

4 DESCRIPTION OF ADOPTED CODES

4.1 RELAP5/ MOD 3.3 CODE

The light water reactor (LWR) transient analysis code, RELAP5, was developed at the Idaho National Engineering Laboratory (INEL) for the U.S. Nuclear Regulatory Commission (NRC). Code objectives include analyses required to support rulemaking, licensing audit calculations, evaluation of accident mitigation strategies, evaluation of operator guidelines, and experiment planning analysis. RELAP5 has also been used as the basis for a nuclear plant analyzer. Specific applications have included simulations of transients in LWR systems such as loss of coolant, anticipated transients without scram (ATWS), and operational transients such as loss of feedwater, loss of offsite power, station blackout, and turbine trip. RELAP5 is a highly generic code that, in addition to calculating the behaviour of a reactor coolant system during a transient, can be used for simulation of a wide variety of hydraulic and thermal transients in both nuclear and non-nuclear systems involving mixtures of steam, water, non-condensable, and solute.

The RELAP5/MOD3 code is based on a non-homogeneous and non-equilibrium model for the two-phases that is solved by a fast, partially implicit numerical scheme to permit economical calculation of system transients. The objective of the RELAP5 development from the outset was to produce a code that included important first-order effects necessary for accurate prediction of system transients but that was sufficiently simple and cost effective so that parametric or sensitivity studies were possible.

The code includes many generic component models from which general systems can be simulated. The component models include pumps, valves, pipes, heat releasing or absorbing structures, reactor point kinetics, electric heaters, jet pumps, turbines, separators, accumulators, and control system components. In addition, special process models are included for effects such as form loss, flow at an abrupt area change, branching, choked flow, boron tracking, and non-condensable gas transport.

The system mathematical models are coupled into an efficient code structure. The code includes extensive input checking capability to help the user discover input errors and inconsistencies. Also included are free-format input, restart, renodalization, and variable output edit features. These user conveniences were developed in recognition that generally the major cost associated with the use of a system transient code is in the engineering labor and time involved in accumulating system data and developing system models, while the computer cost associated with generation of the final result is usually small.

The development of the models and code versions that constitute RELAP5 has spanned approximately 17 years from the early stages of RELAP5 numerical scheme development to the present. RELAP5 represents the aggregate accumulation of experience in modeling reactor core behaviour during accidents, two-phase flow processes, and LWR systems. The code development has benefited from extensive application and comparison to experimental data in the LOFT, PBF, Semiscale, ACRR, NRU, and other experimental programs.

4.1.1 Development of RELAP5/MOD 3.3

The MOD3 version of RELAP5 has been developed jointly by the NRC and a consortium consisting of several countries and domestic organizations that were members of the International Code Assessment and Applications Program (ICAP) and its successor organization, Code Applications and Maintenance Program (CAMP). In addition, improvements have been made on behalf of several Department of Energy sponsors. The mission of the RELAP5 development program was to develop a code version suitable for the analysis of all transients and postulated accidents in LWR systems, including both large and small-break loss-of-coolant accidents (LOCAs) as well as the full range of operational transients.

RELAP5/MOD3 was produced by improving and extending the modelling base that was established with the release of RELAP5/MOD2 [17], [18], [19], [20] in 1985. Code deficiencies identified by members of ICAP and CAMP

through assessment calculations were noted, prioritized, and subsequently addressed. Consequently, several new models, improvements to existing models, and user conveniences have been added to RELAP5 Mod3.3.

4.1.2 Code architecture

RELAP5 Mod 3.3 is written in FORTRAN77 for a variety of 64-bit and 32-bit computers.

RELAP5 is coded in a modular fashion using top-down structuring. The various models and procedures are isolated in separate subroutines. The top level structure is shown in figure 4.1 and consists of input (INPUT), transient/steady state (TRNCTL), and stripping (STRIP) blocks.

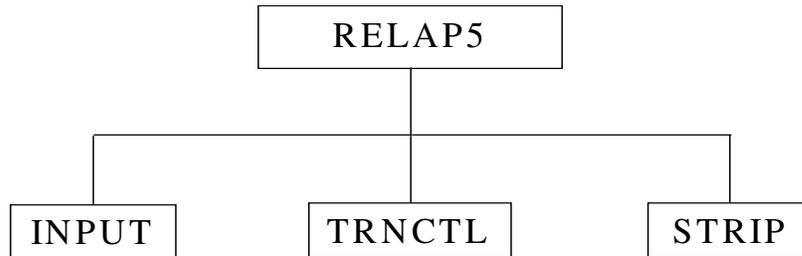


Figure 4.1: RELAP 5 top level structure

The input block (INPUT) processes input, checks input data, and prepares required data blocks for all program options.

The transient/steady state block (TRNCTL) handles both transient and the steady state options. The steady state option determines the steady state conditions if a properly posed steady state problem is presented. Steady state is obtained by running an accelerated transient until the time derivatives approach zero. Thus, the steady state option is very similar to the transient option but contains convergence testing algorithms to determine satisfactory steady state, divergence from steady state, or cyclic operation. If the transient technique alone were used, approach to steady state from an initial condition would be identical to a plant transient from that initial condition. Pressures, densities, and flow distributions would adjust quickly, but thermal effects would occur more slowly. To reduce the transient

time required to reach steady state, the steady state option artificially accelerates heat conduction by reducing the thermal capacity of the conductors.

The strip block (STRIP) extracts simulation data from a restart plot file for convenient passing of RELAP5 simulation results to other computer programs.

RELAP5 provides detailed input checking for all system models using three input processing phases.

- The first phase reads all input data, checks for punctuation and typing errors (such as multiple decimal points and letters in numerical fields), and stores the data keyed by card number such that the data are easily retrieved. A list of the input data is provided, and punctuation errors are noted.
- During the second phase, restart data from a previous simulation are read if the problem is a RESTART type, and all input data are processed. In a NEW-type problem, dynamic blocks must be created. In RESTART problems, dynamic blocks may be created, deleted, added to, partially deleted, or modified as modeling features and components within models are added, deleted, or modified. Extensive input checking is done, but at this level, checking is limited to new data from the cards being processed. Relationships with other data cannot be checked because the latter may not yet be processed.
- The third phase of processing begins after all input data have been processed. Since all data have been placed in fixed common or dynamic data (common) blocks during the second phase, complete checking of interrelationships can proceed. The initialization required to prepare the model for the start of the transient advancement is done at this level.

Figure 4.2 shows the functional modular structure for the transient calculations, while figure 4.3 shows the second-level structures for the transient/steady state blocks or subroutines.

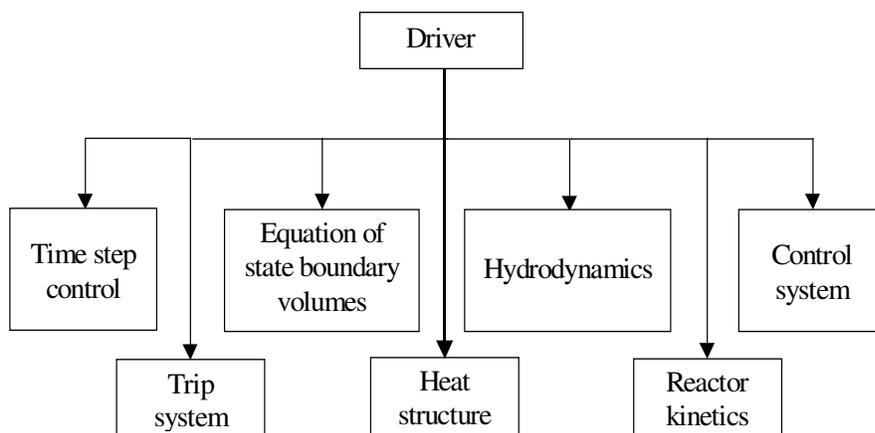


Figure 4.2: Modular structures of transient calculation in RELAP 5

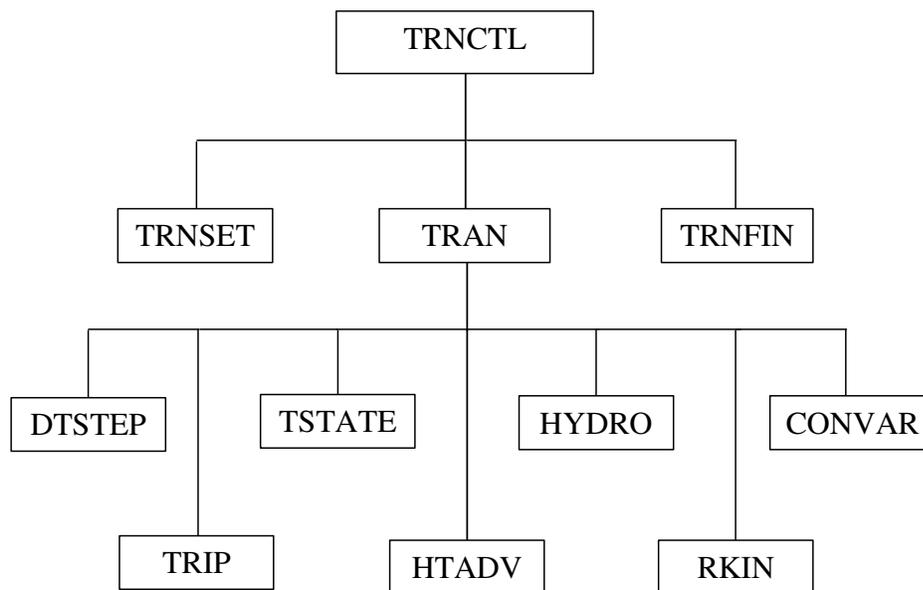


Figure 4.3: Transient/Steady state block structure

The subroutine TRNCTL shown in figure 4.3 consists only of the logic to call the next lower level routines. Subroutine TRNSET performs final cross-linking of information between data blocks, sets up arrays to control the sparse matrix solution, establishes scratch work space, and returns unneeded computer memory. Subroutine TRAN, the driver, controls the transient advancement of the solution. Nearly all the execution time is spent in this block, and this block is the most demanding of memory. Nearly all the dynamic data blocks must be in the

central memory, and the memory required for instruction storage is high, since coding to advance all models resides in this block. When transient advances are terminated, the subroutine TRNFIN releases space for the dynamic data blocks that are no longer needed. A description of the functions of all of the modules (subroutines) driven by TRAN is provided in [21].

4.1.3 Hydrodynamic model

The RELAP5 hydrodynamic model is a one-dimensional, transient, two-fluid model for flow of a two-phase steam-water mixture that can contain non-condensable components in the steam phase and/or a soluble component in the water phase.

The RELAP5 hydrodynamic model contains several options for invoking simpler hydrodynamic models. These include homogeneous flow, thermal equilibrium, and frictionless flow models. These options can be used independently or in combination. The homogeneous and equilibrium models were included primarily to be able to compare code results with calculations from the older codes based on the homogeneous equilibrium model.

The two-fluid equations of motion that are used as the basis for the RELAP5 hydrodynamic model are formulated in terms of volume and time-averaged parameters of the flow. Phenomena that depend upon transverse gradients, such as friction and heat transfer, are formulated in terms of the bulk properties using empirical transfer coefficient formulations. In situations where transverse gradients cannot be represented within the framework of empirical transfer coefficients, such as subcooled boiling, additional models specially developed for the particular situation are employed. The system model is solved numerically using a semi-implicit finite-difference technique. The user can select an option for solving the system model using a nearly-implicit finite-difference technique, which allows violation of the material Courant limit¹⁰. This option is suitable for steady state calculations and for slowly varying, quasi-steady transient calculations.

The basic two-fluid differential equations possess complex characteristic roots that give the system a partially elliptic character and thus constitute an ill-posed initial boundary value problem. In RELAP5, the numerical problem is rendered well-posed by the introduction of artificial viscosity terms in the difference equation formulation that damp the high frequency spatial components of the solution. The ill-posed character of the two-fluid model is a result of the spatial averaging process and neglect of higher-order physical effects in the momentum formulation.

The semi-implicit numerical solution scheme uses a direct sparse matrix solution technique for time step advancement. It is an efficient scheme and results in an overall grind time per node on the CRAY XMP/24 of ~0.00053 seconds, on the DEC Alpha 3000 of ~0.00057 seconds, and on the DECstation 5000 of ~0.00259 seconds. The method has a material Courant time step stability limit. However, this limit is implemented in such a way that single-node Courant violations are permitted without adverse stability effects. Thus, single small nodes embedded in a series of larger nodes will not adversely affect the time step and computing cost. The nearly-implicit numerical solution scheme also uses a direct sparse matrix solution technique for time step advancement. This scheme has a grind time that is 25 to 60% greater than the semi-implicit scheme but allows violation of the material Courant limit for all nodes.

The RELAP5 thermalhydraulic model solves eight field equations for eight primary dependent variables. The primary dependent variables are pressure (P), phasic specific internal energies (U_g , U_f), vapor volume fraction (void fraction) (α_g), phasic velocities (v_g , v_f), non-condensable quality (X_n), and boron density (ρ_b). The independent variables are time (t) and distance (x). Non-condensable quality is defined as the ratio of the non-condensable gas mass to the total gaseous phase mass, i.e.,

¹⁰ i.e. if the nearly-implicit scheme is employed to solve the field equations, the time step used in the scheme can be greater than the material Courant limit

$$X_n = \frac{M_s}{(M_s + M_n)} \quad (4.1)$$

where M_n is the mass of non-condensable in the gaseous phase and M_s is the mass of the steam in the gaseous phase.

The secondary dependent variables used in the equations are phasic densities, phasic temperatures (T_g , T_f), saturation temperature (T_s), and non-condensable mass fraction in non-condensable gas phase (X_{ni}) for the i -th non-condensable species, i.e.,

$$X_{ni} = \frac{M_{ni}}{\sum_{i=1}^N M_{ni}} = \frac{M_{ni}}{M_n} \quad (4.2)$$

where M_{ni} is the mass of the i -th non-condensable in the gaseous phase, M_n is the total mass of non-condensable gas in the gaseous phase and N is the number of non-condensables.

The basic field equations for the two-fluid non-equilibrium model consist of two phasic continuity equations, two phasic momentum equations, and two phasic energy equations. The equations are recorded in differential stream tube form with time and one space dimension as independent variables and in terms of time and volume-average dependent variables.

The semi-implicit solution scheme is based on replacing the system of differential equations with a system of finite-difference equations partially implicit in time. The method has a material Courant time step stability limit. However, this limit is implemented in such a way the single-node Courant violations are permitted without adverse stability effects. Thus, single small nodes embedded in a series of larger nodes will not adversely affect the time step and computing cost.

For problems where the flow is expected to change very slowly with time, it is possible to obtain adequate information from an approximate solution based on very large time steps. This would be advantageous if a reliable and efficient means could be found for solving difference equations treating all the terms by implicit differences. Unfortunately, the state of the art is less satisfactory here than in the case of semi-implicit schemes.

To reduce the number of calculations required for solving fully implicit difference schemes, fractional step methods have been tried. This is the basic idea in the nearly-implicit scheme.

4.1.4 Heat structure models

Heat structures provided in RELAP5 allow the calculation of the heat transferred across solid boundaries of hydrodynamic volumes. Modeling capabilities of heat structures are general and include fuel pins or plates with nuclear or electrical heating, heat transfer across steam generator tubes, and heat transfer from pipe and vessel walls. Heat structures are assumed to be represented by one-dimensional heat conduction in rectangular, cylindrical, or spherical geometry. Surface multipliers are used to convert the unit surface of the one-dimensional calculation into the actual surface of the heat structure. Temperature dependent thermal conductivities and volumetric heat capacities are provided in tabular or functional form either from built-in or user-supplied data.

Finite differences are used to advance the heat conduction solutions. Each mesh interval may contain different mesh spacing, different material, or both. The spatial dependence of the internal heat source may vary over each mesh interval. The time-dependence of the heat source can be obtained from reactor kinetics, one of several tables of power versus time, or a control system variable. Boundary conditions include symmetry or insulated conditions, a correlation package, tables of surface temperature versus time, heat transfer rate versus time, and heat transfer coefficient versus time or surface temperature. The heat transfer correlation package can be used for heat structure surfaces connected to hydrodynamic volumes, and contains correlations for convective, nucleate boiling, transition

boiling, and film boiling heat transfer from the wall to water and reverse transfer from water to wall including condensation.

Temperature distributions in heat structures are assumed to be represented adequately by a one dimensional form of the transient heat conduction equation in rectangular, cylindrical, or spherical coordinates. The spatial dimension of the calculation is along any one of the coordinates in rectangular geometry and is along the radial coordinate in cylindrical or spherical geometry. The one-dimensional form assumes no temperature variations along the other coordinates.

Finite differences are used to advance the heat conduction solution. Each mesh interval may contain different mesh spacing, different material or both. For each material specified, corresponding thermal property data must be entered to define the thermal conductivity and the volumetric heat capacity as functions of temperature; the temperature-dependence can be described by tabular data or by a set of functions.

Heat structures can have an internal volumetric heat source that can be used to represent nuclear, gamma or electrical heating. The source is assumed to be a separable function of space and time. The space function is assumed to be constant over a mesh interval but may vary from mesh interval to mesh interval. The time function may be: total reactor power, fission power or fission product decay power from the reactor kinetics calculation; a control variable; or may be obtained from a table of power versus time.

For each heat structure, boundary condition input specifies the type of boundary condition, the possible connection of a heat structure surface to a hydrodynamic volume and the relation of the one-dimensional heat conduction solution to the actual three-dimensional nature of the structure. Boundary conditions include: symmetry or insulated conditions; a set of heat transfer correlations that cover the various modes of the heat transfer from a surface to fluid (convection, nucleate boiling, transition boiling, film boiling) and vice versa (including condensation); tables of surface temperature versus time; heat transfer

rate versus time heat transfer coefficient versus time or versus surface temperature.

A user option allows an explicit or an implicit coupling between the heat conduction-transfer and hydrodynamic time advancements. With the explicit option, the changes in hydrodynamics temperatures are assumed zero at this point and the new time temperatures are given by equations. If the implicit advancement is used, the final temperatures are computed after the fluid temperatures are computed.

4.1.5 Trip system

The trip system consists of the processing of logical statements. Each trip statement is a simple logical statement that has a true or false result and an associated variable, TIMEOF. The TIMEOF variable is -1.0 whenever the trip is false, and contains the time the trip was last set true whenever the trip is true. This variable allows for time delays and unit step functions based on events during the transient.

Within the structure of RELAP5, the trip system is considered to be only the evaluation of the logical statements. The decision of what action is needed, based on trip status, resides within other models. For example, valve models are provided that open or close the valve based on trip values; pump models test trip status to determine whether a pump electrical breaker has tripped.

Two types of trip statements are provided: variable and logical trips. Since logical trips involve variable trips and other logical trips, complex logical expressions can be constructed from simple logical statements. Both types of trips can be latched or unlatched. A latched trip, once set true, is no longer tested and remains true for the remainder of the problem or until reset at a restart. An unlatched trip is evaluated every time step.

4.1.6 Control system

The control system provides the capability to evaluate simultaneous algebraic and ordinary differential equations. The capability is primarily intended to simulate control systems typically used in hydrodynamic systems, but it can also model other phenomena described by algebraic and ordinary differential equations. Another use is to define auxiliary output quantities, such as differential pressures, so they can be printed in major and minor edits and be plotted.

The control system consists of several types of control components. Each component defines a control variable as a specific function of time-advanced quantities. The time-advanced quantities include hydrodynamic volume, junction, pump, valve, heat structure, reactor kinetics, trip quantities, and the control variables themselves (including the control variable being defined). This permits control variables to be developed from components that perform simple, basic operations.

4.1.7 Special techniques

The mass from the state relationship is compared to the mass from the continuity equation, and the difference is a measure of the truncation error inherent in the numerical solution. This is the main method used to control the time step and thus control the truncation error. Other methods are also used.

Special techniques are also used to mitigate mass and energy errors. These are:

- 1) a second evaluation of the semi-implicit scheme equations using non-linearized time derivatives,
- 2) velocity flip-flop situations,
- 3) non-condensable gas appearance situations.

Special methods are provided in the code for use in obtaining initial conditions. Generic control component options are available to allow the user to minimize the time, effort, and cost to achieve steady state.

4.2 PARCS 2.4 CODE

4.2.1 Introduction

PARCS is a three-dimensional (3-D) reactor core simulator which solves the steady state and time dependent neutron diffusion equations to predict the eigenvalue and the dynamic response of the reactor to reactivity perturbations such as control rod movements or changes in the temperature/fluid conditions in the reactor core. The code is applicable to both PWR and BWR cores loaded with either *rectangular* or *hexagonal* fuel assemblies. The neutron diffusion equation is solved with two energy groups for the rectangular geometry option, whereas any number of energy groups can be used for the hexagonal geometry option. PARCS is coupled directly to the thermalhydraulics systems codes TRAC-M and RELAP5 which provide the temperature and flow field information to PARCS during the transient. The thermalhydraulic solution is incorporated into PARCS as a feedback into the few group cross-sections. The *coarse mesh finite difference* (CMFD) formulation is employed in PARCS to solve for the neutron fluxes in the homogenized nodes. In rectangular geometry, the *analytic nodal method* (ANM) is used to solve the two-node problems for accurate resolution of coupling between nodes in the core, whereas the *triangle-based polynomial expansion nodal* (TPEN) method is used for the same purpose in hexagonal geometry.

Since the initial release of the NRC version of PARCS (V1.01) in November 1998 [22], there have been numerous functional improvements and code feature extensions: addition of a *pin power reconstruction* feature, input/output system renovation, modification of TRAC-M coupling routines, *1D kinetics capability* addition, dynamic memory allocation, *automatic thermalhydraulic to neutronic mapping*, a Windows user interface, UNIX on-line graphics, and finally the hexagonal geometry option was implemented initially with only two-group solutions and then was extended to multigroup solutions.

The major calculation features in PARCS include the ability to perform *eigenvalue* calculations, *transient (kinetics)* calculations, *Xenon transient*

calculations, *decay heat* calculations, *pin power* calculations, and *adjoint* calculations. The primary use of PARCS involves a 3-D calculation model for the realistic representation of the physical reactor. However, various one-dimensional (1D) modeling features are available in PARCS to support faster simulations for a group of transients in which the dominant variation of the flux is in the axial direction, as for example in several BWR applications.

Numerous sophisticated spatial kinetics calculation methods have been incorporated into PARCS in order to accomplish the various tasks with high accuracy and efficiency. For example, the CMFD formulation provides a means of performing a fast transient calculation by avoiding expensive nodal calculations at times in the transient when there is no strong variation in the neutron flux spatial distribution. Specifically, a conditional update scheme is employed in PARCS so that the higher order nodal update is performed only when there are substantial changes in the core condition to require such an update. The temporal discretization is performed using the theta method with an exponential transformation of the group fluxes. A transient fixed source problem is formed and solved at each time step in the transient. For spatial discretization, the stabilized ANM two-node kernel or the multigroup TPEN kernel is used to obtain the nodal coupling relation that represents the interface current as a linear combination of the node average fluxes of the two nodes contacting the interface.

The solution of the CMFD linear system is obtained using a Krylov subspace method which utilizes a BILU3D preconditioner in rectangular geometry and a point ILU preconditioner in hexagonal geometry. The eigenvalue calculation to establish the initial steady state is performed using the Wielandt eigenvalue shift method. The pin power calculation method employs a reconstruction scheme in which predefined heterogeneous power form functions are combined with a homogeneous intranodal flux distribution. The homogeneous flux shape is obtained by solving analytically a two-dimensional boundary fixed source problem consisting of the surface average currents specified at the four boundaries.

PARCS was written in FORTRAN90 and its portability has been tested on various platforms and operating systems, which include SUN Solaris Unix, DEC Alpha Unix, SGI Unix, HP Unix, LINUX, and various Windows OS (i.e. 95, 98, NT, and 2000). During the testing, it was determined that there are minor platform dependencies that need to be treated by compiler directives. A typical compiler directive is F90MODULE that must be used on the LINUX machines for dynamic memory allocation using F90 modules instead of the pointer commons which are the basis of the dynamic memory allocation scheme of PARCS.

4.2.2 Calculation features

PARCS is equipped with various calculation modules needed to predict the global and local response of the reactor in steady state and transient conditions. The various features of PARCS are described in this section along with the corresponding modules.

4.2.2.1 Eigenvalue calculation

In order to establish the initial steady state, it is necessary to perform an eigenvalue calculation. PARCS performs the eigenvalue calculation using the Wielandt eigenvalue shift method. The eigenvalue obtained is used to adjust the ν values in the subsequent transient calculation in order to make the initial reactor state critical. In addition to the standard *k-eff calculation* for a given reactor configuration, the *critical boron concentration (CBC) search* function is available. The type of search is defined in the **SEARCH** card of the CNTL block input.

4.2.2.2 Transient (kinetics) calculation

This is the primary function of PARCS that solves the time dependent neutron diffusion equation involving both delayed and prompt neutrons. The transient calculation option is turned on and off by the **TRANSIENT** card in the CNTL block. The temporal differencing based on the *exponential transform* and the *theta method* yields a transient fixed source problem at each time step. The fixed source problem is solved using the Coarse Mesh Finite Difference (CMFD) method in which a *conditional nodal update* scheme is employed. The temporal discretization schemes can be specified by the **THETA** and **EXPO_OPT** cards in

the TRAN input block. Exponential extrapolation can be used to obtain an initial flux guess at each new time step by activating this option in the expo_opt card. The conditional nodal update scheme activates the higher order nodal update only when there are substantial local cross-section changes. The conditional nodal update is controlled by the **EPS_XSEC** card in the TRAN block.

4.2.2.3 Xenon transient calculation

For slow reactor transients, it is essential to provide a Xenon transient capability. In PARCS, the conventional quasi-static treatment of xenon transients is used that employs the eigenvalue problem solver instead of the transient fixed source problem. The number densities of Xenon and Samarium are updated by solving the respective balance equations using the fluxes resulting from the eigenvalue calculation. The Xenon option is controlled in the **XE_SM** card in the CNTL block by choosing one of the following options: 1) *No Xenon*, 2) *Equilibrium Xenon*, or 3) *Transient Xenon*.

4.2.2.4 Decay heat calculation

A simplified decay heat model involving six groups of decay heat precursor groups is employed in PARCS. The *6 group decay heat precursor equation* is treated in the same way as the delayed neutron precursor equation. The solution of the precursor equation is thus nodewise and provides at each time step the decay heat to be summed with the fission power in order to determine the total power produced in each node. Default values of the precursor fraction and decay constant of the 6 groups are provided for UO₂ fueled cores which are operated for a sufficiently long period of time. The option exists for the user to specify alternate values for the precursor fractions and decay constants. The decay heat option is specified in the **DECAY_HEAT** card in the CNTL block and the input parameters are specified in the **DHP_BETA** and **DHP_LAMBDA** cards in the XSEC block.

4.2.2.5 Pin power calculation

The primary dependent variables in PARCS are the node average fluxes and interface currents. In order to obtain local pin power distributions, it is thus necessary to “reconstruct” pin powers. This is performed in PARCS by

multiplying the *heterogeneous power form functions* with the homogeneous intranodal flux distribution. The homogeneous intranodal flux is calculated by performing an *analytic 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. Pin power reconstruction is performed for both the steady state and transient conditions. The pin power calculation option is activated by the **PIN_POWER** card in the CNTL block and the heterogeneous power form functions are supplied via the PFF block. Corner discontinuity factors can be used to enhance the accuracy of the pin power distribution and can be specified in the **CDF** card in the XSEC block. In order to save computing time for the pin power calculation, only certain fuel assemblies can be selected for the pin power calculation in the **PINCAL_LOC** card of the GEOM block. The transient pin power calculation need not be performed at every time step since the pin-to-box factor does not change appreciably unless there is a substantial change in the core configuration. The transient pin power calculation frequency is activated by the **PIN_FREQ** card in the TRAN block.

4.2.3 Modeling features

One of the essential neutronics problems for a reactor core is to represent the physical system with an accurate numerical model. Among the various fundamental modeling issues in the reactor kinetics calculation are the geometric representation, the cross-section representation, and the thermalhydraulics (T/H) feedback modeling. PARCS provides a 3-D geometric representation of the core that can be reduced to 2D, 1D, or 0D by the choice of the appropriate boundary conditions. However, a special 1D kinetics capability is also available for more accurate and versatile 1D modeling. Various geometric representation features will be described in the first and fourth subsections below. The basic cross-section representation scheme in PARCS is to functionalize the macroscopic cross-sections with linear or quadratic dependence on the T/H state variables. The details of the PARCS cross-section representation schemes are provided in the second subsection. For thermalhydraulics, PARCS uses an external T/H solver which greatly extends its range of applicability. The original internal T/H solver

of PARCS is intended primarily for code testing and not for practical applications, and therefore only the external T/H features are detailed in the third subsection. The input description for the internal T/H solver is still provided since PARCS continues to process this block.

4.2.3.1 Geometric representation

In PARCS, the reactor core is modeled by a group of homogeneous computational nodes. Radially, the computational node can be either rectangular or hexagonal. The size of the node is of the order of fuel assembly pitch, while axially the size is 10~30 cm. In the rectangular option, the core geometry is specified in the GEOM block, whereas the block name is changed to GEOMH for the hexagonal option. The core radial configuration is specified using the unit of a fuel assembly by the **RAD_CONF** card in the GEOM block. The radial node size is then specified by the number of subdivisions of the assembly node by the **NEUTMESH_X** and **NEUTMESH_Y** cards. So the *number of nodes per assembly* can be freely chosen as $n_{subx} * n_{suby}$. Normally, one or four nodes per assembly are used for practical calculations. However, it is possible to perform a fine mesh calculation using the geometry input structure. Also by taking the assemblywise configuration as the pinwise configuration, a pin-by-pin heterogeneous core representation is also possible. The refined geometry feature is, however, limited to the rectangular case only. In the hexagonal case, a hexagon is always represented with one node although internally it is divided into 6 *triangular nodes* within the TPEN kernel.

4.2.3.2 Cross-section functionalization

PARCS uses macroscopic cross-sections which can be input in either the two-group or multi-group form using the same input cards. The macroscopic nodal cross-sections are functionalized on boron concentration (B , in ppm), square root of fuel temperature, moderator temperature and densities, void fraction and the effective rodged fractions. Only the *linear dependence* of cross-sections is considered on these state variables except for the moderator density and void fractions for which the quadratic variation is provided. Symbolically, the cross-sections are functionalized as:

$$\begin{aligned} \sum(B, T_f, T_m, D_m, \alpha, \xi) = & \Sigma_0 + a_1(B - B_0) + a_2(\sqrt{T_f} - \sqrt{T_{f0}}) + \\ & + a_3(T_m - T_{m0}) + a_4(D_m - D_{m0}) + a_5(D_m - D_{m0})^2 + a_6\alpha + a_7\alpha^2 + \xi\Delta\Sigma_{CR} \end{aligned} \quad (4.3)$$

Here the effective rodded fraction is defined as the product of the volumetric rodded fraction and the flux depression factor that is computed by the *decusping* routine for the partially rodded node. For Xenon calculations, the Xenon and Samarium microscopic cross-sections are represented in the same form.

Currently, two special benchmark cross-section representation types are available: one for the OECD MSLB (Main Steam Line Break) problem and the other for the OECD PBTT (Peach Bottom-2 Turbine Trip) problem.

4.2.3.3 Thermalhydraulics feedback

PARCS can operate either with an internal T/H model or coupled with other codes (TRAC or RELAP) that provide the T/H model (external T/H).

The time step size used in the system T/H calculation is often selected very small because of numerical stability considerations. Sometimes it is so small that no considerable changes occur in the core T/H condition and performing a neutronic calculation with such a small change would be unnecessary since the flux variation would also be small. In order to improve code efficiency, a *skip factor* can be used in the coupled calculation such that the T/H code calls PARCS based on this user defined frequency. Different skip factors can be specified for the steady state and transient calculations in the **EXT_TH** card. In addition, since the T/H time step size can in some cases be smaller than that specified in PARCS input, a conditional update scheme is implemented in the code such that PARCS performs the spatial kinetics calculation only when the “accumulated time step size” (i.e. the sum of time step sizes sent from the T/H code since the last PARCS update) is greater than or equal to the input PARCS time step size.

4.3 RELAP5/PARCS COUPLING

4.3.1 Features of the RELAP5/PARCS codes

The coupled RELAP5/PARCS code utilizes a General Interface, which manages the mapping of property data and solution variables between thermalhydraulics and spatial kinetics codes. To meet the requirements of this General Interface separate data map routines for both RELAP5 and PARCS are utilized. The RELAP5-Specific Data Map Routine (RDMR) functions as a secondary interface between RELAP5 and the General Interface, and the PARCS-Specific Data Map Routine (PDMR) functions as a secondary interface between PARCS and the General Interface. In this design, RELAP5, the General Interface, and PARCS are executed as separate processes and communicate with each other through the use of message-passing protocols in the Parallel Virtual Machine (PVM) package.

The coupled RELAP5/PARCS code utilizes an internal integration scheme in which the solution of the system and core thermalhydraulics is obtained by RELAP5 and only the spatial kinetics solution is obtained by PARCS. In this scheme, PARCS utilizes the thermalhydraulics solution data (e.g. moderator temperatures/densities and fuel temperatures) calculated by RELAP5 to incorporate appropriate feedback effects into the cross-sections. Likewise, RELAP5 takes the space-dependent powers calculated in PARCS and solves for the heat conduction in the core heat structures.

The temporal coupling of RELAP5 and PARCS is explicit in nature, and the two codes are locked into the same time step. For this implementation, the RELAP5 solution lags the PARCS solution by one time step. Specifically, the advancement of the time step begins with RELAP5 obtaining the solution to the hydrodynamic field equations using the power from the previous time step. The property data obtained from this solution is then sent to PARCS and the power at the current time step is computed.

PARCS is coupled with either TRAC-M or RELAP5 by the **EXT_TH** card in the CNTL block. The coupling between PARCS and the T/H (Thermal-Hydraulic) code is achieved by the interprocess communication protocol, **PVM**. The two processes are loaded in parallel and the PARCS process transfers the nodal power data to the T/H process. The T/H process then sends back the temperature (fuel and coolant) and density data back to the PARCS process. Originally, it was necessary to execute a third intermediate process, **GI**, which manages the data transfer between the two processes. But this process has been integrated into PARCS in the release adopted in the present work so that only two processes are to be run in parallel.

In general, the neutronic node structure is different from the T/H node structure. The difference is reconciled by a mapping scheme. The original mapping was explicit in that the fractions of different T/H nodes belonging to a neutronic node had to be specified in a file called **MAPTAB** for all the neutronic nodes. In order to reduce the user effort to prepare the MAPTAB file, **automatic mapping** schemes were developed for the coupled TRAC-M/PARCS code that take data from both PARCS and TRAC-M input files to generate the mapping information internally.

4.3.2 The MAPTAB file

The MAPTAB file basically allows the association among thermal hydraulic and neutronic nodes (see figure 4.4). Nevertheless through its functions it is possible to set the quantity of bypass associated to each node, to set the reflector properties, the method of calculating Doppler temperature and the trip logics (e.g. SCRAM).

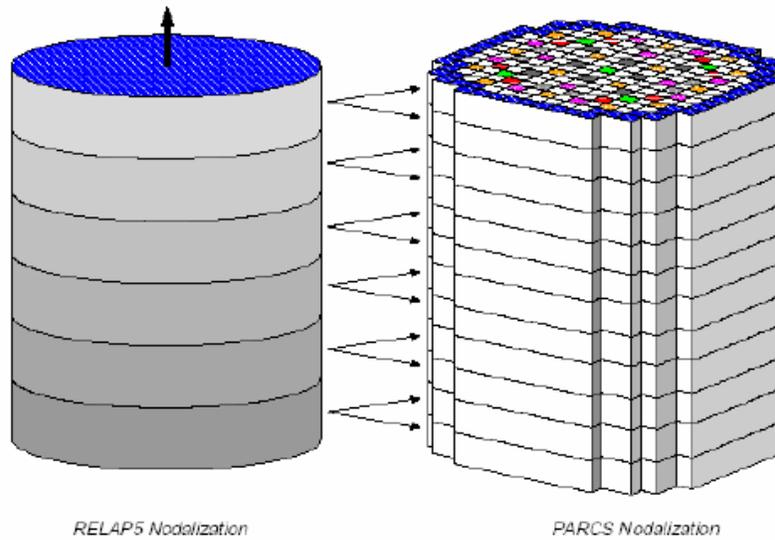


Figure 4.4: Association between thermalhydraulic and neutronic nodalization

The MAPTAB (mapping table). It is composed by five card options: “REFLPROP”, “TABLE1”, “TABLE2”, “TRIP”, and “DOPL”. The function of these cards is the following:

TABLE1: This card relates to the mapping between thermalhydraulic volumes and neutronic nodes. Each line of data which follows is read in free format and contains three numbers (integer, integer, real) corresponding to:

RELAP5 Volume Number PARCS Node Number Weighting Factor

TABLE2: This card relates to the mapping between heat structures and neutronic nodes. Each line of data which follows is read in free format and contains three numbers (integer, integer, real) corresponding to:

RELAP5 Heat Structure Number PARCS Node Number Weighting Factor

TRIP: This card provides the trip unit number used in RELAP5 for detecting control rod scram. Only one line of data needs to be read following this card, and this line, which is read in free format, contains the single integer:

RELAP5 Trip Unit Number

DOPL: This card is used to determine the method of calculating Doppler temperature in the PARCS-Specific Data Map Routine. Only one line of data needs to be read following this card, and there are only two options for calculating Doppler temperature:

LINC *weighting factor*

or

AVG

If “AVG” is read, then the average fuel temperature from RELAP5 will be used and no other data is required. If “LINC” is read, then a linear combination of the centerline and surface fuel temperatures will be used, and an additional weighting factor of type real is required to be input. This weighting factor is applied in the following manner:

$$T_{dopl} = (1 - \omega)T_f^0 + \omega T_f^S \quad (4.4)$$

4.3.3 Calculation methodology

The coupling between PARCS and RELAP5 is external and the calculation is performed via PVM as above mentioned. The methodology to perform calculations is explained hereafter and is illustrated in figure 4.5.

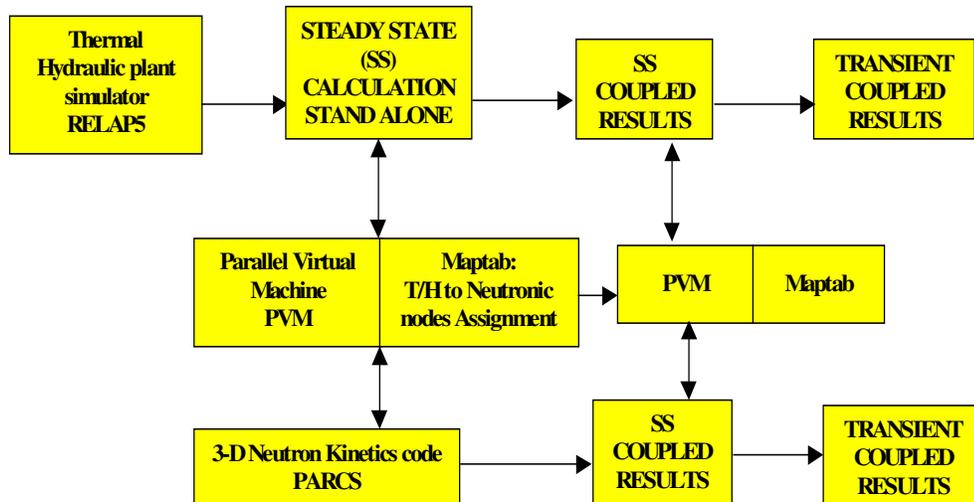


Figure 4.5: RELAP5/PARCS external coupling calculation methodology

PARCS can be executed with either TRAC-M or RELAP5. The user must run two programs simultaneously; the following sequence can be used with either TRAC-M or RELAP:

1. run RELAP in the stand-alone mode for flow initialization (invoking no PARCS calculations) and generate a restart file at the end of the run (RELAP steady state stand alone).
2. using the above restart file run the coupled steady state case and generate the steady state restart files for both PARCS and TRAC-M. Adjust the *external T/H skip factor* so that excessive calls of PARCS are avoided. (RELAP/PARCS coupled steady state).
3. using the restart files, run the coupled transient case (RELAP/PARCS coupled transient).

4.4 VALKIN CODE

4.4.1 Introduction

To simulate the behaviour of a nuclear power reactor it is necessary to be able to integrate the time dependent neutron diffusion equation inside the reactor core.

In this way, a modal method has been developed based on the assumption that the solution for the neutron flux of the studied transient can be expressed approximately as a linear combination of the dominant eigenfunctions of a static auxiliary problem, known as the Lambda modes, associated with a static reactor configuration. To calculate this auxiliary eigenfunctions has been used a nodal collocation method to discretize the equations, and an iterative method to obtain the dominant eigenvalues and their corresponding eigenfunctions.

This nodal modal method has been implemented in a code written in FORTRAN called VALKIN, which calculates the dominant Lambda modes associated with a given configuration of the reactor core, making use of the implicitly restarted Arnoldi method.

This process allows integrating the neutron diffusion equation using a moderate number of modes. This updating process generalizes the quasi-static method for the neutron diffusion equation and allows to decrease the frequency of the actualization of the shape function without loss of accuracy.

The code VALKIN is useful to study transients where the number of necessary eigenmodes is limited; it can also be useful to analyze transients with a physical interpretation for the different amplitudes of the neutron flux expansion, as the ones appearing in the study of in-phase and out-of-phase oscillations of boiling water reactors.

4.4.2 Nodal modal method

Under general assumptions regarding the behavior of the neutron flux in a thermal nuclear power reactor [23], its evolution in time can be modeled by the two-energy groups approximation of the neutron diffusion equations, which in standard notation can be expressed as:

$$[v^{-1}] \dot{\phi} + L\phi = (1 - \beta)M\phi + \sum_{k=1}^K \lambda_k C_k \chi \quad (4.5)$$

$$\dot{C}_k = \beta_k [v\Sigma_{f1} v\Sigma_{f2}] \phi - \lambda_k C_k \quad k = 1, \dots, K \quad (4.6)$$

where it has been considered K groups of neutron precursors and

$$[v^{-1}] = \begin{bmatrix} \frac{1}{v_1} & 0 \\ 0 & \frac{1}{v_2} \end{bmatrix}, \quad \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}, \quad \chi = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$L \text{ is the net loss operator} \quad L = \begin{bmatrix} -\vec{\nabla}(D_1 \vec{\nabla}) + \Sigma_{a1} + \Sigma_{a2} & 0 \\ 0 & -\vec{\nabla}(D_2 \vec{\nabla}) + \Sigma_{a2} \end{bmatrix},$$

$$\text{and } M \text{ is the fission-production operator} \quad M = \begin{bmatrix} v\Sigma_{f1} & v\Sigma_{f2} \\ 0 & 0 \end{bmatrix}.$$

The diffusion constants and cross-sections appearing in the net loss operator (L) depend on the reactor materials, that is, they are position dependent functions.

On the other hand, a reactor core is called critical when the fuel loading is such that the neutron-production rate exactly equals the neutron-loss rate. It is possible to force the criticality dividing the neutron-production rate due to fission by a positive number, λ , in such a way that a physically acceptable solution exist for the equation

$$L\phi = \frac{1}{\lambda} M\phi \quad (4.7)$$

This equation is known as the Lambda Modes equation and it is assumed that there exists an infinite set of positive eigenvalues, λ_n , and their corresponding

eigenfunctions, Φ_n , satisfying the continuity and boundary conditions imposed to the neutronic flux in the reactor core.

The biggest eigenvalue, λ_0 , satisfying the equation (4.7), is associated with the effective multiplication factor of the reactor as $k_{\text{eff}}=\lambda_0$, and its corresponding eigenfunction, Φ_0 , is called the fundamental mode and describes the steady state of the neutron flux in the reactor core.

In order to obtain the kinetic modal expansion equations, it is supposed that the neutron flux can be expanded in terms of the λ -modes as

$$\phi(\vec{x}, t) = \sum_{l=0}^{\infty} n_l(t) \phi_l(\vec{x}) \quad (4.8)$$

where $\phi_l(\vec{x})$ are the Lambda modes of the stationary configuration of the reactor core. That is, they are the eigenfunctions associated with the problem

$$L_0 \psi_l = \frac{1}{\lambda_l} M_0 \psi_l \quad (4.9)$$

where L_0 and M_0 are the loss and production operators of the stationary configuration of the reactor.

Then, from the neutron diffusion equation and the neutron precursor concentration equations using the expansion in these equations and the biorthogonality relation of the eigenfunctions and introducing the feedback reactivities and the delayed feedback reactivities, it has been obtained the modal expansion equations.

A modal interpretation for the in-phase and the out-of-phase oscillations of a BWR reactor has given associating the in-phase oscillation with the amplitude of the fundamental mode of the reactor and interpreting the out-of-phase oscillations as a contribution of the subcritical modes amplitudes to the oscillations.

For a numerical analysis of these oscillations, it is necessary to choose an efficient and fast way of calculating a set of dominant eigenvalues associated with a static configuration of the reactor core and their corresponding eigenfunctions, and also the adjoint eigenfunctions.

The first step to do this consists of discretizing the reactor in N nodes, where the nuclear properties are supposed to be constant. Next, a nodal collocation method [24], which assumes a truncated expansion of the neutron flux in terms of Legendre polynomials, is used. This method permits to discretize the spatial part of the equations, approximating the equations (4.5) and (4.6) by a set of ordinary differential equations (where L and M now are matrix).

$$L\psi = \frac{1}{\lambda} M\psi \quad (4.10)$$

Then, using an iterative method it is obtained the dominant eigenvalues and their corresponding eigenfunctions [25].

4.4.2.1 Modes updating

For realistic transients, the nuclear cross-sections are time dependent functions and to use the modal method proposed above, it would be necessary to calculate a large amount of modes. This is prohibitive from the computational point of view. Thus, it has been adopted some simplifying assumptions: only a set of dominant modes are considered to be sufficient to describe the neutronic flux in BWR instability events¹¹; so it is calculated only a small number of modes and they are updated each certain time step, Δt_i in order to consider the time dependence of the cross-sections. In this way, to integrate the neutron diffusion equation in the time interval $[t_i, t_{i+1}]$, it is used the Lambda modes associated with the problem

$$L^i \psi_i^i = \frac{1}{\lambda_i^i} M^i \psi_i^i \quad (4.11)$$

¹¹ It is important to remark that this expansion will be accurate also taking a small number of modes as far as the Lambda modes of the reactor do not change very much in time.

where L^i and M^i are the matrices associated with the reactor configuration at time t_i .

To obtain the initial conditions for the time integration, VALKIN starts from a critical configuration of the reactor. To achieve this critical reactor configuration the Lambda modes problem has been solved for a given initial configuration.

The thermalhydraulic coupling for the equations obtained is given by the terms including the (dynamical) feedback reactivities and the delayed feedback reactivities, due to the dependence of the nuclear cross-sections on the local distribution of voids and temperatures.

4.5 SIGNAL MODAL DECOMPOSITION

Under oscillating conditions, the neutronic power measured by LPRM's in a reactor core has several contributions: the in-phase or global oscillation associated with the fundamental mode of a stationary configuration of the reactor, and the out-of-phase ones, to be considered as oscillations due to the excitation of subcritical modes through thermohydraulic mechanisms together with the noise inherent in the signal. Therefore, it will be interesting to decompose the LPRM's neutronic signals in its different contributions to study the nature of the oscillation

If the neutronic flux in the reactor core is known, the local neutronic power can be expressed as:

$$P = \alpha(\Sigma_{f1}\phi_1 + \Sigma_{f2}\phi_2) \quad (4.12)$$

The modal methods are based on the expansion of the neutronic flux in terms of Lambda modes, taken as basis functions. Similarly, the power harmonic modes are introduced

$$P_n = \alpha(\Sigma_{f,1}\phi_{1,n} + \Sigma_{f,2}\phi_{2,n}) \quad (4.13)$$

where $\Phi_{g,n}$ $g=1,2$ are the fast and thermal components of the n -th Lambda mode.

The method assumes that the neutronic power (P) can be approximated as a truncated expansion in terms of these power modes

$$P \approx \sum_n a_n P_n \quad (4.14)$$

It is interesting to remark that this expansion will be accurate taking also a small number of modes as far as the Lambda modes of the reactor do not change very much in time.

Supposing to have an ideally monitored reactor core, i.e. to know the neutronic power in each of its points and also to know the fast adjoint Lambda modes, it is possible obtain the amplitudes of the power modes, a_l , which vary in time as

$$a_l(t) = \frac{V}{N_l} \int_R \phi_{l,l} P dx \quad (4.15)$$

and, with these amplitudes, the power decomposition in local modes has been already obtained. Hence, depending on which of these amplitudes oscillate, it is possible classify the instability as in-phase, out-of-phase, or both.

But, the only information available is the neutronic power measured by the LPRM's. Therefore, for each cell, (i, j, k) , of the reactor core discretization, it has been to consider the local harmonic power modes $P_{n,i,j,k}$.

For a given LPRM, l , it has been defined the n -th modal power contribution to the LPRM as

$$LP_{n,l,k} = \sum_{i,j} P_{n,i,j,k} \quad (4.16)$$

where i, j , sum over the adjacent nodes to LPRM l shown in Fig.4.6.

Now, is supposed that the LPRM's signal can be expressed as

$$LPRM_{l,k}(t) = \sum_n a_n^k(t) LP_{n,l,k} \quad (4.17)$$

and analogously to the continuous case, it is used the fast adjoint modes to construct a weighting factor to obtain the power amplitudes $a_n(t)$:

$$a_n(t) = \frac{V}{\bar{N}_n} \sum_{ijk} \phi_{l,ijk,n} P_{ijk}(t) V_{ijk} \quad (4.18)$$

with

$$\bar{N}_n = \sum_{ijk} \phi_{l,ijk,n} (v \Sigma_{f1ijk} \phi_{1,ijk,n} + v \Sigma_{f2ijk} \phi_{2,ijk,n}) V_{ijk} \quad (4.19)$$

V_{ijk} is the volume of the node (i,j,k) , $\Phi_{g,ijk,n}$ is the average flux of group g in node (i,j,k) corresponding to mode n , and $\Phi_{l,ijk,n}$ is the average fast adjoint neutronic flux in node (i,j,k) corresponding to mode n .

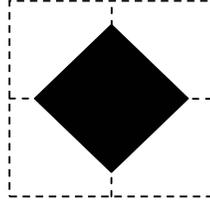


Figure 4.6: Adjacent nodes to LPRM

The analysis of the LPRM signals is based on the in-house codes programmed in Fortran77 developed in the Universidad Politecnica of Valencia [25].

4.6 TIME SERIES ANALYSIS

4.6.1 Basic definitions

With reference to a nuclear reactor, it is important to determine whether the actual operating conditions are inside the stable region (see Chapter 2). In order to study the reactor stability, the neutronic power signals obtained from APRM and LPRM can be used to determine the Decay Ratio.

The Decay Ratio (DR) is a fundamental quantity in a BWR stability analysis and is defined as *the ratio of two consecutive maxima of the impulse response*.

Although the neutron noise cannot be fitted with continuous second order system, the DR definition coming from such a system is given first (section 4.5.1.1) Alternative approaches imply the consideration that the neutron power signals can be considered as time series.

The value of the signal at instant t_n is denoted hereafter as by $\chi(n)$. To study the time series, these are supposed stationary, that is, the main statistical properties of the series remain constant over time. Also, the value of the observed series are assumed as linear combinations of present and past values of the series, and a parametric model can be used to fit their behaviour.

The Decay Ratio parameter is calculated for each of these time series. Basically, two kinds of methods for extracting the DR from neutronic signals are common use, those based on the autocorrelation function (ACF) of the signal and the ones based on the impulse response function (IRF), obtained using a parametric model to fit the behaviour of the system.

4.6.1.1 Analytical definition of DR

If the system that models the signals is a second-order oscillator of the form

$$\ddot{x} + 2\alpha\dot{x} + \omega^2 x = 0 \quad (4.17)$$

the general solution for the system is

$$x(t) = Ae^{-\alpha t} (\cos(\sqrt{\omega^2 - \alpha^2}t + \varphi)) \quad (4.18)$$

The DR parameter gives us a measurement of the damping of the system and it is defined as the ratio between two consecutive maxima of the signal. For the second-order system this parameter is a constant, and is given by

$$DR = e^{-\frac{\alpha 2\pi}{\sqrt{\omega^2 - \alpha^2}}} \quad (4.19)$$

As already mentioned the neutron noise cannot be fitted with a continuous second order system. Consequently the above defined DR should not be considered as a constant. This method it is not used for deriving the value below.

4.6.1.2 Autocorrelation function

The autocorrelation function of a process as the considered one, is defined as the following limit

$$r_{xx}(m) = \lim_{M \rightarrow \infty} \frac{1}{2M+1} \sum_{n=-M}^M x(n+m)x^*(n) \quad (4.20)$$

In practice as a finite number of points for the signal are available, the autocorrelation sequence is rarely known, and must be estimated with the available finite data records. Assuming N data samples, a discrete-time autocorrelation estimate is for the signal x(n):

$$\hat{r}_{xx}(m) = \frac{1}{N-m} \sum_{n=0}^{N-m-1} x(n+m)x^*(n) \quad 0 \leq m \leq N-1 \quad (4.21)$$

$$\hat{r}_{xx}(m) = \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} x(n+|m|)x^*(n) \quad -(N-1) \leq m \leq 0 \quad (4.22)$$

The discrete sequence $\hat{r}_{xx}(m)$ forms unbiased estimates of the true autocorrelation function, because it is satisfied that the expected value of the estimator gives the different points of the autocorrelation function $E[\hat{r}_{xx}(m)] = r_{xx}(m)$.

4.6.1.3 Parametric model

An autoregressive moving-average model for a stationary process, can be defined as

$$x(n) = -\sum_{k=1}^p a(k)x(n-k) + \sum_{k=0}^q b(k)u(n-k) = \sum_{j=0}^{\infty} h_j u(n-j) \quad (4.23)$$

where $x(n)$ can be seen as the output sequence of a causal filter ($h_j=0$ for $j<0$) that models the observed data, h_j is the discrete impulse response function of the system, $u(n)$ is a noise process with zero-mean and constant variance (white noise process) and $a(k)$ and $b(k)$ are the autoregressive (AR) and moving-average (MA) parameters, respectively. The notation ARMA(p,q) is used to indicate an ARMA model with p autoregressive parameters and q moving-average ones.

If all the autoregressive parameters are zero, then equation (4.23) can be written as

$$x(n) = \sum_{k=0}^q b(k)u(n-k) \quad (4.23a)$$

This equation represents a moving-average process of order q , MA(q). And if all the moving-average parameters are zero, except $b(0)=I$, equation (4.23) can be expressed as

$$x(n) = -\sum_{k=0}^q a(k)x(n-k) + u(n) \quad (4.23b)$$

that represents an autoregressive process of order p , AR(p).

Considering the Z-transform of equation (4.23), it is obtained

$$X(z) = H_{ARMA}(z)U(z) \quad (4.24)$$

where $X(z)$ and $U(z)$ are the Z-transform of $x(n)$ and $u(n)$, respectively, and the transfer function of the system is given by

$$H_{ARMA}(z) = \sum_{j=1}^{\infty} h_j z^{j-1} = \frac{\sum_{k=0}^q b(k) z^{-k}}{\sum_{k=1}^p a(k) z^{-k} + 1} \quad (4.25)$$

and if we denote by β_j ($j=1, \dots, p$) the p simple poles of the transfer function and by α_j ($j=1, \dots, q$) the q simple zeros, (4.25) can be written as

$$H_{ARMA}(z) = \sum_{j=1}^p \frac{\omega_j}{(1 - \beta_j z^{-1})} \quad (4.26)$$

where the weights w_j are given by

$$\omega_j = \frac{\prod_{i=1}^q (\beta_j - \alpha_i)}{\prod_{i \neq j}^p (\beta_j - \beta_i)} \beta_j^{(p-q-1)} \quad (4.27)$$

If the above results are introduced in an AR (p) model, the transfer function of the system is

$$H_{AR}(z) = \frac{1}{\sum_{k=1}^p a(k) z^{-k} + 1} = \sum_{j=1}^p \frac{\omega_j}{1 - \beta_j z^{-1}} \quad (4.28)$$

where now, the weights are given by

$$\omega_j = \frac{1}{\prod_{j \neq i}^p (1 - \beta_i / \beta_j)} \quad (4.29)$$

The impulse response function of the system is given by

$$h_j = \sum_{i=1}^p \omega_j \beta_i^j \quad (4.30)$$

To approximate the signals behaviour by means of a parametric model, the orders of the ARMA model are supposed to be known. Thus, to determine the model order constitutes a previous problem to obtain an adequate model for the signal.

To obtain the model order, the Akaike's Final Prediction Error Criterion (FPE) is used. This criterion for a general ARMA(p, q) model, is given by:

$$FPE(k) = \frac{n+k}{n-k} \hat{\rho} \quad (4.31)$$

where $n=N-p$, being N the number of sampled data in a series, $k=p+q+1$, and $\hat{\rho}$ is the residual variance of the process. The basis of the criterion is to use the couple of orders (p, q) at which, the FPE attains its minimum.

4.6.2 Definitions of the Decay Ratio

Other than the analytical definition already given, the DR definitions, presented hereafter, can be used and are used in practical applications.

4.6.2.1 Standard method (DR1 and DR2)

This method proposes two ways of calculating the Decay Ratio. The first one, based on the autocorrelation function (ACF) and the second one, based on the impulse response function (IRF) calculated using an autoregressive moving-average model (ARMA (p, q)) or an autoregressive model (AR(p)) to fit the behaviour of the system. Once the ACF and IRF have been obtained, for a second order system, the DR would be defined as the ratio of two consecutive maxima of these functions (see figure 4.7).

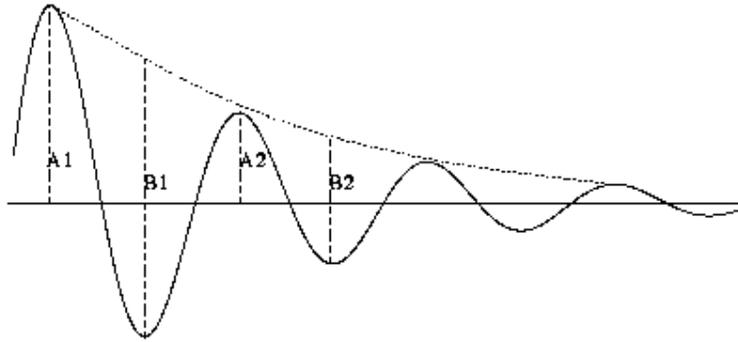


Figure 4.7:Definition of the DR

$$DR = \frac{A_{i+1}}{A_i} \quad (4.32)$$

As a reactor is not strictly a second order system, this definition of DR gives not a constant for the system and so, it is needed to use a new definition of the DR to improve the results. For the first three maxima of the function, the quotients are calculated

$$DR1_i = \frac{A_{i+1}}{A_i}, \quad (i=1,2) \quad (4.33)$$

and the DR parameters of the system, DR1, is defined as the mean value of DR1₁ and DR1₂.

To avoid any disturbance of either the ACF or the IRF, due the influence of the parasitic noise or the time series finiteness, an alternative definition, is considered. For the first three minima of the function, the quotients are calculated

$$DR2_i = \frac{B_{i+1}}{B_i}, \quad (i=1,2) \quad (4.34)$$

and the DR parameter of the system, DR2, is defined as the mean value of DR2₁ and DR2₂. In order to obtain the points B_i, the following steps are considered. The first four maxima of the ACF or the IRF denoted by (x_{Mi}, y_{Mi}) are identified. Once these points are obtained, they are fitted with a cubic polynomial that, using the Lagrange Interpolating Polynomial formula, can be expressed as

$$P(x) = \sum_{i=1}^4 \frac{\prod_{j \neq i}^4 (x - x_{Mj})}{\prod_{j \neq i}^4 (x_{Mi} - x_{Mj})} y_{Mj} \quad (4.35)$$

After this, the first three minima of the ACF of the IRF are identified (x_{mi}, y_{mi}); the image points, P(x_{mi}), of the fitted polynomial can be characterized, and thus, the B_i are given by

$$B_i = P(x_{mi}) + |y_{mi}| \quad (4.36)$$

4.6.2.2 AR-Lyapunov approximation (DR3)

It is known that instabilities take place when two biggest Lyapunov exponents of the system approach simultaneously to zero, which is equivalent to say that the real part of a pair of complex conjugate eigenvalues of the Jacobian matrix of the system approach to zero [26].

It is assumed that the reactor behaviour can be fitted with a second order model with transfer function of the form

$$H(s) = \frac{1}{(s - \gamma)(s - \gamma^*)} \quad (4.37)$$

where γ is the dominant pole of the system and γ^* is its complex conjugate in the continuous s-plane.

To obtain this dominant pole, the dominant pole of the transfer function associated to the parametric model of the signal, β , is considered; this is defined as the system pole with the biggest weight.

Once, it has been determined, it is transformed to the continuous s-plane using the relation with the discrete z-plane, $z = \exp(sT)$. Thus, the dominant pole in the continuous plane can be written as

$$\gamma = \frac{\log(\beta)}{T} = \text{Re}(\gamma) + i \text{Im}(\gamma) \quad (4.38)$$

and the continuous impulse response of the system is given by

$$y(t) = 2 \text{Re} \left[\frac{\gamma^*}{|\gamma - \gamma^*|^2} e^{\text{Re}(\gamma)t} (\cos(\text{Im}(\gamma)t) + i \sin(\text{Im}(\gamma)t)) \right] \quad (4.39)$$

The system oscillation period can be calculated from the imaginary part of γ and is given by

$$L = \frac{2\pi}{\text{Im}(\varphi)} \quad (4.40)$$

And the DR parameter, defined as the quotient between the amplitude of two points separated by L, has the form

$$DR3 = e^{\frac{\text{Re}(\gamma)2\pi}{\text{Im}(\gamma)}} \quad (4.41)$$

4.6.2.3 ARMA-AR model (DR4)

An alternative method to calculate the Decay Ratio, is based on the impulse response obtained from parametric models of the system, either autoregressive or autoregressive moving-average ones [27]. Once the IRF is available, using the expression

$$h_j = \sum_{i=1}^p \beta_i^j \omega_i \quad (4.42)$$

the DR can be calculated as the quotient between two consecutive maxima of the impulse response:

$$DR_j = \frac{h_{L+j}}{h_j} \quad (4.43)$$

where j is the instant corresponding to the maximum of the function and L represents the ‘lag’, or length between two consecutive maxima. Approximating the behaviour of the system by a continuous second order model, L is given by (4.40), where γ is the dominant pole of the transfer function in the continuous s -plane. Using this method, consecutive quotients of the impulse response function in points separated by a ‘lag’ L are taken. Since an asymptotic value is achieved, this value is associated with the Decay Ratio parameter DR4.

4.6.3 Decay Ratio calculations

The results of a computational or experimental BWR system analysis relating to the stability behaviour of this dynamical system are time series of local and global reactor parameters, particularly of the local in-core monitoring: LPRM and APRM.

The task is to analyze the predicted or measured APRM and LPRM time series generated by an arbitrary reactor perturbation. The analysis involves the calculation of the stability characteristic (e.g. DR) from the power time series and possibly other parameter time series.