Control rod drop transient analysis with the coupled parallel code pCTF-PARCSv2.7 [☆]

Enrique Ramos, Jose E. Roman[*]

Dpto. de Sistemas Informáticos y Computación, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain

Agustín Abarca, Rafael Miró

Inst. de Seguridad Industrial, Radiofísica y Medioambiental, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain

Juan A. Bermejo

Iberdrola Ingeniería y Construcción S.A.U, Av. Manoteras, 20. Edificio C, 28050 Madrid, Spain

Abstract

In order to reduce the response time when simulating large reactors in detail, a parallel version of the thermal-hydraulic subchannel code COBRA-TF (CTF) has been developed using the standard Message Passing Interface (MPI). The parallelization is oriented to reactor cells, so it is best suited for models consisting of many cells. The generation of the Jacobian matrix is parallelized, in such a way that each processor is in charge of generating the data associated with a subset of cells. Also, the solution of the linear system of equations is done in parallel, using the PETSc toolkit. With the goal of

[☆] This work has been partially supported by the Universitat Politècnica de València under Projects COBRA_PAR (PAID-05-11-2810) and OpenNUC (PAID-05-12), and by the Spanish Ministerio de Economa y Competitividad under Projects SLEPc-PFE (TIN2013-41049-P) and NUC-MULTPHYS (ENE2012-34585). The authors would like to acknowledge the technical support provided by CNAT and IBERDROLA GENERACIÓN S.A. for the realization of this work.

[*] Corresponding author.

Email addresses: ramos@dsic.upv.es (Enrique Ramos), jroman@dsic.upv.es (Jose E. Roman), aabarca@iqn.upv.es (Agustín Abarca), rmiro@iqn.upv.es (Rafael Miró), jbpi.iberinco@cnat.es (Juan A. Bermejo)

creating a powerful tool to simulate the reactor core behavior during asymmetrical transients, the 3D neutron diffusion code PARCSv2.7 (PARCS) has been coupled with the parallel version of CTF (pCTF) using the Parallel Virtual Machine (PVM) technology. In order to validate the correctness of the parallel coupled code, a control rod drop transient has been simulated comparing the results with the real experimental measures acquired during an NPP real test.

Keywords: sub-channel simulation, parallel computing, PETSc, COBRA-TF, PARCSv2.7

1. Introduction

Detailed fuel element descriptions are increasingly important for Light Water Reactor (LWR) safety analysis, in order to model the fuel elements accurately and obtain the local safety parameters (PCT, Fuel Enthalpy or DNBR). The desired level of detail of the thermal-hydraulic model can be achieved using the so-called subchannel codes, also in the case of coupled local neutronics/thermal-hydraulics evaluations. This setting requires to have an efficient subchannel code in order to obtain reasonable running times for large problems. This is even more important in the case of two-fluid codes such as CTF [1, 2, 3], due to the extended set of complex equations.

In this paper we present a parallelization of CTF so that it can be used in small clusters of computers in order to speedup the computation. In particular, we use a message-passing parallelization (with MPI) following a domain decomposition approach and using parallel linear solvers provided by PETSc. This is similar to the parallelization recently accomplished by Kucukboyaci *et al.* [4], although there are also important differences. Kucukboyaci's code assigns individual fuel assemblies to different MPI processes, and requires both a preprocessing step that produces a CTF input file for each process, as well as a postprocessing step to combine all the files generated by the different processes. In contrast, our parallel code operates directly with the CTF input deck that would be used in a sequential run, and also produces output files equivalent to the sequential run.

The rest of the paper is organized as follows. First we describe the code and its main features, and then we present the approach followed for the parallelization, together with the obtained results in the simulation of control rod drop transient in a three-loop KWU-Siemens Pressurized Water Reactor

(PWR) core.

2. Description of CTF

COBRA-TF (CTF) is an advanced subchannel simulation code, that can be used to analyse reactors of both Pressurized Water (PWR) and Boiling Water (BWR) types. In this section, we provide a brief overview of this code, and interested readers can find further details in [5].

CTF uses a representation of the two-phase flow based on three fields, namely continuous vapor, continuous liquid, and entrained liquid drops in the region of annular flow. The conservation equations for multidimensional flow of the three fields are discretized by means of a semi-implicit finite difference method with donor cell differencing for the convective terms. From these, a set of simultaneous algebraic equations is established, that must be solved for each cell and also for the aggregate computational domain, resulting in the parameter distributions for the different fields.

2.1. Solution scheme

At the beginning of the solution process, estimates of the mass flow rates of the three fields (f_l, f_v, f_e) corresponding to the new time step are computed by solving the momentum equations for each cell, using currently known values for all variables. All explicit terms in the momentum equations are also obtained at this point, which will remain constant for the rest of the time step. The semi-implicit momentum equations are written in terms of the following coefficients: constants representing explicit terms such as momentum efflux and the gravitational force; the explicit portion of the pressure gradient force term; and explicit multiplicative factors for the liquid, vapor and entrained liquid flow rates in the wall and interfacial drag terms.

As a second stage, the code computes tentative velocities that are used in the linearization of the mass and energy equations. A Newton-Raphson iteration is applied to obtain the corrections to the independent variables so that residual errors tend to zero. These residual errors are due to using the new velocities to compute the convective terms in these equations. Applying the Newton-Raphson method a matrix equation is obtained for each cell,

$$[R(x)]\,\{\delta(x)\} = -E, \tag{1}$$

where $[R(x)]$ is the system Jacobian evaluated for the set of independent variables x and made up of analytical derivatives of each equation with respect to the linear variation of independent variables; δ is the solution vector containing these linear variations; and E is the vector of errors.

After all derivatives have been computed, the above system of equations is analytically reduced using Gaussian elimination to get solutions for the independent variables. Void fraction related variables and pressure of the actual cell depend on the pressure of adjacent cells. After reducing the system, an equation of the form

$$\delta P_j = a + \sum_{i=1}^{\mathrm{NCON}} g_i \delta P_i \tag{2}$$

is derived for each cell. Hence, to obtain the pressure variation for each cell, a linear system with as many equations as the number of computational cells needs to be solved. After this, the linear variation of the other independent variables is unfolded.

2.2. Computational tasks

The main computational tasks required in each time step are the following:

1. Update state variables with respect to the current timestep.
2. Compute the Jacobian matrix as well as the right hand side vector (the residual of the non-linear equations) for the Newton-Raphson iteration.
3. Solve the linear system of equations.

Due to the finite-difference discretization scheme, the first two steps involve neighbour-wise computations, that is, when computing the value of a magnitude in a given cell it is necessary to use the current value of the same magnitude in at least one neighbouring cell.

Regarding the third step, it is worth mentioning that the coefficient matrix of the linear system is sparse (it has a small percentage of non-zero entries), so iterative linear solvers are preferred instead of factorization-based solvers.

In order to determine the relative weight of the different computations, the code was profiled with the valgrind tool[1]. The profiling shows that the

[1] `http://valgrind.org`

above three steps (outer iterations) are the most computationally expensive part, with around 75% of the total time. The inner iteration loop (step 3 above) is just about 20% of the total time. From the rest of computations, the most significant ones are the solution of energy equations (about 10%) and the computation of the water-steam properties (about 6%).

3. Structure of the parallel CTF code

We aim at being able to solve large-scale problems with CTF by making use of parallel computing. Via parallelization, several processors can be used to cooperate in the solution of a single problem, thus reducing the computation time as well as increasing the amount of available (aggregate) memory (in a cluster).

Basically, the parallelization consists of splitting the computation in separate subtasks that are assigned to different processors. An effective parallelization of CTF must address two important points: (1) all three stages mentioned in the previous section must be parallelized, not just the linear system solution, and (2) memory must be managed appropriately, splitting the state variables across the different processes. We have considered a coarse-grain parallelism based on a domain decomposition strategy with a message-passing paradigm, in particular with MPI [6].

3.1. Domain decomposition

The domain decomposition approach for the parallelization is appropriate for the type of computations carried out in CTF. We have chosen to decompose the computational domain in the axial dimension, that is, each MPI process is assigned a contiguous chunk of axial levels. Normally the number of axial levels is rather small, so scalability of the parallel code will be quite limited in this dimension. This could be compensated with a second level of parallelism based on multi-thread computation associated to each subdomain (assuming a cluster of multi-core processors is being used), but we have not addressed this issue yet (this would entail shared-memory parallelization with, e.g., OpenMP, as is done in [7]).

Most of the computations in CTF are neighbour-wise, that is, magnitudes depend on values from neighboring cells. Therefore, it is necessary to represent the state variables in such a way that each process has enough memory space to hold all axial levels assigned to it, plus additional space for the *ghost cells*. These ghost cells are extra locations to store values received from neighboring processes. In our case, the ghost cells are two complete axial levels, one to communicate with the process *above* and another one for the process *below*. Whenever necessary during the computation, these ghost points are updated with the values of the corresponding cells stored in the neighbour processes, and this is done with point-to-point communication.

3.2. Linear solver

The development started from a version of CTF that used the SPARSKIT library for the Krylov iterative methods to solve linear systems of equations associated with the Jacobian (this represented a significant improvement over Gaussian elimination and Gauss-Seidel used in the original version). For the parallel version, SPARSKIT has been replaced with PETSc, that provides parallel linear solvers that fit well the domain decomposition paradigm. Also, PETSc is much more flexible in terms of selecting different iterative solvers and preconditioners (this can be done at run time).

PETSc (Portable, Extensible Toolkit for Scientific Computation) [8, 9], is a framework for the parallel solution of partial differential equations. It uses object-oriented programming techniques to encapsulate mathematical algorithms, hiding the complexity of numerical message-passing codes. PETSc is freely available and used in many application areas including nuclear engineering. With PETSc, the application programmer interacts with a few data and algorithmic objects via a generic interface, without knowing the details of the underlying data structures. The three basic abstract data objects are index sets, vectors and matrices. Built on top of these are various classes of solver objects, including linear, non-linear and time-stepping solvers.

For solving linear systems of equations, PETSc provides a variety of iterative methods such as GMRES, that can be combined with different preconditioners such as Jacobi or block Jacobi [10]. A performance study shows that BiCGStab(ℓ) [11] with block Jacobi (with incomplete LU in each subdomain) has a good convergence behaviour, so we have used this combination for the results presented later in this paper.

4. CTF/PARCSv2.7 coupled code

In previous works a fully explicit coupling scheme between CTF and PARCS based on an external data exchange was developed [12]. The coupling algorithm used the GI routines implemented in PARCS and new routines

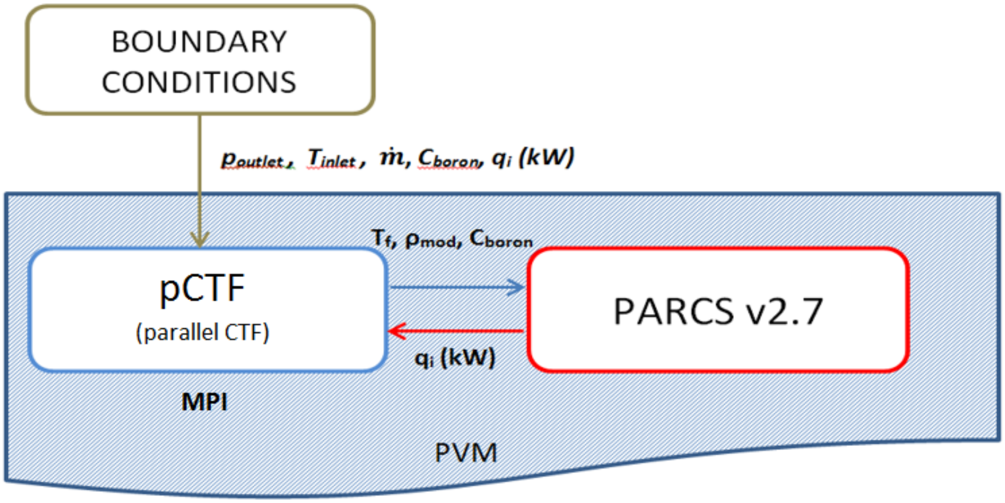

BOUNDARY CONDITIONS
p_{outlet}, T_{inlet}, $\dot{m}$, C_{boron}, q_i (kW)
T_f, ρ_{mod}, C_{boron}
pCTF
(parallel CTF)
PARCS v2.7
q_i (kW)
MPI
PVM

Figure 1: CTF/PARCSv2.7 coupling scheme and data flow.

added to CTF code to make possible the communication and data exchange on a PVM environment [13], as can be seen in the scheme of Figure 1.

Originally, an auto-mapping routine implemented in CTF provided a robust averaging methodology to adapt the data for different neutronic and thermal-hydraulic nodalizations when the number of neutronic and thermal-hydraulic nodes are proportional, most of the cases in which the whole core or a group of fuel bundles are simulated. However in more complete CTF models, like the one developed in this work that includes the plenums of the core, an external file could be necessary to provide the correspondence between thermal-hydraulic and neutronic nodalization *mapping*.

This file, called MAPTAB, has been added to the original coupling scheme and it contains the information about the correspondence between sub-channels and heat structures in the thermal-hydraulic model with the nodes of the neutronic model.

5. Control Rod Drop Transient

The modeled core corresponds to a three-loop KWU-Siemens reactor type. The core has 177 fuel elements, being the number of rods per fuel element equal to 236 with 20 guide tubes. The operating conditions during the test corresponds to Hot Full Power (HFP), where the nominal power and

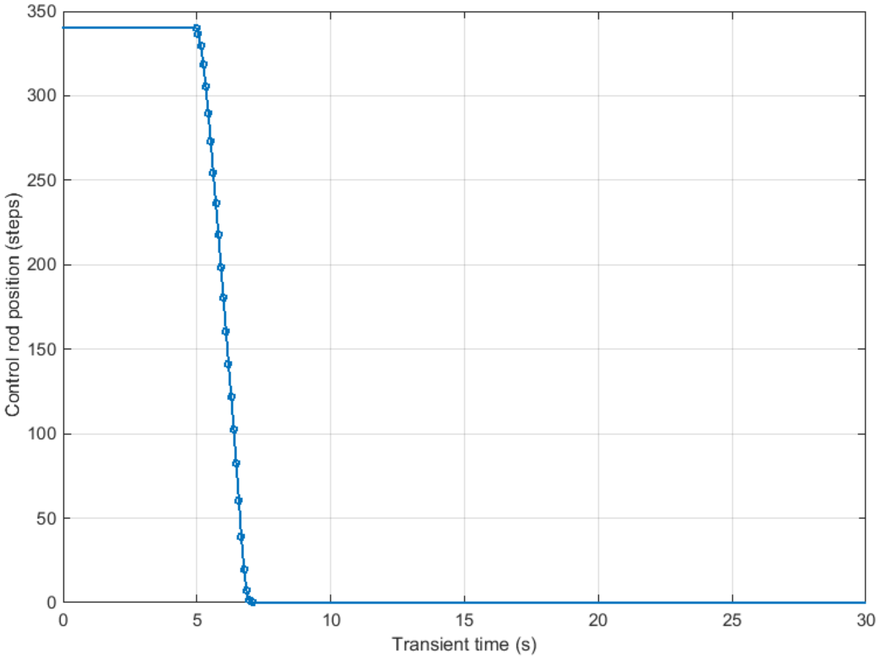


Figure 2: Temporal evolution of the control rod position (in cm.).

core flow are of 3010 MWth and 15605.6 kg/s respectively at the Beginning of Cycle (BOC).

The control rod insertion is defined in the input deck of PARCS. The drop lasts 2.1 seconds, and begins at 5.0 seconds on a 30.0 seconds transient. The first 5.0 seconds are a null transient with the purpose of ensuring a stationary stable conditions at the beginning of the transient. Figure 2 shows the evolution of the control rod insertion during the transient, being 340 cm fully withdrawn and 0 cm fully inserted control rod position.

6. Models and methodology

6.1. Thermal-hydraulic model

The CTF pin by pin model consists of 45313 flow channels, 45312 represents the active core (blue colored nodes in Figure 3) and 1 for the whole core bypass (red colored nodes in Figure 3), axially the domain is divided into 34 nodes equally spaced. The 41772 fuel rods comprising the reactor core are modeled by the same number of thermal structures as well as the 3540 control rod guide tubes represented by their respective unheated structures. In summary, the active core has been represented by a 3D mesh of 45312 x 34 nodes with the thermal structures by which fuel rod and guide tubes are connected to the flow channels. Figure 3 shows the 3D mesh representing the CTF model of the core, in which all the red reflector nodes are represented by a single flow channel in the real CTF model.

6.2. Neutronic model

The neutronic model has been developed for the widely proven neutron diffusion code PARCS v2.7. As can be seen in Figure 4, this model consists in a 3D mesh of 241 x 34 nodes, in which the active core has 177 x 32 nodes. The cross sections tables, that have been provided by Iberdrola Generación S.A., considered 1379 different isotopic compositions according to the fuel assembly design and the burn-up. The active core has 1376 isotopic compositions, while the remaining 3 are assigned to the top, bottom and surrounding reflector nodes respectively. The cross section sets were generated using the SIMTAB methodology, developed in collaboration with Iberdrola Generación S.A. This methodology is fed with data from CASMO4-SIMULATE [14, 15] codes and generates the cross sections as a function of the fuel temperature, moderator density, boron concentration and control rod insertion.

6.3. Thermal-hydraulic to neutronic mapping

The mapping represents the correspondence between the nodalization of the flow channels and heat structures in the thermal-hydraulic domain and the nodes of the neutronic domain. In order to provide this information to the coupled code an external file called MAPTAB is used. The mapping file contains the numerical correspondence between the subchannels and the neutronic nodes, and includes a weighting factor to take into account the volumetric relation between both nodalizations.

For the developed models, each active neutronic node corresponds with 256 thermal-hydraulic flow channels, as shown in Figure 5, so that the weighting factor for each subchannel is 0.0039 (one divided by 256). Since a single flow channel in the thermal-hydraulic model represents the bypass, all the radial reflector nodes of the neutronic model (yellow nodes in Figure 4) have been linked with the bypass channel through the mapping.

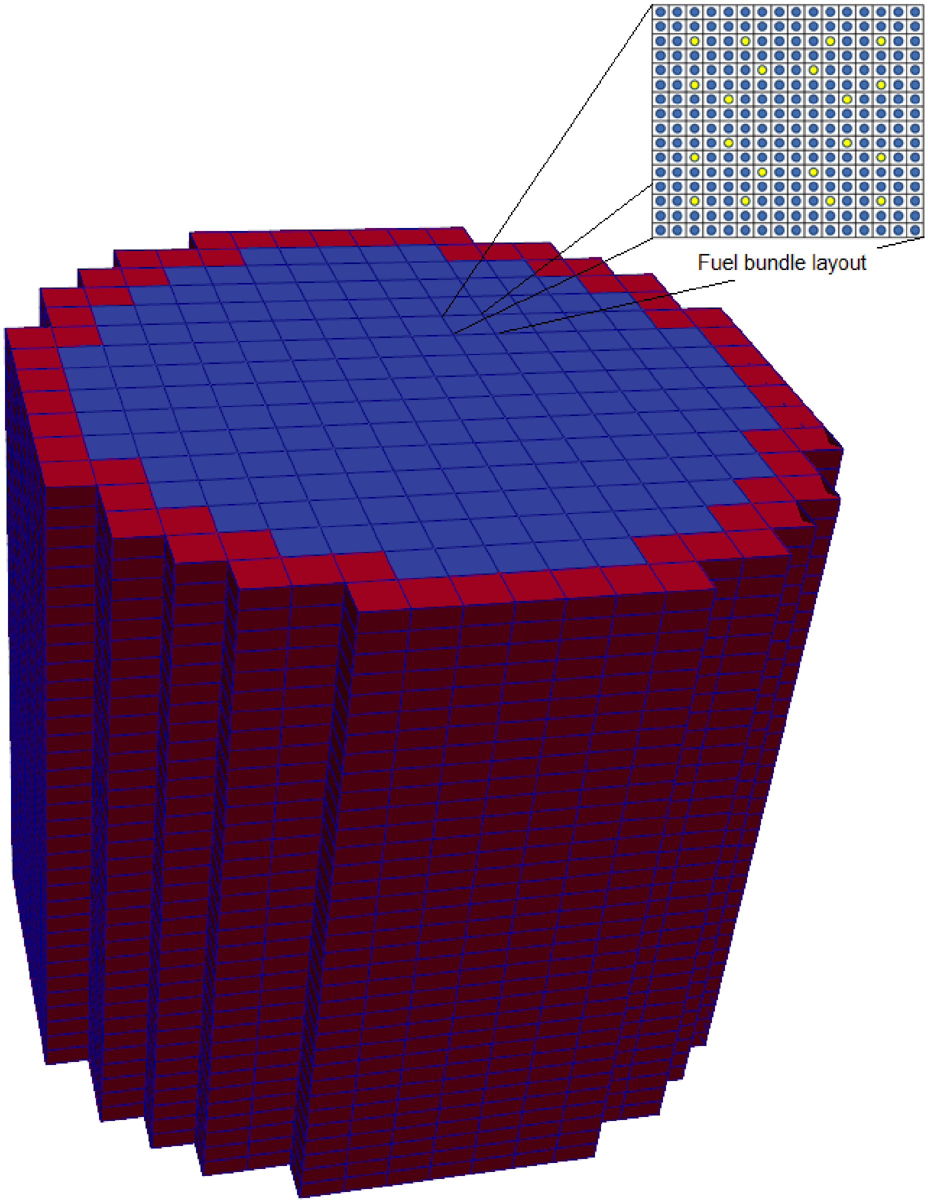


Figure 3: 3D thermal-hydraulic mesh used by CTF.

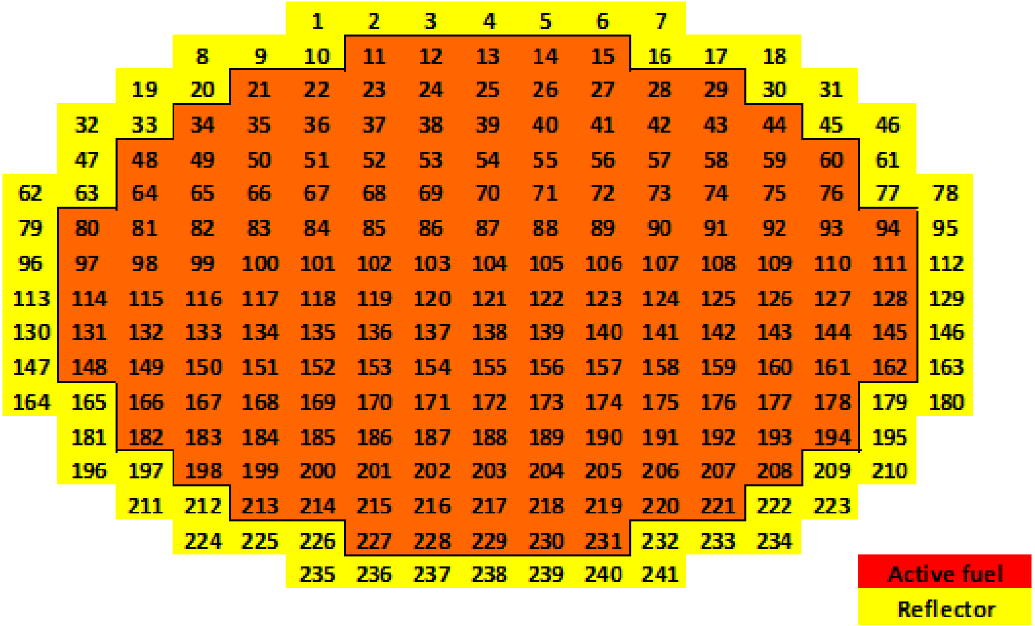
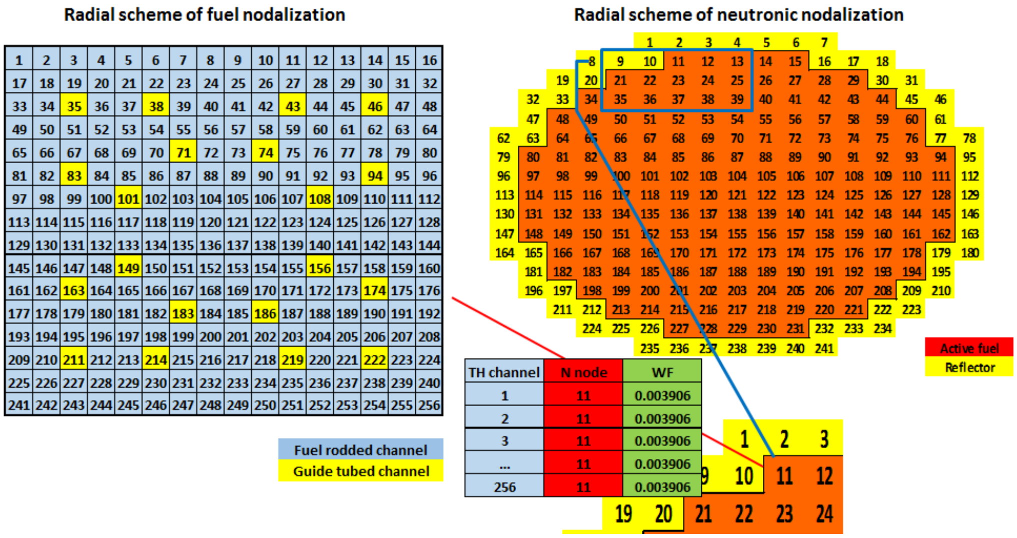


Figure 4: Radial scheme of the neutronic model.


Radial scheme of fuel nodalization
Radial scheme of neutronic nodalization
Fuel rodded channel
Guide tubed channel
Active fuel
Reflector

TH channel	N node	WF
1	11	0.003906
2	11	0.003906
3	11	0.003906
...	11	0.003906
256	11	0.003906

Figure 5: Example of thermal-hydraulic to neutronic mapping of a single fuel assembly.

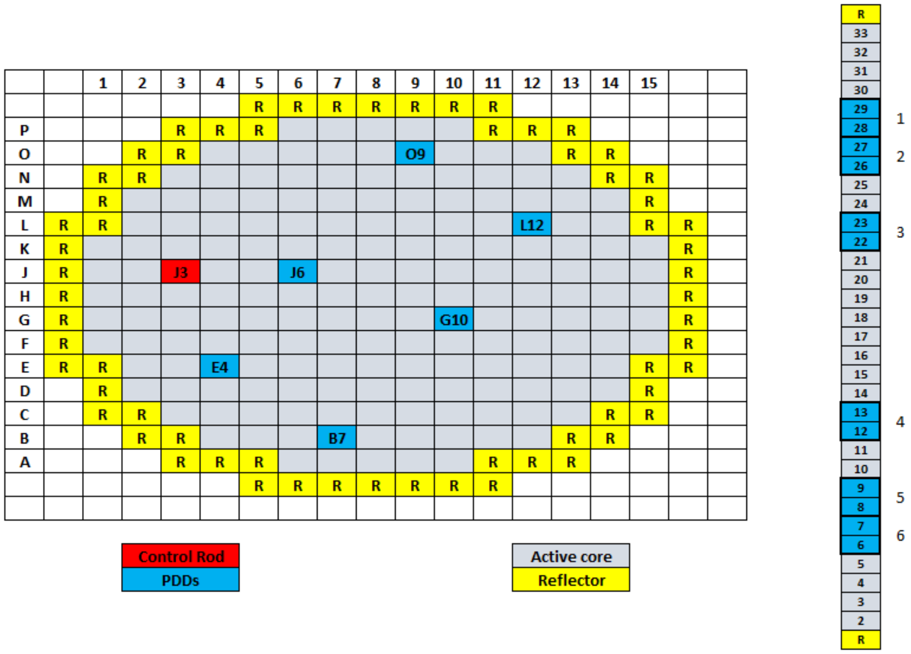


Figure 6: Location of PDDs in the reactor core.

6.4. Power Density Detectors.

In order to obtain the response of the Power Density Detectors (PDDs) using the neutron flux calculated by PARCS, the Equation 3 was used, where F_{ID} is a conversion factor for each detector, it corresponds to the mean power given by the detector at the stationary case, and ϕ is the thermal flux calculated for each detector position.

$$P_{ID} = \frac{\phi}{\phi_{t=0}} \times F_{ID} \tag{3}$$

The real PDDs position inside the reactor core is shown in Figure 6. In the same Figure, the control rod drop trial is shaded in red.

The PDDs radial position corresponds to a single neutronic node, but it is necessary to take the thermal flux of two consecutive axial nodes and multiply them by the proper weighting factor to obtain the detector response. These weighting factors by which the real PDD positions are link up with the axial nodes elevations are shown in Table 1.

Table 1: Axial weighting factors for the PDDs detectors.

Detector axial level	Current Position (node)	Nodes	Axial weighting factor
1	6.6 (in)	29	0.4
	(28.4)	28	0.6
2	8.4 (in)	27	0.6
	(26.6)	26	0.4
3	12.4 (in)	23	0.6
	(22.6)	22	0.4
4	22.4 (in)	23	0.6
	(12.6)	12	0.4
5	26.7 (in)	29	0.3
	(8.3)	8	0.7
6	28.2 (in)	7	0.8
	(6.8)	6	0.2

7. Computational results

The computer system used for the computational experiments is Quasar, a computer cluster consisting of 4 nodes, each of them with AMD Opteron processors with 32 cores running at 2.4 GHz, and 96 Gb of RAM memory per node. The nodes are connected with Gigabit Ethernet.

7.1. Steady State Results

The results of the coupled steady state (HFP conditions at the BOC) are presented in the section. These results are used to verify the correctness of the pCTF/PARCS coupling scheme, the developed models and the cross sections. Besides, the results were the starting point for the final control rod drop transient by using the restart capability of the parallel coupled code.

In order to verify the coupled steady state conditions, the results obtained with the SIMULATE-3 (S3) code are used as a reference. The power profiles and the eigenvalue of the loaded core are the reference neutron kinetic variables to compare the results of related codes.

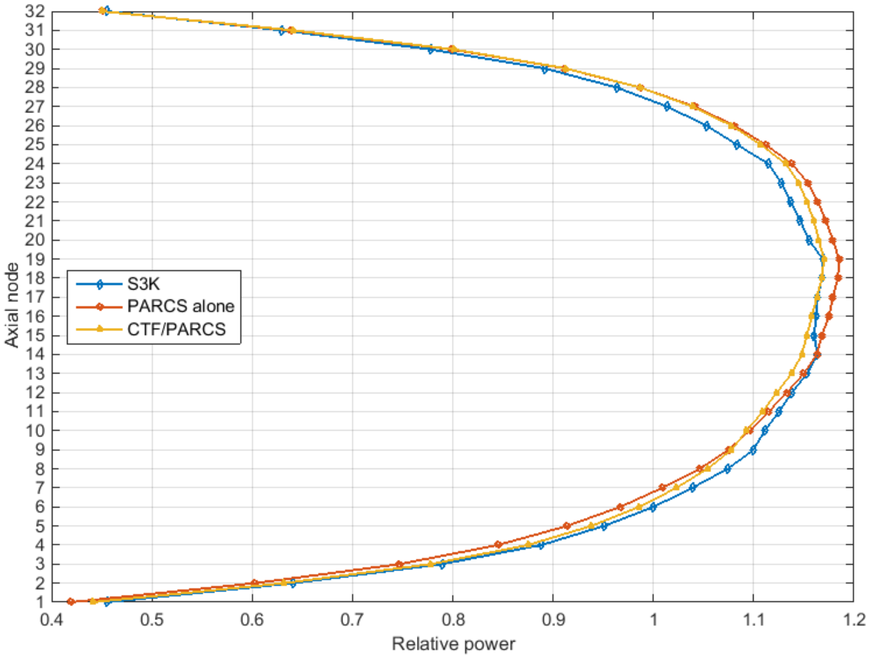
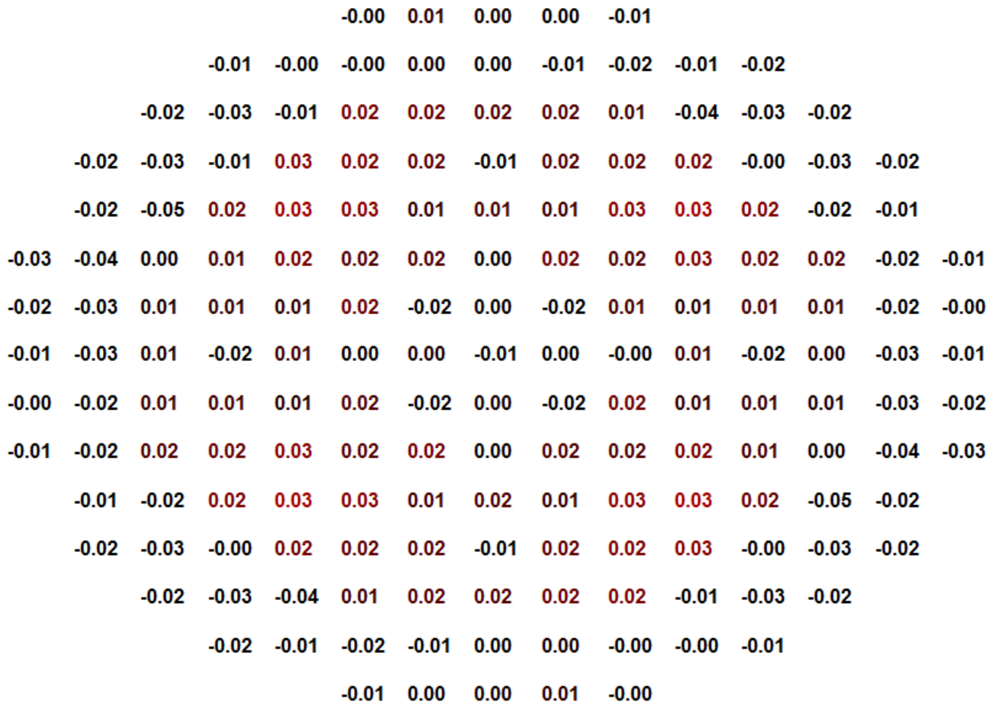


Figure 7: Steady state S3 and CTF/PARCS axial power profiles.

Figure 7 presents the comparison between the axial power profiles obtained with S3, PARCS alone (with S3 thermal-hydraulic boundary conditions) and pCTF/PARCS codes. The relative errors between the pCTF/PARCS and the reference code S3 is presented in Figure 8. As can be observed the relative errors in the power profiles are very small and the power profiles are practically identical. These results, together with those summarized in the Table 2 confirm the validity of the models and the cross sections for the analyzed case. The higher relative errors in the axial power profile are in the top and bottom axial nodes (in the central part they are always below the 2%), and the small differences in shape of the power profiles is due to the different models for taking into account the pressure drop in the spacer grids between S3 and CTF codes.

7.2. Transient Results

In this section the results of the control rod drop transient are presented. As pointed above the control rod drop lasts 2.1 seconds, and begins at 5.0



					-0.00	0.01	0.00	0.00	-0.01					
			-0.01	-0.00	-0.00	0.00	0.00	-0.01	-0.02	-0.01	-0.02			
		-0.02	-0.03	-0.01	0.02	0.02	0.02	0.02	0.01	-0.04	-0.03	-0.02		
	-0.02	-0.03	-0.01	0.03	0.02	0.02	-0.01	0.02	0.02	0.02	-0.00	-0.03	-0.02	
	-0.02	-0.05	0.02	0.03	0.03	0.01	0.01	0.01	0.03	0.03	0.02	-0.02	-0.01	
-0.03	-0.04	0.00	0.01	0.02	0.02	0.02	0.00	0.02	0.02	0.03	0.02	0.02	-0.02	-0.01
-0.02	-0.03	0.01	0.01	0.01	0.02	-0.02	0.00	-0.02	0.01	0.01	0.01	0.01	-0.02	-0.00
-0.01	-0.03	0.01	-0.02	0.01	0.00	0.00	-0.01	0.00	-0.00	0.01	-0.02	0.00	-0.03	-0.01
-0.00	-0.02	0.01	0.01	0.01	0.02	-0.02	0.00	-0.02	0.02	0.01	0.01	0.01	-0.03	-0.02
-0.01	-0.02	0.02	0.02	0.03	0.02	0.02	0.00	0.02	0.02	0.02	0.01	0.00	-0.04	-0.03
	-0.01	-0.02	0.02	0.03	0.03	0.01	0.02	0.01	0.03	0.03	0.02	-0.05	-0.02	
	-0.02	-0.03	-0.00	0.02	0.02	0.02	-0.01	0.02	0.02	0.03	-0.00	-0.03	-0.02	
		-0.02	-0.03	-0.04	0.01	0.02	0.02	0.02	0.02	-0.01	-0.03	-0.02		
			-0.02	-0.01	-0.02	-0.01	0.00	0.00	-0.00	-0.00	-0.01			
					-0.01	0.00	0.00	0.01	-0.00					

Figure 8: Steady state CTF/PARCS relative errors in the radial power profile.

Table 2: Summary of the steady state results.

Code	keff (-)	keff abs. error (pcm)	RMS error axial profile (%)	RMS error radial profile (%)	FXY (-)	Avg. Doppler (C)
S3 (ref.)	1.000141	--	--	--	1.638	636.61
PARCS alone	1.004725	458.5	2.30	1.86	1.538	636.46
pCTF/PARCS	0.999931	20.9	1.59	1.90	1.557	646.78

second. During the transient the information of PDDs signal is extracted with the objective of comparing with the real plant signals.

The simulation was performed on the Quasar cluster using 5 cores. The total wall time for the 30s control rod drop transient was 70 hours. Approximately 15 minutes of that time was spent in PARCS, and the rest of run time can be attributed to pCTF.

The same sequential execution took 135 hours. The speedup of the parallel execution with 5 processors was 3.86 and the efficiency was about 77%.

The total power evolution compared with the power measured during the test using two different methods (ex-core detector signals and heat balance), together with the behavior of the reactivity during the transient are shown in Figures 9 and 10, and reflect that when the control rod drops inside the core, the power decreases due to the negative reactivity feedback introduced by the control. Shortly after that, the total power is partially recovered by the positive reactivity feedback introduced by the decrease of fuel temperature and an increase of moderator density (Doppler and moderator density reactivity coefficients). As can be observed in Figure 9, the total power calculated by pCTF/PARCS is between the two bands that form the total power measured by the two methods available in the test data.

Figures 11, 12 and 13 represent the comparison of the real and simulated PDDs signals in the J6 position, the nearest PDD of the dropped control rod, at different axial levels. As can be observed in Figures 11 to 13 the simulated response of the PDDs during the transient is very close to the real signal.

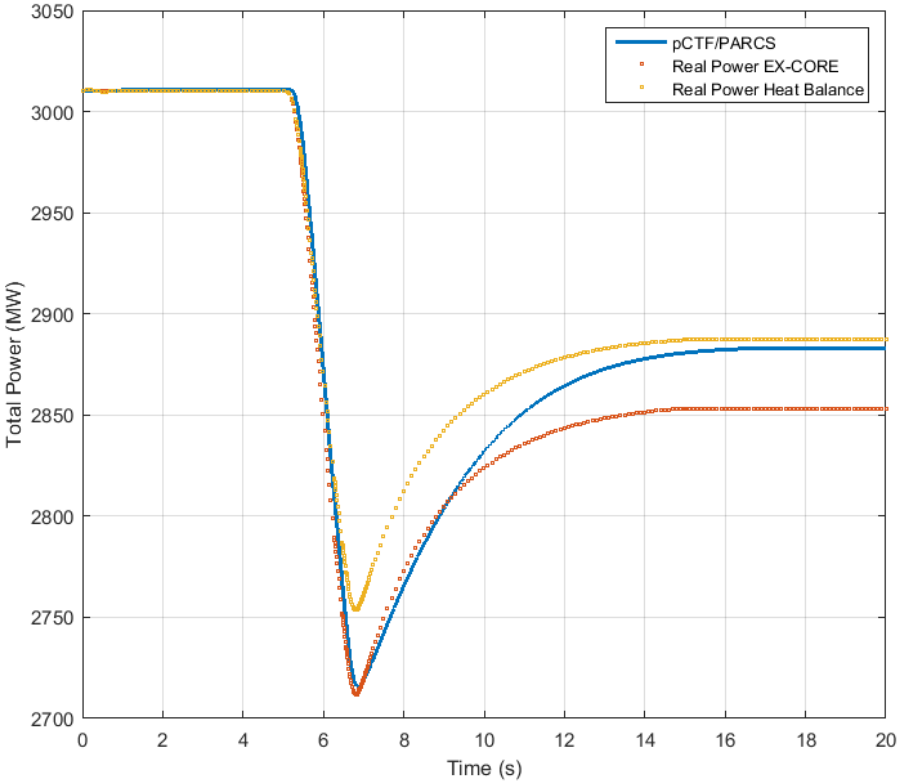


Figure 9: Total reactor power computed by pCTF/PARCS.

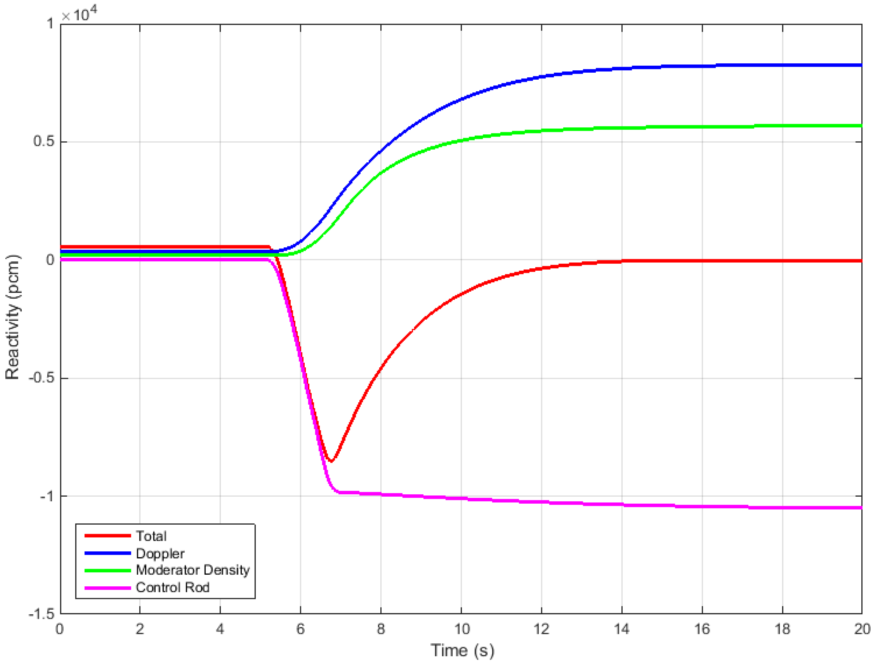


Figure 10: Evolution of the different components of the reactivity.

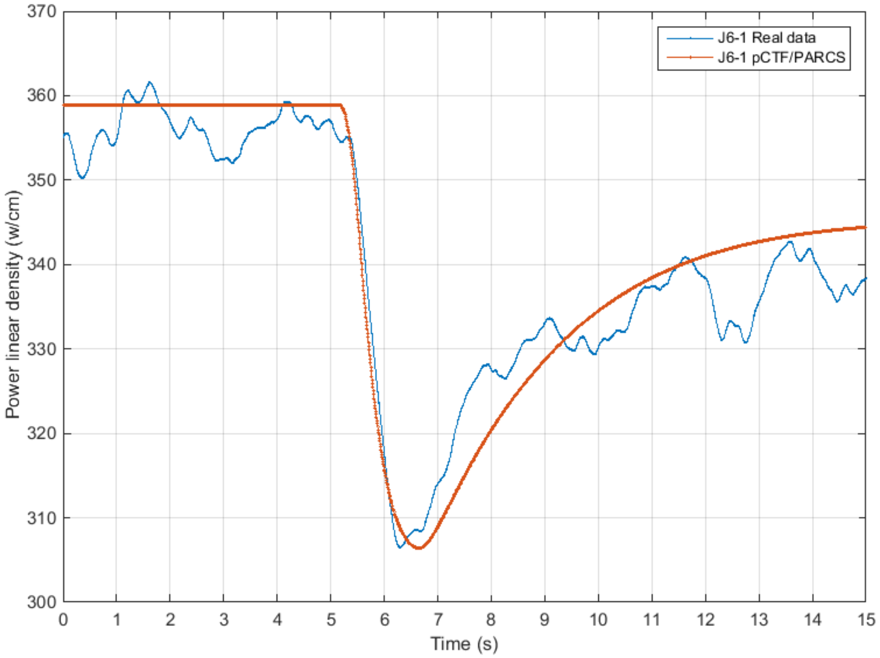


Figure 11: Real and simulated signal for incore detector J06, axial level 1.

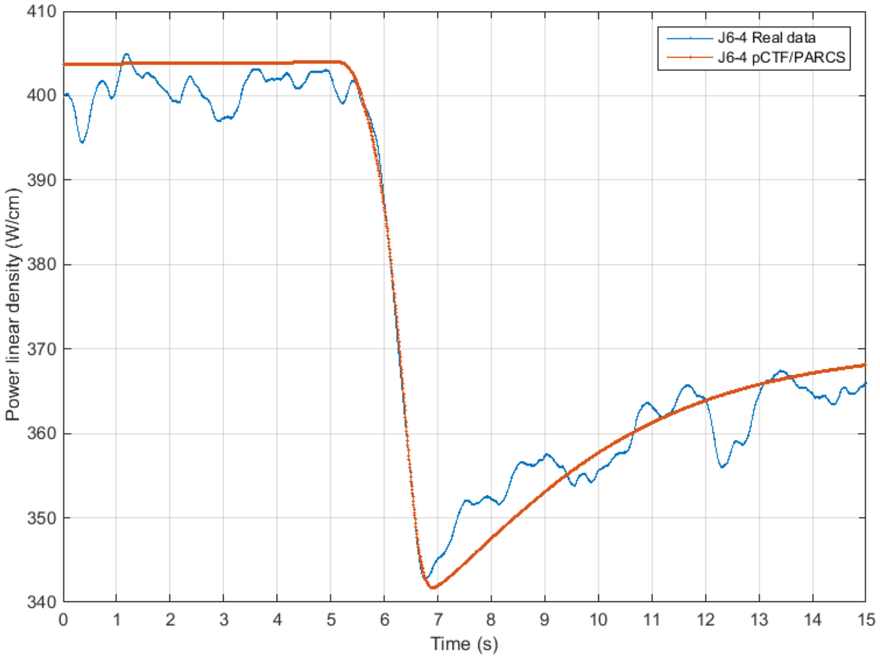


Figure 12: Real and simulated signal for incore detector J06, axial level 4.

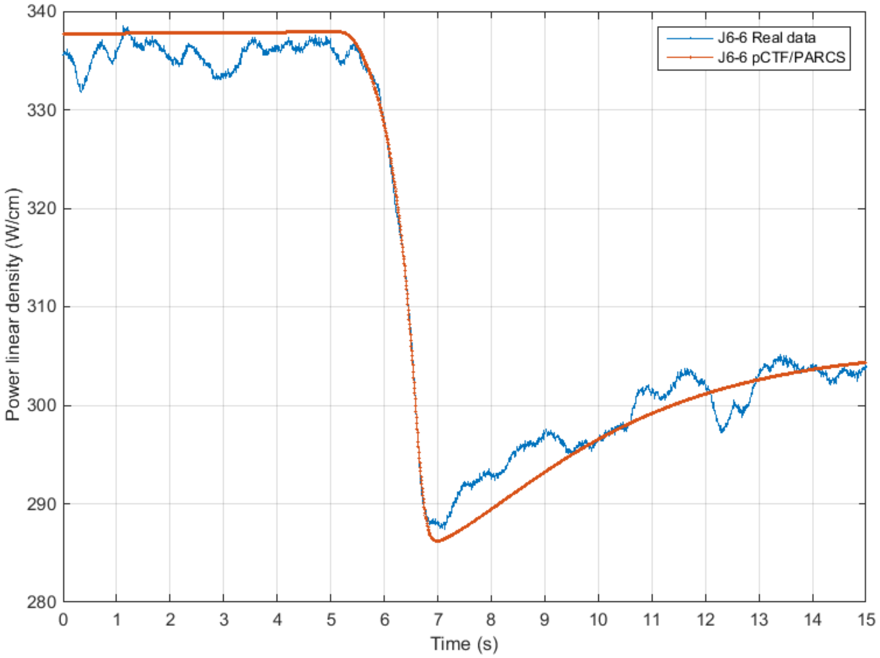


Figure 13: Real and simulated signal for incore detector J06, axial level 6.

8. Conclusions

A fully 3D pin by pin thermal-hydraulic model of a whole three-loop KWU-Siemens core has been developed for the CTF subchannel code. The previously developed coupling of CTF and PARCSv2.7 neutronic code permits the high resolution neutronics/thermal-hydraulics nuclear safety evaluations, like the control rod drop presented in this study.

The modern software engineering techniques as well as current parallel computing paradigms have been applied to an existing thermal-hydraulic subchannel code CTF. The original CTF has been parallelized using the message-passing technology to run the code on a cluster of computers. The parallel version called pCTF allows a significant reduction of the simulation time when it is applied to thermal-hydraulic detailed models of nuclear fuel assemblies.

The pCTF code has been benchmarked against a control rod drop transient. Results have been compared with the real data of control rod drop trial performed at the NPP, and have shown a good agreement with them. Thus, the models are validated against a control rod drop transient. The magnitude of the power decrease during the control rod drop is similar for all the analyzed PDDs detectors, and the evolution of the transient fits with the reference data.

The use of modern computational toolkits, PETSc in our case, provides a lot of flexibility as well as computational efficiency. In this way, it is very easy to experiment with various iterative linear solvers and preconditioners, and choose the most appropriate one for a particular scenario.

Regarding the parallelization, the goal of reducing the computation time for detailed models has been accomplished. However, as mentioned previously, scalability is quite limited because our domain decomposition approach considers full axial levels. A better approach would be to split the domain at the cell level, but this would require a significant redesign of the original CTF code. It remains as a future task to perform a second level of parallelization, with shared memory within a node, that will require a detailed data-dependency analysis of the computations performed at the cell level.

References

[1] COBRA/TRAC, A thermal-hydraulics code for transient analysis of nuclear reactor vessels and primary coolant systems. equations and constitutive models. vol. 1, Tech. Rep. NUREG/CR-3046, PNL-4385 (1983).

[2] COBRA/TRAC, A thermal-hydraulics code for transient analysis of nuclear reactor vessels and primary coolant systems. users manual. vol. 3, Tech. Rep. NUREG/CR-3046, PNL-4385 (1983).

[3] C. Y. Paik, et al., Analysis of FLECHT-SEASET 163-rod blocked bundle data using COBRA-TF, Tech. Rep. NUREG/CR-4166, PNL (1985).

[4] V. Kucukboyaci, Y. Sung, R. Salko, COBRA-TF parallelization and application to PWR reactor core subchannel DNB analysis, in: ANS MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method, 2015.

[5] D. Cuervo, M. Avramova, K. Ivanov, R. Miró, Evaluation and enhancement of COBRA-TF efficiency for LWR calculations, Annals of Nuclear Energy 33 (9) (2006) 837–847.

[6] MPI Forum, MPI: a message-passing interface standard, Int. J. Supercomp. Applic. High Perf. Comp. 8 (3/4) (1994) 159–416.

[7] R. van Geemert, M. Glück, M. Riedmann, H. Gabriel, Parallelized preconditioned BiCGStab solution of sparse linear system equations in F-COBRA-TF, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2011), 2011.

[8] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.5, Argonne National Laboratory (2014).

[9] S. Balay, W. D. Gropp, L. C. McInnes, B. F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A. M. Bruaset, H. P. Langtangen (Eds.), Modern Software Tools in Scientific Computing, Birkhaüser, 1997, pp. 163–202.

[10] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd Edition, SIAM Publications, 2003.

[11] G. L. G. Sleijpen, D. R. Fokkema, BiCGstab(ℓ) for linear equations involving unsymmetric matrices with complex spectrum, Electron. Trans. Numer. Anal. 1 (1993) 11–32.

[12] A. Abarca, et al., A procedure for coupled thermal-hydraulic subchannel and neutronic codes using COBRA-TF and PARCS, Tech. rep., INAC (2011).

[13] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderman, PVM: A Users' Guide and Tutorial for Network Parallel Computing, MIT Press, 1994.

[14] J. T. Cronin, K. S. Smith, D. M. V. Planck, SIMULATE-3. advanced three-dimensional two-group reactor analysis code, Tech. Rep. SOA-95/18, Studsvik (1995).

[15] J. Rhodes, et al., CASMO-4 users manual rev. 4, Tech. Rep. SSP-01/400 Rev 4, Studsvik Scandpower (2004).