

## COCAGNE: EDF new neutronic core code for ANDROMÈDE calculation chain

Ansar Calloo, David Couyras, François Févotte, Matthieu Guillo\*,  
 Coline Brosselard, Bertrand Bouriquet, Alexandre Dubois, Enrico Girardi, Fabrice Hoareau, Marie Fliscounakis,  
 Hadrien Leroyer, Etienne Noblat, Yann Pora, Laurent Plagne, Angélique Ponçot, Nadine Schwartz

EDF R&D: 7 boulevard Gaspard Monge, 91120 Palaiseau, France

\*Corresponding author: matthieu.guillo@edf.fr

**Abstract** - EDF has been developing a new calculation chain, ANDROMÈDE, to replace the current one, CASSIOPEE. This work has been underway for more than 10 years now, and among the different components, there is the neutronic core code COCAGNE, which is the subject of this paper. COCAGNE has state-of-the-art flux solvers, an efficient microscopic (isotopic) depletion solver, an interface for great flexibility to carry out studies for R&D research as well as coupling with different physics through the SALOME platform. Although COCAGNE has yet to undergo extensive V&V before licensing, it has reached the required maturity and robustness for R&D studies.

### I. INTRODUCTION

EDF is developing a new core code named COCAGNE. This code is part of the new EDF calculation chain, ANDROMÈDE. COCAGNE is a major step forward for an industrial core code, since it has state-of-the-art flux solvers (SPn and Sn), an efficient microscopic depletion solver, and different levels of cross section homogenizations (including pin-homogenized or pin-by-pin data) and relies on a state-of-the-art lattice calculation scheme using the APOLLO2.8/REL2005/CEA2005v.4 [1, 2] package developed at CEA.

These new features, alongside with its new flexible architecture that employs heavy use of Object-Oriented paradigm and the python scripting language, make it perfectly suited for both industrial uses and advanced best-estimate studies.

### II. ARCHITECTURE AND INTERFACE OVERVIEW

#### 1. Core Code Architecture

As is the case with modern softwares, COCAGNE is not a stand-alone code, but rather a dynamic library that provides services, organized using a layer approach (Fig. 1)

The python interface is intended to be used by the physicist in charge of modeling the core and writing core procedures (critical boron search and so on). Important objects, such as solvers, exchange containers from the C++ layer which are available to the python interface using the “swig” toolkit. On the top of that, and in order to organize responsibilities, COCAGNE introduces the notion of *Core Modeling Services* (CMS). Each of these is in charge of one consistent aspect (data and processing) of the simulation of a nuclear core. For instance, *Neutronics* relates to flux solvers, *Sections* to cross sections production and so on. A specific CMS, *Etude*, has no other role than the knowledge of all others CMS, their responsibilities and how they relate to each other. *Etude* knows all CMS which in return all know *Etude* (Fig. 2). Following the *Mediator* design pattern in Object Oriented design, *Etude* ensures an organized communication between them.

As such, *Etude* also helps for automatic tasks: for instance when *ThermalHydraulic* has performed some calculations it

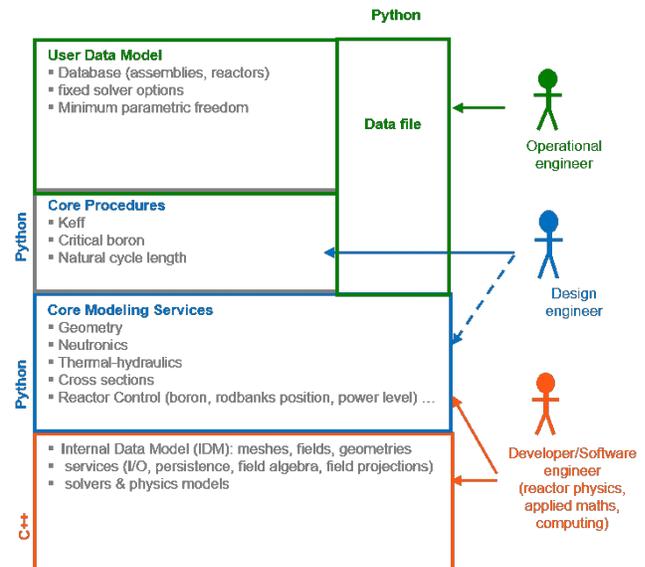


Fig. 1: Software layers and different types of users

sends a signal to *Etude* which in turn sends a signal to all CMS it knows that may be affected by this outcome, such as *Sections*. But *Etude* has no idea what *Sections* may do, it is not its responsibility. It may be noted that for specific use, we can turn off some automatic tasks.

This design avoids side effects encountered in many codes where many entities can manipulate the same data and no entity is responsible for anything.

Using CMS, writing Core Procedures is a high level process for which the engineer doesn't have to worry about details of data management. Rather he only has to focus on the physics involved and how data are exchanged among CMS during the simulation.

#### 2. Interface to EDF simulation platform: SALOME

COCAGNE is part of EDF numerical simulation environment, SALOME which provides common services with the following philosophy: “one physics, one code”. This means

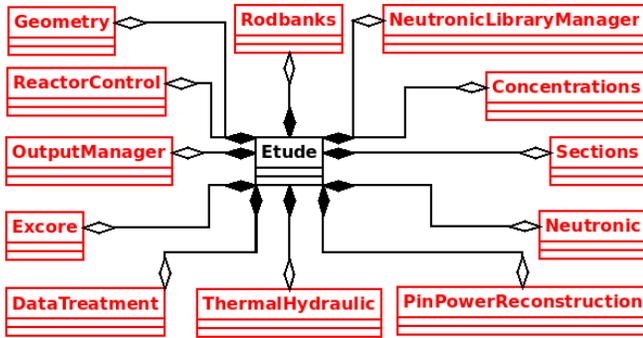


Fig. 2: COCAGNE Core Modeling Services

that for the state-of-the-art calculations, every code should concentrate on just one physics, and to perform a full core calculation SALOME should provide the tools to supervise the coupling, the data exchanged between codes (field values on meshes along with mesh projectors) and also being able to display and post-processing results. COCAGNE has simplified thermal-hydraulic and fuel thermal-mechanics codes. The latter are used to compute thermal feedback parameters. They are a good compromise between speed and calculations quality but there is ongoing work to allow COCAGNE to couple with more precise thermal-hydraulics such as the THYC code, developed at EDF. To achieve this, COCAGNE has been made a component of SALOME.

One nice side-effect is that once COCAGNE is a component of SALOME, the latter automatically provides visualising tools (Fig. 3) [3].

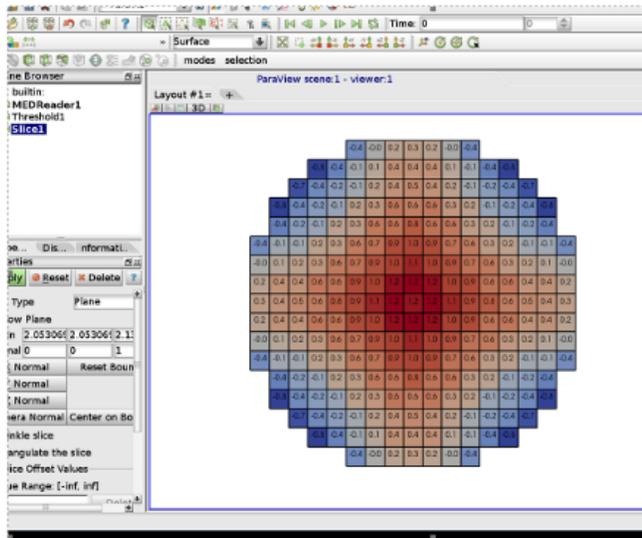


Fig. 3: COCAGNE embedded inside SALOME

### III. PHYSICAL FEATURES

#### 1. Nuclear Data Library Generation

The new EDF calculation scheme is based on a classical two-step approach (Fig. 4).

The transport calculations of the lattice scheme are

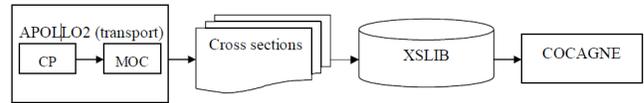


Fig. 4: EDF new calculation scheme

based on the deterministic Light Water Reactor computational scheme, REL2005 [2], developed at the *Commissariat à l'Énergie Atomique et aux Énergies Alternatives, CEA*. The 2D lattice transport deterministic code used for this purpose is APOLLO2 [1], which uses a multicell collision probability approach (CP) and a full heterogeneous transport solver based on the method of characteristics (MOC). The multigroup cross section library associated with the REL2005 scheme uses the SHEM 281-group energy mesh [4]. Isotopic cross sections in the multigroup library used by APOLLO2 are taken from CEA2005v.4 library that is mainly based on the JEFF 3.1.1 evaluations.

In order to produce data depending on burnup, the assembly calculation is coupled to the APOLLO2 depletion solver for the Bateman equations. This depletion calculation is performed at reference (nominal) conditions with a given fuel and moderator temperatures and a fixed boron concentration. Then, isotopic concentrations associated with each burnup step are stored in the library.

Cross sections depend on a few feedback parameters such as fuel temperature, boron concentration and so on which span a phase space. A calculation point is an APOLLO2 calculation for a given point in this phase space. Since there are thousands of calculation points to perform for a given assembly, an application software has been developed at EDF R&D named GAB v2 (Automatic Library Generator). For every calculation point (i.e. set of feedback parameters values), GAB v2 generates an APOLLO2 deck file, distributes all APOLLO2 calculations on a cluster, and finally merges the results in a single library file (CEA SAPHYB file for cross sections). This file is then converted into a more convenient one, in HDF5 format, the XSLIB.

Besides cross sections and concentrations, XSLIB contains all the input data needed to perform calculations such as technological data (geometry of components – assemblies, pins or grids, positions of burnable poisons, instrumentation types, clad constituents. . .) or depletion chain (will be further detailed in this paper).

Furthermore, the energy mesh may vary from 2 to 26 groups, the latter limit is due to the energy range of the second step of the lattice calculation scheme.

#### 2. Cross-sections Homogenization

From APOLLO2 calculations on an assembly, COCAGNE can use three different levels of homogenization, shown in fig. 5: assembly, multi-domain (such as 4x4 regions with 3 different materials) and cell level (17x17 cells and 1/8 symmetry, leading to 45 different materials).

All these homogenizations are available inside the cross section library (if needed) and the COCAGNE user just has to specify which homogenization he chooses using a specific keyword for the corresponding data to be loaded: cross-sections,

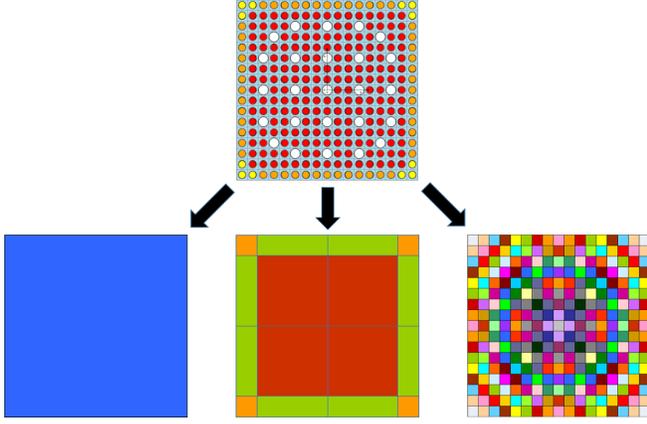


Fig. 5: Homogenization levels in COCAGNE

equivalence factors, mesh at the assembly level and materials positions. However, since COCAGNE accepts only conform, Cartesian meshes, the homogenization level must be the same for the whole core.

For all heterogeneous homogenizations, standard flux-volume homogenized cross-sections are performed during the APOLLO2 calculation. With:

- $\Sigma_{X,m}^g$  the cross-sections for reaction  $X$  and energy group  $g$ , in cell  $m$ ;
- $\phi_m^g$  the flux for energy group  $g$ , in cell  $m$  (whose volume is  $V_m$ ).
- $M$  a macro-cell, gathering one or several cells  $m$  from the reference geometry;
- $G$  a macro energy-group, gathering one or several energy group  $g$ ;

the flux in macro-cell  $M$  and macro energy-group  $G$  is:

$$\phi_M^G = \frac{\sum_{m \in M, g \in G} V_m \phi_m^g}{\sum_{m \in M} V_m}, \quad (1)$$

and preserving the reaction rates results into:

$$\Sigma_{X,M}^G = \frac{\sum_{m \in M, g \in G} V_m \Sigma_{X,m}^g \phi_m^g}{\sum_{m \in M, g \in G} V_m \phi_m^g}. \quad (2)$$

However, solving the problem on macro-cells and macro energy-groups using  $\Sigma_{X,M}^G$  leads to a flux  $\tilde{\psi}_M^G$  different from  $\phi_M^G$ . Therefore, one needs to account for the loss of information from the energy collapsing, spatial homogenization and also the potential change of operator (diffusion or SPn), which is done in COCAGNE by using SPH coefficients [5] to preserve reaction rates at the assembly level.

The SPH equivalence process consists in modifying the cross-sections in order to preserve the reaction rates from the reference APOLLO2 calculation. Let  $\tilde{\Sigma}_{X,M}^G$  be such a modified

cross-section, then reaction rates preservation ensures that the ratio

$$\lambda_{X,M}^G = \frac{\tilde{\Sigma}_{X,M}^G}{\Sigma_{X,M}^G} = \frac{\left( \frac{\tilde{\Sigma}_{X,M}^G \tilde{\psi}_M^G}{\Sigma_{X,M}^G \phi_M^G} \right)}{=1. \text{ from equivalence}} \frac{\phi_M^G}{\tilde{\psi}_M^G} = \frac{\phi_M^G}{\tilde{\psi}_M^G} = \lambda_M^G \quad (3)$$

is the same for all reactions.

Iterative research is performed to solve this equivalence problem (i.e. compute  $\lambda_{X,M}^G$ ). At this stage, one should remember that those coefficients depend not only on the cross sections feedback parameters, but also on the choice of the (approximate-)transport operator and spatial discretization (meshing, and nodal or finite element). In order to ensure the consistency with all the solvers available in COCAGNE, the SPH equivalence is performed with COCAGNE during the library computation process (SPH coefficients and flux-volume weighted cross sections are stored separately, so that switching from one solver to another can be done within the same calculation).

Lastly, in order not to spread the use of SPH factors all over the code, the flux definition takes into account those coefficients:

$$\psi_M^G = \lambda_M^G \tilde{\psi}_M^G$$

and therefore can be used straightforwardly with flux-volume weighed cross-sections to compute reaction rates everywhere else in the code (e.g. microscopic depletion):

$$\tau_{X,M}^G = \tilde{\Sigma}_{X,M}^G \tilde{\psi}_M^G = \Sigma_{X,M}^G \psi_M^G.$$

To solve SPH equivalence, reaction rates preservation usually doesn't provide enough constraints to fully close the system. As a matter of fact, one degree of freedom per energy group is usually available (called in COCAGNE cross section normalization coefficients), and can be used to select which additional quantity one can preserve (such as mean flux, boundary flux, or boundary partial current) [6]. When applied to fully homogenized assembly, those coefficients act as assembly discontinuity factors.

### 3. Cross sections fitting and microscopic depletion

Cross sections vary from one (physical-)point of the core to another and depend on local feedback parameters. They are computed using tabulated sections from the nuclear library. COCAGNE has two cross-sections models:

- the legacy one, Lefebvre-Seban, used in the current core code, COCCINELLE [7],
- and the new one, relying on multi-linear interpolation.

Lefebvre-Seban model is used for V&V purposes only (see the corresponding section) since it has far less features than the new one. It is worth noting that in COCAGNE, changing the interpolation model is almost straightforward since the user has just to write at the beginning of its study which model he chooses, what library to read from, and all the rest of the script is the same since both CMS interface and Core

Procedures are model independent. Both models have been implemented in COCAGNE like “plugins”, meaning one model is completely independent from one other. This will allow us in the near future to implement a third model that has been studied through a Ph.D thesis work, the Tucker-decomposition model [8] without impacting the existing ones.

Macroscopic cross-sections reconstructions using linear interpolation can be computed using either interpolation of the macroscopic cross-sections stored in the library, or using microscopic cross-sections according to the following formula:

$$\Sigma_X^g = \sum_{i=1}^N N_i \tilde{\sigma}_{X,i}^g + \tilde{\Sigma}_{X,res}^g \quad (4)$$

where:

- $N_i$  is the concentration of nuclide  $i$ ,
- $\tilde{\sigma}_{X,i}^g$  is the tabulated microscopic cross-section for particularized nuclide  $i$ , reaction  $X$  and energy group  $g$ ,
- $\tilde{\Sigma}_{X,res}^g$  is the residual cross-section (taking into account all non-particularized nuclides).

Instead of (4), the following alternative formula allows choosing a subset  $I$  of nuclides to be particularized:

$$\Sigma_X^g = \sum_{i \in I} (N_i - \tilde{N}_i) \tilde{\sigma}_{X,i}^g + \tilde{\Sigma}_X^g, \quad (5)$$

where the concentrations and macroscopic cross-section in nominal depletion conditions are denoted by  $\tilde{N}_i$  and  $\tilde{\Sigma}_X^g$  respectively.

Concentrations are computed solving Bateman’s equations, and in COCAGNE this is done using either:

- Runge-Kutta method of order 4 with constant time step,
- or embedded Runge-Kutta methods of orders 4 and 5 with adaptive steps length (Cash-Karp method [9]).

In any case, in COCAGNE we assume that the depletion matrix is constant during all depletion time. A predictor-corrector scheme has also been implemented that allows to improve the precision while having reasonable steps size for industrial use [10].

We will not discuss here the exact depletion chain used in COCAGNE, suffice to know that it is composed of roughly 20 heavy nuclides and 20 fission products.

The depletion chain used in COCAGNE is made of particularized isotopes and a fictive one (the so-called "residual") from the more complete depletion chain used in APOLLO2. For consistency purposes, the choice has been made to store it inside the XSLIB ; that prevents discrepancies that may arise if the code got a chain from another external file and that chain was not the one used in APOLLO2. At that time for a given assembly there is one depletion chain associated to it. It is worth noting that COCAGNE depletion module deals with different depletion chains associated to different assembly types [11].

Another point of interest : both macroscopic and microscopic models have historical corrections that improve accuracy (the latter one uses Pu239 concentration as a spectral index) [12].

Cross-sections reconstruction using microscopic depletion is calculation intensive and it would be useless unless heavily optimized, especially since flux solvers are very fast. Therefore a particular care has been taken in order to minimize memory bandwidth bottlenecks as well as to parallelize execution in a shared memory environment [11]. Vectorization has yet to be implemented though.

#### 4. Pin power distribution

For reactor safety analyses, the power distribution at the pin level has to be computed. Whereas it is straightforward if core computation is pin-cell resolved (at the cost of longer computing time), pin power reconstruction has to be done for any other homogenization, since detailed informations were lost during the homogenization step.

This reconstruction relies on the hypothesis that the flux in a reactor can be approximated by the product of [13]:

- a macroscopic distribution,  $\psi$ , computed from homogenized cross-sections;
- a local form function  $\Psi$ , taking into account the local heterogeneities within the fuel assembly (water holes, burnable absorber pins, enrichment variations, ...).

The first term, though, has to be projected on a pin-size mesh, whereas it is usually computed on much coarser cells. See part IV.1. for more details.

Considering the form function, it used to be a rather simple normalization for each energy group of the single-assembly heterogeneous flux, but this definition only stands for full assembly homogenization. For the more general case of multi-domain homogenization, the form functions are defined as:

$$\Psi_c^G = \frac{\int_c \phi_c^{G,\infty} dV}{\int_c \psi_M^{G,\infty} dV} \quad (6)$$

where:

- $\phi_c^{G,\infty}$  is the APOLLO2 heterogeneous flux (averaged at the pin-cell level, and integrated on macro-energy group  $G$ ),
- $\psi_M^{G,\infty}$  is the COCAGNE flux computed with multi-domain homogenized cross-sections on macro-cells  $M$ ,

both calculated from single-assembly configuration.

Therefore, the form functions  $\Psi$  depends not only on the homogenization geometry, but also on the transport operator ( $SP_n, S_n$ ) and the spatial discretization. Also, one should remember that  $\Psi$  implicitly depends on SPH equivalence factors that were used to compute  $\psi_M^{G,\infty}$ .

The pin power reconstruction using total power form function is also available in COCAGNE, and usually leads to very

similar results. Discrepancies can only be seen when flux spectrum shifts from single-assembly value, and energy group-wise power distribution at pin cell is far away from homogenized distribution. One typical example is the peripheral pins of a zoned MOX assembly.

#### IV. FLUX SOLVERS

There are currently two flux solvers that are part of COCAGNE, both of them implemented for 3D Cartesian geometries. The first one is the solver named Diabolo which is a simplified transport ( $SP_n$ ) solver and the second one is named Domino and is a Discrete Ordinates ( $S_n$ ) transport solver. Both of them can solve static and kinetics equations. Both solvers use the multi-group formalism to handle energy, and rely on a Cartesian mesh for their spatial discretization.

##### 1. Diabolo $SP_n$ solver

Diabolo [14] uses Raviart-Thomas-Nedelec finite elements ( $RT_k$ ) to solve the Simplified  $P_n$  ( $SP_n$ ) equations in a mixed-dual formulation. Diabolo can solve the  $SP_1$ ,  $SP_3$  and  $SP_5$  sets of static equations, but kinetic calculations are limited to  $SP_1$ . The first three spatial discretization orders ( $RT_0$ ,  $RT_1$  and  $RT_2$ ) are available, and can be used independently for any direction. Mixed spatial discretizations are thus allowed, such as the  $RT_{221}$  configuration: a second-order approximation in the  $x$  and  $y$  directions combined with a first-order approximation along the  $z$  axis.

Using  $RT_k$  elements in a mixed-dual formulation on Cartesian meshes is a good choice as far as flux solvers are concerned: with such a discretization, degrees of freedom (DOFs) for the current along different directions are only coupled through the flux DOFs. This allows for a very efficient solution of the resulting  $SP_n$  system using an alternate directions iterative method. However, discretizing the mixed dual formulation also has a non-negligible drawback: much more DOFs are dedicated to represent the neutron current (which is projected on the space of polynomials of order  $k + 1$  in  $RT_k$  discretization), than to represent the neutron flux (projected on polynomials of order  $k$ ). This means that, when trying to reconstruct the shape of the flux within an homogenized region (as explained in section 4.), using only the information contained in flux DOFs will yield a fairly low order of spatial convergence. In order to overcome this limitation, methods have been developed in COCAGNE to recover fine, within-region flux variations from the more detailed information contained in current DOFs [15].

##### 2. Domino $S_n$ solver

Domino [16] uses the Discrete Ordinates formalism ( $S_n$ ) to angularly discretize the multi-group transport equation. Domino is able to solve both static and kinetic problems [17]. A choice is offered between various Level Symmetric quadrature formulas. A Diamond Differencing scheme (DD) is used for the spatial discretization on Cartesian meshes.

As always when  $S_n$  solvers are used in highly optically-thick and diffusive media such as found in PWRs, source

iterations need to be accelerated. This is done in Domino using a Diffusion Synthetic Acceleration (DSA) method. Using a lowest-order DD scheme for the spatial discretization of the transport equation presents the advantage that the acceleration scheme can re-use the  $SP_1$ ,  $RT_0$  solver in Diabolo, which is consistent.

Although this has not yet led to any new feature in Domino, some theoretical work has recently been devoted to a more in-depth analysis of the Diamond Differences scheme and the derivation of error estimators for it [18].

#### 3. Solvers efficiency

Since the internal data model of COCAGNE is not distributed, these solvers are not parallelized for distributed memory architecture. This is consistent with the fact that COCAGNE is primarily a tool for industrial uses where the availability of distributed computation nodes in production is scarce and priority has been given to shared memory platforms. As such, these solvers have been optimized using the high efficiency concept: to achieve high performance calculations, optimization of the available resources is preferred over multiplying the resources. Both Diabolo and Domino thus try and take advantage of all computation resources available on a single node [19, 20]:

- all CPU cores on a node are harnessed via shared memory parallelization, which is implemented using the Intel® Threading Building Blocks library<sup>1</sup>;
- vector units, i.e. SSE or AVX units implementing SIMD calculations in modern CPUs, are addressed using the Eigen library<sup>2</sup>.

Also, since most calculations are limited by memory bandwidth (memory bound), on-the-fly calculations have been preferred over keeping data in memory. Most of the linear algebra taking place in the solvers thus involves sparse, structured matrices whose coefficients are never stored, but rather recomputed on-the-fly. The practical implementation of this technique is delegated to a specific library called Legolas [21].

Another benefit of having multiple solvers in the same platform is that hybrid approaches can be implemented, where a fast  $SP_n$  computation helps initializing the  $S_n$  solver to get a more precise solution at a fraction of the total cost of the transport solver [22].

The first 3 lines in Table I give examples of computing times for 3D PWR criticality calculations with the solvers in COCAGNE, on a single cluster node. It is interesting to note that industrial 2-group diffusion calculations take less than a minute to run, while finer computations can be run with the same platform to obtain reference results in a very reasonable amount of time.

#### 4. Next-generation solvers

Although it is not yet available in the platform, much work has been devoted in the past few years to develop a distributed-

<sup>1</sup><http://www.threadingbuildingblocks.org/>

<sup>2</sup><http://eigen.tuxfamily.org/>

Solver	Energy groups	Discretization	Spatial mesh (#meshes / pin cell)	#DOFs	#CPU cores	Time
Diabolo (see §V.3.)	2	SP <sub>1</sub> , RT <sub>1</sub>	1×1	$3.3 \times 10^8$	24	46 sec.
Diabolo (see §V.3.)	8	SP <sub>3</sub> , RT <sub>1</sub>	1×1	$1.6 \times 10^9$	24	98 min.
Domino [16]	8	S <sub>8</sub>	2×2	$1.1 \times 10^9$	32	61 min.
Domino (next-gen) [23]	26	S <sub>16</sub>	2×2	$1.9 \times 10^{12}$	1536	46 min.

TABLE I: Examples of solvers performance for 3D PWR core criticality calculations

memory version of the Domino transport solver [23].

This solver achieves very high performance by harnessing the three levels of parallelism available on modern computation infrastructures: cluster nodes, CPU cores and vector units. The two levels of parallelism (distributed and shared memory) are coordinated by the PaRSEC [24] runtime system, while computation kernels take advantage of local vector units using the Eigen library. An example of the performances achieved by this new version of Domino is presented in the last line of Table I (these results were obtained using 64 computing nodes of the Athos<sup>3</sup> cluster).

Another distributed-memory solver, that is not integrated in the COCAGNE platform yet, is Micado [25]. This solver solves the multigroup,  $S_n$  transport equation in prismatic geometries described as an unstructured 2D radial mesh, extruded in the axial direction. Data structures in the rest of the platform are not yet ready to represent such detailed geometries, which makes Micado’s integration into COCAGNE a mid- to long-term perspective.

The spatial resolution in Micado uses a fusion-like iterative method that couples 2D radial MOC computations on geometric “slices” (using a Step Characteristics or a Diamond Differences discretization along characteristic lines) with 1D axial resolutions of the transport equation on cell “columns”. The whole iterative algorithm is accelerated by a pCMFD method. Like the next-generation Domino mentioned above, Micado features three levels of parallelism: distributed memory (MPI), shared-memory parallelism (tbb), and vectorization (Eigen) [19].

## V. VERIFICATION AND VALIDATION (V&V)

Before the licensing of new tools or their application for production purposes, extensive V&V is required. For COCAGNE, V&V has started right from the initial developing process. This section describes the V&V which has been on-going since the very outset of COCAGNE.

### 1. Verification

Verification of the code is first carried out through unit tests in both the C++ and python parts and are the developers’ responsibility to write and document them. COCAGNE has its own utilities for C++ tests suites which is useful for non-regression where the result is not expected to be exact but up to some precision defined by the test maker. Reference

results files (in both XML format and HDF5 format) can be more complex than just floating values and includes values for fields defined on meshes. Besides, the code relies on the standard `cctest` utilities.

For the upper layer where python is involved, we have also developed an automatic test generator that strips python doc-strings written with the `restructuredText` format that allows tables, pictures and complex formulas using latex when needed. Stripped data are fed to a template for production of a pdf document. Everything is automatic as much as possible.

Preliminary validation at assembly and core level is done by comparing results to the current code used by engineering and production teams on an everyday basis to compute the reactors of the EDF fleet: COCCINELLE. This has been made possible because COCAGNE has the ability to read the same nuclear data as COCINELLE from the same cross-sections library with the same feedback model, specially implemented for this purpose. Since the SP1 solver in COCAGNE solves the same diffusion equations than the nodal code in COCINELLE, and the simplified thermalhydraulic code is the same, results should be identical (provided the solvers have converged). Those tests checked the correctness of the core calculations sequence which can be tricky especially with feedback parameters. Once we have the same results, it is simple to change the new feedback model modifying just a few lines at the beginning of the script. We can then analyse the effects due to the new assembly scheme (REL2005) and the cross section reconstruction model only.

### 2. Validation of physical models

As far as the developing phase is concerned, a highly efficient framework has been set up to ensure that verification and validation results are maintained. Furthermore, during the past years, several studies have been carried out to validate the physical models implemented in COCAGNE as well as the improvement of cross section libraries.

#### A. Reactivity and pin power distribution

COCAGNE calculations are based on nuclear data stemming from the JEFF3.1.1 data and the REL2005 calculation scheme [2]. The assembly calculation scheme has been thoroughly validated on experimental data from mock-up reactors by the CEA. Thus, the quality of the homogenized few-group cross sections for the core code is ensured.

For its industrial scheme, ANDROMEDE uses two-group diffusion calculations with cross sections generated through the REL2005 scheme. The extensive validation of the core

<sup>3</sup>Athos: <https://www.top500.org/system/178201>

calculations will be carried out in the forthcoming years by comparing core calculations from ANDROMEDE to measured data from the fifty-eight reactors of the EDF fleet which have been under production for more than thirty years. In the meantime, COCAGNE has been validated using numerical benchmarks and measurements on some selected reactors. For confidentiality purposes, results on measured reactor data are not published and thus, only those from numerical benchmarks are available.

Preliminary validation has been carried out on assemblies in infinite lattice and  $3 \times 3$  assembly clusters (to get more realistic interface effects without being subjected to reflector problems for analysis). Equivalence, normalization and pin power reconstruction models have been tested, corrected and studied thoroughly on such configurations [26, 6] before applying them full-scale on reactor problems. The results proved to be very encouraging, especially with reactivity and pin power distributions being compared to multigroup best-estimate computations from APOLLO2 (26-group fully-heterogeneous MOC) – less than 250 pcm and 2%.

In the past few years, we have been actively working on 2D benchmark from the KAIST institute in Korea, called the KAIST 1A benchmark [27]. It is a small core with high leakage and reflector effects, along with very heterogeneous assemblies (MOX, Gadolinia pins). In 2014, we published the results on this benchmark ranging from Monte Carlo calculations to industrial calculation scheme, and showed that the reactivity is predicted with less than 130 pcm for the unrodded configuration and 350 pcm for the rodded one. Pin power distributions are predicted with an RMS discrepancy of 2.5% [28] for the rodded configuration.

These results are extremely satisfactory for an industrial calculation scheme and should be confirmed by extended validation works in years to come.

### B. Microscopic depletion

Compared to its predecessor, one of the main features of the COCAGNE code is the use of a highly performing microscopic depletion solver [10]. COCCINELLE uses macroscopic depletion and therefore cannot track the depletion of the main isotopes, thereby leading to historical effects over the cycles. Such effects are alleviated with the use of a microscopic depletion solver. Much work has been devoted in the last years to validate the depletion calculation of the COCAGNE code. As for the previous calculations, selected reactor measurements for critical boron concentration have been computed with COCAGNE, but are confidential. Nevertheless, published results include the work on assembly clusters [12] and on the extension of the KAIST 1A benchmark to depletion calculation for validation of industrial schemes [29]. The results were very good with the boron concentration being predicted below 10 ppm and pin power distributions over the cycle follow the same trend as given in the previous paragraph.

Moreover, the microscopic depletion solver of COCAGNE has been employed to set up the *library approximation method* [30]. The latter consists in using the depletion solver to approximated technological heterogeneities such as enrichment or the cooling of MOX assemblies before they

are actually loaded in the core. This method is useful for decreasing the number of cross section libraries required for core calculations.

### C. Advanced core computation schemes and models

One of the main features of the COCAGNE code is the possibility of engaging into pin-homogeneous computations. Indeed, COCCINELLE is limited to assembly-homogeneous calculations only. In COCAGNE, several homogenization levels are possible, ranging from coarse assembly-homogeneous to pin-homogeneous calculations. For industrial calculation schemes, an intermediate homogenization level has been defined for dealing with heterogeneous configurations. Besides, pin-homogeneous calculations with two group diffusion has been employed for validating more accurate schemes.

Furthermore, pin-by-pin calculations have been optimized for their use with the  $SP_n$  and  $S_n$  solvers of the COCAGNE platform [31]. COCAGNE provides a unique framework whereby the user can access two-group diffusion calculations up to multigroup transport calculations for a given problem set. Multigroup cross section libraries are generated using the REL2005 which ensures high-quality cross sections up to 26 groups, derived from the SHEM 281-group energy mesh. Thus, thanks to COCAGNE, *advanced industrial schemes* based on  $SP_n$  calculations have been conceived and employed recently for validating industrial methods [28, 29]. These schemes have been applied to the previous problems for zero-burnup and depletion calculations. It has been observed that thanks to the simplified transport method, the pin power discrepancy decreases below 1%.

Furthermore, multigroup calculations implied that multigroup reflector models were required. For two group diffusion calculation, albedo are conserved at the core-reflector interface using the Lefebvre-Lebigot model. However, for multigroup calculations, a new methodology was required and has been set up in [32]. It consists in using a 1D slab geometry to generate cross sections which are tabulated with the feedback quantities. The model proved to be particularly satisfying as it led to results which were very satisfying on power distributions as given by [32].

To boot, the COCAGNE code coupled to the data assimilation module (ADAO) of the SALOME platform has been very useful for studies with data assimilation techniques as given by [33] for the reproduction of an azimuthal power disbalance or by [34] for determining key reflector parameters.

## 3. Application

We will illustrate here one example of the many applications of COCAGNE: calculations of the new Evolutionary Power Reactor (EPR).

Since the EPR is quite different from previous reactors (it has for instance an heavy neutron reflector), when creating a new industrial calculation scheme for it it was important to have reference calculations.

This is where COCAGNE shines since from the same general scheme, it allows to switch very easily from one solver to another and thus from a reference scheme using transport equation ( $S_n$  and many groups) to an advanced scheme using

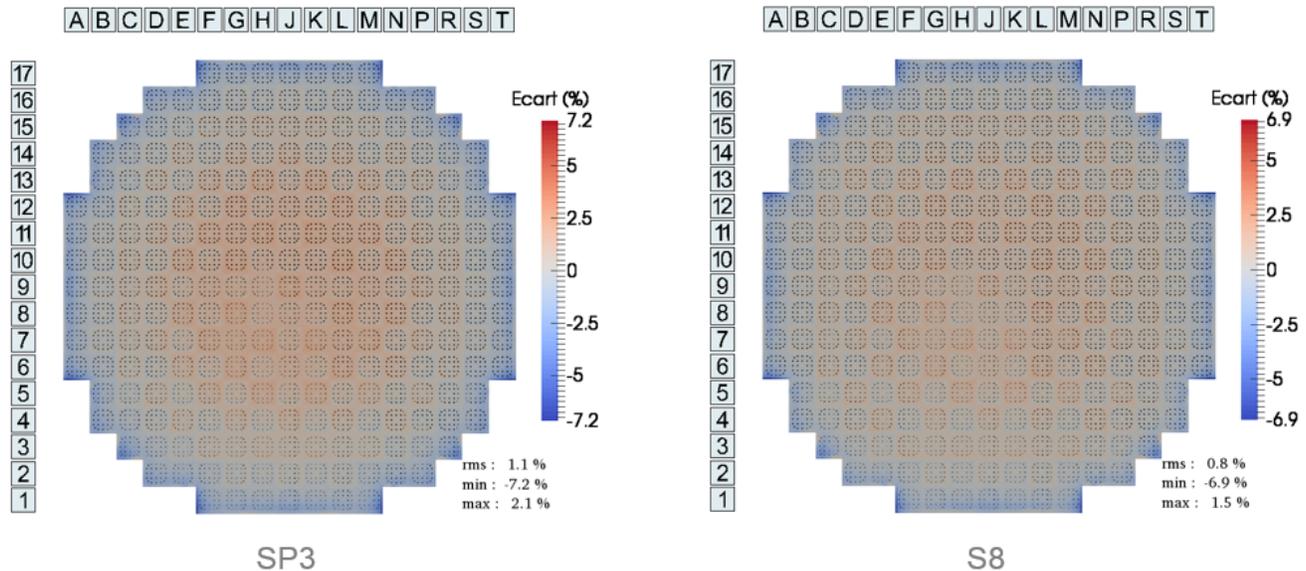


Fig. 6: Power distribution with advanced calculation schemes compared to APOLLO2

simplified transport equation ( $SP_n$  with  $n > 1$  and many energy groups) and finally to the industrial scheme (2 groups diffusion solver). Moving from one scheme to another, we gained each time some speed, while controlling the quality of results. Reference calculations are performed only once in a while when new configurations arise, while the advanced scheme is used frequently to compare with the industrial scheme in development. But those calculations use a *two-step approach* with a lattice code in the first step. In order to have a real, *one-step calculation* reference, we used the APOLLO2 code, the drawback being that it is a 2D reference calculation only. At this time a real 3D *one-step* calculation is out of reach, the EPR being too big for a Monte-Carlo approach. We hope that in the near future, APOLLO3<sup>®</sup> [35] will allow us to have true 3D reference calculations.

Table II and Fig. 6 present some of the results obtained after carefully optimizing the energy mesh. One aspect of particular importance in this study was the reflector model: we used techniques similar to the ones presented in [32] with a 1D slab reflector. As we can see, the scheme using an  $S_3$  solver gives very good results when compared with APOLLO2 in 2D. Although it might seem at first that large negative discrepancies can occur, these are actually limited to a very small number of pins at corners of the core–reflector interface. All in all, global RMS discrepancies are maintained below 1%, which gives us good confidence that such a scheme could be considered as an industrial 3D reference.

Once the industrial scheme was established, we moved on to a simplified one, that would be fast enough to be used on a regular basis. This scheme uses an  $SP_3$  simplified transport solver and is, as we can see, very close to the reference scheme. Global RMS discrepancies with respect to APOLLO2 are just above 1%. Once again, the rather large negative discrepancies are only obtained for a limited number of pins located at the

corners of the core–reflector interface.

2D Pin Power comparison to APOLLO2			
	Min.	Max.	RMS
$SP_3$ pin by pin 8 gr.	-7.2 %	+2.1 %	1.1 %
$S_3$ pin by pin 8 gr.	-6.9 %	+1.5 %	0.8 %

TABLE II:  $SP_3$  and  $S_3$  comparison to APOLLO2

## VI. CONCLUSIONS

We presented in this paper the new core code COCAGNE, with some of its features and state of the art solvers. One of its strengths is that advanced models and schemes are available in the same simulation platform, which allows

- setting up the industrial scheme and strengthen its V&V basis (evaluation of each effect separately),
- promoting expertise works (fine understanding and quantifying a phenomenon),
- putting up with the fact that some measurements are not available (kinetics, new designs. . .)

When compared to the previous code COCCINELLE, it has a far greater flexibility due to the python language and allows engineers to create innovative and advanced schemes. Furthermore, since it is part of the SALOME component, it can be coupled to other codes specialized in different physics aspects such as thermalhydraulics to achieve a high fidelity simulation. COCAGNE is therefore a cornerstone for EDF's future core calculation chain ANDROMEDE.

## REFERENCES

1. R. SANCHEZ, I. ZMIJAREVIC, M. COSTE-DELCLAUX, E. MASIELLO, S. SANTANDREA, E. MARTINOLLI, L. VILLATE, N. SCHWARTZ, and N. GULER, "APOLLO2 Year 2010," *Nuclear Engineering and Technology*, **42**, 474–499 (2010).
2. J.-F. VIDAL, O. LITAIZE, and E. A. D. BERNARD, "New modelling of LWR assemblies using the APOLLO2 code package," in "Proceedings of Joint International Topical Meeting on Mathematics and Computations and Supercomputing in Nuclear Applications, M & C + SNA," Monterey, CA (April 2007).
3. H. LEROYER, G. BOULANT, A. CALLOO, Y. PORA, M. FLISCOUNAKIS, T. CLERC, and R. BARATE, "SALOME-CORE platform: uses for EDF R&D neutronic studies," in "Proceedings of PHYSOR 2014," Kyoto, Japan (2014).
4. N. HFAIEDH and A. SANTAMARINA, "Determination of the Optimized SHEM Mesh for Neutron Transport Calculations," in "Topical Meeting on Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Application (M&C)," Avignon, France (2005).
5. A. HÉBERT, *Applied Reactor Physics*, Presses Internationales Polytechnique (2009).
6. C. BROSSELARD, H. LEROYER, M. FLISCOUNAKIS, E. GIRARDI, and D. COUYRAS, "Normalization methods for diffusion calculations with various assembly homogenizations," in "ANS Reactor Physics Topical Meeting (PHYSOR)," Kyoto, Japan (2014).
7. J.-P. WEST, F. BLANCHON, J. PLANCHARD, J.-P. GREGOIRE, S. MARGUET, D. VERWAERDE, G. NICOLAS, H. SCHAEFFER, A. VASSALLO, and J.-C. BARRAL, "COCCINELLE: a consistent software for light water reactor physics calculations; Design, Safety, Management, Monitoring and Surveillance," in "International Conference on the Physics of Reactors: Operation, Design and Computation (PHYSOR)," Marseille, France (April 1990).
8. T.-H. LUU, M. GUILLO, and P. GUÉRIN, "Reconstruction of Neutronic Macroscopic Cross-Sections using Tucker Decomposition," in "Proceedings of M&C 2015," Knoxville, Tennessee, USA (May 2015).
9. J. R. CASH and A. H. KARP, "A variable order Runge-Kutta method for initial value problems with rapidly varying right-hand sides," *ACM Transactions on Mathematical Software*, **16**, 201–222 (1990).
10. F. HOAREAU, N. SCHWARTZ, and D. COUYRAS, "A Predictor-Corrector Scheme for the Microscopic Depletion Solver of the COCAGNE Core Code," in "18th International Conference on Nuclear Engineering (ICONE)," Xi'an, China (2010).
11. M. GUILLO, D. COUYRAS, F. FÉVOTTE, and F. HOAREAU, "The Optimized Algorithm for the Microscopic Depletion Model in the COCAGNE Core Code – A 2-level Core Partitioning Approach," in "ANS Reactor Physics Topical Meeting (PHYSOR)," Kyoto, Japan (2014).
12. F. HOAREAU, E. GIRARDI, C. BROSSELARD, and M. FLISCOUNAKIS, "Verification of the COCAGNE core code using cluster depletion calculations," in "ANS Reactor Physics Topical Meeting (PHYSOR)," Kyoto, Japan (2014).
13. K. R. REMPE, K. S. SMITH, and A. F. HENRY, "SIMULATE-3 Pin Power Reconstruction: Methodology and Benchmarking," *Nuclear Engineering and Technology*, **103**, 334–342 (1989).
14. W. KIRSCHENMANN, L. PLAGNE, A. PONÇOT, and S. VIALLE, "Parallel SPN on Multi-Core CPUs and Many-Core GPUs," *Transport Theory and Statistical Physics*, **39**, 2–4, 255–281 (2010).
15. D. COUYRAS, F. FÉVOTTE, and L. PLAGNE, "Novel Fine Flux Integration Methods for the DIABOLO Simplified Transport ( $SP_N$ ) solver in COCAGNE," in "International Conference on Mathematics and Computations (M&C)," Sun Valley, ID, USA (May 2013).
16. T. COURAU, S. MOUSTAFA, L. PLAGNE, and A. PONÇOT, "DOMINO: a Fast 3D Cartesian Discrete Ordinates Solver for Reference PWR Simulations and SPN Validation," in "International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C)," Sun Valley, Idaho, USA (2013).
17. A. CALLOO, A. PONÇOT, and D. COUYRAS, "Time-dependent  $S_n$  Method in the DOMINO Solver of the Cocagne Platform," in "International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C)," Jeju, Korea (2017).
18. R. S. JEFFERS, J. KÓPHÁZI, M. D. EATON, F. FÉVOTTE, F. HÜLSEMANN, and J. RAGUSA, "Goal-based h-adaptivity of the 1-D diamond difference discrete ordinate method," *Journal of Computational Physics*, **335**, 179 – 200 (2017).
19. S. MOUSTAFA, F. FÉVOTTE, B. LATHULIÈRE, and L. PLAGNE, "Vectorization of a 2D–1D Iterative Algorithm for the 3D Neutron Transport Problem in Prismatic Geometries," in "International Conference on Supercomputing in Nuclear Applications (SNA)," Paris, France (Sep. 2013).
20. S. MOUSTAFA, I. DUTKA-MALEN, L. PLAGNE, A. PONÇOT, and P. RAMET, "Shared Memory Parallelism for 3D Cartesian Discrete Ordinates Solver," *Annals of Nuclear Energy*, **82** (2013).
21. W. KIRSCHENMANN, L. PLAGNE, and S. VIALLE, "Multi-Target Vectorization with the MTPS C++ Generic Library," in "Applied Parallel and Scientific Computing International Conference (PARA)," Reykjavik, Iceland (2010).
22. T. COURAU, L. PLAGNE, A. PONÇOT, and G. SJODEN, "Hybrid Parallel Code Acceleration Methods in Full-Core Reactor Physics Calculations," in "ANS Reactor Physics Topical Meeting (PHYSOR)," Knoxville, Tennessee, USA (2012).
23. S. MOUSTAFA, *Massively Parallel Cartesian Discrete Ordinates Method for Neutron Transport Simulation*, Ph.D. thesis, Université de Bordeaux (2015).

24. G. BOSILCA, A. BOUTEILLER, A. DANALIS, T. HÉRAULT, P. LEMARINIER, and J. DONGARRA, “DAGuE: a Generic Distributed DAG Engine for High Performance Computing,” *Parallel Computing*, **38**, 1–2 (2012).
25. F. FÉVOTTE and B. LATHUILLIÈRE, “MICADO: Parallel implementation of a 2D–1D iterative algorithm for the 3D neutron transport problem in prismatic geometries,” in “International Conference on Mathematics and Computations (M&C),” Sun Valley, ID, USA (May 2013).
26. M. FLISCOUNAKIS, E. GIRARDI, and T. COURAU, “A generalized pin-power reconstruction method for arbitrary heterogeneous geometries,” in “Proceedings of M&C 2011,” Rio de Janeiro, Brazil (May 2011).
27. N. Z. CHO, “KAIST Nuclear Reactor Analysis and Particle Transport Laboratory, *Benchmark Problem 1A*,” <http://nurapt.kaist.ac.kr/benchmark>.
28. A. CALLOO, H. LEROYER, M. FLISCOUNAKIS, and D. COUYRAS, “Core Neutronics Methodologies Applied to the MOX-loaded KAIST 1A benchmark: reference to industrial calculations,” in “Proceedings of PHYSOR 2014,” The Westin Miyako, Kyoto, Japan (October 2014).
29. A. CALLOO, S. HUY, D. COUYRAS, C. BROSSELDARD, and M. FLISCOUNAKIS, “Validation of the  $SP_n$  depletion schemes of the EDF GABv2-COCAGNE tools using the KAIST 1A benchmark,” in “Proceedings of PHYSOR 2016,” Sun Valley, Idaho, USA (May 2016).
30. A. CALLOO, X. SHANG, and H. LEROYER, “The library approximation method for the future EDF chain ANDROMEDE,” in “Proceedings of M&C 2015,” Knoxville, Tennessee, USA (May 2015).
31. M. FLISCOUNAKIS, E. GIRARDI, T. COURAU, and D. COUYRAS, “Potential pin-by-pin SPN calculations as an industrial reference,” in “Proceedings of ICAPP 2012,” Chicago, USA (2012).
32. S. HUY, M. GUILLO, A. CALLOO, C. BROSSELDARD, and D. COUYRAS, “Multigroup 1D-Reflector Modelling for EDF PWR,” in “Proceedings of PHYSOR 2016,” Sun Valley, Idaho, USA (May 2016).
33. C. BROSSELDARD, B. BOURIQUET, M. DE SEQUEIRA, H. LEROYER, D. COUYRAS, and J. ARGAUD, “Reproduction of a Power Azimuthal Disbalance Using Data Assimilation: Application to Twin Experiments,” in “Proceedings of PHYSOR 2016,” Sun Valley, Idaho, USA (May 2016).
34. J.-P. ARGAUD, B. BOURIQUET, A. CALLOO, and E. LUIS, “Determination of the reflector parameter through data assimilation with the COCAGNE calculation code using a TRIPOLI4 reference simulation on KAIST benchmark,” in “Proceedings of PHYSOR 2016,” Sun Valley, Idaho, USA (May 2016).
35. D. SCHNEIDER, F. DOLCI, F. GABRIEL, J.-M. PALAU, M. GUILLO, and B. POTHET, “APOLLO3: CEA/DEN Deterministic Multi-Purpose Code for Reactor Physics Analysis,” in “Proceedings of PHYSOR 2016,” Sun Valley, Idaho, USA (May 2016).