McSAFE

High-Performance Monte Carlo Methods for SAFEty Demonstration-From Proof of Concept to realistic Safety Analysis and Industry Applications

Research and Innovation actions

Topic: NFRP-2016-2017-01: Continually improving safety and reliability of Generation II and III reactors

2020- Research and Innovation Framework Programme

Contract Number: 755097
Start date: 01/09/2017 Duration: 36 Months

- Deliverable -

D6.1.1: First Synthesis Report

Summary

In this report the progress and achievements obtained during the first 18 months of the McSAFE project are presented for interested parties and possible users of the advanced computer codes and methods for accurate nuclear reactor calculations that will be available after completion of the project.
It is shown that considerable progress has been made already in coupling the general-purpose Monte Carlo computer codes Serpent2, Tripoli4, Monk and MCNP6 with a thermo-hydraulic code and in some cases also with a thermo-mechanical code. Coupling of codes is realised with internal coupling and/or with external coupling implementing all relevant codes in the Nuresim computer platform, based on the Salomé platform.
Moreover, a stable scheme for depletion calculations has been developed and implemented in Serpent2. For 3 of the 4 Monte Carlo codes a dynamic version has been developed to calculate reactor transients taking into account the generation and decay of delayed neutron precursors using advanced simulation techniques. Also various methods have been implemented to reduce the variance in the Monte Carlo calculation and to increase the efficiency of parallel execution of the Monte Carlo codes.

McSAFE – Contract Number: 755097
High-Performance Monte Carlo Methods for SAFEty Demonstration-From Proof of Concept to realistic Safety Analysis and Industry Applications

Document Information

Document title	***First Synthesis Report***
Author(s)	**J. Eduard Hoogenboom (DNC)**
Number of pages	45
Document type	Deliverable
Work Package	WP6
Document number	D6.1.1
Issued by	DNC
Date of completion	25/02/2019
Dissemination level	Public

Document History

Rev.	Date	Comments & status	Author
0	25/02/2019	Final version	DNC

Document Approval

Rev.	Main author(s)	Verification	Approval
Names	J.E. Hoogenboom	J. Dufek, KTH	V. Sanchez, KIT
Date	25/02/2019	26/02/2019	28/02/2019
Signatures			

Distribution list

Name	Organisation	Comments
Roberto Passalacqua	EC DG RTD	

Partner n°	Beneficiary Short name	Type of Beneficiary	Country
1	KIT	University and research centre	Germany
2	DNC	Small and medium company	The Netherlands
3	CEA	Research centre	France
4	UJV	Private company	Czech Republic
5	HZDR	Research centre	Germany
6	WOOD	Private company	United Kingdom
7	VTT	Research centre	Finland
8	KTH	University	Sweden
9	JRC Karlsruhe	European research centre	EU
10	PEL	Utility	Germany
11	CEZ AS	Utility	Czech Republic
12	EDF	Utility	France

Distribution to the partners will be through the project SharePoint.

Content

1. Introduction

The overall objective of the McSAFE project [1] is to move the Monte Carlo based stand-alone and coupled solution methodologies to become valuable and widespread numerical tools for realistic core design, safety analysis and industry-like applications of LWRs of generation II and III. As Monte Carlo codes are very flexible with respect to reactor geometry and composition, the results of the project will also be useful for future types of nuclear reactors as well as to research reactors with complicated geometries and unus ual contructions for experimental research applications. Main targets in the development are advanced depletion, optimal coupling of Monte Carlo codes to thermal-hydraulic and thermo-mechanical solvers, time-dependent Monte Carlo and m ethods for massively parallel simulations.

On one hand, the envisaged developments will permit to predict important core safety parameters with less conservatism than current state-of-the-art methods. On the other hand, they will make possible the increase in performance and operational flexibility of nuclear reactors.

We foster the dissemination of knowledge gained in the project by experienced researchers to become available to any interested party and possible future users of the codes and code systems under development. To that end s everal actions are planned, like a tailored training course, or already undertaken, like digital newsletters. The first newsletter was prepared using the Sendinblue platform and sent out in August 2018. A copy is available via Ref. 2.

Within this scope of dissemination of results, the current synthesis report is issued to inform interested parties about the goals and progress made in the McSAFE project. This first synthesis report coverst he first one and a half year of execution oft he McSAFE project. The planned second and third synthesis report will cover he next periods of the project.

Moreover, to boost the exploitation of project results, an exploitation plan was worked out as well as a detailed dissemination and communication plan and potential users of the McSAFE tools and methods were identified with the help of all project partners.

The McSAFE project will also be r epresented at international events like the NUGENIA forum meetings and at scientific conferences, such as the SNA+MC, M&C, PHYSOR and FISA conferences.

2. The McSAFE Project

Monte Carlo methods are nowadays in wide spread use for nuclear reactor calculations. Although their capabilities have increased considerably over the last decades, supported by the increase in computer power, calculation of detailed power distributions throughout the core of a full-size reactor with sufficient statistical accuracy in acceptable computing time is still questionable.

Moreover, for reactor operation evaluation the actual temperature distribution in the reactor core must be taken into account. Hence, coupling of the Monte Carlo calculation with a thermal-hydraulic calculation is necessary, which must normally be executed in an iterative way. Also the thermo-mechanical behaviour of the reactor core must be taken into account. Hence, coupling of the Monte Carlo code with a t hermo-mechanical code has become a pr erequisite for reactor calculations.

Another challenging task for Monte Carlo is a proper depletion calculation. Most general-purpose codes devoted to reactor physics have an option for depletion calculations, but they take long calculation times and their stability can often be questioned.

For safety analysis not only the static properties of the reactor core are important, but also the dynamic behaviour of the core. This requires the extension of the Monte Carlo code with time dependence. For prompt neutrons this time dependence is often included in the original codes, but for transient reactor calculations delayed neutrons and their precursors are of utmost importance and require a separate treatment in the Monte Carlo code. Transient calculations also requires adequate modelling of control rod movements and, for instance, boron dilution.

This gives a sketch of the main lines of the McSafe project [1]: to develop Monte Carlo codes suitable for Monte Carlo calculations coupled with thermal-hydraulics and thermo-mechanics for a full-size reactor core, including depletion (burnup) and time-dependence for reactor transients. Coupled codes can be implemented in the NURESIM computer platform [3,4], which is based on the Salomé platform [5].

Participants in the McSAFE project are:

Karlsruhe Institute of Technology (KIT, coordinator; Germany)

Delft Nuclear Consultancy (DNC; The Netherlands)

Commissariat à l'Energie Atomique et aux Energies Alternatives (CEA; France)

UJV Rez, a.s. (UJV;Czech Republic)

Helmholtz-Zentrum Dresden-Rossendorf (HZDR; Germany)

Wood (Wood; United Kingdom)

Technical Research Centre of Finland (VTT, Finland)

Royal Institute of Technology (KTH; Sweden)

Joint Research Centre (JRC; EU)

E.ON Kernkraft GmbH (EKK; Germany)

CEZ a. s. Nuclear Power Plants (CEZ; Czech Republic)

Electricité de France (EdF; France)

Each partner brings in its typical experience to one or more of the subjects under development in the McSAFE project. This can be a general-purpose Monte Carlo reactor code, a thermo-hydraulics or thermo-mechanics code, experience in coupling codes or in depletion or time-dependent calculations or providing reactor plant data for validation of the developed code combinations.

In the McSAFE project 4 different Monte Carlo codes are considered, 3 of them being developed in Europe: Serpent2 (VTT), Tripoli4 (CEA) and Monk (Wood). The US Monte Carlo code MCNP6 (LANL) is extended and tailored for the project by DNC. The major thermo-hydraulics code used in the project is SubChanFlow (SCF; KIT). The major thermo-mechanics code used in the project is TransUranus (TU, JRC).
The McSAFE projects benefits from the achievements of the former EU HPMC project. Its goals are in line with the priority topics of NUGENIA and the SNETP platform.

3. Progress in Monte Carlo methods and depletion calculations

In the McSAFE project the Monte Carlo method is the corner stone for neutron transport. Although each of the Monte Carlo codes used in the project has been develped for a long time and has a history of continuous extentions and improvements, several items are identified for further improvement to achieve the efficiency needed for detailed full-core calculations. Therefore, various participants worked on smaller and larger improvements, which cannot all be discussed in this report. Hence, only a selection of subjects will be detailed here.

3.1 Bracketed search for cross section lookup

In Monte Carlo transport codes the energy parameter of a neutrom will change continuously during a history simulation. This requires very frequent lookup of cross section data out of vast arrays of data for each nuclide present in the system. Hence, efficient lookup methods will pay off in a simulation.

Significant efficiency improvements can be achieved by modifying the classic binary search algorithm, used for cross section lookups, to include an additional step known as bracketed search. In this algorithm the binary search is preceded by a step that limits the range of the search. The previous search solution is stored for each nuclide and this is used as the starting point for a new search. Starting at the previous search solution, the range of the grid to be searched is progressively extended by 1, 2, 4, 8, … nodes in the direction that is required in order to give a range that envelopes (brackets) the target energy. If the energy range does not contain the target energy, then the search can be moved to exclude the range that is known not to contain the target energy. Once an envelope has been found, binary search is used as before.

The bracketed search method was implemented in the MONK Monte Carlo code and will be implemented later in other Monte Carlo codes. To briefly illustrate the very significant efficiency gain that can be obtained, Table 1 shows a comparison of run-times for four cases.

Table 1. Comparison of CPU times (seconds) for illustrative MONK calculations using *Binary Search alone, or with Bracketed Search incorporated*

	Binary search	Bracketed search	Ratio (Bracketed/Binary)
Validation case	667	399	0.60
PWR – fresh fuel	1177	1001	0.85
PWR – depleted fuel	7445	4597	0.62
RDB case	1296	569	0.44

The first is one of the validation cases used for MONK, incorporating three clusters of rods separated by boral plates with light water moderator. The second is a model of a single PWR fuel assembly with fresh fuel. The third replaces the fresh fuel composition with a representation of a depleted fuel composition. The fourth is a test case incorporating elevated temperatures in order to invoke the so-called Runtime Doppler Broadening (RDB) model available in MONK.

It can be seen that in three of the four cases an improvement of around 40% or more was achieved using the bracketed search, the exception being the fresh fuel PWR case which gives a more modest overall improvement in runtime – presumably because cross-section look-up does not dominate the calculation runtime.

It is worth noting, that in a sense the bracketed search captures some of the idea of trying to make use of the physics, as it is starting with an assumption that the next search solution is likely to be in the vicinity of the previous search solution.

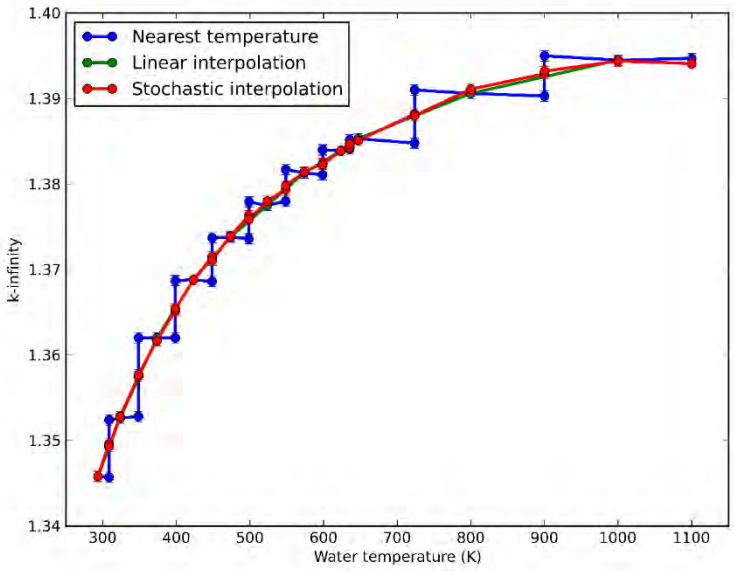
3.2 Stochastic interpolation of bound thermal scattering data

In previous MONK versions only thermal scattering cross-sections were interpolated in temperature, while for secondary particles the code used data from the closest tabulated temperature point. For certain models this gave rise to non-physical step changes in the results. This usually happened when the temperature changed from a value just below the mid-point between two tabulated temperatures to a value just above the mid-point, as the code was switching from the S(α, β) data at the lower temperature to those at the higher temperature.

In the stochastic interpolation method the S(α,β) data are sampled via a method which randomly selects the data tabulated at temperatures T_1 and T_2 with probabilities based on where T_{broad} lies in the range T_1 to T_2.

A single test case has been used to demonstrate the effect of the stochastic interpolation method. This is based on a single pincell model, consisting of a single 3.82 wt % enriched UO_2 unclad fuel rod of radius 0.48 cm, in the centre of a water-filled box with x and y dimensions of 1.2 cm. The hydrogen is all modelled as ^{1}H bound in water and the oxygen is all modelled as ^{16}O. The fuel temperature in this model is fixed at 500 K and the water temperature is varied between 293.6 and 1100 K. Calculations are performed at each of the eleven tabulated temperatures and additionally at points 0.1 K above and below the mid-points between each library temperature in order to demonstrate the step change in k_∞ at the mid-points.

Fig. 1: Results of the test case, comparing interpolation methods
(error bars indicate ±3 standard deviations)



The results obtained using the nearest temperature approach show the expected step changes at the mid-points between tabulated temperatures where the collision processor switches from the data at the lower bounding temperature to the data at the upper bounding temperature (see Fig. 1). As said, these step changes are non-physical and are an artefact of using the nearest temperature data without interpolation.

The results obtained using the stochastic interpolation method completely eliminate the step changes seen in the results based on nearest temperature values and are statistically equivalent to the linear interpolation approach without requiring any post-processing.

The two methods agree, to within the statistical uncertainty, at the tabulated temperatures, and the stochastic interpolation method correctly interpolates at intermediate temperatures, giving confidence that the method behaves as intended.

Furthermore, no significant or systematic differences in run times between the two methods and the stochastic interpolation method have been observed during the calculation.

3.3 Uniform Fission Sites (UFS)

In co-operation with VTT, HZDR evaluates the idea of combining the uniform fission sites (UFS) and response matrix methods in order to obtain acceptable statistical variance in full-core calculations of local parameters throughout the whole reactor core at the same time keeping computation costs as low as possible. The idea of the UFS method is to get more source points in regions where fission power is low, and eventually improve statistics in the outermost fuel pins in a full-core calculations. The core geometry is covered with a mesh in which the code collects the distribution of collision points, flux or fissions during the inactive neutron cycles. This distribution is then used to adjust the number of emitted fission neutrons during the active cycles. A full-core model of the BEAVRS benchmark representing a realistic PWR core was created and used for studying the effect of the UFS method within Serpent2.

The full-core model was calculated without UFS (“standard” calculation) and with UFS. To assess variation in radial and axial power distributions standard and UFS calculations were repeated 40 times with different random number seeds. The mean values of local powers and their relative standard deviations were obtained from these 40 independent tests. To make results of standard and UFS calculations comparable, the number of neutron histories in UFS calculations was adjusted to preserve calculation time.

The BEAVRS benchmark is based on a real PWR core with relatively flat radial power distribution. The variance of radial power in the centre of the core and in the periphery are similar even in the standard calculation. Correspondingly, the application of the UFS method has no significant effect on radial power variance. On the other hand, the axial power in the considered case has a typical cosine-like distribution and the variance of local power in top and bottom axial layers is a few times higher than in the central layers. The application of the UFS method equalizes the axial distribution of variance of local power, decreasing variance in periphery layers and increasing it in central layers.

It is also worth to mention that application of UFS does not affect the resulting mean values of the global multiplication factor and local power distributions, which indicates that a fair Monte Carlo game is assured.

3.4 Response Matrix methodology

VTT is developing a response matrix based methodology for variance reduction purposes. The existing solver was modified to produce an improved initial guess for the fission source distribution, leading to faster source convergence. The solver obtains coupling coefficients required for the response matrix solution from forward Monte Carlo simulations and provides a spatial distribution that approximates the converged fission source. The implemented methodology has been demonstrated by single-assembly and full-core PWR calculations. The results show that the improved initial guess leads to faster source convergence in terms of both inactive cycles and overall running time. The implementation supports rectangular and hexagonal core geometries.

The work was presented at the PHYSOR 2018 international conference and a paper has also been published in Annals of Nuclear Energy (Ref. 6).

3.5 Optimisation of MC and MC-TH coupled criticality calculations

KTH developed a source convergence acceleration method for Monte Carlo criticality calculations. The method increases the neutron population size over the successive criticality cycles. This helps to iterate the fission source faster at the beginning of the simulation where the source may contain large errors coming from the initial cycle. As the neutron population size grows over the cycles, the bias in the source gets reduced. Unlike previously suggested acceleration methods that aim at optimising the neutron population size, the new method does not have any significant computing overhead. Moreover, it can be easily implemented into existing Monte Carlo criticality codes. The effectiveness of the method was demonstrated on a number of PWR full-core criticality calculations. The method was implemented into the Serpent2 code and published in the form of a journal article (Ref. 7).

3.6 Parallelisation of the thermal-hydraulics code SCF

For parallelisation of computer codes two basically different methods exists: MPI (Method Passing Interface) and OpenMP (Open Message Passing).

OpenMP executes a selected loop in a program or subroutine on different processor cores with different loop variables, which can be used to simulate different particle histories on different processor cores. It can be run for two or more cores on the same computer node of a computer cluster and uses shared memory on the computer node.

MPI executes identical copies of a computer program on different processor cores, where different particle histories can be simulated using the different MPI copy number in each copy. Data can be exchanged between different MPI copies, for instance for the master MPI copy to collect all history results from other MPI copies. MPI can be run on different processor cores on the same computer node or on a different computer node.

MPI and OpenMP can be applied to the same computer code. For large-scale parallelisation on a large computer cluster MPI can be run with one copy on each computer node, using OpenMP for all available processor cores on each node. However, special efforts are often needed in programming the MPI communication between MPI copies and in the OpenMP loops to take maximum advantage of using large numbers of processor cores.

Monte Carlo codes are by their nature of simulating independent particle histories quite suitable for parallelisation and all general-purpose Monte Carlo codes are prepared for both MPI and OpenMP parallel execution. For a thermal-hydraulics code like SubChanFlow (SCF) this is clearly not the case. The code contains numerous loops and iterations over axial levels, fuel rods, coolant channels and many others, which are not independent. Some parts of some subroutines are programmed for OpenMP parallel execution, but with limited speed-up. No MPI programming is present in the original version developed by KIT.

However, when executing a coupled MC-TH calculation on a large computer cluster, using all processor cores for efficient execution of the Monte Carlo part, only one processor core or at best one computer node can be used for the thermal-hydraulic calculation, leaving all other processor cores or computer nodes idle for the time the TH code is running. Although the time spent to the TH calculation may be relatively small as compared to the time spent in the MC calculation, for a

full-size core the TH calculation can consume considerable computer time and for an efficient parallel calculation MPI parallelisation of the TH code is desirable.

For the thermal-hydraulics code SCF the main time-consuming subroutines are determined using the profiling option of the compiler. In the SCF code within each outer iteration and within the loop over axial levels a number of subroutines are called for various tasks with regard to the axial level. In each of these subroutines there are one or more loops over all fuel rods, or coolant channels or "gaps" (the hypothetical surface between two coolant channels). These loops are subject to MPI parallelisation and results are collected by the master MPI copy and broadcasted to all other MPI copies when the loop has finished.

The performance of the parallel execution for a stand-alone run of SCF is tested for a ¼ core of a large PWR with 15,120 fuel rods and 18,144 coolant channels. Table 2 shows calculation times (in ms) per subroutine call for execution with 1, 2 and 8 processor cores. The results show that a gain in parallel execution is obtained, but that further research is needed to scale up to higher number of processor cores.

Table 2. Computing times per subroutine call and speedup for different number of processor cores

		computing time (ms or s)				acceleration	
subroutine	# processor cores: goal subroutine	1	2	8		2	8
heat_pin	fuel rod internal temperatures	146	83	21		1.8	7.0
prop	thermal-hydraulic properties	80	47	37		1.7	2.2
energy	solves energy equations	25	24	17		1.0	1.5
heat_average	average fluid temperature and heat transfer coefficient	35	18	3		1.9	11.7
pgrbcgs	solves pressure gradient by conjugate gradient solver	129	131	90		1.0	1.4
	total axial level	436	322	182		1.4	2.4
SCF	single outer iteration (s)	99,8	73,2	44,7		1.4	2.2

3.7 Improved algorith for depletion calculation

For an equilibrium Xe-135 calculation there are specific problems in stability of Monte Carlo simulation with fuel depletion and possibly other feedbacks considered. Under specific conditions the feedbacks can make the simulation unstable, rendering results unusable. This problem has been the subject of a number of studies in the past. As a result, new methods, such as the implicit Euler based coupling schemes, have been adopted for Monte Carlo simulations.

KTH focusses on how to optimise the efficiency of the numerically stable coupled Monte Carlo simulations. The efficiency of Monte Carlo depletion calculations is affected by a number of parameters, such as the time step length, the number of Monte Carlo criticality runs per time step, the number of inactive and active cycles performed in each criticality calculation and the number of neutron histories per criticality cycle. Setting these parameters at non-optimal values results in suboptimal computing efficiency. The objective is to automatise the selection of the free parameters for achieving the best efficiency.

The principal problem of studying the optimal configuration is the need for a vast number of Monte Carlo simulations, which is very expensive in terms of computing cost. It was decided eventually to

develop a new numerical tool based on a diffusion solver, mix random noise into its solution and include a simplified fuel depletion feedback. This way allows carrying out a large number of coupled simulations relatively quickly. This step has been completed. Numerical tests confirmed that this tool allows studying stability-related phenomena of true Monte Carlo depletion codes.

Preliminary tests showed that no matter what coupling scheme and other configurations are used, it is always advantageous to include the Xenon feedback in all Monte Carlo criticality runs, i.e., the equilibrium Xenon concentration should always be considered.

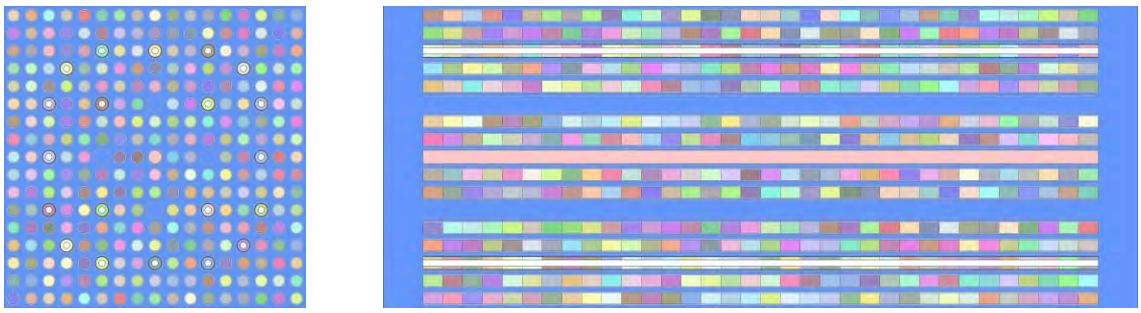
Preliminary results also suggest quite clearly that the error in results is very sensitive to changes in the actual statistics per single Monte Carlo criticality simulation. Quite surprisingly, the results appear to be much less sensitive to the length of the time step. Hence, coupling schemes that perform a number of Monte Carlo criticality runs per step may achieve about the same results as schemes that execute the criticality simulation only once when the statistics per Monte Carlo criticality calculation are about the same (while other parameters may differ).

3.8 Stability and scalability of MC-based full core depletion calculations

To assess and where possible extend the current capabilities and limitations of the Serpent2 code on large depletion calculations a simplified 3-D x-y reflected model of a fuel assemly was defined, based on the BEAVRS PWR 17x17 FA with 20 burnable poisons. Several simplifications were performed to avoid axial dependence of the geometry.

For the automatic depletion zone capability of Serpent2 and using 34 axial subdivisions for burnable poison (BP) and FR as a whole leads to 68 burnable zones. Using these axial zones for each BP and FR pin leads to 9656 burnable zones. Fig. 2 illustrates this case.

Fig. 2. Simple FA problem with 34 axial divisions for each pin modelled (total of 9656 zones)



a) x -y cut

b) z-x axial cut in FA center (not to scale)

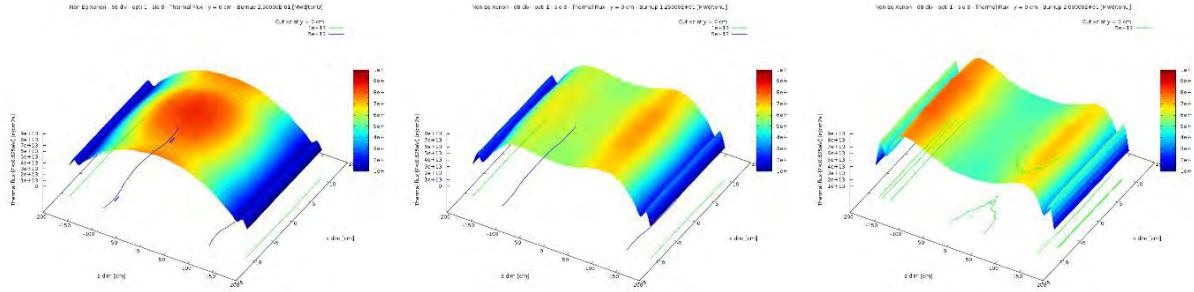
Several modeling alternatives were considered with regard to the Xenon treatment (either in equilibrium or not), the size of the source population and the method for solving the depletion equations using the predictor-corrector (PC) method or the Stochastic Implicit Euler (SIE) method.

The FA level calculation was performed to evaluate potential instabilities and scalability of conclusions from pin level to FA level. The results for non-equilibrium Xenon were calculated for the PC method and with SIE 8 method.

The results of the thermal (E<0.625 eV) neutron flux z-x profiles and integrated axial power distributions are presented in Fig. 3 for the SIE8 method for three depletion times and no equilibrium Xenon considered in the model.

It can be seen from this figure that oscillations are observed in this case. From many different cases it is concluded that adding depletion zones tends to stabilise the solution. However, the SIE8 method does not present a stable solution.

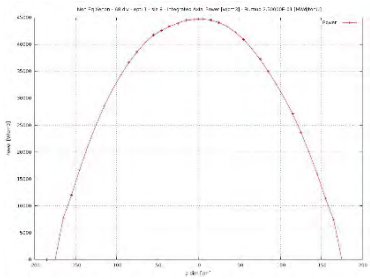
Fig. 3. Thermal neutron flux and power profiles with 96 burnable zones, without Xenon equilibrium and using the SIE8 method.



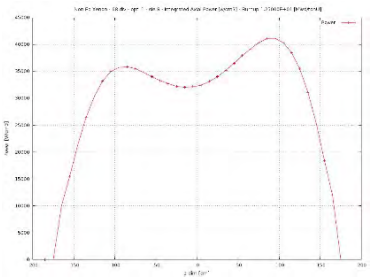
a.1) Thermal x-z flux profile for Bu=0 MWd/kgU

b.1) Thermal x-z flux profile for Bu=12.5 MWd/kgU

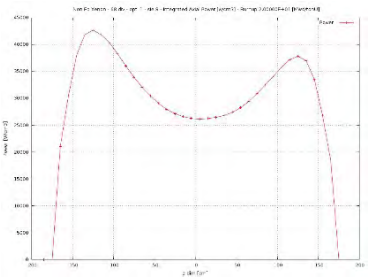
c.1) Thermal x-z flux profile for Bu=20 MWd/kgU



a.2) Integrated axial Power [W/cm3] for Bu=0 MWd/kgU



b.2) Integrated axial Power [W/cm3] for Bu=12.5 MWd/kgU



c.2) Integrated axial Power [W/cm3] for Bu=20 MWd/kgU

As a second step, in order to analyse the scalability of the results presented ,a square array of FAs was considered, where the BEAVRS PWR FA pitch is used as basis. Based on the previous analysis, an FA array of fully divided FAs is considered in order to check the RAM memory requirements and to estimate the parallelisation scalability of the problem.

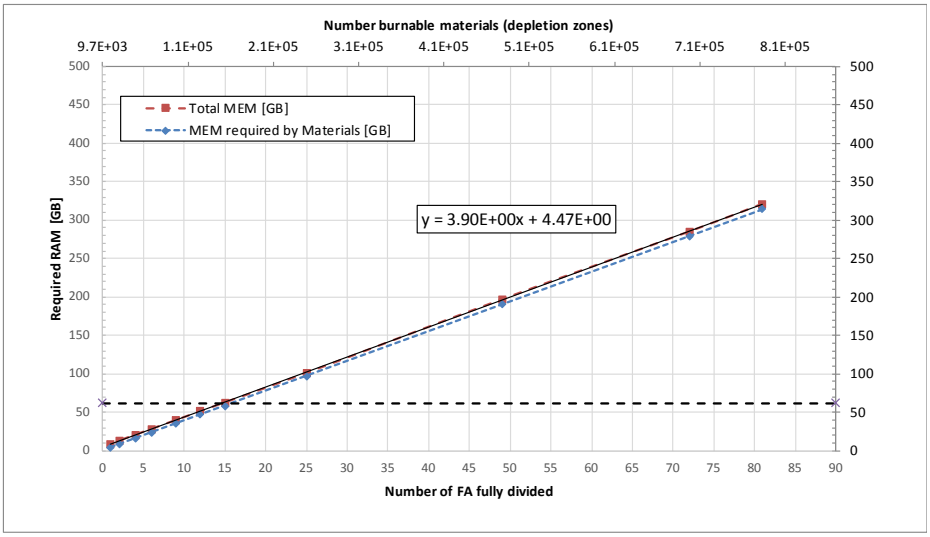
It should be noted that the direct extrapolation of results up to the 64 GB RAM limit (Serpent2 uses by default 80% of available memory as limit, but this can be changed by input to higher values) sets the maximum number of FAs to be modeled to ~13, while the full 3-D BEAVR problem includes 193 FAs. Therefore, the following approach was conducted:

- Consider an 11 x 11 square array. This case considers 20 BP, which represents a conservative approach to include additional materials for each FA.
- Fill the array with increasing number of FAs and obtain RAM memory requirements, considering the lower memory optimisation mode available in Serpent v2.1.29 (option 1) and 20 OpenMP processors. These preliminary tests were performed in an ad-hoc node with up to 480 GB of RAM.
- Identify the limits (i.e. maximum case) and proceed to full-scale test of depletion and parallel scalability.
- Perform a test in an HPC computer cluster for the most demanding case that could be run, in order to estimate the resources required and identify other potential issues.

The scalability of RAM memory requirements was obtained for each of the alternatives considered. The results are presented in Fig. 4. In addition a 90 % of the 64 GB limit is included in Fig. 4 together with a linear fit of memory requirement in order to perform the estimation of maximum number of FAs that could be represented in the computer cluster used.

It can be observed from Fig. 4 that the foreseen RAM memory requirement for a 193 FA core with full division of depletion zones is around 750 GB (830 GB if a 90% limit is stated). The actual 64 GB limit applicable for the computer cluster sets a maximum foreseen number of FAs to be managed in the range up to 12 (if a 90% limit of total RAM is stated).

Fig. 4 Multithread memory requirements



As a result, if a full PWR depletion case is to be developed the following alternatives are to be considered:

- Implementation of the Domain Decomposition Scheme (see Sect. 3.9). This approach can allow dividing the memory requirements on each node fitting the 64 GB limits. 15 – 20 zones will be enough to face actual memory limitations using optimisation mode 1 in Serpent2. It should be regarded that higher performance clusters usually have lower RAM availability (order ~10 GB).
- To assess high-performance goals, higher optimisation memory modes should be used, which states additional requirements for RAM consumption.
- Model 1/8 of a full-size PWR core: This option will fit the memory requirements, but will not allow to model several safety parameters (such as RIA, REA, etc).
- Perform calculations in special nodes with large memories, possibly up to 1 TB. Often the problem with this option is a maximum running time and only a limited number of such computer nodes available.

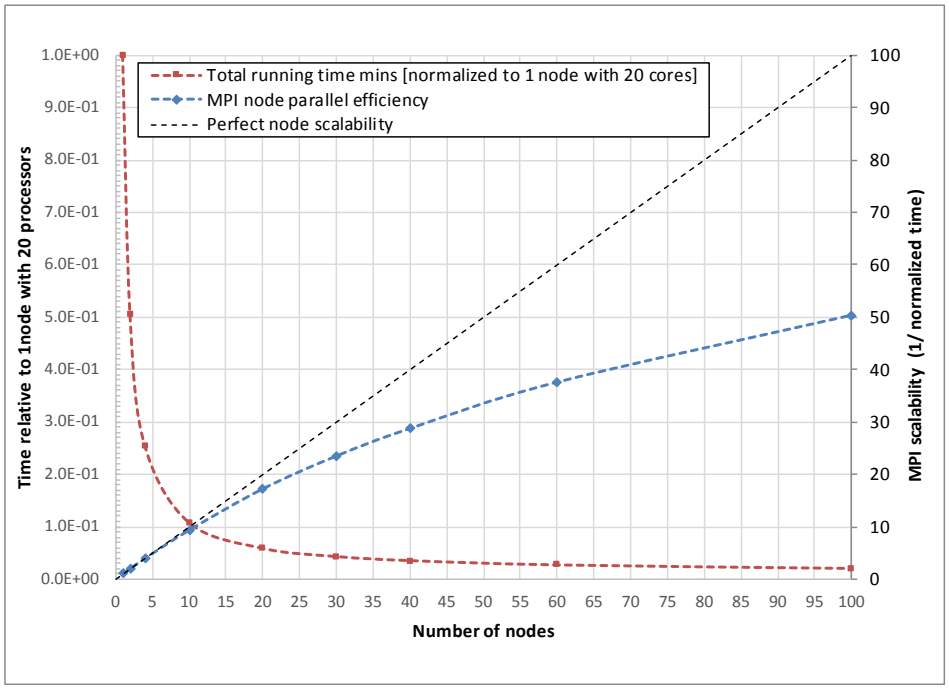
Based on the results and identified issues, a 3x3 array was chosen, filled with 8 FAs. This test of CPU scalability was developed using the lower memory optimisation mode. The model totalises 77248 burnable zones.

In order to analyse the scalability the same cases were run considering more active cycles (5 10^8 total active histories). The results are presented in Fig. 5. Good scalability is found up to ~2000 cores.

It should be regarded that this fact is related to the hybrid OpenMP-MPI implementation in Serpent2, which by default divides the particles for each MPI executable. Further investigation of this aspect should be developed in order to assess whether this approach leads to potential statistical oscillations in fission source convergence.

The scalability of memory requirements behaves linear with the number of depletion zones in lower optimisation memory mode. It should be noted that scalability is developed for a multithread scheme.

Fig. 5. Scalability plot for the 8 FA case with full division and 10 times more cycles



Previous sections presented the pin to array approach, that allowed defining the basis of the expected requirements, namely convergence levels and RAM requirements. As far as the McSAFE final goal intends to develop high-fidelity calculations, a verification of capabilities for the maximum foreseen case that could be run in available HPC computer clusters has to be developed. As a result, based on the results the most demanding problem that could be allocated in the current computer nodes was considered, With this model, an estimation of main run parameters can be carried out, which are presented in Table 3.

Table 3. Main parameters of job execution

Parameter	Value	Comment
RAM memory (Total)	50.456 GB	Includes all. The node limit is 64GB. Additional memory is to be used by other codes in coupling
RAM memory (materials)	46.063 GB	Includes all
Running time @ 2.6 GHz CPU	~80e3 mins [1]	Per Iterator or corrector step
Running time @ 2.6 GHz CPU w/o IFC	~50e3 mins[1]	Per Iterator or corrector step
Calculation time overhead due to IFC	**~60%**	
Total active histories	**2e9**	2000 active cycles of 1e6 histories each

[1]43200 mins ~1 month CPU

3.9 Collision-based Domain Decomposition (CDD)

The Serpent2 Monte Carlo code has advanced depletion options and is therefore used to explore the possibilities for full-core depletion. For a high-fidelity full-core PWR depletion calculation up to 1 TB of memory or more may be needed even using the lowest optimisation mode in Serpent2, i. e. the one that consumes the least amount of memory. The in-node memory of a high-performance computer typically ranges from 4 GB to 64 GB and in some rare cases can be as much as 512 GB. Therefore, the most demanding problems aimed at in McSAFE are expected not to fit in the memory available to most users as shown in the previous subsection. Moreover, it

would be desirable to use higher optimisation modes in Serpent2 in order to get shorter runtimes, although this would lead to larger memory demands.

Considering this issue, two general approaches are feasible. On the one hand, the total memory demand can be reduced to fit in the memory of each compute node. Serpent's optimisation modes are related to this and strategies such as discarding nuclides with negligible impact on neutronics could also be implemented. However, being able to fit in memory a problem that takes about 1 TB with the current Serpent2 version using this approach is very unlikely.

On the other hand, a parallel scheme with memory scalability can be implemented. Using such a scheme, compute nodes can be added until the total available memory is sufficient to fit the memory demand of the problem. Hence, the model has to be partitioned and distributed among nodes using MPI parallelisation until every part of the problem fits in an individual node. The drawback of this approach is that in principle each partition/process will not have the complete information about the system, and therefore the tracking algorithm has to be modified accordingly. Particle-based parallelism can still be used in this case, as all particle histories are independent from each other. This strategy was chosen to tackle the issues posed in this task.

At least two methods with memory scalability have been proposed, namely Data Decomposition (DD) and Spatial Domain Decomposition (SDD).

In Data Decomposition, information such as material compositions is distributed across computer nodes, scaling down the in-node memory, and is communicated between processes when needed. For example, if a particle being tracked by a process ends up in a region which compositions are stored by a different process, this data has to be transferred to continue the tracking. The implementation of this method in Serpent2 would be cumbersome due to the complexity of the data structures, and besides its efficiency is not clear given the fact that it has never been implemented in an actual code.

In Spatial Domain Decomposition, the geometry is partitioned and each subdomain is assigned to a different process that has all the information about its part of the model but none about the rest, hence also decreasing the in-node memory demand. In this case particles that reach domain boundaries have to be communicated between domains/processes so that all histories are properly simulated. This method has been implemented in several codes with very good results. The main issue to develop an SDD scheme for Serpent2 is the partition of the CSG geometry, which is an exceedingly complex problem and is not feasible within the McSAFE project.

Combining ideas of both DD and SDD, a Collision-based Domain Decomposition (CDD) scheme was developed, implemented and tested in Serpent2. This work is presented here. The main idea behind CDD is to partition the model in a way similar to Data Decomposition and to use a tracking method as the one used in Spatial Domain Decomposition.

In CDD, the model is decomposed material-wise, i. e. each process is assigned a list of materials and it only contains the data for the ones it owns. Every process has the complete information about the geometry, with a set of materials for which it has no data. Given that the huge memory demand in depletion problems is caused by burnable materials, the decomposition is only done over the fuel. Other parts of the model such as coolant, cladding and structural components are replicated in every process. An example of this type of decomposition is shown in Fig. 6, where the fuel materials of a VVER fuel assembly are decomposed in three domains, while the rest of the materials are replicated.

During the tracking, a particle can have a collision in a material for which the data is not available. In a DD scheme, this information has to be transferred from the process that owns that material to the current process in order to simulate the collision. In the new CDD method, the particle is sent to this other process instead, as in SDD, and the collision is simulated there. After the collision in

Fig. 6. Material-wise decomposition for a VVER fuel assembly. The colours of the pins indicate the domains of its materials. The water and cladding materials are not decomposed.

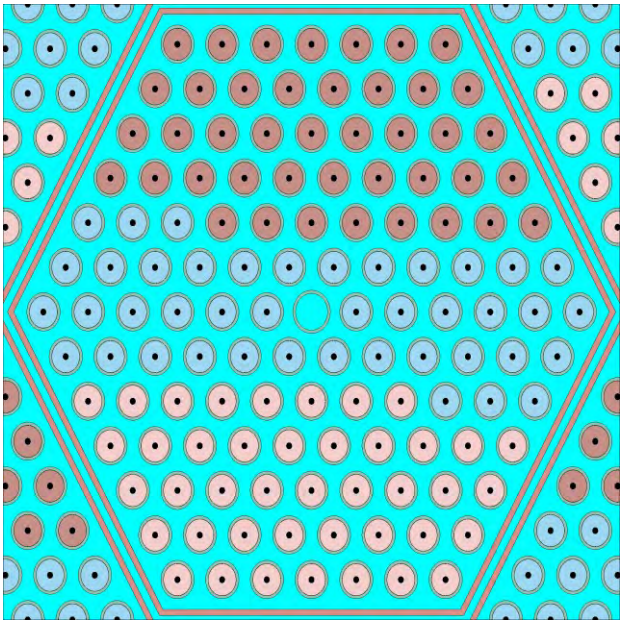
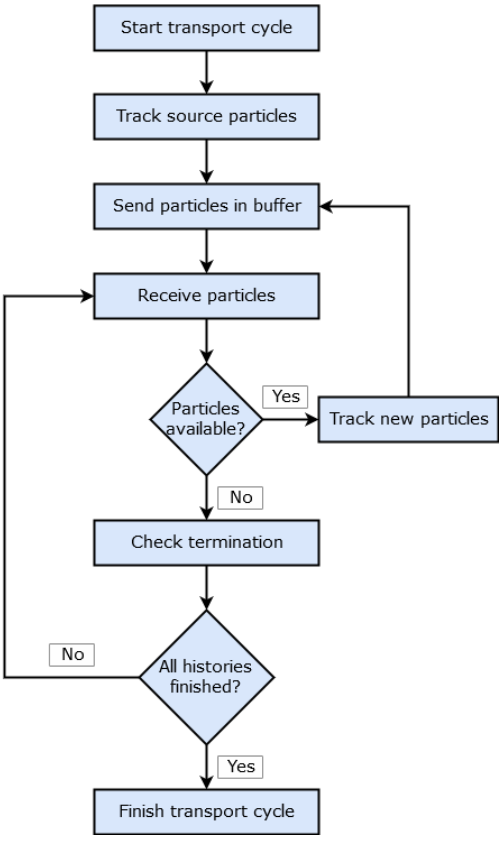


Fig. 7. Tracking scheme for the SDD method



the new domain, the tracking of the particle resumes there until its history is finished or it crosses to a different domain. This has to be done unt il all particles created in all domains have been tracked.

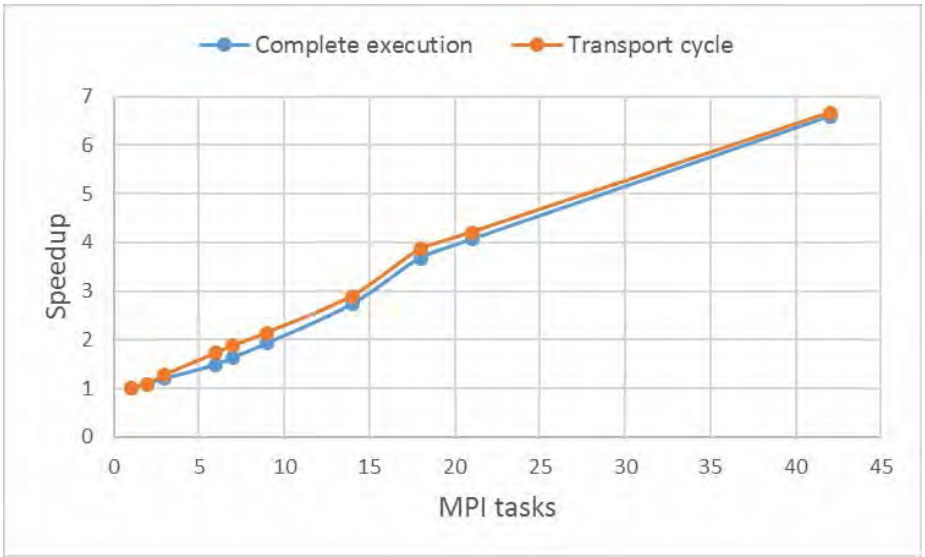
A modified tracking method has to be used in each process in order to track not only the particles born from the fission source in that domain, but also the particles born in other domains that have collisions in materials owned by that process. The algorithm implemented in Serpent2 is shown in

Fig. 7. First, the transport cycle is started and the source particles are tracked. During tracking, particles that have collisions in materials outside the domain are not sent right away, but are stored in a buffer. When the tracking is finished, the particles in this buffer are sent to the corresponding domain and the particles coming from other processes are received. If new particles have actually arrived this tracking scheme continues in a loop. If no incoming particles are received, a global check is performed, synchronising all the processes and checking if all histories started have been finished. If this is the case the transport cycle is finished, and otherwise the tracking continues.

The efficiency and parallel speedup of this tracking scheme is highly dependent on the degree of synchronisation between MPI tasks, as is the case for most parallel algorithms. Therefore, the particle transfers are done asynchronously, in order to minimize the synchronised operations. Particles are sent to the corresponding processes without waiting for them to be received using the MPI non-blocking function calls.

The speedup when running multiple MPI tasks is shown in Fig. 8 for the complete execution and for the transport cycle only. The acceleration is linear, but the efficiency of the algorithm is quite poor (about 16% for 42 tasks). Moreover, the performance appears to be limited by the transport cycle, since the overall speedup is that of the transport cycle.

Fig. 8. Speedup with the CCD method using multiple MPI tasks.



The speedup of the proposed CDD scheme depends strongly on the method used to check for the termination of the transport cycle. This is because at this point all the MPI tasks need to be synchronised, which leads to a loss in performance if the load imbalance of the problem is significant. The particle transfers are very efficient due to the use of non-blocking communications to send and receive them asynchronously.

A first version of CDD was implemented in Serpent2 with relative success. The test cases used for verification show a small yet clear bias in the results caused by the new scheme, which will have to be addressed before having a reliable version. Regarding performance, a more efficient termination scheme is needed in order to improve the speedup of the transport calculation, which is the most time-consuming part of a high-performance Serpent2 simulation.

Once the CDD scheme is verified and optimised, tests with large-scale problems have to be carried out to assert its potential to achieve the objectives proposed for the McSAFE project. In particular, memory scalability tests will be done in connection with the memory bottleneck already identified for full-core depletion problems.

3.10 Validation

One of the most important potential areas of application of coupled Monte Carlo (MC) – thermal-hydraulic (TH) – thermo-mechanical (TM) codes under development are the reference core calculations, for both steady-state and short-time kinetics problems. The approach to the solution of the problem of neutron transport in a reactor core avoids most of the approximations used in standard up-to-date tools for reactor core calculations so it should lead to very trustworthy results. Nevertheless the precision is still limited due to slow convergence of MC calculations and uncertainty both in fundamental nuclear data and in core description (thermal expansion, fuel assembly deformation, etc.). Validation of the newly developed coupled codes against experimental data is therefore needed in order to assess the precision of the coupled MC calculations, prove that the results are reliable and determine how the precision compares to current industrial tools. The objective of our validation is to furnish background for answering these questions – full validation is out of scope of this project.

The main part of work related to the validation and testing will start in the second half of the project, when the corresponding tools are ready. The first version of the tools for the steady state calculations should be ready in March 2019, hence the calculations will start afterwards. The innovative tools allowing for the short-time kinetics calculations will be ready only before the end of the project and their partial validation is planned accordingly. In the first half of the project the main effort have been oriented to acquisition and completion of the data suitable for the validation.

As planned, the validation of the software for steady state calculations will be done against operational data from the commercial power plants. Data have been obtained from the Czech VVER-1000 power plant, and from German Konvoi plant. Both fundamental fuel geometries – square and hexagonal – will be covered. Currently it is not possible to proceed through multicycle depletion calculations using the Monte Carlo codes. It was thus important to find out the operational data corresponding to fresh fuel loadings.

This task was relatively easy in case of Czech VVER plant, because the whole core have been loaded with fresh fuel less than 10 years ago and all the operational data are available in digital format. The data have been collected to th e “experimental benchmark” report, which contains reactor and fuel description and description of the first fuel loading and subsequent reloads. Provided data allow the model preparation for the system of coupled codes. The operational data contain the operational history of 4 fuel cycles (power, control rods position, boric acid concentration, etc.), the power distribution data (measured SPND currents) and the start-up test data (all rods out hot zero power critical states, measurements of control rod worth, reactivity coefficient measurements). The collected data are sufficient for the intended purpose. The model preparation is currently in good progress. The models for the stand-alone codes have been developed and verified. Fig. 9 shows the comparison of the power distribution between the Serpent2 and core monitoring data for the fresh core state at low reactor power. The available data will be converted later into the model for the coupled codes.

In case of PWR Konvoi plant the situation is more complicated, because the reactor have been started with fresh fuel loadings at time when the digital data acquisition systems were not widely available. It have not been an easy task to find out the archived data (see an example in Fig. 10) and convert them to the format suitable for the foreseen application. The data have been nevertheless successfully collected to the form of the “experimental benchmark” report, so the computational models can be build.

Fig. 9: Comparison of the Serpent2 core model prediction to the core monitoring system data: power distribution at the beginning of fresh fuel cycle.

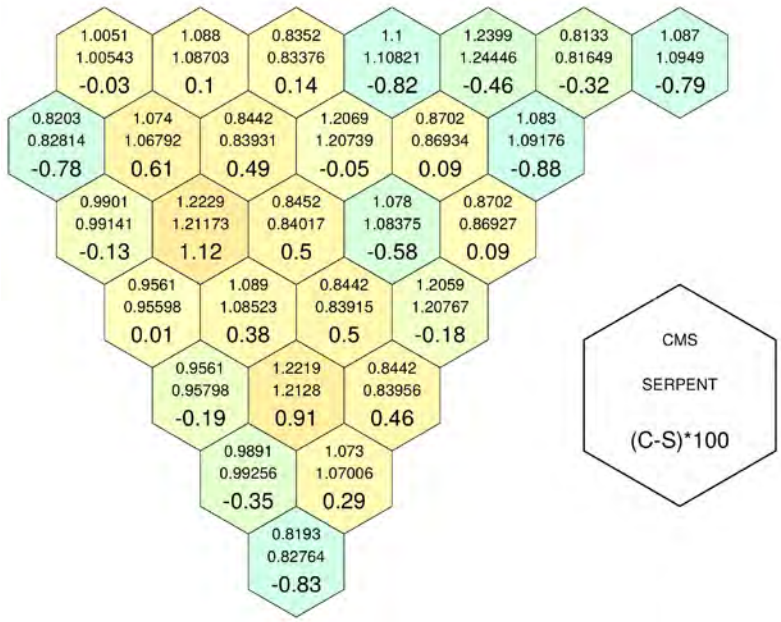
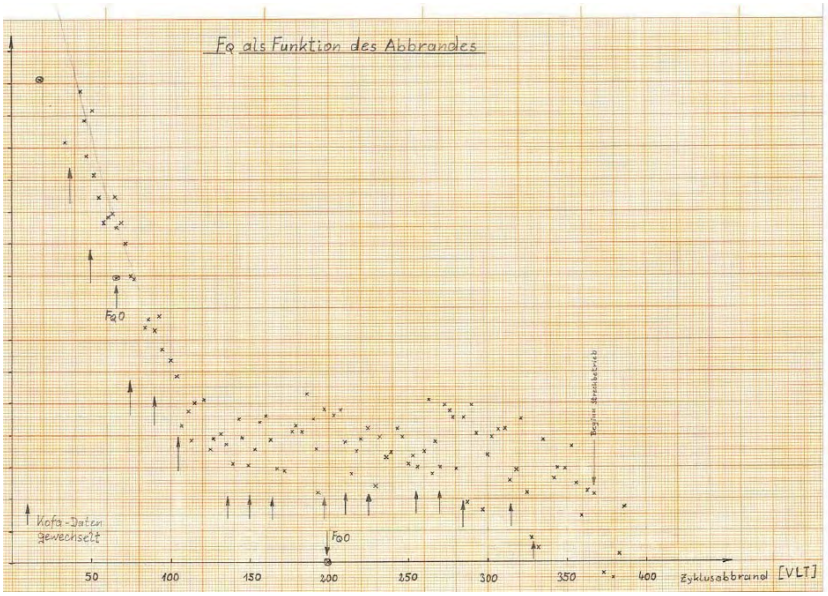


Fig. 10: Example of the format of available operational data for depletion from the Konvoi plant.



4. Progress in coupling codes

For realistic reactor core calculations the effects of the temperature distribution in the core at full power operation must be t aken into account, as well as thermo-mechanical effects due to temperature and pressure. To that end the Monte Carlo (MC) code for neutron transport must be coupled with a thermal-hydraulics (TH) code and a t hermo-mechanical (TM) code. This can be achieved by internal and external coupling.

In the internal coupling the main routines of the TH and TM code are organised as a subroutine of the MC code. The combined codes are executed in an iterative way and the TH and TM parts are

called after the neutronics simulation to update the temperature field and mechanical state, after which the iteration process is repeated until convergence. The internal coupling has the advantage of sharing memory of the MC, TH and TM parts of the integrated code, which allows direct access to relevant data produced in one part and needed in another part of the code.

In the external coupling the MC, TH and TM codes are used as separate codes and a supervisor program at the computer platform must take care of calling the codes in the right order for iteration until convergence. Data exchange from one code to the other must take place by writing intermediate files by one code and reading them as input to the other code.

For running externally coupled codes the Salomé computer platform is adopted. Salomé is open-source software that provides a generic pre- and post-processing platform for numerical simulation. It facilitates integration of new components using ICoCo (Interface for Code Coupling).

In this chapter the progress in integration of the Monte Carlo codes Serpent2, TRIPOLI-4® and Monk in the Salomé platform is shown, as well as the coupling in Salomé with the thermal-hydraulics code SubChanFlow (SCF) and the thermo-mechanics code Transuranus (TU).

Serpent2 has been successfully coupled with SCF both using the ICoCo coupling approach on the Salomé platform and directly using an internal master-slave approach. This allows the user to combine the high-fidelity Monte Carlo neutron (and photon) transport solution of Serpent2 with the pin-by-pin level thermal-hydraulics and fuel behaviour solution of SCF. The capabilities have been demonstrated in steady state and depletion calculations.

The multi-physics interface of Serpent2 has been updated to allow the easy modelling of pin-by-pin distributions in full-level geometries, where the assembly and pin lattices form a two-level nested lattice geometry. The internal source code level coupling interface in Serpent2 has been updated to allow a less invasive, and thus more easily maintainable, source code level coupling to other solvers (SCF and TU).

4.1 Coupling scheme for codes integrated in the Salomé platform

The coupling between neutron transport, fuel thermo-mechanics and thermal-hydraulics is key for the description of fundamental reactivity feedback mechanisms such as the Doppler and moderator effects, which are the main counter-reactions responsible for reactor stability. For the McSAFE project, it has been decided that all the simulation codes should be coupled via the Salomé platform [5], jointly developed by CEA, EdF and OpenCascade. Data exchanges between the simulation codes within Salomé are performed by using the MED-coupling library and the ICoCo (Interface for Code Coupling) API, developed by CEA for the NURISP FP7 project (NUclear Reactor Integrated Simulation Project, 2009-2012).

For this purpose, TRIPOLI-4® has been endowed with an ICoCo API for the exchange of power, temperature and density fields, in view of the coupling with thermal hydraulics. This new interface can now be controlled by a C++ supervisor, which allows performing efficient coupling with other simulation codes.

A major challenge consists in reconciling the geometrical modeling of the Monte Carlo code, which is based on macroscopic volumes, with the one of the coupled codes, which is typically based on meshes. We pragmatically tackle this issue by ensuring that the geometry used by TRIPOLI-4® for the simulated configurations is exactly superposed to the (Cartesian) mesh of the coupled thermal hydraulics code.

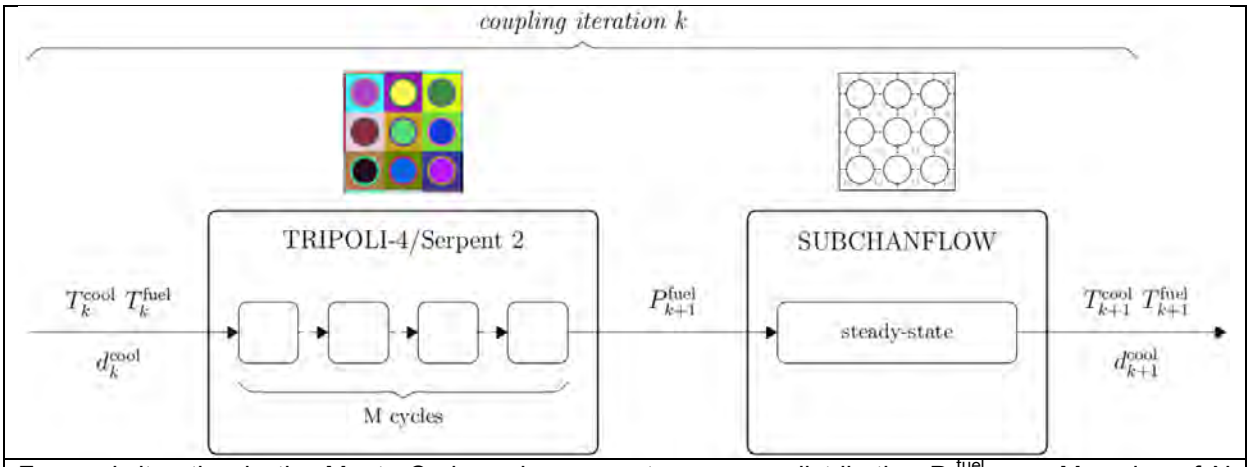
In the context of the McSAFE project, the native (distributed) parallel simulation scheme of TRIPOLI-4® has been extended in order to accommodate the dispatch of updated temperatures and densities across the processors.

As a preliminary test for the newly developed multi-physics interface of TRIPOLI-4®, we have performed a coupled simulation of a fuel assembly taken from the TMI-1 benchmark model in stationary conditions. This kind of simulations is also a prerequisite to determine the initial conditions of time-dependent neutron transport with fuel and moderator feedbacks.

For the purpose of these simulations, we have selected the sub-channel code SUBCHANFLOW (SCF) [8], developed at KIT (Karlsruhe Institute of Technology) and based on the COBRA family. Further details of the integration of SCF into the Salomé platform are given in Sect. 4.2.

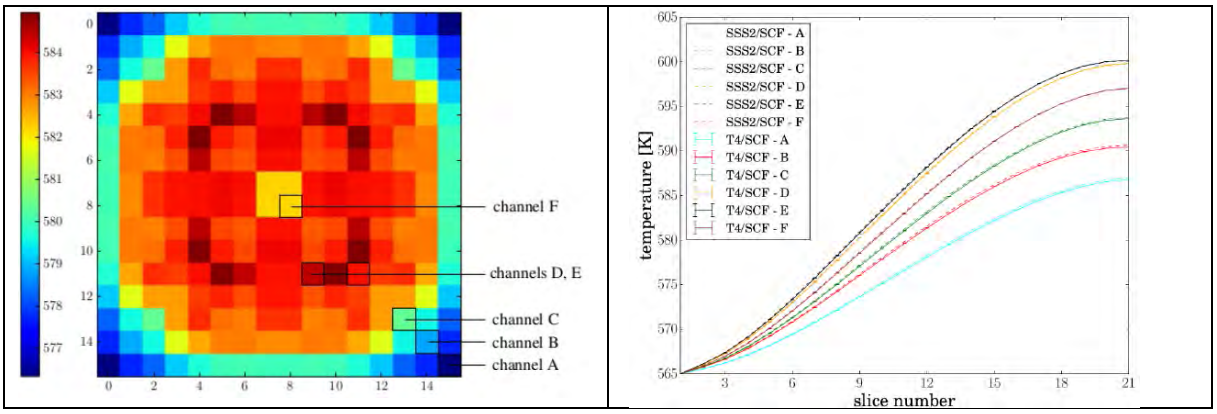
For the boundary conditions, the temperature at the inlet and the pressure at the outlet must be provided. The sub-channel is divided into axial slices and the system is solved with an implicit scheme. In order to compute the fuel rod temperature, each axial slice of each fuel rod is divided into radial rings and the heat diffusion equation is solved with a finite-volume method. We consider here the volume-averaged temperature over these radial nodes.

Fig. 11. Architecture of the TRIPOLI® coupling scheme.



For each iteration k, the Monte-Carlo codes compute a power distribution P_k^{fuel} over M cycles of N particles. This distribution is then transferred to SCF, which computes the corresponding steady-state. The updated temperatures in the fuel rods and the coolant as well as the new densities in the coolant are transferred to the Monte-Carlo codes.

Fig. 12. Coupled simulation of a fuel assembly taken from the TMI-1 benchmark model in stationary conditions with thermal hydraulics feedbacks. Left: coolant temperature map (in Kelvin) obtained with TRIPOLI-4®/SCF, averaged across all axial slices. Right: coolant temperatures obtained with TRIPOLI-4®/SCF (solid lines) and Serpent2/SCF calculations (dashed lines) in 6 channels, at the inlet of each slice. Slice number 1 is located at the bottom of the assembly



The coupled simulation scheme for TRIPOLI-4® and SCF is depicted in Fig. 11. Some of the simulation findings for the TMI-1 benchmark model in stationary conditions are illustrated in Fig. 12, where we have compared the TRIPOLI-4®/SCF results to those obtained by KIT and VTT by coupling Serpent2 to SCF on the same model. An excellent agreement has been found between the two coupling schemes.

4.2 Integration of SubChanFlow into the Salomé platform through the ICoCo interface

The SubChanFlow (SCF) main source code is written in Fortran90 and therefore an interface to C++ is necessary according to the methodology defined for McSAFE. This was done according to recommended practices for Fortran90, this is by using the iso_c_binding module for mapping of data types. The Fortran90 source code is compiled into a dynamic library (subchanflow.so) that provides the capabilities needed by the ICoCo interface.

The interface generated is suitable for coupling with neutronic (deterministic and Monte Carlo), fuel behaviour and system codes. The execution control, mesh support and feedback variables are prepared for coupling with these three kinds of programs. Since the focus in McSAFE is the coupling of SCF with fuel behaviour (TU) and Monte Carlo (Serpent2, TRIPOLI, Monk) codes, the emphasis will be made for this two cases, and the interaction with system codes is out of the scope of this document.

The first step for the implementation of SCF as an ICoCo module is the restructuring of the code as a library that provides capabilities to perform a simulation in a flexible manner and to exchange feedback variables at runtime, to ultimately use this features to develop coupled simulations. This library, written in Fortran90, is wrapped into a shared object (.so in Linux) and called from C++ to perform the actual work exposed by the ICoCo interface.

The relation of this library to the ICoCo interface is essentially one to one, meaning that for each ICoCo class method there is a Fortran90 subroutine that does the work. For this reason, the functions of this library will not be detailed, as the idea of using ICoCo is to have a standardised C++ interface for each code. From the point of view of the users and of the developers of supervisor programs, it is important to have a precise description of the behaviour of the ICoCo

methods. For SubChanFlow developers, special emphasis is put in documenting the source code to facilitate its understanding.

The geometrical model in SCF consists of a set of channels and a set of fuel rods. Channels are connected to each other using a graph-like structure that specifies neighbourhood relations consisting on a pair of channels, a distance between their centroids and a gap width used for the lateral flow calculation. Rods are connected to channels with another graph that describes the fraction of the power of each rod that goes to a set of channels. Furthermore, each channel is not characterised with a specific shape, but the flow area and the wetted and heated perimeters are given instead. This representation is used both to build the model and in the numerical solver, and no actual mesh is used anywhere in the code. Regarding the feedback fields, fuel variables such as power and Doppler temperature are associated to rods, whereas coolant variables such as coolant temperature and density correspond to channels.

Since the ICoCo specification prescribes the MED format for meshes and fields used in feedback exchanges between codes, a problem arises when trying to associate a mesh structure to the fields calculated by SCF. Two types of meshes need to be created: 3D meshes for the fields spanning the whole volume of the core and 2D meshes for boundary conditions at core inlet and outlet. In previous versions of the SCF NURESIM component, assumptions were made about the graph-like geometry used internally and a mesh was built using additional information provided by the user, such as the type of geometry and some key dimensions. This approach only works for specific cases, and its generalisation and extension to new cases, in particular to pin-by-pin geometries, is cumbersome.

Therefore, a different strategy was adopted for the new ICoCo module. Instead of trying to build the MED meshes from the incomplete information in SCF, now they have to be provided by the user. The meshes can be built with a pre-processor or can come from another program that uses the MED format, as long as its cell numbering is consistent with the channel indexes in the SCF model. The convention chosen for the mapping of cells and channels in 3D meshes is given by

$$i_{cell} = (i_{channel} - 1) + (j - 1)N_{channels},$$

where i_{cell} is the (0-based) index of a cell in the MED mesh, $i_{channel}$ is the (1-based) number of the corresponding channel, j is the (1-based) axial level starting from the bottom of the core and $N_{channels}$ is the total number of channels. Hence, the cell indexing is the same as the channel numbering for the first axial level, and for successive levels the indexing is shifted by the number of channels, following a common strategy to convert 2D to 1D indexing. For 2D meshes the same convention is used, with j fixed to 0, so only a shift between 0-based and 1-based indexing is needed.

One restriction does exist in the current version of the ICoCo interface: in order to use a single mesh for channel and rod variables, the MED input and output is restricted to models where each fuel rod is fully contained in one channel, so that a one-to-one relation between rods and channels exists. In this case, fuel variables such as power can be taken as associated to channels, and they can be set and obtained in the same meshes as coolant variables. This approach is well suited for pin-by-pin simulations, in particular for coupling with Monte Carlo and fuel behaviour codes, and hence is appropriate for the McSAFE project. The extension of the MED feedback exchange to other models is not foreseen, but can be done at the request of the partners.

Along with the ICoCo module of SCF, a pre-processor will be delivered to the partners. This pre-processor can generate the SCF input tables describing the geometry and the associated MED meshes for pin-by-pin Cartesian and hexagonal geometries, including full-core models.

The standard way of coupling SCF with a neutron transport code is to get the power distribution from the neutronic calculation and to feedback the coolant density, coolant temperature and fuel Doppler temperature to the transport calculation.

In the following demonstration, first the thermal-hydraulic state is iterated until convergence. Then, the coolant density, coolant temperature and fuel temperature are set to the neutronics module. With this thermal-hydraulic state the neutronic calculation is performed and the power distribution is transferred to SCF. Finally, the error flag is returned to the supervisor. With this scheme, a time step will be converged according to the criteria of the neutronic code.

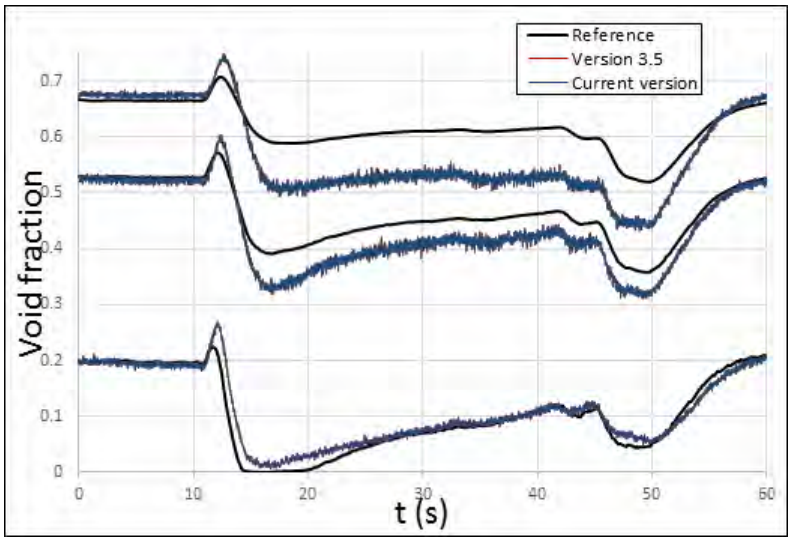
This example shows only one possible scheme for neutronic-thermal-hydraulic coupling. One of the main advantages of the ICoCo interface is that it provides capabilities to implement different coupling schemes in a relatively simple and flexible manner. Therefore, the order of the calculations and feedbacks and the convergence criteria can be easily modified.

In the curent example it is also assumed that the meshes for the neutronic and thermal-hydraulic calculations are identical, so the fields can be exchanged directly. In cases where interpolation is needed, methods in the MED library can be used to interpolate MED fields between MED meshes.

When coupling SCF with a fuel performance code the idea is to replace the simple heat conduction solver for the fuel rods with the solver of the fuel performance code. In this scheme, SCF calculates the coolant conditions taking the cladding outer temperature as boundary condition. In turn, the fuel performance code uses the coolant pressure and temperature and the rod-cladding heat transfer coefficient.

As a tests for the implementation of SCF and its ICoCo interface results are shown in Fig. 13 for a turbine trip case from the BFBT benchmark (case number 4102-01-09), where the results of the version using the ICoCO interface are identical to that from the standalone version of SCF, from which it can be concluded that the current version of SCF and its ICoCo interface are implemented correctly.

Fig. 13. Comparison of results of the current version (blue) and the official version 3.5 b(red) of SCF. The reference results from the benchmark test are shown in black



4.3 Integration of SCF for hexagonal geometry

After a series of difficulties with the original Serpent2-SCF coupling developed at KIT, where we tried to adapt the internal automatic preprocessor for hexagonal systems, it was decided to abandon the original idea and we took a different approach for the internal coupling. The geometry

processing would not be done automatically based on Serpent2 inputs, but rather prepared by the standalone preprocessor before the calculation. The coupled code would then use the generated geometry descriptions and respective mapping files to execute the calculation. The main advantages of this approach are:

1) the coupled code would be much easier to debug, test and maintain. Since the geometry processing is done externally, it can be verified and developed without the need to run the actual coupled calculation, which was a major drawback in testing the former approach (any development resulted in rerunning the same transport calculation over and over, which meant unacceptable turnaround time).

2) the coupled code itself is inherently agnostic of the geometry. The geometry and mapping input decks have the same format for square, hexagonal or whatever other geometries and the coupling does not need any notion of the geometry arrangement.

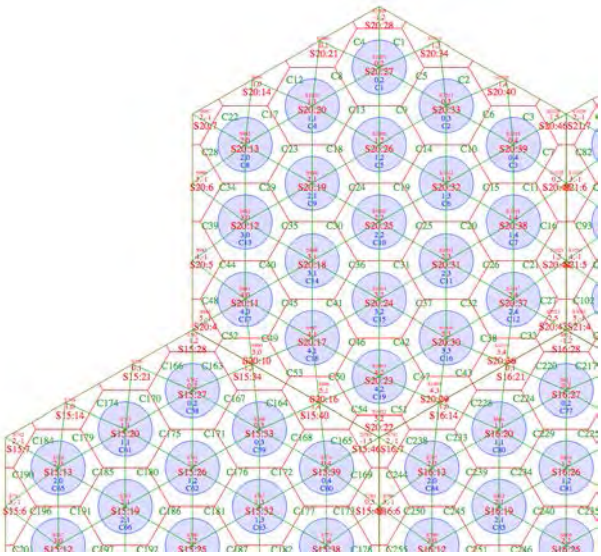
3) geometry features (mainly handling the more complicated hexagonal systems) are much easier to implement in the stand-alone preprocessor than in the tightly coupled Serpent2-SCF-geometry package.

The actual coupling reimplementation of Serpent2-SCF was done by KIT and VTT, while the work on the preprocessor was done by UJV. The initial version generated the SCF geometry descriptions for square and hexagonal cores with both fuel-centered and coolant-centered channels.

To use the new internal coupling it was needed to generate the mapping files between SCF channels and rods to Serpent2 cells for both geometries. This mapping is not always straightforward due to complicated shape of assembly boundary cells. Fig. 14 shows an example of a hexagonal minicore with an overlay of channel and cell meshes.

The resulting mappings provide for each Serpent2 cell the respective SCF pin and list of SCF channels. Those mappings allow transforming the needed power and temperature distribution between the two codes. The mappings are generated in 3D and for arbitrary number of fuel assembly / fuel pin types. In addition to the mapping files the SCF geometries and Serpent2 fuel assembly and core lattices are generated as well.

Fig. 14. Example of mapping for a hexagonal minicore



The preprocessor is designed as a C++ program with a Python frontend. A simple Python script is used as an input file, which allows for great flexibility in the fuel assembly and core definitions. Should the need arise it would be easy to implement other means of case definition (such as text input files) but for the development phase it is more useful to have better flexibility instead of higher comfort for the end user.

The new approach to internal coupling and the extended preprocessor capabilities fulfill one of the project goals of having a second coupling implementation besides the ICoCo coupling.

4.4 Benchmarking of results and HPC scalability

In order to test various forms of the integration of the coupled Serpent2-SCF codes, the VERA benchmark is adapoted. This benchmark consists of a series of calculations for a standard Westinghouse 17x17 PWR design. In this benchmark diverse FA configurations are analysed in terms of burnable poisons, control rod configurations and thermal-hydraulic parameters. For the scope of this work, only 2D and 3D cases of a single FA are developed. Thus, reactivity and power profiles comparisons are developed for such 2D and 3D cases against reported data.

As far as no reference results are provided within the benchmark specification for this case, the reference was collected from reported results from similar projects. In the framework of current task challenges, the following approaches are investigated:

a. Internal coupling with master-slave approach (namely Serpent2-SCF).

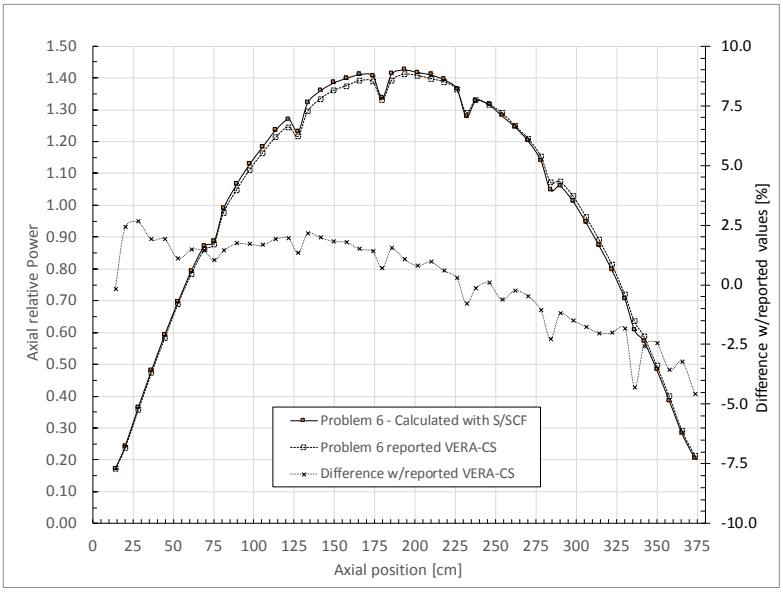
b. External coupling (i.e. Serpent2 + SCF)

c. Supervised ICoCo coupling (externally supervised internal coupling).

A detailed analysis is made and compared with high-quality results with respect to reactivity, axial power profiles, axial coolant temperatures (by channel), axial pin powers (by rod), integrated pin relative power and outlet coolant temperatures (by subchannel),

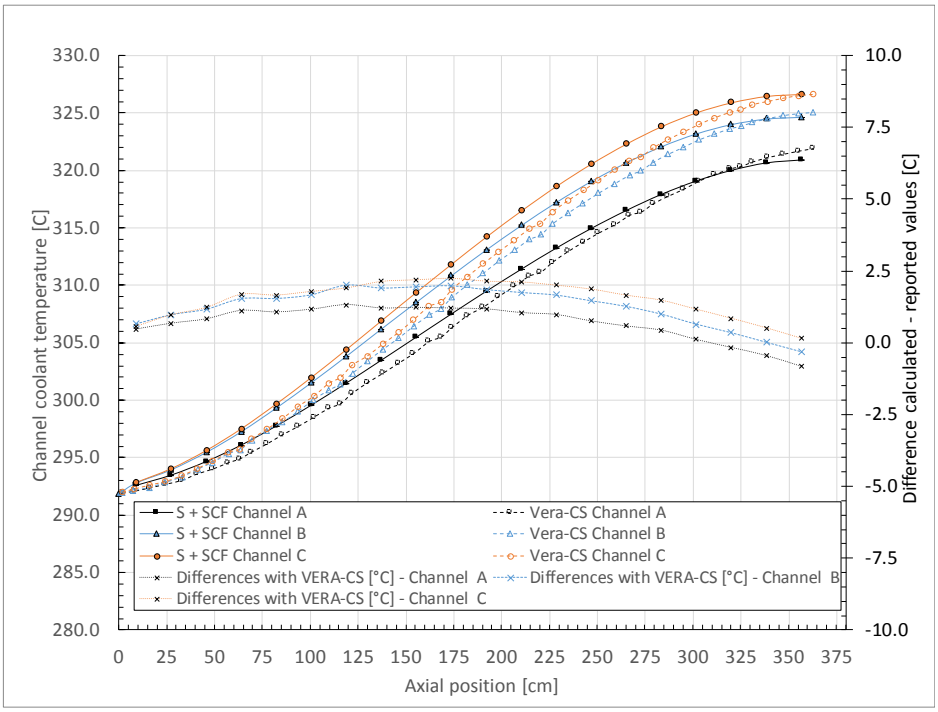
As an example Fig. 15 shows the axial behaviour of the integrated power for the new internal coupling approach, where fuel volume average temperatures are considered.

Fig. 15. Serpent2-SCF axial power behaviour in comparison with reported results



The axial coolant temperatures are compared in Fig. 16 with reported results for a few coolant channels.

Fig. 16. Serpent2-SCF axial coolant temperature profiles



The scalability of the S+SCF internal master-slave coupling was analised for problem #6 using a HPC architecture. For such purpose, the hybrid MPI/OpenMP Serpent2/SCF internally coupled executable was considered (where the slave SCF is run internally in sequential mode).

It should be noted that the stochastic approach implies that the number of neutronic-TH iterations might differ, thus the results are set here to 5 iterations in order to calculate the scalability. The main results are then presented in terms of normalised time (to 1 node of 20 processors) and its inverse (basically the normalised scalability), where a total number of 2.4 10^8 active histories were considered for all cases.

To obtain results for external Serpent2 + SCF coupling, a Python script was considered. In this case, the average TH fields by rod reported by SCF were used to feedback the fuel and coolant parameters in Serpent2 regular IFC file.

Several specific Python3 functions were developed to handle the TH fields and p ower by pin mapping between Serpent2 and SCF, which are available for further couplings. This mapping was developed considering a power detector instead of IFC based power output to improve the control of results statistics. In addition, the communication files were used in Serpent2 as flow control. The reader should note that for this case the management of all external files represented a negligible CPU time.

In addition, to improve global convergence, a dummy cosine axial profile was used as first iteration in SCF. Then the following iterations started for an initial TH solution that approaches to the final expected results. It is important to note that this approach allows a detailed control over convergence aspects and relaxation scheme.

4.5 New internal master-slave coupling development

A new philosophy for the internal master-slave coupling approach was proposed by KIT, where the main aspects are:

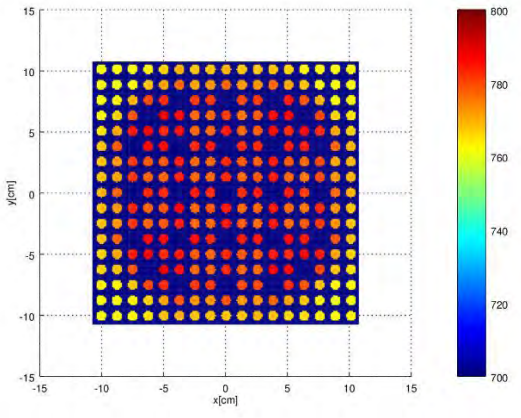
- Improve code maintainability and ease of use by a third party: foresee the impact of code updates and make the implementation of those changes easy in the coupled approach.
- Easy compilation and use by third party: maintain the Serpent2 “easy to compile” philosophy: The idea here is to avoid a complicated compiling process.
- Re-focus the programming approach: the main aspect is to keep both codes (Serpent2 & SCF) well separated, where the interaction is done over some specific routines to be maintained by both developers.
- Make use of main capabilities developed and insights from the ICoCo approach: during the ICoCo implementation inside the NURESIM platform, a high-level access C routines library was developed for SCF to build the ICoCo component. As a result, all capabilities can be used from main C coupling routines just linking this shared C library at compiling time.
- Do not alter Serpent2 code flow: in order to ease implementation of further capability upgrades (such as transient and depletion capabilities), the main Serpent2 code flow must not be altered.
- Maintain all Serpent2 capabilities: no limitations of Serpent2 modeling should arise from the coupling scheme.
- Incorporate all IFC enhancements developed in Serpent2 recently: Several improvements in IFC (Multi-physics Interface) are available, such as nested regular mesh based interface, the capability to assign multiple materials to a file and the availability of remapping of IFC TH and power with the help of external files. Those capabilities, which add flexibility to the coupling, are to be used in the current approach.
- Use the same SCF preprocessor developed for ICoCo approach: The advanced geometry preprocessor developed in the framework of ICoCo coupling is to be used for the SCF model developments and the internal preprocessor is deprecated.

As far as the SCF models are developed externally and the coupling relies in the updated capabilities of Serpent2 IFC (basically through the use of nested lattices in IFC multi-physics interface definitions) no limitations in geometry are now imposed to the coupled case.

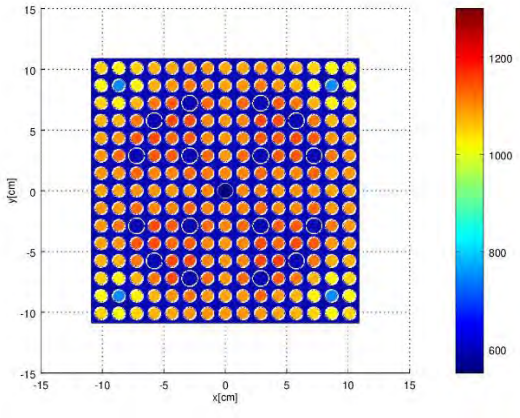
As a result, the newly developed Serpent2+SCF internal master slave-coupling can handle various cases from a single pin to a square or hexagonal FA up to core arrays of FAs using nested square or hexagonal lattices and even abstract models as far as the mapping is correct.

Several verification and testing cases were applied for this coupling. Some of the results of those tests are presented in the Fig. 17 for the VERA (#6) and the TMI FA benchmark test.

Fig. 17. Test cases for the new internal coupling appreoach



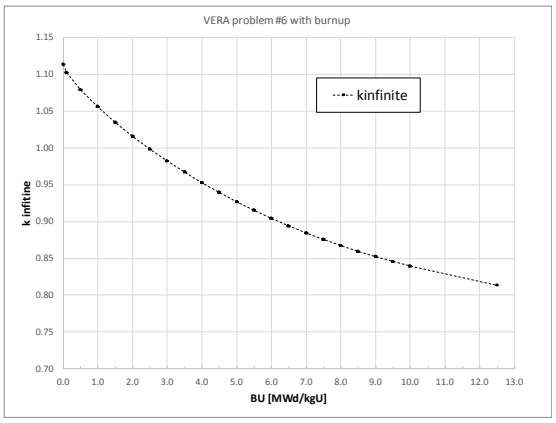
a) VERA Problem 6 – Final temperature iteration



c) TMI FA – Final temperature iteration

Also inherent capabilities of Serpent2 such as depletion and transient calculations are now available without modifications to the code. As an ex ample, the same 3D VERA #6 pr oblem presented inFig. 17a) can be modeled with depletion. Example results for that case are presented in Fig. 18, were depletion is done with the traditional predictor-corrector method, considering equilibrium Xenon.

Fig. 18. Example of a 3D fuel assembly with depletion for the new internal coupling approach



4.6 Benchmarking for hexagonal geometry

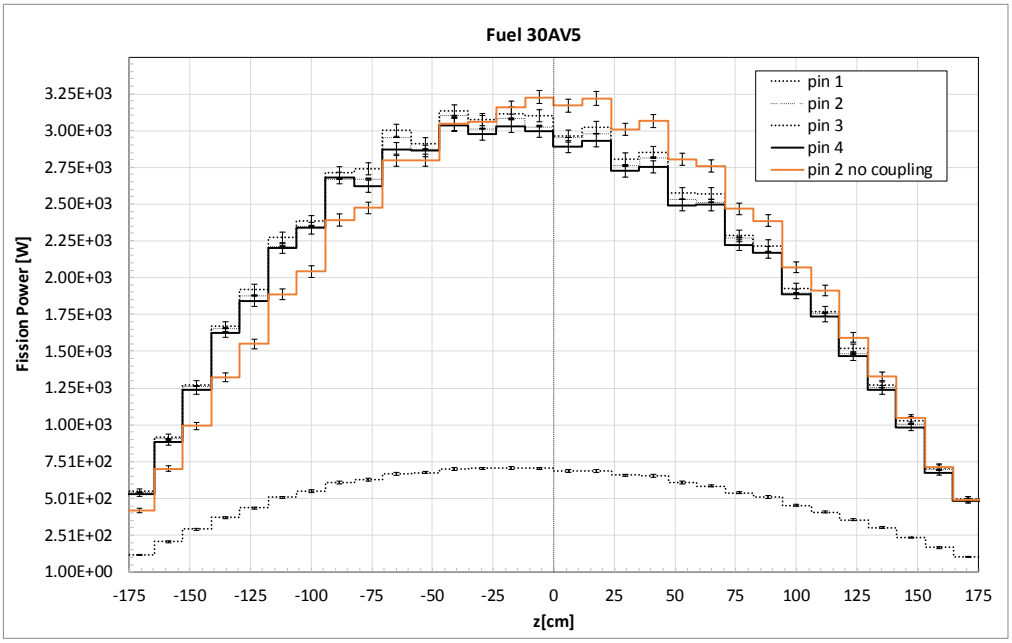
For a benchmark case demonstration a real case benchmark was selected from a VVER-1000, where 2D cases and 3D cases (and core arrays) are proposed and some solutions are available. The selected benchmark provides several FA VVER-1000 TVSA design data to model at 2D and 3D. However, results are only available for 2D cases, it represents a fairly well-documented benchmark case. In addition it provides core level configurations and measurement data.

The FA proposed are several combinations of enrichments and burnable poisons, where the basic geometry (TVSA design with “30AV5” fuel) is maintained.

The SCF model was built with the available SCF preprocessor. A rod-centered model was selected for the sake of simplicity, as far as the mapping to IFC in Serpent2 is clearly understandable.

Regarding convergence aspects, 10^5 histories per cycle and 2000 active cycles were considered (with 50 inactive cycles), which summarizes to 2 10^8 active histories.

Fig. 19 Axial fission power profile for selected pins. The results for the case without feedback is included for pin 2.

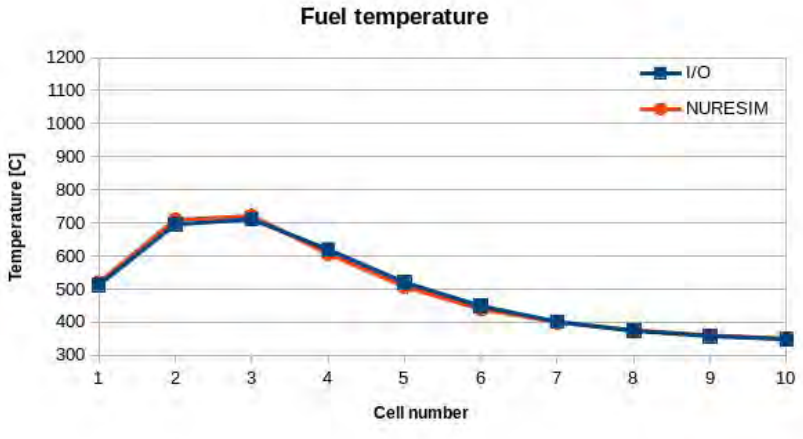


The obtained axial fission power distribution for the selected pins 1 to 4 are presented in Fig. 19, together with a case for pin number 2 obtained without TH feedback. It can be seen from this figure that the feedback pushes the power to the bottom zone of the core, as expected.

4.7 MONK coupling within the Salomé platform

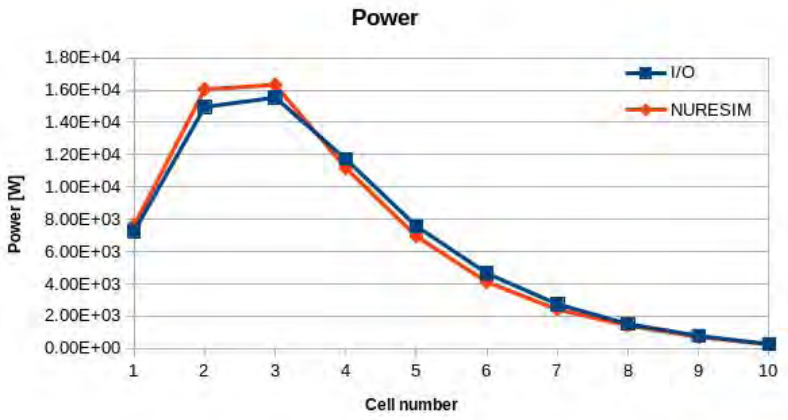
MONK has successfully been converted to a shared library with exposed ICoCo interface methods which have been tested by a Fortran unit test framework and a C++ test program. A MONK Salomé component has been implemented in the form of a C++ class, and this has been tested within the Salomé platform using a Python script. Methods have been implemented to transfer data fields necessary to couple MONK to SCF.

Fig. 20. PWR pincell benchmark fuel temperature profile comparison, 10 axial cells discretisation



A benchmark test case based on a PWR pincell has been used to demonstrate the successful coupling between MONK and SCF integrated within the Salomé platform. The results in Fig. 20 and 21 show a comparison of the two codes, MONK and SCF, coupled via the Salomé framework and via an I/O interface.

Fig. 21 - PW iscretisation



4.8 Integration of TRANSURANUS into the Salomé platform

The Joint Research Cente (JRC) of the Eurpean Union in Karlsruhe maintains, develops and distributes the nuclear fuel rod performance code TRANSURANUS, which describes the thermo-mechanical behaviour of a nuclear fuel rod during normal operation up to design basis accident conditions such as a LOCA or RIA. For the McSAFE project, JRC has prepared the user licence agreements for project participants using the TRANSURANUS code (HZDR, KIT, KTH, UJV, CEZ and VTT), and completed successfully all formalities (e.g. applying for approval from German export authority, BAFA).

HZDR defined in co-operation with CEA, KIT and JRC the specifications for the integration of thermo-mechanical codes into the Salomé platform. On that basis, the TRANSURANUS (TU) fuel performance code was integrated into the platform. During this work, the TU code was modularised in order to be able to interact with the platform.

In close co-operation with VTT and KIT, HZDR developed the coupling of the three codes TU, Serpent2 and SUBCHANFLOW (SCF) within the Salomé platform. For the first time, three codes are interacting under the supervision of the Salomé tools. The interaction of the three codes was tested on a single pin and a one-fuel assembly problem for rectangular geometry. In the next step, the coupling of the three codes was extended to fuel elements with hexagonal cross-section geometry, as well as the coupling of TU with Serpent2 and SCF using the internal multi-physics interface of Serpent2.

5. Progress in time-dependent Monte Carlo calculations

5.1 Development of dynamic Monte Carlo codes

In the McSAFE project the development of dynamic versions of the Monte Carlo codes Serpent2, TRIPOLI-4® and MCNP6 is the cornerstone for realistic calculation of transient behaviour of nuclear reactors.

The time-dependent behaviour of transported particles is taken into account by assigning each particle a “time label” t that is progressively updated in the course of the simulated history on the basis of the particle position and speed. This is already present in the basis versions of the Monte Carlo codes. However, to simulate reactor transient behaviour the generation and decay of delayed neutron precursors play a very important role and this behaviour must also be simulated correctly, which requires extensive modifications of the Monte Carlo codes.

A major scientific challenge for time-dependent Monte Carlo transport simulations is represented by the very different time scales of prompt neutrons and delayed neutron precursors, which demand distinct strategies and variance reduction techniques compared to stationary simulations. Additionally, the delayed neutron fraction (stemming from precursor decay) is very small, which might lead to serious under-prediction biases. To this aim variance reduction techniques based on stratified sampling in time and importance sampling between the neutron and precursor populations are implemented.

Furthermore, during time-dependent simulations, one must prevent the neutron and precursor populations from dying out or growing unbounded. For this purpose population control methods, such as combing [9] and Russian roulette/splitting [10] are implemented.

As an illustration of the new capabilities, Figs. 22 and 23 show time-dependent TRIPOLI-4® simulations of the SPERT III E-core [10] and of the TMI-1 benchmark model with reactivity excursions due to control rod movements. In both cases, the initial conditions correspond to a critical (i.e., equilibrium) regime. The particle source for the kinetic simulations is prepared by running a criticality calculation and appropriately sampling the neutron and precursor populations. Starting from the critical configuration, control rods are extracted and later inserted back into the core. The time scale is such that both neutron and precursors play an important role in determining the response of the simulated systems.

These simulation findings have been compared to the simplified point-kinetics model and numerically verified against the kinetic simulations that have been independently performed by KIT and VTT with the Serpent2 Monte Carlo code in the framework of the McSAFE project. The agreement between Serpent2 and TRIPOLI-4® is excellent.

Fig. 22. Kinetic simulations of the SPERT III E core as a response to control rod movement starting from a critical core at t=0 s; control rods are extracted at t=1 s and are inserted back into the core at t=6 s.
Left: radial cut of the geometry model. Right: evolution of the total neutron flux as a function of time and comparison with point kinetics

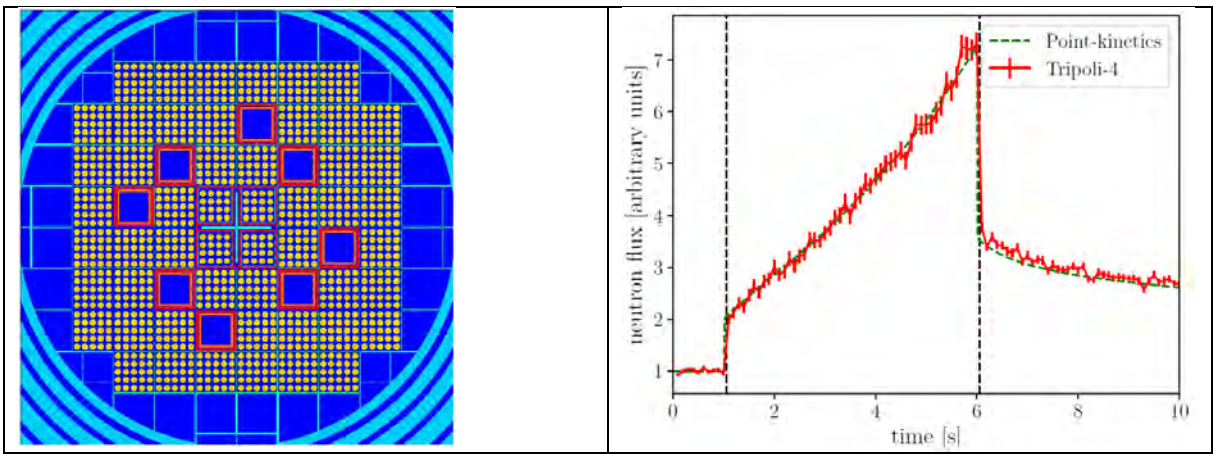
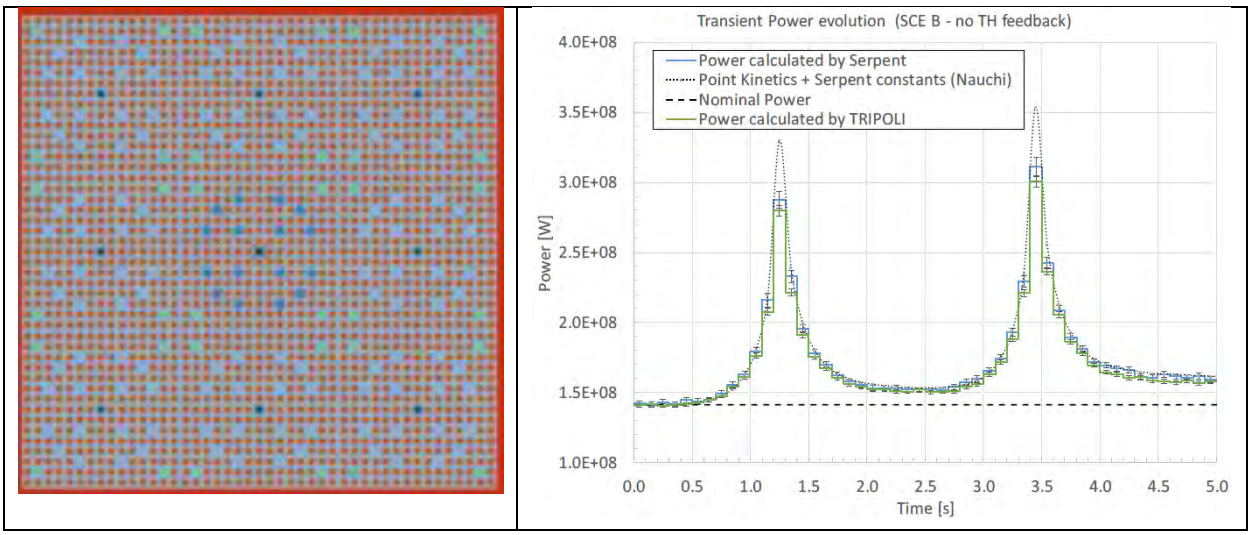


Fig. 23. Kinetic simulations of a mini-core based on the TMI-1 reactor, corresponding to control rod movement in the central assembly: starting from a critical core at t=0 s, control rods are gradually extracted and later inserted back into the core twice. Left: the TRIPOLI-4® model: radial cut. Right: evolution of the reactor power as a function of time and comparison with point kinetics and the analogous Serpent2 simulation.



The time-dependent simulation mode of Serpent2 has been further improved and tested against various experiments with no feedback effects such as control rod insertions of the JSI-TRIGA reactor (in collaboration with Jozef Stefan Institute), control rod insertion in the Giacint facility (in collaboration with Argonne National Laboratory and JIPNR-Sosny) and a control rod ejection transient in SPERT-III E-core (in collaboration with CNL) [11].

In Serpent2 the energy deposition treatment has been improved to yield more accurate spatial distributions of heating by various neutron and photon reactions. Instead of depositing all of the energy locally at fission sites, the energy deposition due to other positive Q-value reactions such as radiative capture as well as direct kinetic energy deposition by neutrons and photons can be accounted for.

For a dynamic version of the MCNP6 code many subroutines of the original code had to be modified and a number of new subroutines added in order to facilitate the generation and decay of delayed neutron precursors and to start a time-dependent calculation from the situation of a critical reactor. Special programming was implemented to allow the continuous movement of a single control rod or a bank of control rods.

Other implementations and improvements of the dynamic MC codes and results of the time-dependent calculations are discussed in the next sections.

5.2 Testing main capabilities of time dependence in Serpent2

Main aspects of Serpent2 capabilities on transient calculations are assessed in order to identify resources requirements and potential bottlenecks. In order to be able to develop a graded approach, ad-hoc calculation tests are used. These cases are based on the available high-quality numerical benchmark for a 15x15 PWR (TMI-1 type fuel assemblies) with Gd_2O_3 as burnable poison (BP), arranged in a 3x3 water reflected minicore.

Several specific transients are simulated. The main intention of each of the cases considered is to check the main capabilities already implemented in Serpent2, considering potential limitations,

foreseen improvements and specific implementation issues that could arise when intended to run in a HPC architecture.

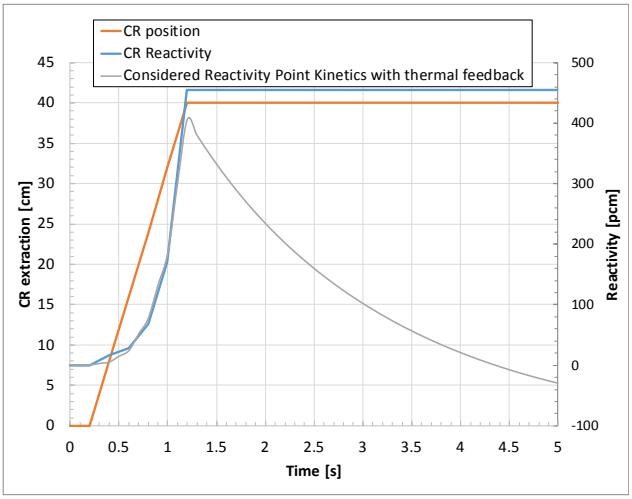
The proposed benchmark considers constant temperatures for the whole model. Regarding the multi-physics goals of the McSAFE project, it is key to be able to model temperature dependences inside the core, with the implemented thermal-hydraulic (TH) feedback.

The initial driver of the transient tests is a control rod (CR) movement A total of 5 control rod scenarios were considered to study the capabilities of Serpent2 for transient calculations. Only for one scenario thermal feedback is taken into account with simplified thermal-hydraulics. A Python script was considered, where the IFC files at FA level were updated only for the temperatures for the UO_2 and UO_2+Gd_2O_3 materials. This update was performed for each time bin. For all scenarios, the main global power is to obtained, together with the pin level power for 10 axial zones.

To check the main behaviour of the results, a comparison is made with an ad-hoc point-kinetics model, which represent the most simple comparison available. Thus the global behaviour is to be compared with the solution of point-kinetics models (obtained using a high-level tool for solving the applicable equation sets), where all kinetic parameters and reactivity transients were obtained from previous criticality calculations with Serpent2.

The main results from transient calculations are presented and compared with the point-kinetics results. As a general rule, a 2-σ statistical uncertainty is shown (i.e. 95% confidence). Unless otherwise stated, the results for the overall power. The control rod movement is depicted in Fig. 24, Together with the calculated reactivity for the point-kinetics calculation.

Fig. 24. Control rod scenario



It can be observed from Figs. 25 and 26 that the expected behaviour is obtained from the Serpent2 calculations. The effect of TH feedback is observed and the shape reproduces the point-kinetics (with feedback) case. The observed differences with respect to point-kinetics for larger times have a cumulative effect that lead to lower temperatures calculated by Serpent2.

For the results presented in Fig. 26 a number of active histories was specifiedin in order to obtain a statistical error of about 3 % (at 1 σ) for the overall system power. The main simulation parameters of these runs are presented in Table 4 for cases A to D.

Fig. 25 Transient without and with simple TH feedback

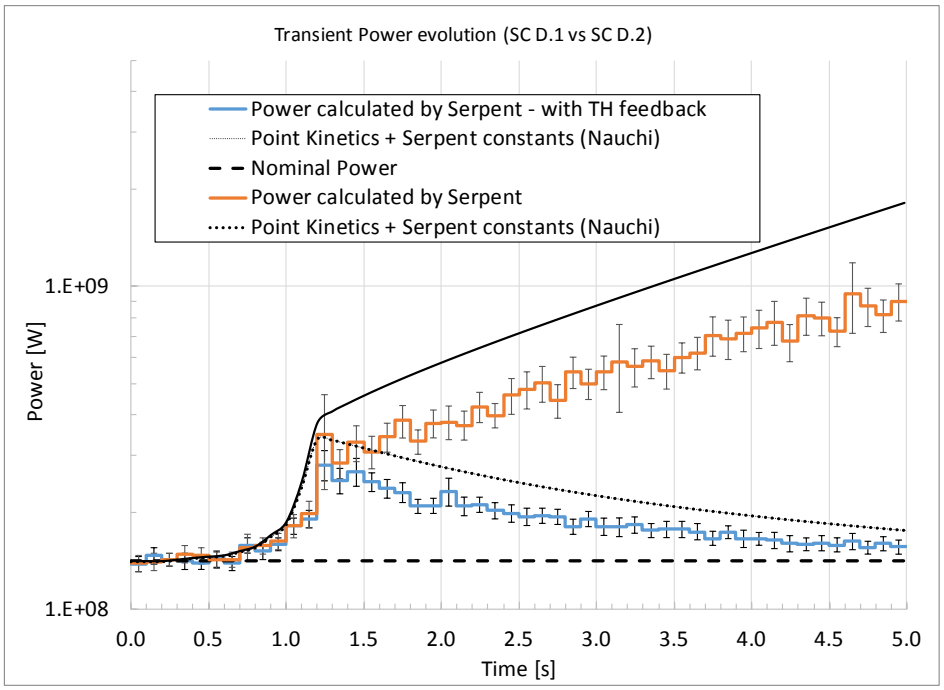


Fig. 26 also shows the detailed power distribution on a pin by pin basis. Tthe statistical noise for this slice for the pin by pin results is in some time bins above 5 % (1σ), which is clearly above the required convergence level required for the foreseen high-fidelity calculations.

Fig. 26. Pin by pin power distribution at z=92 cm below top of fuel and for time interval 4.0-5.0 s

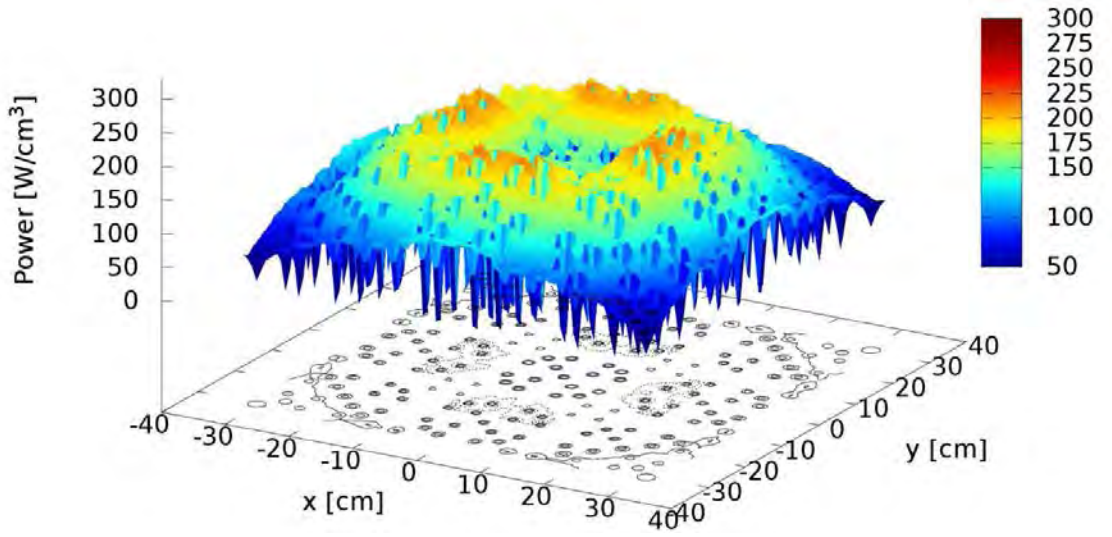


Table 4. Main simulation parameters

Case	Mode	Active histories	Total running time @2.6MHz CPU [h]	Overall power Tally uncertainty – det 1(at 1σ)	
				Average in time bins[%]	Maximum in time bins [%]
A	MPI/OMP	4.00E+05	~4.25E+02	3.1	5.1
B	MPI/OMP	4.00E+05	~4.34E+02	3.6	6.6
C	MPI/OMP	4.00E+05	~4.78E+02	4.9	10.4
D.1	MPI/OMP	5.00E+05	~5.61E+02	5.8	16.3
D.2[a]	OMP	5.00E+05	~7.49E+02	3.0	5.3

[a]Only Serpent2 running time is considered

The main running parameters of these test calculations are presented in Table 4 for various cases. I t should be noted that the amount of calculation time considered is oft he order of a few months to reach this global variance. Clearly, this level of statistical convergence should be improved in order to satisfy the McSAFE goals related to safety parameters calculations in transient cases which are expected to provide high-idelity results.

5.3 Statistics and variance reduction for transient calculations

For the transient in the minicore with thermal-hydraulic feedback calculated with Serpent2 and shown in the previous subsection, the standard deviation in the local power can be obtained. Fig. 27 shows a colour diagram for the relative standard deviation in the local power over the x-y plane for the time interval 4.9 – 5.0 s.

When the analysis is performed by axial slice, the statistical error is as low as 8-10% for the power peak, as presented in Fig. 28.

Fig. 27. Statistical errors in pin power

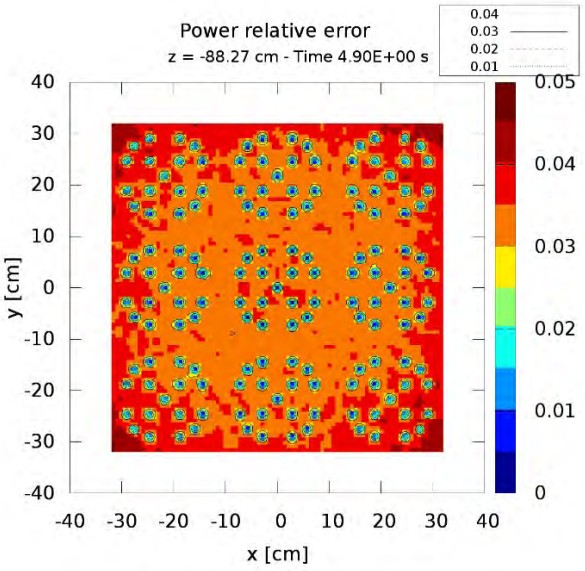
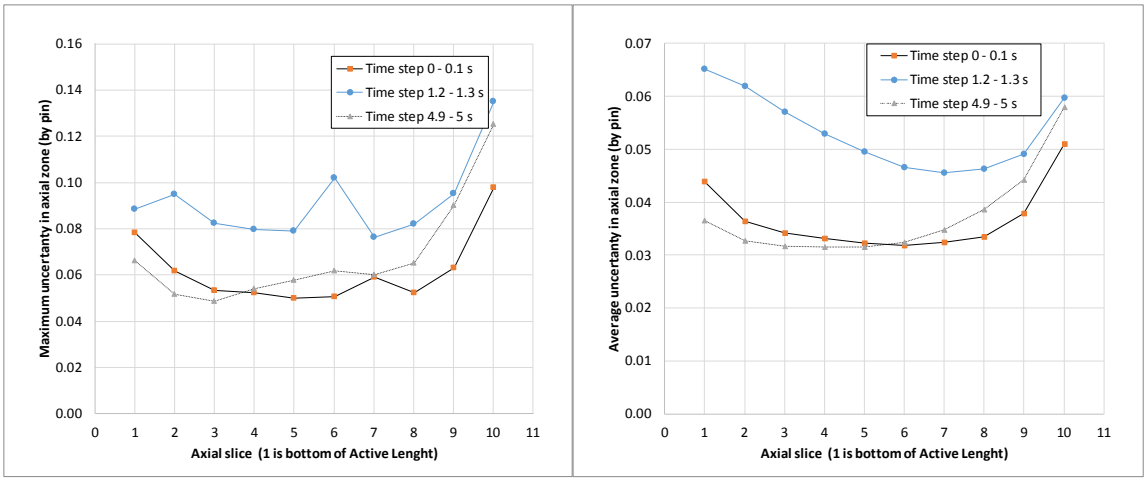


Fig. 28. Maximum and average relative standard deviation in local power



a) Maximum statistical error (1σ) by slice for different time steps

a) Average statistical error (1σ) by slice for different time steps

It is key to note here that these pin by pin results are expected to be used for further safety parameters calculations, thus a higher accuracy will be required in almost all pins. It should be regarded at this point that this accuracy improvement is not necessarily required for all time steps as far as critical values of safety parameters (such as critical heat flux, DNB ratio, etc) are not necessaryly coincident with the spatial and time bins were the statistical error is lower.

Summarising the results presented and discussed above, the following points regarding statistical accuracy and resources requirements should be considered:

1. As stated before, direct extrapolation to obtain statistical convergences as required for the McSAFE goals (i.e. pin by pin results converged up to 0.5 % at 1-σ at pin level) for this simplified case will require calculation times about 0.5 CPU years for a 3x3 FA minicore. The direct extrapolation of the result to a small core to be analyzed inside McSAFE project (for instance. SPERT) should require a similar amount of about 0.5 CPU years. Longet total simulation time or increased number of time bins will rise these requirements. Moreover, direct extrapolation to SMART cores (i.e. about 60 FA) would lead to more than 5 CPU years for this time scope.
2. Use of HPC to solve point 1 is quasi-mandatory, even if some variance reduction techniques are developed.
3. The combination of point 2 with variance reduction techniques is foreseen to be implemented in Serpent2 to tackle the McSAFE objectives. It is important to note here that the definition and implementation of these techniques is not straight-forward and the efficiency is heavily dependent on the problem. With preliminary analysis of the options already developed in other MC codes the implementation of time dependent Weight Windows looks an attractive technique for McSAFE goals. The WW capabilities are already included in Serpent2.1.30 both for the generation and use of such schemes. These capabilities could be then extended to take into account time-dependent variance reduction. Additionally, given the complexity of the problem, the inclusion of some simplified point-kinetics modeling capabilities (even with an external plugin) inside the WW generator should be considered in order to improve the capability to account for a better initial guess for the WW iteration process.

Fortunately, there are many different possibilities for variance reduction in time-dependent Monte Carlo calculations. With only a short indication of the technique, we can list the following methods that are already implemented in one or more of the dynamic Monte Carlo codes or are under development:

- forced decay of precursors per time interval: each precursor decays in each time interval with corresponding modification of the statistical weight of the released neutron
- branchless collisions (BLC): at each collision only one neutron continues with adapted statistical weight
- branchless collisions with analog selection of capture: as before, but capture is first selected according to the capture cross section and the history is terminated if capture is selected
- Russian roulette: neutrons with low statistical weights are terminated or increased in weight to save computer time
- use of importances for selection of precursors and prompt neutrons

For population control over time intervals:

- Russian roulette and splitting at end of each interval to keep the spread in number of particles and in statistical weights within limits

- combing at end of each time interval: redistribution of particles and statistical weights to maintain the original number of particles with equal weights.

The following variance reduction methods are under development or are considered:

- time-dependent fission matrix: estimate and regularly update the fission matrix to guide the particle transport
- time-dependent response matrix: estimate and regularly update the response matrix to guide the particle transport
- time-dependent weight windows: apply Russian roulette and splitting at end of each time interval to keep particle weights between pre-determined weight boundaries.

To reduce computer time to obtain the initial source distribution for a time-dependent calculation in a critical reactor several of the above mentioned techniques are also applied to the stationary criticality case. In addition the following method is applied in one or more Monte Carlo codes:

- Uniform Fission Sites (UFS): increase the number of particles in areas with low neutron flux in combination with adaption of their statistical weight.

General methods to reduce computer time (apart from parallelisation; see next subsection) that are applied in one or more of the Monte Carlo codes are:

- Elimination of most of the standard output of the general-purpose Monte Carlo codes that are not useful for time-dependent calculations with many successive time intervals
- Preparation of a data file with all relevant data to start up a time-dependent calculation in order to eliminate reading in elementary data for system geometry and composition and to eliminate gathering all needed cross section data from basic evaluated data files
- Application of a new tailored tally mesh based on reactor geometry with regularly ordered fuel pins in a fuel assembly and regularly ordered fuel assemblies in a reactor core
- Reorganisation of the logic for parallel execution in order to use the MPI master task also for simulation of particle histories (wich is not always the case, for instance for the standard MCNP code, which uses the master task only for communication with slaves). It may then be useful to reduce the number of histories for the master task somewhat compared to that for the slaves in order for the master to be ready in time for receiving information from the slaves.

Apart from practical implementation and application of these variance reduction methods, a theoretical study is ongoing to determine the variance and standard deviation in power as a function of time using the moments equations with the first moment being the expected value of power and the second moment its variance. The second moment equation is even more complex to solve than estimating the average power, but for very simple systems it can be solved relatively easy for the standard deviation in power as a function of time for a few options of variance reduction like Russian roulette and selection of reaction type (scattering, fission, capture) and number of fission neutrons in case of fission. For a homogeneous infinite system with mono-energetic neutrons it is shown that branchless collisions lead to zero variance for this case.

HZDR has defined a fast-running test case for time-dependent MonteCarlo calculations in order to check new variance reduction techniques. The test case is based on the following assumptions:

- Geometry: 2D reflected PWR fuel assembly
- Perturbation: reduction of boron concentration within one time step
- Feedback: only Doppler feedback

The time-dependent reference solution is calculated by the deterministic multi-group diffusion solver DYN3D. For DYN3D, the few-group cross section library and the reference solution for the initial state are calculated by Serpent2.

As one of the new variance reduction methods, KTH has developed the time-dependent fission matrix approach, which is implemented and tested. The time-dependent fission matrix approach can be used to solve time-dependent reactor physics problems using a set of fission matrices that are pre-computed during a Monte Carlo criticality calculation. We have demonstrated the approach on a set of homogeneous and non-homogeneous perturbations of an initially critical system. The fission matrix approach, however, has an inherent limitation when scaling up to large systems, as the size of the fission matrix scales with a square of the number of nodes. So, applying the method to large systems yields prohibitive memory requirements.

A possible alternative would be a time-dependent version of the response matrix (RM) based solver. The number of responses required scales linearly with the number of nodes, providing an advantage compared to the fission matrix based approach. A steady state RM solver was recently implemented in Serpent2 and used for the fission source convergence acceleration and variance reduction. A time-dependent version of the solver may be useful for reducing the variance in time-dependent Monte Carlo solutions. We are currently working on developing a time-dependent RM solver.

5.4 Parallelisation of time-dependent calculation

Notwithstanding the above-mentioned variance reduction techniques, decreasing the variance by increasing the number of independent particle histories is an obvious choice. However, this may lead to excessive calculation times, unless the Monte Carlo simulations can be executed in parallel on a large number of processors. Although all Monte Carlo codes have the capability to execute a job in parallel, either with MPI or OpenMP or both methods combined, it is still a challenge to obtain high efficiency when using large numbers of processors.

With regard to efficiency of parallel execution, there will be at least two different causes of efficiency decrease when using more processors. One is the time spent to communication between master and slave processor. The more processors used (for MPI parallelisation), the more communication will be needed, forcing other slaves to wait for their own communication. The other reason is the difference in time spent on simulation of the same number of particle histories on different MPI tasks on different processors. Especially for time-dependent calculations the difference in length of a fission chain originating from one neutron (either a prompt neutron starting at the begin of a time interval, or a delayed neutron released from the decay of a precursor during a time interval) can be large. Then it can happen frequently that one MPI tasks has processed all of the particles assigned to it, while another MPI task will continue for some time to complete the simulations. Then all MPI tasks have to wait until each task is finished before the next time interval can be started.

For efficient parallel execution the communication between a slave MPI task and the master task must be minimised. All our Monte Carlo codes are general-purpose codes, implying that programming is present for very diverse situations. This normaly causes all kind of data to be communicated between MPI tasks, which are often not necessary for reactor calculations in which you are only interested in transfering the power distribution as result of tallying. Hence, it will be advantageous to offer the option in the Monte Carlo program that only essential tally information is transfered from a MPI slave tot he master after each time interval.

Moreover, the mechanisms for parallel execution need to be tailored to the specific logic for time-dependent simulations in the Monte Carlo code. In this respect it is important to realise that time-dependent calculations are executed time interval by time interval with each time interval starting with a number of precursors of delayed neutrons and a number of prompt neutrons. As the precursors have decay times normally much longer than a time interval used for estimating the

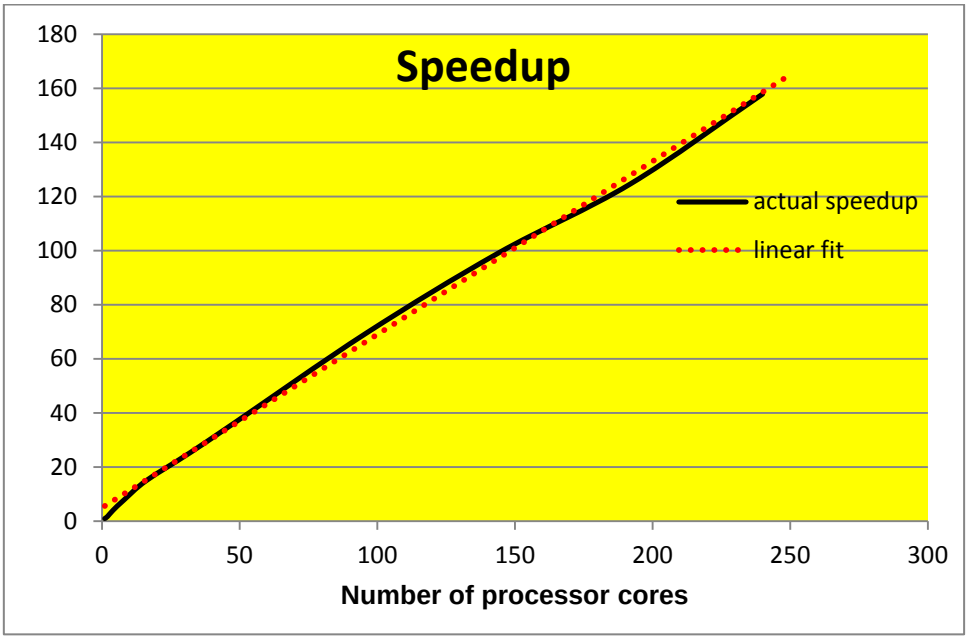
reactor power and the power distribution, the precursors will be alive over many time intervals. Moreover, the collection of precursors in a critical reactor is much larger than the colletion of prompt neutrons at the start of a time interval. This means that the data of these precursors can be transferred from the master MPI task to the slaves only once at the start of the time-dependent calculation, as the position and time of origin of the precursor will not change. During processing of a time interval new precursors will be generated and their origin must be communicated of course, but again these parameters remain the same over many time intervals. Besides the newly generated precursors, neutrons reaching the final time boundary of the time interval must be colleted by the master task, now not only position and time, but also energy, direction and statistical weight. In this way loss of efficiency due to communication can be minimised.

To reduce the waiting time if one task takes (much) loger time for simulating its particle histories then most oft he others, one may apply a form of load balancing in order to distribute the particles more efficiently over the available processors, but this takes more communication and is difficult to realise efficiently.

It would be advantageous to apply parallelisation by OpenMP for the processor cores available on the same computer node. This mechanism selects automatically the next particle in a loop to start its history as soon as a thread using a processor core is available. That means automatic load balancing within the processor core and averaging out different durations of history simulation. Therefore, when using a computer cluster with several or many computer nodes, OpenMP can advantageous be applied for the processor cores on each node, and MPI for all different nodes.

Fig. 29 shows the speedup factor for an increasing number of processor cores obtained with the dynMCNP code. The behaviour is almost linear, although not equal tot he number of processor cores. Important ist hat no actual saturation can be seen with larger number of processor cores, implying that the parallelisation will still be efficient for a much larger number of processor cores.

Fig. 29. Speedup for time-dependent calculation with dynMCNP



5.5 Validation of computer codes and methods

Although validation efforts can only be started when the computer codes to be validated are in more or less final state, which is not yet the case, preparations have been undertaken for a

suitable validation. Experimental data are available for time-dependent experiments with thermal-hydraulic feedback fort he TMI-1 benchmark test and the SPERT IIIE-core experiment.

In sect. 3.10 information is given about the validation steps to be taken. In addition, the validation of the short-time kinetics codes will be done near the end of the project. It will be done against data collected during the SPERT-III E-core experimental program. The data exist already in the needed form. It has yet to be decided which cases among the available experiments will be simulated.

6. Outlook

As shown in this report, important goals have already been obtained in the first half of the McSAFE project. The Monte Carlo codes are basically operational, both for steady-state and dynamic calculations. Advanced algorithms for depletion calculations are implemented in Serpent2. Most of the Monte Carlo codes are internally or externally coupled to a thermal-hydraulic code (SCF) and to a thermo-mechanical code (TU).

However, further development is needed for improved coupling of Monte Carlo codes and the thermal-hydraulic code to limit the computer time for a coupled calculation. Moreover, the algoritm for depletion calculations needs further improvement as well as a solution to deal with large numbers of depletion zones.

With regard to the dynamic calculations, further improvement of variance reduction methods or implementation of new methods is desired to facilitate time-dependent calculations with thermal-hydraulic feedback over many time intervals. At the same time, the parallel execution of coupled calculations must be facilitated for large number of processor cores in order to execute such calculations on a supercomputer.

When all codes and their coupling are optimised, validation calculations will start for the selected cases for which experimental data are available. This will lead to accurate and reliable methods for realistic reactor core design, safety analysis and industry-like reactor applications.

7. References

1. High-Performance Monte Carlo Methods for SAFEty Demostration- From Proof of Concept to realistic Safety Analysis and I ndustry Applications, Grant Agreement ID 755097, https://cordis.europa.eu/project/rcn/211032/factsheet/en
2. McSAFE Digital Newsletter 1, https://www.dropbox.com/s/x0jc5bw55w2sxbl/McSAFE_newsletter.html?dl=0.
3. ChristianChauliac, José-MariaAragonés, DominiqueBestion, Dan GabrielCacuci, NicolasCrouzet, Frank-PeterWeiss, Martin A.Zimmermann, NNURESIM – A European simulation platform for nuclear reactor safety: Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis, *Nucl.Eng.Des.*,**21**, (2011), pp. 3416-3426.
4. NURESIM platform: http://www.nuresafe.eu/docs/NURESIM-flyer_v2.pdf
5. Salomé platform: https://www.salome-platform.org
6. Jaakko Leppänen, Acceleration of fission source convergence in the Serpent 2 Monte Carlo code using a response matrix based solution for the initial source distribution, *Ann. Nucl. En.* ***128*** *(2019), pp. 63-68;* https://doi.org/10.1016/j.anucene.2018.12.044

7. Ignas Mickus, J. Dufek, Optimal neutron population growth in accelerated Monte Carlo criticality calculations, *Ann. Nucl. En.* ***117*** *(2018), pp. 297-304;* https://doi.org/10.1016/j.anucene.2018.03.046
8. U. Imke and V. H. Sanchez, Validation of the subchannel code SUBCHANFLOW using the NUPEC PWR Tests (PSBT), *Science and Technology of Nuclear Installations*, (2012).
9. T.E. Booth, A weight (charge) conserving importance-weighted comb for Monte Carlo, *Proc. Radiation Protection and Shielding Division Topical Meeting*, N. Falmouth, MA, USA (1996).
10. M. Faucher, D. Mancusi and A. Zoia, New kinetic simulation capabilities for Tripoli-4: Methods and applications, *Annals of Nuclear Energy*, 120, 74-88 (2018).
11. A. Levinsky, V. Valtavirta, F.P. Adams, V.N.P. Anghel, Modeloing oft he SPERT transients using Serpent2 with time-dependent capabilities, *Ann. Nucl. En.* ***115*** *(2019), pp. 80-98.*