



# The transmutation chains period folding on the use of the parent pointer tree data structure

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE  
AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Przemysław Stanisław,  
Jerzy Cetnar



Konferencję Użytkowników  
Komputerów Dużej Mocy – KU KDM'17  
10 March 2017, Zakopane



## Software



The Monte Carlo Continuous  
Energy Burn-up Code

$$\text{MCB} = \text{MCNP} + \text{TTA}$$

Two main methodologies required to describe and predict nuclear core behavior

**MCNP** Monte Carlo N-Particle Transport Code is used to describe **Boltzmann equation**.

Describe neutron distribution and spectrum

**TTA** Transmutation Trajectory Analysis analyzes nuclide density by solving **Bateman equation**.

Describe fuel evolution

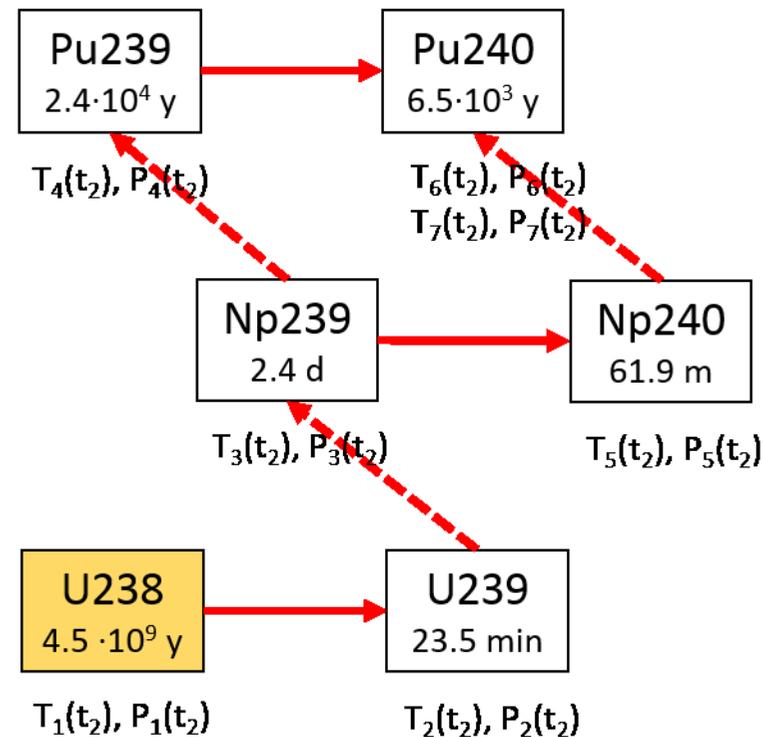
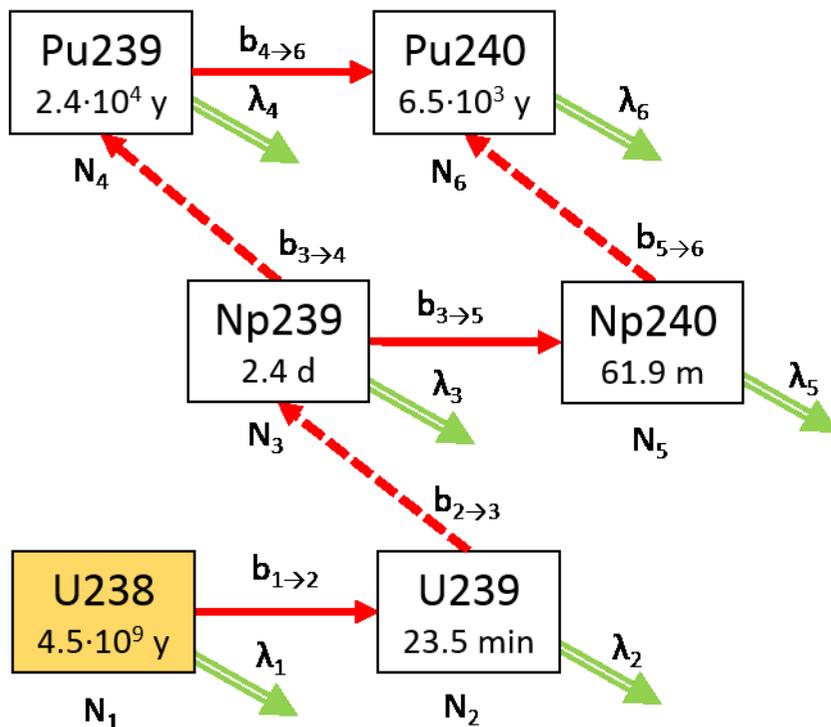
# Bateman Equations Solution (general case)

$$N_n(t) = N_1(0) \frac{B}{\lambda_n} \sum_{i=1}^n \lambda_i \alpha_i e^{-\lambda_i t} \cdot \sum_{m=0}^{\mu_i} \frac{(\lambda_i t)^m}{m!} \Omega_{i, \mu_i - m}$$

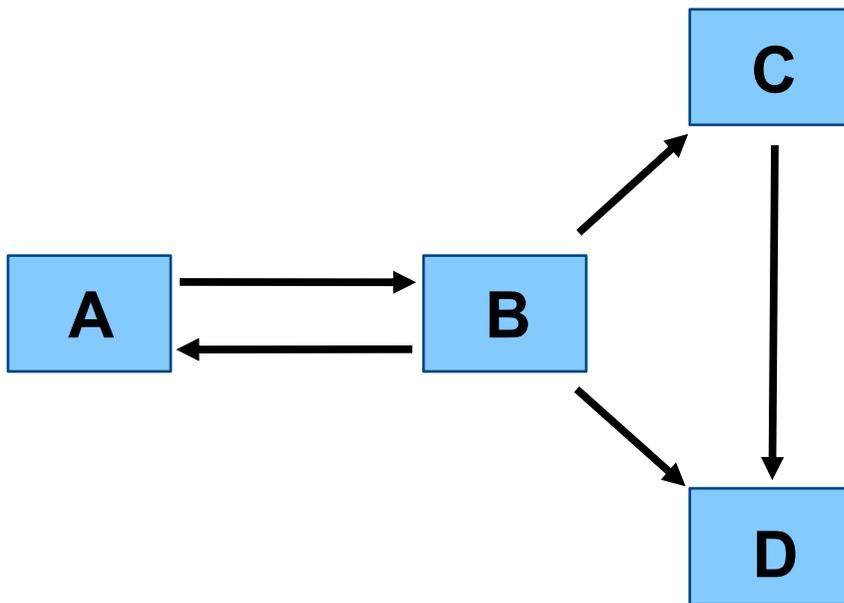
$$\alpha_i = \prod_{\substack{j=1 \\ j \neq i}}^n \left( \frac{\lambda_j}{\lambda_j - \lambda_i} \right)^{m_j}, \quad (i = 1, n)$$

$$\Omega_{i,j} = \sum_{h_1=0}^j \sum_{h_2=0}^j \cdots \sum_{h_{i-1}=0}^j \sum_{h_{i+1}=0}^j \cdots \sum_{h_n=0}^j \prod_{\substack{k=1 \\ k \neq i}}^n \binom{h_k + \mu_k}{\mu_k} \left( \frac{\lambda_i}{\lambda_i - \lambda_k} \right)^{h_k} \delta_{j,p}$$

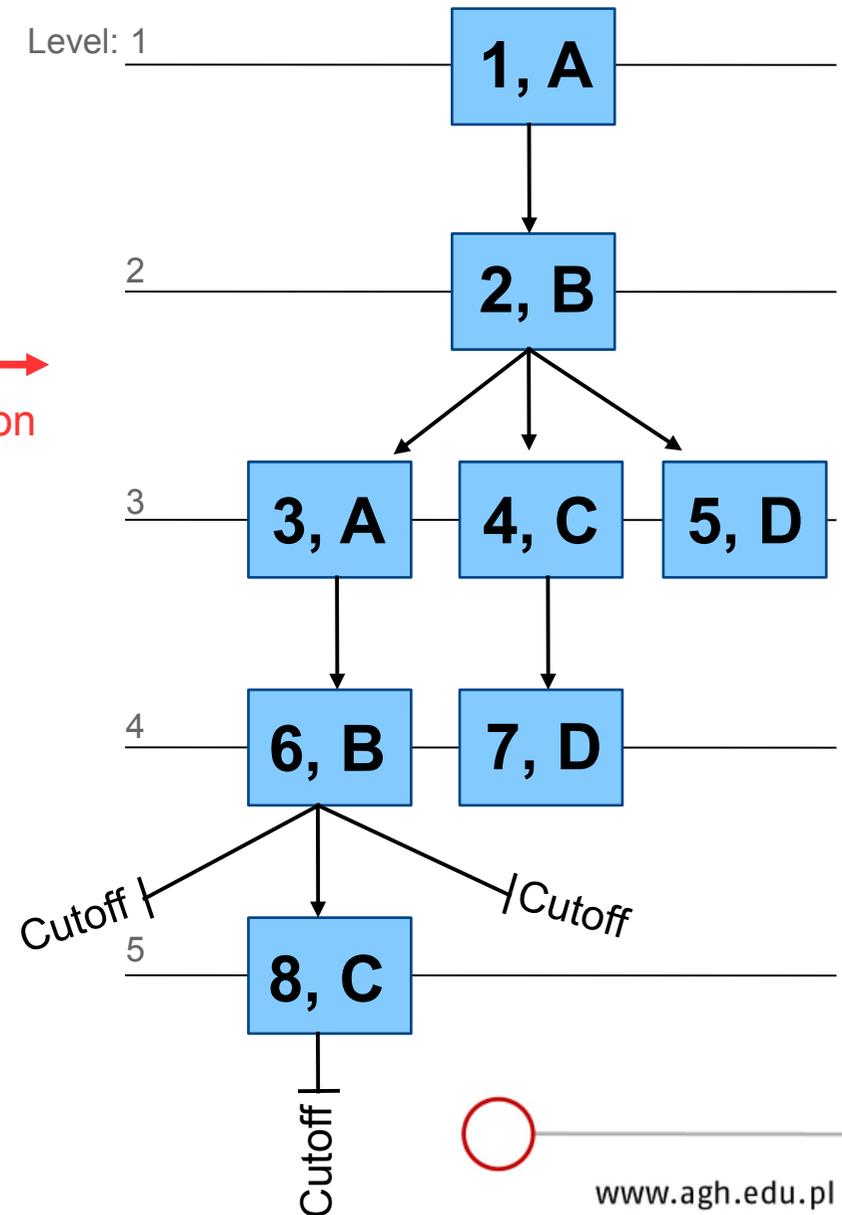
This particularly:  $\Omega_{i,0} = 1$



# The transmutation trajectory analysis



**TTA**  
  
 decomposition



The Bateman equations for transmutation trajectory

$$\frac{dN_i}{dt} = \text{production rate} - \text{destruction rate} \quad (i = 1, n)$$

The trajectory transition calculate the number density that goes from initial nuclide to the formed nuclide for a given time  $t$

$$T_n(t) = N_n(t)/N_1(0)$$

Passage is defined as the total removal rate in the considered trajectory or a fraction of the nuclides in a chain that passed beyond  $n$  nuclide and is assigned (or not) to following nuclides in the chain for the considered period.

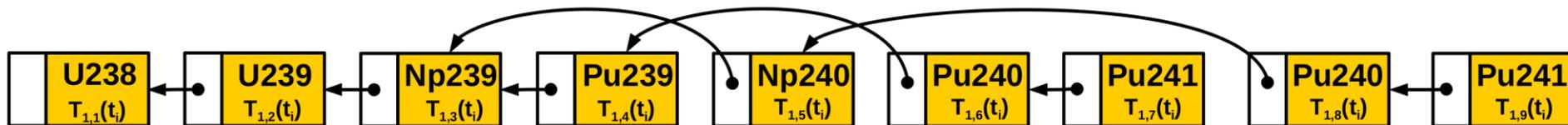
$$P_n(t) = I_n(t)/N_1(0)$$

## Parent pointer tree data structure

The proposed data structure is an N-ary tree in which each node has a pointer to its parent node, but no pointers to child nodes.

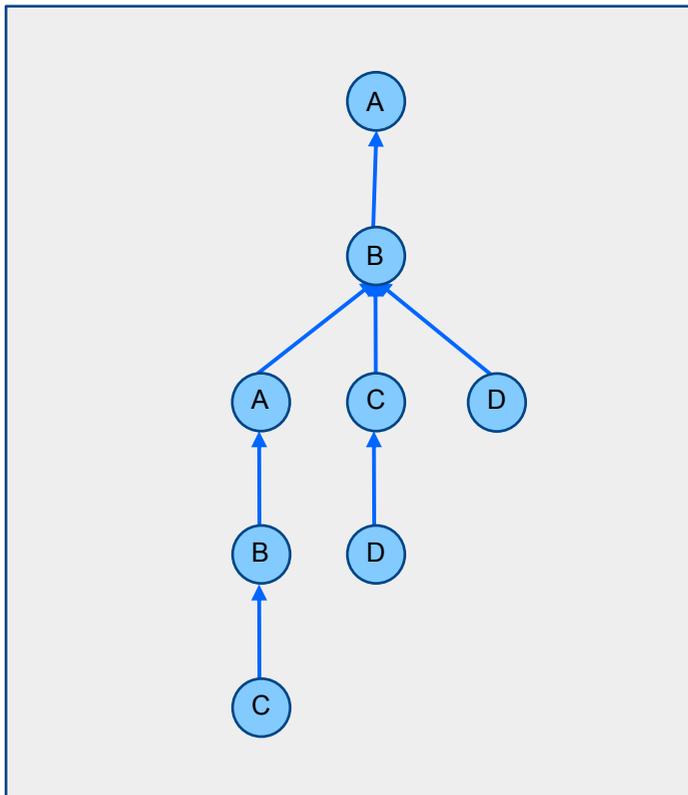
the structure stores:

- nuclide index,
- transition passage values of each chain
- information about generation number (i.e. length of the current / tree level)
- special flags i.e nuclide is last in the chain.

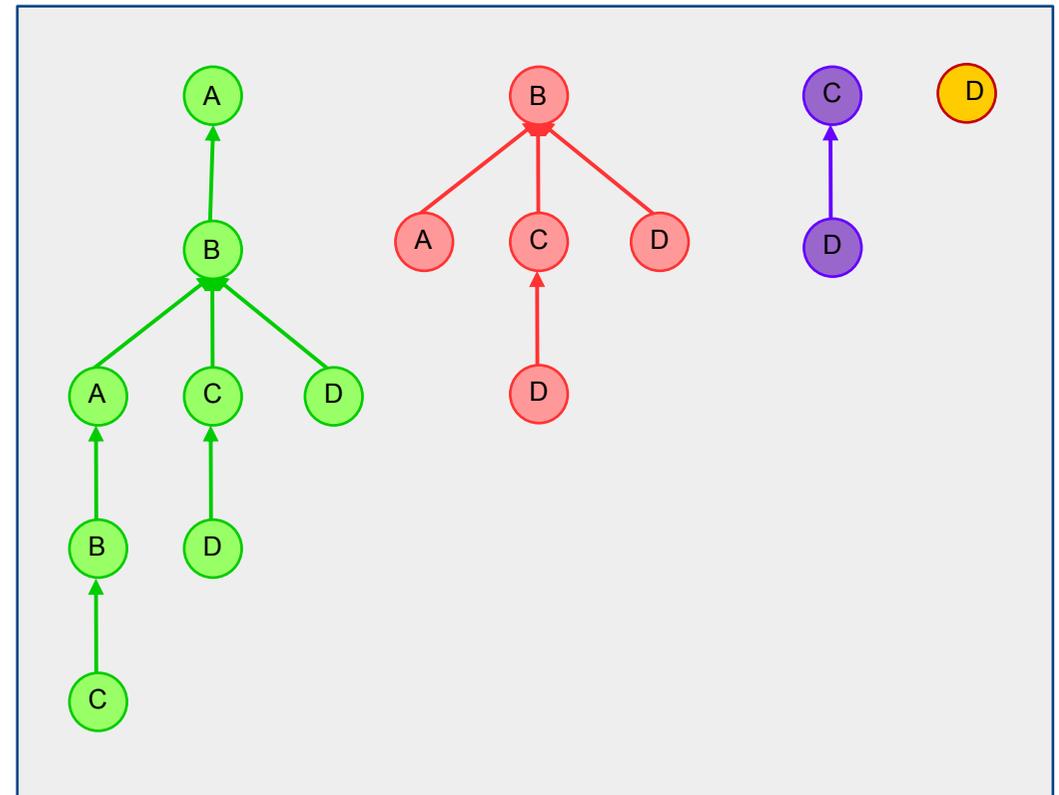


# The transmutation trajectory analysis

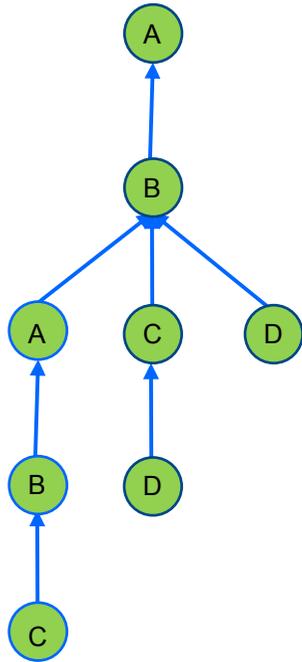
Period 1



Period 2



# Parent pointer tree folding



$$\Leftarrow (1, A) * (1, A)$$

$$\Leftarrow (1, A) * (2, B)$$

$$\Leftarrow (1, A) * (3, A)$$

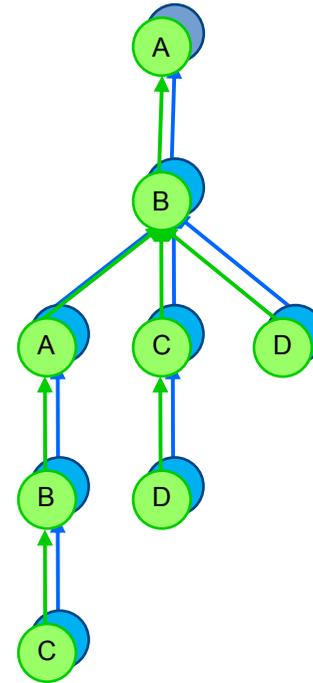
$$\Leftarrow (1, A) * (4, C)$$

$$\Leftarrow (1, A) * (5, D)$$

$$\Leftarrow (1, A) * (6, B)$$

$$\Leftarrow (1, A) * (7, D)$$

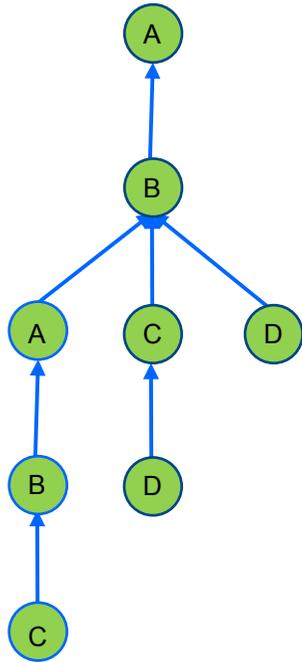
$$\Leftarrow (1, A) * (8, C)$$



Folded trajectories

Trajectories at  $t_1$  and  $t_2$

# Parent pointer tree folding



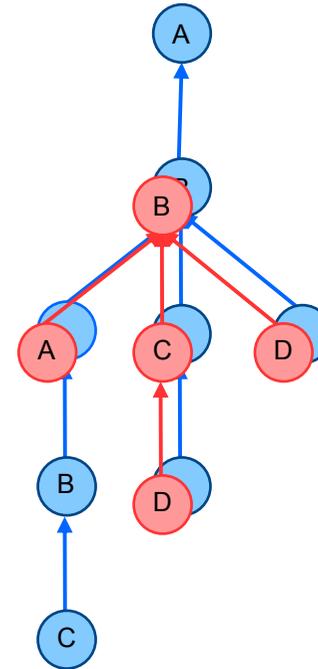
$$\Leftarrow (2, B) * (1, B)$$

$$\Leftarrow (2, B) * (2, A)$$

$$\Leftarrow (2, B) * (3, C)$$

$$\Leftarrow (2, B) * (4, D)$$

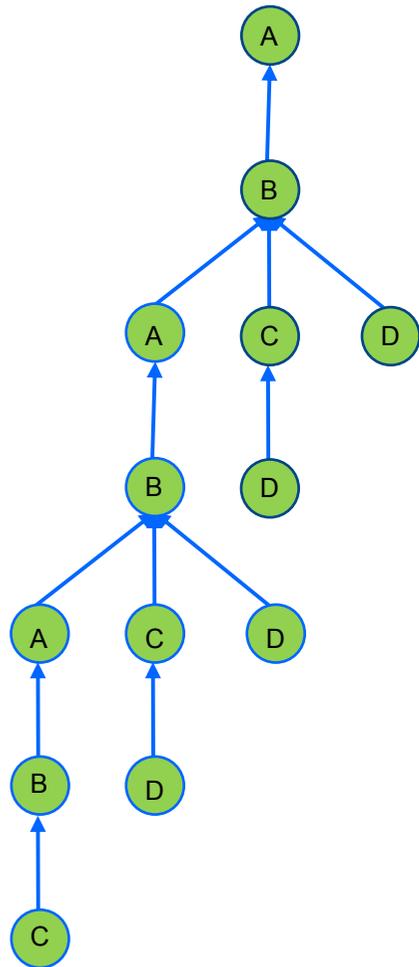
$$\Leftarrow (2, B) * (5, D)$$



Folded trajectories

Trajectories at  $t_1$  and  $t_2$

# Parent pointer tree folding



Folded trajectories

$$\Leftarrow (3, A) * (1, A)$$

$$\Leftarrow (3, A) * (2, B)$$

$$\Leftarrow (3, A) * (3, A)$$

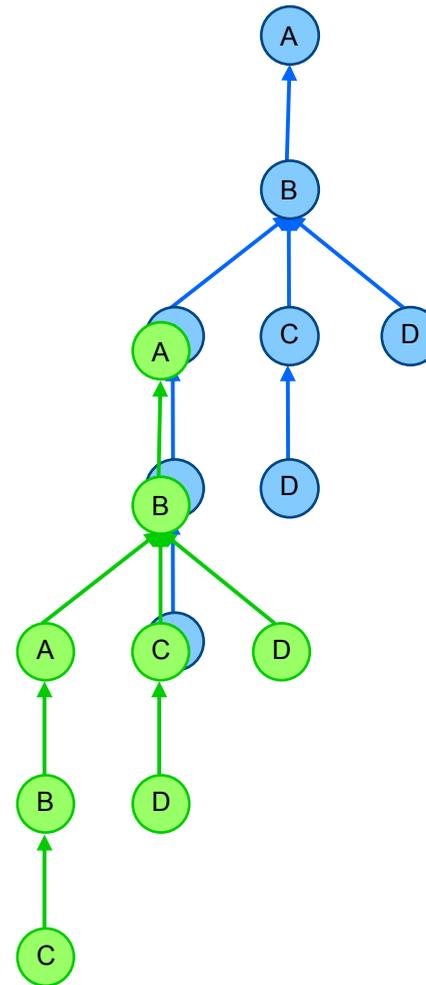
$$\Leftarrow (3, A) * (4, C)$$

$$\Leftarrow (3, A) * (5, D)$$

$$\Leftarrow (3, A) * (6, B)$$

$$\Leftarrow (3, A) * (7, D)$$

$$\Leftarrow (3, A) * (8, C)$$

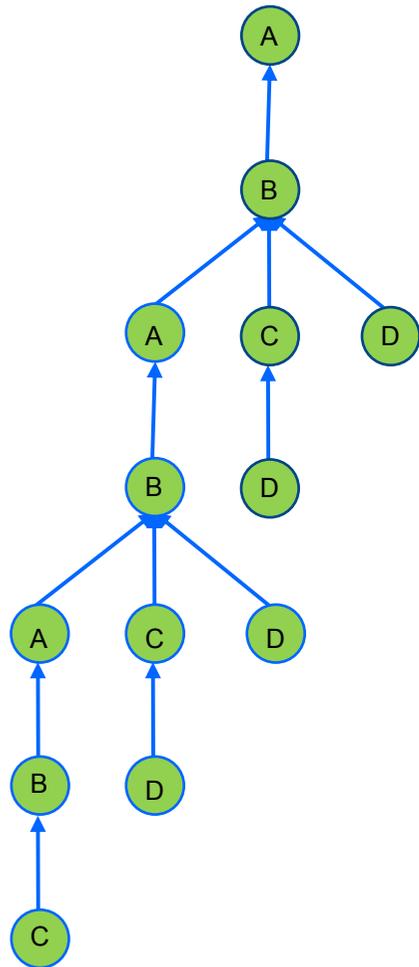


Trajectories at  $t_1$  and  $t_2$





# Parent pointer tree folding



Folded trajectories

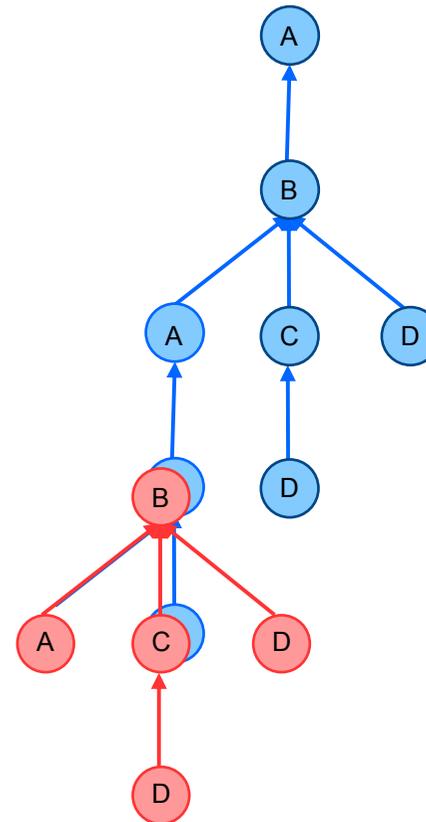
$$\leftarrow (6, B) * (1, B)$$

$$\leftarrow (6, B) * (2, A)$$

$$\leftarrow (6, B) * (3, C)$$

$$\leftarrow (6, B) * (4, D)$$

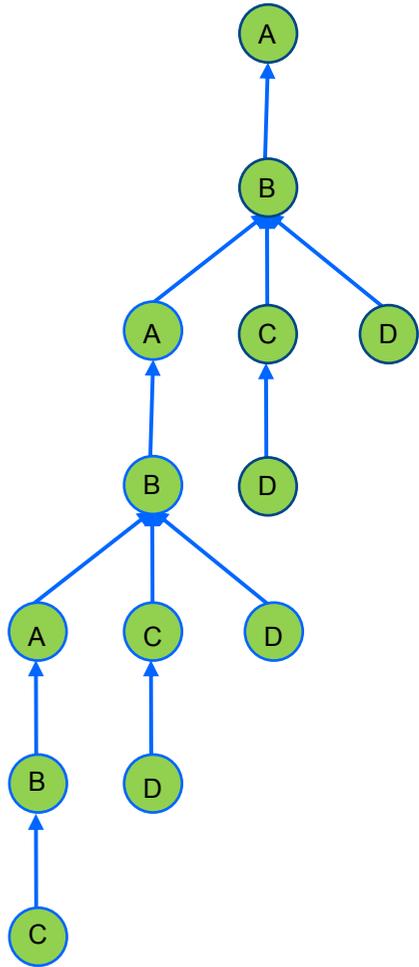
$$\leftarrow (6, B) * (5, D)$$



Trajectories at  $t_1$  and  $t_2$



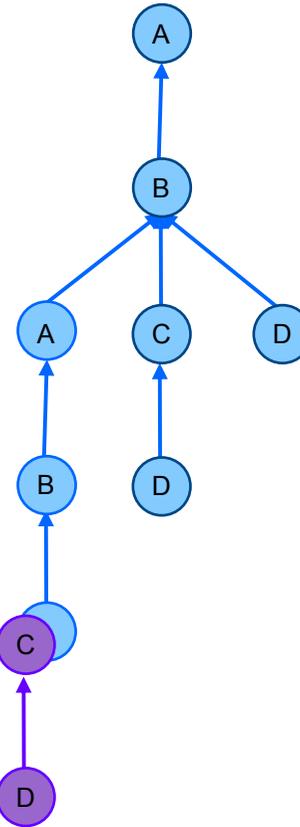
# Parent pointer tree folding



Folded trajectories

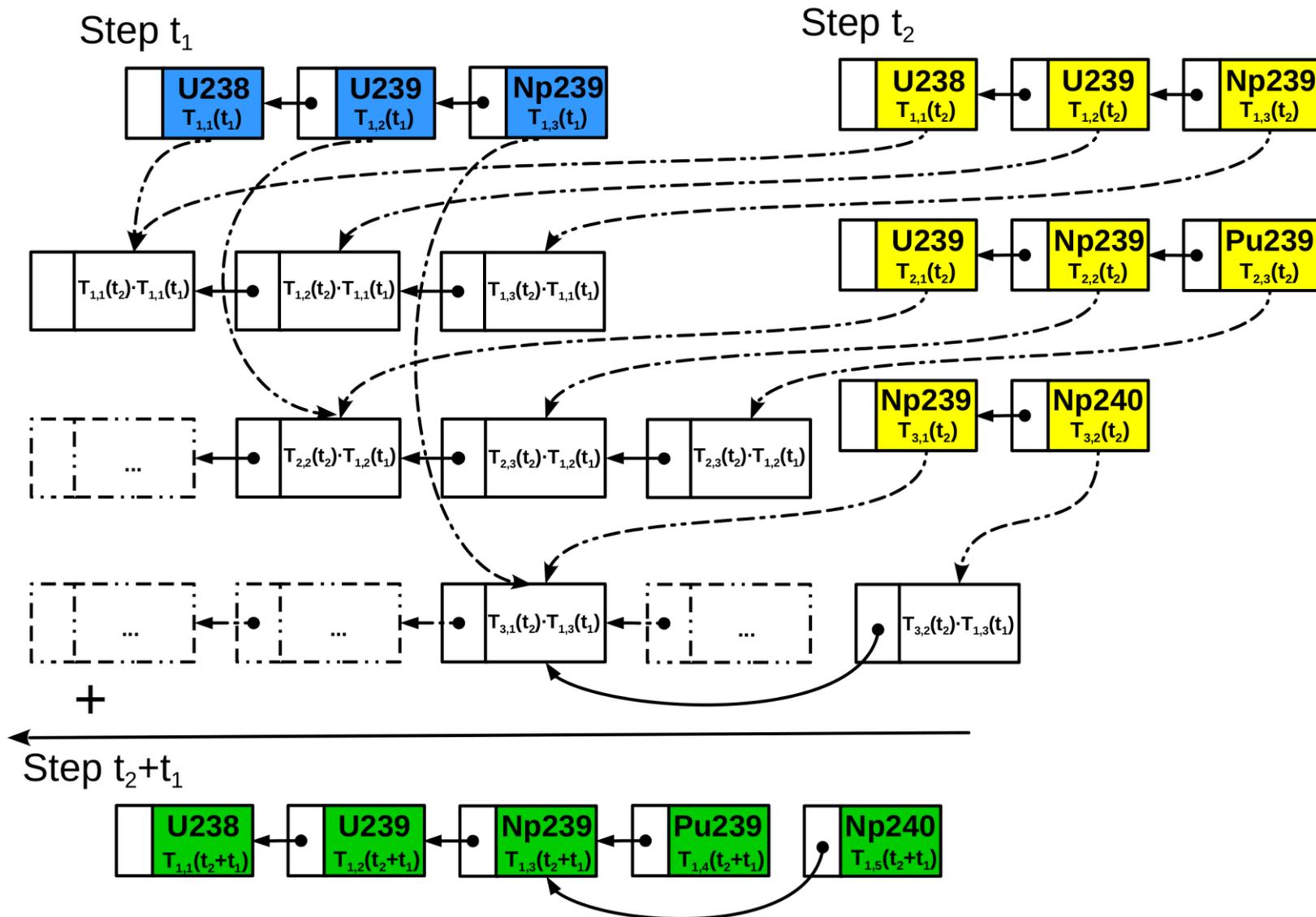
$$\Leftarrow (8, C) * (1, C)$$

$$\Leftarrow (8, C) * (2, D)$$



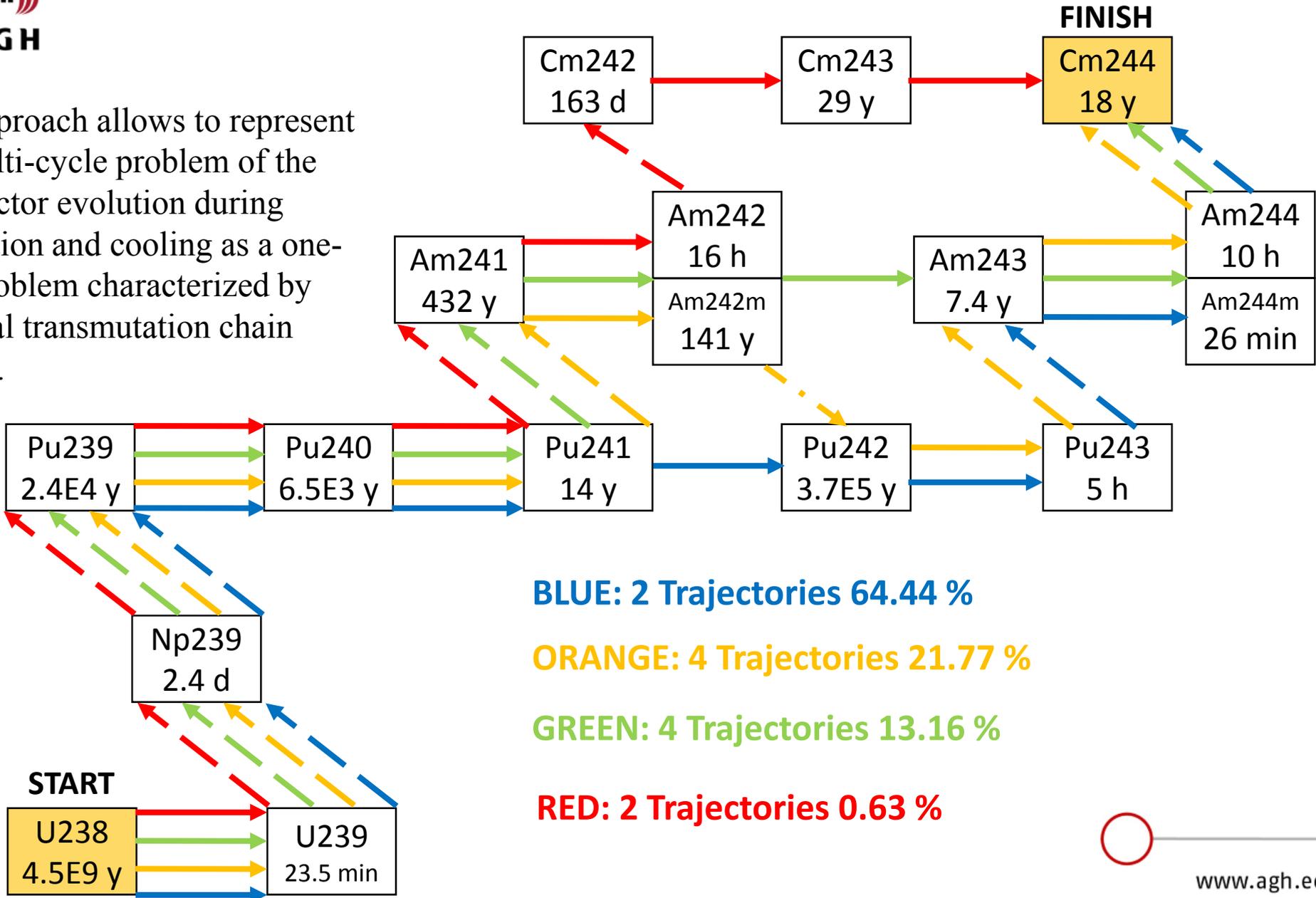
Trajectories at  $t_1$  and  $t_2$

# Parent pointer tree data folding

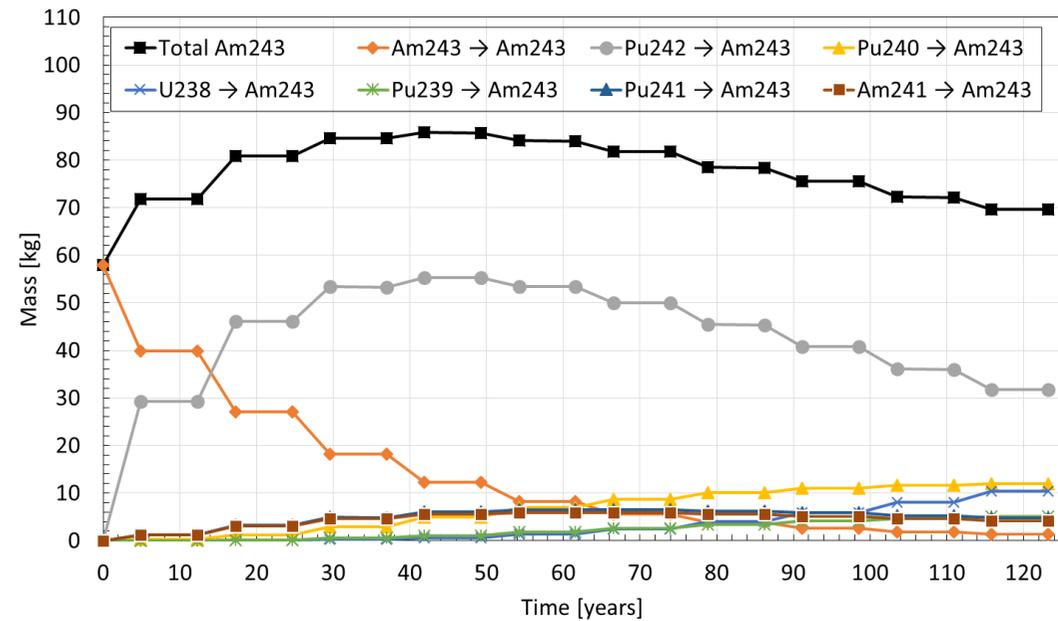
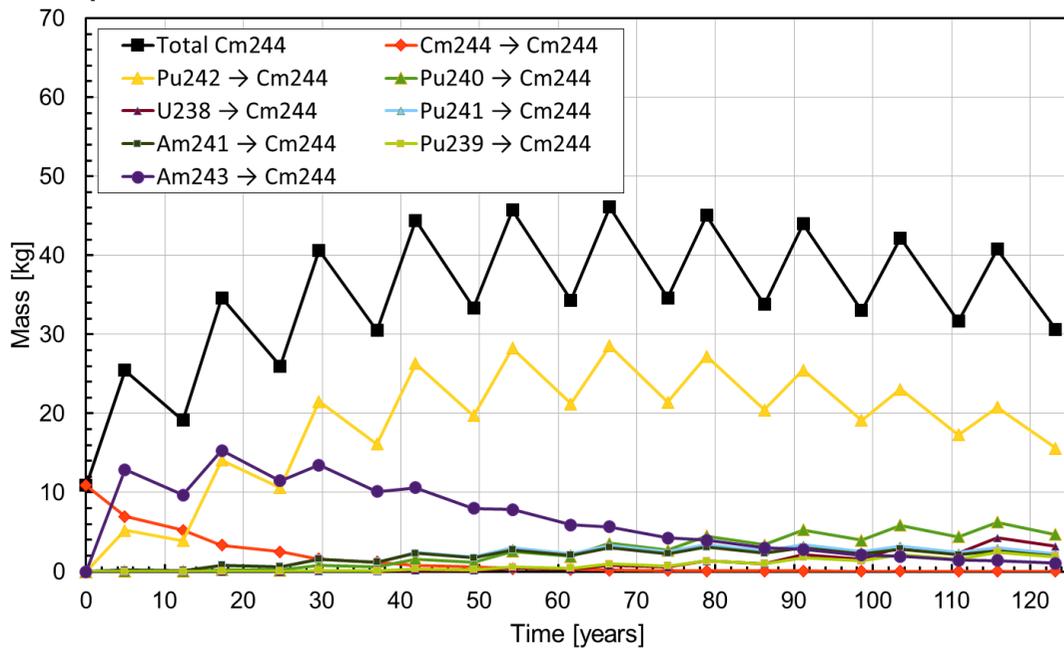


# Application (i)

The approach allows to represent the multi-cycle problem of the fuel vector evolution during irradiation and cooling as a one-step problem characterized by the final transmutation chain system.



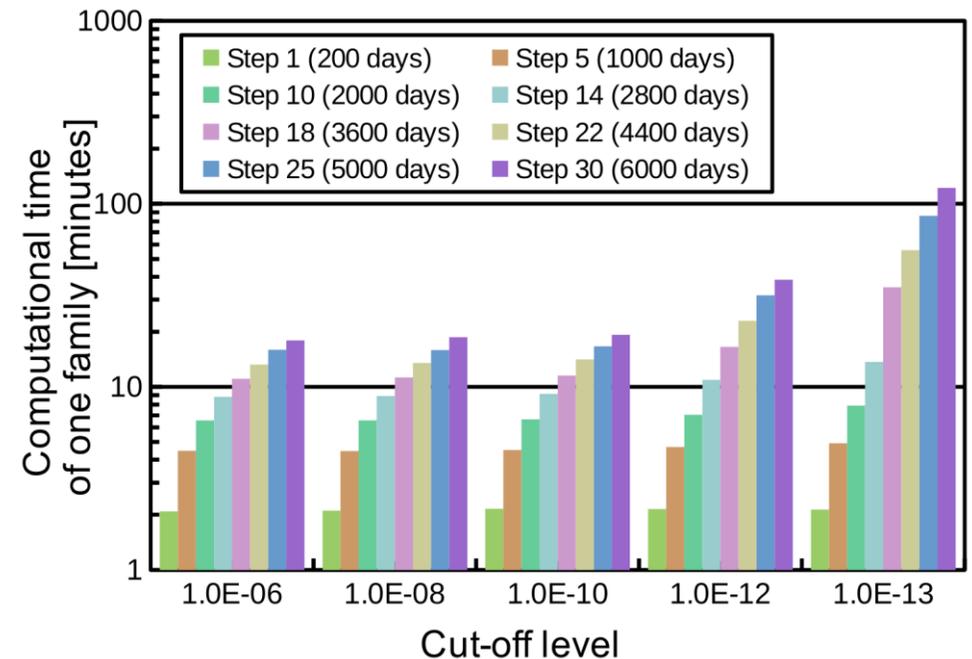
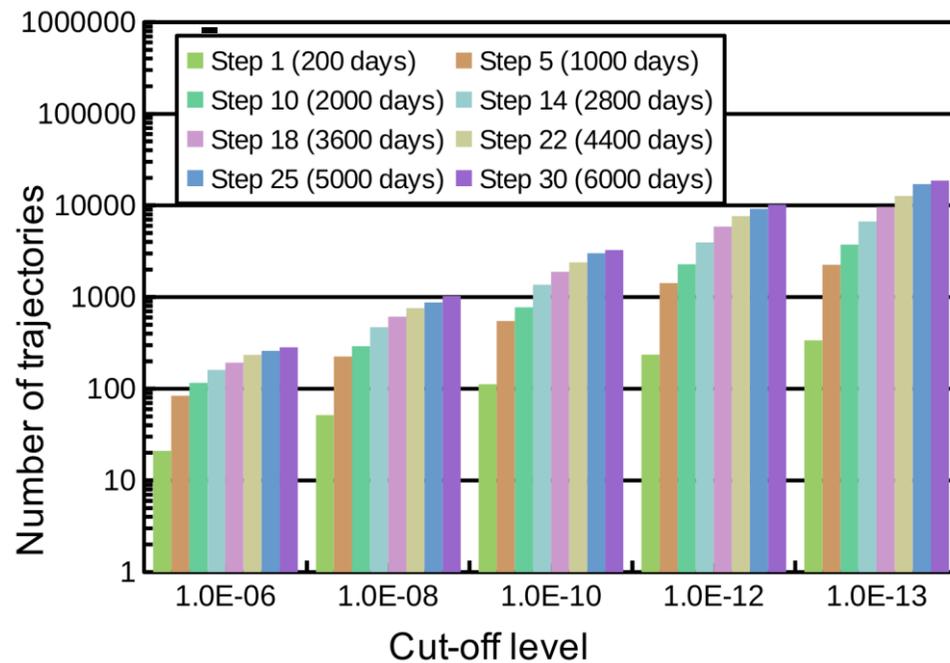
- the TTA folding time procedure can be used to represent single nuclide evolution
- the identification of key trajectories following to formation of transuranic elements is the crucial issues to understand formation mechanism of notably radioactive isotopes, which in turn may help to optimize handling of unloaded fuel before its reprocessing or final disposal,
- the developed numerical method of nuclear transmutation trajectories folding may be used for analysis of any critical or subcritical nuclear system during the arbitrary number of subsequent irradiation or cooling periods,



## Performance simple example

Model of HTR (High Temperature Reactor) assembly:

- representative calculation depends on the cut-off parameter and the size of the considered problem
- reaction rate obtained with Monte Carlo transport
- one initial material Pu239
- one burnup region per one computational node (24 in total)



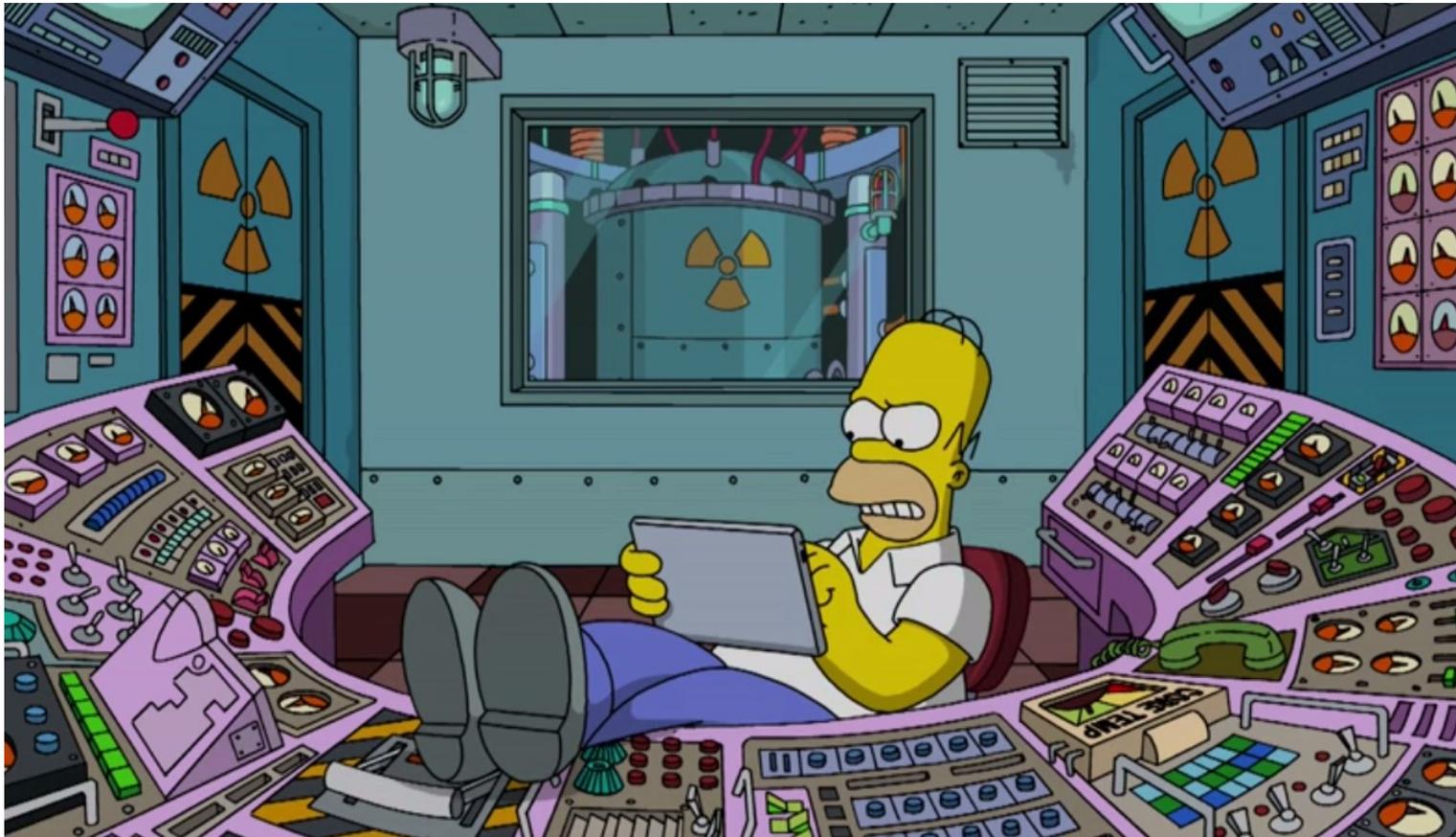
Performance in whole core system:

- mode burnup zones 10 – 1000
- fuel vector 10 – 30 initial material to track
- more computational steps up to 500
- new methodology in recent research in average enlarge computational cost from 1 day to 4 days using 96 cores.

The focal point of the analysis is the implementation and practical application of the novel modules for nuclear transmutation trajectories folding to the MCB code,

the standard solution for the time evolution of nuclide concentrations governed by a set of first-order differential equation may be use to represent the isotope mass evolution as an integrated function for the entire irradiation period including multi-cycling reloading scheme

# Thank you for your attention!



# Hardware - Prometheus



2.4 Pflops

53000 cores

279 TB RAM

10 PB storage

48th position

