	2.10055E-09	2.82429E-12	1.11555E-16	2.30663E-17	1.88870E-17
1.58489E-04	1.00000E+00	1.04395E-02	9.88198E-03	1.97550E-06	1.97548E-09	5.67827E-03	3.23853E-03	3.23785E-03	6.47223E-07	4.69877E-09	8.22729E-12	4.19917E-16	5.77867E-17	5.23536E-17
1.99526E-04	1.00000E+00	1.31247E-02	1.25687E-02	2.51285E-06	2.51283E-09	8.10369E-03	5.24327E-03	5.24217E-03	1.06223E-06	1.00988E-08	2.30866E-11	1.52564E-15	1.36618E-16	1.35758E-16
2.51189E-04	1.00000E+00	1.64948E-02	1.59408E-02	3.18726E-06	3.18724E-09	1.13066E-02	8.12169E-03	8.11998E-03	1.66205E-06	2.08367E-08	6.23277E-11	5.34237E-15	3.04759E-16	3.30448E-16
3.16228E-04	1.00000E+00	2.07212E-02	2.01696E-02	4.03302E-06	4.03300E-09	1.54487E-02	1.20593E-02	1.20568E-02	2.48600E-06	4.12864E-08	1.61814E-10	1.80144E-14	6.42602E-16	7.59386E-16
3.98107E-04	1.00000E+00	2.60161E-02	2.54674E-02	5.09260E-06	5.09258E-09	2.07211E-02	1.72363E-02	1.72327E-02	3.57178E-06	7.87031E-08	4.04222E-10	5.84881E-14	1.28562E-15	1.66049E-15
5.01187E-04	1.00000E+00	3.26414E-02	3.20964E-02	6.41840E-06	6.41838E-09	2.73626E-02	2.38570E-02	2.38520E-02	4.96179E-06	1.44821E-07	9.73354E-10	1.83006E-13	2.45427E-15	3.48679E-15
6.30957E-04	1.00000E+00	4.09180E-02	4.03777E-02	8.07466E-06	8.07464E-09	3.56777E-02	3.21906E-02	3.21839E-02	6.71207E-06	2.58380E-07	2.26594E-09	5.52913E-13	4.50132E-15	7.09910E-15
7.94328E-04	1.00000E+00	5.12370E-02	5.07025E-02	1.01396E-05	1.01396E-08	4.60500E-02	4.26001E-02	4.25912E-02	8.89854E-06	4.49173E-07	5.11896E-09	1.61766E-12	7.98816E-15	1.41391E-14
1.00000E-03	1.00000E+00	6.40701E-02	6.35429E-02	1.27077E-05	1.27077E-08	5.89505E-02	5.55497E-02	5.55380E-02	1.16186E-05	7.64405E-07	1.12674E-08	4.59894E-12	1.38056E-14	2.77498E-14
1.25893E-03	9.99999E-01	7.99795E-02	7.94612E-02	1.58914E-05	1.58914E-08	7.49436E-02	7.16039E-02	7.15889E-02	1.49908E-05	1.27856E-06	2.42583E-08	1.27511E-11	2.33613E-14	5.39629E-14
1.58489E-03	9.99999E-01	9.96244E-02	9.91171E-02	1.98226E-05	1.98226E-08	9.46916E-02	9.14276E-02	9.14084E-02	1.91547E-05	2.10854E-06	5.12580E-08	3.45992E-11	3.88677E-14	1.04362E-13
1.99526E-03	9.99999E-01	1.23760E-01	1.23267E-01	2.46526E-05	2.46525E-08	1.18955E-01	1.15783E-01	1.15759E-01	2.42706E-05	3.43670E-06	1.06591E-07	9.21607E-11	6.37792E-14	2.01175E-13
2.51189E-03	9.99999E-01	1.53228E-01	1.52751E-01	3.05494E-05	3.05494E-08	1.48577E-01	1.45519E-01	1.45489E-01	3.05166E-05	5.54546E-06	2.18597E-07	2.41593E-10	1.03451E-13	3.86964E-13
3.16228E-03	9.99999E-01	1.88920E-01	1.88464E-01	3.76920E-05	3.76920E-08	1.84457E-01	1.81537E-01	1.81499E-01	3.80821E-05	8.86871E-06	4.42742E-07	6.24487E-10	1.66114E-13	7.42881E-13
3.98107E-03	9.99998E-01	2.31723E-01	2.31290E-01	4.62574E-05	4.62574E-08	2.27485E-01	2.24729E-01	2.24682E-01	4.71547E-05	1.40665E-05	8.86340E-07	1.59388E-09	2.64299E-13	1.42268E-12

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
5.01187E-03	9.99998E-01	2.82412E-01	2.82007E-01	5.64008E-05	5.64008E-08	2.78440E-01	2.75879E-01	2.75821E-01	5.78987E-05	2.21312E-05	1.75429E-06	4.01998E-09	4.16849E-13	2.71506E-12
6.30957E-03	9.99997E-01	3.41497E-01	3.41126E-01	6.82246E-05	6.82246E-08	3.37835E-01	3.35502E-01	3.35432E-01	7.04224E-05	3.45333E-05	3.43184E-06	1.00216E-08	6.51697E-13	5.15563E-12
7.94328E-03	9.99997E-01	4.09013E-01	4.08680E-01	8.17354E-05	8.17354E-08	4.05706E-01	4.03633E-01	4.03548E-01	8.47333E-05	5.34137E-05	6.62991E-06	2.46870E-08	1.00952E-12	9.72241E-12
1.00000E-02	9.99996E-01	4.84256E-01	4.83966E-01	9.67926E-05	9.67926E-08	4.81345E-01	4.79561E-01	4.79460E-01	1.00682E-04	8.18225E-05	1.26311E-05	6.00434E-08	1.54828E-12	1.81650E-11
1.25893E-02	9.99995E-01	5.65513E-01	5.65269E-01	1.13053E-04	1.13053E-07	5.63030E-01	5.61558E-01	5.61440E-01	1.17905E-04	1.23993E-04	2.36856E-05	1.43994E-07	2.34843E-12	3.35327E-11
1.58489E-02	9.99993E-01	6.49861E-01	6.49664E-01	1.29932E-04	1.29932E-07	6.47821E-01	6.46673E-01	6.46538E-01	1.35784E-04	1.85618E-04	4.36069E-05	3.39857E-07	3.51817E-12	6.09659E-11
1.99526E-02	9.99991E-01	7.33170E-01	7.33020E-01	1.46604E-04	1.46604E-07	7.31567E-01	7.30741E-01	7.30587E-01	1.53442E-04	2.74075E-04	7.85861E-05	7.87564E-07	5.19775E-12	1.08776E-10
2.51189E-02	9.99989E-01	8.10470E-01	8.10363E-01	1.62073E-04	1.62073E-07	8.09274E-01	8.08745E-01	8.08575E-01	1.69827E-04	3.98542E-04	1.38154E-04	1.78681E-06	7.56159E-12	1.89719E-10
3.16228E-02	9.99986E-01	8.76786E-01	8.76716E-01	1.75343E-04	1.75343E-07	8.75938E-01	8.75664E-01	8.75480E-01	1.83883E-04	5.69957E-04	2.36049E-04	3.95623E-06	1.08176E-11	3.22140E-10
3.98107E-02	9.99983E-01	9.28346E-01	9.28305E-01	1.85661E-04	1.85661E-07	9.27769E-01	9.27693E-01	9.27499E-01	1.94812E-04	8.00895E-04	3.90562E-04	8.51964E-06	1.52047E-11	5.30425E-10
5.01187E-02	9.99978E-01	9.63781E-01	9.63761E-01	1.92752E-04	1.92752E-07	9.63391E-01	9.63451E-01	9.63249E-01	2.02323E-04	1.10557E-03	6.23870E-04	1.77857E-05	2.09930E-11	8.44125E-10
6.30957E-02	9.99973E-01	9.84648E-01	9.84639E-01	1.96928E-04	1.96928E-07	9.84367E-01	9.84508E-01	9.84302E-01	2.06746E-04	1.50028E-03	9.60292E-04	3.58942E-05	2.84923E-11	1.29566E-09
7.94328E-02	9.99966E-01	9.94775E-01	9.94772E-01	1.98954E-04	1.98954E-07	9.94548E-01	9.94727E-01	9.94519E-01	2.08892E-04	2.00464E-03	1.42437E-03	6.99020E-05	3.80750E-11	1.91780E-09
1.00000E-01	9.99957E-01	9.98632E-01	9.98631E-01	1.99726E-04	1.99726E-07	9.98425E-01	9.98619E-01	9.98410E-01	2.09710E-04	2.64345E-03	2.04064E-03	1.31303E-04	5.02124E-11	2.74338E-09
1.25893E-01	9.99945E-01	9.99712E-01	9.99712E-01	1.99942E-04	1.99942E-07	9.99511E-01	9.99710E-01	9.99500E-01	2.09939E-04	3.44888E-03	2.83648E-03	2.38154E-04	6.55155E-11	3.80920E-09
1.58489E-01	9.99931E-01	9.99910E-01	9.99910E-01	1.99982E-04	1.99982E-07	9.99710E-01	9.99910E-01	9.99700E-01	2.09981E-04	4.46246E-03	3.84745E-03	4.18121E-04	8.47735E-11	5.16293E-09
1.99526E-01	9.99914E-01	9.99918E-01	9.99918E-01	1.99984E-04	1.99984E-07	9.99718E-01	9.99918E-01	9.99708E-01	2.09983E-04	5.73711E-03	5.12219E-03	7.12850E-04	1.08992E-10	6.86981E-09
2.51189E-01	9.99891E-01	9.99898E-01	9.99898E-01	1.99980E-04	1.99980E-07	9.99698E-01	9.99898E-01	9.99688E-01	2.09978E-04	7.33946E-03	6.72545E-03	1.18399E-03	1.39437E-10	9.01655E-09
3.16228E-01	9.99863E-01	9.99870E-01	9.99870E-01	1.99974E-04	1.99974E-07	9.99670E-01	9.99869E-01	9.99660E-01	2.09973E-04	9.35298E-03	8.74023E-03	1.92097E-03	1.77694E-10	1.17143E-08
3.98107E-01	9.99828E-01	9.99834E-01	9.99834E-01	1.99967E-04	1.99967E-07	9.99634E-01	9.99834E-01	9.99624E-01	2.09965E-04	1.18820E-02	1.12708E-02	3.05020E-03	2.25744E-10	1.51027E-08
5.01187E-01	9.99783E-01	9.99789E-01	9.99789E-01	1.99958E-04	1.99958E-07	9.99590E-01	9.99789E-01	9.99580E-01	2.09956E-04	1.50565E-02	1.44473E-02	4.74485E-03	2.86060E-10	1.93560E-08
6.30957E-01	9.99727E-01	9.99733E-01	9.99733E-01	1.99947E-04	1.99947E-07	9.99533E-01	9.99733E-01	9.99523E-01	2.09944E-04	1.90382E-02	1.84315E-02	7.23337E-03	3.61713E-10	2.46909E-08
7.94328E-01	9.99656E-01	9.99662E-01	9.99662E-01	1.99932E-04	1.99932E-07	9.99463E-01	9.99662E-01	9.99453E-01	2.09929E-04	2.40277E-02	2.34242E-02	1.08042E-02	4.56514E-10	3.13760E-08
1.00000E+00	9.99567E-01	9.99573E-01	9.99573E-01	1.99915E-04	1.99915E-07	9.99374E-01	9.99573E-01	9.99364E-01	2.09910E-04	3.02726E-02	2.96729E-02	1.58040E-02	5.75166E-10	3.97430E-08
1.25893E+00	9.99455E-01	9.99461E-01	9.99461E-01	1.99892E-04	1.99892E-07	9.99261E-01	9.99461E-01	9.99251E-01	2.09887E-04	3.80767E-02	3.74820E-02	2.26282E-02	7.23445E-10	5.01992E-08
1.58489E+00	9.99314E-01	9.99320E-01	9.99320E-01	1.99864E-04	1.99864E-07	9.99120E-01	9.99320E-01	9.99110E-01	2.09857E-04	4.78110E-02	4.72224E-02	3.17082E-02	9.08397E-10	6.32415E-08
1.99526E+00	9.99136E-01	9.99143E-01	9.99143E-01	1.99829E-04	1.99829E-07	9.98943E-01	9.99142E-01	9.98933E-01	2.09820E-04	5.99237E-02	5.93427E-02	4.35033E-02	1.13854E-09	7.94704E-08
2.51189E+00	9.98912E-01	9.98919E-01	9.98919E-01	1.99784E-04	1.99784E-07	9.98719E-01	9.98919E-01	9.98709E-01	2.09773E-04	7.49504E-02	7.43788E-02	5.85100E-02	1.42405E-09	9.96036E-08
3.16228E+00	9.98631E-01	9.98638E-01	9.98638E-01	1.99728E-04	1.99728E-07	9.98438E-01	9.98637E-01	9.98428E-01	2.09714E-04	9.35214E-02	9.29615E-02	7.72900E-02	1.77690E-09	1.24486E-07
3.98107E+00	9.98277E-01	9.98283E-01	9.98283E-01	1.99657E-04	1.99657E-07	9.98084E-01	9.98283E-01	9.98074E-01	2.09639E-04	1.16363E-01	1.15817E-01	1.00505E-01	2.21089E-09	1.55090E-07
5.01187E+00	9.97831E-01	9.97838E-01	9.97838E-01	1.99568E-04	1.99568E-07	9.97638E-01	9.97838E-01	9.97628E-01	2.09546E-04	1.44288E-01	1.43760E-01	1.28929E-01	2.74146E-09	1.92505E-07

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
6.30957E+00	9.97270E-01	9.97277E-01	9.97277E-01	1.99455E-04	1.99455E-07	9.97077E-01	9.97277E-01	9.97067E-01	2.09428E-04	1.78170E-01	1.77663E-01	1.63429E-01	3.38522E-09	2.37900E-07
7.94328E+00	9.96565E-01	9.96571E-01	9.96571E-01	1.99314E-04	1.99314E-07	9.96372E-01	9.96571E-01	9.96362E-01	2.09280E-04	2.18886E-01	2.18405E-01	2.04890E-01	4.15883E-09	2.92453E-07
1.00000E+01	9.95677E-01	9.95684E-01	9.95684E-01	1.99137E-04	1.99137E-07	9.95485E-01	9.95684E-01	9.95475E-01	2.09094E-04	2.67229E-01	2.66778E-01	2.54119E-01	5.07734E-09	3.57224E-07
1.25893E+01	9.94561E-01	9.94567E-01	9.94567E-01	1.98913E-04	1.98913E-07	9.94369E-01	9.94567E-01	9.94359E-01	2.08859E-04	3.23758E-01	3.23343E-01	3.11685E-01	6.15140E-09	4.32964E-07
1.58489E+01	9.93157E-01	9.93164E-01	9.93164E-01	1.98633E-04	1.98633E-07	9.92965E-01	9.93164E-01	9.92955E-01	2.08564E-04	3.88603E-01	3.88229E-01	3.77721E-01	7.38344E-09	5.19845E-07
1.99526E+01	9.91393E-01	9.91400E-01	9.91400E-01	1.98280E-04	1.98280E-07	9.91202E-01	9.91400E-01	9.91192E-01	2.08194E-04	4.61199E-01	4.60871E-01	4.51655E-01	8.76277E-09	6.17112E-07
2.51189E+01	9.89177E-01	9.89184E-01	9.89184E-01	1.97837E-04	1.97837E-07	9.88986E-01	9.89183E-01	9.88976E-01	2.07729E-04	5.40012E-01	5.39734E-01	5.31926E-01	1.02602E-08	7.22709E-07
3.16228E+01	9.86394E-01	9.86400E-01	9.86400E-01	1.97280E-04	1.97280E-07	9.86203E-01	9.86400E-01	9.86193E-01	2.07144E-04	6.22296E-01	6.22071E-01	6.15740E-01	1.18236E-08	8.32957E-07
3.98107E+01	9.82901E-01	9.82908E-01	9.82908E-01	1.96582E-04	1.96582E-07	9.82711E-01	9.82907E-01	9.82701E-01	2.06411E-04	7.04022E-01	7.03850E-01	6.99000E-01	1.33764E-08	9.42458E-07
5.01187E+01	9.78522E-01	9.78528E-01	9.78528E-01	1.95706E-04	1.95706E-07	9.78332E-01	9.78528E-01	9.78323E-01	2.05491E-04	7.80133E-01	7.80011E-01	7.76559E-01	1.48225E-08	1.04444E-06
6.30957E+01	9.73036E-01	9.73042E-01	9.73042E-01	1.94608E-04	1.94608E-07	9.72848E-01	9.73042E-01	9.72838E-01	2.04339E-04	8.45251E-01	8.45172E-01	8.42947E-01	1.60598E-08	1.13169E-06
7.94328E+01	9.66174E-01	9.66180E-01	9.66180E-01	1.93236E-04	1.93236E-07	9.65987E-01	9.66180E-01	9.65977E-01	2.02898E-04	8.94823E-01	8.94778E-01	8.93533E-01	1.70016E-08	1.19811E-06
1.00000E+02	9.57603E-01	9.57610E-01	9.57610E-01	1.91522E-04	1.91522E-07	9.57418E-01	9.57609E-01	9.57409E-01	2.01098E-04	9.26386E-01	9.26366E-01	9.25819E-01	1.76013E-08	1.24041E-06
1.25893E+02	9.46922E-01	9.46928E-01	9.46928E-01	1.89386E-04	1.89386E-07	9.46739E-01	9.46928E-01	9.46729E-01	1.98855E-04	9.40336E-01	9.40332E-01	9.40212E-01	1.78664E-08	1.25911E-06
1.58489E+02	9.33644E-01	9.33650E-01	9.33650E-01	1.86730E-04	1.86730E-07	9.33463E-01	9.33650E-01	9.33454E-01	1.96066E-04	9.39589E-01	9.39593E-01	9.39691E-01	1.78522E-08	1.25811E-06
1.99526E+02	9.17192E-01	9.17198E-01	9.17198E-01	1.83440E-04	1.83440E-07	9.17015E-01	9.17198E-01	9.17006E-01	1.92612E-04	9.28105E-01	9.28112E-01	9.28296E-01	1.76340E-08	1.24274E-06
2.51189E+02	8.96893E-01	8.96898E-01	8.96898E-01	1.79380E-04	1.79380E-07	8.96719E-01	8.96898E-01	8.96710E-01	1.88349E-04	9.09120E-01	9.09127E-01	9.09335E-01	1.72733E-08	1.21732E-06
3.16228E+02	8.71974E-01	8.71980E-01	8.71980E-01	1.74396E-04	1.74396E-07	8.71806E-01	8.71980E-01	8.71797E-01	1.83116E-04	8.84197E-01	8.84204E-01	8.84412E-01	1.67997E-08	1.18395E-06
3.98107E+02	8.41586E-01	8.41592E-01	8.41592E-01	1.68318E-04	1.68318E-07	8.41423E-01	8.41592E-01	8.41415E-01	1.76734E-04	8.53429E-01	8.53436E-01	8.53637E-01	1.62152E-08	1.14275E-06
5.01187E+02	8.04831E-01	8.04836E-01	8.04836E-01	1.60967E-04	1.60967E-07	8.04675E-01	8.04836E-01	8.04667E-01	1.69016E-04	8.16160E-01	8.16167E-01	8.16360E-01	1.55070E-08	1.09285E-06
6.30957E+02	7.60833E-01	7.60838E-01	7.60838E-01	1.52168E-04	1.52168E-07	7.60686E-01	7.60838E-01	7.60678E-01	1.59776E-04	7.71543E-01	7.71550E-01	7.71731E-01	1.46593E-08	1.03310E-06
7.94328E+02	7.08846E-01	7.08851E-01	7.08851E-01	1.41770E-04	1.41770E-07	7.08709E-01	7.08851E-01	7.08702E-01	1.48859E-04	7.18825E-01	7.18831E-01	7.19000E-01	1.36577E-08	9.62514E-07
1.00000E+03	6.48420E-01	6.48424E-01	6.48424E-01	1.29685E-04	1.29685E-07	6.48294E-01	6.48424E-01	6.48288E-01	1.36169E-04	6.57548E-01	6.57553E-01	6.57708E-01	1.24934E-08	8.80464E-07
1.25893E+03	5.79617E-01	5.79621E-01	5.79621E-01	1.15924E-04	1.15924E-07	5.79505E-01	5.79621E-01	5.79499E-01	1.21720E-04	5.87776E-01	5.87781E-01	5.87920E-01	1.11677E-08	7.87039E-07
1.58489E+03	5.03283E-01	5.03286E-01	5.03286E-01	1.00657E-04	1.00657E-07	5.03186E-01	5.03286E-01	5.03181E-01	1.05690E-04	5.10368E-01	5.10372E-01	5.10492E-01	9.69699E-09	6.83388E-07
1.99526E+03	4.21312E-01	4.21315E-01	4.21315E-01	8.42630E-05	8.42630E-08	4.21231E-01	4.21315E-01	4.21226E-01	8.84761E-05	4.27243E-01	4.27246E-01	4.27347E-01	8.11761E-09	5.72083E-07
2.51189E+03	3.36825E-01	3.36827E-01	3.36827E-01	6.73654E-05	6.73654E-08	3.36760E-01	3.36827E-01	3.36756E-01	7.07337E-05	3.41566E-01	3.41569E-01	3.41650E-01	6.48976E-09	4.57361E-07
3.16228E+03	2.54119E-01	2.54121E-01	2.54121E-01	5.08241E-05	5.08241E-08	2.54070E-01	2.54121E-01	2.54067E-01	5.33653E-05	2.57696E-01	2.57698E-01	2.57759E-01	4.89623E-09	3.45058E-07
3.98107E+03	1.78232E-01	1.78233E-01	1.78233E-01	3.56467E-05	3.56467E-08	1.78198E-01	1.78233E-01	1.78196E-01	3.74290E-05	1.80741E-01	1.80743E-01	1.80785E-01	3.43408E-09	2.42015E-07
5.01187E+03	1.14037E-01	1.14038E-01	1.14038E-01	2.28076E-05	2.28076E-08	1.14015E-01	1.14038E-01	1.14014E-01	2.39480E-05	1.15643E-01	1.15644E-01	1.15671E-01	2.19721E-09	1.54847E-07
6.30957E+03	6.49970E-02	6.49974E-02	6.49974E-02	1.29995E-05	1.29995E-08	6.49844E-02	6.49974E-02	6.49838E-02	1.36495E-05	6.59119E-02	6.59125E-02	6.59280E-02	1.25233E-09	8.82568E-08

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
7.94328E+03	3.20274E-02	3.20276E-02	3.20276E-02	6.40551E-06	6.40551E-09	3.20212E-02	3.20276E-02	3.20208E-02	6.72579E-06	3.24782E-02	3.24785E-02	3.24861E-02	6.17086E-10	4.34887E-08
1.00000E+04	1.31390E-02	1.31391E-02	1.31391E-02	2.62782E-06	2.62782E-09	1.31365E-02	1.31391E-02	1.31363E-02	2.75921E-06	1.33240E-02	1.33241E-02	1.33272E-02	2.53155E-10	1.78409E-08
1.25893E+04	4.27967E-03	4.27970E-03	4.27970E-03	8.55940E-07	8.55940E-10	4.27884E-03	4.27970E-03	4.27880E-03	8.98737E-07	4.33992E-03	4.33995E-03	4.34098E-03	8.24584E-11	5.81120E-09
1.58489E+04	1.04261E-03	1.04262E-03	1.04262E-03	2.08524E-07	2.08524E-10	1.04241E-03	1.04262E-03	1.04240E-03	2.18950E-07	1.05729E-03	1.05730E-03	1.05755E-03	2.00885E-11	1.41572E-09
1.99526E+04	1.76213E-04	1.76214E-04	1.76214E-04	3.52429E-08	3.52429E-11	1.76179E-04	1.76214E-04	1.76177E-04	3.70050E-08	1.78694E-04	1.78695E-04	1.78737E-04	3.39518E-12	2.39273E-10
2.51189E+04	1.87950E-05	1.87951E-05	1.87951E-05	3.75903E-09	3.75903E-12	1.87914E-05	1.87951E-05	1.87912E-05	3.94698E-09	1.90596E-05	1.90598E-05	1.90642E-05	3.62132E-13	2.55210E-11
3.16228E+04	1.12298E-06	1.12299E-06	1.12299E-06	2.24598E-10	2.24598E-13	1.12276E-06	1.12299E-06	1.12275E-06	2.35827E-10	1.13879E-06	1.13880E-06	1.13907E-06	2.16370E-14	1.52485E-12
3.98107E+04	3.23491E-08	3.23493E-08	3.23493E-08	6.46987E-12	6.46987E-15	3.23429E-08	3.23493E-08	3.23425E-08	6.79336E-12	3.28045E-08	3.28048E-08	3.28125E-08	6.23285E-16	4.39256E-14
5.01187E+04	3.71942E-10	3.71945E-10	3.71945E-10	7.43889E-14	7.43889E-17	3.71870E-10	3.71945E-10	3.71867E-10	7.81084E-14	3.77178E-10	3.77181E-10	3.77270E-10	7.16638E-18	5.05046E-16
6.30957E+04	1.34565E-12	1.34566E-12	1.34566E-12	2.69132E-16	2.69132E-19	1.34539E-12	1.34566E-12	1.34538E-12	2.82588E-16	1.36459E-12	1.36460E-12	1.36493E-12	2.59273E-20	1.82720E-18
7.94328E+04	1.13556E-15	1.13557E-15	1.13557E-15	2.27114E-19	2.27114E-22	1.13534E-15	1.13557E-15	1.13533E-15	2.38470E-19	1.15155E-15	1.15156E-15	1.15183E-15	2.18794E-23	1.54194E-21
1.00000E+05	1.53329E-19	1.53330E-19	1.53330E-19	3.06661E-23	3.06661E-26	1.53300E-19	1.53330E-19	1.53298E-19	3.21994E-23	1.55488E-19	1.55489E-19	1.55526E-19	2.95427E-27	2.08200E-25
1.25893E+05	2.06113E-24	2.06114E-24	2.06114E-24	4.12229E-28	4.12229E-31	2.06073E-24	2.06114E-24	2.06071E-24	4.32840E-28	2.09015E-24	2.09016E-24	2.09066E-24	3.97128E-32	2.79873E-30
1.58489E+05	1.51784E-30	1.51785E-30	1.51785E-30	3.03570E-34	3.03570E-37	1.51754E-30	1.51785E-30	1.51753E-30	3.18748E-34	1.53921E-30	1.53922E-30	1.53958E-30	2.92449E-38	2.06101E-36
1.99526E+05	2.88660E-38	2.88662E-38	2.88662E-38	5.77324E-42	5.77324E-45	2.88605E-38	2.88662E-38	2.88602E-38	6.06191E-42	2.92724E-38	2.92726E-38	2.92795E-38	5.56175E-46	3.91960E-44
2.51189E+05	5.50082E-48	5.50086E-48	5.50086E-48	1.10017E-51	1.10017E-54	5.49976E-48	5.50085E-48	5.49970E-48	1.15518E-51	5.57825E-48	5.57830E-48	5.57962E-48	1.05987E-55	7.46935E-54
3.16228E+05	3.18950E-60	3.18952E-60	3.18952E-60	6.37905E-64	6.37905E-67	3.18889E-60	3.18952E-60	3.18885E-60	6.69800E-64	3.23440E-60	3.23443E-60	3.23519E-60	6.14536E-68	4.33090E-66
3.98107E+05	1.25495E-75	1.25495E-75	1.25495E-75	2.50991E-79	2.50991E-82	1.25470E-75	1.25495E-75	1.25469E-75	2.63540E-79	1.27261E-75	1.27262E-75	1.27292E-75	2.41796E-83	1.70404E-81
5.01187E+05	5.06706E-95	5.06709E-95	5.06709E-95	1.01342E-98	0	5.06608E-95	5.06709E-95	5.06603E-95	1.06409E-98	5.13839E-95	5.13843E-95	5.13964E-95	0	0
6.30957E+05	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+05	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
1.00000E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+10	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APPENDIX H. BATEMAN SOLVER (CRAM METHOD) UAF FOR RA-226

APPENDIX H. BATEMAN SOLVER (CRAM METHOD) UAF FOR RA-226

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
0.00000E+00	1.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
1.25893E-05	1.00000E+00	8.32500E-04	3.97957E-04	7.88922E-08	7.88758E-11	2.40855E-05	1.42912E-06	1.42889E-06	1.35510E-10	1.19067E-13	1.31462E-17	4.46220E-23	2.04140E-22	3.15011E-23
1.58489E-05	1.00000E+00	1.04794E-03	5.72683E-04	1.13771E-07	1.13753E-10	4.40409E-05	3.29744E-06	3.29686E-06	3.57955E-10	3.50358E-13	4.91487E-17	2.11478E-22	7.47542E-22	1.41984E-22
1.99526E-05	1.00000E+00	1.31910E-03	8.10519E-04	1.61284E-07	1.61265E-10	7.91491E-05	7.47352E-06	7.47210E-06	9.14945E-10	1.01536E-12	1.81316E-16	9.90458E-22	2.68698E-21	6.24717E-22
2.51189E-05	1.00000E+00	1.66037E-03	1.12739E-03	2.24619E-07	2.24599E-10	1.39513E-04	1.65888E-05	1.65855E-05	2.25509E-09	2.89026E-12	6.58412E-16	4.57363E-21	9.44461E-21	2.66984E-21
3.16228E-05	1.00000E+00	2.08983E-03	1.54136E-03	3.07388E-07	3.07368E-10	2.40753E-04	3.59500E-05	3.59428E-05	5.34363E-09	8.05818E-12	2.34716E-15	2.07709E-20	3.23284E-20	1.10208E-20
3.98107E-05	1.00000E+00	2.63023E-03	2.07359E-03	4.13821E-07	4.13800E-10	4.06165E-04	7.58298E-05	7.58142E-05	1.21443E-08	2.19413E-11	8.19182E-15	9.25298E-20	1.07276E-19	4.36749E-20
5.01187E-05	1.00000E+00	3.31013E-03	2.75020E-03	5.49137E-07	5.49116E-10	6.69293E-04	1.55215E-04	1.55183E-04	2.64238E-08	5.81811E-11	2.79137E-14	4.03257E-19	3.43468E-19	1.65123E-19
6.30957E-05	1.00000E+00	4.16542E-03	3.60475E-03	7.20046E-07	7.20025E-10	1.07676E-03	3.07430E-04	3.07366E-04	5.49738E-08	1.49835E-10	9.26181E-14	1.71480E-18	1.05595E-18	5.91899E-19
7.94328E-05	1.00000E+00	5.24112E-03	4.68077E-03	9.35250E-07	9.35229E-10	1.69106E-03	5.87623E-04	5.87501E-04	1.09253E-07	3.73806E-10	2.98481E-13	7.09717E-18	3.10271E-18	2.00030E-18
1.00000E-04	1.00000E+00	6.59370E-03	6.03406E-03	1.20591E-06	1.20589E-09	2.59259E-03	1.08110E-03	1.08087E-03	2.07227E-07	9.01246E-10	9.32090E-13	2.85212E-17	8.67550E-18	6.34384E-18
1.25893E-04	1.00000E+00	8.29388E-03	7.73520E-03	1.54614E-06	1.54612E-09	3.88027E-03	1.90973E-03	1.90933E-03	3.74808E-07	2.09525E-09	2.81427E-12	1.11049E-16	2.29971E-17	1.88226E-17
1.58489E-04	1.00000E+00	1.04301E-02	9.87266E-03	1.97363E-06	1.97361E-09	5.67015E-03	3.23218E-03	3.23150E-03	6.45914E-07	4.68762E-09	8.19929E-12	4.18068E-16	5.76262E-17	5.21886E-17
1.99526E-04	1.00000E+00	1.31130E-02	1.25570E-02	2.51051E-06	2.51049E-09	8.09283E-03	5.23389E-03	5.23279E-03	1.06028E-06	1.00764E-08	2.30114E-11	1.51914E-15	1.36269E-16	1.35364E-16
2.51189E-04	1.00000E+00	1.64802E-02	1.59261E-02	3.18432E-06	3.18430E-09	1.12924E-02	8.10854E-03	8.10684E-03	1.65931E-06	2.07938E-08	6.21342E-11	5.32038E-15	3.04050E-16	3.29569E-16
3.16228E-04	1.00000E+00	2.07029E-02	2.01512E-02	4.02934E-06	4.02932E-09	1.54305E-02	1.20417E-02	1.20391E-02	2.48231E-06	4.12079E-08	1.61337E-10	1.79429E-14	6.41243E-16	7.57524E-16
3.98107E-04	1.00000E+00	2.59931E-02	2.54444E-02	5.08800E-06	5.08798E-09	2.06981E-02	1.72135E-02	1.72099E-02	3.56699E-06	7.85649E-08	4.03088E-10	5.82645E-14	1.28315E-15	1.65671E-15
5.01187E-04	1.00000E+00	3.26126E-02	3.20676E-02	6.41264E-06	6.41262E-09	2.73337E-02	2.38281E-02	2.38231E-02	4.95572E-06	1.44585E-07	9.70758E-10	1.82332E-13	2.44997E-15	3.47934E-15
6.30957E-04	1.00000E+00	4.08820E-02	4.03417E-02	8.06747E-06	8.06745E-09	3.56416E-02	3.21544E-02	3.21476E-02	6.70446E-06	2.57988E-07	2.26018E-09	5.50949E-13	4.49408E-15	7.08465E-15
7.94328E-04	1.00000E+00	5.11922E-02	5.06577E-02	1.01307E-05	1.01307E-08	4.60050E-02	4.25549E-02	4.25460E-02	8.88906E-06	4.48533E-07	5.10649E-09	1.61211E-12	7.97623E-15	1.41113E-14
1.00000E-03	1.00000E+00	6.40145E-02	6.34873E-02	1.26966E-05	1.26966E-08	5.88946E-02	5.54936E-02	5.54819E-02	1.16068E-05	7.63373E-07	1.12410E-08	4.58363E-12	1.37862E-14	2.76966E-14
1.25893E-03	9.99999E-01	7.99108E-02	7.93924E-02	1.58777E-05	1.58776E-08	7.48744E-02	7.15345E-02	7.15195E-02	1.49762E-05	1.27691E-06	2.42034E-08	1.27098E-11	2.33303E-14	5.38608E-14
1.58489E-03	9.99999E-01	9.95396E-02	9.90324E-02	1.98057E-05	1.98056E-08	9.46065E-02	9.13421E-02	9.13229E-02	1.91368E-05	2.10593E-06	5.11455E-08	3.44898E-11	3.88184E-14	1.04166E-13
1.99526E-03	9.99999E-01	1.23657E-01	1.23163E-01	2.46318E-05	2.46318E-08	1.18850E-01	1.15679E-01	1.15654E-01	2.42486E-05	3.43259E-06	1.06363E-07	9.18755E-11	6.37014E-14	2.00799E-13
2.51189E-03	9.99999E-01	1.53102E-01	1.52625E-01	3.05241E-05	3.05241E-08	1.48450E-01	1.45392E-01	1.45361E-01	3.04899E-05	5.53903E-06	2.18141E-07	2.40859E-10	1.03329E-13	3.86245E-13

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
3.16228E-03	9.99999E-01	1.88768E-01	1.88311E-01	3.76615E-05	3.76615E-08	1.84304E-01	1.81383E-01	1.81345E-01	3.80498E-05	8.85871E-06	4.41839E-07	6.22622E-10	1.65925E-13	7.41506E-13
3.98107E-03	9.99998E-01	2.31542E-01	2.31109E-01	4.62211E-05	4.62210E-08	2.27302E-01	2.24546E-01	2.24499E-01	4.71162E-05	1.40511E-05	8.84571E-07	1.58920E-09	2.64006E-13	1.42006E-12
5.01187E-03	9.99998E-01	2.82198E-01	2.81794E-01	5.63581E-05	5.63581E-08	2.78225E-01	2.75663E-01	2.75606E-01	5.78534E-05	2.21077E-05	1.75087E-06	4.00835E-09	4.16402E-13	2.71011E-12
6.30957E-03	9.99997E-01	3.41250E-01	3.40879E-01	6.81752E-05	6.81752E-08	3.37587E-01	3.35253E-01	3.35183E-01	7.03702E-05	3.44977E-05	3.42530E-06	9.99305E-09	6.51021E-13	5.14635E-12
7.94328E-03	9.99997E-01	4.08734E-01	4.08401E-01	8.16797E-05	8.16796E-08	4.05426E-01	4.03351E-01	4.03267E-01	8.46742E-05	5.33605E-05	6.61762E-06	2.46179E-08	1.00851E-12	9.70522E-12
1.00000E-02	9.99996E-01	4.83950E-01	4.83659E-01	9.67314E-05	9.67313E-08	4.81037E-01	4.79252E-01	4.79151E-01	1.00617E-04	8.17443E-05	1.26084E-05	5.98784E-08	1.54679E-12	1.81336E-11
1.25893E-02	9.99995E-01	5.65188E-01	5.64943E-01	1.12988E-04	1.12988E-07	5.62703E-01	5.61230E-01	5.61112E-01	1.17837E-04	1.23880E-04	2.36446E-05	1.43607E-07	2.34628E-12	3.34765E-11
1.58489E-02	9.99993E-01	6.49531E-01	6.49334E-01	1.29866E-04	1.29866E-07	6.47489E-01	6.46341E-01	6.46205E-01	1.35714E-04	1.85458E-04	4.35348E-05	3.38965E-07	3.51513E-12	6.08678E-11
1.99526E-02	9.99991E-01	7.32854E-01	7.32703E-01	1.46540E-04	1.46540E-07	7.31249E-01	7.30422E-01	7.30268E-01	1.53375E-04	2.73855E-04	7.84633E-05	7.85558E-07	5.19356E-12	1.08610E-10
2.51189E-02	9.99989E-01	8.10187E-01	8.10080E-01	1.62016E-04	1.62016E-07	8.08990E-01	8.08459E-01	8.08290E-01	1.69767E-04	3.98247E-04	1.37952E-04	1.78242E-06	7.55596E-12	1.89447E-10
3.16228E-02	9.99986E-01	8.76554E-01	8.76484E-01	1.75297E-04	1.75297E-07	8.75705E-01	8.75430E-01	8.75246E-01	1.83834E-04	5.69573E-04	2.35732E-04	3.94690E-06	1.08103E-11	3.21715E-10
3.98107E-02	9.99983E-01	9.28176E-01	9.28136E-01	1.85627E-04	1.85627E-07	9.27599E-01	9.27522E-01	9.27328E-01	1.94776E-04	8.00410E-04	3.90089E-04	8.50053E-06	1.51955E-11	5.29792E-10
5.01187E-02	9.99978E-01	9.63673E-01	9.63653E-01	1.92731E-04	1.92731E-07	9.63282E-01	9.63342E-01	9.63140E-01	2.02300E-04	1.10497E-03	6.23203E-04	1.77481E-05	2.09817E-11	8.43233E-10
6.30957E-02	9.99973E-01	9.84590E-01	9.84582E-01	1.96916E-04	1.96916E-07	9.84310E-01	9.84450E-01	9.84244E-01	2.06734E-04	1.49957E-03	9.59406E-04	3.58231E-05	2.84788E-11	1.29447E-09
7.94328E-02	9.99966E-01	9.94750E-01	9.94747E-01	1.98949E-04	1.98949E-07	9.94523E-01	9.94703E-01	9.94494E-01	2.08887E-04	2.00380E-03	1.42326E-03	6.97734E-05	3.80591E-11	1.91631E-09
1.00000E-01	9.99957E-01	9.98624E-01	9.98623E-01	1.99725E-04	1.99725E-07	9.98417E-01	9.98611E-01	9.98402E-01	2.09708E-04	2.64247E-03	2.03930E-03	1.31079E-04	5.01938E-11	2.74160E-09
1.25893E-01	9.99945E-01	9.99710E-01	9.99710E-01	1.99942E-04	1.99942E-07	9.99509E-01	9.99708E-01	9.99498E-01	2.09939E-04	3.44772E-03	2.83493E-03	2.37780E-04	6.54935E-11	3.80712E-09
1.58489E-01	9.99931E-01	9.99910E-01	9.99910E-01	1.99982E-04	1.99982E-07	9.99710E-01	9.99910E-01	9.99700E-01	2.09981E-04	4.46109E-03	3.84567E-03	4.17515E-04	8.47475E-11	5.16055E-09
1.99526E-01	9.99914E-01	9.99918E-01	9.99918E-01	1.99984E-04	1.99984E-07	9.99718E-01	9.99918E-01	9.99708E-01	2.09983E-04	5.73547E-03	5.12013E-03	7.11894E-04	1.08961E-10	6.86705E-09
2.51189E-01	9.99891E-01	9.99898E-01	9.99898E-01	1.99980E-04	1.99980E-07	9.99698E-01	9.99898E-01	9.99688E-01	2.09978E-04	7.33749E-03	6.72305E-03	1.18252E-03	1.39399E-10	9.01335E-09
3.16228E-01	9.99863E-01	9.99870E-01	9.99870E-01	1.99974E-04	1.99974E-07	9.99670E-01	9.99869E-01	9.99660E-01	2.09973E-04	9.35058E-03	8.73741E-03	1.91875E-03	1.77648E-10	1.17105E-08
3.98107E-01	9.99828E-01	9.99834E-01	9.99834E-01	1.99967E-04	1.99967E-07	9.99634E-01	9.99834E-01	9.99624E-01	2.09965E-04	1.18790E-02	1.12674E-02	3.04691E-03	2.25689E-10	1.50982E-08
5.01187E-01	9.99783E-01	9.99789E-01	9.99789E-01	1.99958E-04	1.99958E-07	9.99590E-01	9.99789E-01	9.99580E-01	2.09956E-04	1.50529E-02	1.44433E-02	4.74010E-03	2.85991E-10	1.93506E-08
6.30957E-01	9.99727E-01	9.99733E-01	9.99733E-01	1.99947E-04	1.99947E-07	9.99533E-01	9.99733E-01	9.99523E-01	2.09944E-04	1.90338E-02	1.84267E-02	7.22669E-03	3.61629E-10	2.46844E-08
7.94328E-01	9.99656E-01	9.99663E-01	9.99663E-01	1.99933E-04	1.99933E-07	9.99463E-01	9.99662E-01	9.99453E-01	2.09929E-04	2.40223E-02	2.34183E-02	1.07951E-02	4.56410E-10	3.13681E-08
1.00000E+00	9.99567E-01	9.99574E-01	9.99574E-01	1.99915E-04	1.99915E-07	9.99374E-01	9.99573E-01	9.99364E-01	2.09910E-04	3.02658E-02	2.96658E-02	1.57919E-02	5.75038E-10	3.97334E-08
1.25893E+00	9.99455E-01	9.99461E-01	9.99461E-01	1.99892E-04	1.99892E-07	9.99262E-01	9.99461E-01	9.99252E-01	2.09887E-04	3.80684E-02	3.74732E-02	2.26128E-02	7.23287E-10	5.01875E-08
1.58489E+00	9.99314E-01	9.99320E-01	9.99320E-01	1.99864E-04	1.99864E-07	9.99120E-01	9.99320E-01	9.99111E-01	2.09857E-04	4.78007E-02	4.72117E-02	3.16891E-02	9.08201E-10	6.32271E-08

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
1.99526E+00	9.99136E-01	9.99143E-01	9.99143E-01	1.99829E-04	1.99829E-07	9.98943E-01	9.99143E-01	9.98933E-01	2.09820E-04	5.99110E-02	5.93295E-02	4.34805E-02	1.13830E-09	7.94528E-08
2.51189E+00	9.98913E-01	9.98919E-01	9.98919E-01	1.99784E-04	1.99784E-07	9.98719E-01	9.98919E-01	9.98709E-01	2.09773E-04	7.49347E-02	7.43627E-02	5.84833E-02	1.42375E-09	9.95821E-08
3.16228E+00	9.98631E-01	9.98638E-01	9.98638E-01	1.99728E-04	1.99728E-07	9.98438E-01	9.98638E-01	9.98428E-01	2.09714E-04	9.35022E-02	9.29418E-02	7.72593E-02	1.77653E-09	1.24459E-07
3.98107E+00	9.98277E-01	9.98284E-01	9.98284E-01	1.99657E-04	1.99657E-07	9.98084E-01	9.98284E-01	9.98074E-01	2.09640E-04	1.16339E-01	1.15794E-01	1.00470E-01	2.21044E-09	1.55057E-07
5.01187E+00	9.97832E-01	9.97838E-01	9.97838E-01	1.99568E-04	1.99568E-07	9.97639E-01	9.97838E-01	9.97629E-01	2.09546E-04	1.44260E-01	1.43731E-01	1.28890E-01	2.74092E-09	1.92466E-07
6.30957E+00	9.97271E-01	9.97277E-01	9.97277E-01	1.99455E-04	1.99455E-07	9.97078E-01	9.97277E-01	9.97068E-01	2.09428E-04	1.78135E-01	1.77628E-01	1.63383E-01	3.38456E-09	2.37853E-07
7.94328E+00	9.96565E-01	9.96572E-01	9.96572E-01	1.99314E-04	1.99314E-07	9.96373E-01	9.96572E-01	9.96363E-01	2.09280E-04	2.18845E-01	2.18364E-01	2.04839E-01	4.15805E-09	2.92398E-07
1.00000E+01	9.95678E-01	9.95685E-01	9.95685E-01	1.99137E-04	1.99137E-07	9.95486E-01	9.95684E-01	9.95476E-01	2.09094E-04	2.67180E-01	2.66729E-01	2.54060E-01	5.07642E-09	3.57159E-07
1.25893E+01	9.94562E-01	9.94569E-01	9.94569E-01	1.98914E-04	1.98914E-07	9.94370E-01	9.94568E-01	9.94360E-01	2.08859E-04	3.23702E-01	3.23287E-01	3.11620E-01	6.15033E-09	4.32889E-07
1.58489E+01	9.93159E-01	9.93165E-01	9.93165E-01	1.98633E-04	1.98633E-07	9.92967E-01	9.93165E-01	9.92957E-01	2.08565E-04	3.88539E-01	3.88165E-01	3.77649E-01	7.38223E-09	5.19759E-07
1.99526E+01	9.91395E-01	9.91402E-01	9.91402E-01	1.98280E-04	1.98280E-07	9.91204E-01	9.91402E-01	9.91194E-01	2.08194E-04	4.61128E-01	4.60800E-01	4.51577E-01	8.76143E-09	6.17017E-07
2.51189E+01	9.89179E-01	9.89186E-01	9.89186E-01	1.97837E-04	1.97837E-07	9.88988E-01	9.89186E-01	9.88978E-01	2.07729E-04	5.39937E-01	5.39659E-01	5.31844E-01	1.02588E-08	7.22608E-07
3.16228E+01	9.86397E-01	9.86403E-01	9.86403E-01	1.97281E-04	1.97281E-07	9.86206E-01	9.86403E-01	9.86196E-01	2.07145E-04	6.22219E-01	6.21994E-01	6.15658E-01	1.18222E-08	8.32854E-07
3.98107E+01	9.82905E-01	9.82911E-01	9.82911E-01	1.96582E-04	1.96582E-07	9.82715E-01	9.82911E-01	9.82705E-01	2.06411E-04	7.03949E-01	7.03776E-01	6.98921E-01	1.33750E-08	9.42359E-07
5.01187E+01	9.78526E-01	9.78533E-01	9.78533E-01	1.95707E-04	1.95707E-07	9.78337E-01	9.78533E-01	9.78327E-01	2.05492E-04	7.80067E-01	7.79945E-01	7.76489E-01	1.48213E-08	1.04435E-06
6.30957E+01	9.73042E-01	9.73048E-01	9.73048E-01	1.94610E-04	1.94610E-07	9.72853E-01	9.73048E-01	9.72844E-01	2.04340E-04	8.45198E-01	8.45118E-01	8.42890E-01	1.60588E-08	1.13161E-06
7.94328E+01	9.66181E-01	9.66187E-01	9.66187E-01	1.93237E-04	1.93237E-07	9.65994E-01	9.66187E-01	9.65984E-01	2.02899E-04	8.94785E-01	8.94741E-01	8.93494E-01	1.70009E-08	1.19806E-06
1.00000E+02	9.57612E-01	9.57618E-01	9.57618E-01	1.91524E-04	1.91524E-07	9.57427E-01	9.57618E-01	9.57417E-01	2.01100E-04	9.26365E-01	9.26346E-01	9.25797E-01	1.76009E-08	1.24038E-06
1.25893E+02	9.46933E-01	9.46939E-01	9.46939E-01	1.89388E-04	1.89388E-07	9.46750E-01	9.46939E-01	9.46740E-01	1.98857E-04	9.40331E-01	9.40327E-01	9.40207E-01	1.78663E-08	1.25910E-06
1.58489E+02	9.33657E-01	9.33664E-01	9.33664E-01	1.86733E-04	1.86733E-07	9.33477E-01	9.33663E-01	9.33468E-01	1.96069E-04	9.39596E-01	9.39599E-01	9.39697E-01	1.78523E-08	1.25812E-06
1.99526E+02	9.17209E-01	9.17215E-01	9.17215E-01	1.83443E-04	1.83443E-07	9.17032E-01	9.17215E-01	9.17023E-01	1.92615E-04	9.28119E-01	9.28126E-01	9.28311E-01	1.76343E-08	1.24276E-06
2.51189E+02	8.96913E-01	8.96919E-01	8.96919E-01	1.79384E-04	1.79384E-07	8.96740E-01	8.96919E-01	8.96731E-01	1.88353E-04	9.09140E-01	9.09148E-01	9.09355E-01	1.72737E-08	1.21735E-06
3.16228E+02	8.72000E-01	8.72005E-01	8.72005E-01	1.74401E-04	1.74401E-07	8.71831E-01	8.72005E-01	8.71822E-01	1.83121E-04	8.84222E-01	8.84230E-01	8.84437E-01	1.68002E-08	1.18398E-06
3.98107E+02	8.41617E-01	8.41623E-01	8.41623E-01	1.68325E-04	1.68325E-07	8.41454E-01	8.41622E-01	8.41446E-01	1.76741E-04	8.53460E-01	8.53468E-01	8.53669E-01	1.62157E-08	1.14279E-06
5.01187E+02	8.04868E-01	8.04873E-01	8.04873E-01	1.60975E-04	1.60975E-07	8.04713E-01	8.04873E-01	8.04705E-01	1.69023E-04	8.16198E-01	8.16205E-01	8.16397E-01	1.55078E-08	1.09290E-06
6.30957E+02	7.60877E-01	7.60882E-01	7.60882E-01	1.52176E-04	1.52176E-07	7.60730E-01	7.60882E-01	7.60722E-01	1.59785E-04	7.71588E-01	7.71595E-01	7.71776E-01	1.46602E-08	1.03316E-06
7.94328E+02	7.08898E-01	7.08903E-01	7.08903E-01	1.41781E-04	1.41781E-07	7.08761E-01	7.08903E-01	7.08754E-01	1.48870E-04	7.18877E-01	7.18883E-01	7.19053E-01	1.36587E-08	9.62585E-07
1.00000E+03	6.48479E-01	6.48484E-01	6.48484E-01	1.29697E-04	1.29697E-07	6.48354E-01	6.48484E-01	6.48348E-01	1.36182E-04	6.57608E-01	6.57614E-01	6.57769E-01	1.24946E-08	8.80545E-07

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
1.25893E+03	5.79684E-01	5.79688E-01	5.79688E-01	1.15938E-04	1.15938E-07	5.79572E-01	5.79688E-01	5.79566E-01	1.21734E-04	5.87844E-01	5.87849E-01	5.87988E-01	1.11690E-08	7.87130E-07
1.58489E+03	5.03356E-01	5.03360E-01	5.03360E-01	1.00672E-04	1.00672E-07	5.03259E-01	5.03360E-01	5.03254E-01	1.05706E-04	5.10442E-01	5.10446E-01	5.10567E-01	9.69840E-09	6.83488E-07
1.99526E+03	4.21389E-01	4.21392E-01	4.21392E-01	8.42784E-05	8.42784E-08	4.21308E-01	4.21392E-01	4.21304E-01	8.84923E-05	4.27321E-01	4.27325E-01	4.27426E-01	8.11910E-09	5.72188E-07
2.51189E+03	3.36903E-01	3.36905E-01	3.36905E-01	6.73810E-05	6.73810E-08	3.36838E-01	3.36905E-01	3.36834E-01	7.07500E-05	3.41645E-01	3.41648E-01	3.41729E-01	6.49126E-09	4.57467E-07
3.16228E+03	2.54193E-01	2.54195E-01	2.54195E-01	5.08389E-05	5.08389E-08	2.54144E-01	2.54195E-01	2.54141E-01	5.33809E-05	2.57771E-01	2.57773E-01	2.57834E-01	4.89765E-09	3.45159E-07
3.98107E+03	1.78298E-01	1.78299E-01	1.78299E-01	3.56597E-05	3.56597E-08	1.78263E-01	1.78299E-01	1.78261E-01	3.74427E-05	1.80807E-01	1.80809E-01	1.80852E-01	3.43534E-09	2.42103E-07
5.01187E+03	1.14090E-01	1.14091E-01	1.14091E-01	2.28182E-05	2.28182E-08	1.14068E-01	1.14091E-01	1.14067E-01	2.39591E-05	1.15696E-01	1.15697E-01	1.15724E-01	2.19823E-09	1.54918E-07
6.30957E+03	6.50347E-02	6.50351E-02	6.50351E-02	1.30070E-05	1.30070E-08	6.50221E-02	6.50351E-02	6.50215E-02	1.36574E-05	6.59502E-02	6.59508E-02	6.59663E-02	1.25305E-09	8.83081E-08
7.94328E+03	3.20508E-02	3.20510E-02	3.20510E-02	6.41020E-06	6.41020E-09	3.20446E-02	3.20510E-02	3.20442E-02	6.73070E-06	3.25019E-02	3.25022E-02	3.25099E-02	6.17537E-10	4.35205E-08
1.00000E+04	1.31511E-02	1.31512E-02	1.31512E-02	2.63024E-06	2.63024E-09	1.31486E-02	1.31512E-02	1.31484E-02	2.76175E-06	1.33362E-02	1.33363E-02	1.33395E-02	2.53388E-10	1.78574E-08
1.25893E+04	4.28463E-03	4.28466E-03	4.28466E-03	8.56932E-07	8.56932E-10	4.28380E-03	4.28466E-03	4.28376E-03	8.99778E-07	4.34495E-03	4.34498E-03	4.34601E-03	8.25540E-11	5.81793E-09
1.58489E+04	1.04413E-03	1.04414E-03	1.04414E-03	2.08828E-07	2.08828E-10	1.04393E-03	1.04414E-03	1.04392E-03	2.19269E-07	1.05883E-03	1.05884E-03	1.05909E-03	2.01178E-11	1.41779E-09
1.99526E+04	1.76537E-04	1.76538E-04	1.76538E-04	3.53076E-08	3.53076E-11	1.76503E-04	1.76538E-04	1.76501E-04	3.70730E-08	1.79022E-04	1.79024E-04	1.79066E-04	3.40142E-12	2.39713E-10
2.51189E+04	1.88385E-05	1.88386E-05	1.88386E-05	3.76772E-09	3.76772E-12	1.88349E-05	1.88386E-05	1.88347E-05	3.95611E-09	1.91037E-05	1.91038E-05	1.91084E-05	3.62970E-13	2.55800E-11
3.16228E+04	1.12625E-06	1.12626E-06	1.12626E-06	2.25252E-10	2.25252E-13	1.12603E-06	1.12626E-06	1.12602E-06	2.36514E-10	1.14211E-06	1.14212E-06	1.14239E-06	2.17000E-14	1.52929E-12
3.98107E+04	3.24678E-08	3.24680E-08	3.24680E-08	6.49360E-12	6.49360E-15	3.24615E-08	3.24680E-08	3.24612E-08	6.81828E-12	3.29249E-08	3.29251E-08	3.29329E-08	6.25572E-16	4.40868E-14
5.01187E+04	3.73661E-10	3.73663E-10	3.73663E-10	7.47327E-14	7.47327E-17	3.73589E-10	3.73663E-10	3.73585E-10	7.84693E-14	3.78921E-10	3.78924E-10	3.79014E-10	7.19950E-18	5.07380E-16
6.30957E+04	1.35348E-12	1.35349E-12	1.35349E-12	2.70698E-16	2.70698E-19	1.35322E-12	1.35349E-12	1.35321E-12	2.84233E-16	1.37254E-12	1.37255E-12	1.37287E-12	2.60782E-20	1.83784E-18
7.94328E+04	1.14389E-15	1.14390E-15	1.14390E-15	2.28779E-19	2.28779E-22	1.14367E-15	1.14390E-15	1.14366E-15	2.40218E-19	1.15999E-15	1.16000E-15	1.16028E-15	2.20399E-23	1.55324E-21
1.00000E+05	1.54746E-19	1.54747E-19	1.54747E-19	3.09495E-23	3.09495E-26	1.54716E-19	1.54747E-19	1.54715E-19	3.24969E-23	1.56925E-19	1.56926E-19	1.56963E-19	2.98157E-27	2.10124E-25
1.25893E+05	2.08514E-24	2.08515E-24	2.08515E-24	4.17030E-28	4.17030E-31	2.08474E-24	2.08515E-24	2.08471E-24	4.37882E-28	2.11449E-24	2.11451E-24	2.11501E-24	4.01753E-32	2.83133E-30
1.58489E+05	1.54013E-30	1.54014E-30	1.54014E-30	3.08028E-34	3.08028E-37	1.53983E-30	1.54014E-30	1.53982E-30	3.23429E-34	1.56181E-30	1.56182E-30	1.56219E-30	2.96744E-38	2.09128E-36
1.99526E+05	2.94007E-38	2.94009E-38	2.94009E-38	5.88019E-42	5.88019E-45	2.93950E-38	2.94009E-38	2.93948E-38	6.17419E-42	2.98146E-38	2.98149E-38	2.98219E-38	5.66478E-46	3.99221E-44
2.51189E+05	5.62940E-48	5.62944E-48	5.62944E-48	1.12589E-51	1.12589E-54	5.62831E-48	5.62944E-48	5.62826E-48	1.18218E-51	5.70865E-48	5.70870E-48	5.71004E-48	1.08464E-55	7.64394E-54
3.16228E+05	3.28365E-60	3.28367E-60	3.28367E-60	6.56734E-64	6.56734E-67	3.28301E-60	3.28367E-60	3.28298E-60	6.89570E-64	3.32987E-60	3.32990E-60	3.33068E-60	6.32675E-68	4.45873E-66
3.98107E+05	1.30176E-75	1.30176E-75	1.30176E-75	2.60353E-79	2.60353E-82	1.30150E-75	1.30176E-75	1.30149E-75	2.73370E-79	1.32008E-75	1.32009E-75	1.32040E-75	2.50815E-83	1.76760E-81
5.01187E+05	5.30613E-95	5.30617E-95	5.30617E-95	1.06123E-98	0.00000E+00	5.30511E-95	5.30617E-95	5.30505E-95	1.11430E-98	5.38083E-95	5.38087E-95	5.38214E-95	0.00000E+00	0.00000E+00
6.30957E+05	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
7.94328E+05	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+06	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+07	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Time (year)	Ra-226	Rn-222	Po-218	At-218	Rn-218	Pb-214	Bi-214	Po-214	Tl-210	Pb-210	Bi-210	Po-210	Hg-206	Tl-206
5.01187E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+08	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.25893E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.58489E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.99526E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2.51189E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.16228E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3.98107E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5.01187E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6.30957E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7.94328E+09	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.00000E+10	0	0	0	0	0	0	0	0	0	0	0	0	0	0