Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications (M&C + SNA 2007)
Monterey, California, April 15-19, 2007, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2007)

THE ARTEMIS CORE SIMULATOR: A CENTRAL COMPONENT IN AREVA NP's CODE CONVERGENCE PROJECT

Andreas Pautz*, Hans-Wilhelm Bolloni, Karl-Albert Breith, Rene van Geemert, Jochen Heinecke, Greg Hobson, Stephan Merk, Baptiste Pothet, and Florin Curca-Tivig
AREVA NP - An AREVA and Siemens Company

ABSTRACT

The reactor core simulator ARTEMIS is a central component of AREVA NP's CONVERGENCE project, which aims for the development of a new, unified neutronics/thermal-hydraulics code system for world-wide usage in all AREVA NP geographical regions. The ARTEMIS code is partly based on well-established and mature modules from current production code systems, but also contains numerous completely newly developed components, for responding on the customer's needs of improved accuracy and performance. The paper gives an overview of the most prominent computational modules, lists some first results obtained with the core simulator, and discusses several new software approaches, which have been applied throughout the project.

Key Words: Core Simulator, Coupled Code Systems, Core Analysis and Design

1. INTRODUCTION

A couple of years ago, AREVA NP decided to launch the development of a next-generation coupled neutronics/thermal-hydraulics code system, in order to keep pace with the latest trends in neutronics and also to meet the customer's increasing demands in terms of accuracy and performance. The new code system ARCADIA™, contains as its central components the spectral code APOLLO2-A (based on APOLLO2 from CEA [1] and adapted by AREVA NP for its industrial needs), as well as the reactor core simulator ARTEMIS, which has been entirely developed in-house.

Historically, the need for developing such a new, "converged" code system became increasingly necessary due to the unsatisfactory situation of having to maintain several code systems at different locations world-wide, and to account appropriately for the regionally strongly varying core design procedures and licensing processes. The present situation inside AREVA NP is depicted in Fig. 1 on the left hand side, where the currently used spectral codes and reactor core simulators of the three geographical regions are listed, namely: the SCIENCE system for Pressurized Water Reactors (PWR) in the French region, the CASCADE system, which is in use in Germany and the United States for PWR, the CASMO/NEMO system for American PWR, and CASMO/MICROBURN in general for all AREVA NP international activities on Boiling Water Reactors (BWR).

* Corresponding author: Andreas Pautz, AREVA NP GmbH, Freyeslebenstraße 1, 91058 Erlangen (Germany). E-mail contact: Andreas.Pautz@areva.com, Tel.: +49-(0)9131-18-97580

The ultimate goal of the CONVERGENCE project is depicted on the right hand side of Fig. 1: for PWR, the spectral code APOLLO2-A and the core simulator ARTEMIS form the computational core of the new code system, which is in turn embedded inside a shell of so called "peripheral services", acting as the only visible interface to the end-user; for BWR it was decided to temporarily keep the MICROBURN-B2 code as the reactor core simulator (however, with the clear option to replace it by ARTEMIS at a later stage), while the currently used spectral codes will be replaced by APOLLO2-A for all regions of AREVA NP. The paper at hand mainly concentrates on the features of the new reactor core simulator ARTEMIS, both with respect to state-of-the-art physical and numerical modeling aspects as well as to new software engineering strategies and methodologies. Chapter 2 will give an overview of some of the central computational components of ARTEMIS (e.g. the neutronics and thermal-hydraulics flux solvers, cross section and depletion module etc.), while chapter 3 presents some first results based e.g. on benchmark calculations and code-to-code comparisons. Finally, Chapter 4 introduces some of the new software strategies and approaches being rigorously applied throughout the project.

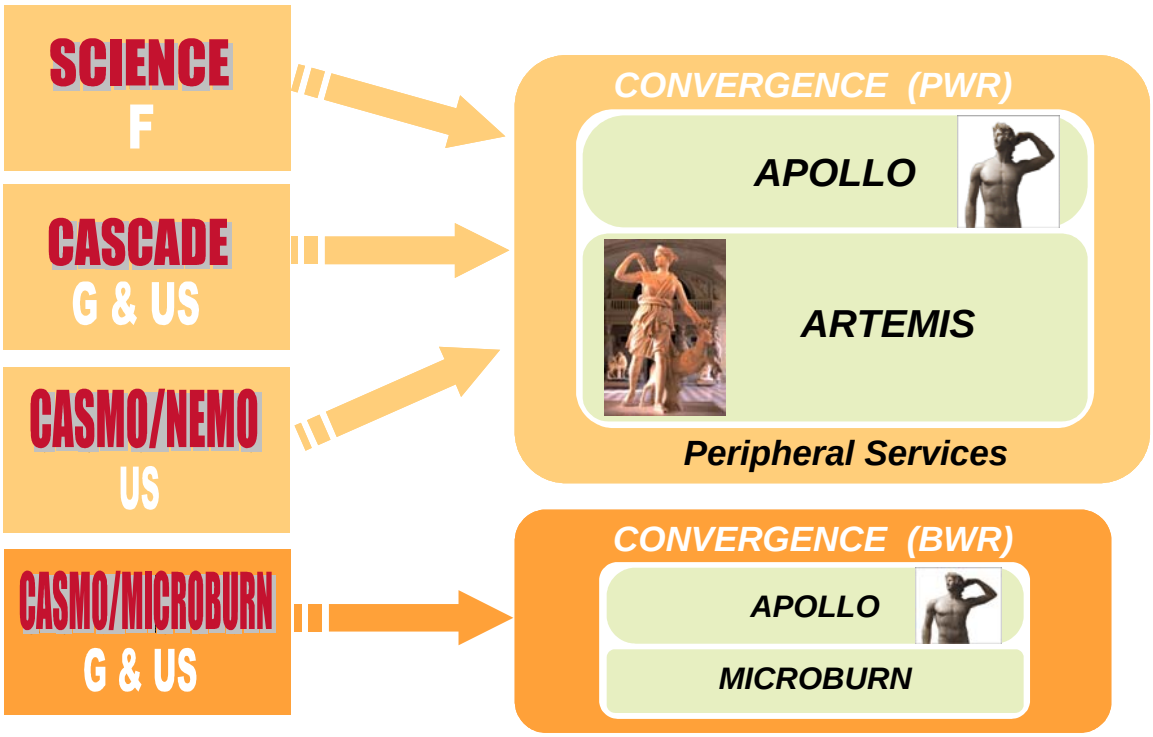

Figure 1: Overview of the currently used code systems within the different AREVA NP regions (left hand side) and the goal of the CONVERGENCE project on the right.

2. THE ARTEMIS CORE SIMULATOR

The ARTEMIS core simulator represents AREVA NP's future 3D coupled neutronics/thermal-hydraulics code system for both steady-state and transient core analysis and design, that will

allow not only for standard 2-group nodal diffusion calculations, but as well for more advanced, pin-by-pin multi-group transport calculations. ARTEMIS has originally been designed for PWR applications mainly, but care has throughout been taken during the early software design stage to include BWR features as well.

The computational kernel of the new code system is essentially formed by the following six major modules:

- The flux solver module, which solves the neutronics field equations either in diffusion approximation of via the Simplified P_N-approach (SP_N). It makes use of the by now classical Nodal Expansion Method (NEM) and features several sophisticated acceleration algorithms, e.g. Multi-Level Coarse-Mesh-Rebalancing (CMR)
- The thermal-hydraulics module is based on the well-known COBRA3-CP code [2]. In a joint effort of all AREVA NP regions, this code has been recoded and enriched by numerous additional options, to address the needs from user communities using other TH-codes, like e.g. FLICA-III [5] or LYNXT. The thermal-hydraulics module can also be driven in stand-alone mode and has recently been baptized COBRA-FLX.
- The cross section module provides macroscopic cross sections to the flux solver, as well as microscopic cross sections for solving the depletion equations. This module is an entirely new development, and uses a much more advanced cross section representation than older, traditional production codes. Instead of using tables or low-order polynomials to describe the dependency of cross sections on thermal-hydraulics state parameters, boron and xenon particle number density, it uses a generalized B-spline approach (with splines of arbitrary order) to obtain maximum accuracy.
- The depletion solver has also been completely developed from scratch, in order to guarantee maximum flexibility with respect to the choice of the nuclide chain, accurate modeling of "feedback" reactions like α-decay of n->2n, and efficient solution methodologies. In doing so, several different solvers have been implemented in parallel, featuring either the "classical" matrix exponential approach as in ORIGEN [8], an extension of this approach making heavy usage of Krylov subspace ideas, or a Runge-Kutta approach to address more complex cross section behavior, e.g. parabolic shapes of Gadolinium cross sections.
- The fuel rod module replaces the original, rather old-fashioned fuel rod component of the COBRA code, and relieves several of its restrictions. It allows for example for arbitrarily complex thermal conductivity and heat capacity laws, different porosity correlations, and makes use of gap conductance tables to describe the heat transfer from fuel to cladding. Large effort was spent to bring the physical modeling of the fuel rod as close as possible to AREVA NP's principal fuel rod design code COPERNIC. In fact, the excellent agreement between both codes has been impressively demonstrated in a large test matrix for a broad range of applications.
- Finally, the dehomogenization module has also been completely developed from scratch, using best-practice approaches from all of AREVA NP's geographical regions. It calculates the 3D-pin-power distribution for each assembly by means of given discontinuity factors and form functions and is fully multi-group capable. It allows the evaluation of fully integrated 3D pin burnup using pin powers and pin segment weighting, as well as tracking of different fuel pin types and detector response calculations.

Besides these major modules, several auxiliary codes are also part of the ARTEMIS code system. The construction of the spline representation of cross sections from spectral code data is e.g. accomplished via the HERMES code, which has to be run prior to ARTEMIS calculations to build cross section libraries in appropriate format. These libraries store the spline coefficients necessary to construct cross sections for arbitrary nodal state parameters. Another example is the POSEIDON code, which is the main module for handling the processes of in-core shuffling, core re-loading and fuel disposal. It stores the current status of fuel assemblies in the core as well as in wet and dry storage on an always-up-to-date fuel repository file; moreover it is capable of handling fuel assembly rotations, fuel assembly repair actions and the geometrical re-meshing of assemblies.

The next sections will describe some of the computational and functional modules of ARTEMIS in more detail, highlighting some of the major new features and pointing out, wherever applicable, the state-of the-art physical and numerical modeling inside the new core simulator.

2.1 The ARTEMIS flux solver module

The ARTEMIS steady-state and transient flux solver modules enable accurate three-dimensional steady-state and transient flux computations within adequate time frames in an optionally sequential or parallel computing environment. Their neutron physics models are characterized by a nodal diffusion approach based on the nodal expansion methodology (NEM) in both polynomial and semi-analytical variants. To meet future trends, a nodal SP_N (Simplified P_N method) transport theory option has also been implemented very recently.

The flux solver module is a central part of the well-known PANBOX and PRISM codes [3,4], originally developed at Siemens KWU for PWR, and has been extensively used for more than 25 years by user communities worldwide. It can therefore be regarded as one of the most mature components of ARTEMIS. To enable modular usage of the flux solver in the ARTEMIS framework, it was extracted from the original code, updated for full Fortran95 compatibility and equipped with a completely new, unified Fortran95 interface for standardized communication with other ARTEMIS modules.

From the methodological point of view, the real strength of the flux solver lies in its sophisticated acceleration mechanisms and its compact data organization. It uses a priori a red-black ordering of the computational nodes for optimized interface current handling and enhanced efficiency of the basic iterative procedure. Moreover, such an ordering scheme allows for easy parallelization, since the iterations for the “red” nodes is essentially decoupled from the black nodes and can therefore be performed in parallel. For doing so, the flux solver possesses a multi-thread based, efficient parallel version of the inner iteration loops, which will gradually be changed towards a modern OpenMP implementation.

The principal acceleration method is based on multi-level coarse mesh rebalancing (ML-CMR), augmented by an occasional additional convergence boost by means of an asymptotic extrapolation procedure. The specific need for such elaborate acceleration schemes stems from

the fact, that unaccelerated iterations tend to show very unfavorable error decay ratios for typical LWR applications (typically 0.98 or larger), which would result in impractical computing times for realistic core sizes. The CMR algorithm is an efficient way to damp higher solution modes of the eigenvalue problem and to give rise to faster convergence towards the basic mode. It essentially consists of a restriction of the complete problem to a set of problems of lower dimension (in both energy and space) and to derive driving factors from the coarse mesh solution to rescale the NEM fluxes by multiplicative correction. Since these re-balancing equations typically have much lower dimension than the full NEM equations, not only a single, but a whole hierarchy of different restrictions of the original problem may be used, resulting in so called V-cycle with subsequent "smoothing" or prolongation towards the overall fine solution. By doing so, error decay ratios of 0.9 or smaller can readily be achieved. The principle of NEM combined with multi-level CMR acceleration is shown in Fig. 2 below.

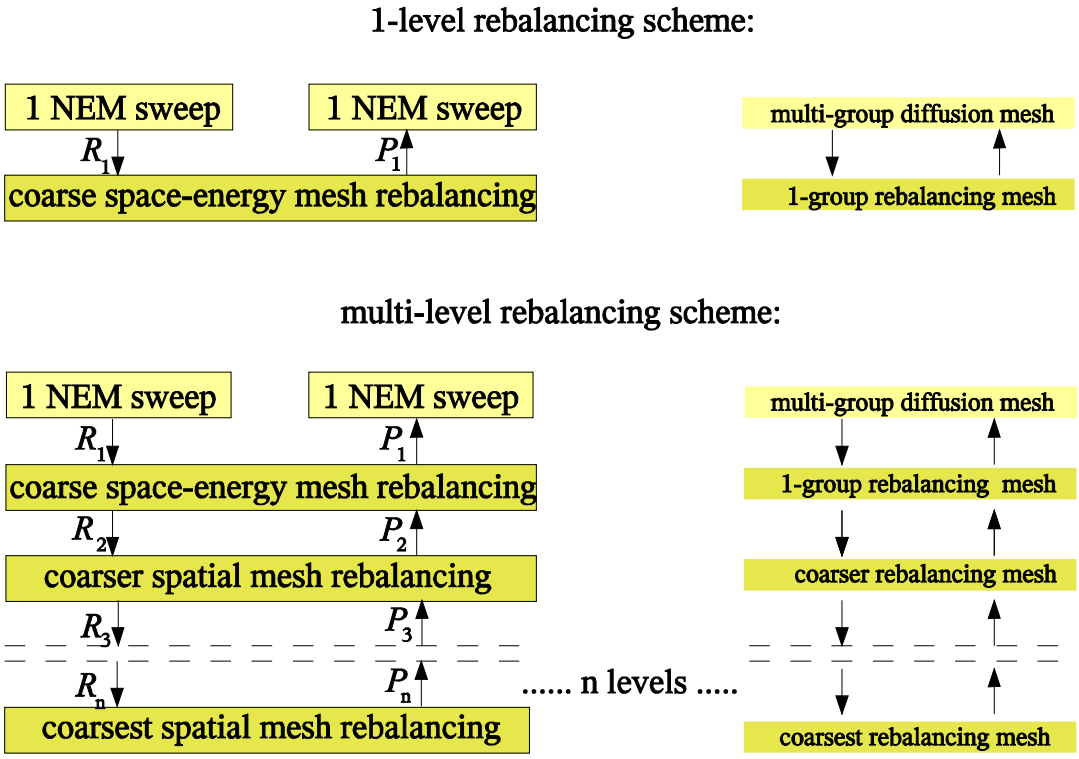

Figure 2: The principle of Multi-Level Coarse Mesh Rebalancing, as it is used in the ARTEMIS flux solver module.

While this scheme has been proven to work extremely successful for a large variety of problems, recently investigated Krylov subspace methods have a strong potential to yield even more efficient acceleration than the classical CMR. Such methods are currently under investigation and are expected to give further speedup by a factor of two and beyond [17].

Most recently, a simplified transport model was implemented into the flux solver module. The SP_N-method (especially SP_3) has become the method of choice in many contemporary LWR core simulators, mainly due to its simple theoretical formulation, its computational efficiency and its strong similarity to the diffusion equation. Especially the latter is the reason, why SP_N can readily be implemented into an already existing diffusion code structure. In ARTEMIS, major effort was spent to make the SP_N formulation compliant with the nodal expansion model and to fit it into the elaborated acceleration schemes, i.e. it was deemed mandatory that CMR must well collaborate with the SP_N-solver. That this is indeed the case, is shown in Fig. 3, where the error decay on the fast flux is plotted vs. the number of NEM-sweeps for an SP_3-calculation (applied to the IAEA-3D core benchmark problem [6]). It is clearly visible, that the CMR has indeed a strong impact on convergence behavior, when compared to the unaccelerated iteration history.

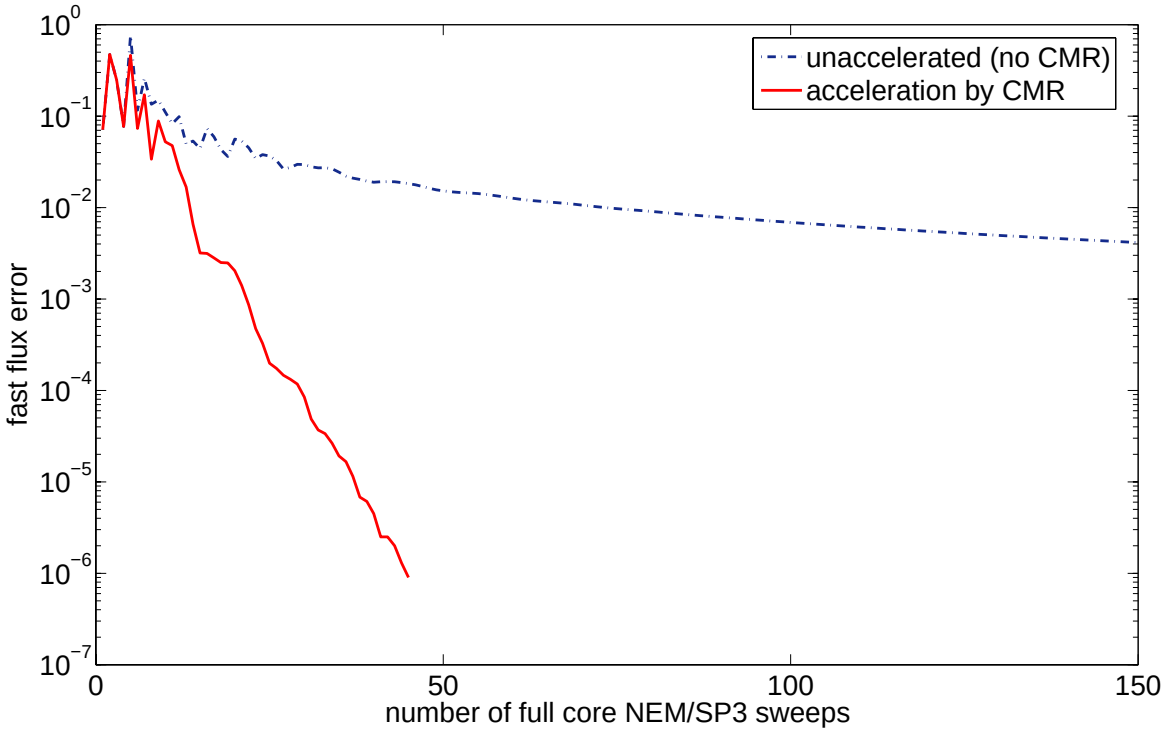

Figure 3: The efficiency of Coarse Mesh Rebalancing for an SP_3-calculation, applied to the IAEA-3D core benchmark problem.

2.2 The ARTEMIS thermal-hydraulics module

The thermal-hydraulics module is primarily based on the well-known code COBRA3-CP, which has been extensively used as production code at AREVA NP. However, it was a major requirement in the frame of the CONVERGENCE project, that also features of the other production codes FLICA-III and LYNXT (like NRC-approved cross flow resistance or turbulent mixing models), had to enter into the new, globally converged code system. This goal was widely achieved, and a large set of benchmark calculations was defined to verify that results from the new code system are typically in good to excellent agreement with those from former production codes. The new, "converged" thermal-hydraulics code has recently been baptized COBRA-FLX. Besides its stand-alone capability, this new code can also be coupled as a module to the ARTEMIS code; hence the new core simulator benefits from all extensions and improvements which have been added to the TH code system.

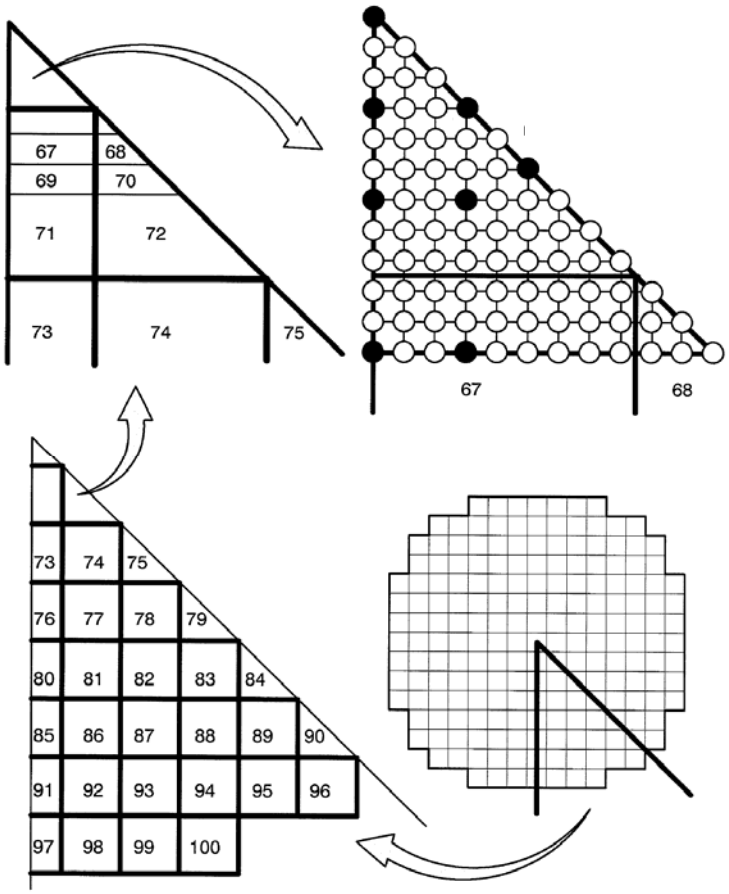

Figure 4: The meshing for an octant of a core with different degrees of spatial resolution.

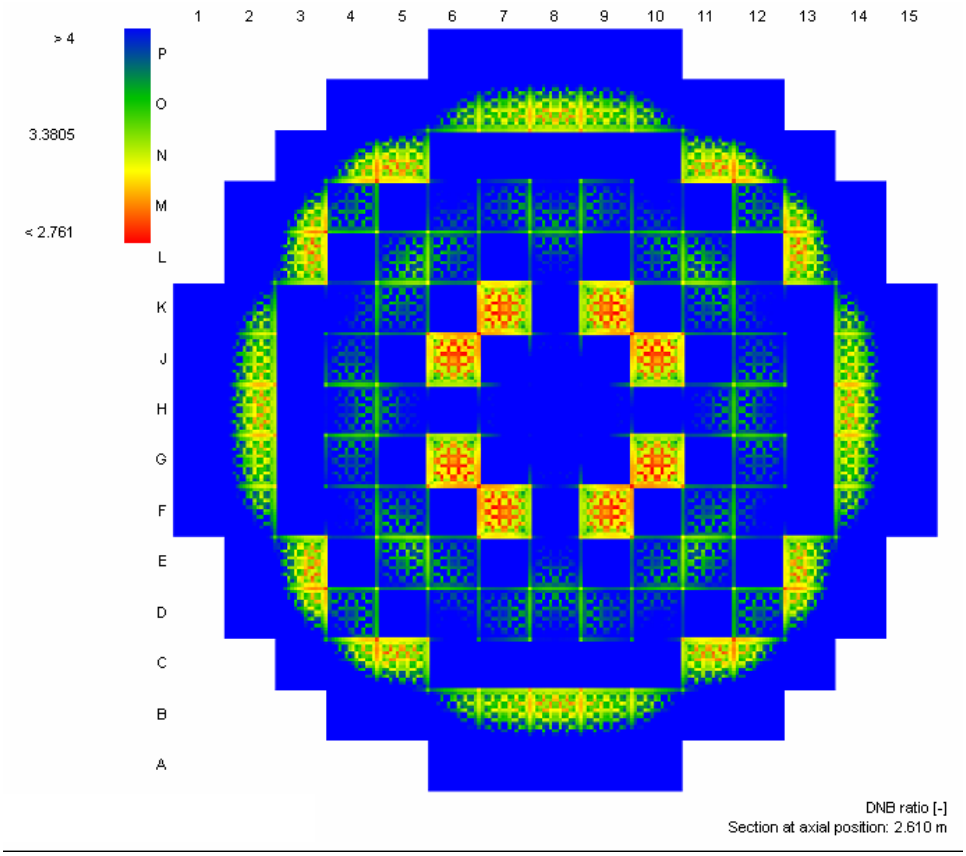

Figure 5: Visualization of thermal-hydraulic results (here: DNB-ratio) with the tool COREVIEW-3D.

The COBRA-FLX code solves the mass, energy and momentum conservation equations, which are expressed as 4-equation system on an integrated sub-channel basis. A quasi three-dimensional modeling is possible via cross flow and pressure iterations between channels. This allows in principle for a full-core pin-by-pin modeling of the complete reactor core. Moreover, it is possible to model a reactor core with different levels of detail: it may e.g. be desirable to consider only a small section of a core in pin-by-pin resolution, while other parts may be sufficiently accurately modeled on a per-assembly basis. For that reason, a graphical input processor was developed, which allows to define, via simple “drag-and-drop” operations, arbitrary patterns of spatial meshes. An example of such a flexible meshing is shown in Fig. 4, where an octant of a core is hierarchically sub-divided with different spatial resolutions, ranging from assemblies to single pin cells.

The thermal-hydraulics module also had to undergo a sound revision of the code structure. At the same time, several improvements were made to the numerical solution algorithms, like e.g. the implementation of a spatial red-black scheme and a pre-conditioned conjugate gradient method to solve the thermal-hydraulics balance equations. Similar to the flux module, the red-black scheme allows for loop-based parallelization of the iterative solver, which has been done by means of OpenMP. All these measures led to a massive performance boost of the new TH

module, as compared to the original COBRA code and make it capable of performing thermal-hydraulics full-core pin-by-pin calculations within less than one minute on recent Linux machines.

For the visualization of results, a dedicated tool, COREVIEW-3D was developed, that allows to plot two- and three-dimensional viewgraphs of relevant thermal-hydraulics quantities for any radial or axial region of the reactor core. A typical example is shown in Fig. 5 above, where the minimum DNB-ratio at a representative 2D-core plane has been depicted.

2.3 The ARTEMIS cross section module

The cross section module of ARTEMIS is an entirely new development and represents one of the major innovations within the core simulator development. It actually consists of two parts:

- The interface code HERMES, which takes as input cross section libraries from the spectral code APOLLO2-A, typically containing several hundred sets of cross sections at different discrete state points (i.e. thermal-hydraulic parameters, burnup, xenon and/or boron concentration etc.). APOLLO2-A libraries typically come with some 140 different nuclides and with 2 to 26 energy groups. The main task of HERMES is to compute for each single cross section so called B-spline coefficients, which yield a reasonably good approximating/interpolating curve for its state-parameter dependency. These coefficients are then stored in cross section libraries suitable for later use with ARTEMIS. B-spline coefficients are the most natural and appropriate choice, since they maximize the degree of "smoothness" of the interpolating curve, whilst at the same time minimizing the numbers of coefficients needed.
- Inside ARTEMIS, these B-splines must be evaluated efficiently for arbitrary combinations of state parameters, as they do occur in each of the reactor nodes. Since ARTEMIS uses a microscopic concept for its cross section representation, this evaluation must take place for each nuclide of the ARTEMIS nuclide chain and each required cross section type (microscopic and macroscopic cross sections, discontinuity factors, form functions etc.). This is in fact one of the most time-consuming tasks in a core calculation.

The reader may note that ARTEMIS employs indeed a much more advanced concept for cross section generation and representation, as it has been commonly used in "traditional" production codes. Most of these codes use relatively simple table interpolation or low-degree polynomials to describe feedback effects. As opposed to such approaches, which always suffer from inherent inaccuracies and are limited in their range of applicability, the ARTEMIS approach is in principle capable of reproducing the spectral code data exactly. This does, however, require that all so called "cross terms" between state parameters have to be taken into account, which usually leads to an unacceptable amount of CPU time needed for evaluation. In general, the choice of these cross terms is a clear trade-off between the desired accuracy of the cross section representation and the performance penalty one is willing to accept.

In order to evaluate the most appropriate choice of cross terms, an automation tool was developed, which performs statistical evaluations of the deviations between spectral code data and their corresponding B-spline reconstructed values. By definition, the B-splines always reproduce the cross sections exactly at the support/fitting points of the spectral code library, however, larger deviations may occur in areas, which are more distant from the user-chosen reference point of the library. Depending on the choice of cross terms, a certain spread can be observed here. Reasonable choices of cross terms seem to be triplets of combinations of burnup, moderator density and one more state parameter (e.g. fuel temperature, boron concentration). In Fig. 6, we have shown a histogram for such a cross term combination; however, instead of sketching individual cross sections, for brevity we have chosen to show the global effect on the assembly eigenvalue. The reader may note, that even for a relatively simple cross term choice, which requires a reasonably low effort for evaluation, the eigenvalues deviate from the APOLLO2-A values by at most 200 pcm (at a standard deviation of ~35 pcm).

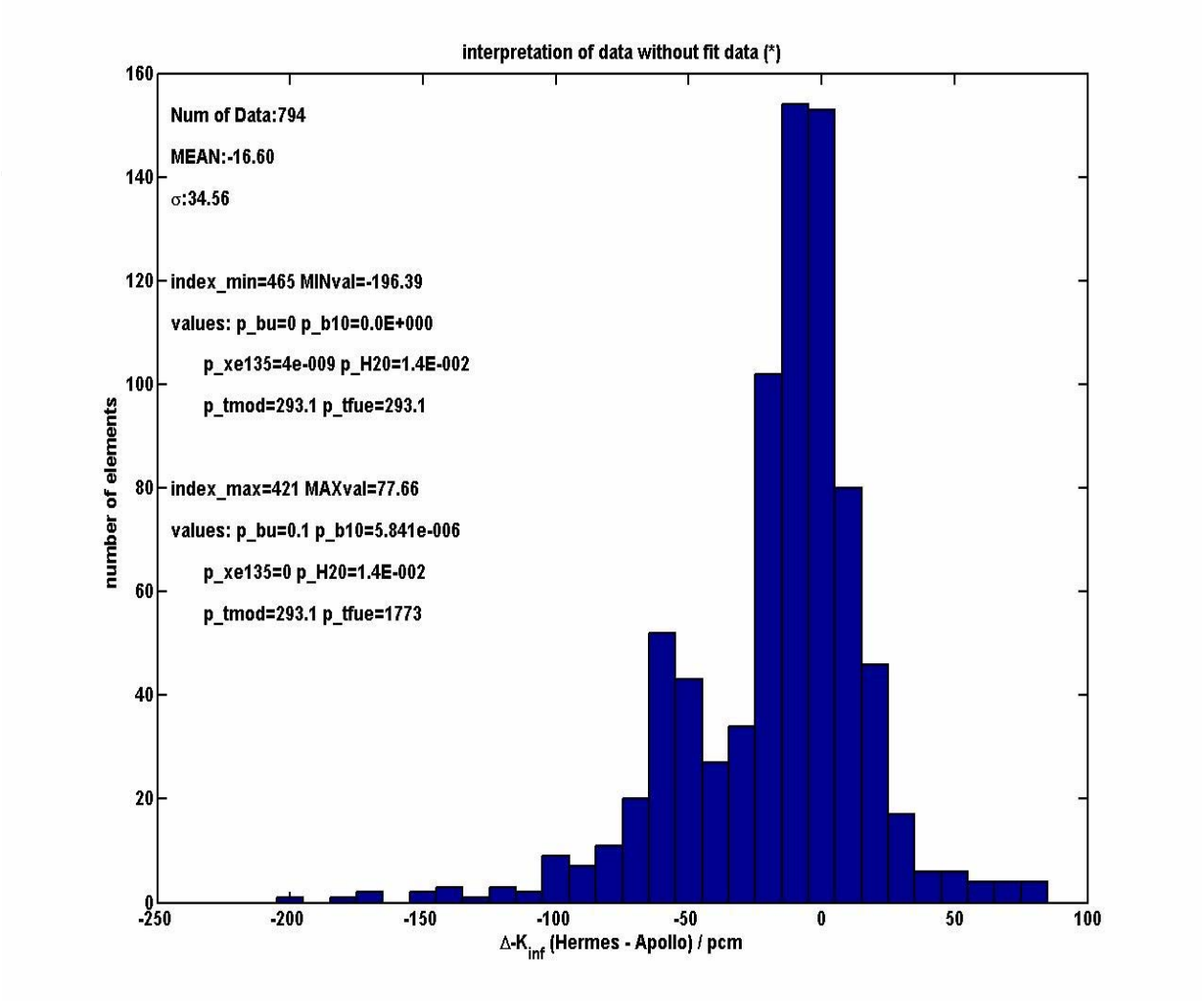

Figure 6: Output of the evaluation tool, which compares K_{inf}-values from APOLLO2-A with those from ARTEMIS' B-Spline representation to characterize the quality of the cross term specification.

2.4 The ARTEMIS depletion solver module

The ARTEMIS depletion solver module was also re-developed to keep pace with increasing user demands for accuracy and performance. One of the requirements for the new, converged code system was a description as accurate as possible of all nuclide reactions, including "feedback" reactions like alpha decay or n-2n transitions. As opposed to many traditional depletion solvers, which take advantage of nuclide chain linearization, such an extended approach makes necessary the usage of a fully featured differential equation solver or of so called matrix-exponential series, as they are most prominently used in the point depletion code ORIGEN [8]. Furthermore, the depletion solver should also be fully consistent with its counterpart from APOLLO2-A.

To achieve these goals, different solver methodologies for the nuclide depletion equations were implemented:

- A high-performance Runge-Kutta solver, which allows to solve the depletion equations even with variable coefficients. This may e.g. be required, if a parabolic dependence on burnup for cross sections or flux shapes shall be described between two flux solutions
- The ORIGEN approach of expanding the matrix exponential into a Taylor polynomial series. This method might suffer from serious round-off errors and may fail to converge for large time/burnup-steps.
- A Krylov-subspace-based representation of the matrix exponential series. Expanding the exponential in a series different to the Taylor series may have a positive effect on the overall convergence and typically leads to much fewer terms to be taken into account in the expansion. This is particularly true, if longer burnup steps are to be performed.

To take into account the variations in flux and cross section shape between two flux solutions, several approaches have been implemented and tested. The simplest approach was already present in older production code systems and works by simply rescaling the fluxes used for burn-up, such that the power produced in the node stays constant over the time step. A more complex approach is the predictor/corrector method, which basically performs the depletion calculation twice, once for the flux/cross section conditions at the beginning and once for those at the end of the time step. This methodology is e.g. implemented in the TRITON depletion sequence [9] of the SCALE5 package. The penalty of performing twice as many depletion calculations is typically more then overcompensated by the savings due to the option for longer time steps.

The depletion module has been tested against numerous different solvers, and for a wide variety of nuclide chains, ranging from very simple chains with only the major actinides and a couple of fission products present up to the complex chains of APOLLO2-A with more than 140 nuclides. Especially for the latter, much effort was spent to find as good agreement as possible for consistency reasons within the ARCADIA system. That this goal was in fact achieved is shown in Fig. 7, where the assembly eigenvalue differences between APOLLO2-A and ARTEMIS are plotted vs. burnup for a UO_2-assembly with 4% enrichment. As the reader may notice, the deviations are not larger than 50 pcm, even for the highest burnup of 80 MWd/kg. Additionally, Fig. 8 shows the nuclide densities for ARTEMIS and APOLLO2-A, and their mutual deviation for the same case. It is remarkable that even nuclides of minor importance like Pu236 or Cm248 with very low concentrations do not deviate significantly from each other.

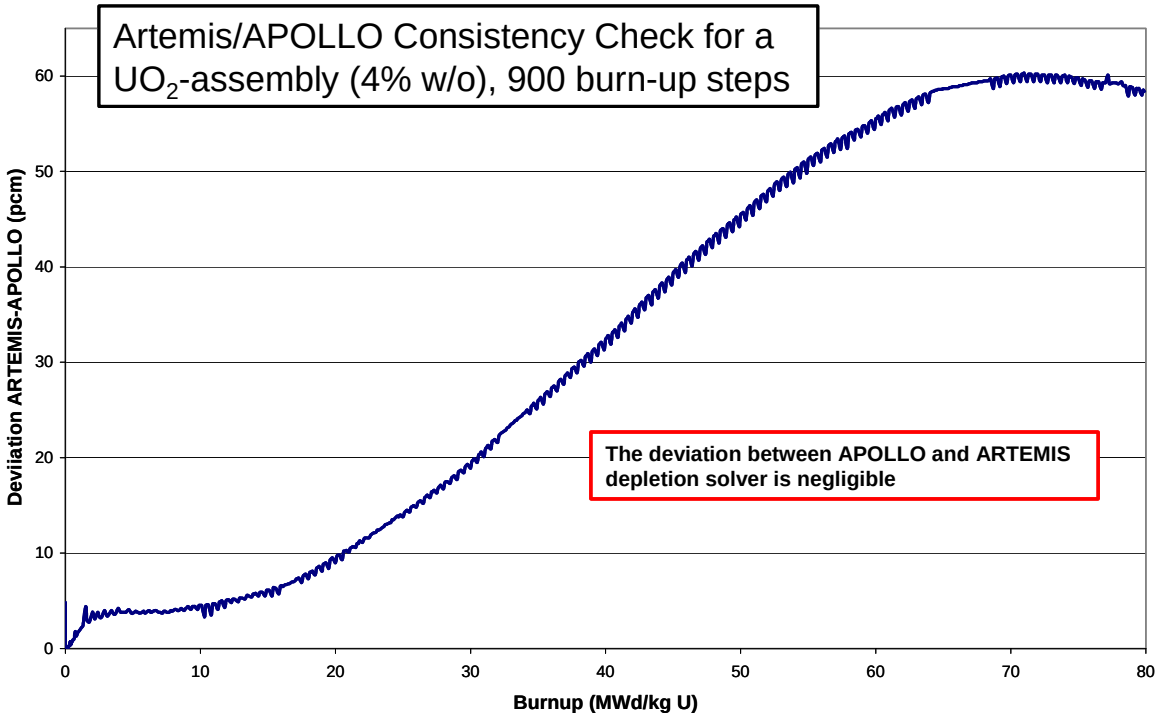

Figure 7: Proof of consistency between APOLLO2-A and ARTEMIS depletion solver.

Nuclide	ARTEMIS	APOLLO	Deviation
U234	0.72235E-06	0.72463E-06	-0.316 %
U235	0.14398E-04	0.14428E-04	-0.196 %
U236	0.35439E-04	0.35477E-04	-0.107 %
U237	0.11151E-06	0.11125E-06	0.227 %
U238	0.59032E-02	0.59027E-02	0.009 %
NP239	0.64473E-06	0.64520E-06	-0.072 %
NP238	0.30938E-07	0.30825E-07	0.363 %
NP237	0.76935E-05	0.76917E-05	0.023 %
PU236	0.33203E-10	0.32835E-10	1.108 %
PU238	0.61286E-05	0.61324E-05	-0.062 %
PU239	0.28691E-04	0.28809E-04	-0.411 %
PU240	0.87188E-05	0.87562E-05	-0.428 %
PU241	0.12427E-04	0.12479E-04	-0.412 %
PU242	0.12667E-04	0.12691E-04	-0.184 %
AM241	0.37710E-06	0.37999E-06	-0.769 %
AM242M	0.67146E-08	0.67660E-08	-0.766 %
AM243	0.51048E-05	0.51067E-05	-0.036 %
CM242	0.29511E-06	0.30097E-06	-1.987 %
CM243	0.12771E-07	0.13012E-07	-1.880 %
CM244	0.44381E-05	0.44269E-05	0.251 %
CM245	0.38982E-06	0.38843E-06	0.357 %
CM246	0.10526E-06	0.10455E-06	0.677 %
CM247	0.22675E-08	0.22482E-08	0.852 %
CM248	0.36546E-09	0.36145E-09	1.095 %

Actinides (after 80 MWd/kg)

Fission Products

Nuclide	ARTEMIS	APOLLO	Deviation
ND143	0.12892E-04	0.12925E-04	-0.256 %
ND144	0.34667E-04	0.34655E-04	0.034 %
ND145	0.14190E-04	0.14211E-04	-0.148 %
ND146	0.20014E-04	0.20011E-04	0.018 %
ND147	0.78925E-07	0.78961E-07	-0.045 %
ND148	0.93924E-05	0.93975E-05	-0.054 %
ND150	0.46997E-05	0.47031E-05	-0.073 %
PM147	0.18914E-05	0.18973E-05	-0.309 %
PM148	0.11897E-07	0.11851E-07	0.387 %
PM148M	0.16234E-07	0.16353E-07	-0.735 %
PM149	0.17156E-07	0.17164E-07	-0.050 %
PM151	0.30661E-08	0.30670E-08	-0.031 %
SM147	0.14864E-05	0.14926E-05	-0.415 %
SM148	0.48134E-05	0.48041E-05	0.194 %
SM149	0.24261E-07	0.24350E-07	-0.370 %
SM150	0.69959E-05	0.70066E-05	-0.154 %
SM151	0.21721E-06	0.21769E-06	-0.219 %
SM152	0.19932E-05	0.19980E-05	-0.239 %
SM153	0.20094E-07	0.20075E-07	0.095 %
SM154	0.11150E-05	0.11159E-05	-0.081 %
EU153	0.31596E-05	0.31649E-05	-0.168 %
EU154	0.12553E-05	0.12564E-05	-0.090 %
EU155	0.22936E-06	0.22957E-06	-0.094 %
EU156	0.12428E-06	0.12401E-06	0.218 %
EU157	0.51373E-09	0.51229E-09	0.281 %
I131	0.92615E-07	0.92665E-07	-0.054 %
I135	0.58134E-08	0.58182E-08	-0.082 %
XE131	0.74946E-05	0.75172E-05	-0.300 %
XE132	0.35445E-04	0.35449E-04	-0.010 %
XE133	0.11648E-06	0.11654E-06	-0.056 %
XE135	0.18068E-08	0.18133E-08	-0.355 %
XE136	0.62150E-04	0.62169E-04	-0.030 %
CS133	0.24383E-04	0.24423E-04	-0.164 %
CS134	0.54963E-05	0.54881E-05	0.148 %

Figure 8: Comparison of nuclide densities at 80 MWd/kg between APOLLO2-A and ARTEMIS depletion solver.

2.5 The ARTEMIS fuel rod module

Although the COBRA-3CP code already contained a relatively simple fuel rod model, it was decided for ARTEMIS that a more sophisticated module must be implemented. One of the major requirements was the consistency with AREVA NP's principal fuel rod code COPERNIC. To ensure this kind of consistency, the major physical correlations from COPERNIC were directly taken over to ARTEMIS:

- The laws for thermal conductivity and heat capacity, as they depend on temperature, burn-up and material composition,
- The radial power distribution in the fuel pellet, calculated by means of APOLLO2-A and tabulated versus burn-up, radius and fuel composition (MOX, UOX, UGd),
- Gap conductance tables are calculated from COPERNIC and are tabulated vs. burnup, pellet temperature, and cladding temperature and then supplied to ARTEMIS,
- Porosity tables are as well calculated from COPERNIC, and tabulated versus radius and burn-up,
- The effective temperature for calculation of the Doppler feedback is evaluated by a correlation particularly suited for fast transient conditions.

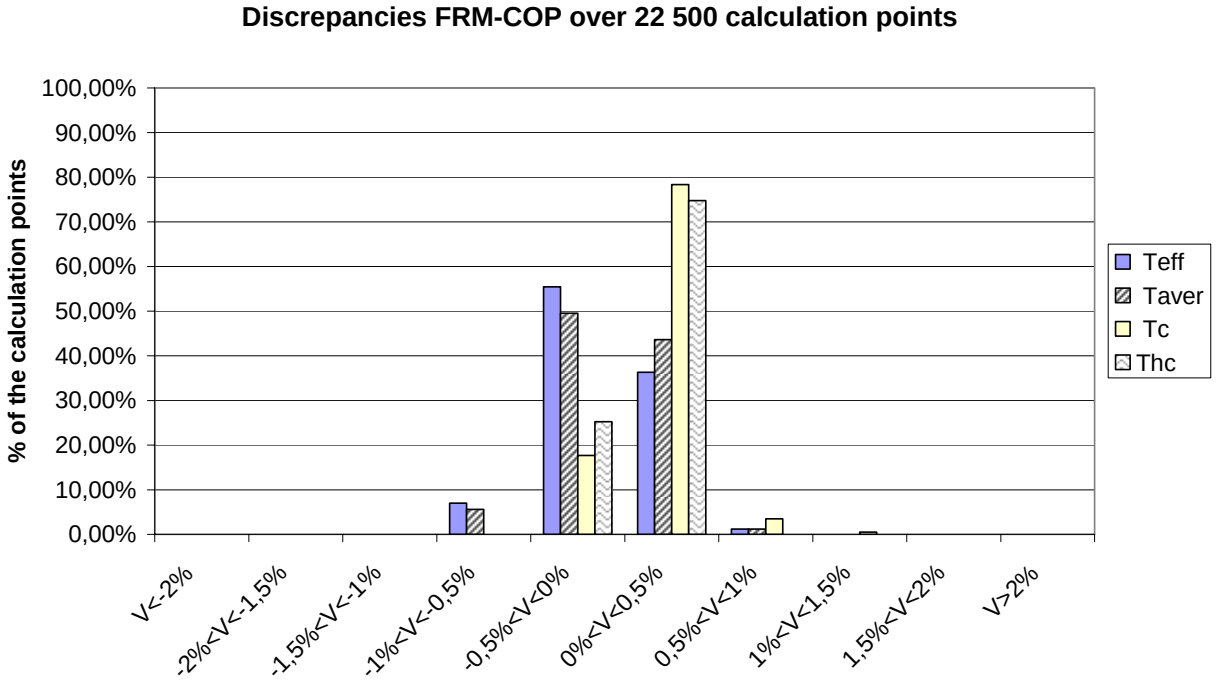

Figure 9: Deviations between ARTEMIS fuel rod module and COPERNIC for the test matrix; compared are effective, averaged, central and surface temperature

To test the accuracy of the new fuel rod module, a large test matrix was defined, covering a large variety of burn-ups, fuel types, pellet diameters etc. The combination of these different fuel types with the different state parameters resulted in approximately 22500 calculation points for COPERNIC respectively the ARTEMIS fuel rod module. The outcomes of this vast number of calculations were statistically evaluated; as a major criterion for acceptance it was postulated that none of the temperatures calculated by the fuel rod module should deviate by more than 1% from the COPERNIC reference. In Fig. 9 it is shown that this goal was indeed widely achieved, with only a very minor fraction of test cases falling outside this acceptance limit.

2.6 The ARTEMIS dehomogenization module

The latest major development, which entered into the new ARTEMIS code system, is the dehomogenization module, which has been completely newly designed and implemented. Its major tasks are the evaluation of pin powers, pin fluxes and pin burn-ups from the nodal core solution and by applying the form functions and discontinuity factors delivered by the spectral code APOLLO2-A.

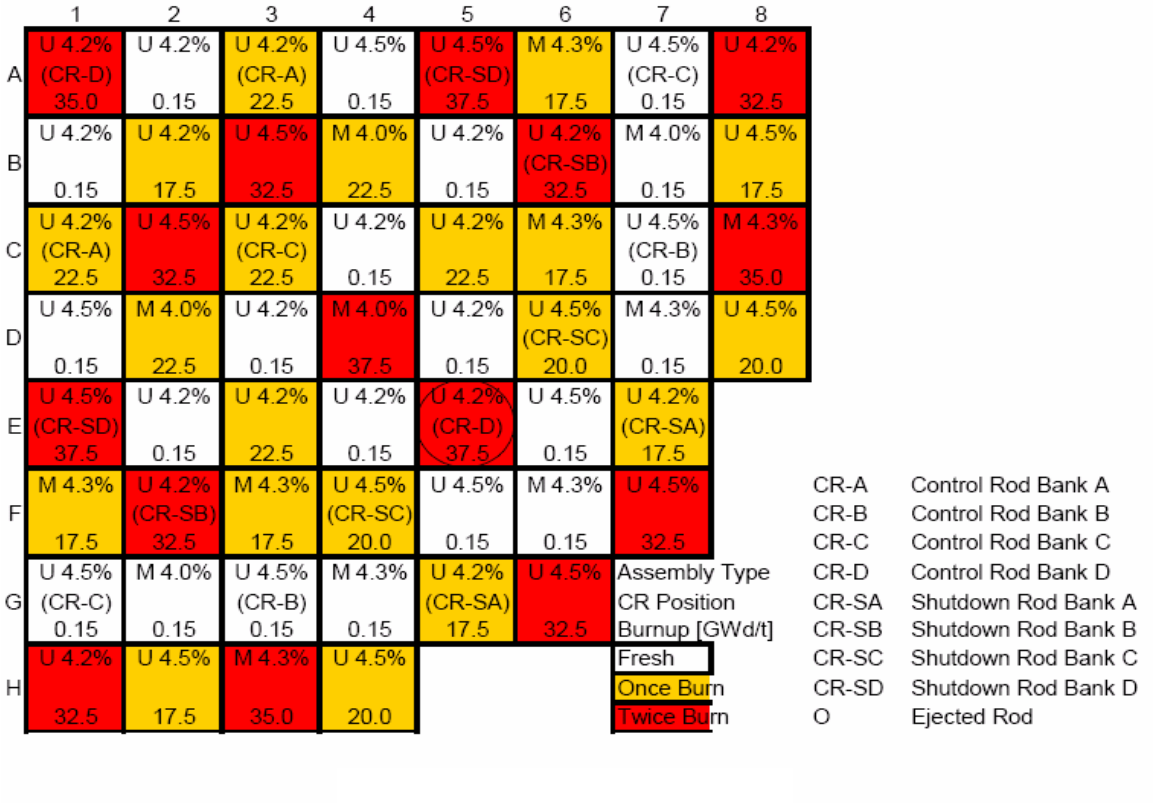

Figure 10: The design of the PWR-MOX benchmark, used for testing the ARTEMIS dehomogenization module.

Compared to earlier versions in current production codes, the new dehomogenization module is fully multi-group capable and supports the evaluation of fully three-dimensional integrated pin-burnup, using pin powers with pin segment weighting. Moreover, it allows the tracking of fuel rods by type and offers detector rate calculation capabilities, which are fed from the corresponding spectral code information. Much care has been taken to correctly describe history effects on form functions, e.g. due to the presence of control rods or burnable poisons for axial power shaping. The evaluation of pin burn-ups can also optionally be based on the explicit depletion of nuclides not only on the nodal average, but also on edges and corners of the assemblies. Despite the increased numerical effort, this option is more accurate than the traditional macroscopic burnup correction used in other production code systems.

The dehomogenization module has undergone a large series of different tests, to validate it against reference pin-by-pin calculations and computational benchmarks. Prominent examples are the KWU-2D burnup benchmark [10] and the very recent OECD PWR-MOX benchmark [7], both for which high-fidelity reference solutions exist. Furthermore, a large test suite has been created from APOLLO2-A colorset calculations and mini-core (e.g. 5x5) problems. Fig. 10 and Table I show as an example the basic layout of the PWR-MOX benchmark and the outcomes of the ARTEMIS dehomogenization module, as it compares with the 47-group Method of Characteristics (MOC) reference solution. The reader may note that the standard deviation for almost all assemblies is far below 1%, which can be regarded as very satisfactory.

Table I: The deviations between MOC reference solution and ARTEMIS for the PWR-MOX benchmark

Assembly Location	Fuel Assembly Type	Dehomogenization Module 4 nodes per assembly	
		Standard Deviation (%)	Maximum Absolute Diff. (%)
A1	UOX 4.2%, BU = 35.0	0.47	1.23
B2	UOX 4.2%, BU = 17.5	0.39	1.08
C3	UOX 4.2%, BU = 22.5	0.43	1.44
D4	MOX 4.0%, BU = 37.5	0.60	1.72
E5	UOX 4.2%, BU = 37.5	0.53	1.70
F6	MOX 4.3%, BU = 0.15	1.80	5.72

3. ARTEMIS VALIDATION

Currently, the ARTEMIS core simulator system is undergoing an extensive code validation and verification phase. Besides the stand-alone tests on the individual computational modules, as they have already been mentioned in the previous sections, integrated tests on the overall system are also performed, both to check for non-regression and for code verification purposes. The following prominent benchmarks belong to the ARTEMIS test suite:

- The IAEA-3D core benchmark, a fully-featured three-dimensional core problem with a large number of different fuel compositions, characterized by macroscopic cross sections
- The OECD Main Steam Line Break benchmark (MSLB, [11]), a relatively recent, quite complex benchmark problem, which tests the transient capabilities of the ARTEMIS code system. Thermal-hydraulics boundary conditions are prescribed at core inlet and outlet.
- The KWU-2D benchmark problem, a depletion calculation for a typical 1300-MW German PWR. The core contains both poisoned and un-poisoned fuel, depletion is based on fixed microscopic cross sections and a simplified nuclide chain [10].
- The OECD PWR-MOX benchmark (see section 2.6), featuring a typical MOX-loaded core with differently depleted fuel assemblies; high fidelity reference solutions do exist for this problem [7].

For brevity, only the KWU-2D benchmark is discussed here. Despite its simplifications, it can be regarded as a fully featured problem with all the main characteristics of a typical full-cycle calculation. Major computational outcomes comprise the power distributions at BOC and EOC, as well as the boron letdown curve, which are compared to the reference solutions. The power distributions are depicted in Fig. 11; the deviations from the reference solution for BOC and EOC are nowhere larger than 0.1% and are thus not shown here. Similarly well described is the boron letdown curve (Fig. 12): at BOC, the boron concentration deviates less than 0.2 ppm from the reference solution; at EOC one finds a natural end of cycle length of 397 equivalent full power days, which is again closer than 0.5 days to the reference.

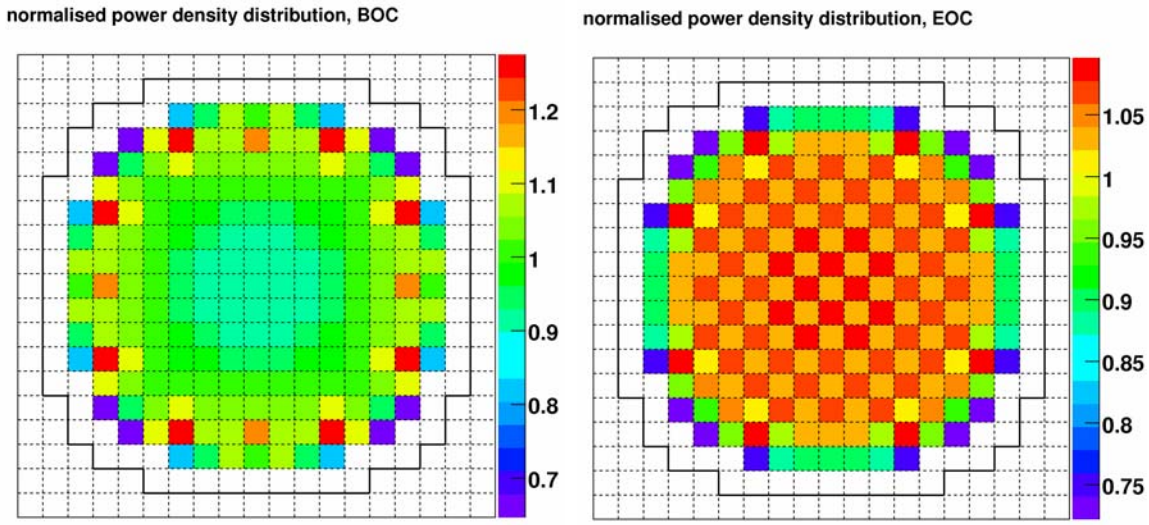

Figure 11: The power distribution of the KWU-2D benchmark at BOC and EOC

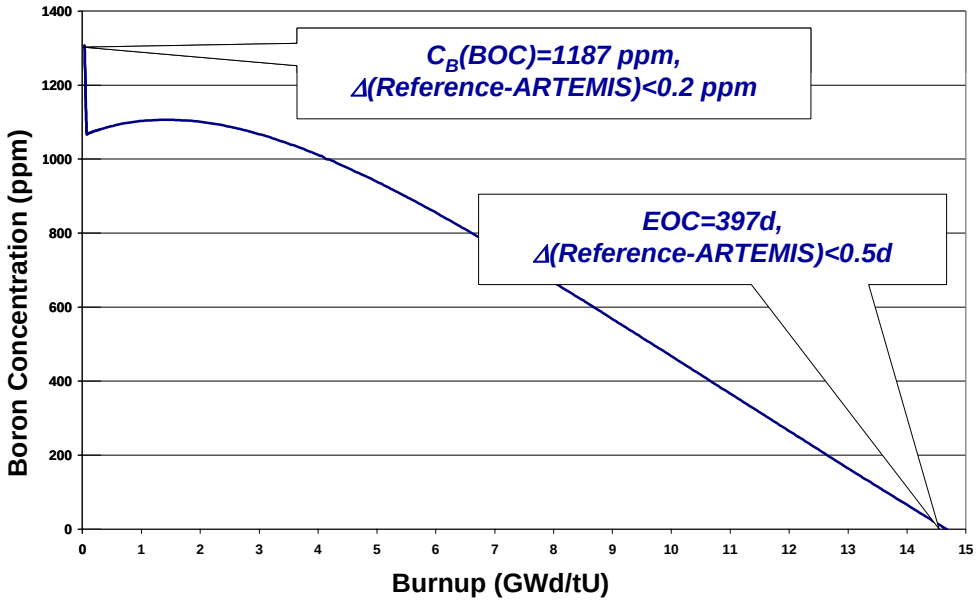

Figure 12: The boron letdown curve of the KWU-2D benchmark

4. ARTEMIS SOFTWARE ENGINEERING ISSUES

The chance to develop an entirely new code system for coupled neutronics/thermal-hydraulics analyses of PWR opened up the possibility to change and re-think about several coding and design practices and paradigms of the past. Several decisions were made for ARTEMIS, which aim for higher standardization, cleaner interface programming and the usage of the most recent software technologies. To illustrate this, several examples are listed below:

- All data files (i.e. cross section libraries, restart and output files, fuel repositories) are stored throughout in the so called HDF5-format [12]. This format offers hierarchical data organization, easy data accessibility via a clean application programming interface (API), while at the same time retaining the advantage of high-performance binary data storage.
- The input to all modules as well as to the overall system is based on the KBF (Keyword-Based Format), which offers an intuitive and ergonomic way of setting up and running ARTEMIS core calculations. For compatibility with e.g. automatic documentation and report generation, the output is also stored in XML-format.
- The usage of Fortran95 as the programming language of choice has been forced throughout the project. Fortran95 yields a high numerical performance, well comparable to FORTRAN77, and better than any other high-level language, while at the same time offering the advantages of modularity and even some object-oriented features.
- The usage of standardized mathematical libraries, like e.g. LAPACK [13] is highly encouraged. Typically, such libraries come as vendor-optimized packages and achieve floating point operation rates close to peak performance.
- Shared and distributed memory parallelization is applied wherever possible. Especially for those modules, which act on in principle un-coupled nodes (e.g. depletion, cross

section and fuel rod module), such parallelism is generally easy to achieve by means of OpenMP directives [14]. However, distributed-memory parallelism via the Message Passing Interface (MPI) is also foreseen [15], e.g. to support spatial domain decomposition.

Besides these changes and improvements, a major step was taken with respect to the basic software analysis and design processes. It is in fact commonly observed, that scientific and engineering software development often lags far behind state-of-the-art methods, when it comes to the practices employed in analyzing software requirements and designing applications. Part of the problem is the vast amount of existing legacy code and the usage of old-style programming languages like FORTRAN77, as well as a certain skepticism against modern programming paradigms amongst scientists and engineers.

Due to these deficiencies, it was decided in the ARTEMIS project to employ a UML (Unified Modeling Language) approach for analyzing and designing the new core simulator software architecture. UML is a graphical, diagram-based language, which impresses by its simplicity. Nevertheless, it is capable of capturing all those features object-oriented high-level languages like C++ or Java typically offer. In a feasibility study, the Objecteering UML modeler [16] was chosen as the future favorite design tool. A strong argument for the use of such a tool has been the integrated development environment, which allows at the same time drawing meaningful diagrams, maintaining up-to-date code documentation (compliant with AREVA NP's high quality assurance standards) and generating source code.

An outstanding application of UML has been developed in the framework of the ARTEMIS project together with Objecteering: the first-of-its-kind Fortran95 code generator (UML2FORTRAN), which generates Fortan95 code directly from UML class diagrams. Although Fortran95 is strictly speaking not an object-oriented (OO) language, it offers many features it shares with other modern OO programming languages, like e.g. data encapsulation, function and operator overloading, modularity etc. With some more effort, also inheritance constructs and static/dynamic polymorphism can be managed. The process of building such quasi-object-oriented Fortran95 applications typically requires a relatively large overhead of source code, which is however strongly template-based and thus ideally suited to be generated by a dedicated tool. UML2FORTRAN generates such code directly from the class diagrams in the UML modeler; the resulting code is 100% Fortran95-standard compliant, extremely clean and has a unique, uniform appearance. Moreover, all code documentation from the UML modeler is straightly implemented in the source code. The tool offers enough flexibility to fulfill all the Fortran programmers needs, even for very special code constructs and statements. Code dependencies are automatically resolved during generation, makefiles and even basic unit test modules can easily be produced. Due to all these virtues, the UML2FORTRAN tool is ideal to build stable, maintainable and quality-assured Fortran code, which is much cleaner than any software engineer would ever develop it "by hand".

The investment in the UML2FORTRAN tool has certainly paid back: today, the UML Modeler is the exclusive development tool for AREVA NP's next-generation reactor simulator and has significantly boosted both code quality and development speed.

5. CONCLUSIONS

The last chapters have illustrated the current development status of the new reactor core simulator system ARTEMIS in the framework of AREVA NP's code convergence project. ARTEMIS is a combination of mature and well-maintained modules on the one hand (e.g. the flux solver) and of new and innovative, modern-style modules for the other central tasks (like cross section evaluation, depletion and dehomogenization) on the other hand. Due to its modular code structure, maintenance and interface issues are much easier addressed than for legacy codes, and responsibilities can clearly be distributed among development team members. This is of particular importance for a highly-distributed team, as this is the case for the CONVERGENCE project, which is geographically spread over all regions of AREVA NP. The introduction of modern software paradigms and of a dedicated iterative development strategy is also a clear "must" in this context.

Currently, validation and verification of the new code system are ongoing, and first users have been involved in testing the software on practical applications, e.g. cycle calculations or safety parameter evaluations. A first industrial version is expected for 2008.

REFERENCES

[1] S. Loubiere, R. Sanchez, M. Coste, A. Hebert, Z. Stankovski, C. Van Der Gucht, I. Zmijarevic, "Apollo2 Twelve Years Later", Mathematics And Computation, Reactor Physics And Environmental Analysis In Nuclear Applications, Vol. 2, pp. 1298-1315, Madrid, Espagne (1999).
[2] Jackson, J.W., and N.E. Todreas, "COBRA IIIC/MIT-2: A Digital Computer Program for Steady State and Transient Thermal-Hydraulic Analysis of Rod Bundle Nuclear Fuel Elements", MIT-EL 81-018.
[3] H. Finnemann, F. Bennewitz, M. Wagner, "Interface current techniques for multidimensional reactor calculations", Atomkernenergie (ATKE) 30 (1977).
[4] H. Finnemann, W. Gundlach, "Space-time kinetics code IQSBOX for PWR and BWR", Atomkernenergie (ATKE) 37 (1981).
[5] Fillion, Ph., Bergeron, A.: FLICA4v1.8.FRA. Modèles physiques, CEA, SFME/LETR/ RT/02-020/A, August 2002
[6] Benchmark Problem Book, ANL-7416, Suppl. 2, Argonne National Laboratory (1977)
[7] T. Kozlowski and T.J. Downar, OECD and U.S. NRC PWR MOX/UO2 core Transient Benchmark, Final Specifications, Rev. 2, December 2003
[8] M. J. Bell, "ORIGEN - The ORNL-Isotope Generation and Depletion Code", Report No. *ORNL/TM-4628*, Oak Ridge National Laboratories, (1973)
[9] M. DeHart, "Lattice Physics Calculations using the TRITON Sequence of SCALE", Physor 2006, Vancouver, Canada, Sept. 2006.
[10] "Benchmark Source Situation BSS-19," National Energy Software Center: Benchmark Problem Book, Report ANL-7416, Suppl. 3, pp. 880 & 892, 1985.

[11] “PRESSURISED WATER REACTOR MAIN STEAM LINE BREAK (MSLB) BENCHMARK”, Volume I: Final Specifications, NEA document NEA/NSC/DOC(99)8, (1999).
[12] “What is HDF5”, visit the website at: http://hdf.ncsa.uiuc.edu/whatishdf5.html
[13] “LAPACK – Linear Algebra Package”, visit the website at: http://www.netlib.org/lapack/
[14] “OpenMP: Simple, Portable, Scalable SMP Programming”, visit the website at: http://www.openmp.org
[15] “Message Passing Interface Forum”, visit the website at: http://www.mpi-forum.org/
[16] “Objecteering Software”, visit the website at: http://www.objecteering.com/
[17] R. van Geemert, “Synergies of Acceleration Methodologies for Whole-core N/TH-Coupled Steady-State and Transient Computations”, Physor 2006, Vancouver, Canada, September 2006.