

# 2016 RPA Narrative RPA-16-10565

## Error Propagation for Fuel Cycle Calculation

February 10, 2016

**Technical Workscope Identifier:** FC-5.1b

**Principal Investigator:** Paul P.H. Wilson, Professor, University of Wisconsin-Madison

**Co-PI:** Dr. Baptiste Mouginot, University of Wisconsin-Madison

**Time Frame:** 3 years

**Estimated Cost:** \$800,000

## 1 Proposed Scope/Context Description

The fuel cycle simulation tool can have a large scope of application, from the study of the behavior of some type of fuel or reactor inside an existing nuclear fleet to the prospective analysis of a complete nuclear transition. Beside the study of fleet evolution it can also be used to assess the possibility of material hijacking in the context of non-proliferation study or policy...

Each application field of the nuclear fuel cycle calculation requires a specific level of confidence, which are up to now very un-precisely assessed, if assessed. There is a real need of validation the those kind of calculation, which can hardly be reached. Indeed, the only existing way to validate any fuel cycle calculation/tool, is the benchmark with other similar tools, or other existing data... When the first is generally conclude with a list of why the different softwares end up with different results (without concluding on the precision of any), the second allow only the validation on existing concept and have no impact on calculation implying the use of new concept. The aim of this project is to add error propagation capability to the CYCLUS fuel cycle simulator [1]. By their usage for predicting the evolution of a large industrial enterprise in an uncertain future, nuclear fuel cycle simulations are generally based on approximations and uncertain input data. Since validation is largely considered to be impractical, such simulations are seen as indications of future behavior rather than predictions of that behavior.

Nevertheless, it would be valuable to be able to place some confidence bounds on those indications, both to assess the robustness of conclusions that derive from those indications and

to provide information about the sensitivity of those conclusions to the uncertain data and algorithms.

Having a broad distribution for each metric calculated in a fuel cycle simulation instead of unique values will allow a better comparison between different fuel cycle scenarios.

Moreover for some critical analysis such as retrospective non-proliferation analysis, it could be extremely valuable to add some degree of confidence on the simulation performed. This could allow at least to confirm or invalidate the possibility to use those calculation tools for such purpose.

This project will extend the Cyclus concept of resources to include error information and then develop a number of archetypes that can perform operations to propagate that error in a rigorous fashion. Ultimate calculation of fuel cycle performance metrics will also need to be updated in order to represent final results as distributions rather than single values.

## 2 Logical Path to Work Accomplishment

The goal of this project is to add optional extensions to Cyclus that will allow an assessment of the error as it propagates through a fuel cycle.

### 2.1 Add uncertainty to ressources

### 2.2 Archetypes uncertainty management

In a fuel cycle calculation, there are three different uncertainty/error sources :

- the material uncertainty,  $\delta\vec{N}_{in}$ , which is the input and the output of each archetypes,
- the "tolerance",  $\tau_i$ , on the input parameter, which corresponds the possible variation range of any physics parameter,  $i$ ,
- the error,  $\epsilon_{mod}$ , introduce by the archetypes modeling/operation on the material.

The uncertainty on the output material is a function of all those uncertainties :

$$\delta\vec{N}_{out} = \mathcal{F}\{ \delta\vec{N}_{in} , \tau_0 , \dots , \tau_n , \epsilon_{mod} \} \quad (1)$$

The work we are proposing to do on this project, is to allows all archetypes of CYCLUS, to combine all those sources and computes the resulting uncertainties on the output materials. In the following parts, one will tried to detail the work which need to be done on the different archetypes, we are proposing to update.

#### 2.2.1 Enrichment facility

The enrichment facility is probably the easiest facility to model, since the only error come from the tails and feed stream enrichments, the enrichment process could be linearly models. This can be easily analytically combine and propagate.

*Is it true with any enrichment technic ?  
do we want the exact error propagation formula ?*

### 2.2.2 Separation

As the enrichment process, the separation should be very easy to model including uncertainty. Indeed, the only parameter which can introduce extra variance is the separation efficiency,  $\tau_{eff}$ .

*do we want the exact error propagation formula ?*

### 2.2.3 Reactor

As all archetypes the reactor, are subject to physical parameter fluctuation. The different parameter one should consider are the discharge burnup, the effective power/capacity factor... Variation consideration on those parameters will affect the discharge time, which should be very difficult to include in the uncertainty calculation. Nevertheless, a brute force study (also sometimes called "Total Monte Carlo method") could allow to determine their impact, running many times the same simulation choosing randomly the parameter value at each cycle of the reactor, the distribution of the results will provide a precise measurement of the sensitivity.

We are envisaging two kinds of the reactor, which will be both capable to handle uncertainty. The first one will be built as an upgrade of the existing CYCAMORE reactor, making it error aware. The CYCAMORE reactor is a recipe based reactor, in addition of all the classical reactor parameters (batch number, cycle length, power, capacity factor,...) the user provides the input and output fuel recipes. The only requirement is to force the user to provide the output recipe with the corresponding sensitivity to input uncertainty. From this sensitivity, one will be able to compute directly the output uncertainty on the output composition from the input one.

The second version of reactor will be able to calculate the evolution of the fuel provided by the fuel fabrication (see §2.2.4). To do so, we propose investigating two ways, both using pre-trained models, allowing the prediction of key physics parameters needed to compute the evolution of a fuel during the irradiation. It has been proven that from pre-trained neural network models, one can predict the evolution of the one group cross section during the irradiation of the fuel from its initial isotopic composition, and is working for a various range of reactor, from LWR to SFR [?, ?].

The first application using the neural network predictive models, is to train a model to directly predict the composition evolution as the function of the burnup. This application might not work, since the usage of neural network has been proven to predict one group macroscopic cross section.

If the neural network model fails to directly predict precisely the isotopic composition evolution, one has to consider the second option, which implies to predict the 1 group cross

section, then integrate the Bateman equation.

The main reason leading the try of direct prediction of the isotopic evolution, is the error/uncertainty propagation. As express previously :

$$\delta \vec{N}_{out} = \mathcal{F}\{ \delta \vec{N}_{in} , \tau_0 , \dots , \tau_n , \epsilon_{mod} \} \quad (2)$$

The determination and the propagation of all uncertainty source is, for this kind of reactor, a complicated matter. The error due to the computation of the depletion calculation are wild:

- the error of the neural network predictor,  $\epsilon_i^{nn}$ , with  $i \in [0..N]$ ,  $N$  the number of predicted parameter,
- the convolution of the uncertainty on the material composition with the neural network prediction.
- the calculation error on the data sets used to train the neural network,  $\epsilon_{T.D.}$ .

And then can be expressed as :

$$\epsilon_{mod} = \mathcal{G}\{ \delta \vec{N}_{in} , \epsilon_0^{nn} , \dots , \epsilon_n^{nn} , \epsilon_{T.D.} \} \quad (3)$$

Because, the predictive model are trained on sample populated using few thousand of depletion calculation, which are subject to computation error,  $\epsilon_{T.D.}$ . On one side, a depletion calculation take as input, the nuclear data. Those nuclear data are interpolated/extrapolated from many different experimental measurement using many different models. Therefore the nuclear data contain uncertainty... Those uncertainty are extremely difficult to propagate properly through a full depletion calculation because of the coupling between neutron transport and depletion calculation: the composition of the fuel impact the shape of the neutron spectrum, which impact the reaction rate on the nuclei... On the other side, the depletion calculation require different approximation to be completed. There is nearly impossible using Monte Carlo technique on a PWR full core calculation due to source convergence issue. It is also extremely complicate to follow precisely the different reactor parameter, such as boron concentration, rod control management, charge factor evolution, neutron leakage... The study and the propagation of the modeling uncertainty, such as the modeling simplification and the nuclear data uncertainty is way beyond the scope of this project...

This require a full dedicated research project (and probably more). Therefore, those error,  $\epsilon_{T.D.}$ , will not be considered on the first version of this work. This might need to be reconsidered when the depletion calculation error propagation capacity will have done important progress.

The direct error induced by the use of predictive model,  $\epsilon_i^{nn}$  on each parameter,  $i$ , need to be assessed. This could be performed with a mapping the error on the isotopic space

populated with the training sample. This will allow to determine the error of the model on each point on the isotopic space populated. We might use then a other neural network, or other interpolation method to predict the error of the prediction as the function of the isotopic composition.

Once we have a working predictive model and a map of the error on the prediction, one need to build the covariance matrix which will allow to convolute the uncertainty on the input material,  $\delta N_{in}$ , with the prediction of the model.

If the direct prediction of the composition is not precise enough, one have to use solve Bateman equation using predicted one group cross section and then compute the error coming from a numerical resolution of the Bateman equation and propagate the error of the one group cross section. The Bateman equation resolution will be a step by step process. After having discretize the irradiation time (or burnup), one will use the model to predict the cross section at each time step (closer the time step are preciser the calculation will be) and then solve numerically the Bateman equation step after step, ending with the final isotopic composition of the fuel. Because the predictive model will be able to predict the one cross section as the function of time (or burnup), one should be able to propagate those error using sensitivity analysis. The time discretization as well as the other approximation require to solve the Bateman equation will also introduce computation error that need to be determined and added to the final uncertainty. This could be made with the comparison of depletion calculation made with those extra approximation with the reference one performed using the same modeling approximation as the training depletion all along the isotopic space.

#### 2.2.4 Fabrication

The aim of the fuel fabrication is to mix different incoming material streams in order to build a fuel which validate the neutronics/physics requirement of the reactor. Depending of the reactor, the criterium could be various. We are considering in the first time including fabrication model for MOX fuel only, in LWR and SFR, used as burner and breeder for SFR. One will use algorithm building fuel allowing to build fuel reaching the targeted burnup according to either criticality criterium either conversion ratio criterium. Those algorithm will relies on the capabilities of some predictive model to predict the maximal achievable burnup depending on the the criticality or conversion ratio evolution according to burnup. The predictive model as the reactor model, will be based on the use of neural network formerly trained on the same set of training depletion calculation used for the reactor models. The capability of the neural network have been proven to predict the evolution of the criticality in PWR reactor using MOX fuel [?, ?], as well as the the initial criticality of MOX fuel in SFR [?] and should be easily extended to conversion ratio evolution.

The error/uncertainty propagation for fuel fabrication should be pretty similar than for reactor. Indeed the uncertainty can be expressed as :

$$\delta \vec{N}_{out} = \mathcal{H} \{ \delta \vec{N}_{in} , \epsilon_{mod} \}. \quad (4)$$

In this case there is no tolerance, since the parameter are goal to achieve, not physical characteristics. As well as previously the error of the model can be expressed as :

$$\epsilon_{mod} = \mathcal{K}\{ \delta \vec{N}_{in} , \epsilon_0^{nn} , \dots , \epsilon_n^{nn} , \epsilon_{T.D.} \}, \quad (5)$$

where  $\epsilon_i^{nn}$  represent the error on the prediction of the parameter  $i$  by the neural network predictive model and  $\epsilon_{T.D.}$  the error due to the error on the depletion calculation composing the training set, which will not be considered since we don't have any way to correctly estimate it.

As for the reactor model, the  $\epsilon_i^{nn}$  component of the error can be determined through a mapping of the error along the isotopic space, and the impact of the material input uncertainty needed to be assessed by the calculation of the covariance matrix which need to be build.

## 2.3 Problems/Applications/Validations

During the realization of this work we would like to validate each step of the uncertainty propagation process. First, with a mid/early-term validation, after the enrichment facilities, the separation and the recipes reactor will be implemented, it will be possible to confirm the uncertainty is correctly propagated, using a brute force validation (with the "Total Monte Carlo" method). This validation will be continued with new archetypes will be implemented in CYCLUS...

An other goal we want to achieve, is a sensitivity analysis, allowing to determine the impact of the different uncertainty/error/tolerance on the final calculation uncertainty. This should allow to define precisely where the future effort should be focus to reduce those uncertainty sources accordingly to the object of the calculation and also to validate (or un-validate) the use of fuel cycle simulation tool for some specialized study (such as non-proliferation study...) With all the necessary components in place, a series of demonstration simulations will be conducted using fuel cycles of increasing complexity: once through, MOX LWR recycle, and fast reactor recycle. These scenarios will be constructed to highlight the role of error and uncertainty and identify metrics in which the presence uncertainty may impact fuel cycle analysis conclusions.

### 3 Milestone Task Listing

#### **TASK 1:** CYCLUS update for uncertainty awareness

- subtask 1: update material to uncertainty,
- subtask 2: validate the backward compatibility,
- subtask 3: Add a default uncertainty behavior when using both uncertainty aware archetypes and standard one in the same time ?

#### **TASK 2:** Modeling development

- subtask 1: isotopic space definition, training sample realization
- subtask 2: reactor models development: parameter prediction, error analysis
- subtask 3: fuel fabrication model development: parameter prediction, error analysis

#### **TASK 3:** Updating the CYCLUS Archetypes to uncertainty management

- subtask 1: enrichment facility,
- subtask 2: separation facility,
- subtask 3: recipe reactor,
- subtask 4: reactor archetypes, depletion calculation/prediction, uncertainty propagation,
- subtask 5: fuel fab archetypes, mixing calculation/prediction, uncertainty propagation.

#### **TASK 4:** Validation & application

- subtask 1: validation of the overall process with simple calculation : enrichment + separation + recipe reactor
- subtask 2: validation of modeling capabilities (Fab + reactor)
- subtask 3: exemple calculation: PWR, transition from PWR to FBR
- subtask 4: full sensitivity analysis
- subtask 5: time dependent parameters sensitivity analysis (discharge burnup, capacity factor...)
- subtask X: comparison with other physic modeling capabilities such as Bright-Lite ?





### 3.1 Brief Project Description

Different steps have to be followed to accomplish it. The first step is to extend the standard resources to contain their own error information and develop methods to propagate the errors as agents operate on those resources. With such an infrastructure in place, specific archetypes will be developed to perform operations that depend on this error information, with a focus on recycled fuel fabrication, advanced fast reactors that consume recycled fuel, and separations facilities. When used as part of an actinide recycle scheme, these facilities will trade in approximations to ideal material compositions and hence introduce errors. Analysis tools that can express the outcomes of this error propagation on fuel cycle performance metrics will also be developed. Demonstration of error capability will begin with once through fuel cycles, followed by a simple MOX-recycle system, and culminating with multi-recycle fast reactor systems.

#### 3.1.1 Update CYCLUS

Expressing uncertainty in resources will require a fundamental change to the Cyclus kernel to extend the resource objects to include support for their own error estimates. Prior examples of this type of capability [e.g. 2] will serve as a guide for implementation in the Cyclus context. When complete, this capability should allow every resource to carry a measure of its uncertainty. In addition, all operations that are available to modify resources will be extended to also modify error estimates appropriately. This work will focus on analytical approaches to error propagation, that can be implemented in a way that reduces to the current behavior in the absence of uncertainty information. This will ensure backward compatibility with existing Cyclus archetypes that may not be aware of uncertainty propagation.

#### 3.1.2 Models development

The recipe-based fuel management of Cycamore is not well suited to perform precise error estimation and propagation: by definition, the output fuel composition is precisely defined by the input fuel composition. By contrast, when exploring recycling in nuclear fuel cycles, it is necessary to introduce archetypes that seek to achieve an integrated performance goal by blending streams of material. One such algorithm has been developed as part of the French nuclear fuel cycle tool, CLASS [3]. Using either a polynomial regression [4] or a neural network approach [5], two material streams are mixed to achieve a particular neutronics performance, e.g. burnup of PWR MOX fuel.

Both models we are proposing using, are based on the neural network prediction. Those neural networks are priorly trained using on a set of data composed by many different fuel isotopic compositions (between thousand and few thousands depletion calculations). Different examples have shown the great potential of neural networks predicting different neutron spectrum integrated parameters (macroscopic cross section, multiplication neutron factor...) for many types of reactor (PWR, Sodium cooled - FBR, lead cooled ADS) on many types of fuel, MOX, Pu-MA ADS fuel, UOX...

To elaborate such kind of model, one need to build the training sample used to feed the different models, used for fuel fabrication and macroscopic cross section prediction. This training sample are composed by few thousand different couple of input and output parameters. In our case, the input parameter are the initial fuel composition, and the output parameter are the direct (i.e. macroscopic cross section,  $k_{eff}$ ...) or indirect (  $< k_{eff} >$ , maximal burnup accessible...) results of depletion calculation. In any case, one need to use a depletion calculation tool, to populate the training sample. To generate our training sample data we are envisaging using the SERPENT tool as it seems to be the actual best depletion calculation tool available (in term of precision and speed).

*Moreover the quality of the the model will depend on the geometrical repartition and density of the data points in the training sample. It seems that using a latin hyper square repartition [] is the best to train a neural network model when a square repartition may probably netter for a polynomial models*

Since all the output parameters have been computed, one can train the model. Two different kind of model have to be generated:

- the fabrication model: allowing to mix  $N$  different material streams to build a fuel which will have the correct neutronic/physics properties,
- cross section model/depletion model: allowing to determine the isotopic evolution of the fuel during irradiation.

## Fuel fabrication Model

### Cross Section/Depletion Model

Cross section model are used to predict the one group cross section evolution as a function of the burnup (or the time) for all reaction on any nuclei that might appear in the fuel during the irradiation. Using them it is possible to solve the Bateman equation and then determine the fuel composition evolution as the function of the burnup (or the time).

Instead prediction each different cross sections, it might be able to predict directly, using properly trained neural network, the isotopic fuel composition.

Since both kind of model should be trained on complete depletion calculation, containing both one group cross section evolution and the isotopic composition evolution, it could be very interesting to try both, keeping only the most interesting solution.

## 3.2 Uncertainty

The following project aims to introduce uncertainty capability to material metrics, or material related metrics such as separation tails, burnup... into CYCLUS fuel cycle simulator. Moreover, it could be consider as a base work for the management and the propagation of any metrics in CYCLUS tool.

### 3.3 Estimation

#### 3.3.1 input parameter

The CYCLUS archetypes will be updated to allow them deal with the uncertainty on metrics related to the flow process itself (separation efficiency, enrichment tails, burnup achievement...). Those uncertainty should be fixed by the user to the appropriate values. Some default values could also be set...

Even if those uncertainty should have a moderate impact on the full fuel cycle calculation, it should be quite simple to implement it... and it could be used as a validation test on the overall uncertainty propagation in the different archetypes comparing the new build-in CYCLUS capacity and some brute force calculation...

Although, those uncertainty should have a limited impact on the fuel cycle calculation, it is still required to measure it. To do so, it will be interesting to perform a sensitivity analysis on those.

#### 3.3.2 Models

Those models, as explained previously, are trained on a large amount of pre-calculated depletion calculation. Some study have been done to determine the optimal density of a training sample to train the model [B. Leniau private com?]. This could be quickly completed in the context of this project, allowing to determine the precision of the model intrinsic precision as a function of the training sample density.

One also need to perform some bibliography to determine which neural network library will be the most suitable for the use in a fuel cycle simulation tool and/or if other statistical predictive method might be interesting as well.

Finally, one need to assess the precision of the models. This have two main different components: the one directly connected to the prediction capability of the used model, and the uncertainty of the the computed data used as training data for the models.

The computed data correspond to a sample of few thousand of different depletion calculation. Each of the depletion calculation have 3 uncertainty components, a statistical, a nuclear data and a modeling one.

In addition of the model uncertainty, one need to convolute, the isotopic uncertainty (on the isotopic fuel composition) with the model predictions. A solution could be to use a brutal method such as the Total Monte Carlo, realizing many different predictions using as input many different small variation of the isotopic composition, the dispersion on the prediction provides an correct estimation of the corresponding uncertainty. One will also investigate more elegant solution, using analytic uncertainty combination.

## 3.4 Propagation

The main difficulty in this project remain the propagation of the different uncertainty along the fuel cycle calculation.

### 3.4.1 input parameter

The new CYCLUS archetypes developed in a first times be capable to handle error on materials flow, allowing the propagation of the uncertainty on the input materials flow through all the facility process, delivering an material output flow(s) with the corresponding uncertainty.

### 3.4.2 Models

It should be possible use a simplified version the "Global Perturbation Method" apply on the depletion calculation developed by [Williams, Takeda, Salvatores], allowing the propagation of the different uncertainty across the integration of the Bateman equation.

One solution that could be investigate, is simply to use a TMC method [TMC] to evaluate the sensitivity to the cross section and initial inventory uncertainty to the final composition. Unfortunately those sensitivity should depends on the composition (for a PWR reactor, the macroscopic cross section accordingly to the shape of the neutron spectrum shape highly depending on the isotopic composition depending of the initial composition). In consequence, using TMC to determine the different sensitivity imply a very high preliminary CPU cost, but should improve the speed on the fuel cycle simulation versus a perturbative propagation of the different uncertainty.

## 4 Final Phase :

We imagine 3 step in the final phase. First we would like to step up a testing mechanism, based on simply brute force the uncertainty propagation in the fuel cycle calculation by using some kind of Total monte Carlo, allowing all parameters to fluctuate accordingly to their respective uncertainty on many integration of the same calculation. The result distribution can be read as a direct measure of the uncertainty on the final parameter and will be compared to the direct uncertainty propagation calculation as a validation of all the method previously presented. This method will (turning on the uncertainty on each parameter one by one) to validate the uncertainty propagation on each parameter...

One need nevertheless keep in mind that everything is never perfect, and one should always try to improve. To do so it is required to identify the most problematic issue and solve it. In this particular case, one need to identify what is the uncertainty gaol depending on the application of the fuel calculation, of course the precision goal is very different when one doing a prospective calculation versus a non-proliferation retrospective analysis... Knowing it, one very important of this project is it could allow through sensitivity analysis to identify the biggest uncertainty source, and help to focus to improve it.

That is why, in addition of the TMC test tool, I would like to develop a tool/method to systematically allows user to perform sensitivity analysis applied to their calculation allowing us to improve future models dedicated to some precise use of the CYCLUS tool.

Finally, we would like Third step : application to simple case : PWR, transition from PWR to FBR...