Proceedings of ICAPP '12
Chicago, USA, June 24-28, 2012
Paper 12286

PHISICS multi-group transport neutronic capabilities for RELAP5

Aaron Epiney, Cristian Rabiti, Andrea Alfonsi, Yaqi Wang, Joshua Cogliati, Gerhard Strydom
Idaho National Laboratory (INL)
2525 N. Fremont Ave., Idaho Falls, Idaho, 83402
Tel: (208)526-2462, Fax: (208)526-0528, Email: aaron.epiney@inl.gov

Abstract – *PHISICS is a neutronic code system currently under development at INL. Its goal is to provide state of the art simulation capability to reactor designers. This paper reports on the effort of coupling this package to the thermal hydraulic system code RELAP5. This will enable full prismatic core and system modeling and the possibility to model coupled (thermal-hydraulics and neutronics) problems with more options for 3D neutron kinetics, compared to the existing diffusion theory neutron kinetics module in RELAP5 (NESTLE). The paper describes the capabilities of the coupling and illustrates them with a set of sample problems.*

I. INTRODUCTION

The RELAP5-3D code was developed for best-estimate transient simulation of light water reactors[1]. The code is able to model coupled behavior of the reactor core and the thermal-hydraulics of the power plant. Applications of the code include simulations of transients such as loss of coolant, anticipated transients without scram, and operational transients such as loss of feed water, loss of offsite power, station blackout, and turbine trip.

The multi-dimensional neutron kinetics model in RELAP5-3D is based on the NESTLE[2] code, which solves the two or four group neutron diffusion equations in either Cartesian or hexagonal geometry using the nodal expansion method.

For the simulation of advanced reactors such as Generation IV systems, in particular the Modular High Temperature Gas Reactor (MHTGR) considered by the US[3], more neutronic capabilities are desirable compared to the available NESTLE package in RELAP5-3D. Therefore the neutronic code package PHISICS has been coupled to RELAP5.

PHISICS (Parallel and Highly Innovative Simulation for INL Code System)[4] is a neutronic code system currently under development at INL. Its goal is to provide state of the art simulation capability to reactor designers. The different modules for PHISICS currently under development are a nodal and semi-structured spherical harmonics based transport core solver (INSTANT) for steady state and time dependent problems, a depletion module (MRTAU) and a cross section mixer-interpolator module.

Table 1 shows a comparison of the RELAP5-NESTLE and RELAP5-PHISICS capabilities so far. The PHISICS code is still in development to extend its capabilities.

TABLE I

RELAP5-NESTLE and RELAP5-PHISICS capabilities

Feature	RELAP5-NESTLE	RELAP5-PHISICS
Energy groups	2 or 4	Not bounded
Diffusion	Yes	Yes
Transport	No	Yes
Triangular Mesh	No	Yes
Unstructured Mesh	No	Yes
Adjoint	No	Yes
Depletion	No	Yes
Multi-Dim Cross Section Tables	No	Yes
Speed	Win	Lose (Future?)
Discontinuity Factors	Yes	Future
Cylindrical Geometry	No	Future
Perturbation Theory	No	Future
Localized refinement	No	Future

II. The PHISICS code package

The neutronic code package PHISICS follows a modular approach to simplify development and long term maintenance of the code. PHISICS is entirely written in FORTRAN 95/2003.

The general software structure is shown in Fig. 1. The main elements of the code are kernels (blue), kernel drivers (orange), global drivers (red) and global data types (green). Each kernel is designed to solve one type of basic problem. A local driver is assigned to each kernel which is able to run it in stand alone. For example, the INSTANT kernel driver is able to read data from an input file into the global kernel data type and run a standalone INSTANT calculation. The global data types hold globally needed information like cross section data, mesh, fluxes, etc. that are needed by more than one kernel to perform more complex calculations involving different kernels. Global drivers (red) coordinate the information flow and call the needed kernels to perform a given type of calculation.

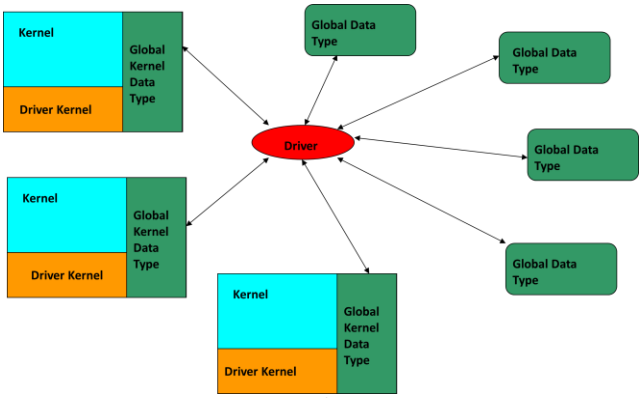

Fig. 1. PHISICS: General software structure.

This structure allows for a high modeling and software flexibility since new drivers can easily be written to combine different kernels needed to solve a particular problem.

The main modules of the code to date are the spherical harmonics transport core solver INSTANT (steady state and transient), the depletion module MRTAU and a MIXER module capable of interpolating and mixing cross sections. These modules are explained in more detail below.

II.A. INSTANT

The transport core solver INSTANT[5] (Intelligent Nodal and Semi-structured Treatment for Advanced Neutron Transport) is the most developed kernel of the PHISICS framework. INSTANT is parallelized and is designed to take full advantage of middle to large clusters (10 to 1000 processors). INSTANT is based on the second order formulation of the transport equation discretized in angle by spherical harmonics while in space it uses horto-normal arbitrary order polynomials[6]. Its key features are:

- 2D/3D Cartesian geometry
- 2D triangular, Z extruded
- 2D hexagonal, Z extruded
- Unlimited number of energy groups
- Unlimited number of up-scattering groups
- Up to P33 anisotropy
- Reflective, vacuum and periodic boundary conditions
- Computation of fundamental mode and source problems forward and adjoint
- Chebyshev acceleration for power iterations
- Diffusion partitioning for inner iterations

In order to be able to solve transient problems, a time dependent scheme has recently been implemented as a new module for the PHISICS suite. The chosen scheme is based on a second order backward Euler scheme with explicit delayed neutron treatment. In the following it is reported a summary of its derivation, while more details could be found in Ref. 7.

As in many time integration schemes also in this case it is possible to rewrite the time dependent transport equations in a form such as that the equations are formally equivalent to the time independent ones. Therefore, time dependent problems could be simulated without changing the steady state INSTANT transport solver. The main elements of this derivation are given below.

The time dependent transport equation coupled to the delayed neutrons equation is given in Eq. (1).

$$\left(\frac{1}{s}\frac{\partial}{\partial t}+\vec{\Omega}\cdot\vec{\nabla}+\Sigma_{\mathrm{T}}(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})\right)\Psi(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t})$$
$$=\int_{E_0}^{E_G}dE'\sum_{n=0}^{N_s}\Sigma_{\mathrm{s}}(\mathrm{E}'\to\mathrm{E},\mathrm{t})\sum_{m=-n}^{n}\Phi_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega})$$
$$+\int_{E_0}^{E_G}dE'\chi_p(\mathrm{E}'\to\mathrm{E})(1-\beta(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t}))(\upsilon\Sigma_{\mathrm{F}})(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})$$
$$+\sum_{\mathrm{f}=1}^{\mathrm{NF}}\chi_f(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})\lambda_f(\vec{\mathrm{r}},\mathrm{t})N_f(\vec{\mathrm{r}},\mathrm{t})+S_{ext}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t})$$
$$\frac{\partial N_f(\vec{\mathrm{r}},\mathrm{t})}{\partial t}=\int_{E_0}^{E_G}dE\beta_f(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})(\upsilon\Sigma_{\mathrm{F}})(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})-\lambda_f(\vec{\mathrm{r}},\mathrm{t})N_f(\vec{\mathrm{r}},\mathrm{t})\qquad f=1,\ldots,NF \tag{1}$$

By defining

$$\left(F_p\Phi_{0,0}\right)(\vec{\mathrm{r}},\mathrm{E},\mathrm{t}) = \int_{E_0}^{E_G} dE'\chi_p(\mathrm{E}' \to \mathrm{E})\left(1 - \beta(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\right)(\upsilon\Sigma_{\mathrm{F}})(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t}) \quad (2)$$

and

$$S_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) = \sum_{\mathrm{f}=1}^{\mathrm{NF}} \chi_f(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})\lambda_f(\vec{\mathrm{r}},\mathrm{t})N_f(\vec{\mathrm{r}},\mathrm{t}) + S_{ext}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) \quad (3)$$

the time dependent transport equation can be rewritten as

$$\left(\frac{1}{s}\frac{\partial}{\partial t} + \vec{\Omega}\cdot\vec{\nabla} + \Sigma_{\mathrm{T}}(\vec{\mathrm{r}},\mathrm{E},\mathrm{t})\right)\Psi(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) = \int_{E_0}^{E_G} dE' \sum_{n=0}^{N_s} \Sigma_{\mathrm{s}}(\mathrm{E}' \to \mathrm{E},\mathrm{t}) \sum_{m=-n}^{n} \Phi_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + \left(F_p\Phi_{0,0}\right)(\vec{\mathrm{r}},\mathrm{E},\mathrm{t}) + S_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}). \quad (4)$$

Introducing the definitions for odd and even functions with respect to the angle

$$f^{\pm}(\vec{\Omega}) = \pm f^{\pm}(-\vec{\Omega})$$
$$f^{\pm}(\vec{\Omega}) = \frac{\left(f(\vec{\Omega}) \pm f(-\vec{\Omega})\right)}{2} \quad (5)$$

for the angular flux, the external and delayed sources and adding and subtracting the equations for +Ω and -Ω (from Eq. 4), the following system of equations can be found:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) + \left(\frac{1}{s}\frac{\partial}{\partial t} + \Sigma_{\mathrm{T}}(\vec{\mathrm{r}},\mathrm{E})\right)\Psi^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) = \int_{E_0}^{E_G} dE' \sum_{n=0,even}^{N_s} \Sigma_{\mathrm{s}}(\mathrm{E}' \to \mathrm{E},\mathrm{t}) \sum_{m=-n}^{n} \Phi_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + \left(F_p\Phi_{0,0}\right)(\vec{\mathrm{r}},\mathrm{E},\mathrm{t}) + S^{+}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t})$$

$$\vec{\Omega}\cdot\vec{\nabla}\Psi^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) + \left(\frac{1}{s}\frac{\partial}{\partial t} + \Sigma_{\mathrm{T}}(\vec{\mathrm{r}},\mathrm{E})\right)\Psi^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) = \int_{E_0}^{E_G} \sum_{n=0,odd}^{N_s} \Sigma_{\mathrm{s}}(\mathrm{E}' \to \mathrm{E}) \sum_{m=-n}^{n} \Phi_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}',\mathrm{t})\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + S^{-}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E},\mathrm{t}) \quad (6)$$

In the above system of equations, it is assumed that the fission emission (prompt and delayed) is isotropic. It therefore appears only in the even equation.

Appling a first order backward Euler scheme in time to the system leads to:

$$\vec{\Omega}\cdot\vec{\nabla}\Psi_{\mathrm{i}}{}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \left(\frac{1}{s\Delta t_i} + \Sigma_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E})\right)\Psi_{\mathrm{i}}{}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = \int_{E_0}^{E_G} dE' \sum_{n=0,even}^{N_s} \Sigma_{\mathrm{s,i}}(\mathrm{E}' \to \mathrm{E}) \sum_{m=-n}^{n} \Phi^{\mathrm{i}}{}_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}')\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + \left(F_p\Phi_{0,0}\right)(\vec{\mathrm{r}},\mathrm{E}) + S^{+,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \frac{1}{s\Delta t_i}\Psi_{\mathrm{i-1}}{}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E})$$

$$\vec{\Omega}\cdot\vec{\nabla}\Psi_{\mathrm{i}}{}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \left(\frac{1}{s\Delta t_i} + \Sigma_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E})\right)\Psi_{\mathrm{i}}{}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = \int_{E_0}^{E_G} dE' \sum_{n=0,odd}^{N_s} \Sigma_{\mathrm{s,i}}(\mathrm{E}' \to \mathrm{E}) \sum_{m=-n}^{n} \Phi^{\mathrm{i}}{}_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}')\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + S^{-,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \frac{1}{s\Delta t_i}\Psi_{\mathrm{i-1}}{}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) \quad (7)$$

were indexes i and i-1 are used to indicate current and previous time step and $\Delta t_i = t_i - t_{i-1}$.

Finally, by posing

$$\tilde{S}^{+,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = S^{+,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \frac{1}{s\Delta t_i}\Psi_{\mathrm{i-1}}{}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) \quad (8)$$

$$\tilde{S}^{-,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = S^{-,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) + \frac{1}{s\Delta t_i}\Psi_{\mathrm{i-1}}{}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) \quad (9)$$

$$\tilde{\Sigma}_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E}) = \Sigma_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E}) + \frac{1}{s\Delta t_i} \quad (10)$$

the system can be rewritten as

$$\vec{\Omega}\cdot\vec{\nabla}\Psi_{\mathrm{i}}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E})+\tilde{\Sigma}_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E})\Psi_{\mathrm{i}}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = \int_{E_0}^{E_G} dE' \sum_{n=0,even}^{N_S} \Sigma_{\mathrm{s,i}}(\mathrm{E}' \rightarrow \mathrm{E}) \sum_{m=-n}^{n} \Phi^{\mathrm{i}}{}_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}')\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + \left(F_p\Phi_{0,0}\right)(\vec{\mathrm{r}},\mathrm{E}) + \tilde{S}^{+,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E})$$

$$\vec{\Omega}\cdot\vec{\nabla}\Psi_{\mathrm{i}}^{+}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E})+\tilde{\Sigma}_{\mathrm{T,i}}(\vec{\mathrm{r}},\mathrm{E})\Psi_{\mathrm{i}}^{-}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) = \int_{E_0}^{E_G} dE' \sum_{n=0,odd}^{N_S} \Sigma_{\mathrm{s,i}}(\mathrm{E}' \rightarrow \mathrm{E}) \sum_{m=-n}^{n} \Phi^{\mathrm{i}}{}_{\mathrm{n,m}}(\vec{\mathrm{r}},\mathrm{E}')\mathrm{Y}_{\mathrm{n,m}}(\vec{\Omega}) + \tilde{S}^{-,i}{}_{ext,D}(\vec{\mathrm{r}},\vec{\Omega},\mathrm{E}) \quad (11)$$

which is formally equivalent to the time independent set of equations that are used for the derivation of the discretization used inside INSTANT[6] . Therefore, the INSTANT solver can be used without modification to solve time problems. The time kernel calculates the time absorption (last term in Eq. 10) and time source (last term in Eq. 9) and adds it to the absorption and external source, respectively, before calling the INSTANT kernel.

In addition, the time kernel also treats the delayed neutrons. It computes the initial conditions at the beginning of the transient and does the time integration for subsequent time steps.

The initial conditions, for a transient, is assumed to be the asymptotic value, thus the one for which the time derivative is set equal to zero. The initial condition is therefore obtained by solving the following system of equations:

$$\int_{E_0}^{E_G} dE'\left(\beta_f\right)(\vec{\mathrm{r}},\mathrm{E})(\upsilon\Sigma_{\mathrm{F}})(\vec{\mathrm{r}},\mathrm{E})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E}) - \lambda_f(\vec{\mathrm{r}})N_f(\vec{\mathrm{r}}) = 0$$

for f = 1, …, NF (12)

The spatial dependency of the cross sections, lambdas and betas can be removed since they are assumed flat in space within a computational cell. With these consideration, the solution for N_f^0 is:

$$N_f^0(\vec{\mathrm{r}}) = \frac{F_0(\vec{\mathrm{r}})}{\lambda_f}$$

$$F_0 = \int_{E_0}^{E_G} dE\,\beta_f(E)(\upsilon\Sigma_{\mathrm{F}})(\mathrm{E})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E}) \quad (13)$$

For the time integration of the precursor densities during the transient, it is assumed that cross sections, lambdas and betas are constant within the computational cell or gauss point. The equation to solve is

$$\frac{\partial N_f(\vec{\mathrm{r}},\mathrm{t})}{\partial t} = \int_{E_0}^{E_G} dE\beta_f(E)\,(\upsilon\Sigma_{\mathrm{F}})(\mathrm{E})\Phi_{0,0}(\vec{\mathrm{r}},\mathrm{E},\mathrm{t}) - \lambda_f N_f(\vec{\mathrm{r}},\mathrm{t})$$

for f = 1, …, NF (14)

Using the definition of F_0 (see Eq. 13) and applying an operator splitting approach were the previous time step flux is used in the computation of the densities, lead to

$$\frac{\partial N_f(\vec{\mathrm{r}},\mathrm{t})}{\partial t} + \lambda_f N_f(\vec{\mathrm{r}},\mathrm{t}) = F_0^{i-1}(\vec{\mathrm{r}}) \qquad t^{i-1} < t < t^i \quad (15)$$

where the solution can be calculated explicitly as

$$N_f^i(\vec{\mathrm{r}}) = N_f^{i-1}(\vec{\mathrm{r}})e^{-\lambda_f\Delta t_i} + \frac{1-e^{-\lambda_f\Delta t_i}}{\lambda_f}F_0^{i-1}(\vec{\mathrm{r}})$$

for f = 1, …, NF (16)

It is worth mentioning that this expression could be unstable for $|\lambda_f| \ll 1$, i.e. for very long living delayed neutron families.

II.B. MRTAU

MRTAU (Multi-Reactor Transmutation Analysis Utility) is a generic depletion/decay/burn-up code developed at INL[8]. The code tracks the time evolution of the isotopic concentration of a given material accounting for nuclear reaction happening in presence of neutron flux and also due to natural decay (Bateman equation). Moreover the code can provide, as additional output, information regarding integral quantities associated with different nuclear reactions like helium production and energy released (pseudo isotopes are used to simulate decay heat). The calculation flow could be controlled in such a way in-core and out-core periods could be alternated freely. The main features of MRTAU are:

- Use of Taylor series expansion based algorithm at arbitrary order and CRAM methodology[9] for computation of the exponential matrix
- Criticality search option, i.e. the isotopic densities are changed in a material to preserve core criticality. Typically, the boron concentration is changed in a region to simulate control rod movement
- Multi transmutation and partitioning loops
- Possibility for the user to deplete only the actinides or actinides and fission products
- Power history specification

- Separation efficiency specification for each isotope

II.C. MIXER

A MIXER module is also part of the PHISICS suite. This module does all the cross section handling for the different kernels. The MIXER can handle macroscopic, microscopic and "mixed" cross sections:

A macroscopic cross section library contains macroscopic cross sections for each type of material used in the calculation tabulated for the state parameters (temperature, burn-up, control rod position, etc.). The mixer just interpolates these cross sections at the requested state parameters. No limits in tabulation dimensions or neutron energy groups exist.

A microscopic or "mixed" cross section library contains the tabulated cross sections for each isotope considered in the calculation. The mixer reads for each material a description containing a list of isotopes and corresponding densities. Microscopic cross sections are interpolated at the requested state parameters and macroscopic cross sections for each material are generated with the corresponding densities. With this capability, "mixed" macroscopic and microscopic cross sections are also possible. For example, it is possible to provide macroscopic absorption cross sections without xenon for a material and the microscopic xenon absorption together with a xenon density. The mixer will calculate the xenon contribution for the absorption and add it to the macroscopic cross section.

Micro, macro or "mixed" cross section libraries can be input in different ways: An easy to use XML structure is available if cross sections are available in text format from any source. In addition, the MIXER reads the AMPX and ISOTXS library formats if cross section libraries are prepared with SCALE, ERANOS[10] or MC2[11]. More library types are planned to be supported in the future.

III. RELAP5-PHISICS coupling

There are two considerations by coupling the PHISICS suite to RELAP5. First, on the RELAP5 side a low impact for the user, i.e. it should be possible to run existing RELAP5 input decks with INSTANT instead of NESTLE. Second, on the PHISICS side, it is desirable to have lean software interdependency, for further maintenance of the coupling. It was decided to couple the different modules of PHISICS directly to RELAP5, i.e. PHISICS is integrated in RELAP5 as a subroutine. Fig. 2 shows the input data flows between the two packages. The calculation is always driven by RELAP5. The RELAP5 input reader decides if parts of PHISICS are needed and calls the RELAP5-PHISICS driver accordingly. The RELAP5-PHISICS driver collects the available input data from RELAP5 (geometry, calculation options) and adds, if needed, data from special PHISICS input files to complete the needed information for the required calculation. Once all the inputs are collected, the RELAP5-PHISICS driver calls the required modules like INSTANT, MRTAU or the MIXER and feeds back the results (power distribution) to RELAP5 for the next iteration.

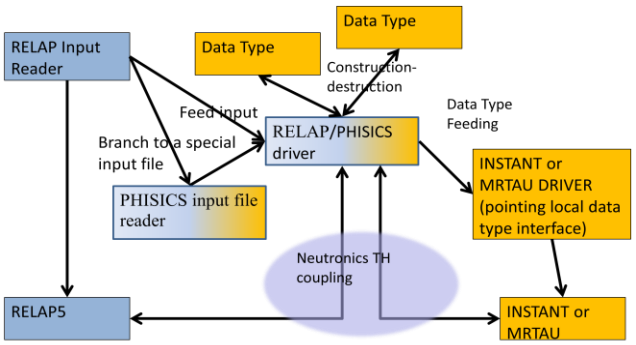

Fig. 2. RELAP5-PHISICS coupling structure.

Different calculation scenarios are possible. For coupled steady state calculations, the INSTANT solver is made accessible from within RELAP5 through a new keyword "INSTANT" extending the existing "POINT" and "NODAL" functions. This new option gives the user access to the full capability of INSTANT from within RELAP5. 2D and 3D problems can be solved using different geometry and mesh types. In addition to the Cartesian and hexagonal geometry, INSTANT supports triangular and triangular extruded meshes. Direct and adjoint fundamental mode problems can be solved using transport theory up to PN33. Furthermore, the INSTANT part of coupled calculations can be parallelized on multiple processors.

The coupling is compatible with the existing RELAP5 cross section and feedback options ("RAMONA", "HWR", "GEN", etc.). Existing cross section inputs and mappings from kinetic nodes to thermal hydraulic zones can be used with the INSTANT option. The RELAP5 control rod model can also be used. Fig 3, shows the coupling scheme for this compatibility mode where the cross section treatment is managed by NESTLE. INSTANT gets the interpolated macroscopic cross sections, computes the new flux distribution in the core and feeds the corresponding power distribution back to RELAP5 for the next iteration.

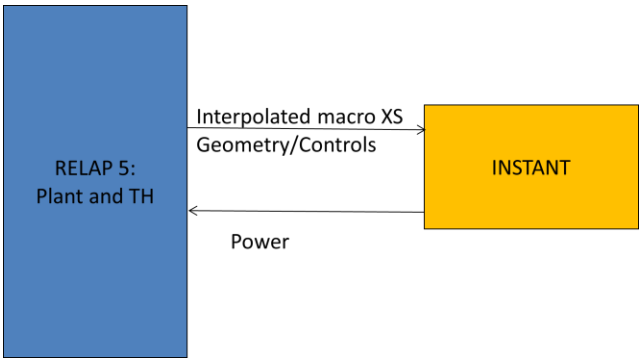

Fig. 3. RELAP5 treats cross sections, core solver replaced by INSTANT

In addition, the coupling supports a new cross section option "PHISICS" which lifts the RELAP5 limitation of four energy groups. The number of neutron energy groups for "PHISICS" is only limited by the available computer memory.

The multi group cross sections for "PHISICS" are input in any format supported by the mixer (see above). Different options such as diffusion coefficients with absorption cross sections or total cross sections with the full scattering matrices are supported by the "PHISICS" option. Mappings from kinetic nodes to thermal hydraulic zones and regions (to define feedback correspondences between the neutronic and thermal hydraulic mesh) for the "PHISICS" option are still input in the RELAP5 input file in the "GEN" form. In this case (shown in Fig. 4), RELAP5 sends the state parameters, like fuel/coolant temperatures and densities and control rod positions to the MIXER, which interpolates and mixes (if microscopic cross sections and material descriptions are provided) the multi-group cross sections. The final interpolated macroscopic cross sections are then sent to INSTANT, which calculates the new flux distribution. The resulting power distribution is sent back to RELAP5.

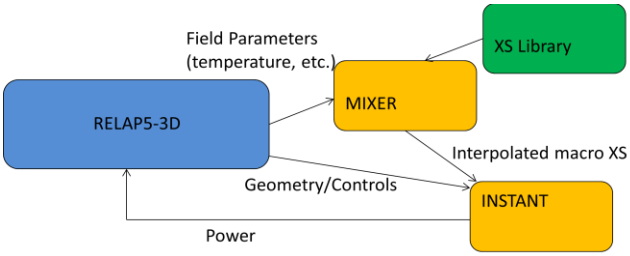

Fig. 4. RELAP5 treats only TH, core solver and cross section manipulation replaced by PHISICS

These two approaches, i.e. cross section manipulation by NESTLE or by the MIXER are also available for time dependent problems (shown in Fig. 5). In this case, the interpolated macroscopic cross section are sent to the time driver either from NESTLE or from the MIXER. The time step is sent from RELAP5 to the time driver. The time kernel evaluates the new delayed neutron precursor densities and calculates the "time absorption" and "time source" as in Section II.A. The manipulated cross sections and the source are then sent to INSTANT, which evaluates the new flux and power distribution in the core. As for steady states, the new power distribution at $t+\Delta t$ is sent back to RELAP5 for the next time step.

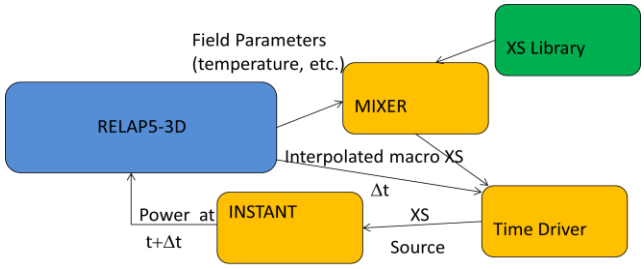

Fig. 5. Time dependent coupling scheme (both options shown, cross section from RELAP5 or from the MIXER)

The burn-up code MRTAU has also been coupled to RELAP5. The possible calculation paths resulting by combining INSTANT, MRTAU and RELAP5 are manifold. The current implementation of the coupling allows for the calculation types detailed in the following section.

Coupled steady-states can be calculated for a given burn-up level. As shown in Fig. 6, MRTAU, the MIXER and INSTANT iterate to advance the core burn-up to a desired level. Optionally, it is possible to use the criticality search module (dotted lines in Fig. 6), which adjusts the control rod position (or any desired isotopic density like the boron in the coolant) to keep the core critical during this pre-burning. The pre-burning is done with constant temperature field, i.e. the MIXER evaluates the cross sections at the initial temperature for each burn-up step. Once the desired burn-up is reached (t>T burn), a coupled thermal-hydraulic/neutronic steady state can be calculated to obtain the plant conditions (temperatures, pressures, etc.) at the desired core burn-up.

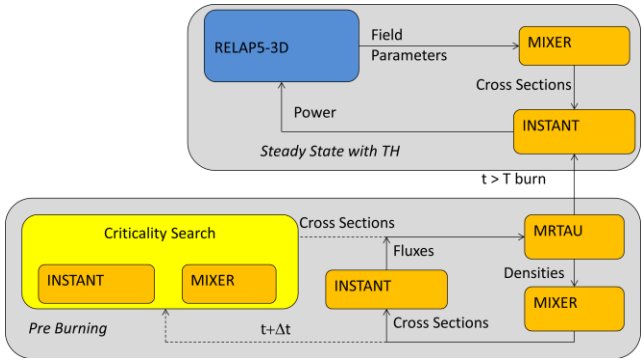

Fig. 6. Steady-state search with pre-burning at constant temperature field.

This scheme can also be iterated, to compute the core burn-up with an updated temperature field for each burn-up step. As shown in Fig. 7, a new core temperature field (Steady State Search) is calculated for every burn-up step with MRTAU.

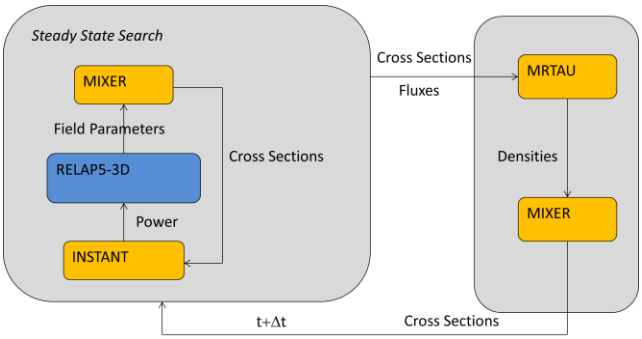

Fig. 7. Burn-up calculation with temperature feed-back

MRTAU can also be used during transients, to track poisons like xenon. As mentioned in the MIXER description, it is possible to use a mix of macroscopic and microscopic cross sections. This allows inputting macroscopic cross sections for each material together with microscopic xenon absorption and initial xenon density. A possible transient calculation scenario is shown in Fig. 8: the core could be pre-burned to a desired burn-up level, before a transient is started where xenon is traced with MRTAU. It is possible to use the critically search module as an option (dotted lines in Fig. 8) in this scenario.

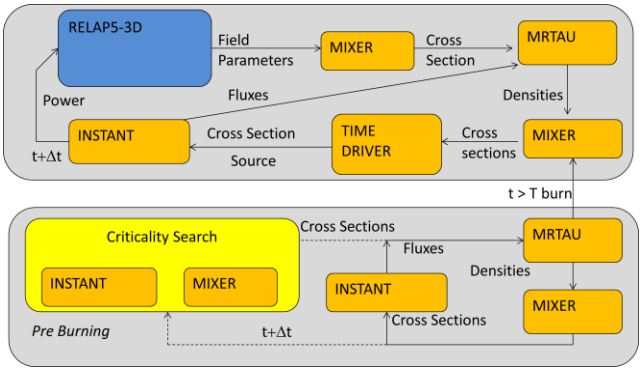

Fig. 8. Transient with poison tracking and pre-burning

IV. Illustrative calculations

In order to demonstrate the correct implementation of the RELAP5-PHISICS coupling, as well as to illustrate some of its capabilities, calculations have been performed on some simple cases:

- Steady state
  - Cartesian geometry
    - PWR with control rod
    - PWR without control rod
  - Hexagonal geometry
    - Takeda 4 benchmark
- Transient
  - Control rod movement in simplified PWR
-

IV.A. Steady state

To test the steady state RELAP5-PHISICS coupling, a Cartesian and a Hexagonal geometry case have been calculated.

Cartesian geometry

The Cartesian geometry case is a typical PWR. The full 3D core has been modeled (17x17x13 nodes (x/y/z)). The core contains 11 different materials (different colors in Fig. 9). 36 different feedback zones have been considered to map the thermal-hydraulic mesh to the neutronic mesh. Two neutron energy groups have been used. Two cases, one with and one without control rod (see Fig. 9) have been calculated. The case with control rod contains one control rod in the core center.

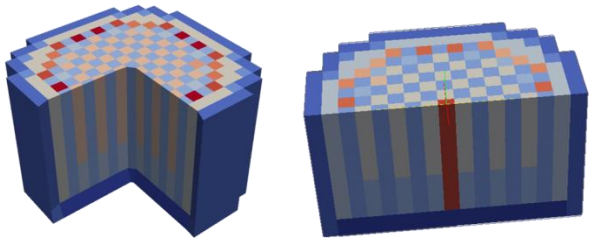

Fig. 9. PWR material distribution. Left: without control rod; Right: including control rod (1 rod in core center (red))

For the case without control rod, k_{eff} and power distributions have been calculated with RELAP5-NESTLE and RELAP5-PHISICS. Table II compares the k_{eff} found with RELAP5-NESTLE and RELAP5-PHISICS for the first iteration, i.e. when NESTLE and INSTANT use the same cross sections evaluated from the initial temperature distribution (Initial), and the converged solution, i.e. when the iterations between thermal-hydraulics and neutronics are converged (Converged). It goes without saying that for the converged k_{eff}, NESTLE and INSTANT use different cross sections according to the evolution of the temperature during the steady state search. INSTANT used PN=1 with surface order 1 and volume order 4, whereas NESTLE uses the nodal expansion method.

TABLE II

RELAP5-NESTLE and RELAP5-PHISICS k_{eff} for the PWR without control rod.

	k_{eff}	
	Initial	Converged
INSTANT	1.01639	1.00348
NESTLE	1.01731	1.00483
Delta	*0.00092*	*0.00135*

As one can see from Table II, NESTLE and INSTANT are within 100pcm if they use the same cross sections (initial iteration), and are still within 135pcm after convergence with the thermal-hydraulics (after ~600 iterations). Fig. 10 shows the relative difference in power distribution between RELAP5-NESTLE and RELAP5-PHISICS for the first iteration and the converged solution. As one can see, the power distribution for the first iteration is about 10% different at the border of the core between NESTLE and INSTANT. For the converged solution, this difference reduces to ~8%, i.e. the thermal-hydraulic feedbacks tend to reduce the difference in power distribution.

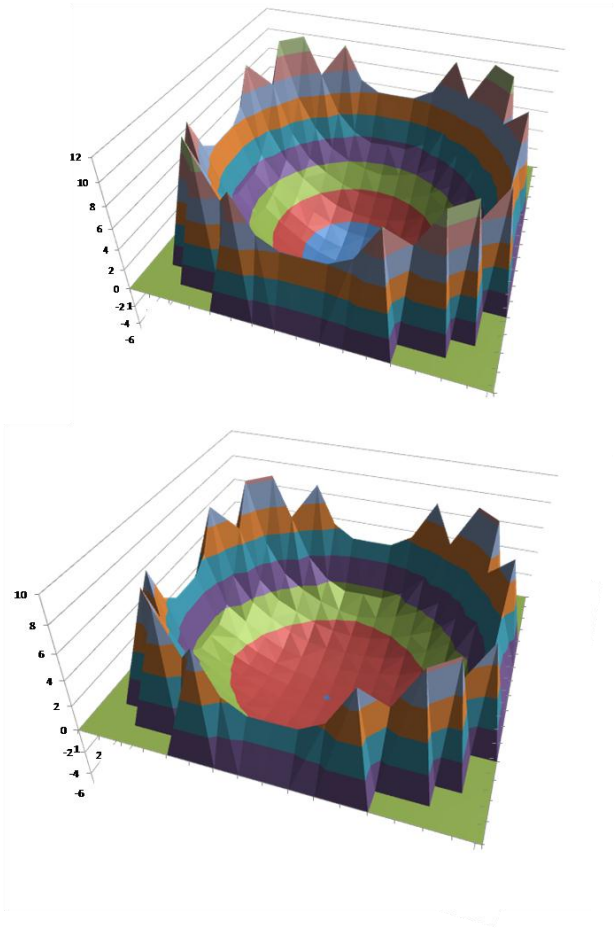

Fig. 10. PWR without control rod: power distribution difference (%) at mid core height. Top: first iteration; Bottom: converged

Table III, shows the converged k_{eff}, for the same exercise where the control rod is inserted.

TABLE III

RELAP5-NESTLE and RELAP5-PHISICS k_{eff} for the PWR including control rod.

	k_{eff}
	Converged
INSTANT	1.03838
NESTLE	1.03919
Delta	*0.00081*

The comparison of RELAP5-NESTLE and RELAP5-PHISICS for steady state in Cartesian geometry shows good agreement for the k_{eff}. The difference in power distribution is higher than expected. Possible reasons for this difference could be the higher spatial resolution in INSTANT. Also different implementations of the vacuum boundary condition in NESTLE and INSTANT could contribute to the difference. Table IV shows the converged RELAP5-PHISICS k_{eff} for different surface and volume orders. It can be seen that INSTANT converges spatially.

TABLE IV

RELAP5-PHISICS k_{eff} for the PWR without control rod: spatial convergence.

Surface order	Volume order			
	3	4	5	6
0	1.00407	1.00440		
1		1.00348	1.00368	
2			1.00361	1.00372
3				1.00371

Excursion: Computational time

It is interesting to compare computational times for RELAP5-NESTLE and RELAP5-PHISICS. Using RELAP5-PHISICS with 4th order spatial approximation increases the computational time by a factor of ~80 for 1500 steady state iterations compared to RELAP5-NESTLE. This factor is significant even by considering the gain in accuracy by using INSTANT. First, INSTANT is a very young code and several optimizations are expected in the future concerning accelerations and second, the computational time using RELAP5-PHISICS can already now be reduced by:

1. Use of a cross section threshold.
2. Parallelization of INSTANT

Using a cross section threshold means that INSTANT is only called if the change in cross sections from the thermal-hydraulic feedbacks exceeds a given value. This option is implemented in the coupling. For the PWR without control rod, a factor of ~10 can be gained in computational time with RELAP5-PHISICS.

The gain in speed using multiple processors depends on how many cores are available. The time scaling for INSTANT is almost perfect for cores on shared memory; otherwise it depends on the node to node communication speed. A gain of a factor of 10 is possible by average users.

By using both, the cross section threshold and parallelization, the initial factor of ~80 can be compensated. The user can choose a tradeoff between accuracy and computational cost.

It is worth mentioning that not only INSTANT is parallelized, but also the MIXER and MRTAU.

Hexagonal geometry

To test the coupling for hexagonal geometry, the Takeda 4 benchmarkREF has been considered. Takeda 4 is a numerical benchmark with a small sodium cooled fast reactor using 4 energy groups. A case without control rods has been calculated with NESTLE and INSTANT and has been compared to the UNIC Diffusion code[12]. The full core in 3D has been modeled. No thermal-hydraulic feedbacks have been considered. Fig 11 shows a section of the core without control rod, where blue is , red is , green is ….

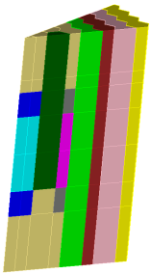

Fig. 11. Takeda 4 hexagonal geometry benchmark without control rod.

Table V shows a spatial convergence study using INSTANT with PN=1. Table VI compares the spatially converged INSTANT k_{eff} to the k_{eff} computed with NESTLE (nodal expansion method) and the UNIC code used in diffusion approximation.

TABLE V

RELAP5-PHISICS k_{eff} for the Takeda 4 without control rod: spatial convergence.

Surface order	Volume order			
	5	6	7	8
0	1.07995	1.07995	1.07995	
1		1.07350	1.07351	
2			1.07343	1.07344
3				1.07342

TABLE VI

INSTANT, NESTLE and UNIC k_{eff} for the Takeda 4 without control rod

	k_{eff}
INSTANT (S3V8)	1.07342
NESTLE	1.07427
UNIC	1.07335

From the above tables, it can be seen that the INSTANT PN1 solution converges towards the UNIC solution. INSTANT and the NESTLE nodal expansion method are within ~90pcm.

IV.A. Transient

To check the RELAP5-PHISICS coupling for transients, a control rod movement has been compared between RELAP5-NESTLE and RELAP5-PHISICS. The core is a simplified PWR. Two group cross sections have been used including temperature feedbacks. The transient has been run for 300 seconds. A control rod is kept at its initial inserted position for the first 100 seconds then withdrawn and reinserted. Fig. 12 shows the control rod position (red) in the right axis. Furthermore, the figure shows the core power computed with RELAP5-NESTLE and RELAP5-PHISICS on the left axis.

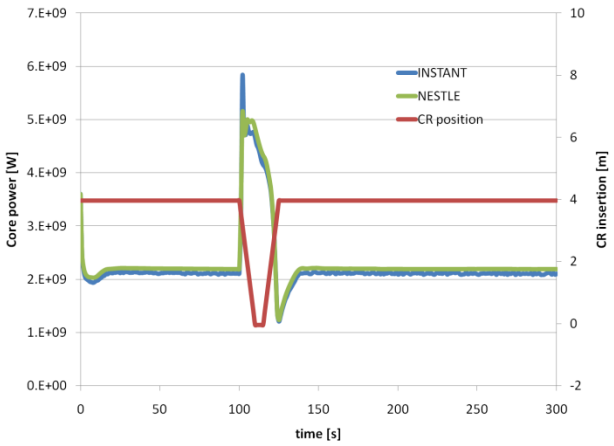

Fig. 12. Control rod movement transient with RELAP5-NESTLE and RELAP5-PHISICS

It can be seen that the power stays constant as expected if the control rod does not move. The power starts increasing when the control rod is removed. Also as expected, the power decreases, undershoots and stabilizes

at the same power level than initially when the control rod is reinserted (due to delayed neutrons and thermal-hydraulic feedbacks). Furthermore, it can be seen that RELAP5-NESTLE and RELAP5-PHISICS are in good agreement.

V. CONCLUSIONS

This paper presents the status of the coupling of PHISICS to RELAP5-3D. First, the different modules of the PHISICS code package have been described. It has been shown that PHISICS is highly modular, which makes it easy to develop new capabilities and maintain existing code. PHISICS has been coupled to the thermal-hydraulic code REALP5-3D for steady state and transient calculations. The coupling is compatible with existing RELAP5 input decks. This coupling adds the following features to the RELAP5 neutronic capabilities:

- Transport calculation up to P33
- Spatial and angular mesh refinement
- Unlimited number of neutron energy groups
- Cross section tabulation
- Microscopic cross section handling

The simple problems presented for steady state and transients show that the coupling is properly implemented, i.e. geometry, controls and cross sections are handled correctly and power is correctly fed back to RELAP5. The results obtained for coupled calculations using RELAP5-PHISICS are in agreement with RELAP5-NESTLE and UNIC calculations if INSTANT uses P1 to approximate the diffusion solution.

For burn-up calculations with temperature feedback, poison tracking and decay heat calculation during transients, MRTAU has been coupled to RELAP5. It has been shown that calculation possibilities with INSTANT, the MIXER, MRTAU and RELAP5 are manifold. Example calculations involving MRTAU will be presented in a subsequent paper.

It is worth mentioning that the coupling benefits directly from further developments made in the PHISICS framework. The mentioned coupling capabilities are currently further tested and validated.

ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, under DOE Idaho Operations Office Contract DE-AC07-05ID14517. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

REFERENCES

1. The RELAP5 Development Team, *RELAP5/MOD3 Code Manual, Volumes 1 and 2,* NUREG/CR-5535, INEL-95/0174, Idaho National Engineering Laboratory, August 1995

2. P. J. Turinsky et al., *NESTLE: A Few-Group Neutron Diffusion Equation Solver Utilizing the Nodal Expansion Method for Eigenvalue, Adjoint, Fixed-Source Steady State and Transient Problems*, EGG-NRE-11406, Idaho National Engineering Laboratory, June 1994.

3. G. Strydom, A. S. Epiney, *PHISICS/RELAP5-3D Results for Phase I of the OECD/NEA MHTGR-350 MW Benchmark*, Proceedings of ICAPP'12, Chicago, USA (2012) (Paper 12290)

4. C. Rabiti, Y. Wang, G. Palmiotti, H. Hiruta, J. Cogliati, A. Alfonsi, *PHISICS: a New Reactor Physics Analysis Toolkit*, Proc. 2011 ANS Annual Meeting, Hollywood, Florida, USA, June 26–30, 2011, American Nuclear Society (2011)

5. C. Rabiti, Y. Wang, G. Palmiotti, H. Hiruta, J. Cogliati, A. Alfonsi, A. Epiney, T. Grimmet, *PHISICS: New Features and Advancements*, Proc 2011 ANS winter meeting, 2011, American Nuclear Society (2011)

6. E. Lewis, C. B. Carrico and G. Palmiotti, *Variational Nodal Formulation for the Spherical Harmonics Equations*, Nucl. Sci. Eng., **122**, (1996)

7. C. Rabiti, A. Rineiski, *Extension of KIN3D, a Kinetics Capability of VARIANT, for Modeling Fast Transients in Accelerator Driven System*, Proceedings of PHYSOR 2004, Chicago, Illinois, USA 2004

8. S. Bays, S. Piet, M. Pope, G. Youinou, A. Dumontier, D. Hawn, *Transmutation Dynamics: Impacts of Multi-Recycling on Fuel Cycle Performances*, INL/EXT-09-16857 (2009)

9. A. Alfonsi, C. Rabiti, A. S. Epiney, Y. Wang, J. Cogliati, *PHISICS TOOLKIT: MULTI-REACTOR TRANSMUTATION ANALYSIS UTILITY – MRTAU*, Proceedings of PHYSOR 2012, Knoxville, Tennessee, USA 2012

10. Rimpault, G. et al., *The ERANOS data and code system for fast reactor neutronic analyses*. Proceedings of PHYSOR 2002, Seoul, Korea 2002

11. H. Henryson, II, B. J. Toppel, C. G. Stenberg, *MC^{2}2: A Code to Calculate Fast Neutron Spectra and Multigroup Cross Sections*, Argonne-8144, June 1976

12. M. A. Smith, C. Rabiti, D. Kaushik, W. S. Yang, G. Palmiotti, *Report on Advanced Neutronics Code Development in FY2007*, ANL-AFCI-209, 2007