

## Verification of the parallel pin-wise core simulator pCTF/PARCSv3.2 in operational control rod drop transient scenarios

Enrique Ramos<sup>1</sup>, Jose E. Roman<sup>1</sup>, Agustín Abarca<sup>2</sup>, Rafael Miró<sup>2</sup>, Juan A. Bermejo<sup>3</sup>, Alberto Ortego<sup>4</sup>, Jose M. Posada<sup>4</sup>

<sup>1</sup>*Dpto. de Sistemas Informáticos y Computación, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain*

<sup>2</sup>*Inst. de Seguridad Nuclear, Radiofísica y Medioambiental, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain*

<sup>3</sup>*Iberdrola Ingeniería y Construcción S.A.U, Av. Manoteras, 20. Edificio C, 28050 Madrid, Spain*

<sup>4</sup>*Centrales Nucleares ALMARAZ-TRILLO, Av. Manoteras, 46 bis, 5ª planta, 28050 Madrid, Spain*

*ramos@dsic.upv.es, jroman@dsic.upv.es, aabarca@isiryum.upv.es, rmiro@isiryum.upv.es, jbpi.iberinco@cnat.es, aoi@cnat.es, jmpb@cnat.es*

### ABSTRACT

Thanks to the advances in computer technology, it is feasible to obtain detailed reactor core descriptions for the safety analysis of Light Water Reactor (LWR), in order to represent realistically the fuel elements design, as is the case of 3D coupled simulations for local neutron kinetics and thermal-hydraulics. This scenario requires an efficient thermal-hydraulic code that can produce a response in a reasonable time for large-scale, detailed models. In two-fluid codes, such as the thermal-hydraulic subchannel code COBRA-TF, the time restriction is even more important, since the set of equations to be solved is more complex. We have developed an MPI parallel version of COBRA-TF, called pCTF. The parallel code is based on a cell-oriented domain decomposition approach, and performs well in models that consist of many cells. The Jacobian matrix is computed in parallel, being each processor in charge of calculating the coefficients related to a subset of the cells. Furthermore, the resulting system of linear equations is also solved in parallel, by exploiting solvers and preconditioners from PETSc. The goal of this study is to demonstrate the capability of the recently developed pCTF/PARCS coupled code to simulate large cores in a pin by pin level of detail in an acceptable computational time, using for this purpose two control rod drop operational transients that took place in the core of a three loop PWR nuclear reactor. As a result, the main safety parameters of the core hot channel have been calculated by the coupled code in a pin level of detail, obtaining best estimate results for this transient.

### KEYWORD

Sub-channel Simulation, COBRA-TF, pCTF, PARCS, Parallel Computing, PETSc

## I. INTRODUCTION

In order to describe the design of fuel elements in safety analysis, an accurately detailed reactor core description is very important for Light Water Reactors (LWR). This is also the case for coupled local neutronics/thermal-hydraulics evaluations. This kind of simulation can take a large amount of computational time, so it is highly convenient to take advantage of an efficient subchannel code that is able to provide the result in reasonable running times for large problems. The importance is more marked in case of thermal-hydraulic two-fluid codes such as COBRA-TF (CTF).

In this work, the two codes used in this study are COBRA-TF (CTF) and PARCS. CTF is an advanced subchannel code, applicable to both PWR and BWR analysis. The code features two-fluid, three-field representation of the two-phase flow [1, 2, 3, 4]. PARCSv3.2 (PARCS) is a three-dimensional (3D) reactor core simulator which solves the steady-state and time dependent, multi-group neutron diffusion and low order transport equation in orthogonal and non-orthogonal geometries [5].

We have developed a modification of the PARCS code that internally couples with pCTF (the parallel version of CTF). Our aim is to create a powerful tool that is capable of simulating the behaviour of reactor cores in asymmetric transients with high detail, in reasonable run times. This parallel coupled code has been already used to simulate a transient analysis related to a control rod drop event [6]. In this work, we continue the validation of this new coupled code, in this case in the context of a fictitious Reactivity Insertion Accident (RIA) scenario that can be reproduced using control rod drop operational transients. The results obtained in the simulated transient match the measurements acquired during an NPP real test.

The response time of the parallel coupled code is significantly smaller than the original sequential codes, since most of the computation is carried out in parallel. Furthermore, the parallel code can be run on a cluster of computers, so in this way the amount of aggregate memory available to the simulation is much larger. This allows addressing complex simulations with a much finer resolution than with the original codes.

The contents of the rest of the paper is the following. Sections II and III briefly describe the parallel COBRA-TF code and how it has been coupled with PARCS, respectively. Section IV provides a short description of the models used in the computational experiment. In Section V we show the main results obtained with the developed codes. We finish with some concluding remarks.

## II. PARALLEL COBRA-TF (PCTF)

In this section, we briefly describe the parallelization of COBRA-TF (CTF). CTF is a code for subchannel simulation that represents the two-phase flow by means of three fields (continuous vapor, continuous liquid, and entrained liquid drops in the region of annular flow). Basically, the code solves a set of nonlinear equations with a Newton-Raphson iteration, where at each step the Jacobian matrix must be evaluated and a linear system of equations must be solved with a sparse iterative method. The description here focuses on parallelization. Additional details about the CTF code can be found in [4, 6].

For solving linear systems in parallel we use PETSc [7, 8], a library intended for use in large-scale application codes. Many ongoing computational science projects are built around the PETSc libraries, and the nuclear engineering community is not an exception. PETSc includes a large suite of parallel linear, nonlinear equation solvers and ODE integrators that are easily used in application codes written in C, C++, Fortran and Python. It also provides many of the mechanisms needed within parallel

application codes, such as simple parallel matrix and vector assembly routines that allow the overlap of communication and computation. In addition, PETSc includes support for parallel distributed arrays useful for finite difference methods.

## II.B. Parallelization of CTF code

For the parallelization of the CTF code, we follow a domain decomposition methodology, where each MPI process is in charge of the unknowns associated with a subdomain. Then all the code must be adapted to this scheme, and interprocess communication must be performed whenever it is necessary to refresh data updated by the neighbouring process. The most important computational tasks are:

1. Update state variables to the current time step: synchronization and communication of all processes using MPI primitives [9].
2. Compute the coefficient matrix and the vector for the Newton-Raphson iteration: parallel distribution with MPI and resolution.
3. Solve the linear system of equations: parallel resolution using PETSc (MPI based).

The above three steps constitute the outer iterations, which are the most computationally expensive part of the simulation. This can be seen with a profiling tool acting on the sequential code, for instance the *valgrind tool*<sup>1</sup>. With the input data considered in section V, outer iterations represent around 75% of the total time. The solution of the linear system (the inner iteration, step 3 above) takes only about 20% of the total time.

Before doing the actual parallelization, the original sequential code was optimized with a re-writing of loops in some subroutines and a very important reduction of memory request for the execution. Storage required for all data structures was reduced as much as possible, and we also made an optimization of disk requirements for saving the execution results (a very important quantity of disk space).

In our code, the decomposition of the domain is done with respect to the axial direction. This decomposition involves the most important data structures, that must be carefully distributed across processes so that the resulting parallel computation is done correctly. Some important remarks are:

1. Decomposition of the computational domain in the axial dimension: a contiguous range of axial levels is assigned to every MPI process with a minimum of 1 axial level per process.
2. The number of levels in the axial dimension is usually small. This means that the parallel code cannot scale to many processes in this dimension, unless the mesh is refined axially (the maximum number of processes is equal to the number of axial levels).
3. State variables have a parallel representation, where each process stores part of the data. In addition to the chunk of axial levels assigned to it, each process needs extra space for the ghost cells. Ghost cells are additional space in the memory

---

<sup>1</sup><http://valgrind.org>

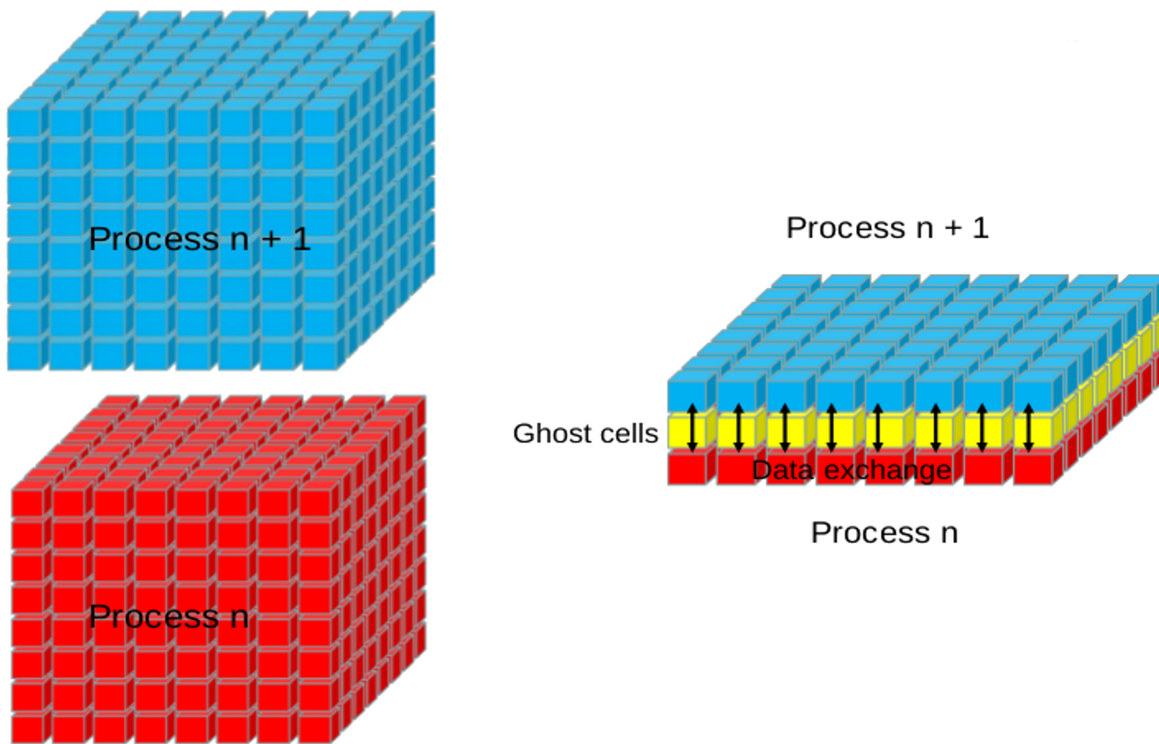


Fig. 1: Axial domain decomposition: subset of axial levels assigned to two neighboring processes.

allocated by each process where values received from neighboring processes are stored. Geometrically, these ghost cells are not part of the simulated domain neither defined in the input deck. Figure 1 shows a schematic representation of the axial domain decomposition. The subset of axial levels assigned to two neighboring processes are represented in the blue and red lattices in the 1. The yellow stripe represents additional memory space for cell data that have to be stored in both processes to store the values that are exchanged between them.

4. When required, point-to-point communication is used to update the ghost cells with values of the analogue cells owned by the neighbor processes.

The next step was the replacement of the SPARSKIT library (that was used in the original sequential code) by the equivalent components of PETSc. For this purpose, the main tasks are:

- Replace the invocation to SPARSKIT subroutines, based on the reverse communication interface idea, with the corresponding PETSc functionality. In PETSc we must create a parallel matrix object, whose entries are set in parallel, and the right-hand side vector object. These are fed to the linear solver object.
- Choose an iterative linear solver that replaces the one originally used from SPARSKIT. In fact, since PETSc is designed in an object-oriented way with basic polymorphism capabilities, it means that it is possible to choose the actual solver at run time. Many possibilities are available, including GMRES and Bi-CGStab [10].
- Similarly, pick the most appropriate preconditioner from the list of all PETSc possibilities.

Once the correctness of the parallel code has been assessed, a final tuning stage is to select the most appropriate strategy for the linear solve. In this work we tested the most important combinations of preconditioners (Jacobi, Block Jacobi, additive Schwarz, SOR) with iterative solver (Bi-CGStab, Bi-CGStab( $\ell$ ), GMRES). The analysis with different test cases showed that Bi-CGStab with Block Jacobi preconditioner is the optimal combination with the smallest execution time.

### III. PIN LEVEL COUPLED SIMULATIONS USING PCTF/PARCS

#### III.A. Overview of the pCTF/PARCS simulation procedure

This paper extends the work of [5], where the authors implemented a fully explicit coupling scheme between CTF and PARCS. This coupling mechanism was based on external data exchange, via the GI routines available in PARCS and a few new subroutines incorporated to CTF to allow the exchange of data by means of the PVM environment [11]. Data must be adapted from the neutronic nodalization to the thermal-hydraulic subchannels, and this was achieved by an averaging methodology with mapping routines implemented in CTF. In complex thermal-hydraulic CTF models, an external file called MAPTAB is required to specify the correspondence between thermal-hydraulic and neutronic nodalization mapping.

In order to perform a pin-wise neutron kinetics simulation with PARCS v2.7 [12], the coupled code has been modified to make possible the use of the *pin power reconstruction* capability of PARCS with the assembly averaged thermal-hydraulic feedback from pCTF. PARCS needs to compute the pin power from the nodal solution using *power form functions* of the fuel assemblies for each energy group. In the approximation presented in this work, these functions are provided from the solution of a previously simulated criticality problem performed with the KENO-VI code of SCALE6.2 suite. Additionally to the nodal power, CTF receive from PARCS the whole 3D pin power distribution of the core. Finally, CTF feeds its rod heat structures with pin powers after a renormalization process. In the scheme of the simulation procedure of the coupled parallel code presented in Figure 2 it can be seen how after the nodal solution, PARCS calls its pin power reconstruction module for computing the pin power, and finally the pin power is passed to pCTF through a PVM communication.

#### III.B. PARCS pin power reconstruction methodology

The PARCS procedure to obtain the pin power distribution is briefly described in this section. The primary unknown in the PARCS nodal solution kernel is the node average flux. In order to obtain local pin power distributions, it is thus necessary to *reconstruct* pin powers from the node average flux values. This is achieved in PARCS by multiplying the heterogeneous power form functions by the homogeneous intranodal flux distribution [13].

On one hand, the heterogeneous group-wise power form functions have to be calculated using a 2D fine mesh transport simulation (for example at the same time that the cross section are generated by a depletion simulation). To obtain the power form functions the fine mesh multigroup fluxes are multiplied by the fission macroscopic cross section and finally collapsed in two energy groups in order to obtain functions that can be multiplied by the two-group homogeneous intranodal flux distribution computed by PARCS.

On the other hand, PARCS needs to calculate the homogeneous intranodal flux distributions. For smooth axial variations of

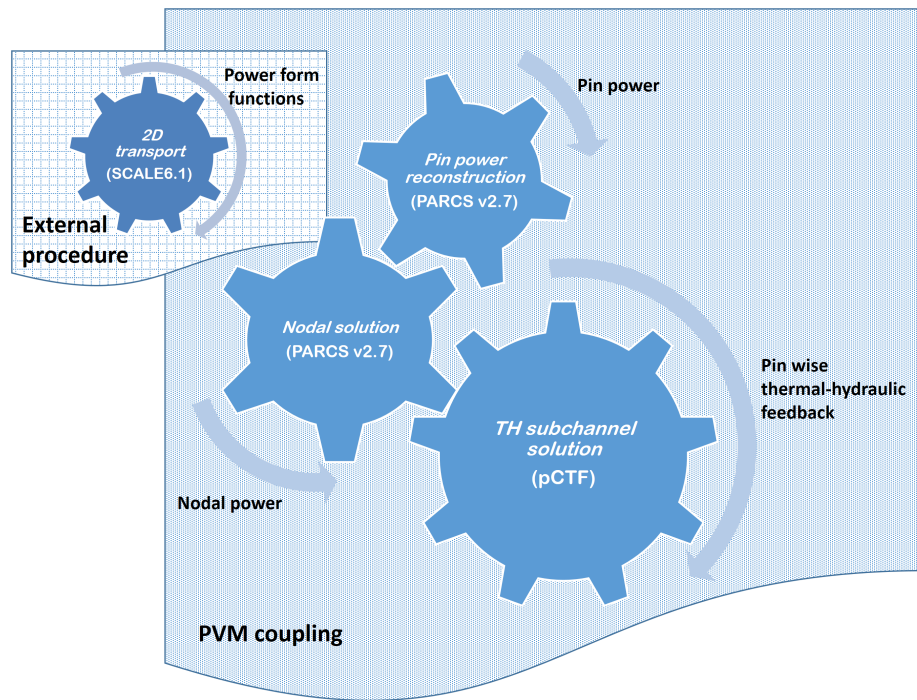


Fig. 2: Scheme of the simulation procedure of a couple pin-wise simulation with pCTF/PARCS

the flux, it can be assumed that the radial and axial dependences of the intranodal flux are separable. For this reason, all the current pin power reconstruction methods deal with a two-dimensional problem. The axial dependence is taken into account during the calculation of the homogeneous intranodal fluxes thanks to the different constraints used for in the polynomial expansions for approximate the flux shape at different axial nodes. The homogeneous intranodal flux is calculated by finding the analytical solution of a 2D fixed source problem in which the surface average currents are specified at the four boundaries. The surface average currents are obtained from the converged node average flux distribution at a given state.

The most widely accepted method assumes that the radial dependence of a pin power can be described by a two dimensional polynomial expansion of the flux. This expansion is augmented by a set of exponential functions applied only to the thermal energy group. In this approach, the determination of the intranodal flux becomes into a Dirichlet problem in which a two dimensional partial differential equation is solved with the function values specified at the boundary. The solution accuracy is sufficiently good using 8 boundary conditions, 4 are surface and 4 corner point fluxes. PARCS was based in this analytic approach with 8 boundary conditions (see Figure 3).

For the estimation of the corner point fluxes, the Method of Successive Smoothing (MSS) approach is used so that the Dirichlet problem can be solved with the desired number of boundary conditions. (MSS) is based on the assumption of a linear flux variation around a corner and the resulting expression for the corner involves only 4 node average and 4 surface average fluxes around the corner point. This approach is computationally efficient and is improved in PARCS with the corner point balance as an additional physical constraint. The set of resulting equations constitute a linear system with the corner fluxes being the unknowns. The linear system needs to be solved simultaneously.

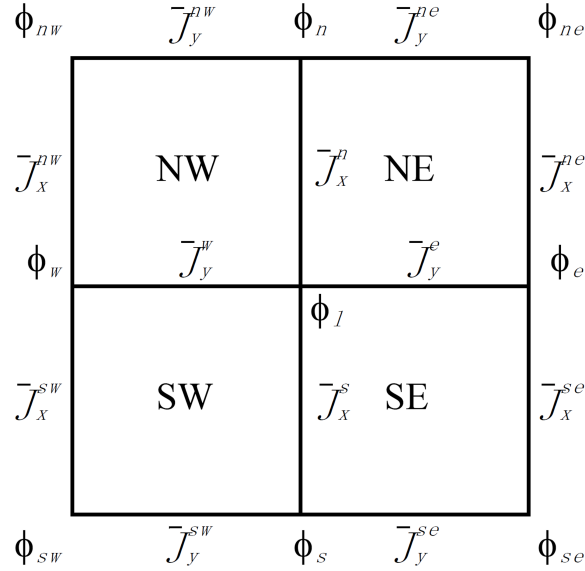


Fig. 3: Variable definition for corner flux calculation

### III.C. Calculation of the power form functions

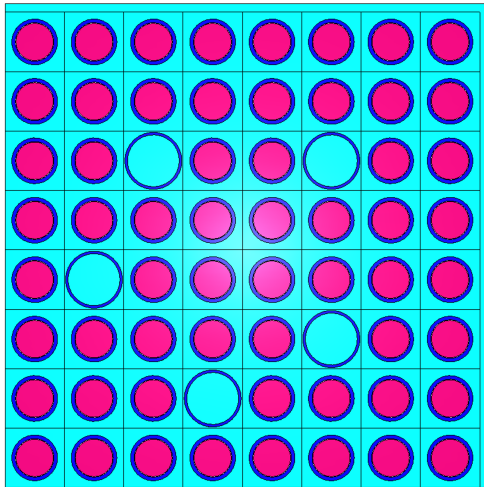
As it was mentioned before, the power form functions have been obtained from the solution of an externally simulated criticality problem performed with the KENO-VI code of SCALE6.2 suite. The pin power reconstruction of PARCS needs to be fed with power form functions for each energy group and for unrodded and rodded fuel assemblies. In the procedure used to obtain the power form function, several assumptions have been made: the fuel rods contain an isotopic composition equivalent to that which is obtained considering the average burnup of the core to do the depletion; the initial fresh fuel (before the depletion) has an enrichment of 4.2%wt U-235; and one quarter symmetry in the assembly is assumed, the boundary conditions are symmetric in the  $x$  and  $y$  boundary limits and zero flux in axial direction limits beyond the active fuel (positive and negative  $z$ -direction). Figure 4 shows the KENO-VI geometry for rodded and unrodded fuel assemblies.

The power form functions were obtained from a 3D mesh tally superimposed over the fuel assembly considering two energy groups (fast and thermal neutron energy groups). The cells of the mesh tally match with pin cells of the fuel assembly so that the obtained fluxes can be directly normalized to be used as power form functions for PARCS. Figures 5 and 6 represent the fast and thermal neutron fluxes calculated by KENO-VI in the mesh tally superimposed over the fuel assemblies.

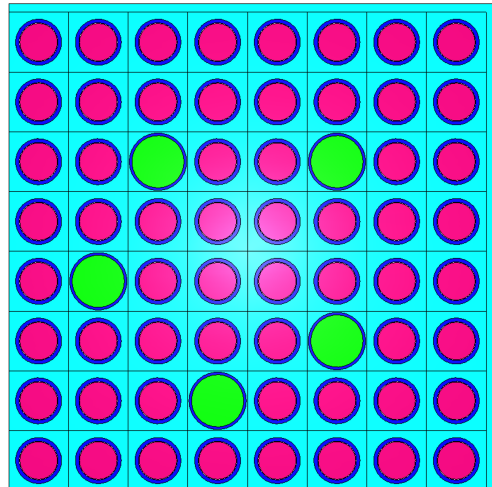
The fast and thermal neutron group fluxes obtained with KENO-VI are homogenized in the fuel pin cells and normalized in order to obtain the power form functions. Finally, these power form functions are coded in the PARCS input deck, so that they can be used by the code to calculate the heterogeneous fluxes to reconstruct the pin power responses from the nodal solution.

## IV. PCTF/PARCS MODELS

We use a model of the core of a pressurized water reactor (PWR) with three loops. The number of fuel elements in the core is 177, with 236 rods per fuel element and 20 guide tubes. During the test, the operating conditions correspond to Hot Full Power

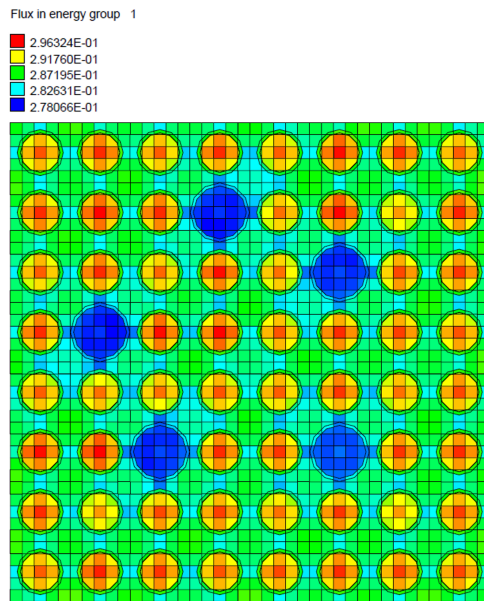


(a) Rodded assembly

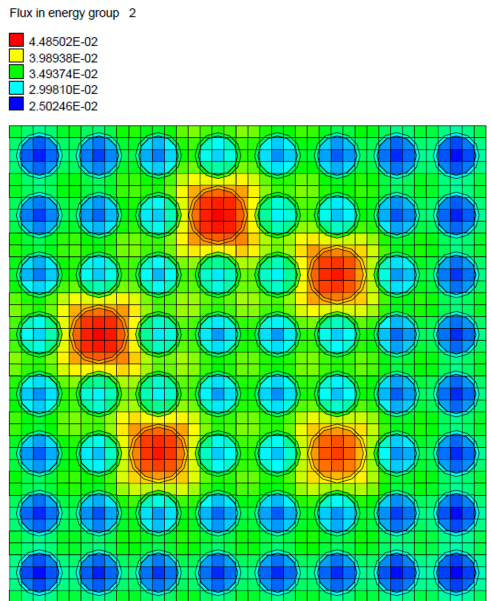


(b) UnroDDed assembly

Fig. 4: Keno-3d representation of the KENO-VI geometry of the PWR fuel assembly.



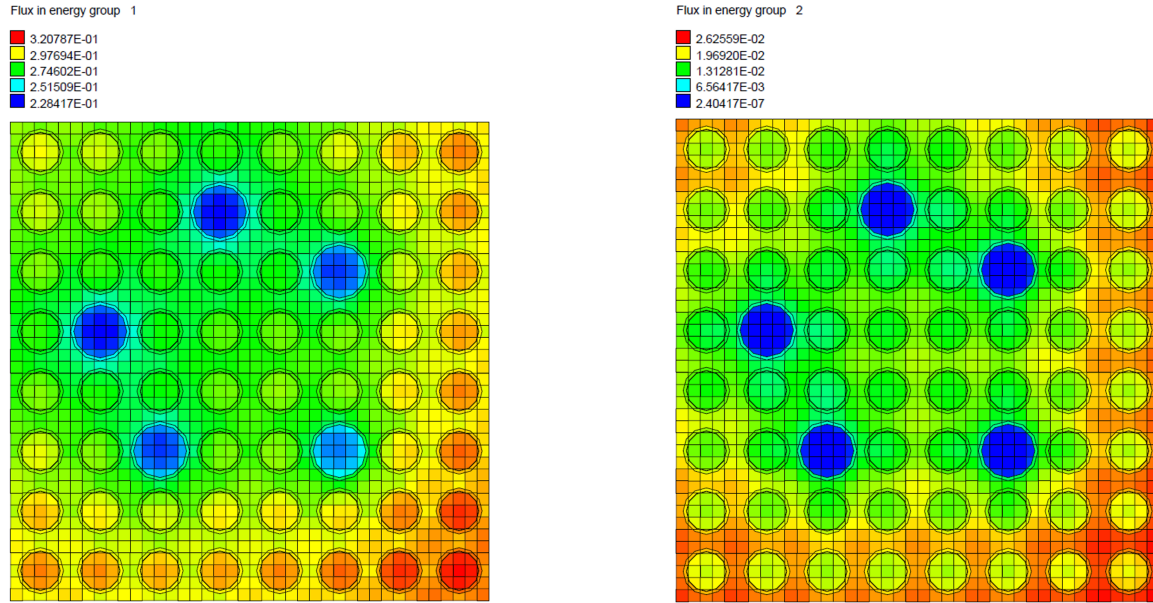
(a) Fast neutron flux (group 1)



(b) Thermal neutron flux (group 2)

Fig. 5: Representation of the KENO-VI pin cell wise neutronic fluxes in the unroDDed assembly.





(a) Fast neutron flux (group 1)

(b) Thermal neutron flux (group 2)

Fig. 6: Representation of the KENO-VI pin cell wise neutronic fluxes in the rodded assembly.

(HFP), where at the Beginning of Cycle (BOC) the nominal power is 3010 MWth and the core flow is 15605.6 kg/s.

#### IV.A. Thermal-hydraulic pCTF model

The core of the three-loop PWR reactor has been modeled, in a subchannel scale, using 45313 flow channels, of which 45312 represent the active core (cells colored in blue in Figure 7) and 1 for the whole external core bypass (red colored cells in Figure 7). The domain is split axially in 34 equispaced nodes of 10.625 cm. Finally, the thermal-hydraulic model has a total of 1540642 cells.

The 41772 fuel rods that comprise 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 dissipative thermal structures. Figure 7 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.

#### IV.B. Neutronic PARCSv3.2 model

The neutron kinetic model has been developed with the widely proven neutron diffusion code PARCSv3.2. The nodal model consists in a 3D mesh of 241 x 34 nodes, in which the active core has 177 x 32 nodes as it is shown in Figure 8. Iberdrola Generación Nuclear S.A. has provided the tables of cross sections, considering 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 SIMTAB methodology was used to generate the cross section sets. This methodology, developed in collaboration with Iberdrola Generación Nuclear S.A., is fed with data from

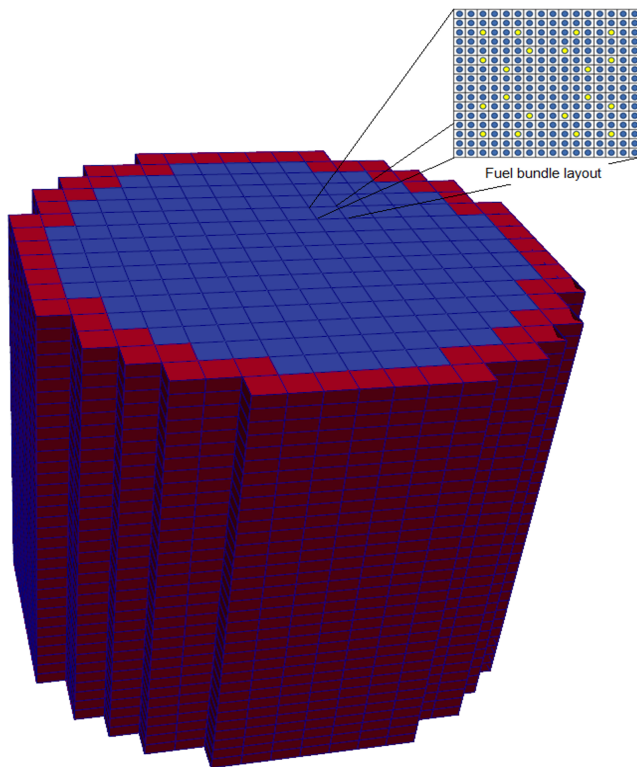


Fig. 7: Scheme of the thermal-hydraulic model of the reactor core

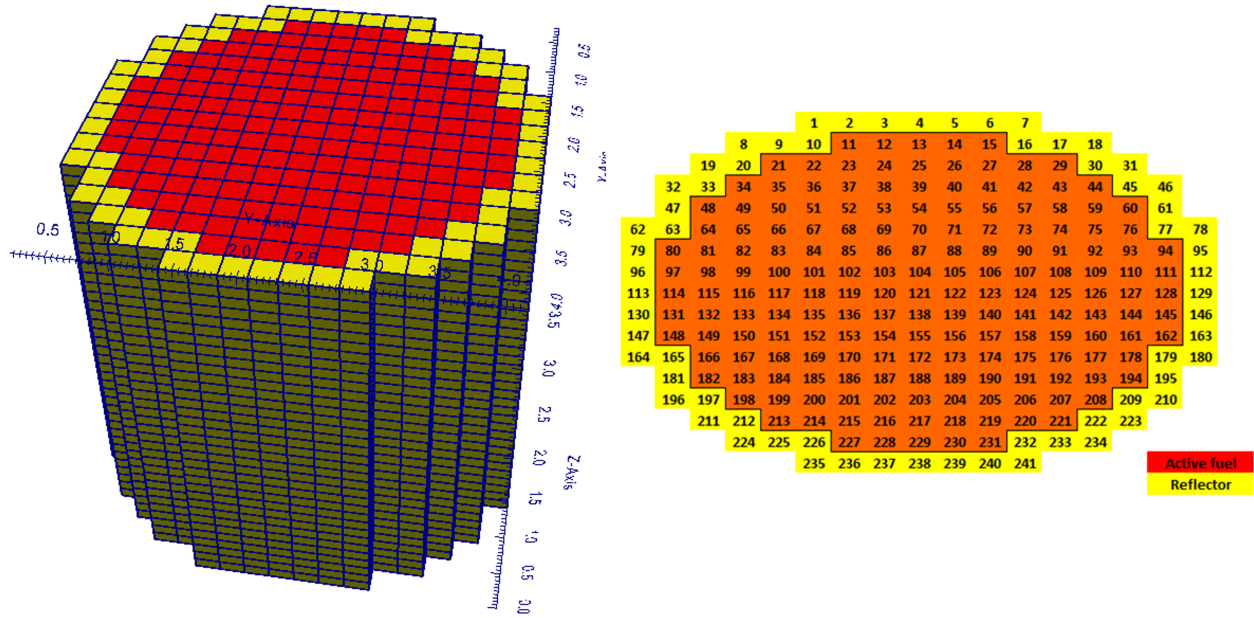


Fig. 8: Neutronic nodal mesh (left) and radial scheme of the nodalization (right)

CASMO4-SIMULATE codes and generates the cross sections as a function of the fuel temperature, moderator density, boron concentration and control rod insertion.

## V. RESULTS

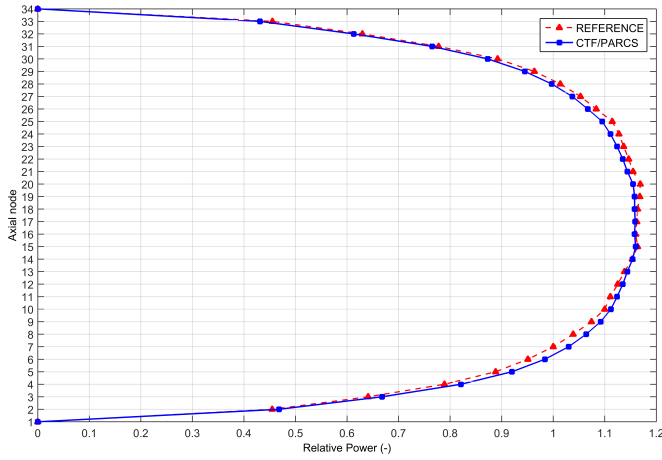
In this section, we describe the results for both the coupled steady state (HFP conditions at the BOC), as well as two control rod drop transients. The computational results correspond to a machine named Quasar, which is a small cluster with 4 nodes. Each node has 32-core AMD Opteron processors at 2.4 GHz, with 96 GB of RAM. The connection between the nodes is Gigabit Ethernet.

### V.A. Steady State Results

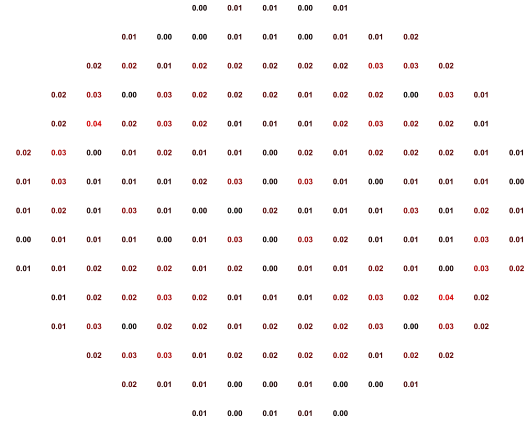
We use the results of SIMULATE3 (S3) as a reference to verify the steady state conditions achieved with the pCTF/PARCS coupled code. The power profiles and the eigenvalue of the loaded core, calculated from the nodal solution of pCTF/PARCS, are compared with the ones obtained with the reference code.

As can be seen in Figure 9a, the axial power profiles obtained with S3 and pCTF/PARCS present good agreement, with relative errors lower than 2 %. The same conclusion can be extracted from the relative errors of the radial power profiles presented in Figure 9b.

Table I shows the summary of the results obtained in the coupled steady state simulation. These values confirm the validity



(a) Comparison of axial power profiles



(b) Radial power profile relative errors

Fig. 9: Power profiles steady state analysis

TABLE I: Summary of steady state results

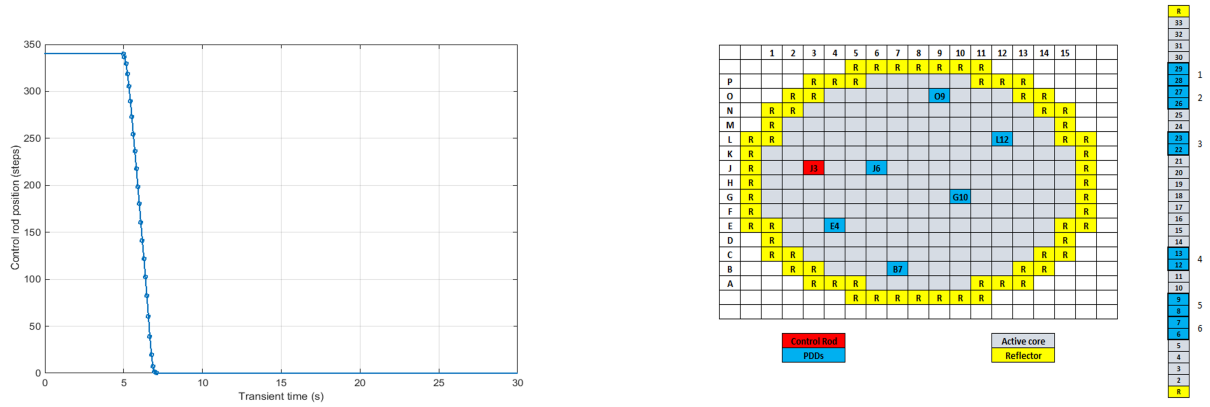
Code	keff	keff absolute error (pcm)	RMSE axial profile (%)	RMSE radial profile (%)
Plant Reference	1.000141	-	-	-
PARCS alone	1.004725	458.5	2.30	1.86
pCTF/PARCS	0.999931	20.9	1.59	1.90

of the cross section and models developed for carrying out the control rod drop tests. The PARCS steady state alone results presented in the Table I are calculated using a fixed nodal thermal-hydraulic boundary conditions from the plant reference code (S3).

## V.B. Transient Results

In this section two different control rod drop operational transients are simulated with the pCTF/PARCS parallel coupled code. Both transients are carried out at the same operational conditions, corresponding with a hot full power (HFP) state, where at the beginning of cycle (BOC) the nominal power is 3010 MWth and the core flow is 15605.6 kg/s.

The RIA transients correspond with two control rod drop tests made by the nuclear power plant at different core locations, the J3 and the N7 according to the scheme of Figure 10b. The simulated control rod insertions are defined into the input deck of PARCS, lasting 2.1 seconds on a 30.0 seconds transient. The first 5.0 seconds are a null transient aiming at ensuring stationary



(a) Control rod movement during insertion

(b) Control rod and PDDs locations

Fig. 10: Temporal evolution of the control rod position

stable conditions at the outset of the transient. Figure 10a shows the temporal sequence of control rod movement.

The simulation of the RIA scenarios was carried out using 5 cores of Quasar, spending a total elapsed time of 70 hours for each 30 s transients. The parallel execution with 5 cores had a speedup of 3.86, with an efficiency of 77%.

In Figures 11 and 12 the total power evolution during the control rod drops are compared against the power measured in the power plant by two methods: using the Excore detectors signals (red squares) and a coolant energy balance of the core (green squares). The transients reflect the expected evolution of the power in a control rod drop event: first the power goes down because of the negative reactivity feedback caused by the control rod. Shortly after that, part of the total power is recovered thanks to the positive reactivity feedback induced by the decrease of fuel temperature and an increase of the density of the moderator. Both transients only differ slightly in the amount of power dropped when the control rod goes down, which depends on the reactivity worth of the tested control rod.

Figure 13 represents the pin by pin radial power distribution in the core after the control rod drops. As can be observed the power shape is disturbed around the area in which the dropped control rod is located in each transient.

The shape of the reactor core power is measured in the PWR nuclear power plant using the Power Density Detectors (PDDs), placed in the guide tubes inside de reactor core. In order to obtain the response of these detectors from the neutron flux returned by PARCS, we used Equation 1, where  $\phi$  is the thermal flux computed for each detector position, and  $F_{ID}$  is a conversion factor for each detector, corresponding to the average power given by the detector at the stationary case.

$$P_{ID} = \frac{\phi}{\phi_{t=0}} \times F_{ID} \quad (1)$$

Finally, with the objective of assessing the parallel coupled code pCTF/PARCS, when both neutronics and thermal-hydraulics are simulated in a pin level of detail, the real plan signals of some PDDs incore detectors are compared with the values obtained by the coupled code. As can be observed in Figures 14–17, in which a comparison of the PDDs located in the positions J6 and O9 at the axial levels 1 and 6 are shown, the prediction of the pCTF/PARCS is very close to the real signal for both scenarios, slightly under predicting the power for the N7 control rod drop transient.

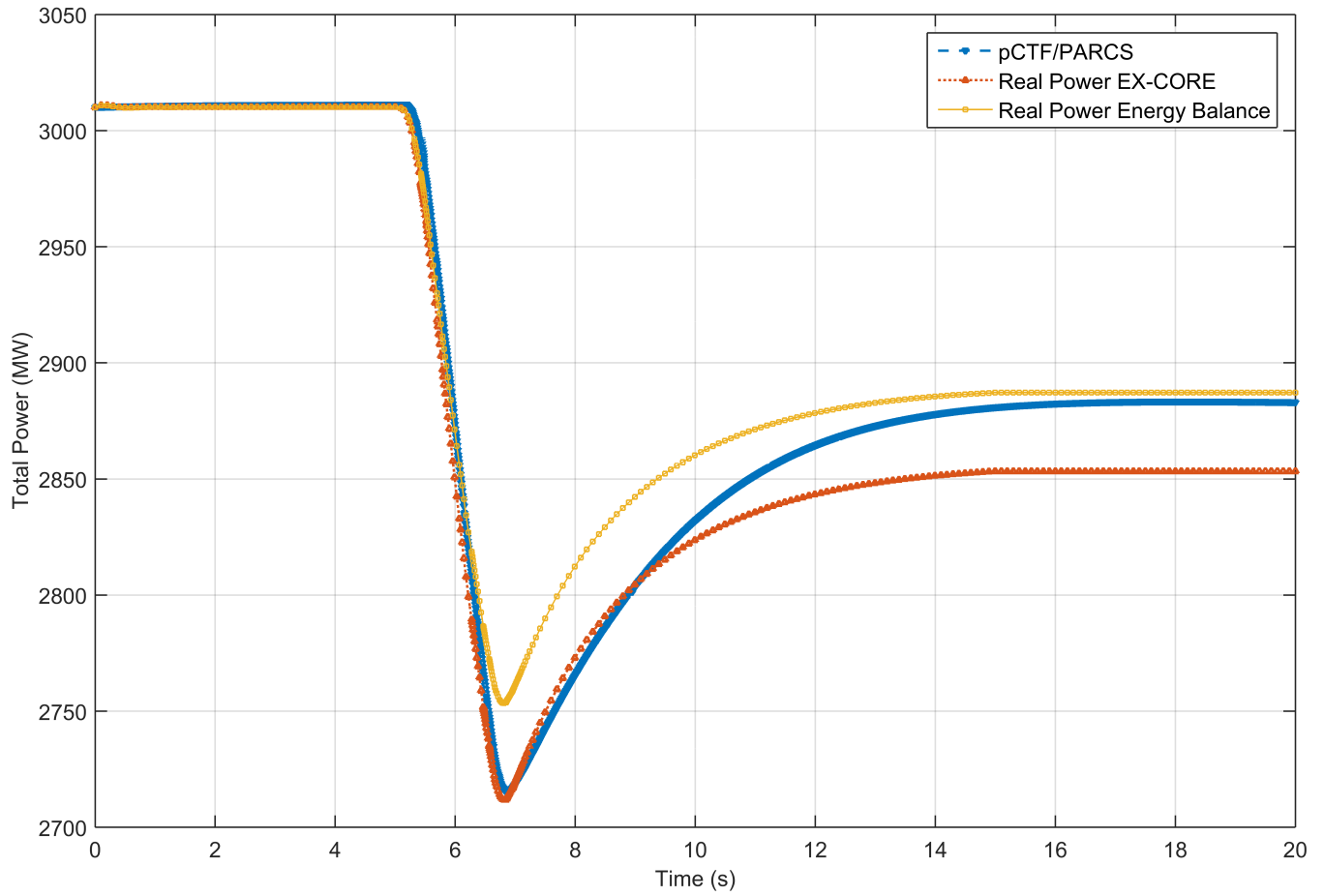


Fig. 11: Power evolution during the control rod drop J3

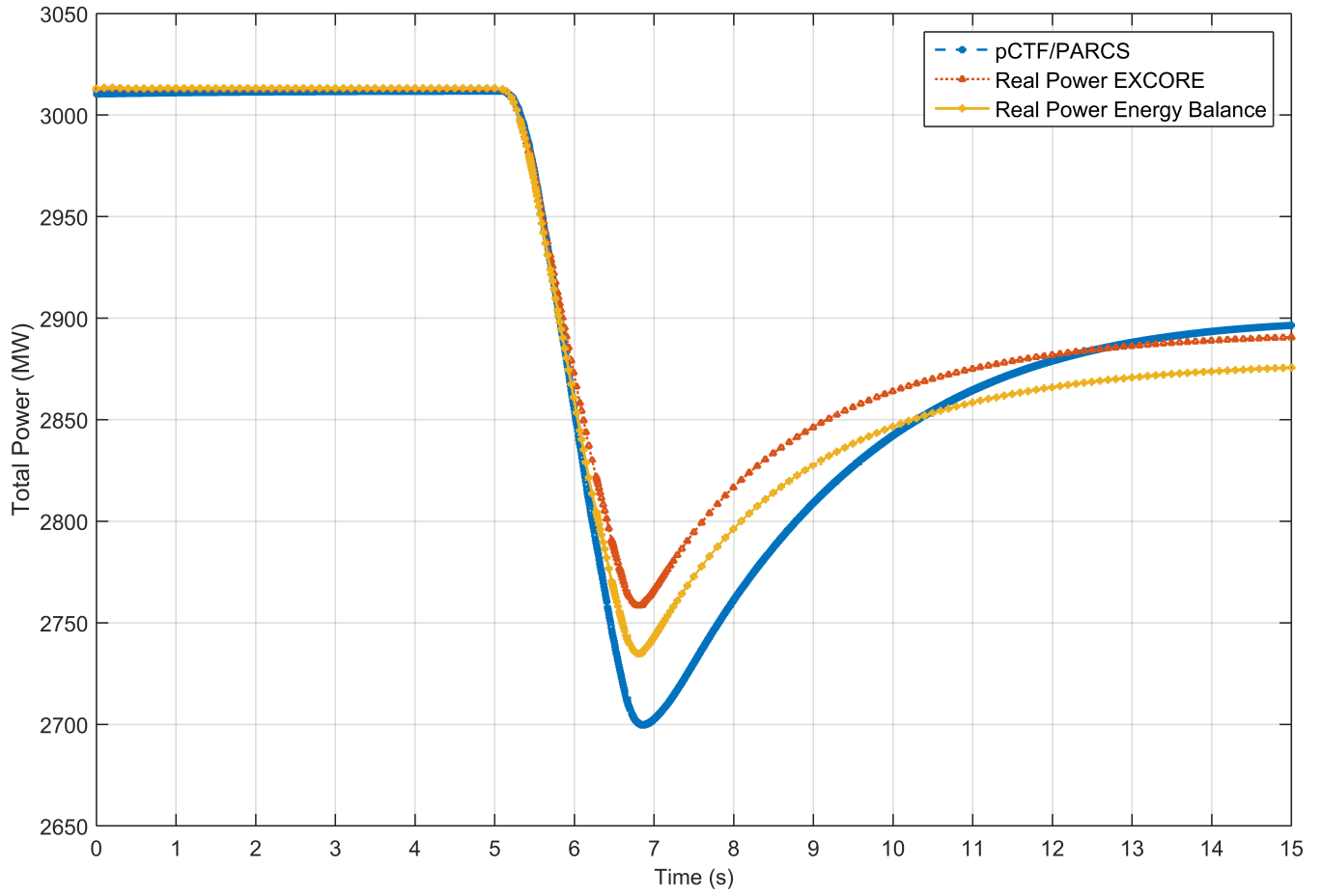
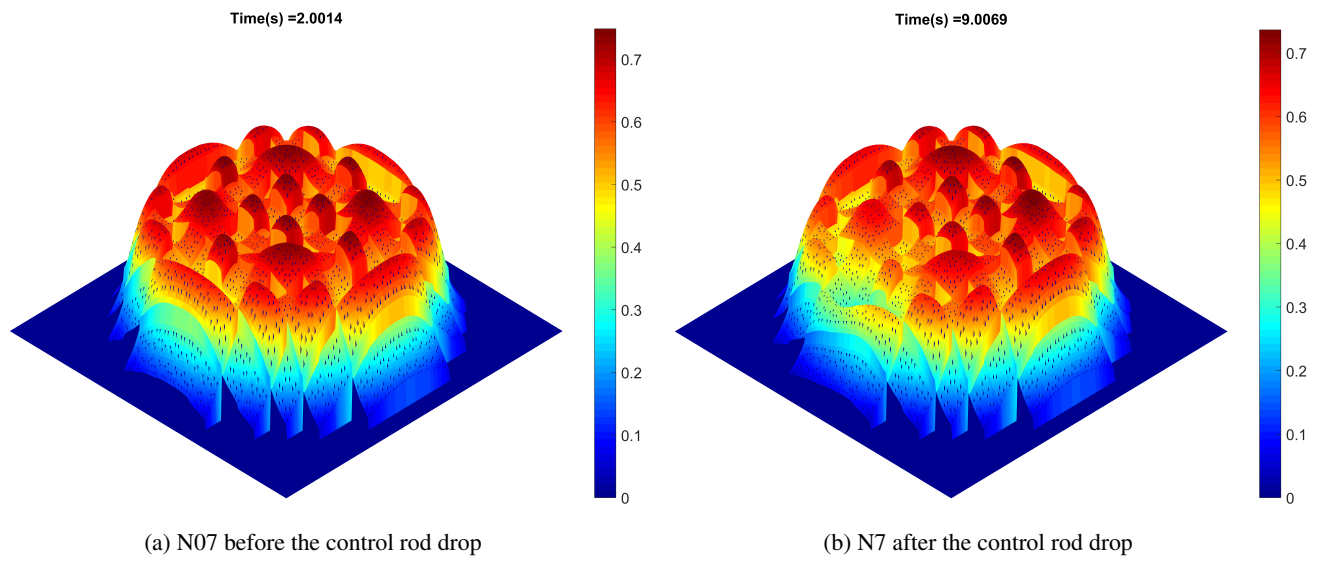


Fig. 12: Power evolution during the control rod drop N7



(a) N07 before the control rod drop

(b) N7 after the control rod drop

Fig. 13: Radial power shape after control rod drops

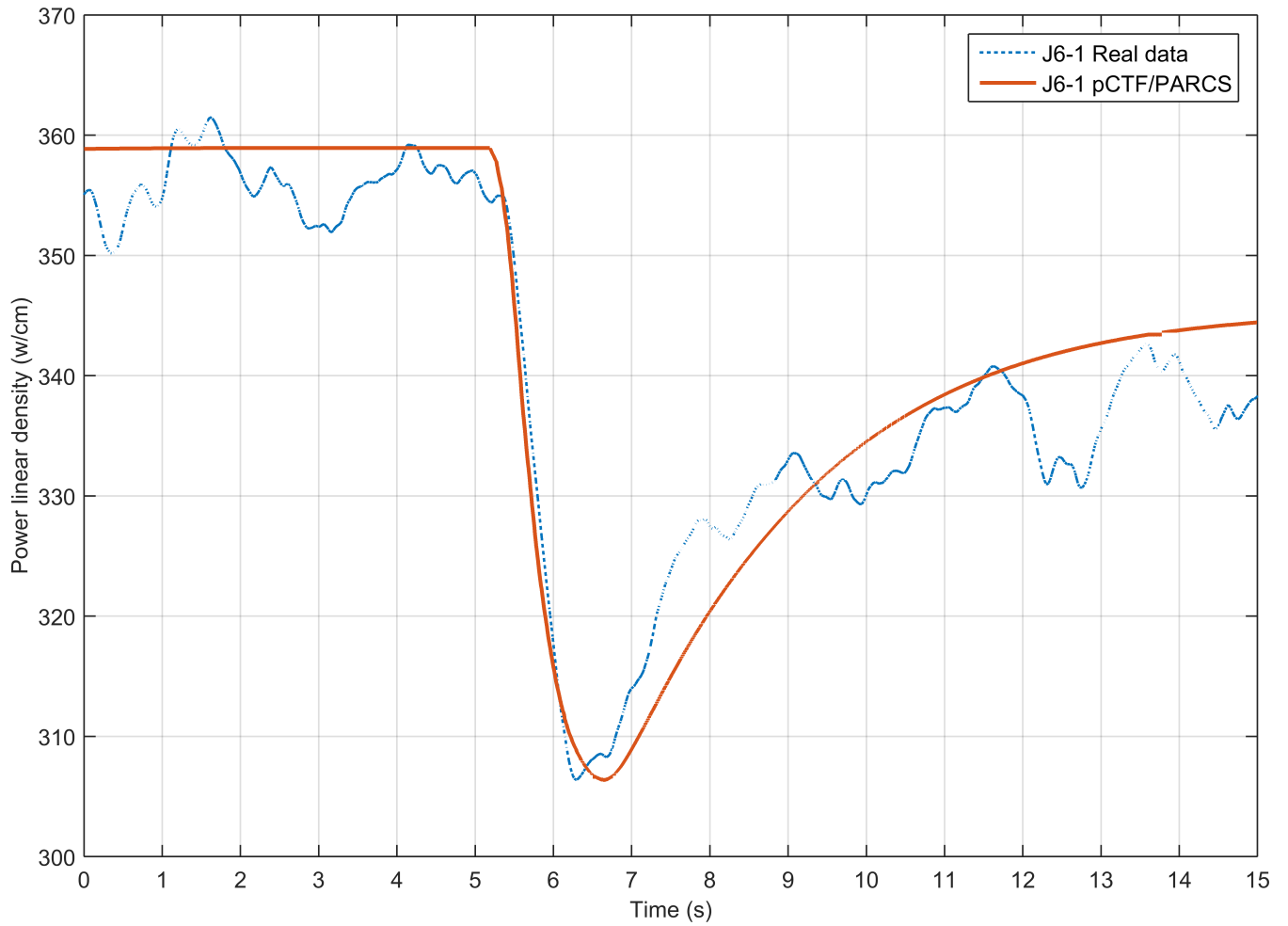


Fig. 14: Real versus simulated signal for incore detector J6 at axial level 1



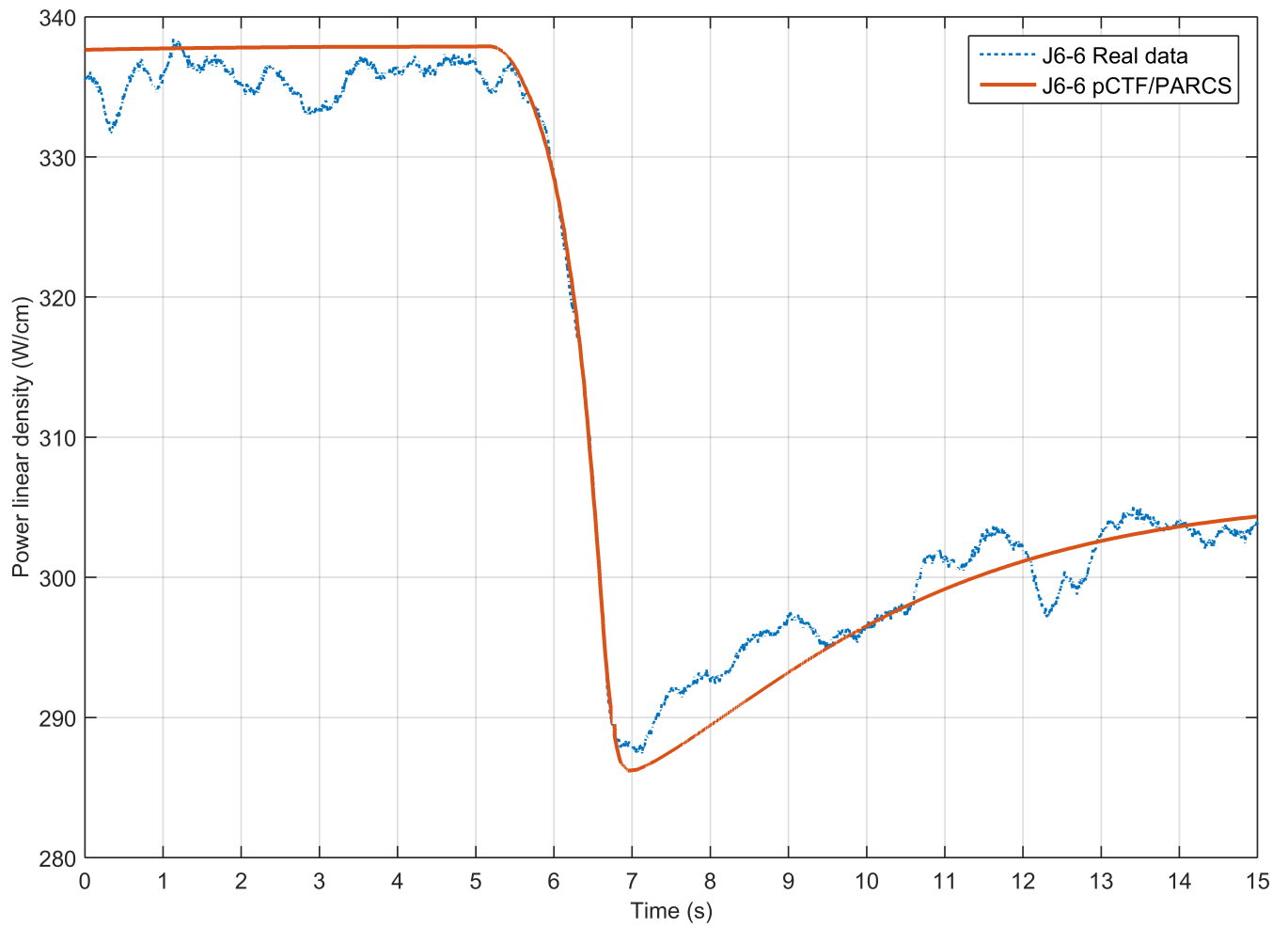


Fig. 15: Real versus simulated signal for incore detector J6 at axial level 6

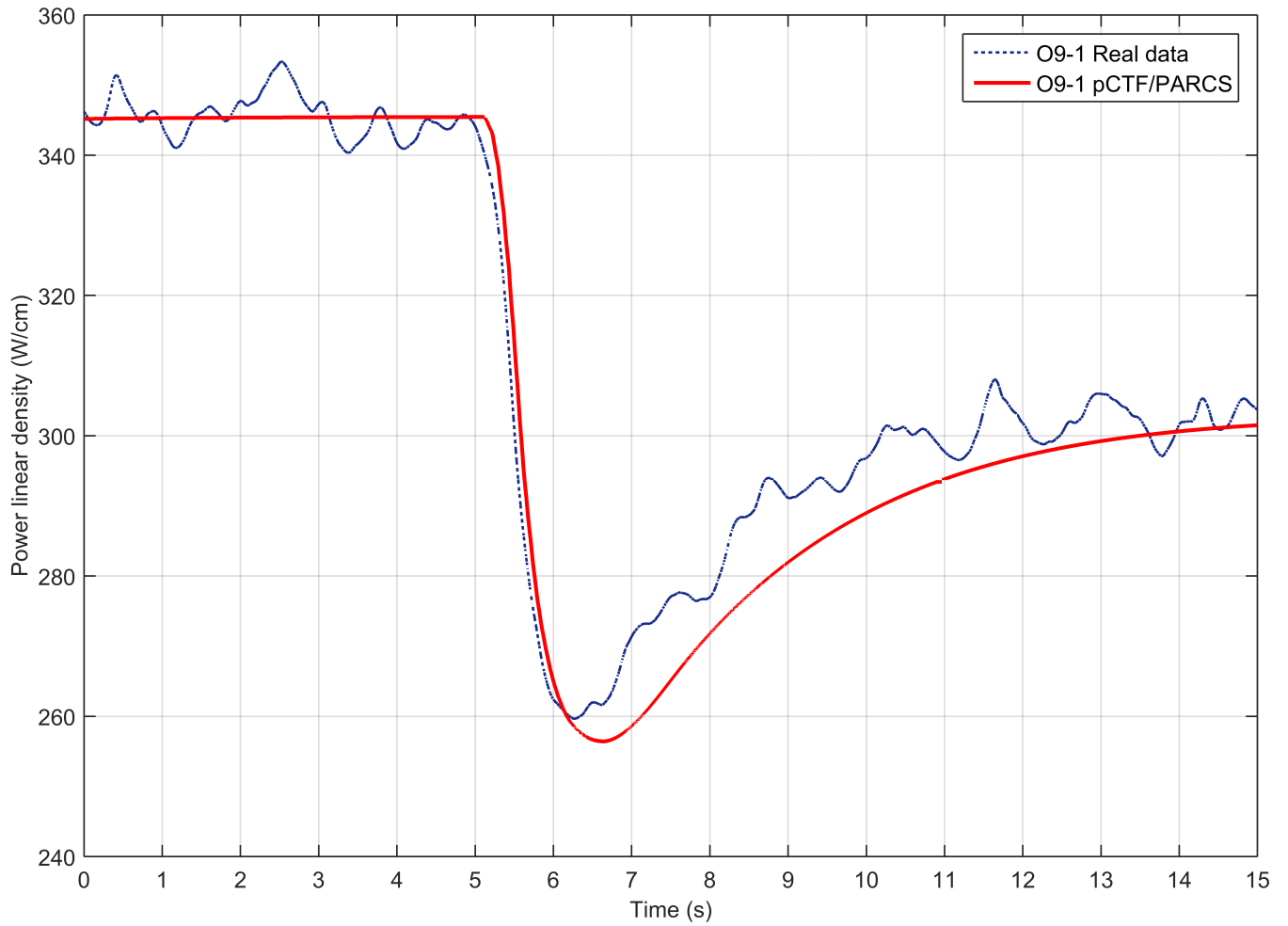


Fig. 16: Real versus simulated signal for incore detector O9 at axial level 1

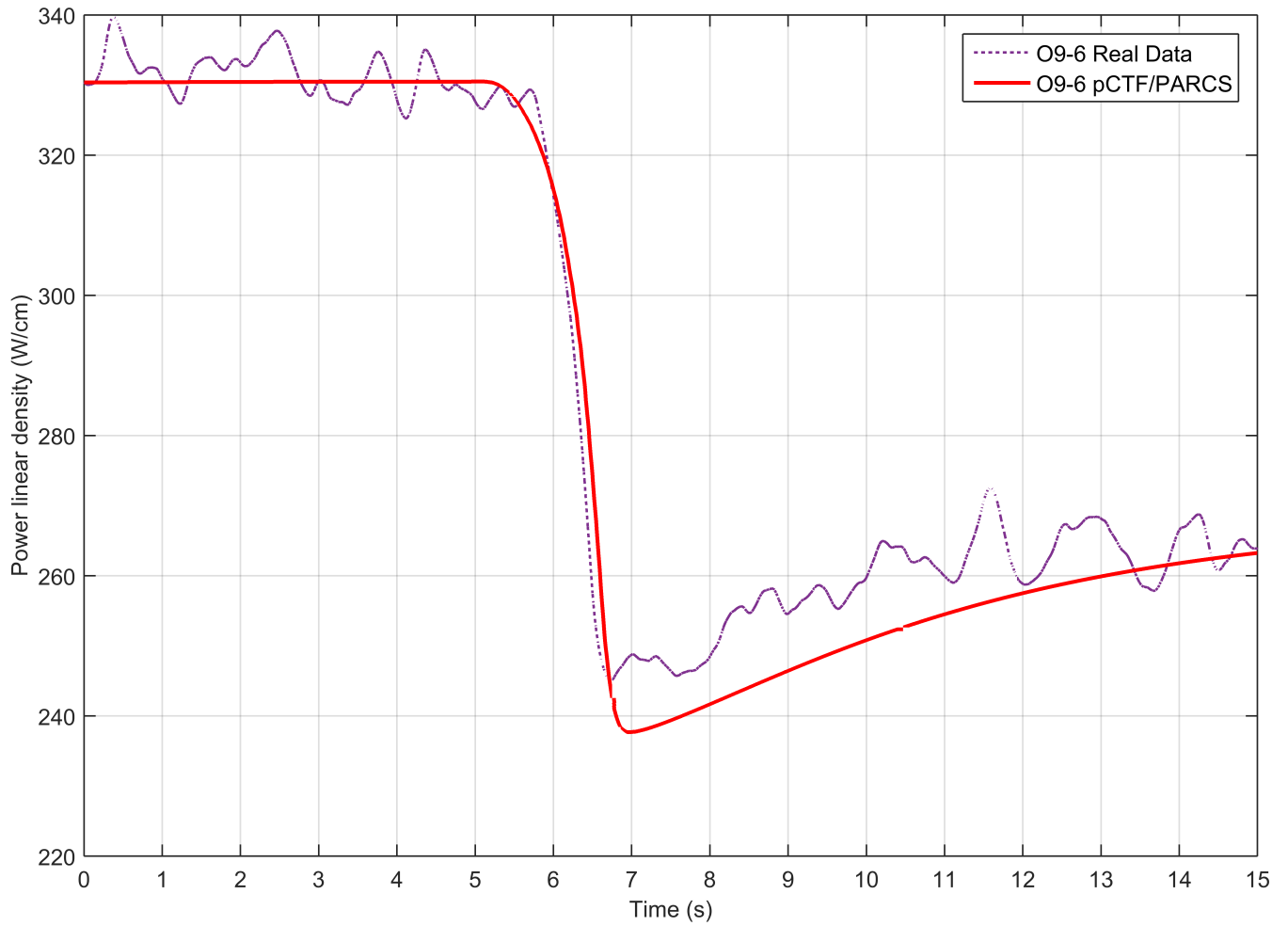


Fig. 17: Real versus simulated signal for incore detector O9 at axial level 6

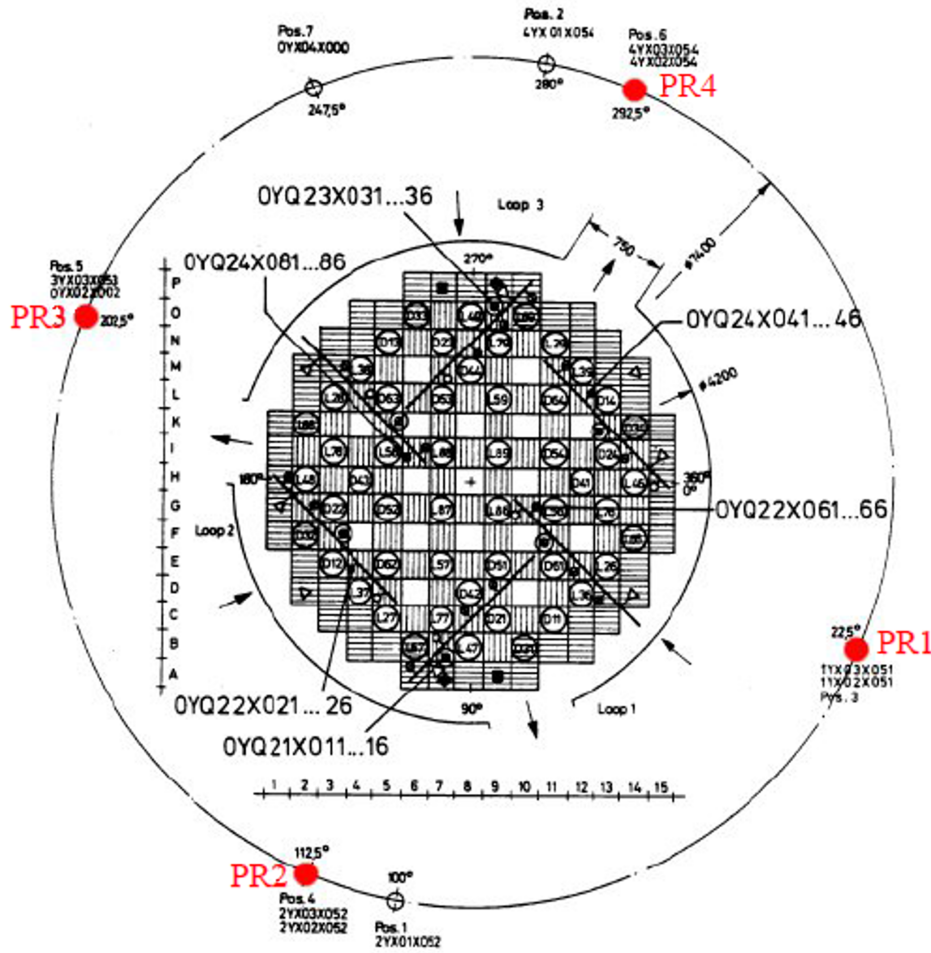


Fig. 18: Scheme of the radial location of the Excore detectors

The 16 Excore detectors (boron-lined ionization chambers) are placed at the biological shield of the containment building of the reactor. Then, a transport model is needed to compute their simulated signal from the core kinetics response. In Figure 18, the 4 radial positions for the Excore detectors are shown, marked as PR1, PR2, PR3 and PR4. Each position has 2 redundancies (bottom and top), and each redundancy has 2 chambers connected to form one measuring channel.

The Excore detectors response is simulated with PARCS by applying a simplified radial transport model described in Equations (2) and (3).

$$\phi(r) \approx 1/r \cdot \exp(-\Sigma_r \cdot r) \quad (2)$$

$$\phi(r) \approx 1/r^2 \quad (3)$$

Equation (2) describes the neutron transport from each fuel assembly to the exterior vessel surface. Equation (3) describes pure geometric transport from the vessel surface to the detectors location. The variable  $r$  is the distance, and the constant macroscopic cross section for the medium used in Equation (2) is given as  $0.115 \text{ cm}^{-1}$ .

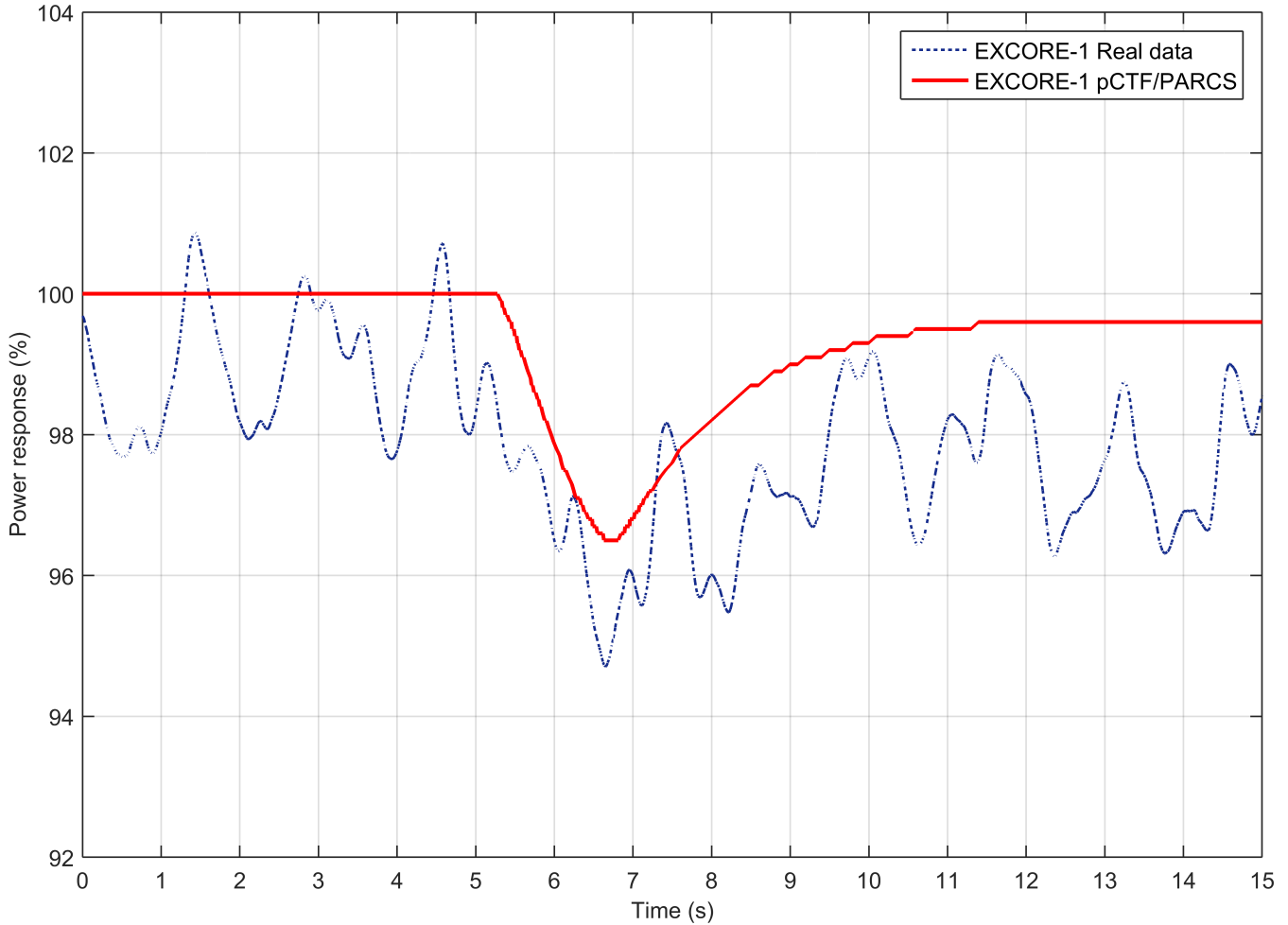


Fig. 19: Real versus simulated signal for excore detector 1

After the simulation, the total flux response from the simulated Excore detectors are normalized with their steady state values. The objective of this normalization is to rescale the flux response of the same way that the nuclear power plant instrumentation to compare the simulated signal with the real measurements.

Figures 19 to 22 show the comparison of the simulated Excore detectors (from PR1 to PR4) with the real plant measurements during the N07 control rod drop transient. As can be seen the obtained responses have very good agreement with the measurements, and then the simplified transport theory applied to obtain the Excore signals can be considered valid. The results from the Excore detector signal confirm that the parallel coupled code pCTF/ PARCSv3.2 can simulate properly control rod drop transient in PWR nuclear reactors.

## VI. CONCLUSIONS

We have developed a fully 3D pin-by-pin coupled thermal-hydraulic/neutronic model of a whole PWR core with three loops and used it with the parallel coupled code pCTF/PARCS to compute the pin power employing the pin power reconstruction capability of PARCS. The original coupling algorithm has been modified with the goal of enabling that pCTF was fed with the

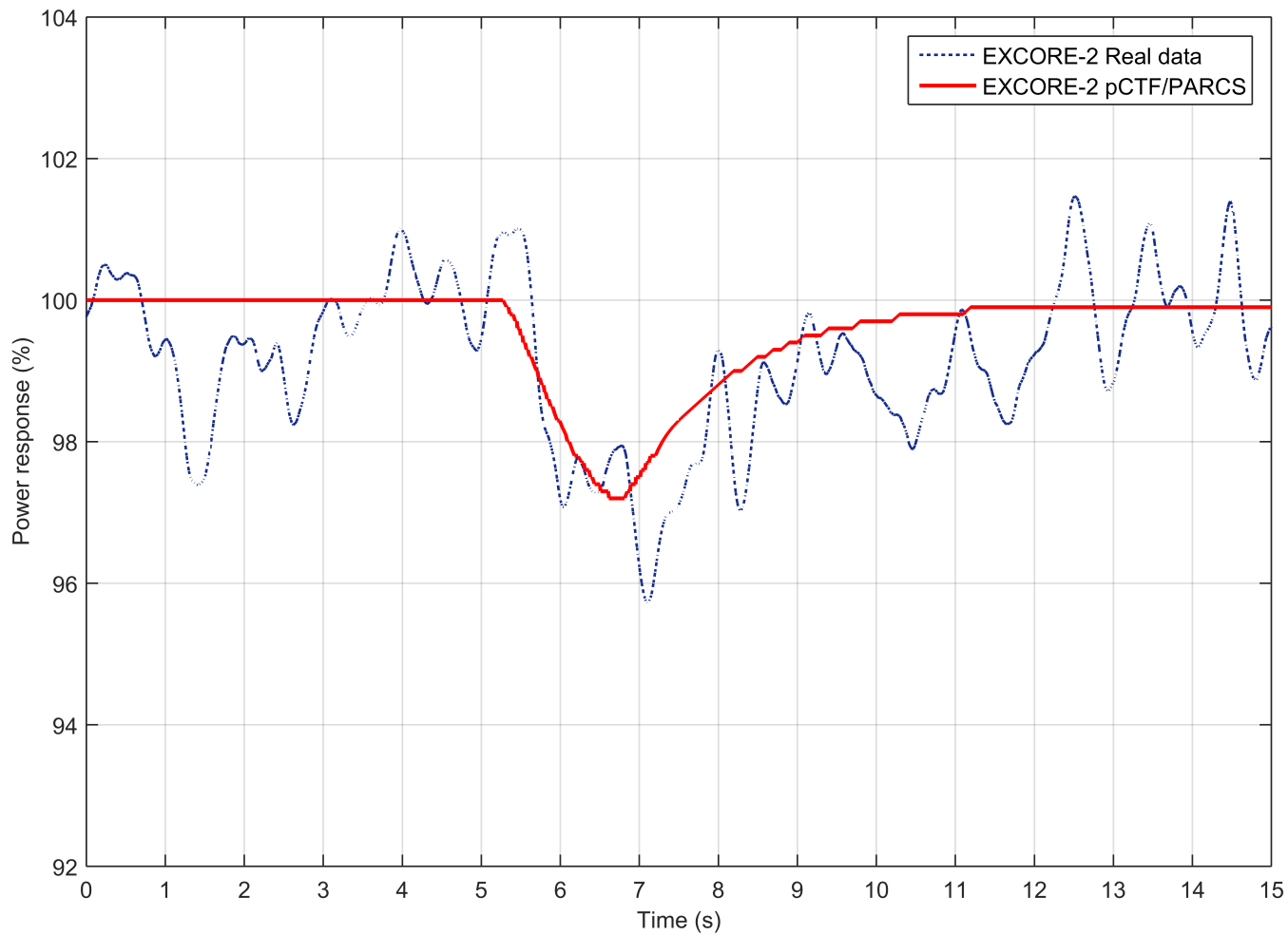


Fig. 20: Real versus simulated signal for excore detector 2

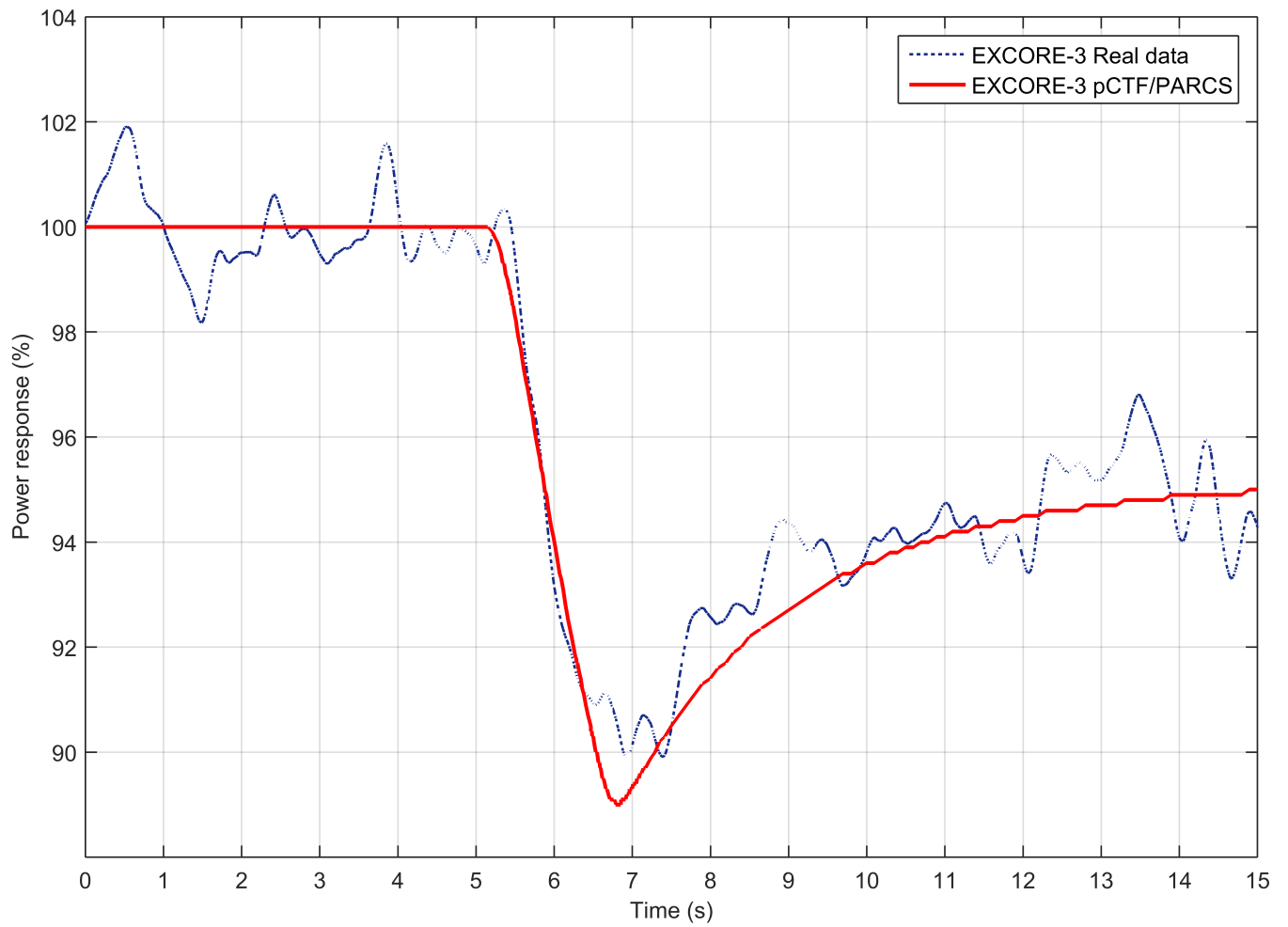


Fig. 21: Real versus simulated signal for excore detector 3

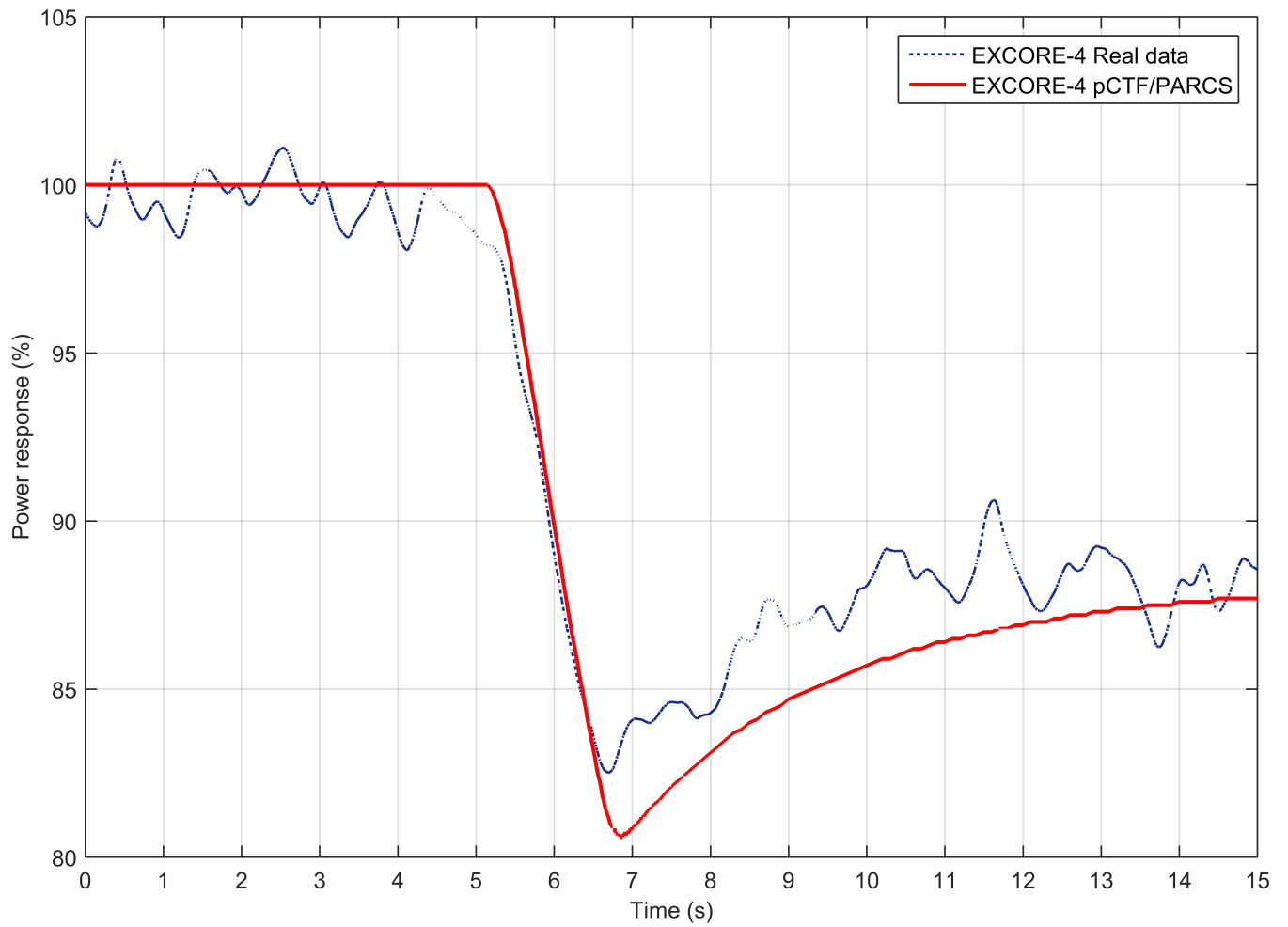


Fig. 22: Real versus simulated signal for excore detector 4



pin powers from PARCS.

The pCTF code has been benchmarked against two RIA scenarios. Results of the simulation have been contrasted with real data coming from control rod drop trials performed at the NPP, showing good agreement. Hence the models are assessed against control rod drop transients. The power decrease during the control rod drop has a similar magnitude for all the analyzed PDDs and Excore detectors, and the evolution of the transient matches the reference data.

Having a parallel version of the COBRA-TF code allows us to carry out analyses with much more level of detail, with significantly shorter simulation times. This opens the door to studying scenarios that were not considered before.

## ACKNOWLEDGMENTS

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 Economía y Competitividad under Projects SLEPc-HS (TIN2016-75985-P) and NUC-MULTPHYS (ENE2012-34585).

## 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. 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.
5. A. Abarca, et al., A procedure for coupled thermal-hydraulic subchannel and neutronic codes using COBRA-TF and PARCS, Tech. rep., INAC (2011).
6. E. Ramos, J. E. Roman, A. Abarca, R. Miró, J. A. Bermejo, Control rod drop transient analysis with the coupled parallel code pCTF-PARCSv2.7, *Annals of Nuclear Energy* 87 (2) (2016) 308–317.
7. S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, K. Rupp, B. Smith, S. Zampini, H. Zhang, PETSc users manual, Tech. Rep. ANL-95/11 - Revision 3.6, Argonne National Laboratory (2015).
8. 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*, Birkhäuser, 1997, pp. 163–202.
9. MPI Forum, MPI: a message-passing interface standard, *Int. J. Supercomp. Applic. High Perf. Comp.* 8 (3/4) (1994) 159–416.
10. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd Edition, SIAM Publications, 2003.

11. 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.
12. T. Downar, D. Lee, Y. Xu, T. Kozlowski, J. Staundenmier, PARCS v. 2.7 US NRC Core Neutronics Simulator: User manual, Tech. rep. (2006).
13. T. Downar, D. Lee, Y. Xu, T. Kozlowski, J. Staundenmier, PARCS v. 2.7 US NRC Core Neutronics Simulator: Theory manual, Tech. rep. (2006).