CLASS: Core Library for Advanced Scenario Simulations

Baptiste Mouginot, Nicolas Thiolliere
Subatech Laboratory, EMN, Université de Nantes, CNRS, France

Abstract

The nuclear reactor simulation community has to perform complex electronuclear scenario simulations. To avoid constraints coming from the existing powerful scenario software such as COSI, VISION or FAMILY, the open source Core Library for Advanced Scenario Simulation (CLASS) has been developed.

The main asset of CLASS is its ability to include any type of reactor, whether the system is innovative or standard. A reactor is fully described by its evolution database which should contain a set of different validated fuel compositions in order to simulate transitional scenarios. CLASS aims to be a useful tool to study scenarios involving Generation-IV reactors as well as innovative fuel cycles, like the thorium cycle.

In addition to all standard key objects required by an electronuclear scenario simulation (the isotopic vector, the reactor, the fuel storage and the fabrication units), CLASS also integrates two new specific modules: fresh fuel evolution and recycled fuel fabrication. The first module, dealing with fresh fuel evolution, is implemented in CLASS by solving Bateman equations built from a database induced cross-sections. The second module, which incorporates the fabrication of recycled fuel to CLASS, can be defined by user priorities and/or algorithms. By default, it uses a "linear Pu equivalent"method [1], which allows predicting, from the isotopic composition, the maximum burn-up accessible for a set type of fuel.

This paper presents the basis of the CLASS scenario, the fuel method applied to a MOX fuel and an evolution module benchmark based on the French electronuclear fleet from 1977 to 2012. Results of the CLASS calculation were compared with the inventory made and published by the ANDRA organisation in 2012 [2].

Introduction

The nuclear reactor simulation community has to perform complex electronuclear scenario simulations. To avoid constraints coming from the existing powerful scenario software such as COSI, VISION or FAMILY, the open source Core Library for Advance Scenario Simulation (CLASS) has been developed. The main asset of CLASS is its ability to include any type of reactor, whether the system is innovative or standard.

CLASS principle

As all electronuclear scenario software, CLASS includes the entire set of key elements to describe an electronuclear fleet, from the reactor to the storage of used fuel, including the fuel reprocessing facility.

The main element of the CLASS library is the reactor. It is described by its power (in W), its mass (in tonne), its fuel type (UOX/MOX/…) and the final burn-up (in GWj/tonne)

of the used fuel. The cycle time of the fuel is calculated from the couple power/BU_{MAX} through the well-known relation Equation (1):

$$BU_{MAX} = \frac{P * CycleTime}{HeavyMetalMass} \quad (1)$$

It is also possible to set another couple of the triplet (BU/Power/CycleTime), depending on the user choices.

At the end of the irradiation, the spent fuel is sent to a cooling facility, from where it will be sent to the storage, at the end of a cooling time (five years by default). In storage, all output fuels from any reactor are stored individually. During all the cooling and storage time, the decay of all radioisotopes present in the fuel is taken into account. In addition to all the key parameters, a database is included to describe the evolution of the fuel during the irradiation.

There are currently two separate types of reactors. The most basic one corresponds to a reactor always filled with the same fresh fuel. This reactor type is used, for instance, to manage PWR reactor filled with uranium oxide (UOX) fuel.

The complex part comes from the other type of reactors, which reprocess fuels. Here, the composition of the used fuel in stock does not depend only on the final burn-up in the reactor, but also on the time spent in storage due to the decay of all radioactive isotopes. Thus, the composition of each reactor refill changes at each cycle. In order to simulate the evolution of the reprocessed fuel, CLASS needs two additional modules: the first one builds the reprocessed fuel with the correct properties in terms of reactivity (k_{eff} evolution) and mass, and the other one describes the irradiated fuel evolution.

At the end of the cycle, the used fuel is sent to the cooling facility. At the end of the cooling (usually five years) the fuel is either sent to a intermediate storage facility (to await reprocessing), or to final storage. During all this time (cooling, storage, waste), radioactive nucleus decays are taken into account.

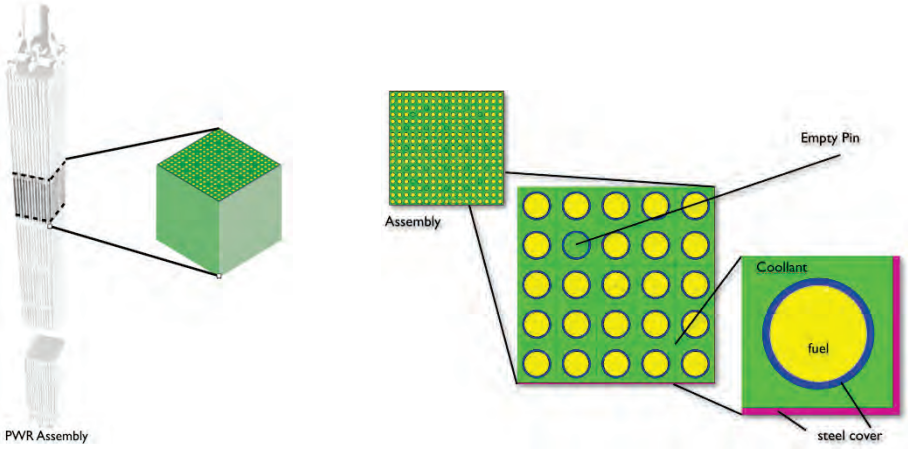
Database, fuel reprocessing and evolution

The following part describes the procedure used to perform all the simulations with the MURE software, and the processing scheme of the two modules mentioned previously: the computation of the reprocessed fuel and the determination of the evolution of the build fuel during irradiation.

Fresh fuel simulation

To compute the new fuel evolution database, the MURE software developed at CNRS has been used. This software allows the evolution from MCNP calculation, which determines at each time step (set by users) the system static parameters (mean cross-section, flux, k_{eff}). For the present example, to avoid full core geometry complexity, cubic assembly simulation was performed (see Figure 1) assuming that the fuel evolution in a cubic assembly with total reflecting surfaces is close to the core evolution.

Figure 1. Schematic representation of the simulated "cubic" assembly



In the present case, a constant power of 2.8 GW_{th} was used for the entire reactor. The simulation was set to perform a new MCNP calculation every quarter of a year. The UOX fuel standard burn-up in France ranges from 33 to 46 GWd/tonnes depending on the fuel campaign, so the irradiation time was set to 5 years, which corresponds to a burn-up of about 75 GWd/tonnes to reach the correct burn-up.

For each simulated composition, a CLASS database (called EvolutiveProduct) was generated, containing all the useful physical parameters (as the effective neutron multiplication factor, k_{eff}, the flux φ, nuclei quantities and fission, capture and (n, 2n) mean cross-sections). Figure 2 shows these parameters stored as an ASCII matrix.

Figure 2 presents the format of the CLASS EvolutiveProduct. The database can be generated by only formatting it in the right ASCII format.

Figure 2. Representation CLASS evolutive product in matrix form

$$\begin{pmatrix} & t_1 & \cdots & t_j & \cdots & t_n \\ k_{eff} & k_{eff,1} & \cdots & k_{eff,j} & \cdots & k_{eff,n} \\ \phi & \phi 1 & \cdots & \phi j & \cdots & \phi n \\ \mathrm{zai}_1 & Q_{1,1} & \cdots & Q_{1,j} & \cdots & Q_{1,n} \\ \cdots & \cdots & & \cdots & & \cdots \\ \mathrm{zai}_i & Q_{i,1} & \cdots & Q_{i,j} & \cdots & Q_{i,n} \\ \cdots & \cdots & & \cdots & & \cdots \\ \mathrm{zai}_m & Q_{m,1} & \cdots & Q_{m,j} & \cdots & Q_{m,n} \\ \sigma_{Fission,\mathrm{zai}_1} & \sigma_{Fission,1,1} & \cdots & \sigma_{Fission,1,j} & \cdots & \sigma_{Fission,1,n} \\ \cdots & \cdots & & \cdots & & \cdots \\ \sigma_{Fission,\mathrm{zai}_m} & \sigma_{Fission,m,1} & \cdots & \sigma_{Fission,m,j} & \cdots & \sigma_{Fission,m,n} \\ \sigma_{Capture,\mathrm{zai}_1} & \sigma_{Capture,1,1} & \cdots & \sigma_{Capture,1,j} & \cdots & \sigma_{Capture,1,n} \\ \cdots & \cdots & & \cdots & & \cdots \\ \sigma_{Capture,\mathrm{zai}_m} & \sigma_{Capture,m,1} & \cdots & \sigma_{Capture,m,j} & \cdots & \sigma_{Capture,m,n} \\ \sigma_{(n,2n),\mathrm{zai}_1} & \sigma_{(n,2n),1,1} & \cdots & \sigma_{(n,2n),1,j} & \cdots & \sigma_{(n,2n),1,n} \\ \cdots & \cdots & & \cdots & & \cdots \\ \sigma_{(n,2n),\mathrm{zai}_m} & \sigma_{(n,2n),m,1} & \cdots & \sigma_{(n,2n),m,j} & \cdots & \sigma_{(n,2n),m,n} \end{pmatrix}$$

Recycling fuel process

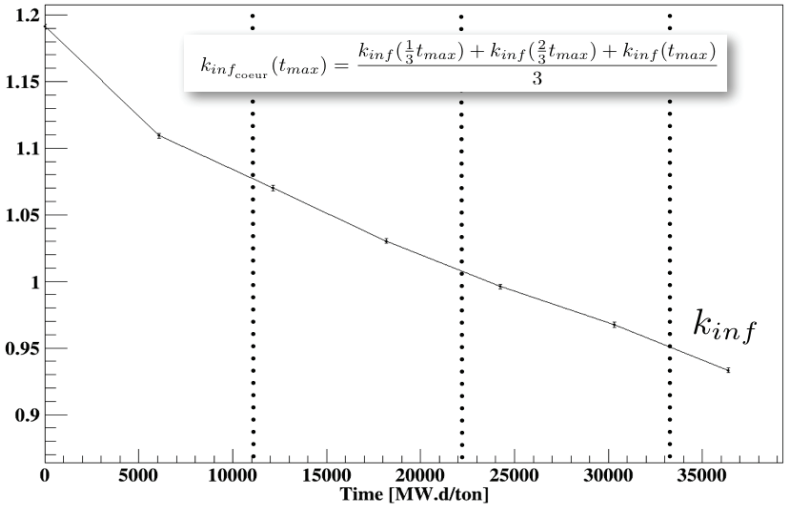
Both modules dedicated to recycled fuel reprocessing rely on a large number of realistic fuel composition evolutions. In order to create them, a random composition of the fuel is first generated within a realistic range (see Table 1) and then its evolution is simulated inside the cubic assembly described previously. The remaining composition is filled with depleted uranium at 3% of ^{235}U.

Table 1. Range of the random composition generation

Isotopes	% Maximum of the final composition	% Minimum of the final composition
^{238}Pu	0.03	0.09
^{239}Pu	1.7	4.8
^{240}Pu	0.65	2.00
^{241}Pu	0.00	0.72
^{242}Pu	0.15	0.45
Depleted U	Remaining	

For each generated fuel, the corresponding maximal burn-up has also been determined. The main limitation for a radioactive fuel is its reactivity, expressed by the effective neutron multiplication factor k_{eff}. In the following work, the maximal burn-up is defined where the mean value (for one-fourth core reloads) of the infinite neutron multiplication factor is equal to 1.06, as illustrated in Figure 3.

Figure 3. Evolution of the k_{inf} as a function of the burn-up and determination of the maximal burn-up



Each different random composition evolution and associated maximal burn-up comprised a database, all databases are stored in the data bank.

Recycled fuel fabrication

To build the proper reprocess fuel, it is necessary to determine the maximal burn-up that each composition can reach. For a set range of burn-ups, the α parameter is adjusted with the "linear Pu equivalent" method [2], which leads to the formula Equation (2):

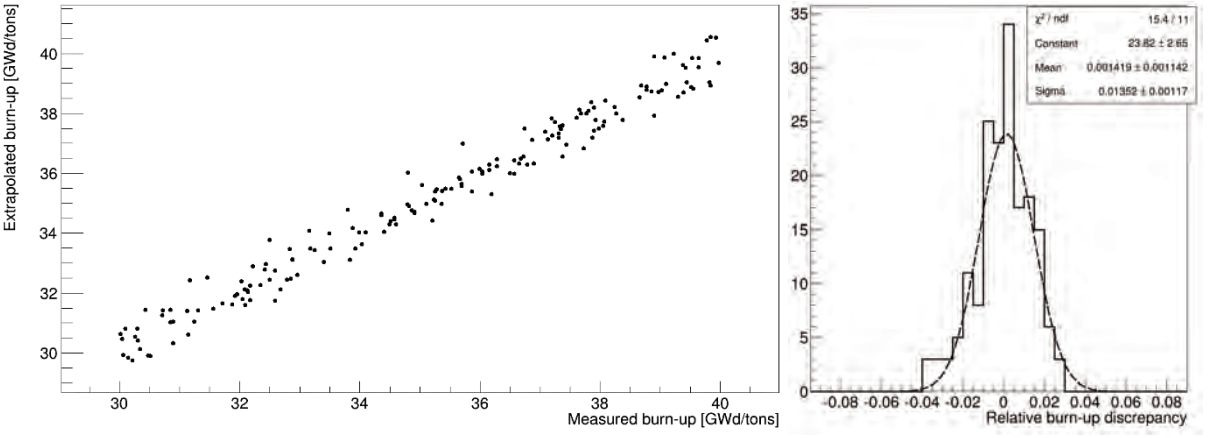
$$BU_{\max} = \sum_i \alpha_i n_i + \alpha_0 \qquad (2)$$

where n_i represents the isotopic fraction of the isotope i before irradiation.

In Figure 4, the formula dispersion can be observed; 1% error is obtained on the estimated burn-up ranging between 30 and 40 GWd/tonnes. The accuracy of the result

could be improved by reducing the considered burn-up range, but it would be computationally intensive.

Figure 4. Dispersion on the burn-up estimation, on the left extrapolated burn-up versus measured one in [GWd/tonnes] and on the right, the relative discrepancy between them



Once the maximal burn-up is estimated for any reprocessed fuel composition, a reprocessed fuel can be built for a reactor set by Equation (3):

$$\begin{cases} BU_{\max} = \sum_i \alpha_i n_i + \alpha_0 \\ M_{reactor} = M_{Pu} + M_{depletedU} \end{cases} \quad (3)$$

In the previous equation, $M_{reactor}$ represents the mass in the reactor using the built fuel, M_{Pu} the mass of plutonium extracted from the recycled stock (composed of used UOX fuel) and $M_{depletedU}$ the mass of depleted uranium needed to build the proper fuel in terms of the maximal burn-up and total mass, and BU_{max} the maximal burn-up reached by the fuel inside the reactor set. From Equation (3), the fraction of the stock required and the amount of depleted uranium needed to build a reprocessed fuel can be determined. If more than the tested stock is required to build the fuel (fraction>1), all the plutonium isotopes of this stock are used and the remaining portion is taken in the next stock.

This way, this built fuel is able to reach the maximal burn-up of the specific reactor that should be filled. The next step is to determine how the composition of this fuel could evolve inside the reactor through irradiation.

Recycled fuel evolution

In order to determine the evolution of the reprocessed fuel composition, the data bank of the fuel evolution (described previously), from which the closest fuel composition is extracted, was sed. Thus, a reference database was defined for the specific reprocessed fuel composition.

Using the cross-sections, σ_i, and flux, φ_i, given by the reference database, and the well-known Bateman equation [Equation (4)], the evolution of the built fuel is calculated after renormalisation of the flux. At each time step, the cross-sections and the flux are updated from the reference database (and the flux renormalised), until the end of the evolution. This module performs the evolution only for the main actinide isotopes.

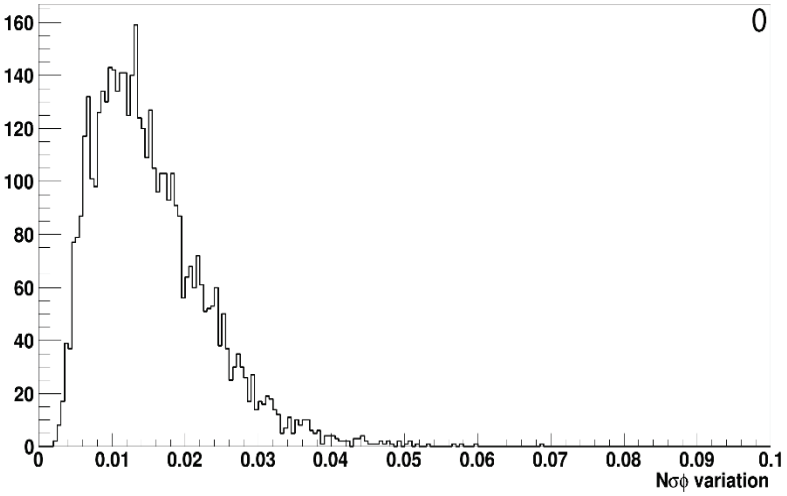
$$\frac{dN}{dt} = -\lambda_i N_i - \sigma_i \varphi N_i + \sum_{j \neq i} \lambda_{j \to i} N_j + \sum_{j \neq i} \sigma_{j \to i} \varphi N_i, \forall i \quad (4)$$

An on-going study allows estimating the error associated with this method. This study consists of reconstructing the evolution of each random composition fuel using Equation 4 from all the other databases in the data bank. So far only one parameter has been studied; the relative differences in reaction rate, "nσφ variation", Equation (5):

$$(n\sigma\varphi \text{ variation}) = \frac{\sum_i \left| (n_i\sigma_i\varphi)_{estimated} - (n_i\sigma_i\varphi)_{simulated} \right|}{\sum_i (n_i\sigma_i\varphi)_{simulated}} \quad (5),$$

where the estimated parameter corresponds to the calculation through previously described method, the simulated parameter corresponds to the simulation, n is the quantity of isotope i, σ the cross-section for a set reaction, and φ the renormalised flux (the sum is made on all isotopes present in the fuel and on the fission, capture and (n, 2n) reactions).

Figure 5. Dispersion of the relative reaction rate differences at first time step



In Figure 4, the dispersion of the "nσφ variation" is about 1% (in Figure 4, each point of the distribution corresponds to the relative difference for one randomly generated fuel), which seems reasonable. As shown in Figures 5-6, this dispersion exponentially increases with time and decreases with the number of random fuel composition initially generated, following a well-known tendency in $\frac{1}{\sqrt{n}}$.

Figure 6. Evolution of the mean of relative reaction rate dispersion as a function of the time step, each curve represents the number of random fuel composition simulated

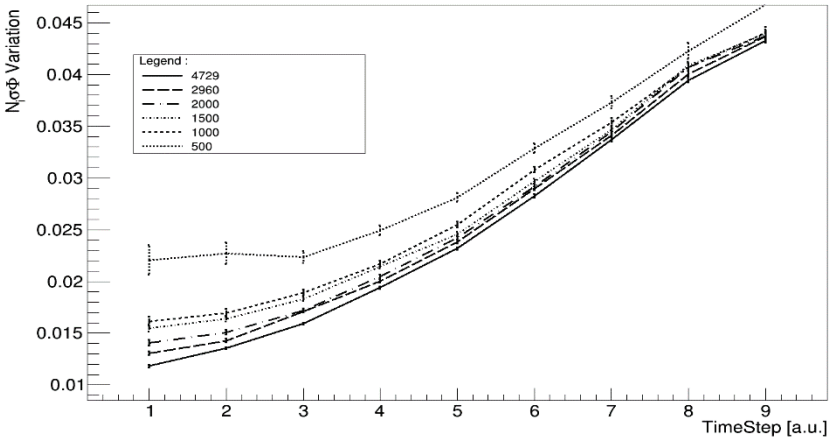
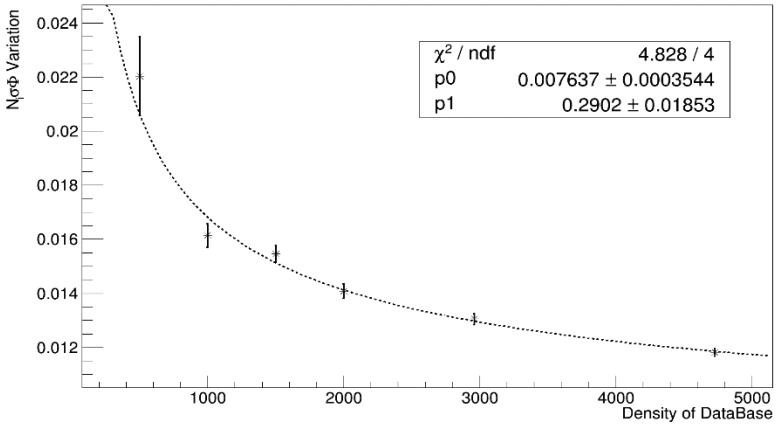


Figure 7. Evolution of the mean of relative reaction rate dispersion as a function of the number of random fuel composition simulated



On-going studies are being performed in order to improve this result. There are two main ideas: one is to improve the flux management in the evolution module and the other is to test different algorithms to determine the distance between two isotopic vectors. Flux management is necessary in order to conserve a constant power. Indeed, it would be better to renormalise the flux many times between two time steps. Finally, the definition of distance between two isotopic vectors is a crucial aspect, because the evolution method is very sensitive to it. To define this distance, different algorithms will be examined.

CLASS application: French electronuclear fleet 1978-2011

In the present study, the CLASS software has been used to reproduce the evolution of the French electronuclear fleet from 1978 until 2011. The Pressurised Water Reactors (PWR) installed after 1978 are considered here, but the Uranium Naturel Graphite Gas (UNGG) reactors used before are not. Only the stocks produced by the French reactors are taken into account.

The French fleet

Six different types of PWR constitute the French electronuclear fleet: the CP0, CP1 and CP2, the P4 and P'4, and the N4. In order to simplify the description of the fleet and because their characteristics are close, we will consider that the CP1 and CP2 are identical, as are the P4 and P'4, and are, respectively, called CPY and P4. Table 2 shows the characteristics in terms of enrichment, burn-up and loading plan for all the considered campaigns.

In order to simplify the realisation of the scenario, the different loading plans have not been taken into account. We are assuming that all the fuels inside the reactor are removed at the end of the cycle, and that the reactor is completely refilled at the beginning of the new cycle. Consequently, the cycle corresponding to a loading plan by a third and by a quarter, respectively, lasts 3 and 4 years. This can fluctuate a little, depending on the reactor and the campaign.

Table 2. Characteristics of the different fuel campaigns

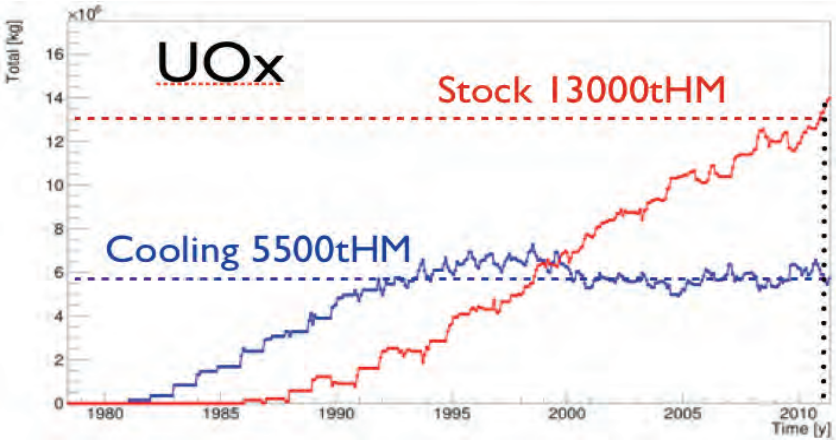
Campaign	^{235}U Enrichment	Pu quantity	Burn-up max	Cycle time (day equiv. full Px)	Loading plan
Standard 1450	3.4 %	None	43	258	1/4
Standard 1300	3.1 %	None	33	300	1/3
Standard 900	3.25 %	None	33	298	1/3
Garance	3.7 %	None	42	275	1/4
Cyclades	4.2 %	None	42	355	1/3
Gemmes	4.0 %	None	43	395	1/3
MOX 16%	Equiv. 3.7 %	7%	35	280	1/3
MOW "low"	Equiv. 3.25 %	5.3%	36	290	1/3

The used fuel stock

Throughout the process from reactor to storage, the CLASS software allows us to follow the composition of each fuel in each facility of the fleet. The aim of this work is to present the total inventory in stock and in cooling for UOX and MOX. This information is also provided by the ANDRA organism [3].

For UOX used fuels, the ANDRA reported 12 006 tonnes of heavy metal in stock, including cooling, versus 18 500 tonnes of heavy metal predicted by CLASS (see Figure 8). The large difference is easily explained by the presence of 56 tonnes of plutonium already separated from used UOX fuel, which corresponds to about 5600 tonnes of heavy metal before separation. However, CLASS still overestimates the amount of UOX fuel in the fleet by about 1000 tonnes. This could be explained by the large tendency of the PWR operator to always use the fuel at maximum burn-up, adding about 20 days' equivalent full power at the normal irradiation of the fuel.

Figure 8. Evolution of the total amount of UOX used fuel in stock and cooling, in tonnes of heavy metal



For the MOX fuel, the predictions are in good agreement: 1250 tonnes is predicted by CLASS versus 1287 tonnes for ANDRA. Overall, CLASS is performing well in estimating the total amount of fuel in the fleet, these results are the first step towards the validation of the CLASS software. A detailed benchmark will be performed in the coming years, including a comparison with the COSI software developed by CEA. In this comparison, the isotopic evolution of the different fuels will be analysed.

Conclusion

In the present work, the current status of the CLASS software is detailed and illustrated with a simple example. The main working scheme is explained through the example of the French fleet. The basic capacities of the CLASS software are verified, nevertheless, a detailed analysis and a precise benchmark are still required.

CLASS is a young software (about 1 year's development), but it has got off to a promising start. Many further developments and analyses are already in progress, and other will come in order to make CLASS, some precise software for electronuclear scenario simulation. During the upcoming year, precise comparisons with different items of scenario software such as COSI will be performed on transitional scenarios. This benchmark will allow validating the CLASS software.

Future improvements will also concern the development of a standard method for data bank generation. Priority will be given to the generation of a sodium fast reactor MOX and Multi MOX data bank and one for ADS in order to consider the management of minor actinides and eventually plutonium in "end game" scenarios.

As mentioned previously, on-going analyses will also study different aspects of the definition of distances between two isotopic vectors and their influence on the evolution module. The evolution modules will also be extended to the entire nuclear chart.

Acknowledgements

The authors would like to thank the young collaboration team involved in CLASS, for their enthusiasm, discussions and ideas present and future, which have been and will be the fuel of CLASS's development.

References

[1] Meplan, O. et al. (2009), "MCNP Utility for Reactor Evolution: Couples Monte Carlo transport with fuel burnup calculations", *MURE, Computer Program Services of OECD/NEA*.

[2] Kepisty, G., David, S. (1970), "Performance of Pu/Th fuels in PWR, analysis of the fissile material effiency for future reactors", *Internship report, Institut de Physique Nucléaire d'Orsay* – CNRS.

[3] Inventaire national 2012, Rapport de synthèse, ANDRA.