

Introduction to ORIGIN

SCALE 6.3

ORNL is managed by UT-Battelle, LLC for the US Department of Energy



U.S. DEPARTMENT OF
ENERGY

Learning goals



Gain insight into ORIGIN's purpose and main capabilities



Become familiarized with generating and running ORIGIN input for a simple decay problem using Fulcrum

What is ORIGIN?



Overview of ORIGIN

- Oak Ridge Isotope Generation code in SCALE
- Irradiation and decay simulation code
- Explicit simulation of 2,237 nuclides in database
 - 176 actinides
 - 1,151 fission products
 - 910 structural activation nuclides
- Explicit simulation of all pathways from neutron transmutation, fission, and decay
 - Includes all nuclides with half-lives > 1 ms
 - Accurately represents the evaluated nuclear data
 - Many physics codes track a small subset of isotopes

ORIGEN's purpose:

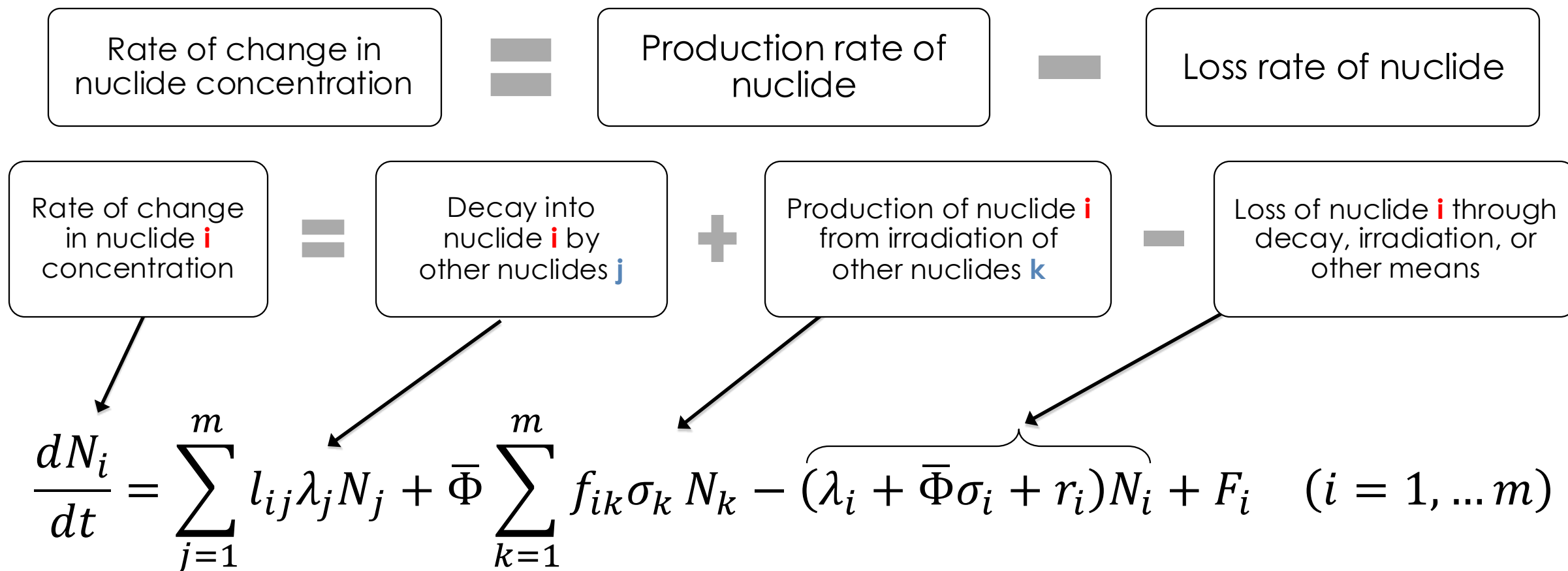
Solve for the nuclide inventory as a function of time: $N_i(t)$

- From the fundamental nuclide inventory result, it is a “unit conversion” to produce decay heat, activity, mass, etc.
- **Applications**
 - **Spent nuclear fuel characterization**
 - Nuclide concentrations (different units)
 - Activities
 - Decay heat
 - Radiation emission rates and spectra (neutron and gamma)
 - **Source terms for accident analyses** (operating reactors, spent fuel handling, storage)
 - **Structural material activation** (in-core, ex-core)
 - **Fuel cycle analysis** (material feed and removal processing)

ORIGEN enables a comprehensive isotopic characterization of fuel over a large time scale, including repository analysis.

ORIGEN is a high-performance ODE solver

General ordinary differential equation (ODE) solved for production and loss rate of a nuclide



Representation of ORIGEN **input** data

The transmutation equation can be written in matrix form:

$$\frac{d\mathbf{N}}{dt} = \overline{\overline{\mathbf{A}}} \cdot \mathbf{N}(t) = (\overline{\lambda} + \phi(t)\overline{\sigma}(t))\mathbf{N}(t)$$

ORIGEN (the =origen sequence) solves for a single material/mixture; the user is responsible for defining the following data:

$\mathbf{N}(0)$ the **initial nuclide vector**

$\overline{\overline{\mathbf{A}}}(t)$ the **transition rate matrix** (stepwise constant in time)

which has components:

$\overline{\lambda}$ the **decay matrix** (constant in time)

$\overline{\sigma}(t)$ the **neutron reaction matrix** (stepwise constant in time, changing as the flux spectrum changes)

$\phi(t)$ the **flux magnitude** (stepwise constant, changing as the flux/power level changes)

Matrix exponential method (MATREX solver)

- The transmutation equation can be written in matrix form (for the homogeneous equation):

$$\dot{\mathbf{N}}(t) = \bar{\bar{\mathbf{A}}} \cdot \mathbf{N}(t) \quad \text{where } \dot{\mathbf{N}}(t) = \frac{d\mathbf{N}}{dt}$$

- Solution for the nuclide vector has the form

$$\mathbf{N}(t) = \mathbf{N}(0) \cdot \exp[\bar{\bar{\mathbf{A}}} t]$$

where

$\mathbf{N}(0)$ = vector of initial atom densities

$\bar{\bar{\mathbf{A}}}$ = transition matrix containing rate coefficients (s^{-1}) for radioactive decay, neutron capture, fission, etc.

$$\bar{\bar{\mathbf{A}}} = \boldsymbol{\lambda} + \bar{\Phi} \boldsymbol{\sigma}$$

Matrix exponential solution (MATREX solver)

- Solves the coupled differential equations that result from tracking the matrix of $> 2,200$ nuclides
- Method: extremely fast for large systems – it solves the matrix in $\sim 10^{-2}$ seconds
- Requirements
 - Initial nuclide concentrations (user input)
 - Neutron flux (user input, derived from input power)
 - Irradiation time (user input)
 - Nuclear decay data (decay constants λ , etc.)
 - Neutron cross sections (probabilities of interaction)

Hybrid matrix exponential/linear chains (MATREX solver)

- Exponential function $\exp[\bar{\bar{A}} t]$ is represented as the series expansion

$$e^{[\bar{\bar{A}} t]} = \sum_{m=0}^{\infty} \frac{(\bar{\bar{A}} t)^m}{m!}$$

- Short lived transitions are difficult to solve due to round off
e.g. $e^{-100} = 1 - 100 + 100^2/2! - 100^3/3! + \dots = 3.72 \times 10^{-44}$
- To prevent loss of numerical accuracy, short-lived nuclides are removed from the matrix, matrix elements are adjusted, and matrix is compressed
- Removed nuclides are solved outside the matrix exponential, using Bateman chains solved by a Gauss-Seidel iterative technique

$$N_i = N_i(0)e^{-d_i t} + \sum_{k=1}^{i-1} N_k(0) \prod_{n=k}^{i-1} \left(\frac{a_{n+1,n}}{d_n} \right) \left[\sum_{j=k}^{i-1} d_j \left(\frac{e^{-d_j t} - e^{-d_i t}}{d_i - d_j} \right) \right] \prod_{\substack{n=k \\ n \neq j}}^{i-1} \left(\frac{d_n}{d_n - d_j} \right)$$

Chebyshev rational approximation method (CRAM solver)

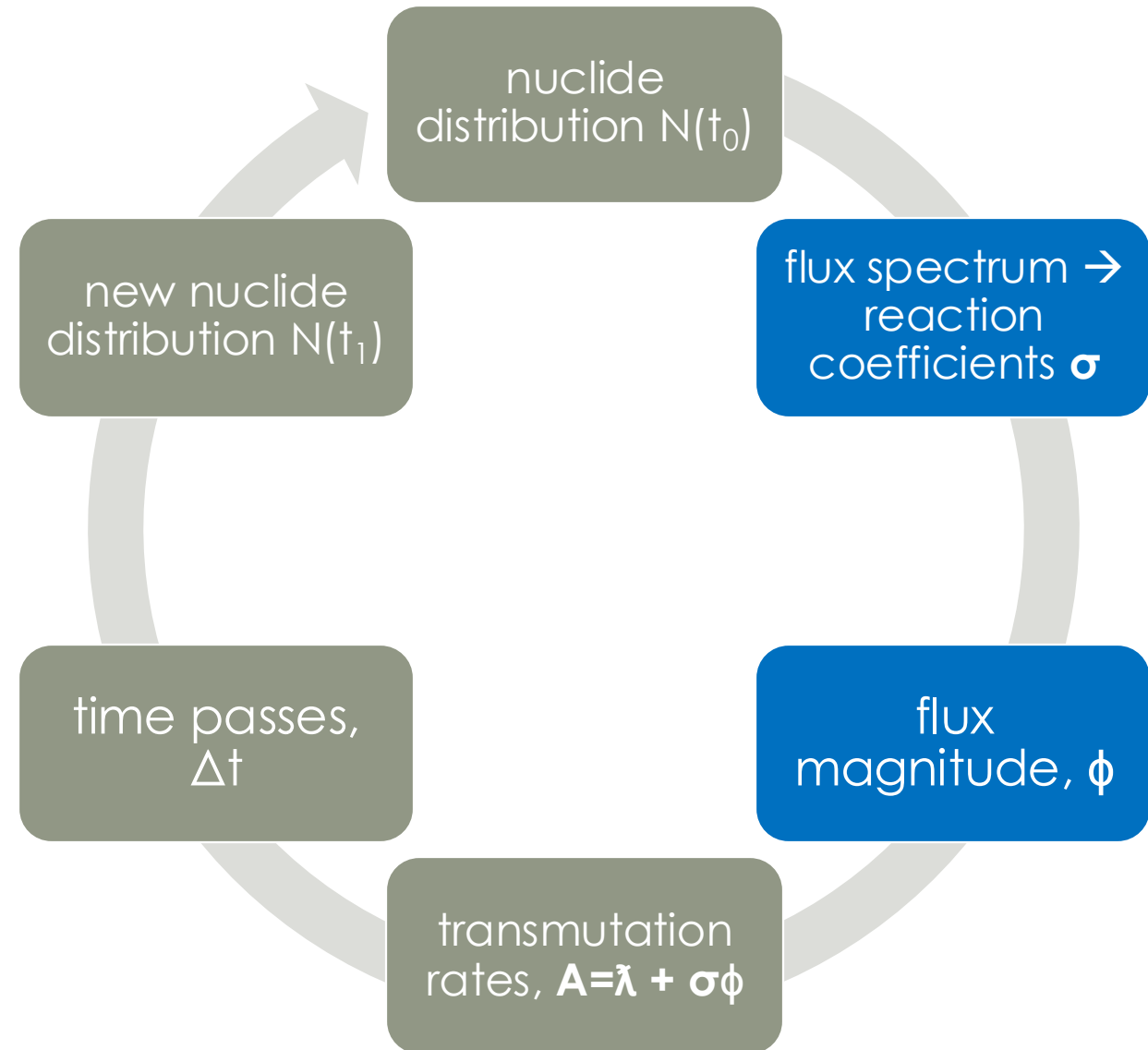
- Matrix exponential method based on Chebyshev rational approximation of the exponential function

$$e^{\bar{\bar{A}} t} = a_0 + 2 \mathbf{Re} \left[\sum_{i=1}^{k/2} a_i (\bar{\bar{A}} t + \theta_i I)^{-1} \right]$$

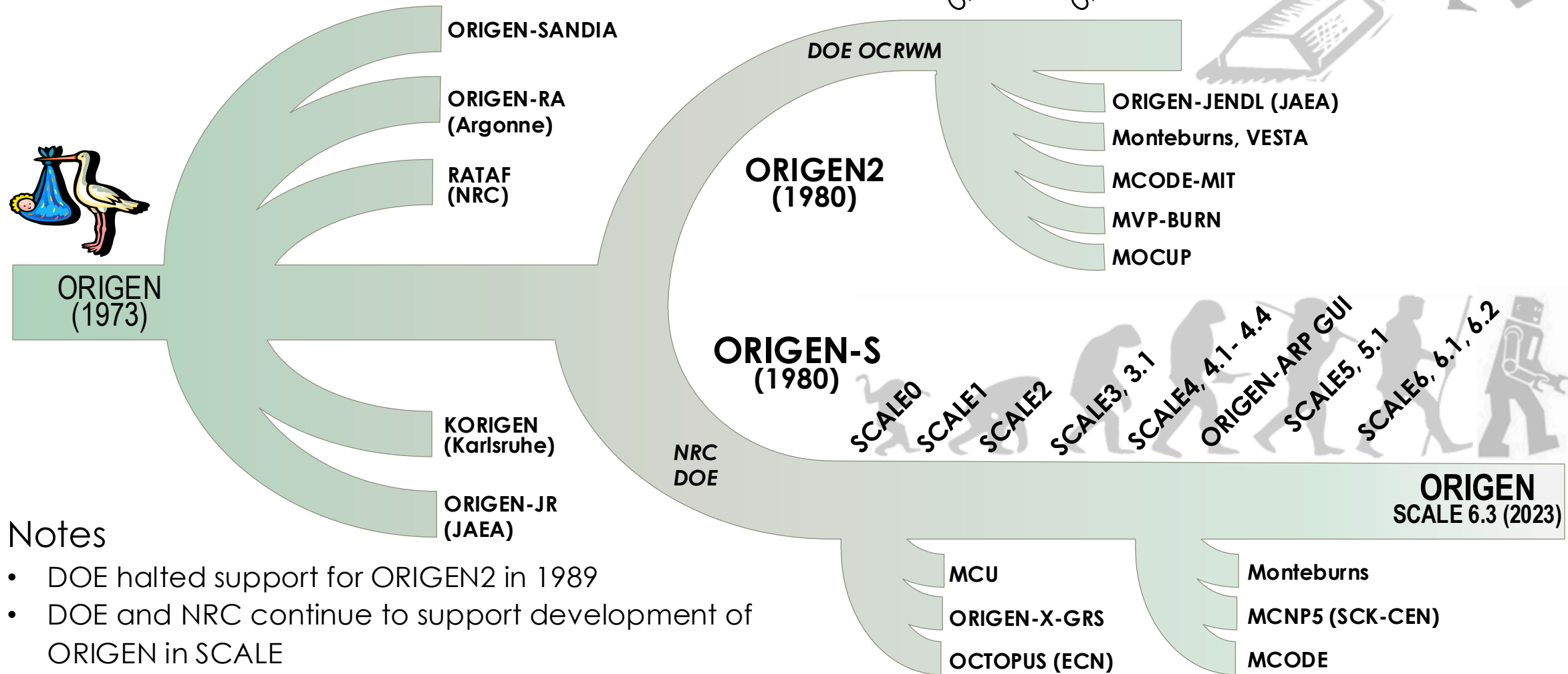
- Fast and accurate, able to handle large systems of nuclides
- Overall accuracy almost independent of step lengths
- Adjoint solution implemented ([*Annals of Nuclear Energy* 85, p.68, 2015](#))

Key to accurate simulations is accurate input data

- Little numerical error in the solvers
- **Decay data λ** are well known and constant in time
- **Reaction coefficient data σ** depends on:
 - Nuclear cross section and yield data
 - Fuel type
 - Enrichment
 - Burnup
 - Assembly design
 - Fuel temperatures
 - Moderator properties
 - Control rod/blade exposure, etc.
- **Flux magnitude data ϕ** depend on operating history



On the ORIGIN species



Comparison of ORIGIN branches

	ORIGIN (1982–present)	ORIGIN2 (1980-1992)
Modern nuclear data	✓	✗
Neutron spectra	✓	✗
User-specified gamma and neutron energy groups	✓	✗
Active development	✓	✗
QA configuration control	✓	✗
Validation	Extensive validation	None in over 20 years

Evolution of ORIGIN capabilities

- **SCALE 5.1 (2006)**
 - Included graphical user interface, new ORIGIN libraries for different reactor types
- **SCALE 6.0 (2009)**
 - Included new base cross section libraries (ENDF/B-VI) and updated gamma ray libraries; retained all previous reactor libraries
- **SCALE 6.1 (2011)**
 - Included ENDF/B-VII decay data, expanded nuclides, energy-dependent fission yields
 - Replaced the old 3-group libraries with 238-group data from JEFF/A-3.0
- **SCALE 6.2 (2016)**
 - Completely rewritten modular ORIGIN source code, dynamic memory allocations
 - New CRAM solver; rewritten, streamlined input format
 - Integrated alpha and beta sources and spectra
 - ORIGIN API to allow embedding depletion calculation in other codes
- **SCALE 6.3 (2023)**
 - Modernized library construction integrated into the ORIGIN sequence instead of externally via COUPLE
 - Sensitivity analysis capability
 - OBIWAN command utility for viewing and manipulating ORIGIN binary files

ORIGEN Documentation

- Section 5.1 of the SCALE 6.3 manual
- Examples show some of the commonly used capabilities

<https://scale-manual.ornl.gov/origen/origen-examples.html>

▣ 5.1. ORIGEN: Neutron activation, transmutation, fission product generation, & radiation source term calculation

5.1.1. Acknowledgements

5.1.2. Version Information

5.1.3. Method of solution

5.1.4. ORIGEN Family of Modules

5.1.5. ORIGEN

5.1.6. ARP

5.1.7. OPUS

5.1.8. Examples

⊕ 5.1.9. Appendices

5.2. OBIWAN: A tool for viewing and manipulating Origen binary files

5.3. ORIGEN Reactor Libraries

5.4. ORIGAMI: A Code for Computing Assembly Isotopics with ORIGEN

<https://scale-manual.ornl.gov/origen/>

References on ORIGEN methods and data

I. C. Gauld, G. Radulescu, G. Ilas, B. D. Murphy, M. L. Williams, D. Wiarda, [*Isotopic Depletion and Decay Methods and Analysis Capabilities in SCALE*](#), Nuclear Technology , 174, 2, 169 (2011).

A. Isotalo and W. Wieselquist, [*A Method for Including External Feed in Depletion Calculations with CRAM and Implementation into ORIGEN*](#), Annals of Nuclear Energy, vol. 85, p.68-77 (2015).

M. T. Pigni, M. W. Francis, and I. C. Gauld, [*Investigation of inconsistent ENDF/B-VII.1 independent and cumulative fission product yields with proposed revisions*](#), Nuclear Data Sheets 123, 231 (2015).

I. C. Gauld, M. T. Pigni, and G. Ilas, [*Validation and testing of ENDF/B-VII decay data*](#), Nuclear Data Sheets 120, p.33-36 (2014).

G. Ilas, J. R. Burns, B. D. Hiscox, and U. Mertzyurek, [*SCALE 6.2.4 Validation: Reactor Physics*](#), ORNL/TM-2020/1500v3, UT-Battelle, LLC, Oak Ridge National Laboratory (November 2022). Web, doi:10.2172/1902818.

G. Ilas and B. Hiscox, "[*Validation of SCALE 6.2.4 and ENDF/B-VII.1 data libraries for nuclide inventory analysis in PWR used fuel*](#)", Transactions of the American Nuclear Society, vol. 124, p.552-554 (2021). <https://www.ans.org/pubs/transactions/article-49653>

G. Ilas and J. Burns, "[*SCALE 6.2.4 Validation for Light Water Reactor Decay Heat Analysis*](#)", Nuclear Technology (2021). <https://doi.org/10.1080/00295450.2021.1935165>

G. Ilas, I. C. Gauld, and H. Liljenfeldt, [*Validation of ORIGEN for LWR used fuel decay heat analysis with SCALE*](#), Nuclear Engineering and Design, vol.273, p. 58-67 (2014)

G. Ilas, I. C. Gauld, and G. Radulescu, [*Validation of new depletion capabilities and ENDF/B-VII data libraries in SCALE*](#), Annals of Nuclear Energy, vol. 46, p.43-55 (2012).

Publicly-available documents can be accessed via OSTI. See <https://www.ornl.gov/scale/references> Depletion and Decay

ORIGEN decay calculations

- Hands-on introduction to ORIGEN decay input data and input generation and validation using Fulcrum



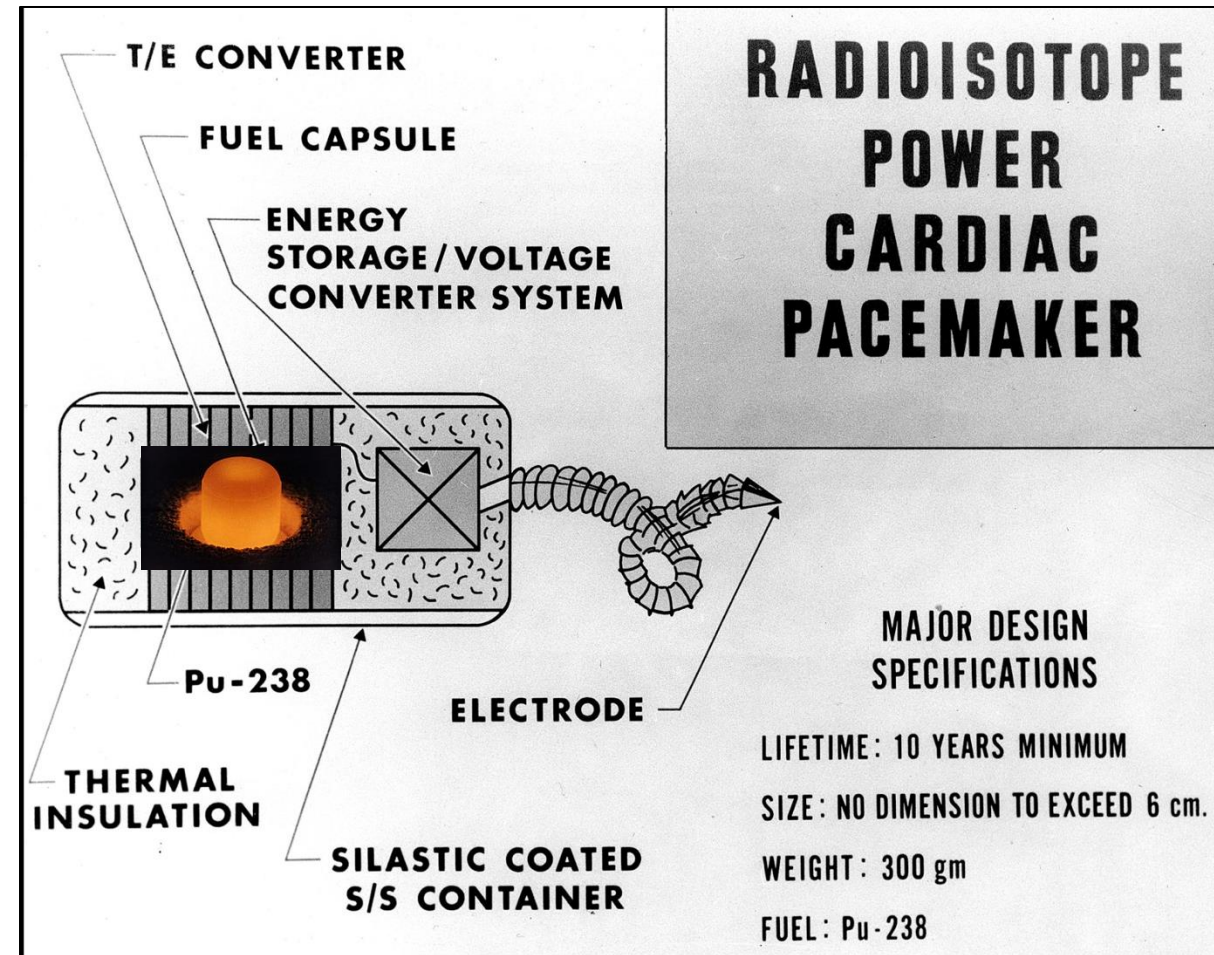
A simple decay heat calculation

- **Problem**

- Determine the 100-year decay heat curve for a radioisotope-powered pacemaker powered by 1 mg ^{238}Pu

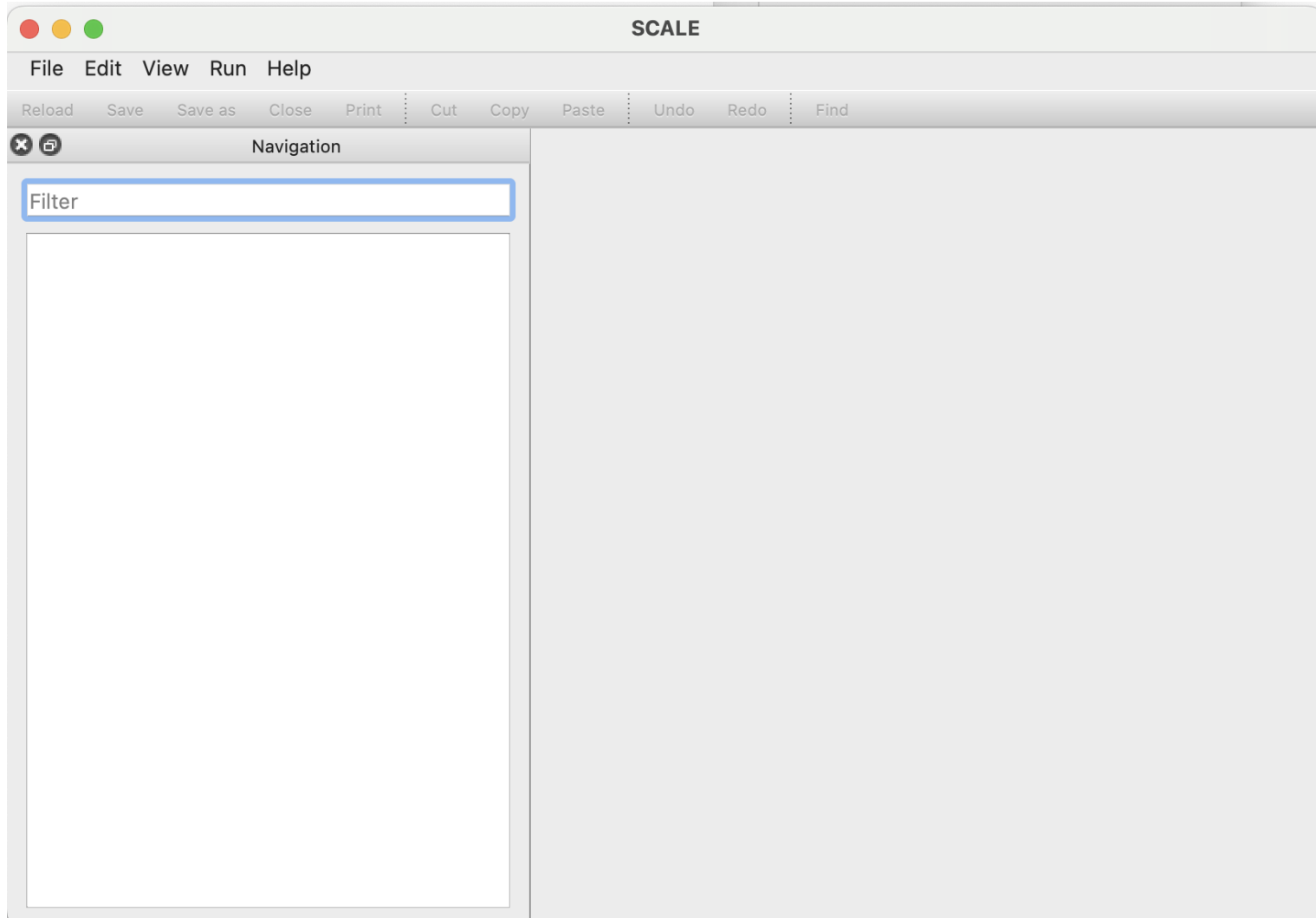
- **Approach**

- Clean slate: start with an empty file and use the autocomplete capability in Fulcrum
- Name your input “**i_heart_pu238.inp**”
- Include **save=yes**
- Open associated F71 file in Fulcrum



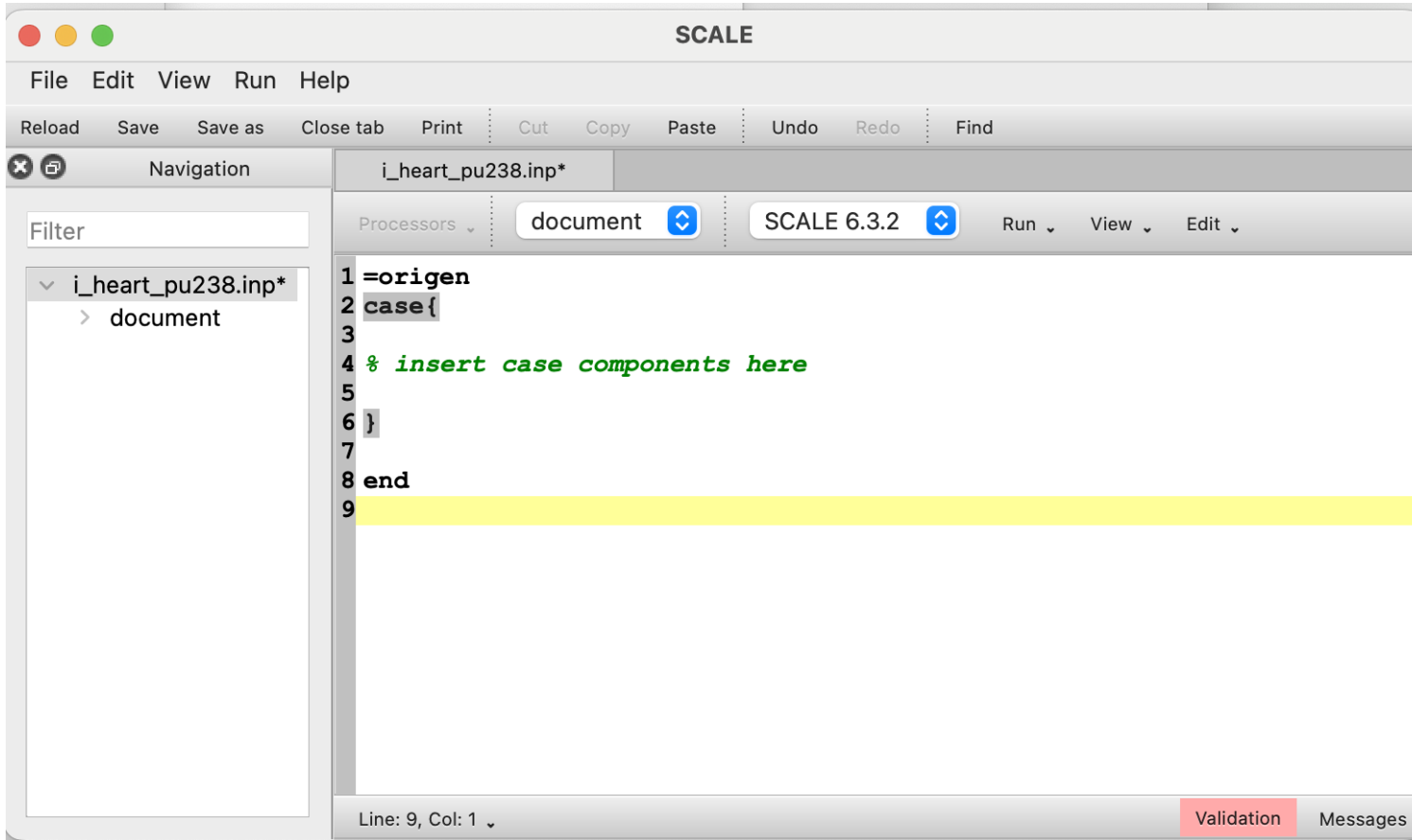
Radioisotope-powered cardiac pacemaker developed by the Atomic Energy Commission, circa 1967.

Simple decay calculation: Open Fulcrum



- Shortcut to Fulcrum is usually in the SCALE folder on Desktop (on Windows computer)

Simple decay calculation: Construct a **case** block



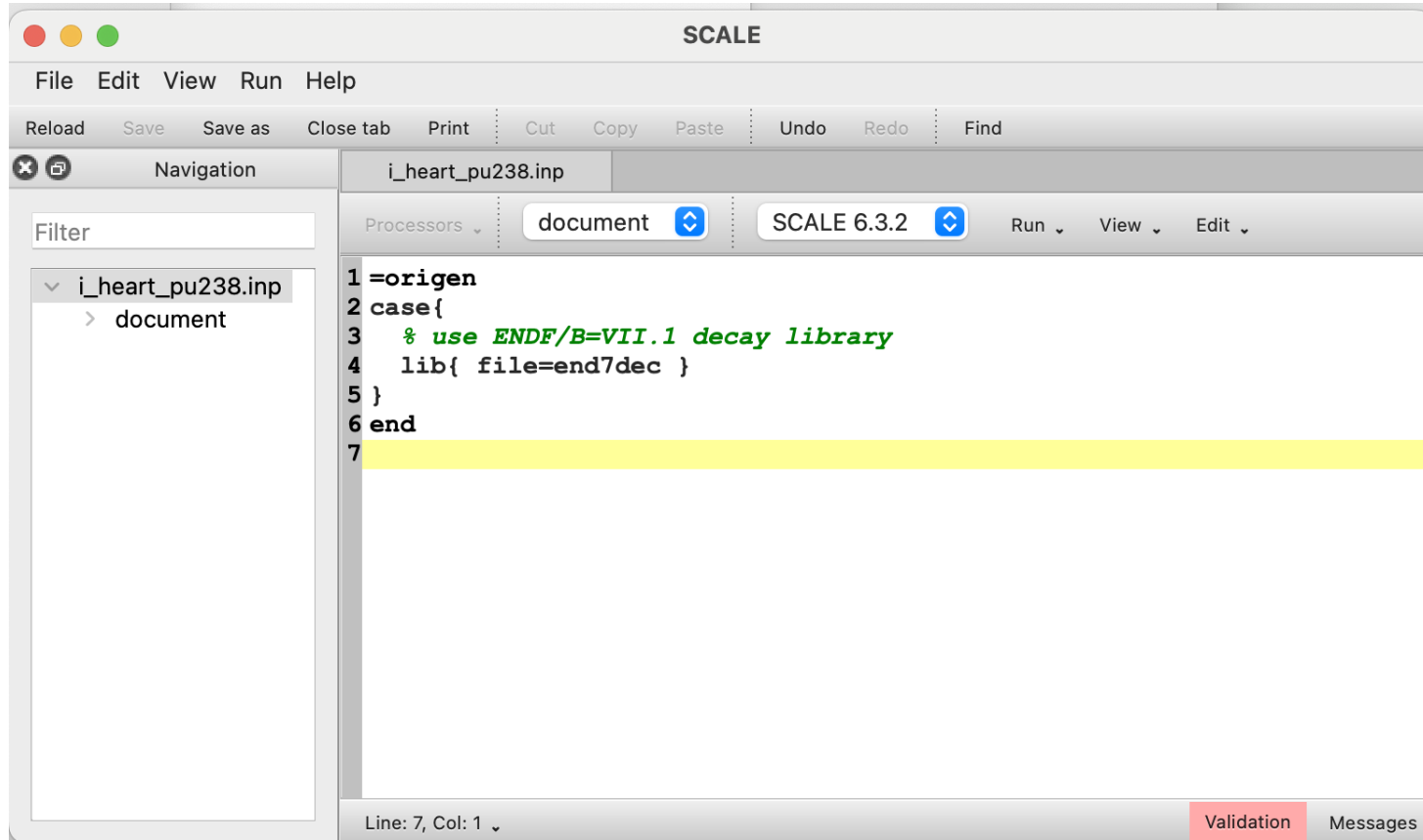
The screenshot shows the SCALE 6.3.2 software interface. The main window is titled "SCALE" and contains a menu bar (File, Edit, View, Run, Help) and a toolbar (Reload, Save, Save as, Close tab, Print, Cut, Copy, Paste, Undo, Redo, Find). A "Navigation" panel on the left shows a tree view with "i_heart_pu238.inp*" and "document". The main editor area displays the following code:

```
1 =origen
2 case{
3
4 % insert case components here
5
6 }
7
8 end
9
```

The code is line-numbered 1 through 9. Line 9 is highlighted in yellow. The status bar at the bottom indicates "Line: 9, Col: 1" and has buttons for "Validation" and "Messages".

“**case**” is always the problem “unit” for each ORIGEN calculation

Simple decay calculation: Add a **lib** block

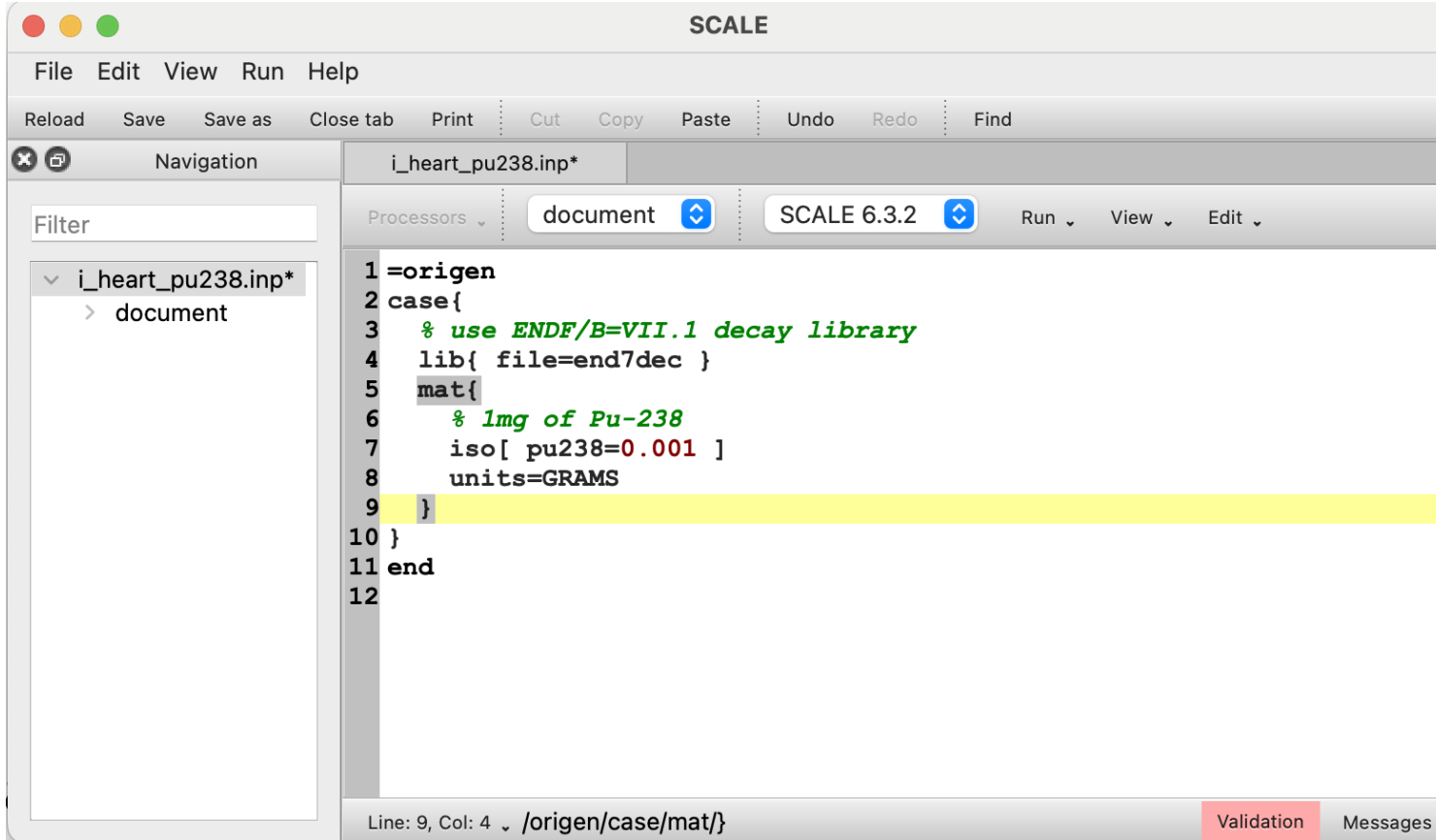


- Syntax is:

```
lib{ file=filename
pos=file_position }
```

- Use the special “**end7dec**” library, which is a **decay-only** library

Simple decay calculation: Add a **mat** block for ^{238}Pu



The screenshot shows the SCALE 6.3.2 software interface. The main window displays a text editor with the following code:

```
1 =origen
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10 }
11 end
12
```

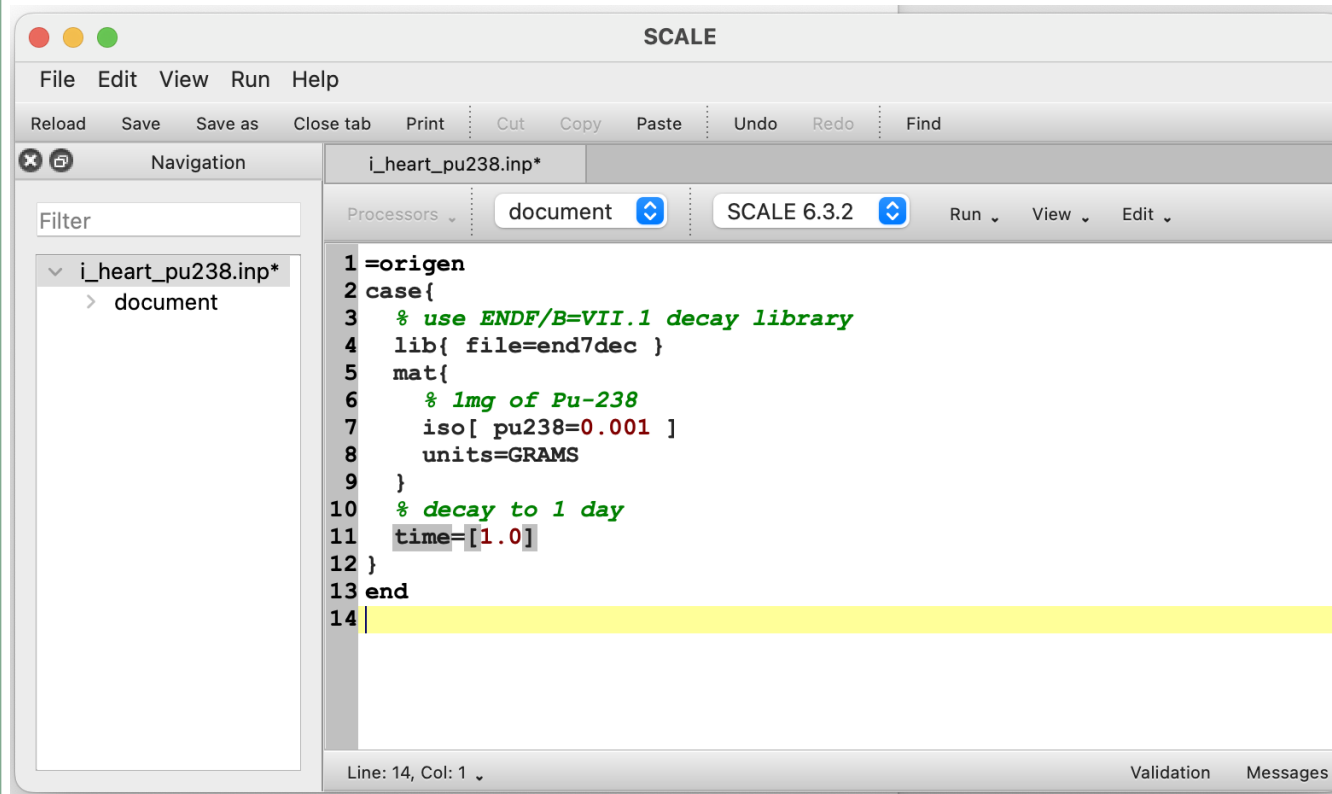
The line numbers 1 through 12 are visible on the left side of the code editor. The line containing the closing brace of the `mat` block (line 9) is highlighted in yellow. The status bar at the bottom indicates "Line: 9, Col: 4" and shows "Validation" and "Messages" buttons.

Material block has two components:

- **iso**= [list of key/value pairs]
- **units**= Concentration units for values in **iso** array (e.g., GRAMS)
- Default concentration units are MOLES

Add 1 mg (0.001) of Pu-238

Simple decay calculation: Add a **time** block



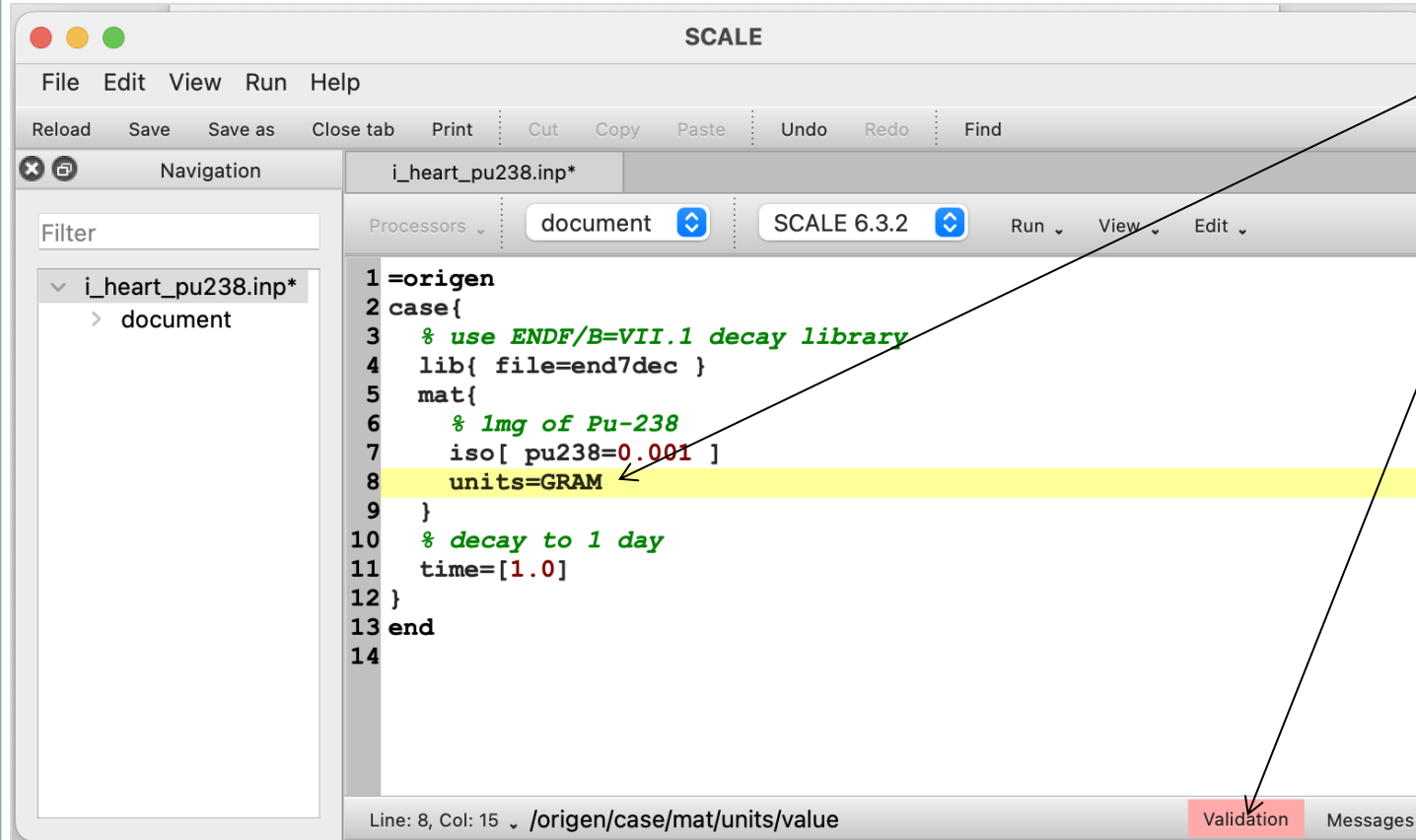
The screenshot shows the SCALE 6.3.2 application window. The menu bar includes File, Edit, View, Run, and Help. Below the menu bar is a toolbar with buttons for Reload, Save, Save as, Close tab, Print, Cut, Copy, Paste, Undo, Redo, and Find. The main window is titled 'i_heart_pu238.inp*' and contains a script editor. The script is as follows:

```
1 =origen
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10  % decay to 1 day
11  time=[1.0]
12 }
13 end
14
```

The script is currently at line 14, column 1. The status bar at the bottom indicates 'Line: 14, Col: 1' and 'Validation Messages'.

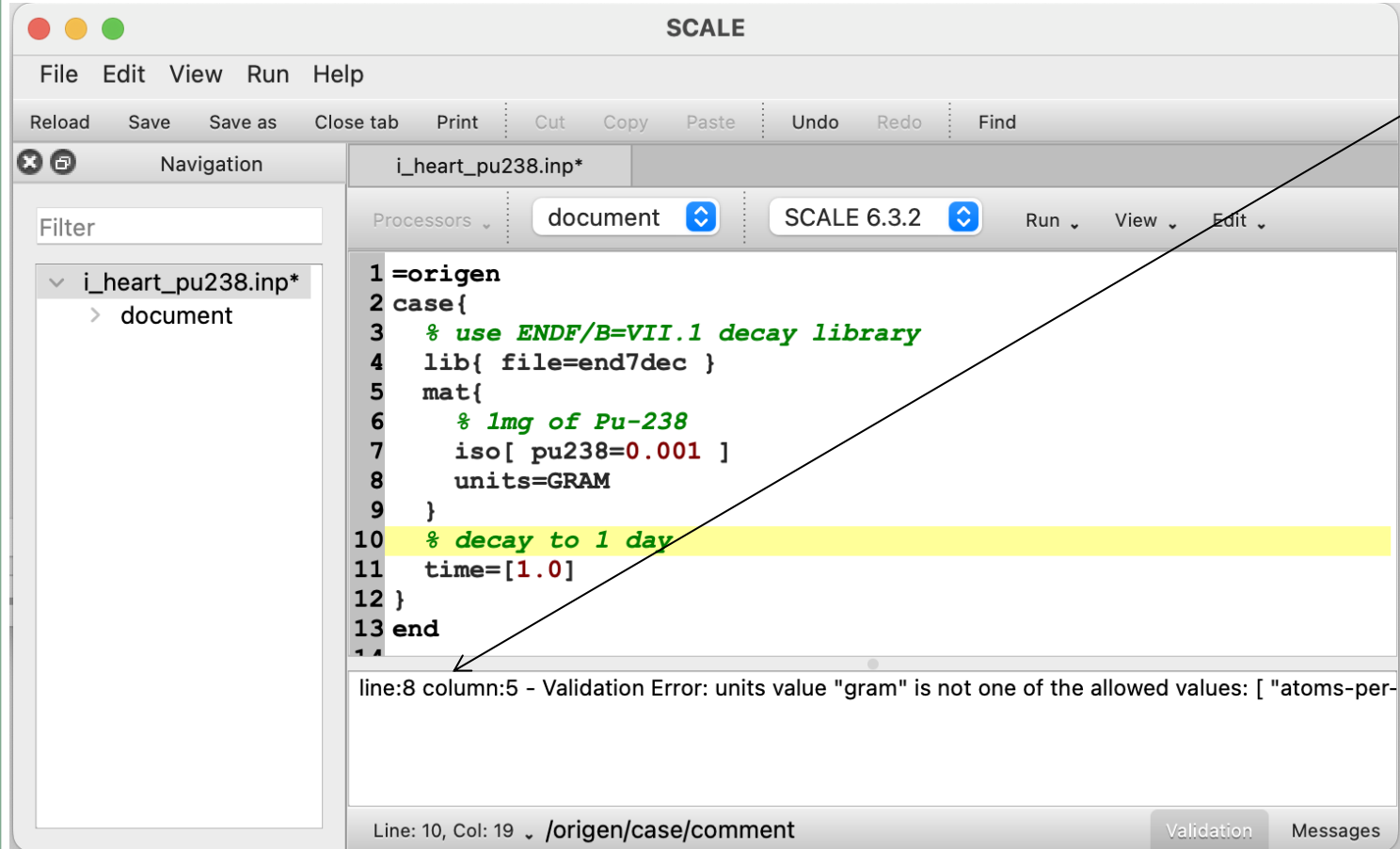
- For now, we'll just do 1 day
- Can use shortcut notation **t=[1.0]**
(default units are **days**)
- We'll go back and add the full time in a moment
- Our problem is ready to run!

Simple decay calculation: **Validate** the ORIGIN input



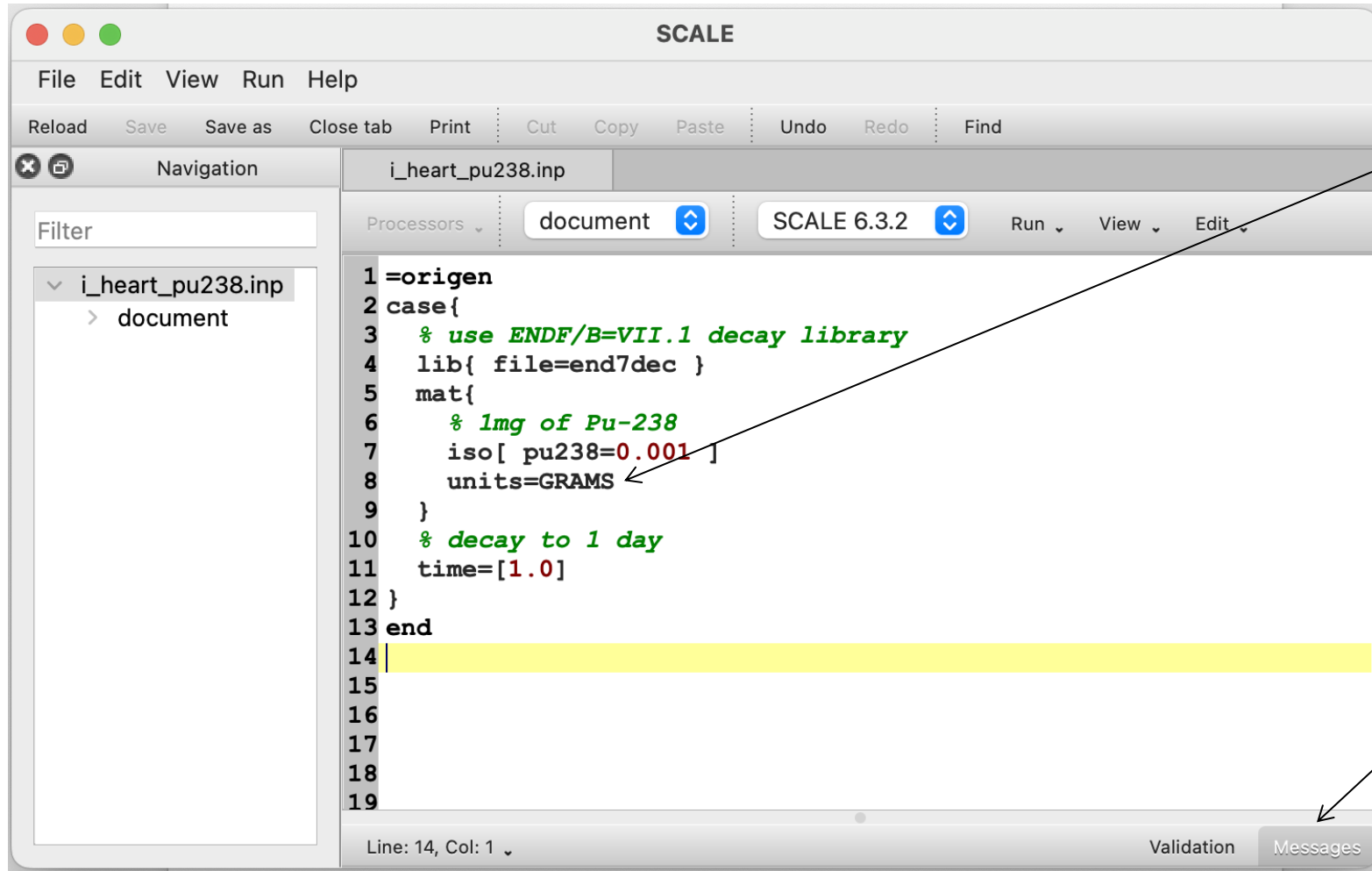
- Remove "S" from "GRAMS"
- Notice **Validation** button in bottom right (red shade)
- Click on button to get validation error details

Simple decay calculation: Find validation errors



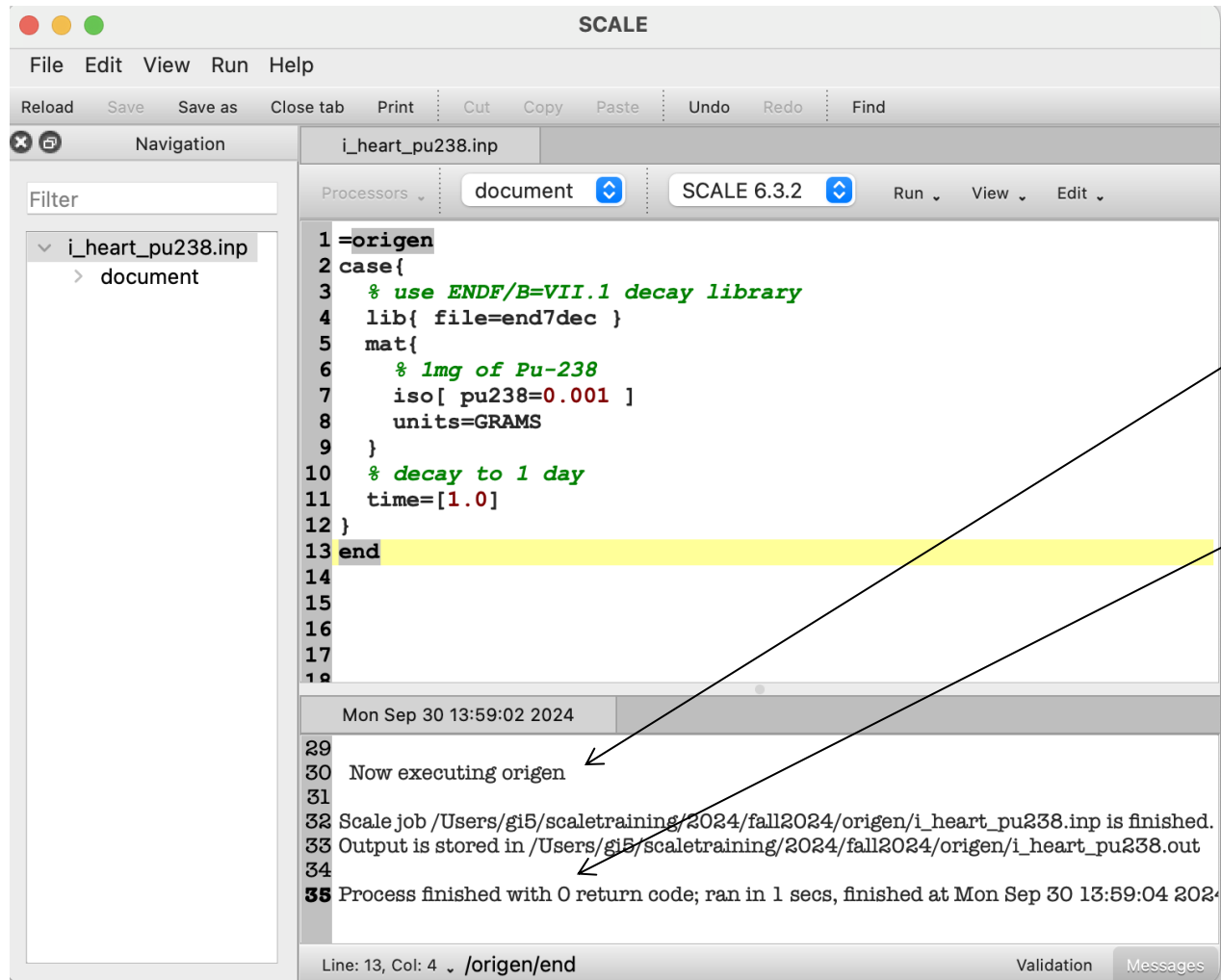
- Validation box shows **LINE: COLUMN** and then a message
- Clicking the message moves the cursor there

Simple decay calculation: Run calculation



- Correct "GRAM" to GRAMS" and save file by click on "Save"
- Click the "Run" button
- Notice shaded Messages button
- Click Messages to see code running

Simple decay calculation: Examine messages



The screenshot shows the SCALE 6.3.2 application window. The top menu bar includes File, Edit, View, Run, and Help. Below it is a toolbar with buttons for Reload, Save, Save as, Close tab, Print, Cut, Copy, Paste, Undo, Redo, and Find. The main window is divided into a left sidebar with a file tree showing 'i_heart_pu238.inp' and 'document', and a central editor area. The editor displays a script with the following content:

```
1 =origen
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10  % decay to 1 day
11  time=[1.0]
12 }
13 end
```

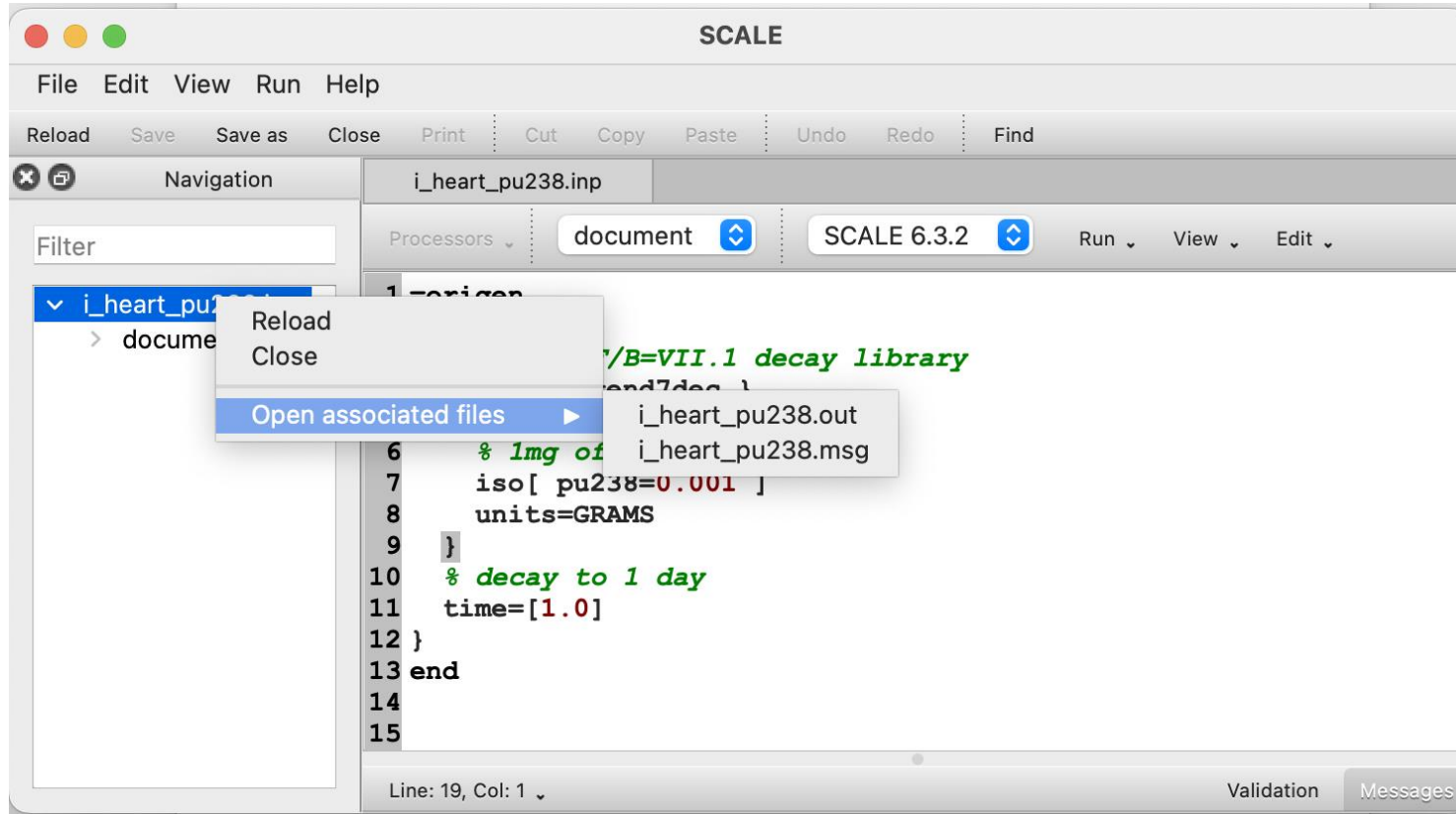
The line '13 end' is highlighted in yellow. Below the editor is a status bar showing 'Mon Sep 30 13:59:02 2024'. At the bottom, a messages pane displays the following output:

```
29
30 Now executing origen
31
32 Scale job /Users/gi5/scaletraining/2024/fall2024/origen/i_heart_pu238.inp is finished.
33 Output is stored in /Users/gi5/scaletraining/2024/fall2024/origen/i_heart_pu238.out
34
35 Process finished with 0 return code; ran in 1 secs, finished at Mon Sep 30 13:59:04 2024
```

The status bar at the bottom indicates 'Line: 13, Col: 4 /origen/end' and has buttons for 'Validation' and 'Messages'.

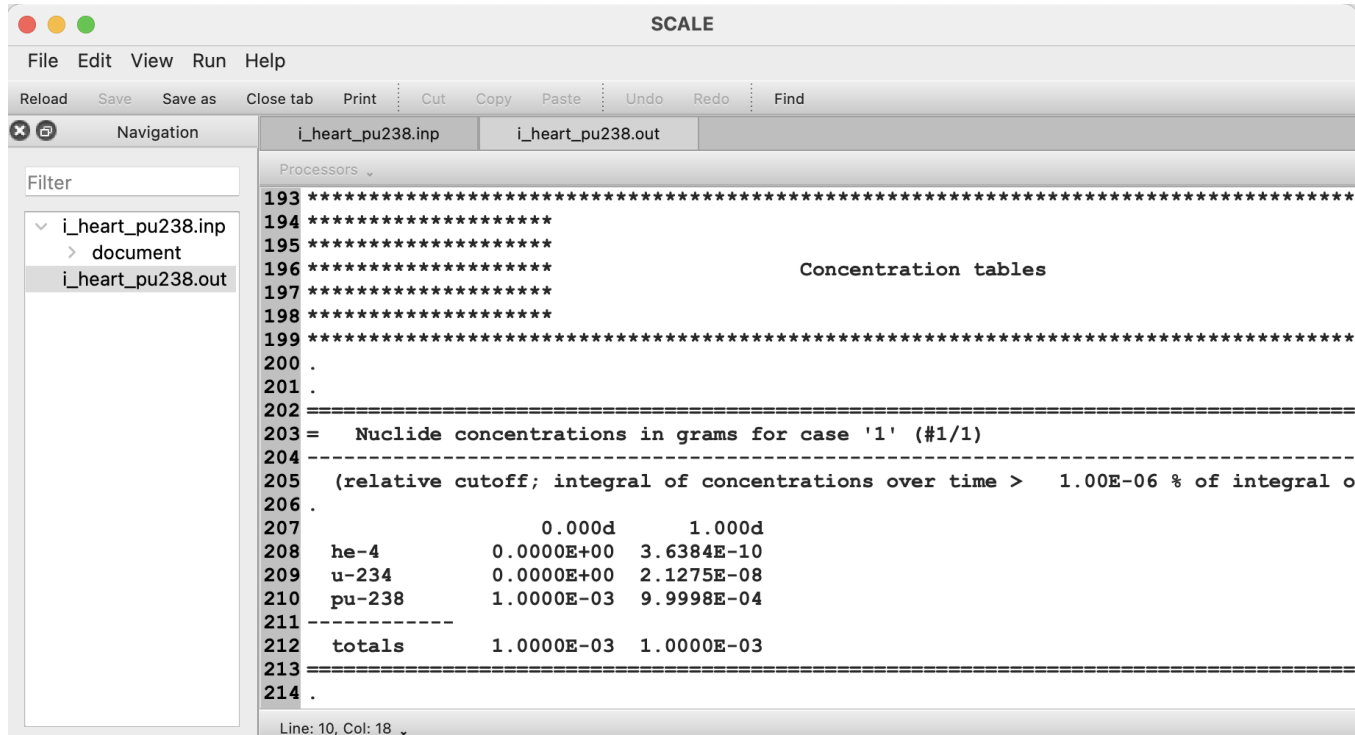
- See sequences executed
- A “0 return code” means success

Simple decay calculation: Open output file



- Easiest way is to right-click on file name in the navigation pane and “Open associated files”
→ i_heart_pu238.out

Simple decay calculation: Examine output file

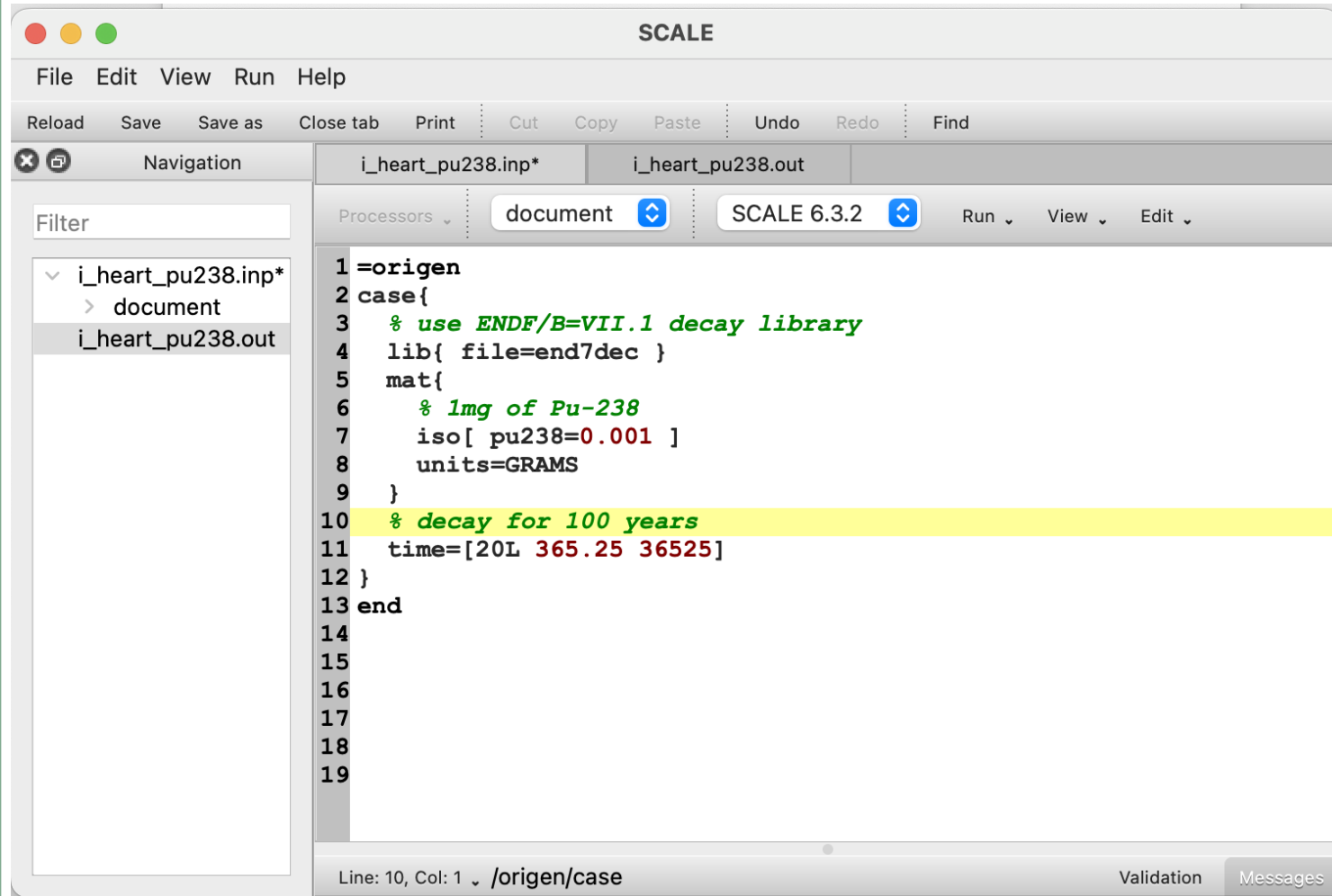


```
193 *****
194 *****
195 *****
196 *****              Concentration tables
197 *****
198 *****
199 *****
200 .
201 .
202 =====
203 =   Nuclide concentrations in grams for case '1' (#1/1)
204 -----
205 (relative cutoff; integral of concentrations over time > 1.00E-06 % of integral o
206 .
207           0.000d      1.000d
208 he-4      0.0000E+00  3.6384E-10
209 u-234     0.0000E+00  2.1275E-08
210 pu-238    1.0000E-03  9.9998E-04
211 -----
212 totals    1.0000E-03  1.0000E-03
213 =====
214 .
```

Line: 10, Col: 18

- “Concentration tables” are printed for each case
- By default, only grams are printed
- Nothing really happens to ^{238}Pu in a day...
 - Half-life is 87.7 years

Simple decay calculation: Add decay time to 100 years



The screenshot shows the SCALE 6.3.2 software interface. The main window displays the input file `i_heart_pu238.inp` with the following content:

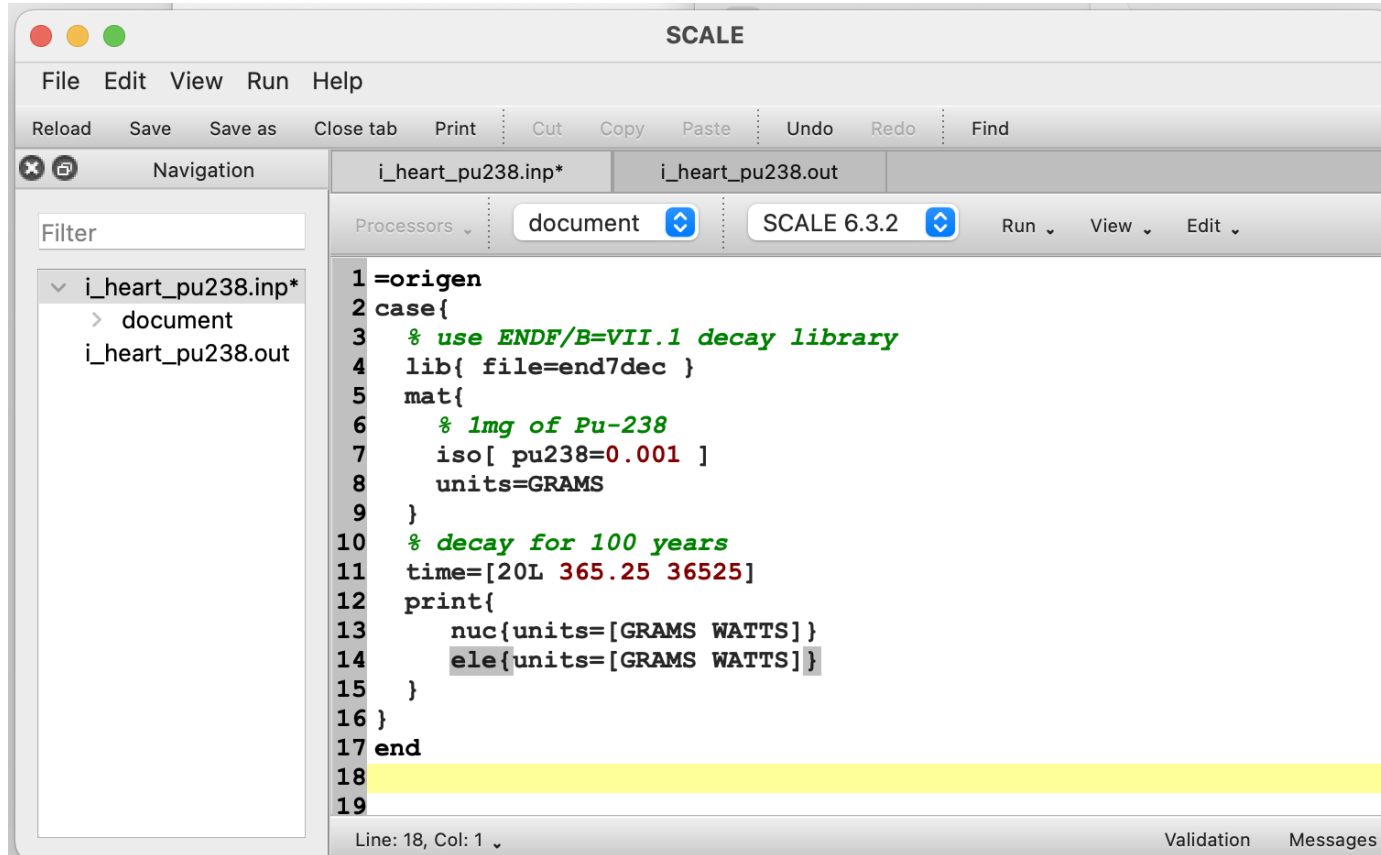
```
1 =origen
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10  % decay for 100 years
11  time=[20L 365.25 36525]
12 }
13 end
14
15
16
17
18
19
```

The line `% decay for 100 years` is highlighted in yellow. The status bar at the bottom indicates the cursor is at Line 10, Col: 1.

- 100 years is 36,525 days
 - Keeping it in days now for simplicity
- Input shortcut:
 - **20L** = 20 logarithmically-spaced intervals between 365.25 and 36,525 days

Simple decay calculations:

Examine activity and elemental breakdowns



The screenshot shows the SCALE 6.3.2 software interface. The main window displays a script for calculating the decay of Pu-238. The script includes comments in green and code in black. The script sets up a material of Pu-238, specifies the decay library, and calculates the activity and elemental breakdown after 100 years. The script is as follows:

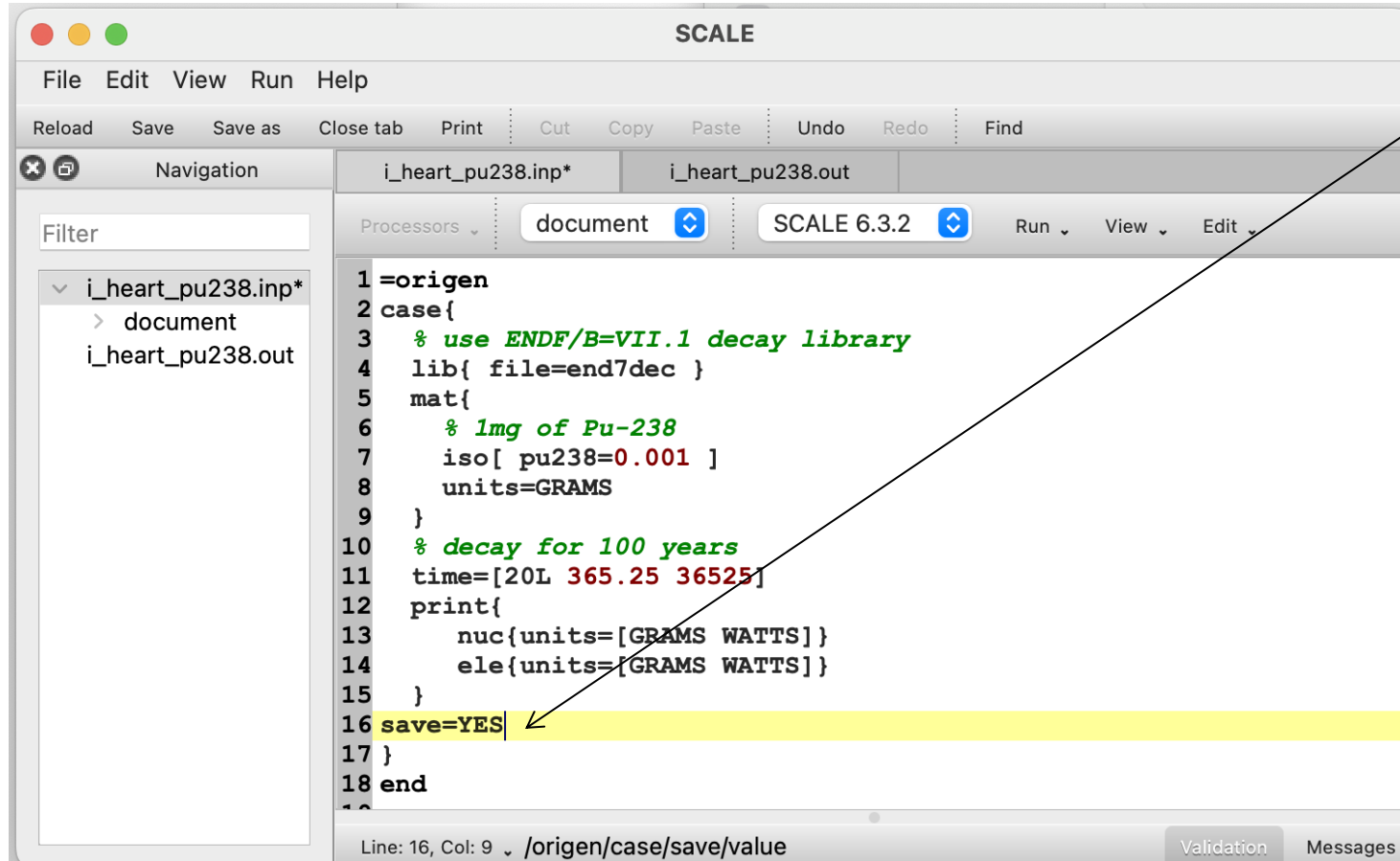
```
1 =origen
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10  % decay for 100 years
11  time=[20L 365.25 36525]
12  print{
13    nuc{units=[GRAMS WATTS]}
14    ele{units=[GRAMS WATTS]}
15  }
16 }
17 end
18
19
```

The interface also shows a navigation pane on the left with a filter and a list of files: i_heart_pu238.inp* and i_heart_pu238.out. The status bar at the bottom indicates 'Line: 18, Col: 1' and 'Validation Messages'.

- Use the **print** block
- **print** block contains nuclide (**nuc**) and element (**ele**) blocks
- Blocks on the left print activity/mass by nuclide and element
- Add printout of decay heat (**WATTS**)

Simple decay calculation:

Save ORIGIN binary concentration file (f71)



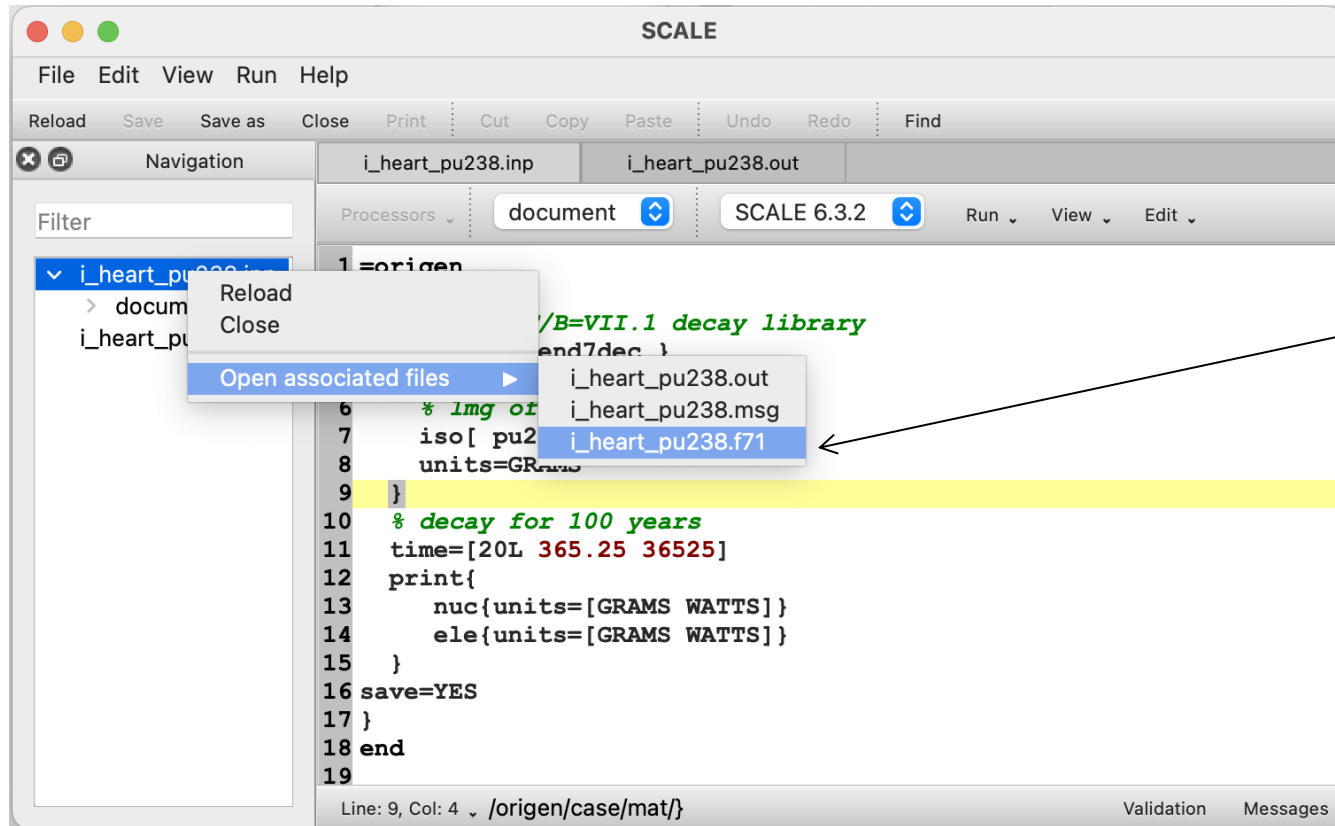
The screenshot shows the SCALE 6.3.2 software interface. The main window displays a script for a simple decay calculation. The script is as follows:

```
1 =origin
2 case{
3   % use ENDF/B=VII.1 decay library
4   lib{ file=end7dec }
5   mat{
6     % 1mg of Pu-238
7     iso[ pu238=0.001 ]
8     units=GRAMS
9   }
10  % decay for 100 years
11  time=[20L 365.25 36525]
12  print{
13    nuc{units=[GRAMS WATTS]}
14    ele{units=[GRAMS WATTS]}
15  }
16  save=YES
17 }
18 end
```

The line `save=YES` is highlighted in yellow. A line points from the first bullet point in the list to this line. The status bar at the bottom indicates the cursor is at Line 16, Col: 9, and the current path is `/origin/case/save/value`.

- Use the **save=yes** to generate binary concentration file
- The **save** block gives more control over the exact data to output
- Run input

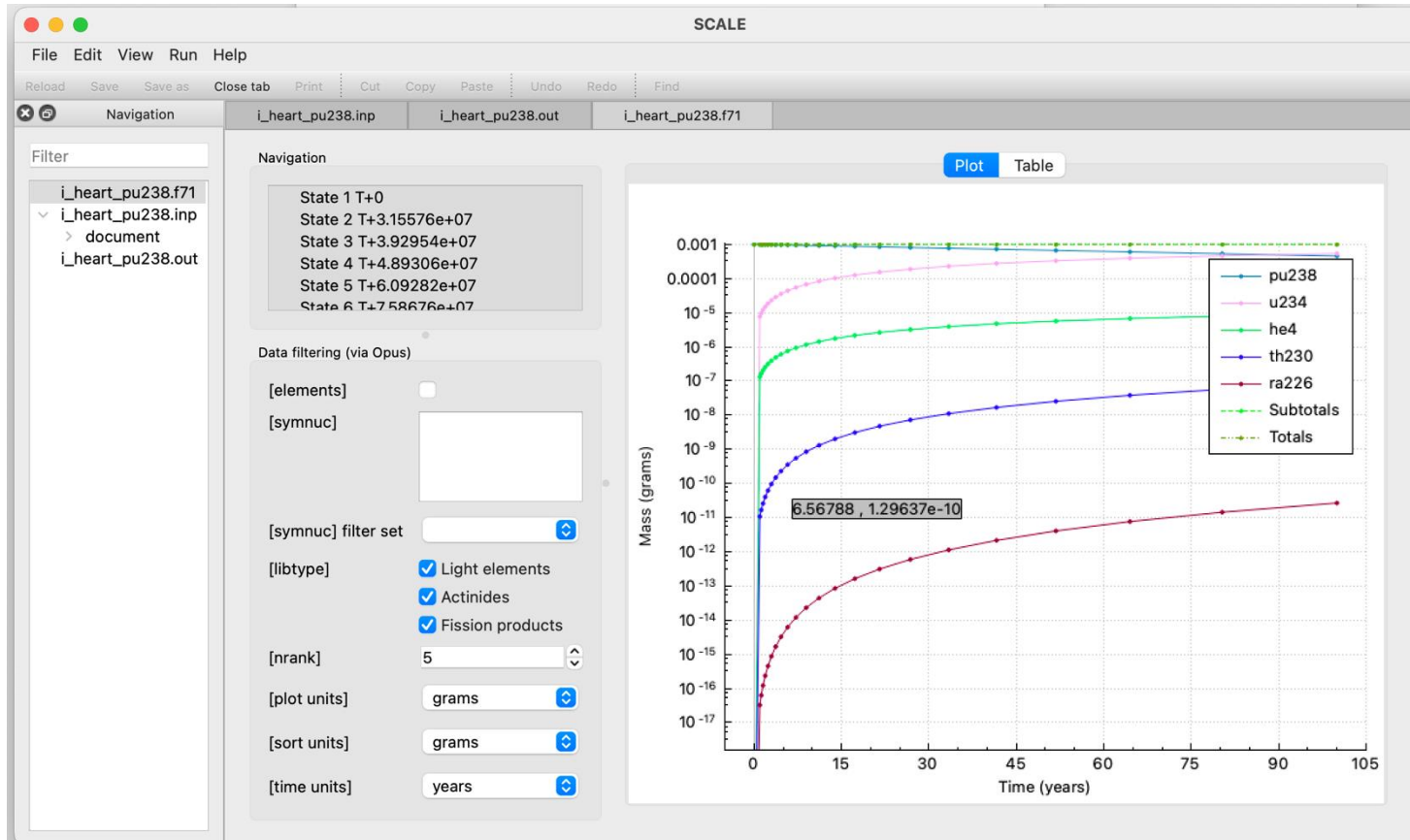
Simple decay calculations: Open f71 file



- Right-click on **i_heart_pu238.inp**
- Open associated files
→ "i_heart_pu238.f71"

Simple decay calculation:

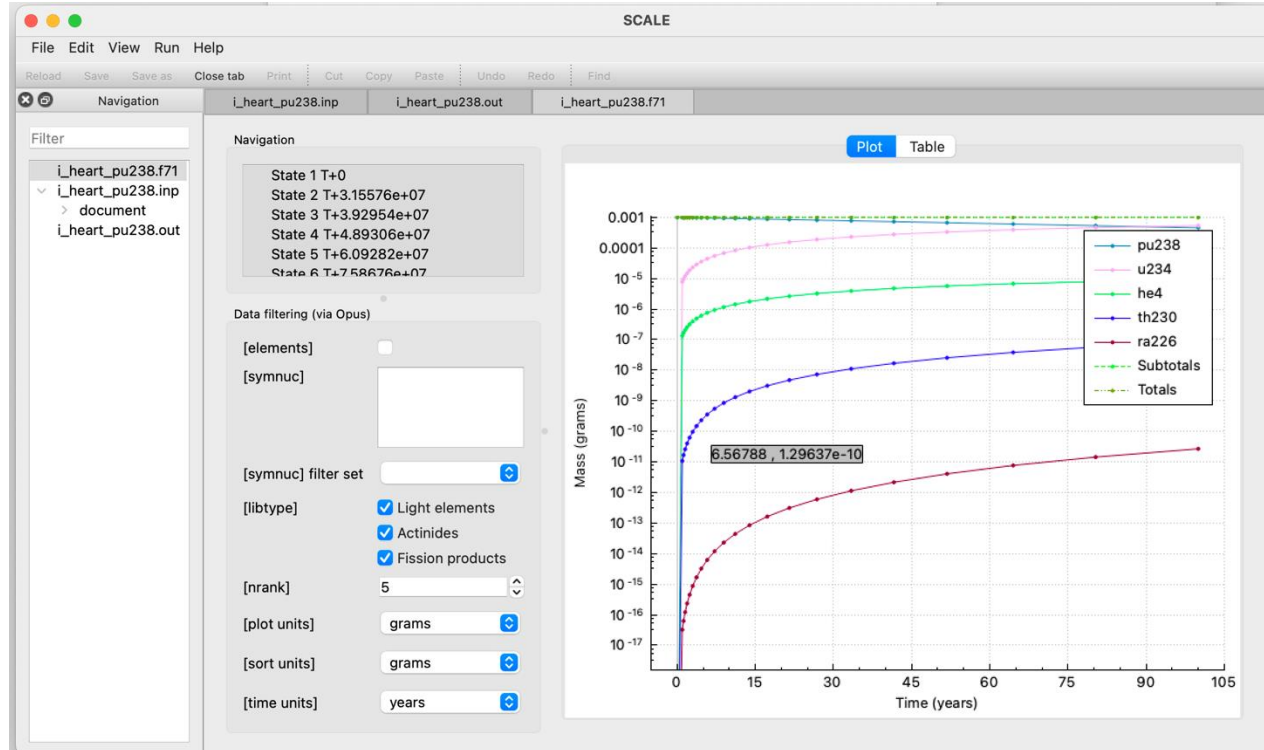
Use f71 viewer to view concentration plot (in grams)



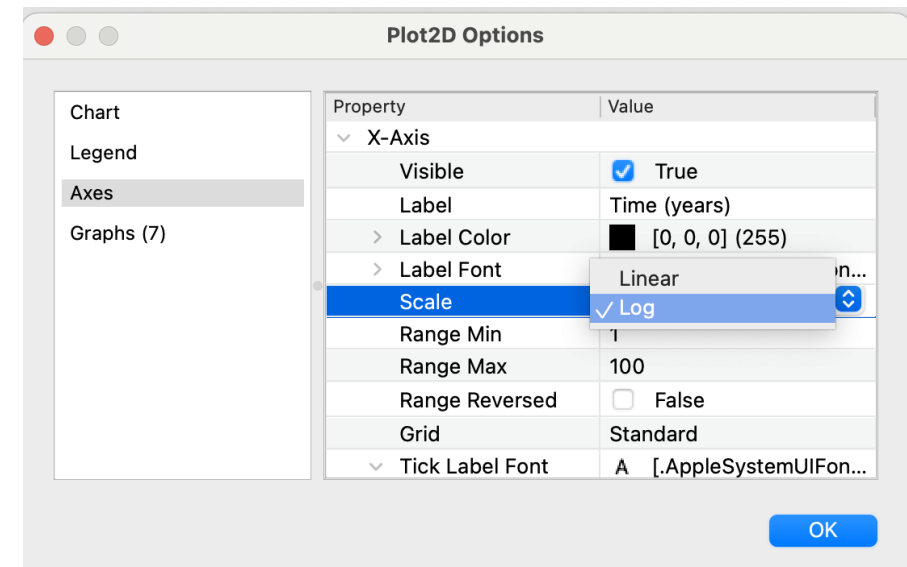
- First select all states (should be 23) any of the following ways
 - Left-click and drag down
 - Hold SHIFT and select range
 - CTRL-A
- Change [plot units] and [sort units] to “grams”

Simple decay calculation:

Can change x-axis time scale to Log or Linear

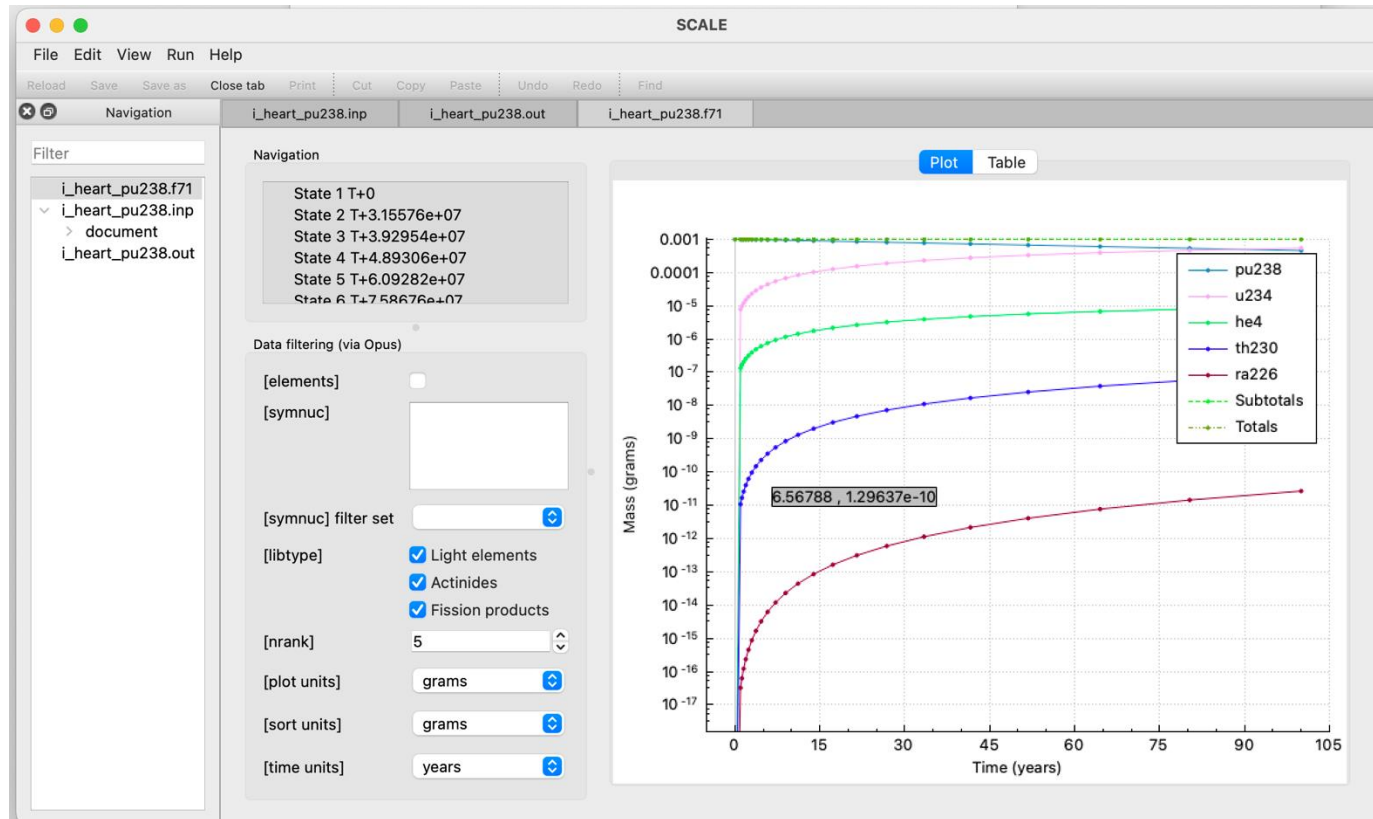


- Right click anywhere on plot and select “**Plot Options**”
- Navigate to “**Axes→X-Axis→Scale**” and change to “**Log**” or “**Linear**”



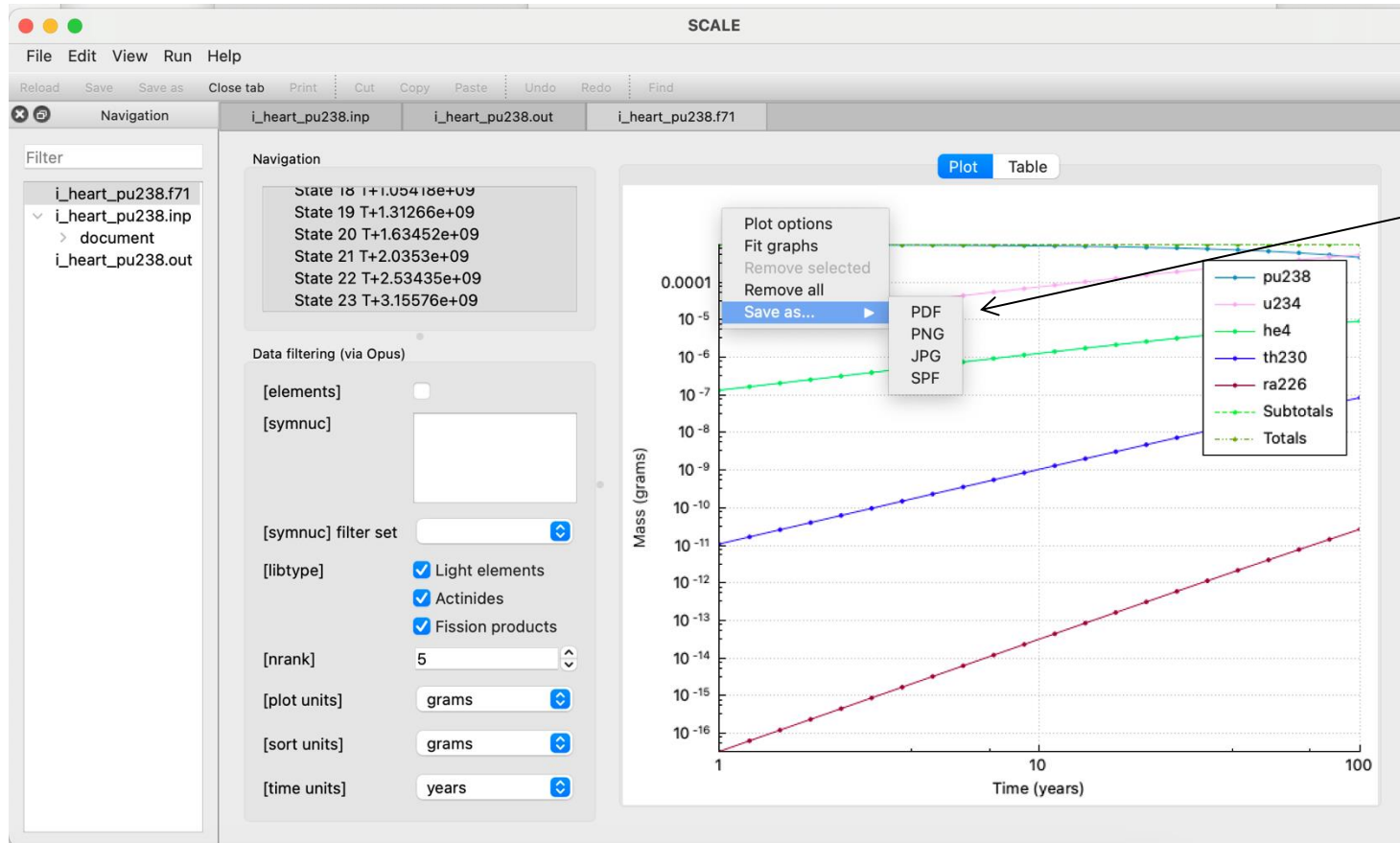
Simple decay calculation:

Experiment with different filtering options



- **nrank** is the number of things to show
- **sort units** is the units to use in the ranking
- **plot units** is the units to show on the plot
- **elements** toggles whether elements or nuclides are shown
- The plot to the left shows the concentration in grams of the top 5 nuclides or elements

Simple decay calculation: Export plot as image



- Right-click on plot and select “**Save as...**” and choose file type
- PDF option uses vector graphics—good choice for embedding in a report

Simple decay calculation:

Export table for use into another program

SCALE

File Edit View Run Help

Reload Save Save as Close tab Print Cut Copy Paste Undo Redo Find

Navigation

Filter

i_heart_pu238.f71

i_heart_pu238.inp

i_heart_pu238.out

i_heart_pu238.f71

document

i_heart_pu238.out

Navigation

State 4 T+1.00000e+07

State 5 T+6.09282e+07

State 6 T+7.58676e+07

State 7 T+9.44701e+07

State 8 T+1.17634e+08

State 9 T+1.46477e+08

State 10 T+1.82393e+08

State 11 T+2.27116e+08

State 12 T+2.82804e+08

State 13 T+3.52146e+08

State 14 T+4.38491e+08

State 15 T+5.46008e+08

State 16 T+6.79888e+08

State 17 T+8.46594e+08

State 18 T+1.05418e+09

State 19 T+1.31266e+09

State 20 T+1.63452e+09

State 21 T+2.0353e+09

Data filtering (via Opus)

[elements]

[symnuc]

[symnuc] filter set

[libtype]

☒ Light elements

☒ Actinides

☒ Fission products

[nrank]

[plot units]

[sort units]

[time units]

	pu238	u234	he4	th230	ra226	Subtotals	Totals
0.000000e+00	0.001	0	0	0	0	0.001	0.001
1.000000e+00	0.000992127	7.74006e-06	1.32372e-07	1.07539e-11	3.24064e-17	0.001	0.001
1.245197e+00	0.000990206	9.62858e-06	1.6467e-07	1.66634e-11	6.2535e-17	0.001	0.001
1.550516e+00	0.00098782	1.1975e-05	2.048e-07	2.5816e-11	1.20659e-16	0.001	0.001
1.930698e+00	0.000984856	1.48889e-05	2.54634e-07	3.99881e-11	2.32772e-16	0.001	0.001
2.404099e+00	0.000981178	1.85051e-05	3.16479e-07	6.19251e-11	4.4897e-16	0.001	0.001
2.993577e+00	0.000976617	2.2989e-05	3.93165e-07	9.58671e-11	8.65763e-16	0.000999999	0.000999999
3.727594e+00	0.000970968	2.85433e-05	4.88157e-07	1.48357e-10	1.66898e-15	0.000999999	0.000999999
4.641589e+00	0.000963979	3.54146e-05	6.05673e-07	2.29479e-10	3.21618e-15	0.000999999	0.000999999
5.779693e+00	0.000955346	4.39015e-05	7.50822e-07	3.54751e-10	6.19482e-15	0.000999999	0.000999999
7.196857e+00	0.000944705	5.43632e-05	9.29746e-07	5.48012e-10	1.19252e-14	0.000999999	0.000999999
8.961505e+00	0.00093162	6.72273e-05	1.14976e-06	8.45795e-10	2.29401e-14	0.000999998	0.000999998
1.115884e+01	0.00091558	8.29968e-05	1.41947e-06	1.30394e-09	4.40899e-14	0.000999998	0.000999998
1.389495e+01	0.000895993	0.000102254	1.74883e-06	2.00747e-09	8.46461e-14	0.000999997	0.000999997
1.730196e+01	0.000872187	0.000125657	2.14912e-06	3.08532e-09	1.62287e-13	0.000999997	0.000999997
2.154435e+01	0.000843427	0.000153932	2.63273e-06	4.73192e-09	3.10623e-13	0.000999996	0.000999996
2.682696e+01	0.000808937	0.000187839	3.21269e-06	7.23852e-09	5.93311e-13	0.000999995	0.000999995
3.340485e+01	0.000767954	0.000228127	3.90184e-06	1.10378e-08	1.13038e-12	0.000999994	0.000999994
4.159562e+01	0.000719813	0.000275451	4.71139e-06	1.67658e-08	2.14688e-12	0.000999993	0.000999993
5.179475e+01	0.000664065	0.000330252	5.6489e-06	2.53461e-08	4.06192e-12	0.000999992	0.000999992
6.449467e+01	0.000600644	0.000392592	6.71549e-06	3.80987e-08	7.64947e-12	0.00099999	0.00099999
8.030857e+01	0.000530072	0.000461957	7.90243e-06	5.68747e-08	1.43246e-11	0.000999988	0.000999988
1.000000e+02	0.000453673	0.000537042	9.18749e-06	8.4211e-08	2.66435e-11	0.000999986	0.000999986

- Click on the Table Tab
- Click on the top left square to highlight all data
- **CTRL-C** to copy contents
- Can open the other program (e.g., MS Excel) and **CTRL-V** to paste contents

Solution

```
=origen
case{
  lib{ file=end7dec }
  mat{
    iso[ pu238=1e-3 ]
    units=GRAMS
  }
  time=[ 20L 1 36525 ]
  print{ nuc{ units=[grams watts] } }
  save=yes
}
end
```

Simple decay calculation: Summary

- Input file structure is simple for simple decay calculations
 - one ORIGIN **case** block
 - use “end7dec” in **lib** block
 - short material specification (**mat** block), e.g.
 - **mat{units=grams iso[u238=1.0]}** % single nuclide
 - **mat{units=moles iso[k=1.0]}** % single element
 - rest is printing in .out file, f71 save options
- Logarithmic interpolation is convenient to create time array, e.g.
time=[5L 1 10]
- For long decay times, make sure the time “**grid**” is refined sufficiently
e.g., try both **time=[5L 1 10]**
and
time=[10L 1 10]
and see if your results change significantly