Uranium-238 decay chain

Maria Arazo[1], Marc Barroso[1], Óscar De la Torre[1], Laura Moreno[1], Ariadna Ribes[1], Patricia Ribes[1], Ana Ventura[1], and David Orellana-Martín[2]

[1] Universitat de Barcelona
Email: {maria.arazo, marc.barroso4, oscar.delatorre.perez, 95morenolaura, aribesmetidieri, ribesmetidieri, a.venturabarroso}@gmail.com
[2] Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
Universidad de Sevilla
E-mail: dorellana@us.es

Summary. The main objective of this article is to modelize the process of decay of Uranium 238 within the framework of Membrane Computing, so the evolution of great numbers of particles can be progressively followed and the results of the desintegrations (nuclei coming from α and β^- decays) can be counted.

In order to model the process in an accurate manner, exploiting the properties of maximal parallelism and non-determinism of *Membrane Computing*, a Population Dynamic P system (or PDP for short) restricted to one environment and a P system conformed by only the skin have been selected.

The difficulty in the characterisation of this reactions lays in the simultaneity of the different decays, since the number of desintegrations of nucleous of each specie depend on the number of atoms of the initial population. In order to solve this problem and keep their attachment, the characteristic time of production of each decay has been translated into probabilities of deintregration of a nucleous using the decay constant λ.

1 Introduction

In this paper we are considering the Uranium-238 decay, which will be explained in the following sections. One of the first objectives was to prove that making use of *Membrane Computing* and the P-lingua simulation, we could obtain the results previously known, e.g. the ways that intermediate products of the decay took to arrive to the final product or the amount of different elements that were produced during the chain. Nevertheless, during the development of this project another interesting problem, which will be explained and discussed in later on, appeared: time implementation. At this point, our main goal was to look for different ways of modeling the physical process as close as possible to reality. Even so, it is still interesting to know which products we obtain in each disintegration, so we are able to proof the most probable ways of decay.

The problem with half-lifetimes.

During the modeling of the decay processes we found out some problems when implementing the time involving the reactions, i.e. the disintegration of one element into another one has an intrinsic half-lifetime associated to it. This parameter $T_{1/2}$, found in equations (1) and (2), determines the time that takes for the element to reduce the number of its nuclei to half of the initial ones. Also, an important constant is τ, the decay constant, defined as: $\tau = \frac{1}{\lambda}$, which represents the probability of a nucleus to decay, per unit of time.

$$\frac{\mathrm{d}N}{\mathrm{d}t} = -\lambda N \tag{1}$$

$$T_{1/2} = \frac{\ln 2}{\lambda} \tag{2}$$

Being N the number of nuclei at a given time t, and λ, the number of disintegrations per second, which is a constant for a given reaction.

From (1) it can be noticed that the rate of disintegration depends not only on the constant of disintegration, λ, but also on the population of nuclei at the time we are calculating the disintegration rate. This is the reason why rather than considering the rate as the parameter to characterize the reactions, sometimes is better to consider what we define as half-lifetime, $T_{1/2}$, which is constant because it only depends on λ.

These processes occur all at the same time, so to say, from the first moment when we obtain the second nucleus of the chain, another reaction begins to take place: it does not wait for all the first elements to react. Taking into account that P-Systems are based in systems that evolve by steps of time we were aware we had to find a way to approximate as close as we could to the fact that time is continuous. To do so while trying not to differ a lot from what happens in reality we went through different models making some changes in the implementation of the time. The two methods that we selected, which will be further explained in following sections, were the following:

- Steps of time: The first approach to the problem of Uranium decay consisted in translating the half-lifetimes of the different decays of the chain by a logarithmic scale so the considered range of variation was reduced enough in other to assign a proportional and arbitrary amount of time for each step. There it has been considered that one reaction must be applied to all nuclei before beginning the following reaction of the chain. In this case, index notation was used to represent the duration of each step. Although not being a model really close to the real situation, one could obtain the expected results. So, for example, the first reaction was assigned a counter that went from 1 to 7. This counter ensured that no reaction could begin before having ended previously the earlier step in the chain.

  In a way, this process roughly simulated the different periods of time required for each element of the chain to vanish. However it doesn't allow that different elements react at the same time. The next step in the chain has to wait until the previous one finished. Therefore, although the simulation that implements this rules approaches reasonably well the amount of particles gathered at the end of the process, it was not a good approach to reality, as in a real decay several reactions of the chain take place at the same time.

- Probabilistic model: in which it has been taken into account that once a nucleus has decayed into the next one, the following reaction can take place for that recently generated nucleus. This model is a useful way to determine which particles were generated at each moment, i.e. one could thoroughly examine the intermediate stages of the decay.

2 Uranium-238 decay chain

2.1 Radioactive series

Nuclear decays [2] are transitions to less energetic —and thus more stable— states. An initial unstable nucleus can naturally decay into another nucleus, usually but not necessarily lighter, following different modes characterized by the emitted particles and the resultant nuclei. The ones concerning our study are the α decay (3) and the β^- decay (4), where the emitted particles can be He nuclei (α particles) or electrons (along with their corresponding antineutrino).

$$ {}^{A}_{Z}X \to {}^{A-4}_{Z-2}Y + {}^{4}_{2}\mathrm{He} \tag{3}$$

$$ {}^{A}_{Z}X \to {}^{A}_{Z+1}Y + e^- + \bar{\nu}_e \tag{4}$$

Where A is the mass number and Z the atomic number. Other possible decay modes are the β^+ decay (with emission of positrons and electronic neutrinos), the gamma emission, and the electronic capture. Of the three kinds of possible emitted particles (α, β and γ), γ particles have the largest penetrating power, while α particles interact more with matter.

The resulting nuclei of a nuclear decay can still be unstable and therefore decay into another nuclei and the corresponding particle. In this way, several decays may take place until a stable nucleus is reached. This process of chained decays that begins on a unstable parent nucleus and end on a number of stable nuclei is called a *radioactive series* or *decay chain*.

The parent nuclei of the radioactive series usually have very large lifetimes (i.e. the time it takes to the initial population to disappear entirely). There are four main radioactive series (three of them being natural), and all of them end in lead, which is stable.

2.2 U-238 decay chain

As we can see in Table 1, the Uranium decay chain consists of 15 main steps (i.e. decay reactions). This table shows the most probable decay modes, but there are other decays with an extremely low probability of occurring, showed with more detail in the second part of Table 1. Nevertheless, independently from the path chosen, the final product is always lead (Pb-206), which is stable. A diagram of the whole U-238 decay chain and its less probable decay modes can be found in Figure 1.

Parent	$\mathbf{T_{1/2}}$ **(s)**	$\boldsymbol{\lambda}$ **(decays/s)**	$\boldsymbol{\tau}$ **(s)**	**Decay modes**	**Reaction**	$f_{r,1}$
$^{238}_{92}\mathrm{U}$ (a)	1.41×10^{17}	4.92×10^{-18}	2.03×10^{17}	α (100%): $^{234}_{90}\mathrm{Th}$	$[a]_0 \to x[b]_0$	9.68×10^{-1}
$^{234}_{90}\mathrm{Th}$ (b)	2.08×10^{6}	3.33×10^{-7}	3.00×10^{6}	β^- (100%): $^{234}_{91}\mathrm{Pa}$	$[b]_0 \to z[c]_0$	5.33×10^{-1}
$^{234}_{91}\mathrm{Pa}$ (c)	2.41×10^{4}	2.87×10^{-5}	3.48×10^{4}	β^- (100%): $^{234}_{92}\mathrm{U}$	$[c]_0 \to z[d]_0$	6.22×10^{-1}
$^{234}_{92}\mathrm{U}$ (d)	7.74×10^{12}	8.95×10^{-14}	1.12×10^{13}	α (100%): $^{230}_{90}\mathrm{Th}$	$[d]_0 \to x[e]_0$	2.29×10^{-1}
$^{230}_{90}\mathrm{Th}$ (e)	2.38×10^{12}	2.92×10^{-13}	3.43×10^{12}	α (100%): $^{226}_{88}\mathrm{Ra}$	$[e]_0 \to x[f]_0$	2.53×10^{-1}
$^{226}_{88}\mathrm{Ra}$ (f)	5.05×10^{10}	1.37×10^{-11}	7.28×10^{10}	α (100%): $^{222}_{86}\mathrm{Rn}$	$[f]_0 \to x[g]_0$	3.30×10^{-1}
$^{222}_{86}\mathrm{Rn}$ (g)	3.30×10^{5}	2.10×10^{-6}	4.77×10^{5}	α (100%): $^{218}_{84}\mathrm{Po}$	$[g]_0 \to x[h]_0$	5.70×10^{-1}
$^{218}_{84}\mathrm{Po}$ (h)	1.86×10^{2}	3.73×10^{-3}	2.68×10^{2}	α (99.98%): $^{214}_{82}\mathrm{Pb}$ β^- (0.02%): $^{218}_{85}\mathrm{At}$	$[h]_0 \to x[v]_0$ $[h]_0 \to z[j]_0$	7.20×10^{-1} 1.44×10^{-4}
$^{214}_{82}\mathrm{Pb}$ (v)	1.62×10^{3}	4.27×10^{-4}	2.34×10^{3}	β^- (100%): $^{214}_{83}\mathrm{Bi}$	$[v]_0 \to z[k]_0$	6.77×10^{-1}
$^{214}_{83}\mathrm{Bi}$ (k)	1.19×10^{3}	5.81×10^{-4}	1.72×10^{3}	β^- (99.979%): $^{214}_{84}\mathrm{Po}$ α (0.021%): $^{210}_{81}\mathrm{Tl}$	$[k]_0 \to x\,[n]_0$ $[k]_0 \to z\,[m]_0$	1.43×10^{-4} 6.83×10^{-1}
$^{214}_{84}\mathrm{Po}$ (m)	1.64×10^{-4}	4.22×10^{3}	2.37×10^{-4}	α (100%): $^{210}_{82}\mathrm{Pb}$	$[m]_0 \to x[p]_0$	1.00×10^{0}
$^{210}_{82}\mathrm{Pb}$ (p)	7.00×10^{8}	9.90×10^{-10}	1.01×10^{9}	β^- (100%): $^{210}_{83}\mathrm{Bi}$ α (1.9×10^{-6}%): $^{206}_{80}\mathrm{Hg}$	$[p]_0 \to x\,[o]_0$ $[p]_0 \to z\,[q]_0$	7.90×10^{-9} 4.16×10^{-1}
$^{210}_{83}\mathrm{Bi}$ (q)	4.33×10^{5}	1.60×10^{-6}	6.25×10^{5}	β^- (100%): $^{210}_{84}\mathrm{Po}$ α (13.2×10^{-5}%): $^{206}_{81}\mathrm{Tl}$	$[q]_0 \to x\,[s]_0$ $[q]_0 \to z\,[r]_0$	7.45×10^{-7} 5.64×10^{-1}
$^{210}_{84}\mathrm{Po}$ (r)	1.20×10^{7}	5.80×10^{-8}	1.72×10^{7}	α (100%): $^{206}_{82}\mathrm{Pb}$	$[r]_0 \to x[t]_0$	4.98×10^{-1}
$^{206}_{82}\mathrm{Pb}$ (t)	Stable	—	—	—	—	—

Other (less probable) decays

$^{218}_{85}\mathrm{At}$ (j)	1.50	4.62×10^{-1}	2.16	α (99.9%): $^{214}_{83}\mathrm{Bi}$ β^- (0.1%): $^{218}_{86}\mathrm{Rn}$	$[j]_0 \to x\,[k]_0$ $[j]_0 \to z\,[l]_0$	8.16×10^{-1} 8.17×10^{-4}
$^{218}_{86}\mathrm{Rn}$ (l)	3.50×10^{-2}	1.98×10^{1}	5.05×10^{-2}	α (100%): $^{214}_{84}\mathrm{Po}$	$[l]_0 \to x[m]_0$	8.92×10^{-1}
$^{210}_{81}\mathrm{Tl}$ (n)	7.80×10^{1}	8.89×10^{-3}	1.13×10^{2}	β^- (100%): $^{210}_{82}\mathrm{Pb}$	$[n]_0 \to z[p]_0$	7.38×10^{-1}
$^{206}_{80}\mathrm{Hg}$ (o)	4.99×10^{2}	1.39×10^{-3}	7.20×10^{2}	β^- (100%): $^{206}_{81}\mathrm{Tl}$	$[o]_0 \to z[s]_0$	7.00×10^{-1}
$^{206}_{81}\mathrm{Tl}$ (s)	2.52×10^{2}	2.75×10^{-3}	3.64×10^{2}	β^- (100%): $^{206}_{82}\mathrm{Pb}$	$[r]_0 \to x[t]_0$	4.98×10^{-1}

Table 1: Half-life times, decay constants, mean lifetimes, decay modes, reaction and probability functions for each reaction in the U-238 decay chain. The letter in brackets corresponds to the letter assigned to each nucleus for implementation. Half-lifes and probabilities for the chain decays obtained from [3] [5]; probabilities for the less probable decays obtained from [4]. The probability function associated to each transformation rule depends on the decay constant and the decay mode probability.

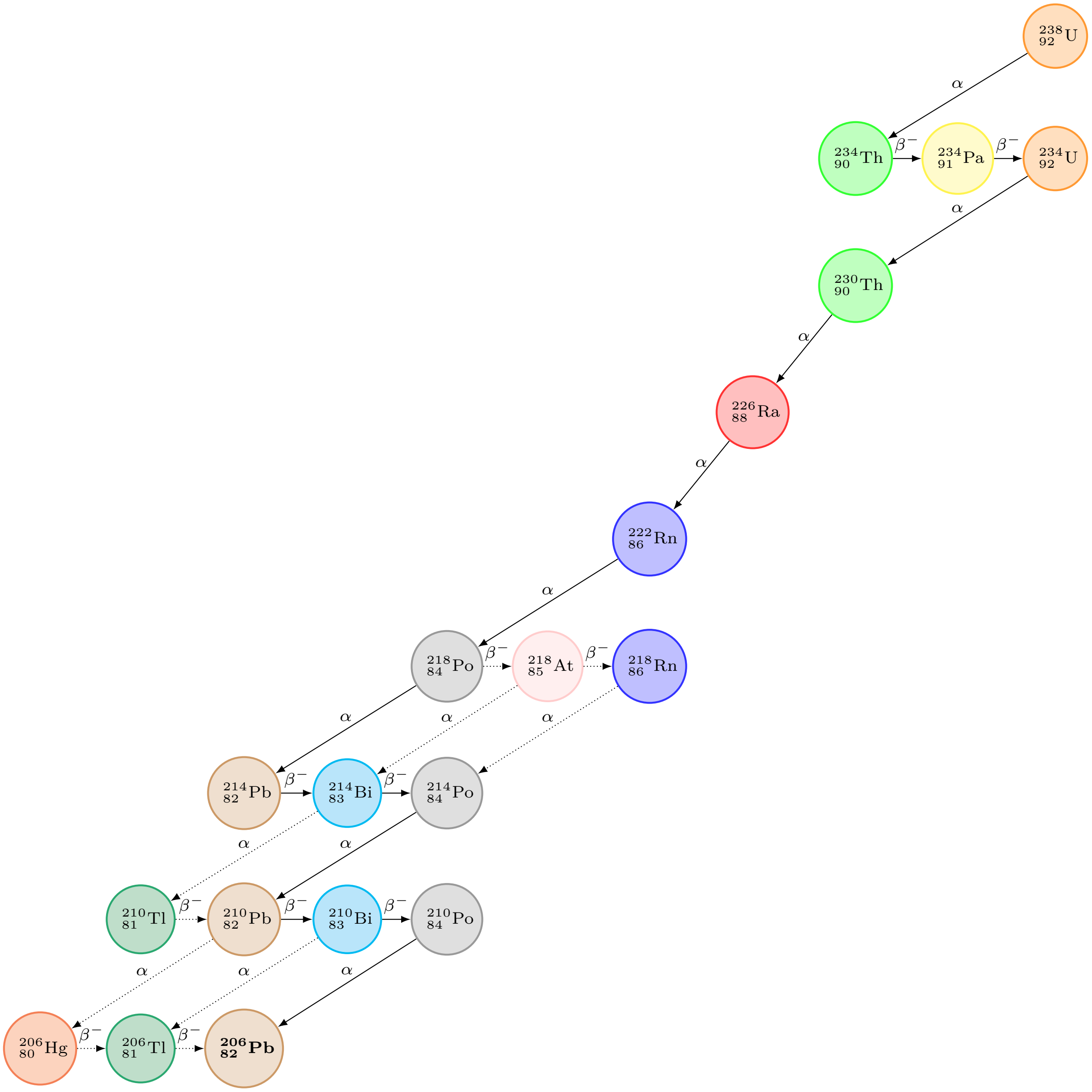


Fig. 1: Uranium decay chain.

3 P-system model

Population Dynamics P systems (PDP systems) [1] are a kind of P systems that combines the characteristics of both cell-like and tissue-like models. A PDP system is constituted by (i) a set of connected environments placed in the nodes of a directed graph (ii) identical cell-like structures of hierarchically arranged membranes placed inside each environment, (iii) a working alphabet of objects and (iv) a set of rules which describe how objects evolve and move inside the P systems ($\mathcal{R}$) and among the environments ($\mathcal{R}_\varepsilon$).

Formally, a Population Dynamics P system of degree (q,m) with $q, m \geq 1$, taking T time units, $T \geq 1$, is a tuple

$$(G, \mathit{\Gamma}, \mathit{\Sigma}, T, \mathcal{R}_\varepsilon, \mu, \mathcal{R}, \{f_{r,j} : r \in \mathcal{R}, 1 \leq j \leq m\}, \{\mathcal{M}_{ij} : 1 \leq i \leq q, 1 \leq j \leq m\}) \tag{5}$$

where:

- $G = (V, S)$ is a directed graph and $V = \{e_1, \ldots, e_m\}$ are the elements called environments.
- $\mathit{\Gamma} \cup \mathit{\Sigma}$ is the working alphabet.
- T is a natural number that represents the simulation time of the system.
- $\mathcal{R}_\varepsilon$ is a set of communication rules between environments of the form
  $$(x)_{e_j} \xrightarrow{p(x,j,j_1,\ldots,j_h)} (y_1)e_{j1} \cdots (y_h)_{e_{jh}} \tag{6}$$
  where $x, y_1, \cdots y_h \in \mathit{\Gamma}, (e_j, e_{jl}) \in S(l = 1, \ldots, h)$ and $p_{(x,j,j1,\ldots,j_h)}(t) \in [0,1]$, for each $t = 1, \ldots T$.
  The previous definition means that, when a communication rule is applied, object x contained in environment e_j passes to environments $e_{j1} \ldots e_{jh}$, possibly modified into objects $y_1, \ldots y_h$. If more than one rule can be applied to $(x)_{e_j}$, then the rule executed is chosen randomly according to the probabilities $p(x, j, j_1, ..., j_h)$.
- μ is the membrane structure of the cells contained in each of the m environments and each consisting on a set of q hierarchically arranged membranes injectively labeled by $1, \ldots q$. The skin membrane, or outer membrane is labeled by 1. The membranes can also have electrical charges or polarizations, $EC = \{0, +, -\}$.
- $\mathcal{R}$ is a set of evolution rules applied within each cell. They are of the form $r : u[v]_i^\alpha \rightarrow u'[v']_i^{\alpha'}$ where $u, v, u', v' \in M(\mathit{\Gamma})$, $i \in 1, \ldots q$, and $\alpha' \in EC$.
- For each $r \in \mathcal{R}$ and for each $j, 1 \leq j \leq m$, $f_{r,j}$ is a computable function which satisfies that, for each $u, v \in M(\mathit{\Gamma})$ all the rules $r \in \mathcal{R}$ whose left-hand side is (i, α, u, v) and the right-hand side have a polarization α', $\sum_{j=1} f_{r,j}(t) = 1$ $\forall t \leq T$.
- $\mathcal{M}_{1j}, \ldots, \mathcal{M}_{qj} \in M(\mathit{\Gamma})$ are the initial multisets of objects for environments $j = 1, \ldots m$ placed inside the membranes $1, \ldots q$ of μ.

The tuple of multisets of objects present at any moment in the m environments and at each of the regions of the P systems (cell-like structures) constitutes the a configuration of the system at any time. At the initial configuration of the system, all environments are assumed to be empty and all the membranes have neutral polarization.

The system evolves from one configuration to another at each time step by executing simultaneously all the applicable rules of the set $\mathcal{R} = \mathcal{R}_\varepsilon \cup \bigcup_{i=1}^{m} \mathcal{R}_{\Pi_j}$[3] in a maximal way. When there are rules acting on overlapping left-hand sides, i.e. $u[v]_i^\alpha$, $u'[v']_i^\alpha$ where $u, u', v, v' \in M(\Gamma)$, $u \neq u' \vee v \neq v'$ and $u \cap u' \neq \emptyset \vee v \cap v' \neq \emptyset$, the rule which is executed is selected randomly according to the probability associated with each rule.

Finally, it is interesting to highlight the fact that a global clock is considered in the system, marking the time for the whole system, so the application of all rules (both from $\mathcal{R}_\varepsilon$ and $\mathcal{R}$) are synchronized in all environments.

[3] $\Pi_j = \{\Gamma, \mu, \mathcal{R}, \mathcal{M}_{1j}, \ldots, \mathcal{M}_{qj}\}$ denotes the P system in environment e_j and R_{Π_j}, the set of rules defined on the considered P system.

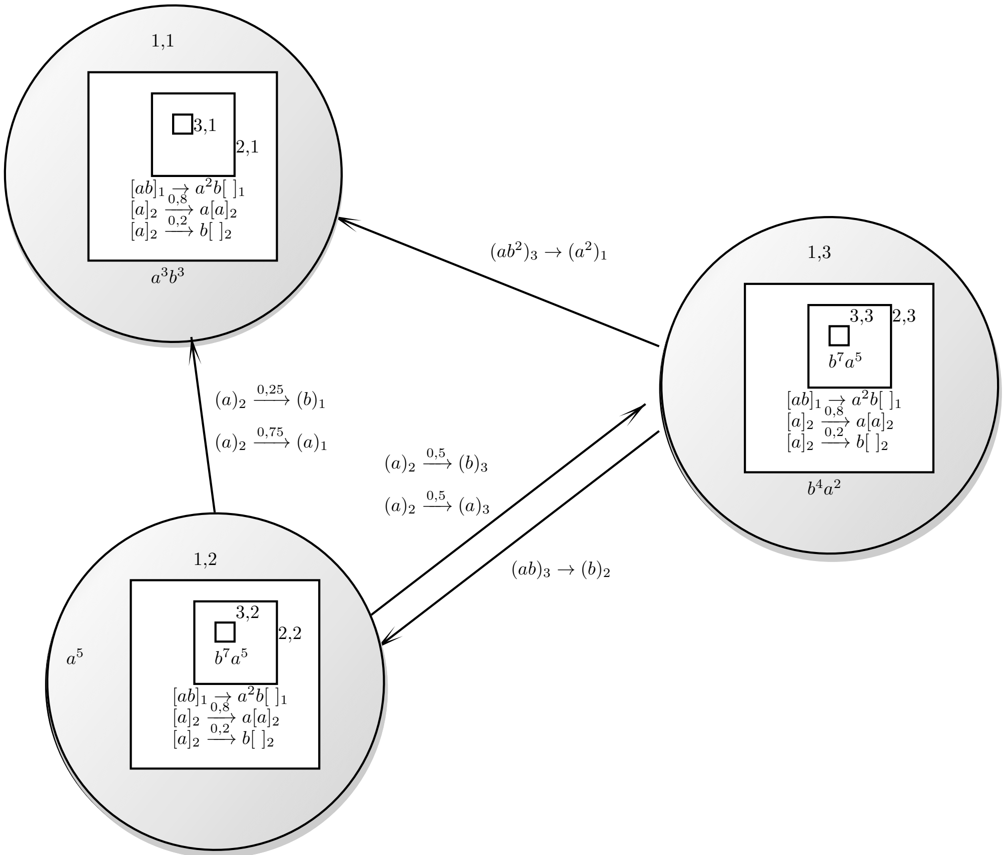

1,1
3,1
2,1
$[ab]_1 \rightarrow a^2b[\ ]_1$
$[a]_2 \xrightarrow{0,8} a[a]_2$
$[a]_2 \xrightarrow{0,2} b[\ ]_2$
a^3b^3
$(ab^2)_3 \rightarrow (a^2)_1$
1,3
3,3
2,3
b^7a^5
$[ab]_1 \rightarrow a^2b[\ ]_1$
$[a]_2 \xrightarrow{0,8} a[a]_2$
$[a]_2 \xrightarrow{0,2} b[\ ]_2$
b^4a^2
$(a)_2 \xrightarrow{0,25} (b)_1$
$(a)_2 \xrightarrow{0,75} (a)_1$
$(a)_2 \xrightarrow{0,5} (b)_3$
$(a)_2 \xrightarrow{0,5} (a)_3$
$(ab)_3 \rightarrow (b)_2$
1,2
3,2
2,2
b^7a^5
a^5
$[ab]_1 \rightarrow a^2b[\ ]_1$
$[a]_2 \xrightarrow{0,8} a[a]_2$
$[a]_2 \xrightarrow{0,2} b[\ ]_2$

Fig. 2: A graphical example of a PDP system

4 Implementation

4.1 First model

The Step of time model is the simplest possible modelization of the Uranium decay problem, as it characterizes the half-lifetime, the characteristic time parameter, of the nuclear reactions using the clock steps of time defined in the membrane computing model. This first rought approximation is based on two assumptions:

- The nuclear reaction of a given element cannot begin until all the progenitor nuclei of this element have reacted.
- The duration of each reaction can be represented assigning different clock steps to every reaction.

The timescale of the reactions (characterized by the half-lifetime, $T_{1/2}$) involved in the network of nuclear reactions is huge, varying 21 orders of magnitude: the $T_{1/2}$ of the fastest reaction is of about $\approx 10^{-4}$ s while for the slowest $T_{1/2} \approx 10^{17}$ s.

In order to translate the half-lifetime of each reaction to a number of clock iterations, a logarithmic scale is considered. The number of clock iterations is therefore calculated assigning a scaled integer number to each reaction according to the "weight" of time for each reaction.

The implementation of this model is done therefore in a single cell-like membrane through rules of the type

$$\begin{aligned}
&[a_i \rightarrow\ a_{i+1}\ 1 \leq i \leq 7]_1\\
&[a_8 \rightarrow\ b_1, z]_1\\
&[b_i \rightarrow\ b_{i+1}\ 1 \leq i \leq 4]_1\\
&[b_5 \rightarrow\ c_1, x, y]_1
\end{aligned}$$

when there's a single via decay or

$$\begin{aligned}
&[h_i \xrightarrow{\quad} h_{i+1}\ 1 \leq i \leq 3]_1\\
&[h_4 \xrightarrow{99{,}98\%} v_1, z]_1\\
&[h_4 \xrightarrow{0{,}02\%} j_1, x, y]_1
\end{aligned}$$

when competition rules are considered.

The letter assignation is specified at Table 1, as it is the same as the one used in the probabilistic model. x, y and z represent α particles, e^- (electrons) and $\overline{\nu}_e$ (electron antineutrino).

As can be seen, after seven clock steps of computation the nuclei a decays into b. The nuclei b then waits 4 steps of computation before evolving. When competition rules apply, the nuclei h also waits a given number of computations after evolving according to a given probability.

Although it is not being a model really close to the real situation, with it one could obtain the results expected. The main application of the *Steps of time* model is then, to obtain the number of nuclei and particles of each kind once the process has ended, taking into account that some nuclei can decay by different vias according to a given probability. However, it cannot predict the number of nuclei of each kind after a given amount of time. The representation of the model is staggered-like, so it doesn't represent a continuous and soft process and therefore the approximation is not accurate.

To sum up, the biggest disadvantage in this modelization was that the next step in the chain had to wait until the previous one had finished completely. Therefore, although the simulation which implemented these rules approached reasonably the amount of particles gathered at the end of the process, it wasn't a good approach to reality, as in a real decay several reactions of the chain take place at the same time.

4.2 Second model

In order to implement the experiment of the Uranium Decay using the framework of *Membrane Computing*, it has been chosen the PDP system model restricted to a single environment containing a membrane structure composed by a sole membrane of neutral polarization $\alpha = 0$, so the system can be described as a P system

$$\Pi_1 = (\Gamma = \{a, b ... z\}, \mu = [\ ], \mathcal{R} = \{r_{1,i}, i = 1, ..., 38\}, \mathcal{M}_{1,q=1} = a^n)$$

Within this membrane, different objects, which represent the intermediate products in the decay chain, evolve in each step of the computation using PDP evolution rules of the type described in the previous section.

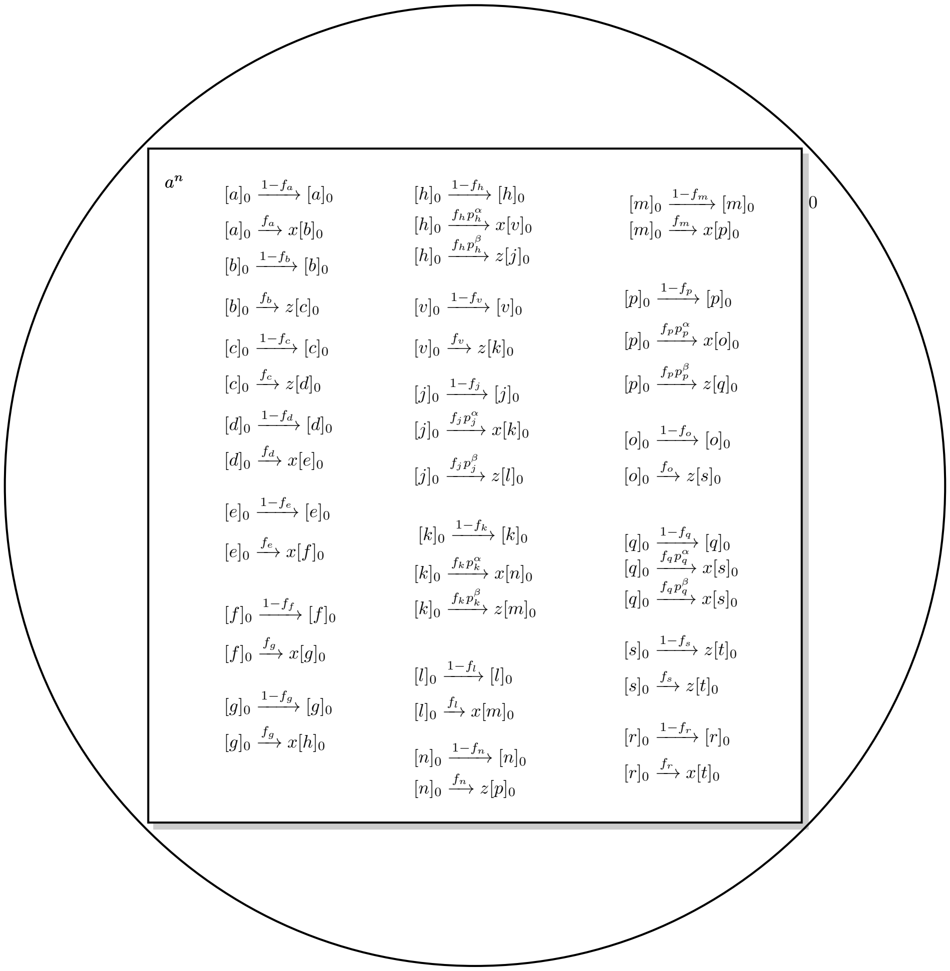
The alphabet of objects Γ is composed by all the intermediate products described in Table 1. A letter has been associated to each decay product in order to enable an easier modeling of the problem. As discussed in section 2, each decay mode α or β^- generates a different kind of particles, which also have a letter associated (x for α particles and z for particles generated in β^- decay), so the total number of particles obtained from the α and β^- decay can be accounted at the end of the computation.

In order to model the smooth and continuous decay of every specie in time, competition rules have been considered. For example, given the first reaction, the decay of U_{92}^{238} into Th_{90}^{234} through the α mode.

$$[a]_1 \xrightarrow{1-f_a} [a]_1$$
$$[a]_1 \xrightarrow{f_a} x[b]_1$$

where a and b are the letters associated to U_{92}^{238} and Th_{90}^{234} respectively. The rule applied will be chosen taking into account the probability associated with it, denoted by $f_{a,1} \equiv f_a$. From this it can be seen why it is so important that the probability sums up to 1.

The decay of a into b takes place through the α decay mode, so a second product x (α particle) is generated outside the membrane.



$[a]_0 \xrightarrow{1-f_a} [a]_0$

$[a]_0 \xrightarrow{f_a} x[b]_0$

$[b]_0 \xrightarrow{1-f_b} [b]_0$

$[b]_0 \xrightarrow{f_b} z[c]_0$

$[c]_0 \xrightarrow{1-f_c} [c]_0$

$[c]_0 \xrightarrow{f_c} z[d]_0$

$[d]_0 \xrightarrow{1-f_d} [d]_0$

$[d]_0 \xrightarrow{f_d} x[e]_0$

$[e]_0 \xrightarrow{1-f_e} [e]_0$

$[e]_0 \xrightarrow{f_e} x[f]_0$

$[f]_0 \xrightarrow{1-f_f} [f]_0$

$[f]_0 \xrightarrow{f_g} x[g]_0$

$[g]_0 \xrightarrow{1-f_g} [g]_0$

$[g]_0 \xrightarrow{f_g} x[h]_0$

$[h]_0 \xrightarrow{1-f_h} [h]_0$

$[h]_0 \xrightarrow{f_h p_h^\alpha} x[v]_0$

$[h]_0 \xrightarrow{f_h p_h^\beta} z[j]_0$

$[v]_0 \xrightarrow{1-f_v} [v]_0$

$[v]_0 \xrightarrow{f_v} z[k]_0$

$[j]_0 \xrightarrow{1-f_j} [j]_0$

$[j]_0 \xrightarrow{f_j p_j^\alpha} x[k]_0$

$[j]_0 \xrightarrow{f_j p_j^\beta} z[l]_0$

$[k]_0 \xrightarrow{1-f_k} [k]_0$

$[k]_0 \xrightarrow{f_k p_k^\alpha} x[n]_0$

$[k]_0 \xrightarrow{f_k p_k^\beta} z[m]_0$

$[l]_0 \xrightarrow{1-f_l} [l]_0$

$[l]_0 \xrightarrow{f_l} x[m]_0$

$[n]_0 \xrightarrow{1-f_n} [n]_0$

$[n]_0 \xrightarrow{f_n} z[p]_0$

$[m]_0 \xrightarrow{1-f_m} [m]_0$

$[m]_0 \xrightarrow{f_m} x[p]_0$

$[p]_0 \xrightarrow{1-f_p} [p]_0$

$[p]_0 \xrightarrow{f_p p_p^\alpha} x[o]_0$

$[p]_0 \xrightarrow{f_p p_p^\beta} z[q]_0$

$[o]_0 \xrightarrow{1-f_o} [o]_0$

$[o]_0 \xrightarrow{f_o} z[s]_0$

$[q]_0 \xrightarrow{1-f_q} [q]_0$

$[q]_0 \xrightarrow{f_q p_q^\alpha} x[s]_0$

$[q]_0 \xrightarrow{f_q p_q^\beta} x[s]_0$

$[s]_0 \xrightarrow{1-f_s} z[t]_0$

$[s]_0 \xrightarrow{f_s} z[t]_0$

$[r]_0 \xrightarrow{1-f_r} [r]_0$

$[r]_0 \xrightarrow{f_r} x[t]_0$

Fig. 3: Implementation of the model with the rules applied

As λ is a physical parameter constant in time and characteristic of each reaction, it has proved to be the most suitable magnitude to compute the the probability functions $f_{r,1}$. The functions $f_{r,1}$ have been calculated as the normalized logarithm of the scaled (> 1) λ 's.

The most significant difficulty in the assignation of $f_{r,1}$ to every transformation rule consisted on the great order of magnitude of the times of decay of the different reagents, as $\lambda \in [5{\cdot}10^{18}, 5{\cdot}10^{-3}]$ decays/s (nearly 21 orders of magnitude). Moreover, in other to compute the probabilities of each reaction, λ needed to be dimensionless and normalized in such a way that the probabilities of the rules with the same left-hand objects sum up to 1.

In order to solve these problems, some assumptions and approximations have been made. First of all, all values of λ in Table 1 have been divided between the order of magnitude of minimum λ. This way, we ensure that when the logarithm is applied the numbers will always be positive (as $\frac{\lambda}{\lambda_{\mathrm{min}}} > 1$), as well as granting that in logarithm scale, the order of the numbers is not changed. As have been mentioned, having such a great range of λ is truly inconvenient for doing the computations, so the rough solution that has been found consists in applying logarithms to the dimensionless quotient $\frac{\lambda}{\lambda_{min}}$. In other to obtain a global probability, which assigns probability equal to one to the rule with the greatest chance to occur, $\log(r)$ is divided by $\log(r)_{max}$ for every rule. When the decay mode is not unique, this function is also multiplied by the probability of occurrence p_r^γ, with $\gamma = \{\alpha, \beta\}$ representing the possible decay modes.

The method used for obtaining $f_{r,1}$ explained above, assigns a probability to the evolving transformation rules, which transforms a reactive into a different object. As has been explained in section 3, the sum of all probabilities applied over the same left-hand object must be one. As a consequence, the probability of occurrence of the non-evolution transformation rule r: $[a]_0 \xrightarrow{1-f_{r,1}} [a]_1$ has been computed as $1 - f_{r,1}$.

The rules start applying when at least one object of the left-hand side of a rule is generated and continue applying until all this kind of objects are consumed. So, as more intermediate products are generated, more reagents begin to evolve in each computation step, so after a given number of time steps several products of the decay chain will be evolving at the same time (modeled as discrete clock steps in the computation). In this manner, the dependence of the decay in the abundances of each reactive has been roughly simulated in a first approximation. Moreover, the problem of the first model, when a reactive couldn't begin to evolve until all reagents of the previous decay have been consumed has been solved. It's necessary to notice that although this model approximates better the decay as a continuous process, the steps of time are still discrete, represented by each computation step.

5 Code

```
@model<probabilistic>
def main()
{
  @mu=[]’1;
  @ms(1) = a*500000;

  [a]’1 --> [a]’1 ::  0.968;
  [a]’1 --> x[b]’1 :: 0.032;

  [b]’1 --> [b]’1 :: 0.467 ;
  [b]’1 --> z[c]’1 :: 0.533 ;
```

```
[c]’1 --> [c]’1 :: 0.378  ;
[c]’1 --> z[d]’1 :: 0.622 ;

[d]’1 --> [d]’1 ::  0.771;
[d]’1 --> x[e]’1 ::  0.229;

[e]’1 --> [e]’1 ::  0.747;
[e]’1 --> x[f]’1 ::  0.253;

[f]’1 --> [f]’1 ::  0.67;
[f]’1 --> x[g]’1 ::  0.33;

[g]’1 --> [g]’1 ::  0.43;
[g]’1 --> x[h]’1 ::  0.57;

[h]’1 --> [h]’1 ::  0.28;
[h]’1 --> x[v]’1 ::  0.71856;
[h]’1 --> z[j]’1 ::  0.00144;

[v]’1 --> [v]’1 ::  0.323;
[v]’1 --> z[k]’1 :: 0.677;
[j]’1 --> [j]’1 ::  0.183;
[j]’1 --> x[k]’1 ::  0.816183;
[j]’1 --> z[l]’1 ::  0.000817;

[k]’1 --> [k]’1 ::  0.317;
[k]’1 --> x[n]’1 ::  0.68285657;
[k]’1 --> z[m]’1 ::  0.00014343;
[l]’1 --> [l]’1 ::  0.108;
[l]’1 --> x[m]’1 ::  0.892;

[n]’1 --> [n]’1 :: 0.262 ;
[n]’1 --> z[p]’1 ::  0.738;
[m]’1 --> [m]’1 ::  0.0;
[m]’1 --> x[p]’1 ::  1.0;

[p]’1 --> [p]’1 ::  0.584;
[p]’1 --> x[o]’1 ::  0.4159999921;
[p]’1 --> z[q]’1 ::  0.000000007904;

[o]’1 --> [o]’1 ::  0.30;
[o]’1 --> z[s]’1 ::  0.70;
[q]’1 --> [q]’1 ::  0.436;
[q]’1 --> x[s]’1 ::  0.563999255;
[q]’1 --> z[r]’1 ::  0.000000745;

[s]’1 --> [s]’1 :: 0.286 ;
[s]’1 --> z[t]’1 ::  0.714;
```

```
  [r]'1 --> [r]'1 ::  0.502;
  [r]'1 --> x[t]'1 ::  0.498;
}
```

6 Results

The results obtained when running the P-lingua code with $a = 5 \cdot 10^5$ are shown in Figure 4 and Figure 5.

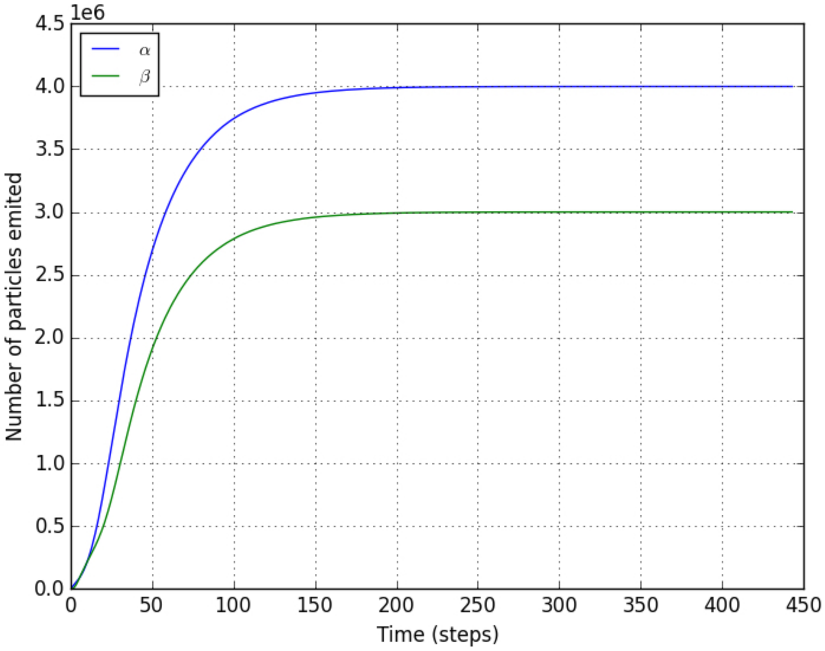


Fig. 4: Number of particles emitted through the different reactions, in logarithmic scale. It can be seen how both particles, α and the ones emitted through β^- reactions (named as β) reach an almost stationary value, with a larger final number of α particles, since α reactions take place more frequently. As reactions begin, a lot of the initial particles evolve giving their products but, as more reactions get to the final product, lead, less reactions take place at each step, so the number of particles emitted at each step reduces considerably, changing only one particle per step so, given the scale of the figure, this becomes imperceptible.

Both figures show the expected behavior of the Uranium decay chain, as they satisfy that

- after a large enough number of computation steps, all the emmited particles in the reaction (α and β^- particles) have reached a stable state.
- the decay chain takes place in a staggered way (we cannot see the discrete increments), as was sought in this second model.
- the slope of the evolution of each product matches the λ coefficients we implemented, showing a correct relationship between the probability functions and λ.
- the process is faster at generating particles in the beginning and at the end of the reaction, whereas the middle products last for a while.

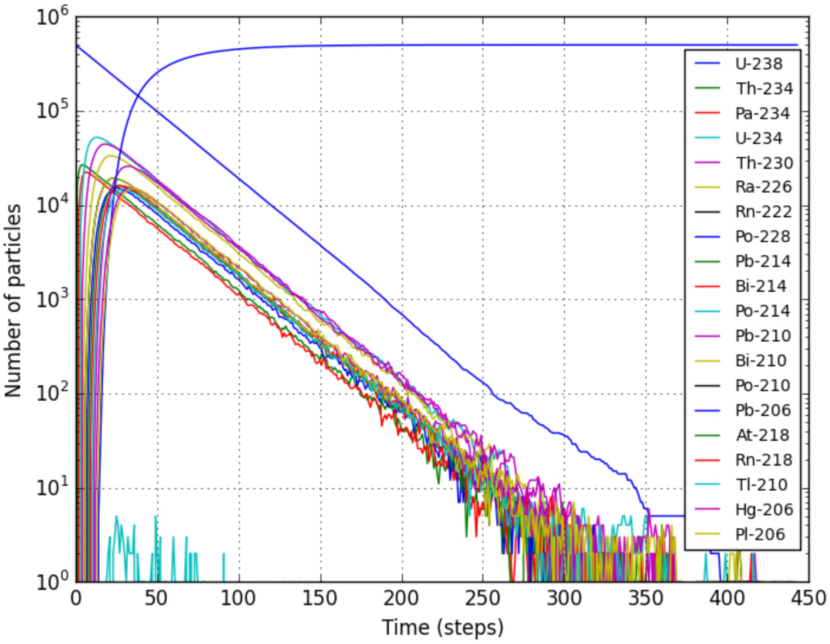


Fig. 5: Nuclei population. It is represented the number of particles at each step of time, in logarithmic scale. It can be seen that the number of initial particles (Uranium-238) decreases whereas the final product increses until it reaches an stationary value, the same as the initial number of particles, as expected. As reactions take place, the new elements are created, showing an impressive increase that slowly decreases then as reactions continue. As we reach more advanced stages (Steps of time ≈ 250 and more) a noise in the number of elements appears: this is because less reactions of the same elements take place simultaneously and so the number of particles changes sharply at each step, depending on if the reaction involving that specific element has taken place or not.

The fluctuations, which can be appreciated in the last steps of time in Figure 5, are probably due to the overlapping of different reactions in the advanced steps of the uranium chain which simultaneously produce and consume a certain nucleus, i.e., as the products evolve, more reactives are generated at the same time that they are evolving to the next product. That way, a dependence on the decay rate on the relative abundances of each reactive is appreciated.

7 Conclusions

Decay chains are based in a system of differential equations that once solved allow to obtain the products at each time, t. However, the solution is reached after solving a coupled system of numerous differential first order equations, which is computationally costly. The *MC* tools allow to reproduce the process and to obtain the expected final products just by making some slight approximations.

This means that competition rules which appear naturally in *MC* can assume the role of the bounds between differential equations almost trivially, so the mentioned system of differential equations does not need to be solved in order to simulate the real situation.

In addition, this article attempts different ways of implementing time in a decay process. Instead of modeling time as an independent parameter, which would be the model where indexes are used (considering steps of computation as time), it has finally been introduced as a part of the probability, given by the decay constant λ. The first method is really unefficient because the system wastes a lot of time just skipping processes (while indexes change) and so by this time, the program is not really working on the chain reaction itself.

Acknowledgments.

We would like to thank: Agustín Riscos-Núñez, Carmen Graciani and Mario J. Pérez-Jiménez from Universidad de Sevilla and Ricardo Graciani from Universitat de Barcelona for their valuable comments and suggestions, as well as for their reviews of the several versions of this paper; the 14th BWMC attendants for their patience and advice on our first ideas; and also Francesc Salvat, Assumpta Parreño, Ricardo Graciani and Bruno Juliá-Díaz from Universitat de Barcelona for offering us the opportunity to attend the Brainstorming.

References

1. M. Colomer, A. Margalida, and M. J. Pérez-Jiménez. Population Dynamics P System (PDP) Models: A Standardized Protocol for Describing and Applying Novel Bio-Inspired Computing Tools. *PLOS ONE*, 8(4):1–13, 2013.
2. K. S. Krane. *Introductory nuclear physics.* John Wiley and Sons Inc., New York, NY, 1987.

3. Live Chart of Nuclides: nuclear structure and decay data. https://www-nds.iaea.org/relnsd/vcharthtml/VChartHTML.html.
4. National Nuclear Data Center: Chart of Nuclides. http://www.nndc.bnl.gov/chart/.
5. N. Radionuclide Half-Life Measurements. http://www.nist.gov/pml/data/halflife-html.cfm.