Direction de l'Energie Nucléaire
Département d'Etudes des Réacteurs
Service de Simulation THermohydraulique
Laboratoire de Développement des Applications pour les Systèmes

BORDEREAU D'ENVOI

Date : 20 mars 2006 Page :

N/Réf : CEA/DEN/GRE/DER/SSTH/LDAS
DO 82 20/03/06

06PPQN000082

diffusé le : 20/03/06

Désignation :
SSTH/LDAS/EM/2005-035
CATHARE 2 V2.5_1 : USER'S MANUAL

Rédacteur : *G. LAVIALLE*

Observation :

B. FAYDIDE
Chef du Service de Simulation
en THermohydraulique

Destinataires :

EDF/SEPTEN :	M. HENRIOT (2 ex)
	Mmes FERRIER*, MALATRAIT*,
	MM. HEMMERICH*, PENTORI*,
	GOY*
EDF/R&D :	Mme SOUYRI*,
	MM. ERHARD*, MARCHAND*
AREVA NP :	M. BARBIER (2 ex)
	MM. GANDRILLE*, SAUVAGE*
TECHNICATOME :	MM. MARIE, ROUX
IRSN/DESTQ (Cad) :	M. REOCREUX*
IRSN/DPAM (Cad) :	Mme CAMOUS (2 ex)
	MM. BARRE*, MARCHAND*,
	Mme GUILLARD*
IRSN/DSDRE/DPER (Far) :	M. de FRANCO*
IRSN/DSR (Far) :	M. DUBOIS (2ex)
	M. DUFEIL*, Mme MESSER*
DEN/DM2S/SFME (Sac)	M. CARUGE*
DEN/DM2S/SFME/LTMF (Sac) :	M. STUDER*
DEN/DM2S/SFME/LETR (Sac)	M. ROYER*, Mme ANIEL*
DEN/DM2S/SFME/LGLS (Sac)	M. DADA*
CS-SI :	Mme EYMARD*

**Résumé uniquement envoyé par courriel*
(le texte complet peut être envoyé sur demande par PDF)

DEN/DSNI :	M. PARADIS*
DEN/DSOE:	M. WILLERMOZ*
DEN/DER/DIR :	M. PORRACCHIA*
DEN/DER/SESI (Cad) :	Mme CHENAUD, M. PIGNATEL,
	MM. ROUAULT*, AUJOLLET*,
	DUMAZ*, MALO*,
	Mme TOSELLO*
DEN/DER/SPRC :	M. GROUILLER*
DEN/DER/SPEX :	Mme MERGUI*
DEN/DER/SRES :	M. CABRILLAT*
DEN/DER/SSTH :	MM. FAYDIDE*, TENCHINE*
DEN/DER/SSTH/LMDL :	MM. DUCROS*, BESTION*
DEN/DER/SSTH/LDAS :	Mmes FARVACQUE, M. LAVIALLE
	Mmes de CRECY, DOR, GEFFRAYE,
	MESSIE, RAMEAU, KADRI
	MM. ANTONI, BAZIN, PARENT,
	BENTIVOGLIO, PIALLA, LYVET,
	GERMAIN P., GERMAIN T., RUBY
	MOREAU, MIEUSSET, MOUTIN,
	MARLIET, PELLISSIER, DUMONT,
	VAN DEN HOVE
DEN/DER/SSTH/LIEX	M. HERVIEU*, Mme NOEL
DEN/DTN/SMTM/LMTR (Cad) :	M. ALBAREDE*
DEN/DTN/SE2T :	M. GARNIER*
DEN/DTN/SE2T/LTGD :	M. CHATAING*

Fichier PDF par courriel

EDF/SEPTEN :	*M. HENRIOT*
AREVA-NP :	*M. BARBIER*
IRSN/DPAM :	*Mme CAMOUS*
IRSN/DSR :	*M. DUBOIS*
DEN/DER/SESI :	*Mme CHENAUD*

Commissariat à l'Energie Atomique
DEN/DER/SSTH/LDAS
17, rue des Martyrs – 38054GRENOBLE Cédex 9
Tel : 04 38 78 50 43 - Secrétariat : 04 38 78 35 93 - Fax : 04 38 78 94 53 – E-mail : martine.farvacque@cea.fr

DIRECTION DE L'ENERGIE NUCLEAIRE

DEPARTEMENT D'ETUDES DES REACTEURS

SERVICE DE SIMULATION EN THERMOHYDRAULIQUE

LABORATOIRE DE DEVELOPPEMENT DES APPLICATIONS POUR LES SYSTEMES

CATHARE

CATHARE V2.5_1 : USER'S MANUAL

G. LAVIALLE

SSTH/LDAS/EM/2005-035 Mars 2006

CEA/DEN/GRE/DER/SSTH/LDAS
DO 78 14/03/06
06PPQN000078
diffusé le : 14/03/06

DIRECTION DE l'ÉNERGIE NUCLÉAIRE
DIRECTION DU CENTRE DE CADARACHE
DÉPARTEMENT D'ÉTUDES DES RÉACTEURS
SERVICE DE SIMULATION EN THERMOHYDRAULIQUE
LABORATOIRE DE DEVELOPPEMENT DES APPLICATIONS POUR
LES SYSTEMES

NT	SSTH/LDAS/EM/2005-035	0	P5H65	A-THACC-02-02		1/212
NATURE	CHRONO UNITÉ	INDICE	UNITÉ	ÉLÉMENT D'OTP	CLASSEMENT UNITÉ	PAGE

Note Technique

TITRE: CATHARE V2.5_1 : User's manual

AUTEUR: G. LAVIALLE

RÉSUMÉ: Ce document est le manuel utilisateur de CATHARE 2 V2.5_1. Il décrit tout d'abord des généralités sur le code (sur la représentation des écoulements, l'approche numérique et les choix de modélisation de CATHARE). Il explique ensuite d'une part comment comprendre et construire un jeu de données CATHARE et d'autre part comment réaliser un calcul et exploiter le résultat obtenu. Enfin, le module CATHACOMB module autonome de CATHARE de calcul combustible est présenté.

ABSTRACT This document is the CATHARE V2.5_1 user manual. General features of the CATHARE code (hydraulic approach, numerical features, modelling features) are given first of all. The manual then goes on to explain on one hand how to understand or to build a CATHARE data file, and on the other hand how to perform a calculation and how to exploit CATHARE calculation results. Finally, the stand-alone fuel computation CATHACOMB is described.

MOTS CLÉS: CATHARE, THERMAL HYDRAULICS, REACTOR SAFETY

Nom du Fichier Informatique: MODNTv12.doc

	Rédacteur	Vérificateur		Approbateur
Fonction				Le Chef du LDAS
visa		Parent		
NOM	G. LAVIALLE	M. PARENT		M. FARVACQUE
Date	14/03/2006	14/03/2006		14/03/2006

Propriété Industrielle	Cadre de réalisation	Classification					Qualité
En l'absence d'accord ou de contrat, la diffusion des informations contenues dans ce document auprès d'un organisme tiers extérieur au CEA est soumise à l'accord du Chef de Département.	DOB.................................... DSOE Domaine .... Simulation Projet....................... THACC-02-02 Contrat...	DR	CC	CD	SD	Sans X	AFAQ ISO 9001 VERSION 2000

Indice	OBJET DES RÉVISIONS	DATE	NATURE et CHRONO UNITÉ
0	Emission initial pour la version CATHARE 2 V2.5	30/06/05	SSTH/LDAS/EM/2004-040
0	Mise à jour pour la version CATHARE 2 V2.5_1 §1 : mise à jour des références §5.3.3.1 : ajout ballonnement §5.4.1 : WALL et WALL3D sub-modules §5.4.2 : EXCHANGER sub-module §5.4.4 : FUELPLAQ élément §5.4.5.2 : CORE sub-module §5.4.5.4 : ajout §5.4.5.5 : ajout §5.5.1 : ACCU gadget §5.5.4.1.1.3 correction §5.5.4.1.1.5 : correction schéma hauteur §5.5.5: SENSOR gadget §5.5.7 : ajout commentaire §5.5.8.1.3 : TEE gadget §5.7.1.1.3 : correction §5.8.1.1.2 : mélange de gaz incondensable §5.8.3 : ajout FLUIDS §5.9.1 : ajout MONOPHAS §5.9.2 : ajout COMPONENT §6.1.4.1.2 : Arithmetic operators §6.1.4.2.3 : Printout management §6.1.4.4.1 : Main modules §6.1.4.4.2 : Thermal component sub- modules §6.1.4.4.3 : Other thermal sub-modules §6.1.4.4.4 : Gadgets §6.2.1 : The junction notion, the junction head loss coefficients and the momentum equations §6.3.1.4.5 : User's file management (new) §6.3.2.5.2 : ajout GOBALLON §6.3.3.1 : User common §Appendix 14 Thematic INDEX (mathematic) §Appendix 14 Thematic INDEX (results) §Appendix 14 Thematic INDEX (wall) Mise à jour réalisée dans le cadre des contrats CSSI AB04B008 et AB03B005/4000054835	13/03/06	SSTH/LDAS/EM/2005-035

TABLE OF CONTENTS

1 REFERENCES

[DOC0] The CATHARE code: SMTH/LMDS/EM/2001-063

CATHARE main characteristics: general presentation about implemented physics and assessment strategy

[DOC1] CATHARE 2 V2.5_1 : User manual: SSTH/LDAS/EM/2005-035

(this document)

[DOC2] CATHARE 2 V2.5_1 : User guide: SSTH/LDAS/EM/2005-34

User help for the model choice and for frequently asked questions about CATHARE use and calculation management

[DOC3] CATHARE 2 V2.5_1 : Dictionary of operators and directives: SSTH/LDAS/EM/2005-019

Description of the syntax of operators and directives

[DOC4] CATHARE 2 V2.5_1 : Post-processing of results: SSTH/LDAS/EM/2005-020

Description of post processing operators and directives and list of parameters available to plot

[DOC5] I CATHARE 2 V2.5_1 : installation manual: SSTH/LDAS/EM/2005-017

How to install the CATHARE code on WINDOWS, UNIX and LINUX platforms

[DOC6] CATHARE 2 V2.5_1 : Description of the base revision 6.1 physical laws used in the 1D, 0D and 3D modules : SSTH/LDAS/EM/2005-038

Description of the 1D physical constitutive laws

Description of the exact formulation of correlation

Description of some justification elements (physical and numerical)

Description of the zone indexes printed on the output listing

[DOC7] Description of the 3D module: SSTH/LDAS/EM/2004-48

Description of the 3D physical models

Description of the 3D space and time discretisation for all balance equations

Description of the 3D boundary conditions (walls)

Description of the solution algorithm in the 3D

[DOC8] Physical improvement of the 3D module of CATHARE 2 V1.5B: SMTH/LMDS/EM/2002-084

Description of recent improvements for the 3D physical laws

[DOC9] CATHARE 2 V2.5_1 : Physical laws used in reflooding sub-module : SSTH/LDAS/EM/2005-036

Description of the reflooding module

Description of the physical laws used during a reflooding calculation

[DOC10] CATHARE 2 V2.5_1 : Description of the Volume module: SSTH/LDAS/EM/2005-37

Description and equations (internal and junctions) of the VOLUME module

Numerical analysis of the VOLUME module

Description of the closure equations (distribution and influence coefficients, mass transfer between sub-volumes)

Description of wall transfer in the VOLUME module

Description of specific model in the VOLUME module

[DOC11] Description of the CATHARE-2 V2.5 REVISION 6.1 pipe module tee sub-module: SSTH/LDAS/EM/2004-65

Basic equations of the 1D model

Terms used in the discretised equations

Discretised equations at a 1D point and at the TEE branch

TEE closures laws

Physical laws used in the TEE sub-modules

[DOC12] CATHARE 2 V2.5_1 : Description of mass source and sink sub modules: SSTH/LDAS/EM/2005-039

List of mass and sink sub-modules (sink, source, SGTR, break)

Physical equation and specific model used in mass and sink sub-modules

Numerical features and discretisation

Data processing

[DOC13] Description of the wall, heat exchanger and 0D steam generator sub-modules: SSTH/LDAS/EM/2004-049

[DOC14] Description of the PUMP sub-module: SSTH/LDAS/EM/2004-041

[DOC15] CATHARE 2 V2.5_1 : Description of the fuel: SSTH/LDAS/EM/2005-043

[DOC16] CATHARE 2 V2.5_1 : Description of the core kinetic sub-module: SSTH/LDAS/EM/2005-042

[DOC17] CATHARE 2 V2.5_1 : Description of the valve sub-module: SSTH/LDAS/EM/2005-040

Description of valve, flow limiter and check valve models

Physical modelling of valve sub-module

Terms added to 1D equations

Specific laws used in valve sub-module

Description of valve failures

[DOC18] Description of the fluid and material constitutive relationships SMTH/LMDS/EM/2004-032

2 APPENDIX TABLE

3 INTRODUCTION

The CATHARE code (Code for Analysis of Thermal-Hydraulics during an Accident of Reactor and Safety Evaluation) is developed to perform best-estimate calculations of pressurised water reactor accidents: PWR loss of coolant (large or small break, primary and secondary circuit).
Specific modules have also been implemented to allow modelling of other reactors like BWR or gas cooled reactors.

It is developed in GRENOBLE by the French Atomic Energy Commission (CEA), ELECTRICITE de FRANCE (EDF), FRAMATOME-ANP and IRSN.

This document aims to provide an understanding of CATHARE data files and how they are constituted and to explain how to perform calculations and process CATHARE calculation results.

Chapter 4 presents an overview of CATHARE code (hydraulic approach, numerical features, modelling features).

Chapter 5 describes the various objects available for modelling reactors, namely one or more hydraulic circuits and their specific features.

Chapter 6 focuses directly on CATHARE input and command language specifications:

- CATHARE language operators
- Reactor/Circuit data
- Steady state
- Regulation
- Transient
- Modification and intervention in the circuit.

Chapter 7 includes a special CATHARE feature, the stand-alone fuel computation.

4 GENERAL FEATURES

4.1 Range of application

CATHARE includes several independent modules that take into account any two-phase flow behaviour:

- Mechanical non-equilibrium:
  - vertical: co- or counter-current flow, flooding counter-current flow limitation (CCFL), etc.
  - horizontal: stratified flow, critical or not critical flow co- or counter-current flow, etc.
- Thermal non-equilibrium: critical flow, cold water injection, super-heated steam, reflooding, etc.
- All flow regimes and all heat transfer regimes.

In order to take into account these phenomena the CATHARE code is based on a two-fluid and six-equation model with a unique set of constitutive laws. Various modules offer space discretisation adapted to volumes (0D), pipes (1D) or vessels (3D) ready to assemble for any reactor description.

CATHARE is limited to transients during which no severe damage occurs to fuel rods; more precisely, fuel ballooning and clad rupture are assumed to have no major effect on water flow in the primary circuit.

Time discretisation is fully implicit (semi-implicit for 3D) and enables solution stability to be achieved over a broad range of time step values. The maximum time step is up to the user and depends on the problem being solved.

4.2 CATHARE software features

A CATHARE computation consists of three parts, corresponding to three separate executables:

1. CATHARE pre-processing (READER.exe): CATHARE data acquisition. This involves a user input deck. These data contain a description of the hydraulic circuit(s) to be simulated, the events occurring during the simulation and how calculation is managed. Chapters 5 and 6 provide all the information users need to build the input deck.

2. CATHARE calculation (CATHAR.exe): execution of the simulation described in the input deck, i.e. basically the thermal hydraulic computation. The present chapter describes the main features of this calculation.

3. CATHARE post-processing results (POSTPRO.exe): users may process CATHARE binary output files to output useful information to a formatted file for analysis (see [DOC4]).

Appendix 1 presents the files (FORTRAN units) needed and generated by each software item and the other files that may be used by them.

4.3 Main CATHARE characteristics

The 1D module is based on a 1D two-fluid model:

- Two energy balance equations,
- Two momentum balance equations,
- Two mass balance equations,
- and 0, 1, 2, 3 or 4 mass balance equations for 0, 1, 2, 3 or 4 non-condensable gases.

The mass and energy balance equations are of primary form whereas the momentum equations are of secondary form.
The six main variables are pressure, liquid enthalpy, gas enthalpy, void fraction, liquid velocity and gas velocity (P, Hl, Hg, α, vl, vg) and, if it exists, xi (i=1, 4) non-condensable mass fraction (with related transport equations).

The model of the 1D module is extended to the 3D and 0D modules.
In the 3D module 10 (+ xi) main variables are considered, with the momentum equations being written in all three directions: P, Hl, Hg, α, vl_x, vg_x, vl_y, vg_y, vl_z, vg_z.
In the 0D modules is divided in two sub volumes. The main variables are pressure, liquid enthalpy, gas enthalpy, void fraction and non-condensable mass fraction (with related transport equations) for each sub volume and the separation level elevation between the two sub-volumes. Two energy and mass balance equations are written for each sub volume. For the volume, a total volume the pressure equation is written.
Remark: The system always has six equations: even in single-phase computations, a residual phase treatment is used:
$\alpha_{min}=10^{-5}$, $\alpha_{max}=1-10^{-6}$,
$\alpha=\alpha_{min}$, Tg=Tsat, Vg=Vl for single phase liquid,
$\alpha=\alpha_{max}$, Tl=Tsat, Vl=Vg for single phase gas.
Other calculations are available:

- Transport of radiochemical components
- Mass and energy balances for each zone
- Radial heat conduction (for multi-layer wall and fuel structures)
- 2D conduction for rewetting (for multi-layer wall and fuel structures)
- Fuel thermo-mechanics (clad deformation, rupture and oxidation)
- Point kinetics model.

4.4 CATHARE numerical features

Discretisation is of the first order in space and time

- In spatial terms CATHARE uses:

Finite volumes (mass, energy) and finite differences (momentum) discretisation
Structured and staggered nodalisation (see Figure 1: 1D space discretisation for an example)
First order upstream scheme for convective terms (donor cell principle).

- In temporal terms CATHARE uses:

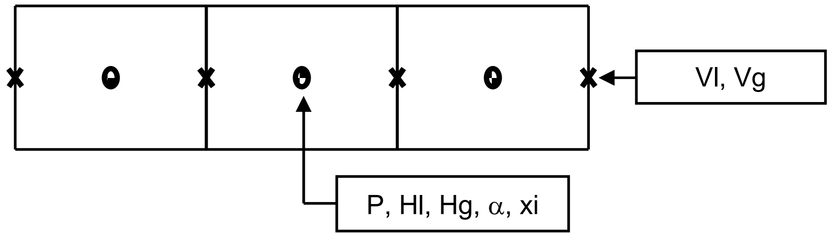
Fully implicit (0 and 1D) and semi-implicit (3D) discretisation.
Implicit wall conduction (+ implicit coupling with hydraulic conditions).

A non-linear system is thus obtained, solved using the NEWTON-RAPHTON iterative method.

Remark: this method is presented in Appendix 1.

More information on time discretisation may be found in [DOC2] WHAT is an IMPLICIT CODE?

Example:



✗ Vector point: Momentum equation resolution

⊙ Scalar point: Mass and Energy equation resolution

Figure 1: 1D space discretisation

4.5 CATHARE model features

Any kind of hydraulic circuit may be represented by elements, which are connected by junctions. These elements are modelled using CATHARE modules.
It consequently has a modular structure with five main modules:

- Basic 1D pipe module
- 0D volume module
- 3D module
- Boundary condition module
- Double-ended break module.

The heat exchange between one primary and several secondary circuits, via heat exchangers, can be calculated.

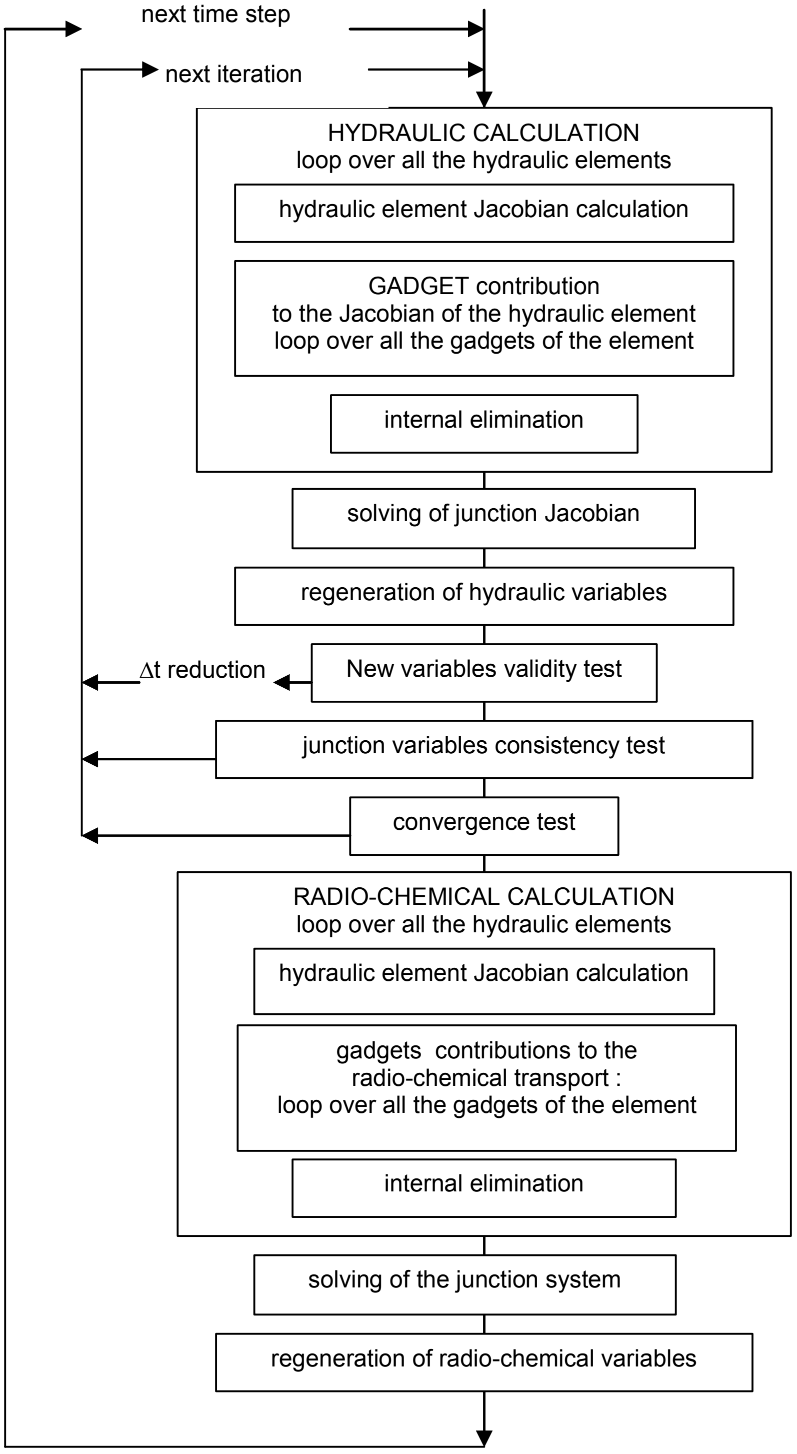
Other types of objects are available to represent:

- specific models related to the other calculations made by CATHARE, notably the main sub-modules (reflood, multi-layer wall, point kinetics model, etc.),
- localised changes to standard thermal hydraulic equations, in the form of gadget sub-modules for representing valves, injections, break, pump, etc.,

Other sub-modules refer to specific aspects of particular PWR transients – modelling steam generator feedwater overflow, mixing effects in the vessel bottom, etc.

More information, especially about the CATHARE code assessment may be found in [DOC0].

4.6 CATHARE general algorithm


next time step
next iteration
HYDRAULIC CALCULATION
loop over all the hydraulic elements
hydraulic element Jacobian calculation
GADGET contribution
to the Jacobian of the hydraulic element
loop over all the gadgets of the element
internal elimination
solving of junction Jacobian
regeneration of hydraulic variables
Δt reduction
New variables validity test
junction variables consistency test
convergence test
RADIO-CHEMICAL CALCULATION
loop over all the hydraulic elements
hydraulic element Jacobian calculation
gadgets contributions to the
radio-chemical transport :
loop over all the gadgets of the element
internal elimination
solving of the junction system
regeneration of radio-chemical variables

5 REACTOR COMPOSITION

All underlined key words (upper case characters) are related to an entry in [DOC3]. Other words in upper case characters are CATHARE key words related to these entries. Refer to this document for further information on key words and to Chapter 6 for generic information on their use.
Appendix 13 contains a summary of these items.

5.1 Object-oriented structure

Various elements in the reactor's hydraulic circuits can be distinguished. Each element is described by a CATHARE module (or CATHARE object type), in other words a set of programs used for modelling this element. Depending on their function, nature and similarity, these modules may be:

- Assembly modules
- Main (or basic) modules
- Thermo sub-modules
- Gadgets
- Special PWR sub-modules or gadgets
- Non-condensable gases and radio-chemical components.

The term object-oriented structure is used because modules are hierarchically organised: assembly modules contain modules that contain gadgets and/or sub-modules. Sub-modules are distinguished from gadgets because they may interfere with several modules, whereas gadgets only modify the behaviour of a single module locally.
The word “contain” should be understood literally in the physical sense but also with respect to the structure of CATHARE software; modules inherit properties from assembly modules; similarly gadgets or sub-modules inherit properties from their "parent" modules.
Non-condensable gases, for example, can consequently be defined as belonging to an assembly module and will not need to be defined for each element making up the hydraulic circuit.

5.2 Assembly modules

These modules are used to group basic modules. There are three assembly elements:

- REACTOR
- CIRCUIT
- ZONE.

5.2.1 REACTOR element

The REACTOR element consists of CIRCUIT elements. Circuits connected in this way may be physically connected by heat exchangers (see §0).
Several reactor elements can be defined in the same input deck, but from the calculation point of view will be separate from one another.
This module (see §6.3.1) is used to control thermal hydraulic computation. It may also be used to launch specific calculations or initialisations (see §6.3.2.5).

5.2.2 CIRCUIT element

The CIRCUIT element consists of all the basic elements making up the hydraulic circuit. Some parts of the circuit can be isolated from the circulating fluid zone (see §6.3.2.2) with valves (see §5.5.9). But as the fluid is the same in the entire circuit, non-condensable gases and radiochemical components (see §5.8) are defined for a circuit element (obviously the concentration or radioactivity of fluid components are not necessarily the same in the entire circuit).
Specific calculations may be launched at this module level (see §6.3.2.5).

5.2.3 ZONE element

The ZONE element is designed to help CATHARE users process calculation results. It enables several elements of the same circuit to be combined in an assembly on which CATHARE performs mass and energy-balance calculations. Specific initialisations may be launched at this module level (see §6.3.2.5).

5.3 Main modules

These modules are used to define the particular hydraulic feature of each part of a hydraulic circuit.
They are also called “hydraulic” or “basic” elements. The list of these modules is as follows:

- AXIAL,
- VOLUME,
- THREED,
- BCONDIT,
- RUPTURE.

For each of these modules, if not already predefined by CATHARE, users will need to provide a topological definition (links to neighbours?), a nodalisation definition, a geometrical definition (object size, for example), and a hydraulic definition (kind of flow, singular pressure drops).

Modules are connected by **junctions**.

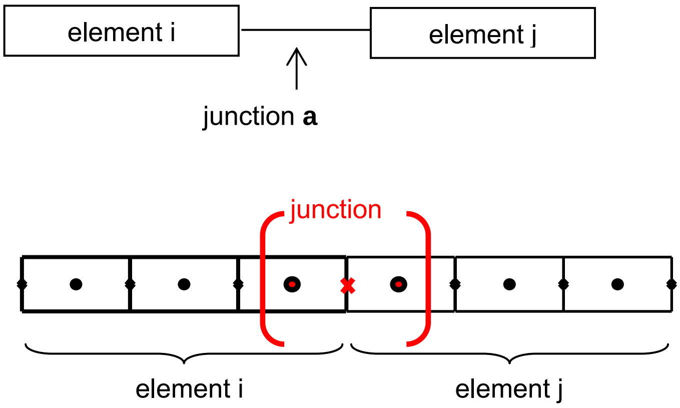


Figure 2: What is a junction?

- A junction is a topological notion used to link two elements.

- A junction is referenced twice; respectively in each topological definition of the two elements which share it. For each element, the junction is associated with the key word USTREAM (for upstream) or DSTREAM (for downstream). The choice of key word is insignificant since it is different for the two elements. In the above example, if the junction is defined as upstream for element i, it must be defined as downstream for element j and vice versa.

- It is represented by a physical vector point and by two scalar points, which are all shared by the two connected elements. The vector point specifications (geometrical parameters, flow area, gravity, etc.) are provided adopting the point of view of each adjacent element; both sets of specifications must be consistent with each other and should satisfy CATHARE reader controls.

5.3.1 AXIAL element

5.3.1.1 Description

CATHARE literature may also contain other terms: "pipe element" or "1D element".
This module is basically used to model a pipe. It may also be used to model hot and cold legs, steam generator-U tubes, the expansion line, core channel, core bypass, downcomer, SG riser, SG vapour line, etc.

From the physical and numerical point of view it is characterised by a staggered mesh and a 2-junction model. See Figure 1: 1D space discretisation.
By definition, a pipe is modelled as a succession of **truncated cones** (or a cylinder) with **continuity of cross-section:**

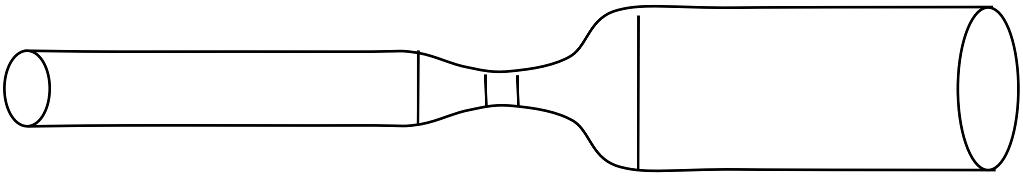


Figure 3: Physical object to model

will become

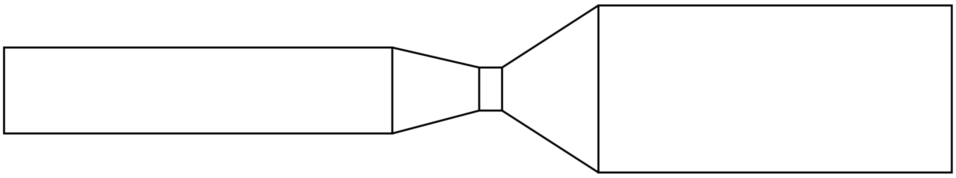


Figure 4: CATHARE representation of a pipe

Users must specify topology, geometry and mesh definitions, as well as the type of hydraulic flow (rod bundle, annular or standard) related to specific correlations. For these three cases of flow geometry, the interfacial friction is different. Several flow geometries can be juxtaposed in the same element.

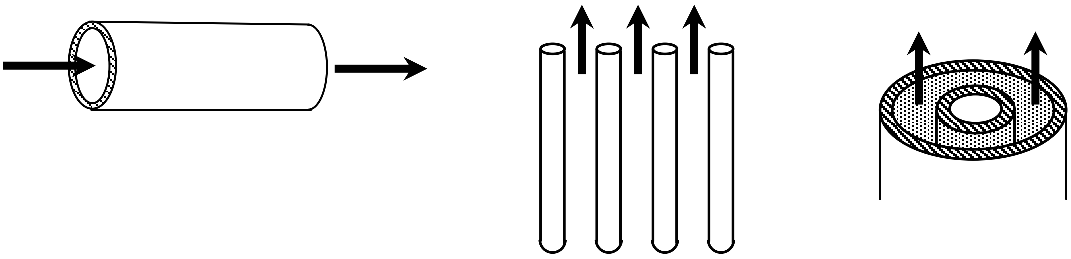
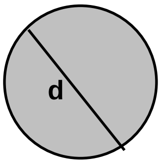


Figure 5: Standard, rod bundle and annular flow

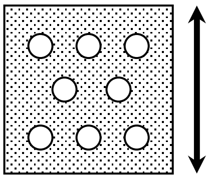
Example of geometrical values to be entered by users (GEOM):

- case of a standard pipe:



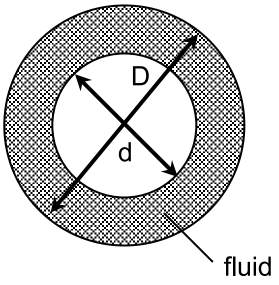
SECT = π $d^2/4$
PERI = π d

- case of a rod bundle pipe:



n rods with diameter D
PERI = 4* L + n * π D
SECT = L^2 - n * π $D^2/4$

- case of an annular pipe:



PERI = π (d + D)
SECT = $\pi/4$ (D^2 - d^2)
SIZE = D

Singularities can be defined on each vector mesh of the pipe.

5.3.1.2 How to proceed?

The following key words are needed to characterise this element in the input deck:

- AXIAL (topological definition),
- GEOM (geometrical definition),
- HYDR (hydraulic definition),
- MESH (mesh definition),
- SINGULAR (optional) (singular pressure-drop definition).

5.3.1.3 Other sources of information

Refer to Appendix 3 for more information on numbering and discretisation of the AXIAL element.

Refer to [DOC2] for more information on MESH, HYDR and GEOM for an AXIAL element:
HOW TO CHOOSE a MESHING for a PIPE?
HOW TO CHOOSE the HYDRAULICS of a PIPE?
HOW TO DEFINE the GEOMETRY of a PIPE?

Refer to [DOC6] and [DOC11] for more information on physical constitutive laws and correlations.

5.3.2 VOLUME element

5.3.2.1 Description

CATHARE literature may also contain other terms: ‘capacity module' or ‘0D module‘.

The so-called volume module calculates a two-node point model. It should be used in two cases:

- to describe large volumes in which gravity effects are dominant,
- to connect several modules.

The main assumptions are that:

- velocities inside the volume are small compared to the velocities at the junctions,
- inertial forces are assumed to be negligible compared to gravity forces.

This implies drastically simplified momentum balance equations.

It can describe single and two-phase flows, as well as well-mixed and totally stratified flows and is therefore used to model, among others:

- parts of the pressure vessel: lower plenum, upper plenum, upper head,
- inlet and outlet steam generator plena and pressuriser,
- the steam dome and steam collector of the steam generator secondary side.

When modelling a pressuriser, special options can be activated. The PRSRIZER key word (see VOLUME) may be used to accelerate heat transfer between upper and lower sub-volumes in the element during initialisation, in order to reduce the time constants associated with pressuriser dynamics. The PWRPRZ option (see VOLUME) allows realistic heat exchange between the spray water and the steam to be predicted when the spray flow does not touch the walls when dropping. When modelling heated walls, the option is used for accurate prediction of the flashing delay due to the heaters and vapour-wall exchange when the pressure is increasing in case of film condensation of the steam above the mixture level.
(see [DOC2]) for more information.

Thanks to new models implemented through optional key words for volume junctions (GEOM), volume modules can also be used to model secondary-side steam generators using separator and dryer volumes to prevent liquid entrainment to the steam turbines. This adds the possibility of representing the effects of such components, at least qualitatively.

Another specific model may be applied to a VOLUME element, namely the film condensation model (VFILM), which allows a specific correlation to be used for the wall-to-fluid flux calculation (the Chen correlation is the standard CATHARE correlation, but it may be changed to the NUSSELT, COPAIN, or USHIDA film condensation correlation). The VOLUME element can thus be used to model the reactor containment using a VOLUME module assembly.

The CATHARE representation of the volume is characterised by:

- Two sub-volumes (Ω+,Ω-), corresponding to two internal scalar nodes where scalar equations are solved, and a separation level (Zc).
- As many junctions as needed. The six equations are solved at the volume port to define the main variables at each junction. The code then introduces a fictitious scalar node inside the volume port associated with the vector point corresponding to the junction.
- Additional closure equations:
  - internal hydrostatic equation (for internal scalar nodes),
  - phase distribution laws at junctions,
  - droplet fall and bubble rise velocities at the interface,
  - condensation and evaporation at the interface.

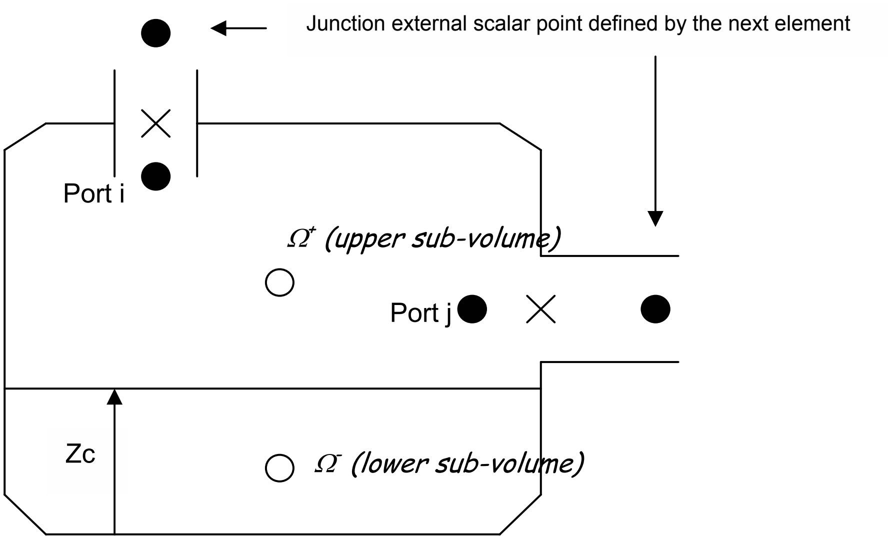


Figure 6: Spatial discretisation of the standard volume module

More information on VOLUME module discretisation and numbering can be found in Appendix 4.

Nodalisation is then predefined by CATHARE as well as the hydraulic flow type (using the simplified momentum equations due to the model).

5.3.2.2 How to proceed?

The following key words are needed to characterise this element in the input deck:

- VOLUME, (topological definition),
- GEOM (geometrical definition),
- SINGULAR (optional) (junctions singular pressure-drop definition).

5.3.2.3 Other sources of information

Refer to [DOC2] for more information:
WHY-WHEN TO USE a VOLUME?
HOW TO DEFINE the GEOMETRY of a VOLUME?

Refer to [DOC10] for more information on physical constitutive laws and correlations.

5.3.3 THREED element

5.3.3.1 Description

CATHARE literature may also contain another term: '3D module'.

The main purpose of the THREED element is to represent large-scale thermal-hydraulic 3D effects in nuclear power plants. One of the main applications is modelling PWR vessels but it can be extended to other geometries.

Review of industrial applications based on large scale 3D hydraulics:

- large-break LOCA studies,
- small and intermediate break LOCA studies,
- studies of accidental procedures (immediate or late pump trip),
- loss of RHRS,
- other PWR transients (loss of steam generator feedwater, SGTR, etc.).

The main phenomena usually studied with the THREED model are the three main phases of a large-break LOCA:

- blowdown phase:
  - core radial power profile (core radial temperature profile),
  - 3D behaviour of rewetting (3D distribution of energy stored in the core),
- downcomer refill:
  - bypass to the break in the water injected by the accumulator,
  - steam/water counter-current flow phenomena in the annular downcomer,
  - production of steam in the core,
  - condensation of steam with non-condensable gases,
- core reflood phase:
  - radial power profile (3D thermal-hydraulic behaviour),
  - 3D behaviour during blowdown phase (initial conditions are really 3D),
  - 3D flow distribution at the core upper tie plate (limit of the water hold-up),
  - below the quench front, cross flows (efficient mixing between hydraulic channels),
  - above quench-front, almost no mixing (strong radial effect on rod temperatures),
  - droplet de-entrainment in the 3D upper plenum (effect on steam binding, effect on quench front progression).

In the CATHARE code, fuel ballooning is taken into account only for local cladding thermo-mechanical behavior, without any induced change in core geometry that remains unaffected by ballooning.

Nevertheless, some physical laws are modified to take into account the effect of ballooning on fluid distribution around the balloon. Mechanical effect of the presence of the ballooning is modeled by adding pressure loss coefficient in the vector points close to location of balloon.

Steam to liquid and wall to fluid heat transfers are also modified to take into account the fluid acceleration, the liquid droplet break up and vaporisation.

The model (operator GOBALLON) is activated when the clad rupture is calculated both during the depressurisation phase and the reflooding phase of a large break LOCA. This model may be used when the core is modelled with a 3D module and when the core rod are represented using either simple walls or fuel sub modules.

Non-standard applications of CATHARE 3D:

- Other PWR transients (low pressure),
- BWR transients (multi-dimensional neutronic coupling),
- VVER steam-generators (secondary side),
- Analysis of experiments in other domains:

  Severe accident: SULTAN,

  Heat exchanger: EPICE,
- Coupling with fine thermal-hydraulic codes.

It is based on the 1D two-fluid 6-equation model. The basic set of equations consists of 10 thermal-hydraulic differential equations (mass and energy balance for each phase, momentum balance for each phase along each coordinate direction). A qualified set of constitutive relationships is directly extrapolated from those of the axial element 1D correlations, extended in all three directions, with certain specific features (no stratification, no added mass term).

The numerical features are the same as in the 1D model with these particularities:

- finite volumes for all scalar equations,
- finite difference for momentum equations,
- rectangular and cylindrical coordinates are available.

Remark: for cylindrical coordinates, only the vertical axis for the Z direction is allowed.

The discretisation scheme is **semi-implicit**:

- mass and energy convective flux terms are time-explicit (time step bounded by CFL stability criterion relating to fluid velocity),
- pressure and velocities are time-implicit in the momentum equations.

Remark: A diffusion model is available for this module (GODIFF) as well as a (k,ε) model for each phase (liquid and gas mixture) (TURBULEN) but there is no general model for two-phase flow turbulence.

The turbulent model is dedicated to very specific applications in which turbulent phenomena must be significant. It can be used to model the turbulent diffusion of one or both phases (gas and/or liquid). The initial turbulence state is required in the input deck, via the initial value of the turbulent kinetic energy (k) and the turbulent dissipation rate (ε).

5.3.3.2 How to proceed?

The THREED element creates a 3D element by defining its topology.

Users define the THREED geometry, nodalisation, hydraulics and connections to other elements using five compulsory key words:

- THREED and CONNECT (topological definition),
- GEOM (geometrical definition),
- HYDR and PHYSCALE (hydraulic definition),
- SINGULAR (optional) (singular pressure drop definitions).

Remark: a THREED element may be connected to a VOLUME, AXIAL, BCONDIT or a THREED element. Two kinds of junctions are available, external (standard – on an outside face of the THREED element) or internal (for guide-tubes or hot leg connection).

5.3.3.3 Other sources of information

Refer to Appendix 5 for more information on mesh numbering in THREED element.

Refer to Appendix 6 for advice and an example of CATHARE definition of a THREED element (topology, geometry, nodalisation, hydraulics, etc.).

Remark: In CATHARE, **hydraulic channel** refers to a part of any 3D hydraulic mesh column (Z).

Refer to [DOC2] for more information on:
WHY-WHEN TO USE a 3D MODULE?
HOW TO CHOOSE a 3D NODALISATION?
HOW TO CHOOSE the HYDRAULICS of a 3D MODULE?
HOW TO CHOOSE the PHYSICAL SCALES of a 3D MODULE?
HOW TO DEFINE the CONNECTIONS of a 3D MODULE?
HOW TO DEFINE the GEOMETRY of a 3D MODULE?

Refer to [DOC7] and [DOC8] for more information on discretisation, physical constitutive laws and correlations of the THREED element.

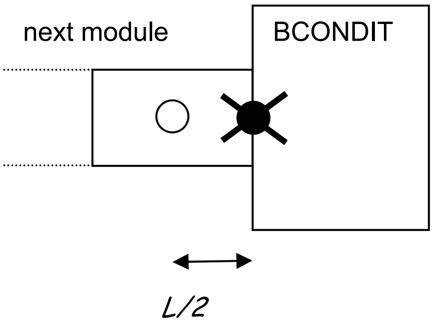
5.3.4 BCONDIT element

5.3.4.1 Description

CATHARE literature may also contain another term: ‘boundary condition module'.

The BCONDIT element is used to model a boundary condition.
It can be connected to an AXIAL, a VOLUME or a THREED element.
It allows one or more hydraulic conditions to be set for each phase (pressure, enthalpy, velocity, mass flowrate for gas or liquid, void fraction or mass fraction of non-condensable gas) depending on what the user wants to model, at the inlet or at the outlet of the element.

CATHARE spatial discretisation of the BCONDIT element:
The scalar and vector points of the boundary condition are superposed.



NB: the BCONDIT element is not a junction.

5.3.4.2 How to proceed?

BCONDIT type objects may be of two types:

- Internal type: variables are defined with respect to time (NB: time laws should begin at time zero).
- External type: variables are not time-dependent but imposed externally (and explicitly) by the user via the input deck.

There are three kinds of boundary conditions:

- inlet boundary conditions,
- outlet boundary conditions,
- mixed boundary conditions.

Users consequently only need to define the topological characteristics of the BCONDIT (BCONDIT key word) and the kind of boundary condition they want to use, i.e. the physical conditions to be imposed (MODEL key word).

Remark: the physical model imposed using the BCMOD key word may be changed during a transient.

5.3.4.3 Other sources of information

Refer to [DOC3] (MODEL) for a description of all available models.

Refer to [DOC2] HOW TO CHOOSE and to use a BOUNDARY CONDITION for more information.

5.3.5 RUPTURE element

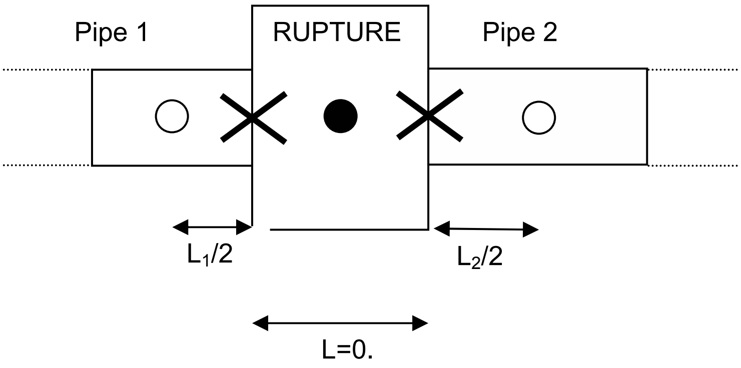
5.3.5.1 Description

The RUPTURE element is used to model a double-ended pipe break, with scope for critical flowrate conditions.

This object can only be connected to two AXIAL elements because critical flow can only be calculated on meshed elements.

The RUPTURE element may be seen as the combination of two BC4A outlet boundary conditions, each one attached to an AXIAL element. No topological or geometrical data are consequently needed (the data for the AXIAL junction are used), and hydraulic data are predefined by CATHARE. But the RUPTURE and ASSIGN key words must be used in the input deck to define topological characteristics, any physical conditions to be set and the time of opening.

The scalar and vector points of each boundary condition are superimposed.



Conditions at the junction with pipe2 and at the junction of pipe1 may differ even if the boundary conditions are physically located at the same place.

5.3.5.2 How to proceed?

The associated RUPTURE module can be of two types:

- **Internal type:** physical conditions are defined with respect to time.
- **External type:** physical conditions are not time-dependent but set at any time by the user.

5.4 Thermal sub-modules

These elements model thermal effects caused by structures and affecting hydraulic behaviour.
The list of these elements is as follows:

- WALL, WALL3D (radial-wall conduction coupled to hydraulic flow),
- EXCHANGER (wall conduction between two hydraulic meshes),
- FUEL, FUEL3D, FUELCHAR (fuel-rod model with calculation of thermo-mechanical behaviour),
- FUELPLAQ (plane-fuel model without calculation of thermo-mechanical behaviour).

Other specific calculations may be attached to these walls (with exceptions):

- REFLCHAR, REFLCH3D (rewetting and reflooding of walls),
- Modelling of a simplified secondary circuit by a point-model steam generator attached to a wall SGPOINT,
- Point kinetics model to share core power between the fuel rods of a circuit (CORE).

The reader should refer to the relevant chapters in this document for more information.

All these walls must be attached to a main module (each element may have several walls). The main hypothesis is that **heat conduction** is assumed to be radial given that axial energy transport is mainly driven by fluid flow and that axial conduction can be neglected (except for reflooding).

In the CATHARE code, 25 standard properties are provided for the structural material. If users wish to use a new material not defined in the CATHARE material library, it will take the name XXXXXXi and its properties must be written in subroutine FWMAXX (see §6.3.3.2) (or YYYYYYij defined in FWMAYY).
A single subroutine describes the physical properties of each predefined material. Density, specific heat capacity and conductivity are calculated as a function of material temperature.
The CATHARE computation will automatically stop if the value of one of these three physical properties becomes negative or if the calculated temperature of the material exceeds the solidus value.

Refer to [DOC18] for a complete description of CATHARE material constitutive relationships and properties.

5.4.1 WALL and WALL3D sub-modules

5.4.1.1 Description

The WALL and WALL3D sub-modules describe heat conduction in wall structures. They take into account heat exchange with the flow and the outside environment, and power generation.

Walls can be connected to any AXIAL or VOLUME module, using the WALL key word, or to a THREED module, using the WALL3D key word. The main features of walls are:

- Cylindrical or plane geometries are allowed,

- Different layers of material can be specified (maximum value is 13 for a simple wall),

- For WALL connected to an AXIAL or a VOLUME (and not for WALL3D) , a contact resistance (option RESIST of WALL) can be added between two layers.
  The heat transfer coefficient (h) input in the data deck is used to add a term (1/h*e) to the thermal conductivity calculation of the last or first mesh of one of the two considered layer (e is the thickness of the considered mesh) . The considered layer is defined using the LEFT or RIGHT keyword. The heat transfer coefficient is either given by the user (VALUE) either calculated by the code from the user's material properties (MODEL).
  This model can be used to model an actual resistance contact or to avoid the modelling of a given material (a fine painted layer for example).

Calculation of the wall is coupled to the fluid through three parameters:

- heating perimeter for WALL (HPERIM), heating surface for WALL3D (HSURF),
- wall temperature (one value for one wall and one hydraulic mesh),
- wall-to-fluid heat flux (see [DOC6]):

q_{pl}	heat flux	wall ↔ liquid
q_{pv}	heat flux	wall ↔ vapour
q_{pi}	heat flux	wall ↔ interface

A one-dimensional heat conduction equation is solved, only considering radial conduction. The solution scheme is time implicit; only the material thermal properties are evaluated with time-explicit temperatures. Fluid flow can be inside the structure (EXTERNAL structure) or around it (INTERNAL structure).

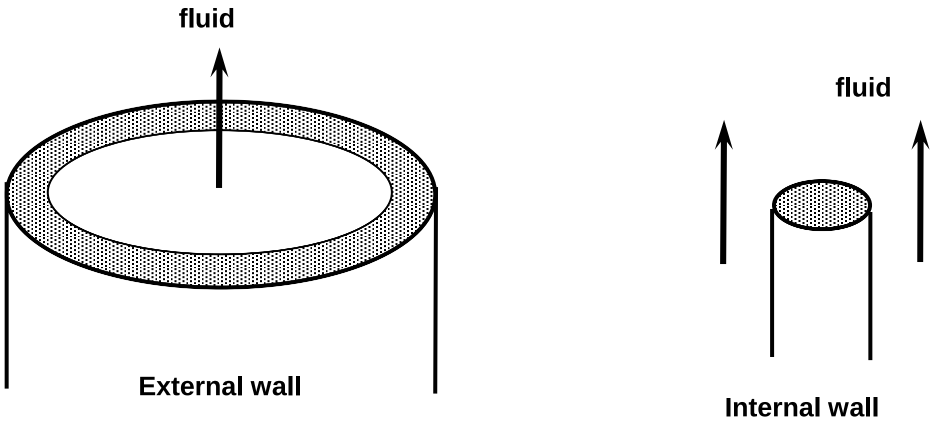


Figure 7: EXTERNAL and INTERNAL walls

5.4.1.2 How to proceed?

Considerable flexibility is possible when modelling structures. Users must define:

- internal and external diameters, radial meshes to be used in the heat conduction equation,
- the heating perimeter (HPERIM) or heating surface (HSURF) .

By combining these data CATHARE determines the wall volume, its thickness and the surface area in contact with the fluid.

Exchanged fluxes are calculated by the flow considering the wet temperature (see [DOC6]) and the heat conduction equation is solved to obtain the wall temperature field.

Radiative exchange may be modelled between walls in the same AXIAL or THREED element using the RADIAT key word in the input deck.

Heat exchange with the outside can be defined (WALL or WALL3D - LOSS), modified (FLUMOD) or cancelled (ADIABWAL).

Generated power (heating structures) can be defined in a specific layer (WALL/WALL3D - SOURCE) or overall (POWER) and a power profile can also be added along the wall (in the axial direction) (PNRSHAPE/PNRSHAPX).

5.4.1.3 Other sources of information

Refer to Appendix 7 for information on meshing and numbering of WALL and WALL3D and to [DOC13] for more information on discretisation and physical correlations.

5.4.2 EXCHANGER sub-module

The heat exchanger model is based on the standard one-dimensional heat conduction model.

Exchangers model the heat exchange through a wall from fluid to fluid. The two fluids may or may not belong to the same hydraulic circuit. In other words two kinds of EXCHANGER sub-module may be defined in CATHARE:

- An EXCHANGER between two circuits,
- An EXCHANGER within a circuit.

An additional and external power can be added to the exchanger wall, using the EXCPOWER directive. It can be used for example to simulate the gamma radiation thermal effect.

NB: Users must define the initialisation value of the exchanged power (ECHPOWER) (see §6.3.2.2 for information on the initialisation step).

5.4.2.1 Heat exchanger between two CIRCUITs

Commonly, they are used to simulate the exchanging part of a steam generator:

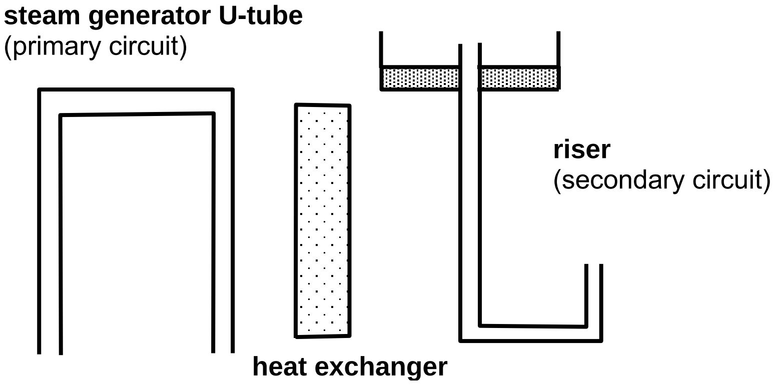


Figure 8: Heat exchanger in a PWR steam generator

This exchanger element has to be defined between parts of two AXIAL elements belonging to two different CIRCUIT elements. The EXCHANGER is considered by each AXIAL element as a WALL with imposed external conditions.

So, as for a WALL, the user defines materials, wall diameters and heating perimeters (EXCHANGER).
The EXCHANGER mesh should also fit (in the WALL definition) with both AXIAL hydraulic meshes. Refer to Appendix 7 for more information on EXCHANGER meshing.

Numerically, the resolution of the heat transfer calculation scheme can be chosen quasi-implicit or explicit. Depending on this choice, the time coupling of the primary and secondary circuits are tightly or loosely coupled.
Refer to Appendix 8 for more information on coupled calculations.

5.4.2.2 Heat exchanger inside a CIRCUIT

In a circuit, heat exchangers model the heat exchange between two AXIAL elements of a circuit, between two parts of the same AXIAL element or between two parts of the same THREED element. Examples are given in Figure 9: Heat exchanger inside a circuit.

They are based on the same model as standard walls, using the one-dimensional heat conduction model.

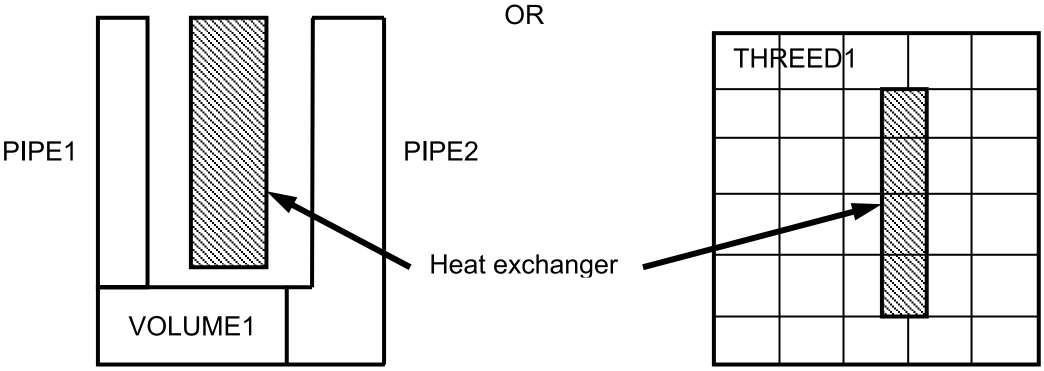


Figure 9: Heat exchanger inside a circuit

In practical terms, these exchangers can be used to model thermal coupling on the secondary side of a steam generator or in different parts of a core vessel.

The syntax and rules for their definition are basically the same as for exchangers between two different circuits.

5.4.3 FUEL, FUEL3D and FUELCHAR elements

5.4.3.1 Main features

The fuel module involves thermal conduction (standard wall model) and a thermo-mechanics model (calculation of internal pressure, clad deformation, rupture and oxidation). The thermo-mechanical behaviour is, however, explicitly calculated whereas thermal conduction is implicitly coupled with the hydraulics.

With the thermo-mechanical behaviour, a specific model can be used (FUELDTMX) to manage the step time before and after the clad rupture).

For more information on the fuel model, discretisation and correlations, refer to [DOC15].
Moreover, the power distribution assigned to this module can be associated with a point kinetics model (CORE). In this case all power definitions included in the module definition are ignored and users must enter specific characteristics (see §5.4.5.2).

There are two ways to perform a calculation with fuel. These options are specific to the fuel module and do not exist for the other CATHARE 2 modules.

The first standard calculation option performs a complete calculation of hydraulics and fuel. In this case, the entire hydraulic circuit is nodalised and there is reciprocal interaction between hydraulic flow and fuel rod behaviour.

With the second "stand-alone" fuel option, hydraulic flow is imposed and is consequently not affected by the thermal and thermo-mechanical behaviour of fuel rods. This enables fuel rods to be calculated more quickly (because the hydraulic circuit is not entirely nodalised). The method for performing such calculations is described in chapter 7 of this document. It should be noted that the FUEL and FUELCHAR elements described below are the same as those used for the stand-alone fuel option.

A fuel rod is basically a wall (see §5.4.1) with special characteristics:

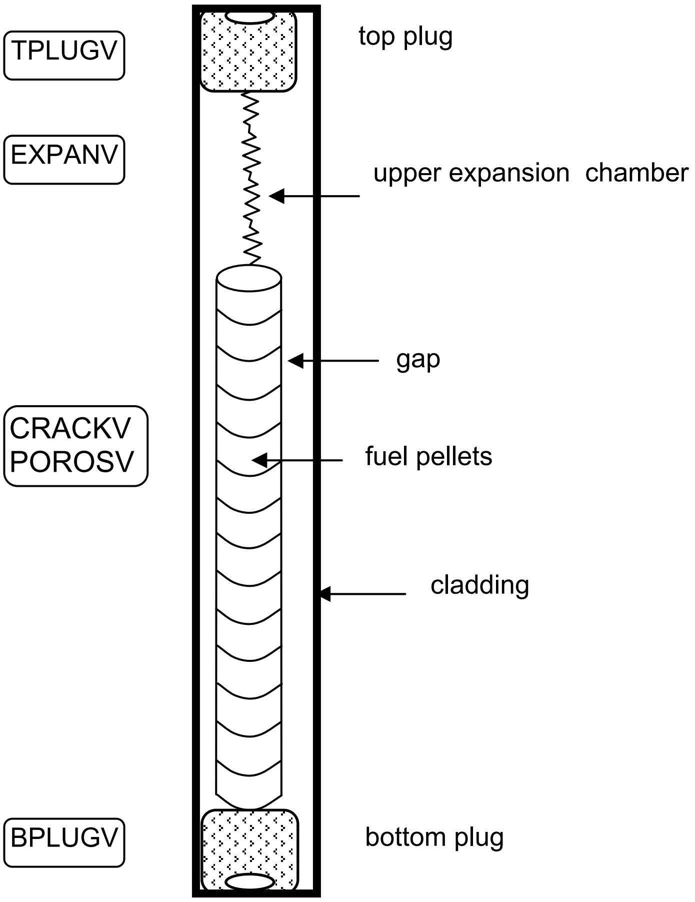


Figure 10: Fuel rod

- A fuel rod is always associated with a vertical AXIAL element or a THREED element.
- Unlike walls, fuel rods are always internal cylindrical structures as shown in Figure 11: Fuel structures.

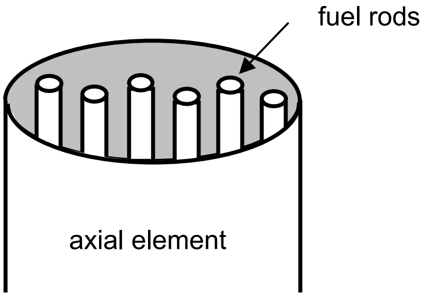


Figure 11: Fuel structures

- Fuel rod layers are assumed to match standard PWR specifications. Therefore, meshing is almost completely predefined in CATHARE (see Figure 12: Materials and radial meshing in a fuel rod). Nevertheless, users may choose between:

- two predefined characteristics for the UO_2 (new or irradiated) or use their own characteristics,
- two predefined characteristics for the cladding (CEA or PHEBUS),

The external oxide layer may also be eliminated (SPALL).

- Moreover, the mesh is predefined except in the UO_2 pellet

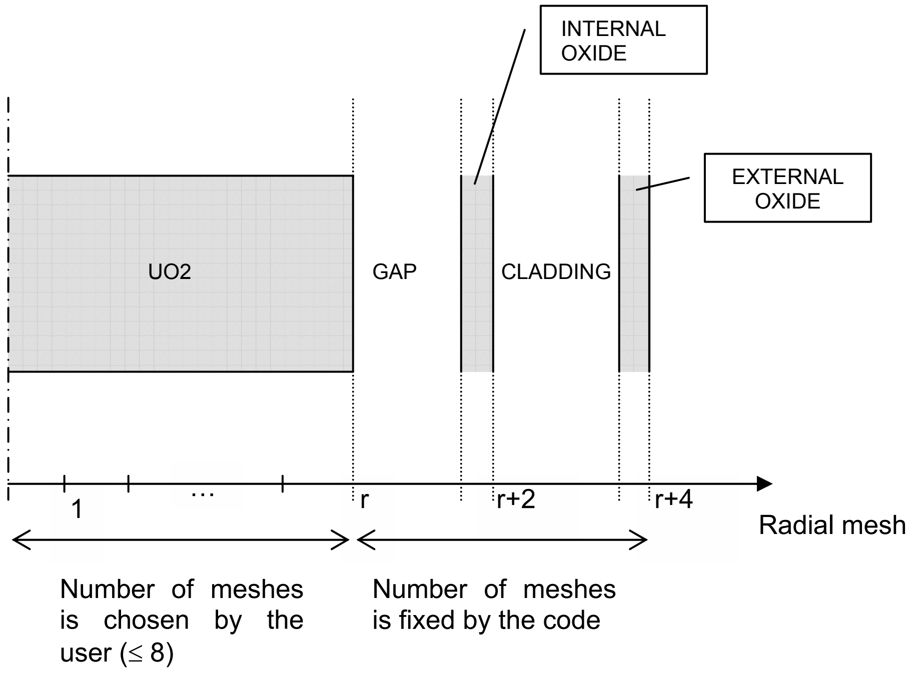

INTERNAL OXIDE
EXTERNAL OXIDE
UO2
GAP
CLADDING
1
…
r
r+2
r+4
Radial mesh
Number of meshes is chosen by the user (≤ 8)
Number of meshes is fixed by the code

Figure 12: Materials and radial meshing in a fuel rod

Remark:
External oxide: oxidation created on the cladding at water contact
Internal oxide: oxidation created after cladding rupture, if infiltration of water oxidises the internal side of the cladding.

- As the fuel model is assumed to represent neutron power distribution, regardless of whether or not the point kinetics model (CORE) is used, users will have to define non-residual and residual power (POWENR, POWRES) as well as the fraction of neutron power released in the fuel (XNEUT), the remainder (1-XNEUT) being released directly into the water. In order to model a heating wall simply without taking into account these particularities, use XNEUT = 1.D0 and POWRES = 0.D0.

- Users should start thermo-mechanical computations with the GOFUEL key word. It cannot be launched before the initialisation.

Remark: if HYDROGEN has been declared as non-condensable in NONCOND (see §5.8.1) and if there are fuel rods in the circuit, after starting the thermo-mechanical computation, there will be a source term of H_2 and a sink term of H_20 in the AXIAL or THREED element containing the fuel rods, because of oxidation of the fuel rod cladding: $Zr + 2H_2O \rightarrow ZrO_2 + 2H_2$
The code automatically takes into account this source and sink term, treating them implicitly.

In addition, the phenomenon of starvation of the oxidation reaction is also taken into account, when there is not enough H_2O to allow normal oxidation. In this case, the oxidation reaction slows down and the H_2 mass flowrate inside the element decreases.

5.4.3.2 Definition steps

Defining a fuel rod in the CATHARE input deck involves four steps.

1. As is the case for walls (WALL/WALL3D), each axial mesh of the fuel rod (or elementary wall) corresponds to an axial mesh of the axial-type element or a Z vector mesh of the THREED element (however, the fuel rod can be shorter than the element). Each elementary wall is defined with FUEL for an axial element or FUEL3D for a THREED element. These key words are used to define the local characteristics of the fuel rod, such as the initial width of the gap, internal and external oxide layers or the initial dimension of the UO_2 hole in the pellet center (see Figure 13: UO2 pellet).

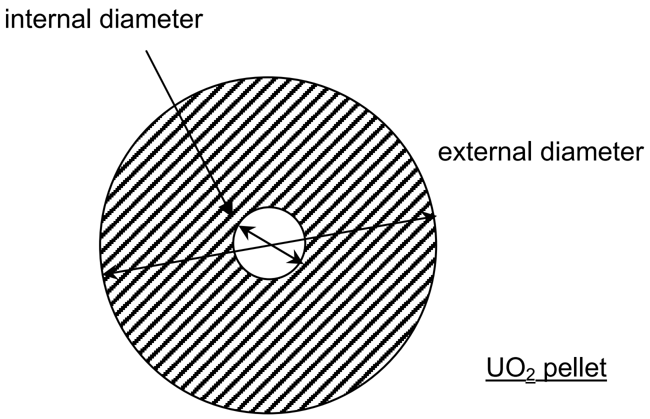


Figure 13: UO_2 pellet

Remark: users input the initial radial mesh, but the radius will change during the thermo-mechanical computation (GOFUEL). The radial conduction calculation takes into account expansion (in radius and volumetric density values).

For NEUTRO specifications, refer to §5.4.6.1.

2. The elementary walls must then be gathered to form an axially complete fuel rod (AND).

Remark: for a fuel rod defined in a THREED element (FUEL3D assembly), to facilitate fuel rod definition, the AND key word may be used to copy one fully defined wall in various 3D hydraulic locations re-using the same elementary walls.

3. The overall properties of the fuel rod must also be defined (FUELCHAR = FUEL CHARacteristics), such as the initial pressure in the gap or the void volume in the upper expansion chamber, in the bottom and top plugs (see Figure 10: Fuel rod), the total void between the UO_2 pellet assembly (see Figure 14: UO2 pellet assembly) or the contact radius between two pellets (see Figure 15: Radius to reach contact).

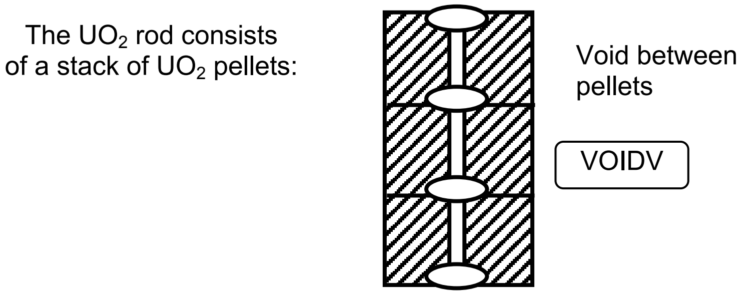


Figure 14: UO_2 pellet assembly

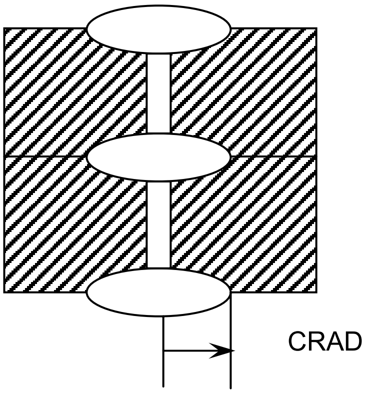


Figure 15: Radius to reach contact between two pellets

Users can describe several identical fuel rods in the same hydraulic element using the NCHANNEL key word in FUELCHAR. It should be borne in mind that in the THREED element this will not “distribute” the structures spatially. CATHARE will consider that they are all located in the same hydraulic channel.
Zero is an acceptable value for NCHANNEL. In such case, the fluid will not take into account thermal fluxes from this fuel rod but CATHARE will nevertheless calculate the effect of hydraulic conditions on the fuel rod (the same behaviour as a similar stand-alone fuel computation – refer to Chapter 7).

Remark1: users can describe several "fuel structures” inside a single AXIAL or THREED element, a fuel structure being a set of identical fuel rods. Each fuel structure differs from the others in one or more respects, such as nominal power or internal pressure. Users may consequently define several fuel rods (several FUELCHAR) in the same module.

Remark2: there is a relationship between NCHANNEL and the weight of the hydraulic element (see §6.2.2 for information on notion of weight).

Example: calculation of the 41184 identical rods in the MEANCORE element. Several methods may be used:
i) Either only one axial MEANCORE element (WEIGHT = 1) is defined, with 41184 identical rods (NCHANNEL = 41184).
ii) Or, for example, 156 identical axial MEANCORE elements (WEIGHT = 156) can be defined, and in each elementary axial element 264 identical rods (NCHANNEL = 264) are defined, because 264 x 156 = 41184.
In each case, the nominal power of the fuel structures (POWNEUT, POWRES in FUELCHAR operator) must be equal to that of a **single** rod.

Users must also define the volume of porosity (POROSV), high temperature cracks (CRACKV), the volume fraction (VOLFRAC) and the coefficient of porosity (POROS) of UO_2. Theoretically, they must be consistent with POROSV and CRACKV but it is not compulsory. VOLFRAC and POROS are used to compute conductivity and heat capacity.

For NEUTRO specifications, refer to §5.4.5.2.

4. Finally, users must link the set of FUEL/FUEL3D walls defined in 1 and 2 with the overall characteristics defined in 3 (FUELCHAR) using the INTEGRATE key word.

At this point, an entire fuel rod (or more precisely a set of identical fuel rods, i.e. a fuel structure) is defined.

5.4.3.3 Special models and features of the FUELCHAR module

- POWER changes the non-residual and residual power during computation when the point kinetics model (CORE) is not being used.

- PNRSHAPE/PNRSHAPEX defines a time-dependent axial profile for non-residual power.
- PRESHAPE/PRESHAPX define a time-dependent axial profile for residual power.
- XNEULIST/XNEULISX defines a time-dependent axial profile for the fraction of neutron power released in the fuel (XNEUT).

- MODXNEUT changes the XNEUT value as a function of a condition involving the ratio (delayed neutron part of residual power) / (total residual power).

- SPALL removes the external oxide layer from calculation.

- OXRATE performs a parallel calculation of the oxidation rate (without modifying CATHARE calculation) for a selected oxidation law:
- CATHCART-PAWELL law, i.e. the standard CATHARE law,
- The CATHCART-PAWELL - law describing total oxygen consumption,
- the BAKER-JUST law is used for the calculation.
-
- SPALLOX simulates removal of the external oxide layer in the OXRATE parallel calculation.

- PICTG activates the calculation and storage in a separate file (PICTG) of the maximum temperature and maximum accumulated thickness of the internal and external oxide layers for each fuel rod and for three time intervals.

- the SCARFUEL option activates the properties of the UO_2 LIBELLULE material:

RSETFUEL adjusts CATHARE UO_2 porosity with LIBELLULE values,

STOPFUEL disables this adjustment model.

- ACTEMIS triggers an activity peak for radioactive chemical components to model the effect of an emergency shutdown. This instruction must be applied to the hydraulic circuit and acts on hydraulic elements in which fuel structures are defined. Users must specify the power decrease as a function of time.

Remark: As seen in §5.4.3.2, users must set the initial dimensions of the fuel pellet, gap, oxide layer and cladding.

5.4.4 FUELPLAQ element

Standard fuel elements are designed to model PWR fuel rods including their thermo-mechanical behaviour. Other types of reactor, such as experimental facilities or nuclear powered ships, use special fuel geometries and require the FUELPLAQ module. In these applications, the fuel thermo-mechanical behaviour is not addressed.
This sub-module can be used in conjunction with the point kinetics model (CORE) and only be defined in an AXIAL element. With the STOPNEUT directive, during the transient, the user can disable the kinetics model calculation and then use the fuel laws defined in the input deck.
A reflood module cannot be attached to the FUELPLAQ module.
FUELPLAQ user specifications are very similar to standard WALL specifications.

For FUELPLAQ, a contact resistance (option RESIST of WALL) can be added between the pellets and the cladding. In this case, the heat transfer coefficient (h) input in the data deck is used to add a term (1/h*e) to the thermal conductivity calculation of the last mesh of the pellet layer (LEFT keyword) or of the first mesh of the cladding layer (RIGHT keyword) ; e is the thickness of the considered mesh. The heat transfer coefficient is either given by the user (VALUE) either calculated by the code from the user's material properties (MODEL)

XNEULIST/XNEULISX, PRESHAPE/PRESHAPX, and PNRSHAPE/PNRSHAPEX key words apply to this element.

5.4.5 Facilities available for thermal sub-modules

5.4.5.1 SGCARACT sub-module

5.4.5.1.1 Description

The point steam generator model provides **simplified discretisation of the secondary side**.
Complete reactor schematisations usually describe the secondary circuit in the same way as the primary circuit, using a set of VOLUME, AXIAL and BCONDIT modules. For accidents (for example large-break LOCAs) in which the secondary side is isolated at an early stage, the point steam generator model offers a cost-effective secondary side representation.

The point steam generator sub-module offers a limited description of the secondary side:

- Sub-cooled liquid is not considered (liquid and steam are at saturated conditions),
- Tube dry-out cannot be described,
- No stratification, no re-circulation,
- No connection between several steam generators.

In the 0-D steam generator model, the secondary side is represented for the heat conducting WALL as an “external” side at a temperature T_{ext} with a given heat exchange coefficient HEXT. To allow “external” temperature changes during the transient, the code calculates the secondary side thermodynamic state in the following way:

- The steam generator is treated as a volume partly filled with saturated liquid at a specified pressure.
- The heat exchange coefficient HEXT is constant during the transient, so no “dry” parts of the SG tubes are represented how ever much water there is in the SG.
- The SG may be fed with a liquid at a specified temperature. It may also discharge (lose) saturated steam through “the steam line” (STMFLOW) and/or through “the relief valve” (VALVFLOW).
- The corresponding flowrates are set when defining the point SG (SGCARACT) and possibly in their modification laws (SGFEED) in the executable block.
- The mass and energy balance equations applied to the SG volume determine the resulting secondary side saturation temperature T_{ext} and the amount of liquid left.

The exchanged power calculation is explicitly coupled with the standard thermal hydraulic computation.

5.4.5.1.2 Other sources of information

The 0-D steam generator equations are specified in [DOC2] WHAT is a 0D STEAM GENERATOR?

5.4.5.1.3 How to proceed?

The SGPOINT option is available for a WALL defined in an AXIAL element (which models the SG tube). For the WALL, the user specifies the heat exchange coefficient with the secondary side HEXT (or HINI and HEND).

The user sets steam generator characteristics (SGCARACT):

- total steam generator volume,
- initial pressure,
- initial volume of water in SG,
- the various laws for feedwater liquid temperature, liquid and steam mass flowrates and the safety valve mass flowrate (these may be modified in the executable blocks in SGFEED),
- the fouling factor E or multiplication factor of thermal resistance between the primary and secondary sides (to modify the resulting heat exchange coefficient between the primary and secondary sides, and available for adjusting the steady state of the plant).

GVPPOWER defines initialisation of primary-secondary exchange. The user can choose:

- Either the power transferred to the secondary side (key word SGP). In that case it is also necessary to indicate the variable to be regulated to obtain this required power. The adjusted variable can be either the exchange coefficient, fouling factor or SG pressure. The solution is obtained by Newton's method.
- Or the feedwater flow.

During the calculation, SGFEED can be used to change steam generator boundary conditions (feedwater flow and temperature, steam outlet and relief valve mass flowrates).

5.4.5.1.4 Caution

Primary side WEIGHT
A steam generator tube, or more precisely the corresponding WALL, has its true internal and external diameters. They are specified in the WALL definition. Normally N steam generator tubes can be modelled with an average tube having a WEIGHT N.
Thus, the point SG volume, initial liquid mass, power, and all flowrates to be given in the input data (i.e. in SGCARACT, GVPPOWER, and SGFEED) should be equal to the total SG volume, initial liquid mass, power, and all flowrates divided by N. Only the secondary pressure and feedwater temperature are actual secondary state variables (not divided by N). In a sense, each primary tube communicates with a “small” SG and CATHARE automatically takes into account the number of tubes thanks to the WEIGHT factor.

Remark: The total power sent from the primary side to the secondary side is Pu:

$$Pu = N * H1 * S1 * (Tp1 – Tsat(Psecondary))$$

N : weight of the axial model of the SG tube,
H1 : actual heat exchange coefficient given by $1/H1 = 1/HEXT + R$ (fouling $E = 1/R$ if $R > 0$),
S1 : exchange surface given by the wall meshes on which the SGCARACT is created ($\pi \cdot D_{ext} \cdot L_{tube}$),
Tp1 : Secondary side wall temperature.

5.4.5.2 CORE sub-module

5.4.5.2.1 Description

The CORE sub-module is used to introduce a point kinetics model in the CATHARE calculation. Only one CORE sub-module may be defined in a CIRCUIT. Using this model, the user will give the initial nominal power of the core and CATHARE will calculate the total power at the beginning of each time step and will dispatch it to the corresponding fuel structures and hydraulics.

This sub-module can only be used with FUELCHAR or FUELPLAQ sub-modules.

The point reactor kinetic equations form a coupled set of 18 equations defining the total power P and the delayed neutron precursor concentrations as a function of time. Six groups of delayed neutrons and 11 groups of fission product decay are taken to keep track of ANS power decay. Nevertheless, in the CORE definition, the constants can be changed (see [DOC3] (CORE) and [DOC2] HOW TO MODEL CORE NEUTRONICS? for more information).

The total **power** is calculated from the kinetic equations and then dispatched to the fuel structures and hydraulic elements. The calculation is time-explicit. Consequently, users should take care when choosing the maximum time step value. A large maximum time step may cause numerical instability.

5.4.5.2.2 How to proceed?

Several CATHARE key words (CORE FUEL/FUEL3D, FUELCHAR, and FUELPLAQ) are used to define CORE characteristics. Refer to §5.4.6.1 for a key word definition summary.

The CORE name must be added to the end of the circuit list of elements (CIRCUIT).

The model must be activated using GONEUT. For the FUELPLAQ sub module, the CORE sub-module can be deactivated using the STOPNEUT directive.

Remarks:

- Power profiles may still be defined (FUELPLAQ or FUEL) and changed during the calculation (PNRSHAPE/PNRSHAPX for non-residual power, PRESHAPE/PRESHAPX for residual power).

- A multiplicative coefficient can be defined for each fuel structure (COEFKNEU or COEFPNEU in FUELPLAQ or FUELCHAR). This coefficient will dispatch non-residual and residual power, and calculate the impact of core reactivity on each fuel structure. These coefficients should consequently be treated as a means of weighting the fuel structure to reflect CORE behaviour. Take care not to use COEFPRES/COEFKRES.

- These coefficients are normalised with respect to the weight of the hydraulic element and fuel structure. For example, for residual and non-residual power for the fuel structure i:

$$\boxed{Power(i) = \frac{COEF(i)}{\sum_{j=1}^{n}\left(COEF(j) \times WEIGHT(j) \times NCHANNEL(j)\right)} \times Ptot}$$

where:

- n: total number of fuel structures (FUELCHAR) in the circuit,
- WEIGHT(i): weight of the hydraulic element to which the fuel structure i belongs,
- NCHANNEL(i): weight of the fuel structure in the hydraulic element,
- COEF(i): fuel structure multiplicative coefficient (COEFKNEU or COEFPNEU),
- Ptot: total power computed by the point kinetics model (residual or non-residual depending on power type).

- The power distribution coefficient (XNEUT) (power heating the fuel structure versus power going directly into the fluid) should be defined for each fuel structure (FUELCHAR or FUELPLAQ) and its profile may be changed during the calculation (XNEULIST/XNEULISX).

For a FUELPLAQ operator, the cancellation of the CORE sub module calculation and the used of specific user's fuel law can be defined as the following way:

- define in the data bloc the user's fuel law,
- on the FUELPLAQ element, specify and complete the keywords POWNEUT/POWRESI/LAWNEUT/LAWRESI,
- in the executable bloc, use the directive STOPNEUT

5.4.5.2.3 Reactivity calculation

CATHARE equations specify the time-dependent behaviour of the core power level with point kinetics **reactivity**. This reactivity is the sum of programmed reactivity (fuel reactivity and control-rod movement) and anti-reactivity based on changes in the average core-fuel temperature (Doppler effect), coolant density (moderator effect) and boron concentration. If this kind of reactivity definition is selected, it overrides those specified in the FUEL and FUELCHAR definitions.

- The reactivity effect due to boron is taken into account only if boron has been defined as a fluid component (see §5.8.2). On the other hand, it must be defined if it is present in the fluid. Users must define the coefficient of a predefined law R_{bor}([Boron]).

- Two methods may be used to take account of the moderator effect and Doppler effect:

  Using the CATHARE calculation: users must define a positive coefficient (A) (COEFM/COEFD in FUELCHAR, COEFKM/COEFKD in FUELPLAQ) and its profile (Za) (MODERAT/DOPPLER in FUEL/FUEL3D, AZKMOD or AZKDOP in FUELPLAQ).

  Defining user law coefficients (MODERAT/DOPPLER in CORE): in this case the corresponding function is integrated along the fuel structure without taking into account any of the profile and coefficient values.

The CATHARE reactivity calculation is described in Appendix 9.

Remark: if the LAWRESID key word is used (CORE), the moderator effect may be defined as a function of core density (LAWMODER) and the Doppler effect should be defined with a law dependent on the mean ROWLANDS temperature.

The key word SCRAM allows the effect of an emergency shutdown on the CORE to be modelled by defining an external anti-reactivity law as a function of time and an activity peak for radioactive chemical release in the hydraulic element where fuel structures are defined.

5.4.5.2.4 Other sources of information

Refer to [DOC16] for more information on the CATHARE point kinetics model.

5.4.5.3 REFLCHAR/REFLCH3D sub-module

5.4.5.3.1 Description

This sub-module can be used to model **the reflooding process**, which is the rewetting of a core by the safety-injection water after a clad temperature excursion.
In a reflooding situation, most of the energy accumulated in the wall of the heating rod (fuel wall or electric power) is sent to the fluid in the vicinity of the quench front where the axial temperature gradient is steep.

This calculation is available only on multi-layer walls (WALL/WALL3D) and fuel rods (FUELCHAR).

Standard wall heat transfer laws can describe this process, but tend to underestimate rod cooling and the speed of the quench front along the rods. This is due to three main reasons:

- Axial conduction in the wall plays an important role. Heat from the dry zone of the wall is transferred axially towards the cold rewetted zone. This transfer accelerates precooling of the wall and the rewetting temperature is reached earlier. This is not described with standard 1-D radial laws.
- In the vicinity of the quench front the quenching process disturbs the flow, increasing local heat transfer coefficients.
- Downstream of the quench front a large number of droplets are produced (by film sputtering).

This model should consequently be used when the calculation involves slow rewetting so that axial heat conduction can play a role in quenching; it may also involve limited coolant capabilities and a sufficiently high wet wall temperature (mandatory > Tmfs, better > 450°C).

To calculate quench front progression, precooling and the corresponding heat flux, a reflooding thermal conduction bi-dimensional model wall has been implemented in CATHARE 2 (this model is also called the PSCHIT model). Both TOP DOWN and BOTTOM UP reflooding sub-modules are available. With respect to the remarks above, this module has the following characteristics:

- A 2D radial and axial heat conduction calculation is added in the vicinity of the quench front. A moving 2D nodalisation is superimposed on the classical 1D wall calculation, advancing at the speed of the quench front along the heated structure.
- A local specific heat flux is added in the vicinity of the quench front to enhance heat transfers due to flow disturbance caused by quenching. It is taken as being proportional to the axial wall temperature gradient ($\varphi = K_2 \partial TW/\partial z$). This term may be very high in a very limited area.
- An enhanced film-boiling type heat flux is added (Qpvi term).

5.4.5.3.2 How to proceed?

5.4.5.3.2.1 START AND STOP CONDITIONS

The REFLOOD key word is used to activate the sub-module.

Whenever the REFLOOD key word has been used, bottom-up reflooding will start depending on the physical conditions of the quench front start location.

Depending on the physical conditions, for bottom-up reflooding, CATHARE will activate the true 2D reflooding model or just a 1D model that only follows the quench front, located in the scalar mesh just below the first wall temperature greater than the burn-out temperature. Refer to [DOC2] HOW to USE the BOTTOM-UP SUB-MODULE? for more information on the conditions under which CATHARE switches from 2D to 1D.

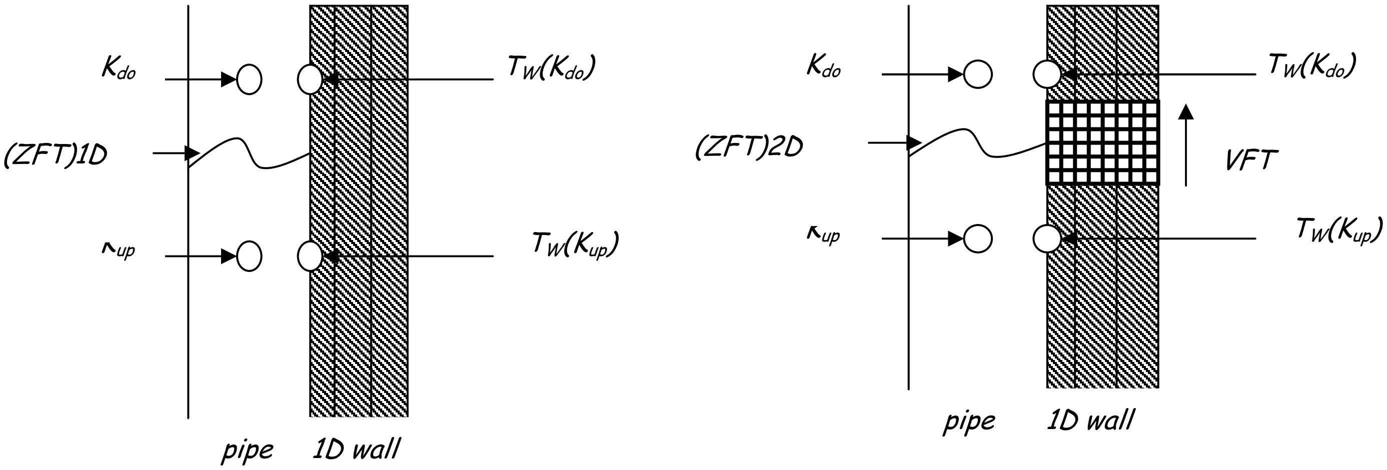


Figure 16: 1D and 2D models of bottom-up reflooding

For top–down reflooding, users should check the local thermal-hydraulic conditions because the code does not manage the launch time automatically. As regards the hydraulic scalar mesh in front of the quench front starting wall mesh, the CATHARE team recommends:

- the downstream mesh wet wall temperature must be higher than (tsat+100°C),
- the upstream mesh must have a wet wall temperature less than (tsat+40°C),
- the void fraction must be less than 0.995.

The REFLOOD key word should be used only when these conditions are fulfilled.
Remark: The key word POINTT (in REFLCHAR/REFLCH3D) enables the starting position of the quench front to be defined (by default it corresponds to the highest wall mesh).

The reflood model stops when the quench front reaches either the top or the bottom of the wall (depending on the kind of reflooding) or, if both bottom and top reflooding have been defined on the same wall, when they meet.

5.4.5.3.2.2 SUB-MODULE DEFINITION

To define the quench front, the user must specify (REFLCHAR/REFLCH3D):

- The local 2D PSCHIT nodalisation: INMI key word to define the number of radial meshes per wall material, DZPPS key word to define the axial length of meshes (and then the number of axial meshes),
- The PSCHIT nodalisation axial mesh number corresponding to the quench front level IQF.

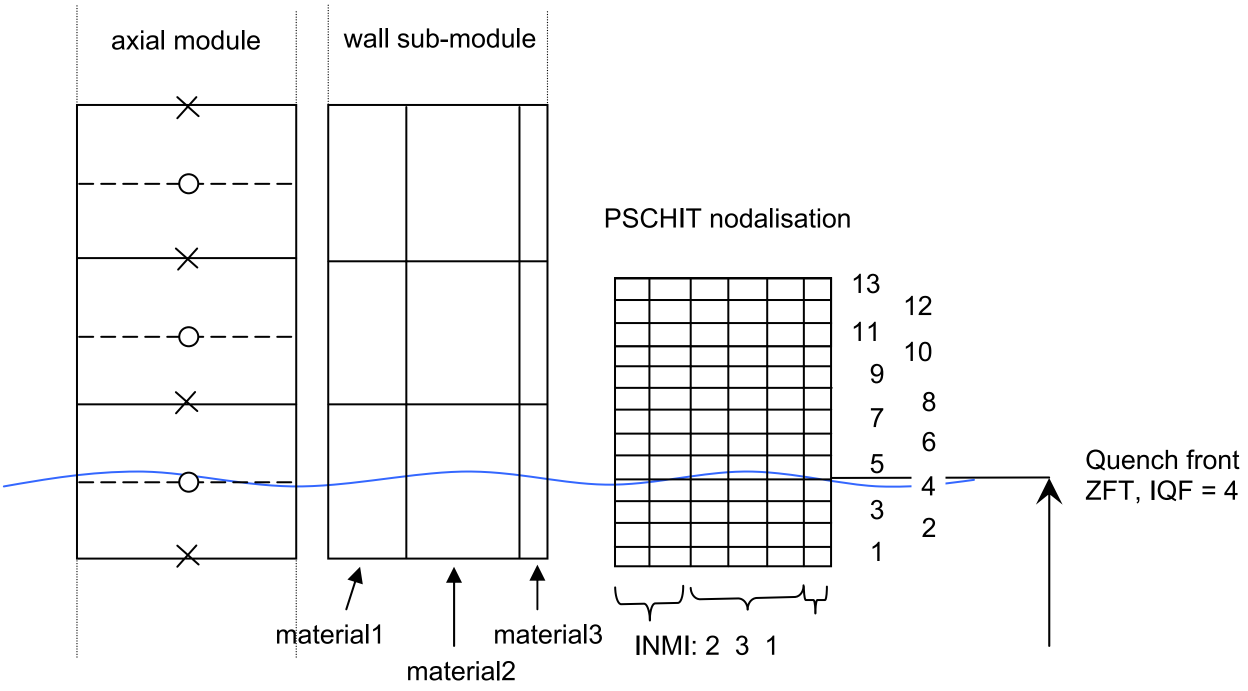


Figure 17: REFLCHAR/REFLCH3D definition

Care should be taken to keep more or less regular axial wall nodalisation (usually the length ratio for successive axial meshes should not exceed 1.2).

Radial nodalisation corresponds to the 1D reflooded wall or fuel rod. For each material layer, the number of radial meshes must not be smaller than in the 1D wall or fuel rod.
The total axial length must be approximately equal to the mean height of a hydraulic cell (+/-20%). If it is too small, physical results may not be consistent. If it is too large, there are some risks of overflow at the end of reflooding (if the reflooding reaches the end of the hydraulic mesh).
The mesh length downstream and upstream of the quench front (given by IQF value) must be similar (+/-10%) (but not the number of meshes).

The axial mesh must be refined in the vicinity of the QF (0.05 mm) to track the very steep axial temperature gradient.

Remark: Calculated temperatures
Locally, during the reflooding calculation, the wall mesh becomes finer. Wall temperatures given by the standard conduction calculation are only replaced in meshes which have been completely reflooded by the PSCHIT nodalisation (e.g. when the quench front level reaches the next wall mesh).

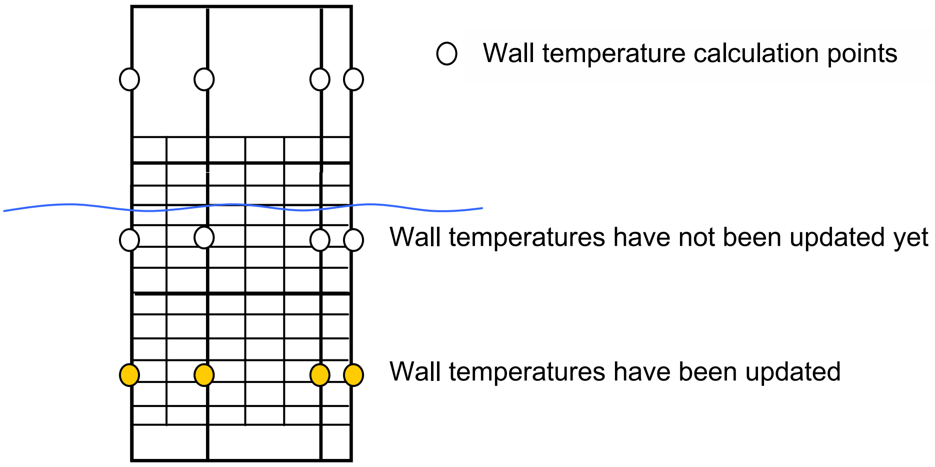

○ Wall temperature calculation points
Wall temperatures have not been updated yet
Wall temperatures have been updated

Figure 18: Wall temperature update during reflooding

5.4.5.3.3 Other sources of information

More information may be found in [DOC2] WHAT IS THE REFLOOD SUB-MODULE?

Refer to 0 for more information on reflooding physical laws.

Refer to [DOC2] for more information on:
HOW TO USE the TOP-DOWN REFLOOD SUB-MODULE ?
HOW TO-USE the BOTTOM-UP REFLOOD SUB-MODULE ?
WHY-WHEN TO USE a REFLOOD SUB-MODULE?

5.4.5.4 EXWALINK sub-module

5.4.5.4.1 Description

An explicit wall to wall link model is available in the CATHARE code (EXWALINK operator). The two walls are coupled thermally through this link, and they can exchange heat depending on the wall temperatures in the two walls. The communication is handled automatically by the code. The heat exchanged can be computed using a thermal resistance model (conduction), a simplified radiation model, or a combination of both.

5.4.5.4.2 How to proceed?

The **EXWALINK** operator is an **EX**plicit **WAll to wall LINK** between two walls. The two wall objects can be of type WALL or FUELPLAQ, and belong to

- the same circuit,
- two different circuits in a reactor.

The EXWALINK operator defines the link which must be activated using the SWITCH directive after the GOPERM directive. The link is not active during the PERMINIT.

5.4.5.4.3 Caution

The link must be defined before the circuit is created. It can only connect WALL and FUELPLAQ objects.
It cannot be defined for walls belonging to VOLUME or THREED objects. The two walls must have a compatible axial mesh. For the radiative heat transfer, it is assumed that:

- the linked walls may be considered of infinite length with respect to their distance;
- the walls are close enough to consider that the flux lost by one wall is equal to the flux received by the other. The model is not able to predict the fluxes exchanged between a small wall inside another one;
- the walls are “grey walls” so their respective emissivities do not depend on their temperatures.

5.4.5.5 AXICOND sub-module

5.4.5.5.1 Description

The AXICOND sub module is a simple model of axial conduction in a wall (1D, 3D WALL or FUELPLAQ sub modules) . The axial conduction calculation resolution is explicit.

5.4.5.5.2 How to proceed?

The AXICOND model is defined in the WALL directive but has to be enable using the AXICOND directive in the command block.

5.4.6 Modelling a reactor core

5.4.6.1 Power and reactivity definition with and without defining a CORE sub-module

	Power	Profile	Coefficient	Distrib. coeff.	Anti-reactivity	Profile
No CORE						
FUEL/ FUEL3D		*Axial Profile* WZUO *Radial Profile* LAW				
FUELCHAR	POWNEUT (nr) POWRES (res) or POWNEUT (nr) POWRESI (res)			XNEUT		
CORE	Nominal Power PO				BORON ([boron]) MODERAT (ρ_{mod}) DOPPLER (T_{UO2})	No profile available
FUEL/ FUEL3D		*Axial Profile* POWNEUT POWRES *Radial Profile* LAW				MODERAT DOPPLER
FUELCHAR			COEFPNEU (nr)	XNEUT	COEFM (mod) COEFD (dop)	
FUELPLAQ		*Axial Profile* ZKPOWNR ZKPOWRS *Radial Profile* LAW	COEFKNEU (nr)	XNEUT	COEFKM (mod) COEFKD (dop)	AZKMOD AZKDOP

NB: defining moderator and Doppler effects in the CORE element overrides those defined using the (FUEL, FUELCHAR) or FUELPLAQ key words.

5.4.6.2 Example of a standard dual-area - 1D representation of a core

5.4.6.2.1 Hydraulics

Consider a reactor with a core. The total power supplied by the core is PCORE. This core can be defined in NCORE areas. Each area can have a hydraulic weight (WEIGHT). For example, in the large-break input deck the core is modelled by two areas:

HOTCORE represents a central fuel bundle (264 rods),
MEDCORE is the remainder of the core (41184 rods).

The total flow area in the core (in this input deck) is 4.4282 m^2. Thus:

4.4 m^2 for MEDCORE,

0.0282 m^2 for HOTCORE.

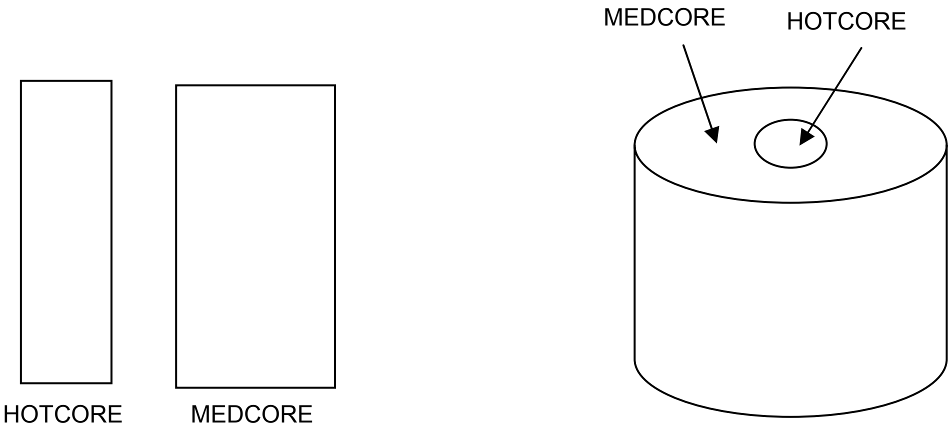


Figure 19: Areas modelling the core

In this example a different choice must be made to model both areas.

→ HOTCORE is modelled with a pipe (AXIAL operator with a flow area of 0.0282 m^2. The weight of this pipe is consequently WEIGHT = 1.

→ MEDCORE is modelled using a pipe with a flow area of 4.4 m^2 / 41184 = 10.6844 x 10^{-5}. Thus the weight of this pipe is WEIGHT = 41184.

5.4.6.2.2 Fuel structures

In each area (i.e. hydraulic element), the NSTRO fuel structure (FUELCHAR) can be defined (attached to this hydraulic element). Each structure can supply a fraction of the total core power.
Two user-defined parameters differentiate a structure:

- One fuel structure may represent several fuel rods. This parameter is set by the NCHANNEL key word (FUELCHAR). Then NCHANNEL identical rods are defined.
- With respect to power distribution, users can act directly upon the power dispatched to the fuel structure: by giving the power law or value directly when no CORE sub-module is defined or by defining a distribution coefficient COEFPNEU in FUELCHAR if the point kinetics model is used (see §5.4.5.2 for more explanation).

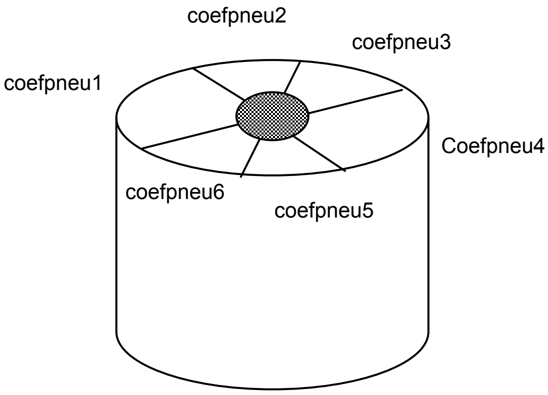


Figure 20: Fuel structures in the core

For one structure, the code normalises the axial power profile given by the user. For one axial mesh of the corresponding axial element (or "area"), the fraction of power supplied by one rod in a given structure is (no CORE sub-module defined):

$$\boxed{\frac{\text{WZUO . mesh length}}{\text{rod length}}}$$

The WZUO value is interpolated from user-defined data (FUEL).

When using the CATHARE CORE module, WZUO should be replaced by the zpownr value given after the POWNEUT key word (in FUEL/FUEL3D after NEUTRO) for non-residual power, and zpowres (after POWRES) for residual power.

In addition, users can indicate a radial profile (named wruo in [DOC3]) for the power supplied in the fissile part of the rod (LAW in FUEL/FUEL3D).

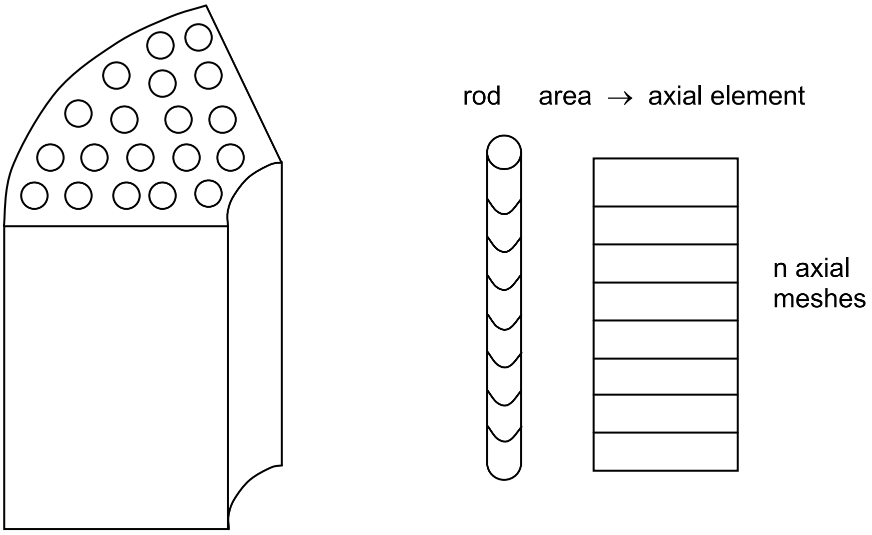


Figure 21: Connection between rod and area

Thus the power supplied by a mesh

$$\Big(\underset{\nearrow\ \text{axial}}{I},\quad \underset{\nwarrow\ \text{radial}}{J}\Big)$$

is: wruo (I, J) $\boxed{\dfrac{\text{WZUO}_I \text{ . length of I}^{\text{th}}\text{ mesh}}{\text{rod length}}}$

Remark: A structure is defined as belonging to an area (or is linked to an AXIAL element) with a weight WEIGHT. This means that the total number of rods of this structure is:
NRODS = WEIGHT x NCHANNEL.

LINKING THE FRACTION OF POWER COEP WITH THE RADIAL FACTOR fi GENERALLY USED IN REACTOR PROBLEMS.

If NTOTRODS is the total number of rods, the average power for one rod is:

$$\text{<P>} = \boxed{\frac{\text{PCORE}}{\text{NTOTRODS}}}$$

For a given fuel structure, the radial factor fi is defined as a function of the average power <P>.

The power of the structure is therefore:

P^s = fi . <P> . NRODS
= fi . (PCORE/NTOTRODS) . NRODS

COEFP can be defined in the input deck:

P^S = COEFP. PCORE

Then:

$$\boxed{\text{COEFP} = \text{fi}\ \frac{\text{NRODS}}{\text{NTOTRODS}}}$$

with NTOTRODS = Σ (NCHANNEL. WEIGHT) over all the fuel structures.

When using a CORE sub-module, COEFP is used directly in the FUELCHAR definition as COEFPNEU or COEFKNEU (FUELCHAR or FUELPLAQ).
If not, it should be taken into account by users when defining the power value law for residual and non-residual power (FUELCHAR).

5.4.6.2.3 Energy balance : summary

- Assuming steady state conditions in an area, or in an axial element with a weight WEIGHT, the power released to the fluid is:

POWER = WEIGHT . Q . [outlet enthalpy - inlet enthalpy]

and: flowrate INSIDE this element

PCORE = $\sum\limits_{1}^{\text{number of areas}}$ POWER is the total power supplied by the core to the coolant.

- In a fuel structure, the power generated is:

$$\boxed{P^S = \frac{\text{COEFP}}{\Sigma\text{COEFP}} POWER}$$

- Power generated by one rod of a structure:

$$\boxed{\frac{\text{COEFP}}{\Sigma\text{COEFP}} \frac{\text{POWER}}{WEIGHT\ .\ NCHANNEL}}$$

-
- Power supplied in one axial mesh

$$\boxed{\frac{\text{COEFP}}{\Sigma\text{COEFP}} \frac{\text{POWER}}{WEIGHT\ .\ NCHANNEL} \frac{length\ of\ mesh\ i}{\text{rod length}} WZUO_i}$$

- Power generated in the fuel radial mesh j in front of the hydraulic mesh i

$$\boxed{\frac{\text{COEFP}}{\Sigma\text{COEFP}} \frac{\text{POWER}}{\text{WEIGHT}\ .\ NCHANNEL} \frac{length\ of\ mesh\ i}{\text{rod length}} WZUO_i\ \text{wruo}(i,j)}$$

- Total power of the core

Total power can be defined by the point kinetics model (see §5.4.5.2)

It can be defined by the user (FUELCHAR) as the sum of POWNEUT and POWRES values for each fuel structure:

$$\boxed{POWNEUT + POWRES \quad = \quad P^S = \frac{\mathrm{COEFP}}{\Sigma\mathrm{COEFP}} POWER}$$

5.5 Gadgets

Gadgets are components set on the main hydraulic modules to represent:

- plant hydraulic devices such as valves, sinks, sources, accumulators and safety injection systems,
- accident description,
- pump and turbine systems.

They can be considered as a local modification of the hydraulic flow and are then represented as point modifications of the standard thermal hydraulic equations.

Generally, to be taken into account in thermal hydraulic computation, these objects not only have to be defined in the input deck but also have to be activated during the calculation (see Appendix 13 – gadget cross reference table).

5.5.1 ACCU gadget

5.5.1.1.1 Description

The ACCU element models **a tank and an injection line**; it can therefore be used to model a safety injection, for example an accumulator. It can be connected to a VOLUME or an AXIAL element.

The tank is filled with water and nitrogen. The discharge of the accumulator occurs automatically as soon as the pressure at the injection point is lower than the internal pressure of the accumulator (and stops otherwise). The nitrogen, considered to be a perfect gas (but with user-defined expansion number), expands and the water is injected into the circuit.

Three equations are written to calculate the discharge flow:

- liquid mass balance,
- adiabatic gas expansion,
- momentum balance in the expansion line (gravity, frictional pressure losses).

Mass, energy and momentum sources terms are added to the pipe or volume equations according to phase separation laws. When connected to a pipe, a specific condensation model is used in the injection mesh to take jet effects into account.

The check valve is modelled and the accumulator will discharge only if conditions for opening the check valve are fulfilled (i.e., $P_{\text{upstream of valve}} < P_{\text{downstream (in accumulator)}}$).

The coupling between accumulator discharge and standard circuit calculation can be either explicit or implicit.

The pressure drop due to the valve hysteresis can be modelled using the keyword “VALVE, A, B”, where A and B are the coefficients of the following equation: $\Delta P = A + B \times Q_G^2$ .This model can be activated or deactivated with the directive ENABLE/DISABLE.

After total discharge of the water, there are two possibilities: either the valve is closed and there is no more injection or the valve remains open and nitrogen is discharged into the circuit. In the latter case, nitrogen should have been defined in the circuit fluid composition (see §5.8.1 and NINCON).

The water in the tank can optionally contain diluted nitrogen and/or boron.

5.5.1.1.2 How to proceed?

The definition of an accumulator requires the geometrical characteristics of the tank and of the injection line, the initial physical state of fluid inside the tank (the initial pressure, the initial water enthalpy and the “adiabatic” expansion coefficient of nitrogen and, optionally, the mass fraction of diluted nitrogen or boron in the water) (see Figure 22, Figure 23 and Figure 24).

First activation is automatic (no need to use OPEN for connection).

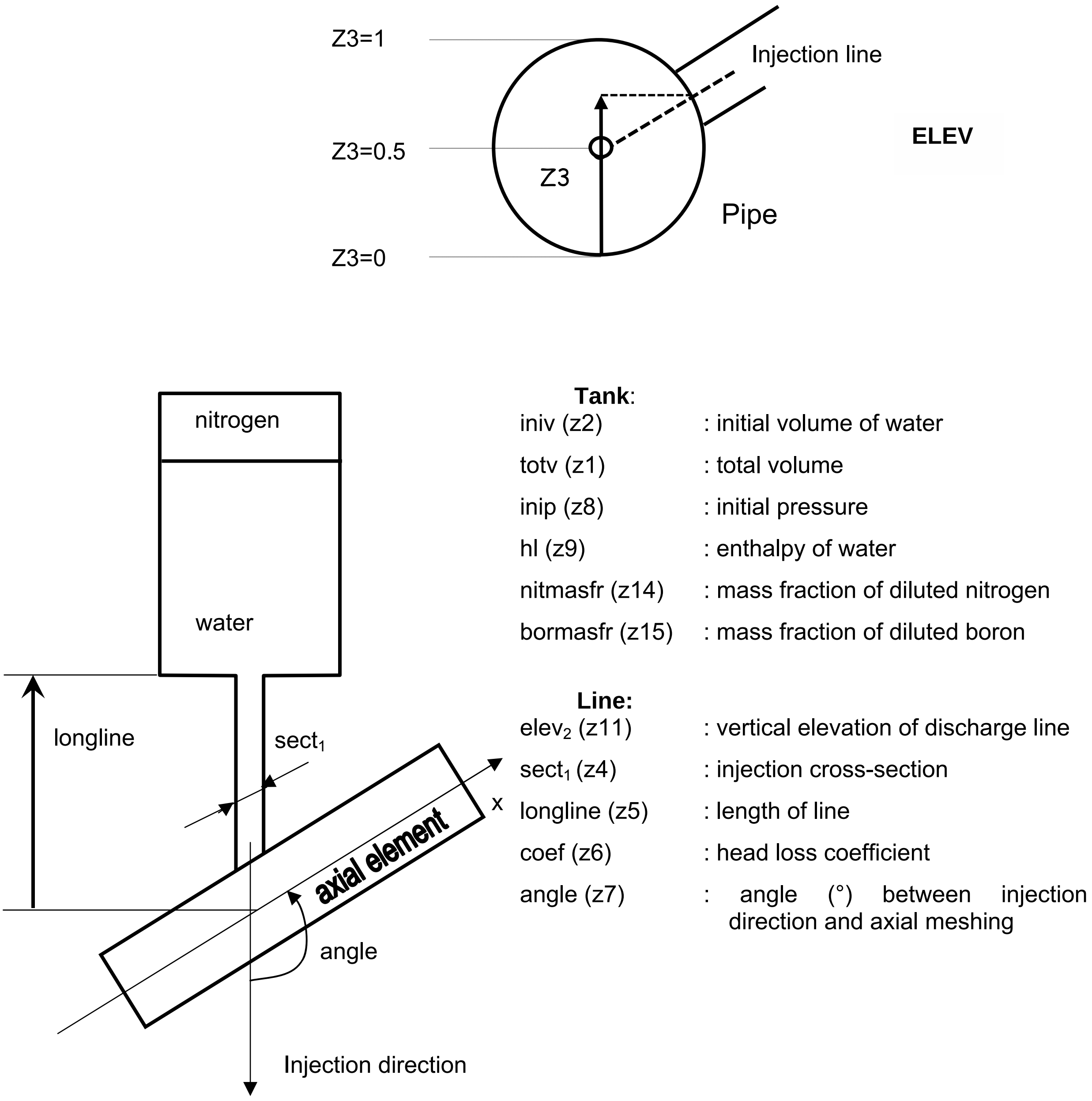
The CLOSE directive can be used to prevent automatic discharge because it disconnects the gadget from the main module (in this case, OPEN may be used to reconnect it).

5.5.1.1.3 Caution

Nitrogen and/or boron must be defined in the circuit in order for it to be injected by the ACCU gadget.

5.5.1.1.4 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12]. Refer also to [DOC2] HOW TO MODEL an ACCU ?



Tank:

iniv (z2)	: initial volume of water
totv (z1)	: total volume
inip (z8)	: initial pressure
hl (z9)	: enthalpy of water
nitmasfr (z14)	: mass fraction of diluted nitrogen
bormasfr (z15)	: mass fraction of diluted boron

Line:

$elev_2$ (z11)	: vertical elevation of discharge line
$sect_1$ (z4)	: injection cross-section
longline (z5)	: length of line
coef (z6)	: head loss coefficient
angle (z7)	: angle (°) between injection direction and axial meshing

Figure 22: Case of an accumulator connected to a pipe

elev$_1$ (z3) : elevation with respect to the bottom of the volume
length (z7) : penetration length of the injection line in the volume

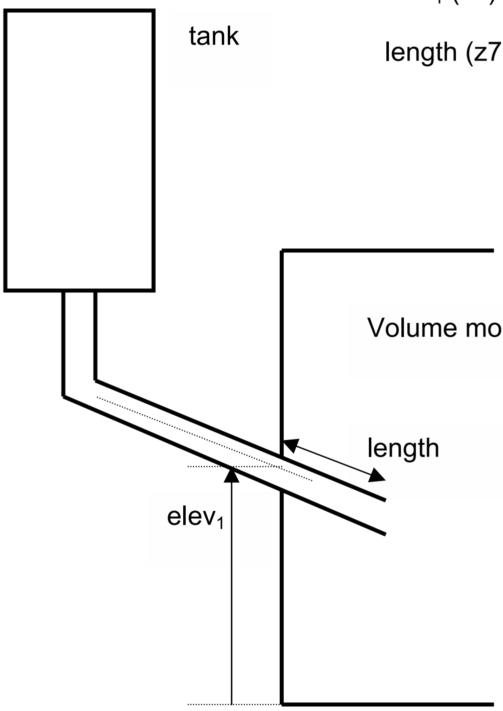
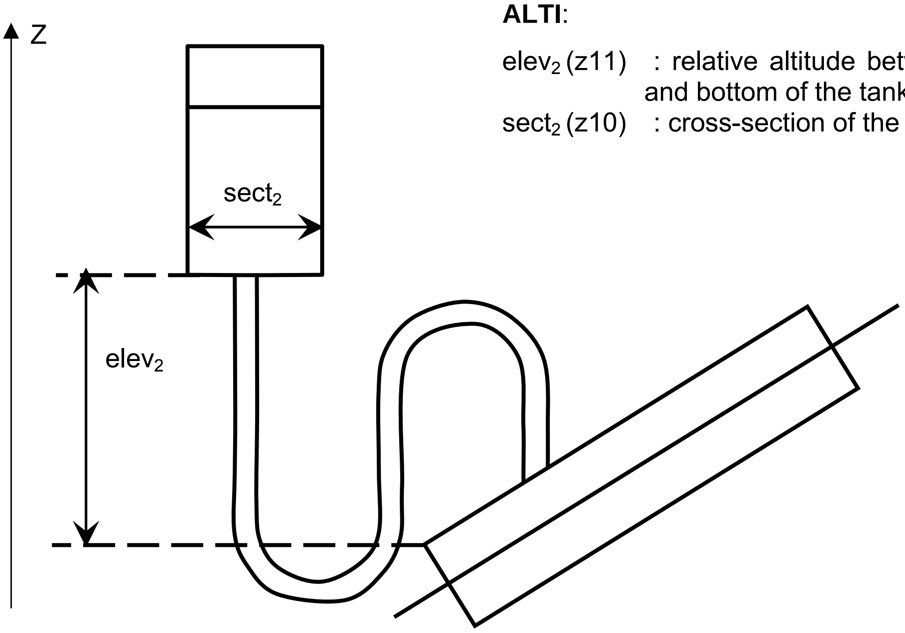


Figure 23: Accumulator connected to a volume

ALTI:

elev$_2$ (z11) : relative altitude between injection point and bottom of the tank
sect$_2$ (z10) : cross-section of the tank



Given TOTV, gravity can be taken into account in the accumulator discharge.

Figure 24: Taking into account the gravity in an accumulator

5.5.2 CANDLE gadget

5.5.2.1.1 Description

This gadget is used to model **a pure energy source or sink**. It can be connected to a VOLUME, an AXIAL or a THREED element.
It allows the user to inject power directly in the liquid or gas phases without phase change or into the interface to be directly used in phase change (not available in a VOLUME element).
It is connected to a scalar point (defined by the user for AXIAL and THREED elements and depending on the level in a VOLUME).
The power injected can be defined externally (can be changed only time step by time step) or internally through user laws (in which case an interpolation is used to define the injected power at each iteration).
Refer to [DOC3] (CANDLE) for more information.

5.5.2.1.2 How to proceed?

The gadget needs to be connected to be taken into account in the calculation (OPEN/CLOSE).

5.5.3 CCFL gadget

5.5.3.1.1 Description

This gadget is used to model the **counter-current flow limit**.
It can be connected to a bottom junction between a VOLUME and an AXIAL element, or to an AXIAL or THREED element (not in a horizontal configuration). It is linked to a vector point (or an assembly of vector points in a THREED element).

In reflux condenser mode, the CATHARE code provided with the CCFL option can predict the liquid down-flow to the core as a function of steam up flow.
CCFL must be controlled at the upper core plate and at the inlet to the SG U-tubes.

5.5.3.1.2 How to proceed?

This gadget needs to be connected to be taken into account in the calculation (ENABLE/DISABLE).

The user enters only the parameters Mn, Cn and ε in the data block:

$$\boxed{J_G^{*\nu} Bo^{\nu\varepsilon/2} + Mn J_L^{*\nu} Bo^{\nu\varepsilon/2} = Cn}$$

with

- for WALLIS-type correlation: ν=0.5, ε=0;
- for KUTATELADZE-type correlation: ν=0.5, ε=1;
- and $\nu \neq 0.5$ when it allows better data fitting.

The classical flooding correlations will automatically be translated for the 6-equation model.

Besides these parameters, when defining this gadget (CCFL), the user has to define several other parameters, including:

- DIAM: Diam is used to define the non-dimensional velocity and the Bond number (Bo).When using a KUTATELADZE -type correlation, this value is not used.
- SRATIO: Sratio is the ratio of the modelled flow section to the flow section used in the correlation (see Figure 25: CCFL definition - SRATIO parameter).

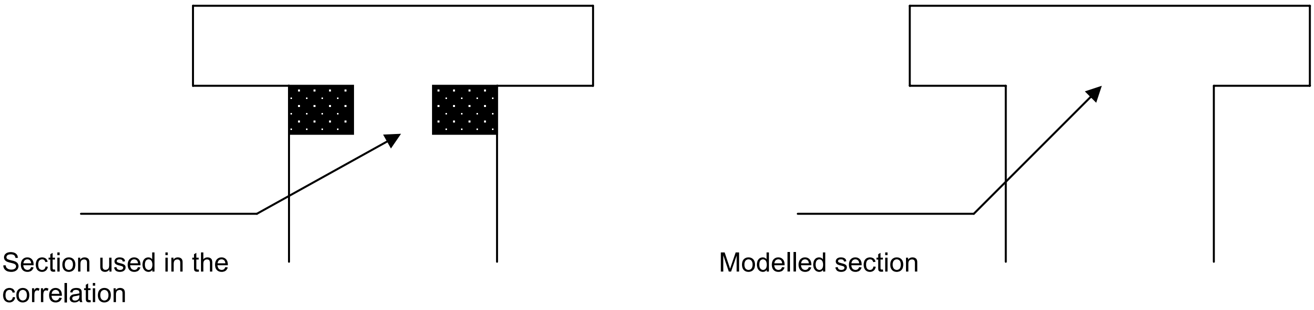
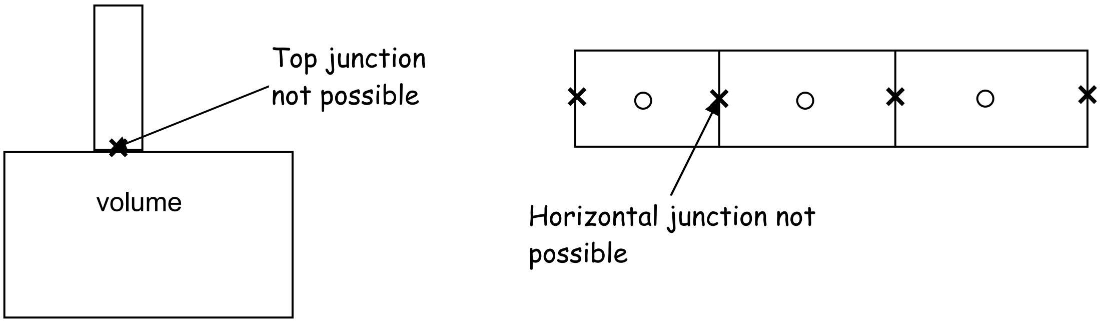


Figure 25: CCFL definition - SRATIO parameter

5.5.3.1.3 Caution

The following configurations are not possible:



5.5.3.1.4 Other sources of information

Refer to [DOC2] HOW to CONTROL CCFL and FLOODING LIMIT? for a general presentation of this physical phenomenon and to HOW to USE the CCFL SUB-MODULE? for more information on this specific gadget.

5.5.4 Pump, turbine and compressor gadgets

5.5.4.1 PUMPCHAR gadget

5.5.4.1.1 Description

The so-called PUMPCHAR element is used to model **a pump**.

The model describes the interaction between fluid and outward flow pump. It calculates the pump head and rotation speed and the heating of the fluid crossing the pump using the head and torque characteristics with a possible degradation in two-phase flow. So the point pump model is treated as a source of energy and momentum connected to a vector point of an axial module.

5.5.4.1.1.1 Variables

The working point of a pump depends on two parameters:

- the angular rotation speed ω (rd/s or RPM),
- the inlet volumetric flowrate q (m^3/s),

and is characterised by two variables:

- the head H (m),
- the hydraulic torque C_h (N.m).

In single-phase flow, the head is independent of the fluid density, so a specific torque T is defined:

$$T = \boxed{\frac{C_h}{\rho}}$$

ρ: density of the fluid (kg/m^3)

5.5.4.1.1.2 Homologous representation and characteristics

The pump head in the first quadrant (see Figure 26), could have the following approximated formula:

$$\boxed{H = A\omega^2 + B\omega q + Cq^2 \qquad (1)}$$

For one reference flowrate q_{ref} and one rotation speed ω_{ref} the head value is h_{ref} and the specific torque value is T_{ref}.

These are called the reference values.

Hence, the reduced variables are defined:

- reduced speed	$\alpha = \omega/\omega_{ref}$
- reduced volumetric flowrate	$\nu = q/q_{ref}$
- reduced head	$h = H/H_{ref}$
- reduced specific torque	$\beta = T/T_{ref}$

By similarity, expression (1) becomes:

$$\boxed{\begin{cases} \dfrac{h}{\alpha^2} = f\left(\dfrac{\nu}{\alpha}\right) \\ \text{or} \\ \dfrac{h}{\nu^2} = g\left(\dfrac{\alpha}{\nu}\right) \end{cases}} \qquad (2)$$

h/α^2 and h/ν^2 are called the "homologous head".

By convention the following are used:

$$\boxed{\frac{h}{\alpha^2} \ for \left|\frac{\nu}{\alpha}\right| \leq 1}$$

and

$$\boxed{\frac{h}{\nu^2} \ for \left|\frac{\alpha}{\nu}\right| \leq 1}$$

thus allowing the homologous head to be represented as a function of a reduced variable in the range -1, +1.

Similarly, the torque T could be changed and represented in homologous variables.

The choice of reduced variables is given by the running octant (for a pump, 4 quadrants and 8 octants, but octants 7 and 8 are not described).

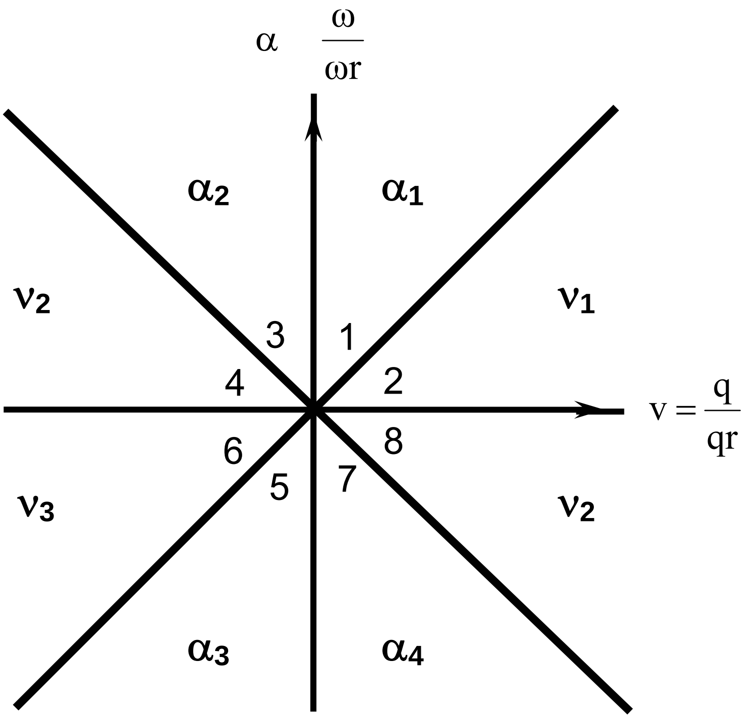


Figure 26: Quadrants of a pump

The index a (resp. ν) indicates α > ν (resp ν > α). The numbers 1, 2, 3 or 4 indicate the quadrant number. The pump characteristics, given by the curves H (q, ω) and C (H, ω), are generally given under the homologous form.

5.5.4.1.1.3 Shaft equation

The basic equation for the pump is the shaft equation:

$$J \frac{\partial \omega}{\partial t} = \Sigma \text{ torques}$$

J = inertia (kg.m^2)

Then:

$$J \frac{\partial \omega}{\partial t} = C_m - C_h - C_f$$

C_m = motor torque
C_h = hydraulic torque
C_f = frictional torque

When the rotation speed is imposed, this equation is not applied. It is only necessary to know the value of the rotation speed.

Motor torque

Reduced form:

$$C_m = \boxed{\frac{C_{nm} \cdot \gamma \left(a_m + b_m |\gamma|\right)}{\left(C_m \cdot \gamma\right)^2 + \left(a_m + b_m |\gamma|\right)^2}} = C_{nm} * F(\gamma)$$

γ : dynamic sliding = 1 - $\omega/\omega s$
ω_s : synchronism speed (rd/s or RPM)
C_{nm} : nominal motor torque (N.m):
a_m, b_m, c_m : constants depending on the power supply
(see Figure 27: Motor torque).

Friction torque

Expression:

$$C_f = \left(a_f + b_f|\omega| + c_f|\omega||\omega| + d_f Pe\right)*\frac{\omega}{|\omega|}$$

P_e : inlet pressure
a_f, b_f, c_f, d_f : frictional torque coefficients.

5.5.4.1.1.4 Degradation in two-phase flow

For two-phase flow, the characteristics given for single-phase flow are used with the addition of a "degradation" function depending of the void fraction "alfa" called M(alfa):

The head in two-phase flow becomes:

$$h_{2\varphi} = [1 - M_h(\text{alfa})] \,.\, h_{1\varphi} + M_h(\text{alfa}) \,.\, h_{1\varphi\,-2\varphi}$$

$h_{2\varphi}$: two-phase head
$h_{1\varphi}$: single-phase head
$h_{1\varphi\,-2\varphi}$: fully degraded characteristics
M_h(alfa) : degradation function for the head (see Figure 28)

A similar expression is used for the torque β.

5.5.4.1.1.5 Cavitation

To take into account the degradations of the pump height and torque due to cavitation, a similar formulation is used. The degradation function becomes $M_h = M_h(\alpha) + M_c(\alpha, T_l\text{-}T_{sat}, \text{NPSH})$.

NPSH (Net Positive Suction Head) characterises the ability of a fluid to cavitate.

A similar expression is used for the torque; the degradation due to cavitation is the same for height and torque.

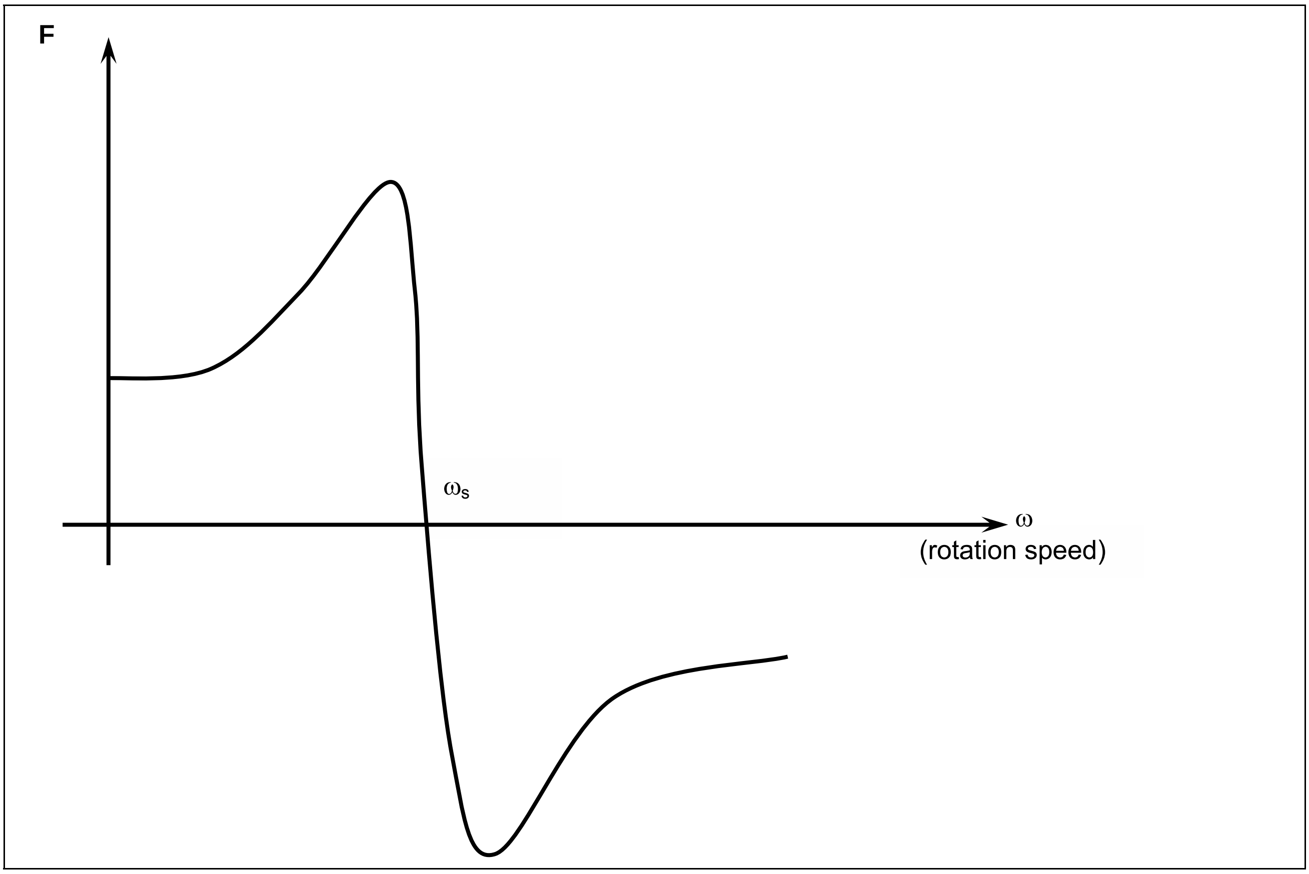


Figure 27: Motor torque

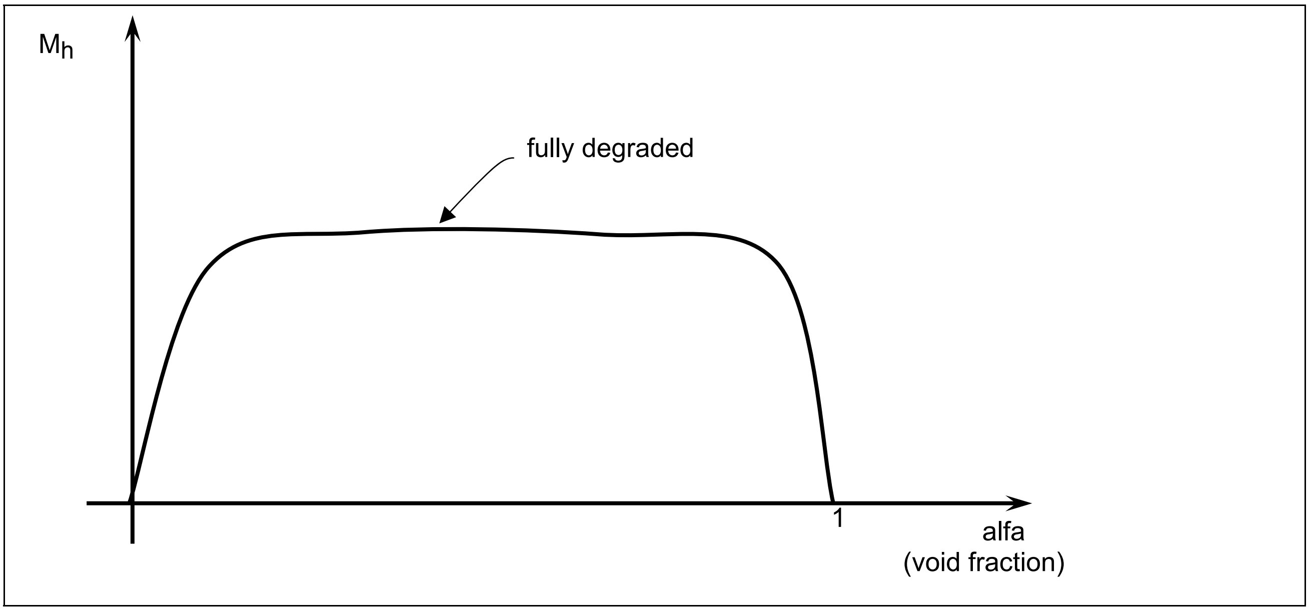


Figure 28: Degradation function for the head

5.5.4.1.2 CATHARE facilities

5.5.4.1.2.1 Predefined models

CP1, N4, P4, RRA, EVA, LOFT, LOBI, BETHSY pump characteristics are available (LIBPOM.f), also in degraded conditions (LIBDEG.f). Cavitation characteristics are available only for CP1, N4, P4 and RRA models.

5.5.4.1.2.2 User models

It is possible to use a personal model using FREE key word. In this case, the following pump characteristics have to be used:

- The characteristics for the pump are given in a discretised manner. They are smoothed by the code using polynomials: head polynomial and torque polynomial.
- For each octant (OCTANT key word) (except 7 and 8) the user must give (single-phase values): discrete values (or points) situated on the head and torque characteristics. These points are given as triplets (x, y, z) satisfying the relation (2).

OCTANT	1	2	3	4	5	6
x (NUALF key word)	ν/α	α/ν	ν/α	α/ν	ν/α	α/ν
y (HOMH key word)	h/α^2	h/ν^2	h/α^2	h/ν^2	h/α^2	h/ν^2
z (HOMTORQ key word	β/α^2	β/ν^2	β/α^2	β/ν^2	β/α^2	β/ν^2

The number of the triplets is given after the NOCTP key word. The degrees of head (resp. torque) polynomials are given after the DHEIGHT (resp DTORQUE) key words.

- The deteriorated characteristics of the pump can be specified.
- Data for the motor torque of the pump (MOTOR key word).
- If the code calculates the rotation speed with the shaft equation (3) the motor torque coefficients a_m, $b_{m,}$ c_m should be given (after the TORQUE key word).
- If the motor torque coefficients are not known, or if the rotation speed is to be imposed for the whole calculation, use the key word NOTORQUE instead of TORQUE. In that case, the imposed rotation speed must be given (after the VELOCITY ROTATION key words). The code will then not apply the shaft equation to calculate the rotation speed.
- Value of inertia (INERTIA key word): the value of inertia is used when the shaft equation is applied; it is not used when the pump rotation speed is imposed.
- Value of the synchronism speed and imposed speed (if necessary) (VELOCITY key word): in both cases (shaft equation applied or not) the synchronism speed should be given (this value is used only in the case where the shaft equation is applied).
- Shaft equation applied: give the value after the RPM or RADSEC key word (depending on the unit).
- Imposed speed (meaning that the shaft equation is not applied): use the ROTATION and IMPOSED key words, then RPM or RADSEC and give:
  - the synchronism speed value,
  - the imposed rotation speed law.

Remark: To perform the rotation speed computation use CALROTA.

- Values of the frictional torque coefficients (FRICTION key word). These values (a_f, b_f, c_f, d_f) are used only when the shaft equation is applied.

- Values for the pump:

Then give the reference values for:

- rotation speed ω_{ref} (NOMV key word),
- flowrate q_{ref} (NOMFLOW key word),
- head H_{ref} (GRAVH key word),
- specific torque $T_{ref} = C_{ref}/\rho_{ref}$ (NOMTORQ key word).

5.5.4.1.3 Caution

Very often the user indicates the rated values as reference values. However, it is not obligatory to choose this state as reference state.

5.5.4.1.4 How to proceed?

Define the PUMPCHAR object in the data block.

Activate it with STARTPUMP, STOPPUMP directives (switch on or off the power supply).

The characteristics of a pump can be modified during the calculation. The PUMPMOD directive modifies the pump characteristics in an axial element.

PCAVIT can be used to activate or stop the cavitation model activation.

BLKROTOR can be used to simulate a pump failure.

5.5.4.1.5 Other sources of information

The basic equations and the numerical analyses are described in [DOC14].
Refer also to [DOC2] HOW TO MODEL a PUMP ?

5.5.4.2 TURBINE element

5.5.4.2.1 Description

This gadget is designed to model **a BWR turbine**.
It can be connected to an AXIAL element.

For this gadget, two ways of calculation are available: the user can define either the power to be imposed or the efficiency. The gamma coefficient (Cp/Cv) also has to be defined by the user.
During the calculation, these values can be changed by the user (WRITE), but to change the calculation mode from power to efficiency (respectively from efficiency to power) the MODPNCST key word (respectively MODPCST) must be used.

A special SINK gadget (SINK with PURGE option) can be connected to the AXIAL element immediately after the TURBINE gadget to eliminate the potential liquid flowrate appearing after the TURBINE. This is characterised by its efficiency (sink real liquid flowrate Ql_{sink} = efficiency * Ql_{axial}).

5.5.4.2.2 How to proceed?

It is automatically connected when defined.

5.5.4.2.3 Caution

Turbine power has to be defined before changing from efficiency to power mode and vice versa.

5.5.4.3 TCOMCHAR and SHAFT gadgets

5.5.4.3.1 Description

The TCOMCHAR module is:

- designed to model a turbine or a compressor of a gas reactor,
- connected to the vector mesh of an AXIAL element.

Several TCOMCHAR gadgets may be grouped together using the SHAFT gadget.

The SHAFT module is:

- designed to model a shaft to slave several TCOMCHAR gadgets,
- attached to a TCOMCHAR gadget that is defined as the master element of the SHAFT, the one which imposes the velocity,
- automatically connected when defined.

5.5.4.3.2 How to proceed? (single TCOMCHAR)

Define it in the data block (TCOMCHAR).
This gadget is automatically connected when defined.
There are three ways to define the TCOMCHAR characteristics:

mod 1 by giving the coefficients of the interpolation law used for head and torque data.
mod 2 by using the library.
mod 3 by using designated non-dimensional characteristics of the turbine or compressor (pressure ratio, isentropic efficiency).

5.5.4.3.3 How to proceed? (coupled TCOMCHAR)

The user must define which TCOMCHAR objects are linked by the shaft and then define a group.
This group may be deactivated (STOPSHAF) and reactivated (STARSHAF) during the calculation.
An alternator may also be defined to add a torque to the group. This coupling may be stopped (STOPALTR) and restarted (STARALTR) during the calculation.
Characteristics of the SHAFT can be changed during the calculation using the SHAFTMOD key word.

5.5.4.3.4 Caution

If the definition uses the ROTATION key word in the data block, the imposed velocity mode is used. Then the velocity will not result from the shaft equation and the alternator disconnection cannot be predicted in this way.
When using SHAFT in a transient such as loss of load (alternator disconnection), the time step tends to be limited to 0.05s. DTMAX should then not be chosen much larger.

5.5.5 SENSOR gadget

- This gadget has been designed for simulator coupling use to provide access to physical variables of the plant. In most cases, it can be successfully replaced by a direct access to CATHARE Calculated Values.
- This is a passive module which has no impact on the calculation.
- It can be connected to AXIAL, 3D, WALL or VOLUME elements depending on the kind of variable requested.
- It is automatically activated when connected.

Nevertheless, its use may be advantageous in order to avoid external post-processing because it allows the evaluation of several variables outside the CATHARE calculation mesh points using interpolation:

Sensor name	AXIAL ELEMENT	3D	VOLUME	WALL
TEMPERATURE	scalar point		**elevation**	**elevation**
PRESSURE	scalar point or **elevation**		**elevation**	
FLOWRATE	vector point			
MAILSIPA	vector point			
ACTI	scalar point		**elevation**	
QACTSIPA	vector point			
UTRASON	scalar point			
SWLLEVEL	whole axial	3D zone		

The SWLLEVEL SENSOR is a 'zone sensor'. That means that it takes into account the whole AXIAL element, or a defined zone for a THREED element. The SWLLEVEL SENSOR is described in [DOC2].

5.5.6 Accident leaks, break gadgets

5.5.6.1 BREAK element

5.5.6.1.1 Description

This gadget is used to model **a break with critical flow conditions**.
It corresponds to an implicit break term (sink term using a critical mass flowrate correlation known as the GROS D'AILLON correlation). No reverse flow is allowed.
In the boron-activity calculation, ejection of boron or activity by a break occurs automatically, the boron or activity concentration of the ejected phase being that of the mesh corresponding to the break term.

5.5.6.1.2 How to proceed?

A break has to be defined on a scalar point in an AXIAL element in the data block.
The break calculation is enabled / disabled using OPEN/CLOSE directives.

5.5.6.1.3 Caution

This allows a simple schematisation of a break along a pipe but the critical flowrate is not very accurate and may lead to different predictions compared to those made when using the recommended schematisation (TEE + AXIAL+ BCONDIT modelling of the break). It cannot take phase separation effects into account.

5.5.6.1.4 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12].

5.5.6.2 PIQBREK element

5.5.6.2.1 Description

This gadget is used model **a leak/fluid removal**.
It can be connected to an AXIAL or a VOLUME element.

The main difference with the BREAK gadget is that sub-critical flowrate is possible (sonic discharge, again using the GROS D'AILLON correlation). The calculation is implicit. Reverse flow is then physically possible and is supported by CATHARE if external conditions have been defined (external temperature or enthalpy for gas and liquid, void fraction and pressure, and if they exist in the circuit, mass fractions/concentrations of non-condensable gases or radiochemical elements).

This gadget also allows a purge (PURGE) or an event (EVENT) to be modelled by defining a void fraction threshold below (respectively above) which only liquid (respectively gas) is extracted.

5.5.6.2.2 How to proceed?

The gadget must be defined in the data block.
The PIQBREK calculation is enabled / disabled using the OPENBREK / CLOSE directives (location can be set up too).
This is an external gadget, meaning that it is controlled by the user within the executable block of the data set using CCV variables.

5.5.6.2.3 Caution

Flow reversal is possible and external conditions have to be defined by the user in this case.

5.5.6.2.4 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12].

5.5.6.3 SGTR element

5.5.6.3.1 Description

This gadget is used to model a single or multiple steam generator tube rupture.
It can be connected to two AXIAL elements, modelling the primary and secondary sides of a steam generator.

5.5.6.3.2 How to proceed?

When defined in a data block, this gadget creates a hydraulic connection between a primary circuit and a secondary circuit. It is defined between two axial elements, modelling primary and secondary side respectively. The two axial elements can belong to the same CIRCUIT, in which case the calculation is implicit, or to two different circuits, in which case the calculation can be implicit or explicit (if an explicit type EXCHANGER has been defined between these two axial elements).
The user gives the characteristics of the SGTR and the flow is calculated by the code. The SGTR can model either small leaks (key word GVFUIT) or one or several fully broken tubes (key word GVGUIL). The physical model is different for these two cases.
The model is connected / disconnected using OPENBREK / CLOSE (location can be set up too).

5.5.6.3.3 Caution

Because it creates a hydraulic connection between a primary circuit and a secondary circuit, the same radio-chemical elements and non-condensable gases have to be defined for the two circuits.

5.5.6.3.4 Other sources of information

[DOC2] HOW TO MODEL A SG TUBE RUPTURE?
The use of this gadget is described in detail in [DOC3] (SGTR).
The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12].

5.5.7 Injection/extraction gadgets

These elements are used to define sinks or sources in an input deck.

When a source gadget (SOURCE, PIQREV) injects subcooled water in a mesh of an axial element, a specific local condensation model is used in the injection mesh, taking into account the increased mixing effects due to the jet (qualified on COSI tests).

5.5.7.1 PIQREV element

5.5.7.1.1 Description

This gadget is used to model **a leak/fluid removal or fluid injection by modelling a branch connected to the main element**. (Example: discharge /let-down or charge/make-up in a nuclear power plant)
It can be connected to an AXIAL or a VOLUME element.

In source mode (flowrate Q>0), the total flowrate is defined by the user as well as external conditions (injection conditions: external temperature or enthalpy for gas and liquid, void fraction and pressure, and if they exist in the circuit, mass fractions/concentrations of non-condensable gases or radio-chemical elements). If no temperature or enthalpy data is input, the CATHARE code uses the saturation temperature calculated at the total pressure injection point.
In sink mode (Q<0), the total flowrate extracted by the gadget is chosen by the user but limited by the sonic conditions. This gadget also allows a purge (PURGE) or an event (EVENT) to be modelled by defining a void fraction threshold below (respectively above) which only liquid (respectively gas) is extracted.
As the flowrate is defined by the user at each time step, the calculation is explicit.

5.5.7.1.2 How to proceed?

The gadget must be defined in the data block.
The PIQREV calculation is enabled / disabled using the OPEN / CLOSE key words.
It is an **external** gadget, meaning that it is controlled by the user within the executable block of the data set using CCV variables.

Remark: refer to TEE gadget figures (page 62) for more explanations of the geometrical characteristics of the PIQREV (SECT, ANGLE) and also ELEV if the source is connected to an AXIAL element.

5.5.7.1.3 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12].

5.5.7.2 SINK element

5.5.7.2.1 Description

This gadget is used to model **an offtake or a leak** or to model the extraction of fluid within the element (the user cannot choose liquid or gas extraction; this depends on the mixture, the kind of flow in the module and the position of the sink – see remark 2).
It can be connected to a VOLUME or an AXIAL element.

5.5.7.2.2 How to proceed?

It has to be defined in the data block.
The SINK calculation is enabled / disabled using the OPEN / CLOSE key words.
Extraction is managed by the user and can be internal (defined by a time-dependent law), in which case the calculation of induced thermal hydraulic terms is implicit, or external (imposed directly during the calculation), in which case the calculation of induced thermal hydraulic terms is explicit.

This element also enables a **sink with a safety valve** (SAFETYVA) to be modelled. In this case, additional data are needed to define calibration pressure (PRESSURE), pressure difference with respect to the maximum opening (DELTAP) and cross-section of the maximum opening (SECT).
This element also enables a **sink with a control valve** (VALVE) to be modelled (only available for VOLUME), in which case only the cross-section has to be defined (SECT). In both cases, the outflow is the product of the sonic flow and the cross-section of fluid aperture.
This model is simple and may be preferred to PIQSEB and PIQVANNE if the user is not trying to model actual actuators but, on the other hand, one drawback is that it gives bad results if subsonic conditions are reached.

Remarks:
1/ Extraction flowrates have to be defined with negative real numbers in the input deck. They are limited by choking conditions.
2/ In a boron-activity calculation, the extraction of boron or activity by a sink is automatic, the boron or activity concentration of the extracted phase being that of the mesh corresponding to the sink term.

5.5.7.2.3 Caution

Only extraction can be defined with this gadget (negative flowrate). For positive flowrate management (injection), the user has to add a SOURCE gadget or can use a PIQREV gadget which is able to manage injection and extraction.

5.5.7.2.4 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12]. It is also described in detail in [DOC3] (SINK).

5.5.7.3 SOURCE element

5.5.7.3.1 Description

This gadget is used to model the injection of liquid or gas or other fluid components.
It can be connected to a VOLUME or an AXIAL element.

5.5.7.3.2 How to proceed?

The gadget must be defined in the data block.
The SOURCE calculation is enabled / disabled using the OPEN / CLOSE key words.
Injection is managed by the user and can be internal (defined by a time-dependent law), in which case the calculation of induced thermal hydraulic terms is implicit, or external (imposed directly during the calculation), in which case the calculation of induced thermal hydraulic terms is explicit. If no temperature or enthalpy data is input, the CATHARE code uses the saturation temperature calculated at the total pressure injection point.

Remarks:
1/ Injection flowrates have to be defined with positive real numbers in the input deck.
2/ If fluid components other than water are injected, they should be defined in the circuit fluid components (see §5.8 and RADCHEMI and/or NONCOND).
3/ Refer to TEE gadget figures (page 62) for more explanations of geometrical characteristics of the PIQREV (SECT, ANGLE) and also ELEV if the source is connected to an AXIAL element.

5.5.7.3.3 Caution

Only an injection can be defined with this gadget (positive flowrate). For negative flowrate management (extraction), the user has to add a SINK gadget or may use a PIQREV gadget which is able to manage injection and extraction.

5.5.7.3.4 Other sources of information

The numerical and physical analyses and software implementation of this gadget are described in detail in [DOC12]. It is also described in detail in [DOC3] (SOURCE).

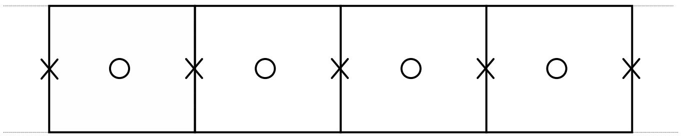
5.5.8 TEE gadget

5.5.8.1.1 Description

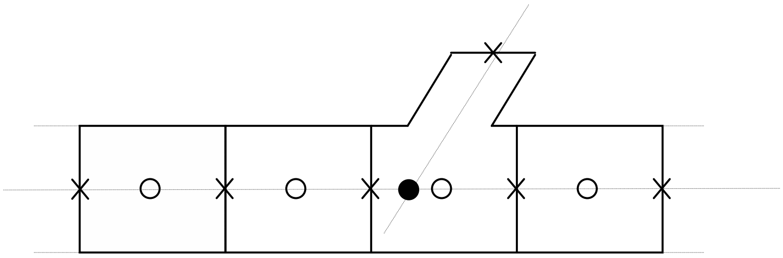
This gadget is used to model **a tee branch of an AXIAL element**.
This tee branch may be connected on the other side to an AXIAL element, a VOLUME or a THREED element. In CATHARE Team, it is only used to connect two AXIAL elements, for example in a reactor input deck for connecting the pressuriser surge line to the hot leg, or a break nozzle to a main pipe of the loop.

5.5.8.1.2 How to proceed?

When defining the TEE gadget properly, an axial element such as this:



becomes



The TEE has to be defined in the input deck (TEE).
The geometrical characteristics of the tee branch then have to be defined by the user (GEOM). Refer to Figure 29 and Figure 30 for a more intuitive explanation on the definition of key words.

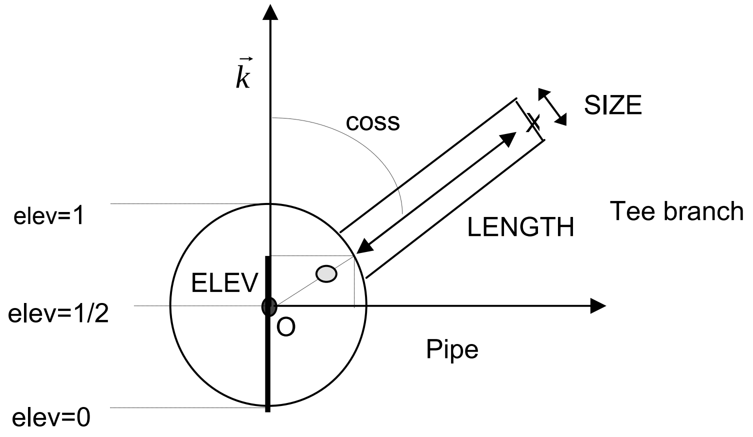


Figure 29: TEE cross-section view

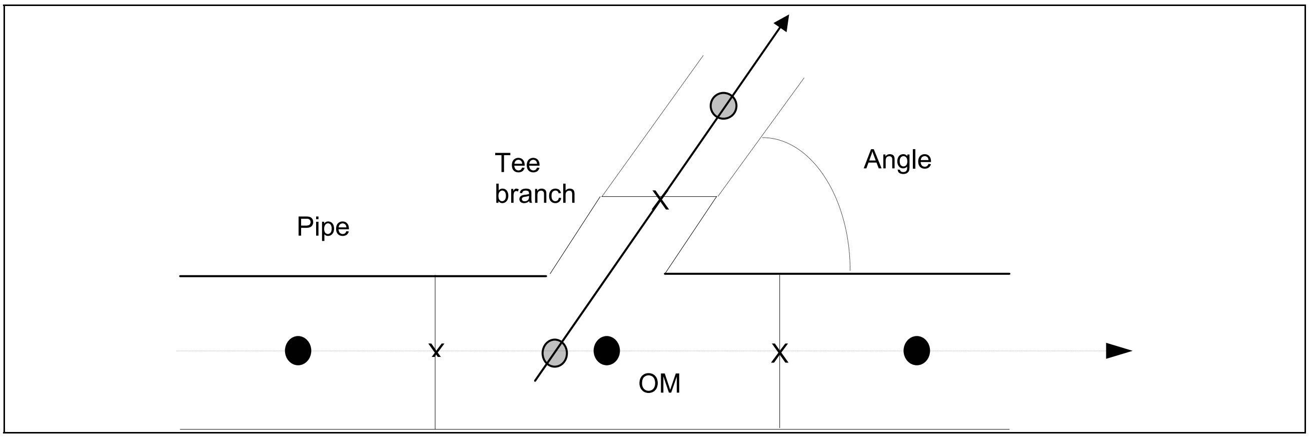


Figure 30: TEE side view

Finally, it has to be connected to the AXIAL element (CONNECT).

5.5.8.1.3 Other sources of information

This gadget is described in detail (physical and numerical aspects) in [DOC11].

Refer to Appendix 3 for an example of mesh numbering of an AXIAL with tee branches.

Refer to [DOC2] for more information:
WHY-WHEN TO USE a TEE?
HOW to define the GEOMETRY of a TEE?

The TEE sub module can be used to model JETPUMP on specific facilities and reactor. This model has to be used with caution because:

- It modifies the momentum equations in the main branch,
- It is not validated in two phase flow situations, so it has not to be used in such cases

Activation and deactivation of JETPUMP model calculation is achieved by the ENABLE and DISABLE directives.

5.5.9 Valve gadgets in CATHARE simulated flow

5.5.9.1 CHECK VALVE, CONTROL VALVE and FLOW LIMITER gadgets

5.5.9.1.1 Description

These gadgets are designed to model **valves**.
They may be connected to the vector mesh of an AXIAL element or a junction between two AXIAL elements (with or without TEE gadgets), or between a VOLUME and an AXIAL element.

A **check valve** prevents flow in one direction and the flow in the other direction depends on the pressure difference.
A **control valve** is designed to regulate the flowrate or to stop the flow.
A **flow limiter** is designed to limit the flowrate.

Remark: these gadgets are also called **VEC gadgets.**

5.5.9.1.2 How to proceed?

To use a CHECK VALVE gadget, the user has to define its location and the valve specifications: the reference fluid density (ROREF), the fully open valve capacity (CV) and the required pressure drop to obtain the full opening of the valve (DP). For convergence reasons, the DP value should be chosen greater than 10^2 Pa (10^3-10^4 are CATHARE Team recommended values).
The user may also define a fouling rate (TCVENCR) and/or a leakage rate (TCVFUITE).
The user may apply failures to this gadget during the computation (see WRITE key word). Several failure models are available.

To use a CONTROL VALVE gadget, the user has to define its location and the valve specifications: the reference fluid density (ROREF) and the law giving the valve capacity as a function of the stem position (CV). The user may also define a fouling rate (TCVENCR) and/or a leakage rate (TCVFUITE).
The user may apply failures to this gadget during the computation (see WRITE key word). Leakage and fouling failure models are available to which the user can associate a failure rate.

To use a FLOW LIMITER gadget, the user has to define its location and the reduced surface area of the flow limiter (SR). The user may optionally define the reference fluid density (ROREF) and the normalised flow-limiter capacity (CV).

These gadgets are enabled / disabled using the OPEN/CLOSE key words.

5.5.9.1.3 Caution

Selecting a ROREF value > 500 kg/m^3 implies a liquid type correlation with liquid vaporisation/condensation for CV and a critical flowrate calculation.
Selecting a ROREF value < 500 kg/m^3 implies a gas type correlation, with a single-phase, perfect gas and adiabatic transformation hypothesis for CV and a critical flowrate calculation.

5.5.9.1.4 Other sources of information

These gadgets are described in detail in [DOC17]

Refer to [DOC3] (CHECK VALVE, CONTROL VALVE, FLOW LIMITER) for more information on defining characteristics.

5.5.9.2 ECHECK and ECVALVE gadgets

5.5.9.2.1 Description

These gadgets are designed to combine two VEC objects located at the same vector point.

The ECHECK gadget is used to manage a CHECK VALVE and a CONTROL VALVE defined at the same vector point of an AXIAL element.

The ECVALVE gadget is used to manage a group of FLOW LIMITER and CONTROL VALVE gadgets defined at the same vector point of an AXIAL element. Using this gadget, the user can also represent a main line bypass on which FLOW LIMITER and/or CONTROL VALVE gadgets are defined.

5.5.9.2.2 How to proceed?

- Define each component.
- Define the group.
- The group is enabled / disabled using the OPEN/CLOSE key words. In this case, the VEC objects which belong to the group do not need to be separately activated.

Failures and position values (opening rate PU) may be applied using CCV (WRITE) on each sub-component.

5.5.9.2.3 Caution

- The grouped VEC gadgets should have the same reference fluid density (ROREF).
- VCC or failure operations: Beware of the component name used for the variable (explanation given in [DOC3]: EQ variables refer to equivalent characteristics.
- The equivalent CV of a group of two valves connected in series is lower than the lowest of the two valves.
- Valves connected in series must be opened via the isolation valve of the group (PU value).

5.5.9.2.4 Other sources of information

These gadgets are described in detail in [DOC17].

5.5.10 Valve gadgets connected outside CATHARE simulated flow

These gadgets should be used only to represent actual actuators. They are designed to model valves at the limit of CATHARE geometry, which means connecting CATHARE simulated flow to external conditions, to the external system model.

5.5.10.1 PIQSOUP gadget

5.5.10.1.1 Description

This gadget is designed to model **an actual safety valve**.
It is an extension of the SAFETY-VALVE sub-type of the SINK sub-module but allows reversible flow and has a time-implicit solution scheme.
Failures can be applied to this gadget.
It also allows a purge (PURGE) or an event (EVENT) to be modelled by defining a void fraction threshold below (respectively above) which only liquid (respectively gas) is extracted.
Reverse flow can be forbidden using NOREVERS key word.

5.5.10.1.2 How to proceed?

The component needs to be defined.
It needs to be activated to be taken into account (OPEN) and can be deactivated (CLOSE). The opening is dependent on the pressure gap between the two sides of the gadget.

5.5.10.1.3 Caution

Unless it is used with the NOREVERS option, as it allows reverse flow, the user should define external conditions before connecting it.
The inexperienced user may prefer to use the SINK gadget with the SAFETYVA option which is easier to define and manage.

5.5.10.1.4 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.5.10.2 PIQVANNE gadget

5.5.10.2.1 Description

This gadget is designed to model **a group of control-valves and check-valves connecting an** AXIAL or a VOLUME **element with the outside** which is not modelled by CATHARE (for example, a connection to other systems).

It represents a branch, connected to a scalar point of the main element, terminated with one or two valves and optionally a check valve. The calculation scheme is time implicit. It allows reversible flow but this may be changed using NOREVERS key word. In this case external conditions are not required.
This gadget also allows the modelling of a purge (PURGE) or an event (EVENT) by defining a void fraction threshold below (respectively above) which only liquid (respectively gas) is extracted.

5.5.10.2.2 How to proceed?

The component needs to be defined.
It needs to be activated to be taken into account (OPEN) and can be deactivated (CLOSE).
The opening rate is defined by the user through CCV (WRITE).
The external conditions have to be defined by the user (case of inflow).

5.5.10.2.3 Caution

The inexperienced user may prefer to use the SINK gadget with VALVE option which is easier to define and manage (but it does not allow proper pressure regulation).

5.5.10.2.4 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.6 Special features for special PWR reactor phenomena

5.6.1 FLOMIXER

5.6.1.1.1 Description

This component is designed to model the **incomplete mixing phenomena in the vessel lower head** (case of Vessel 0D+1D modelling).
It activates an extra relationship between the liquid enthalpies calculated at several junctions of two different VOLUME elements.

This model should be activated only when the fluid primary circuit is a single-phase liquid (except for the pressuriser) with active primary pumps and when vessel inflow rates are slightly dissymmetric. Then the user has to define the maximum permissible void fraction (at vessel inlet junctions) (AGMAX) and the maximum permissible dissymmetric coefficient DQMAX (= Max(Q_i – Q_j) / Min Q_i where Qi and Qj are vessel inflow rates).
The model equations are as follows:
Q_{in-j} is an inflow rate (junction of the first volume), Q_{out-i} an outflow rate (from the secondary volume).

$$Weight_i = \sum_j a_{ij}\, Q_{in-j}\,, \quad i = 1, N$$ where

$$H_{out-i} = \sum_j a_{ij}\, Q_{in-j} H_{in-j} / Weight_i + \Delta H_{vessel}\,, \quad i = 1, N$$ (explicitly imposed)

$\sum_i Q_{out-i}\, H_{out-i} = Q_{out\,tot}\, H_{VOLUMEout}$ where $Q_{out-tot} = \sum_i Q_{out-i}$ and $H_{volume-out}$ is the modelling calculated liquid enthalpy in the secondary volume (energy conservation)
The a_{ij} coefficients may be defined in a personal library (N4 or CP1 key word) or must be given by the user for the first outlet junction (the built matrix is cyclical) (USER key word).

In CATHARE this becomes: at each time step, the liquid enthalpy for the outlet junctions is imposed (instead of standard liquid energy equation).

5.6.1.1.2 How to proceed?

Users must complete the definition of the model (specification of AGMAX and DQMAX criteria).
The model needs to be activated to be taken into account (STARMIX) and can be deactivated (STOPMIX) by the user. Nevertheless, the model deactivates automatically when the criteria are no longer fulfilled.

5.6.1.1.3 Caution

According to experimental data, the volumetric flowrate distribution is not modified. This only modifies the enthalpy flowrate distribution between the ports and not the overall enthalpy outflow rate.

5.6.2 PIQARE

This component is designed to model the **overflow phenomena in a N4 plant serial steam generator** (the Feedwater Flow Control System overflow).

The overflow model is relevant for a SG equipped with an economiser type feedwater system (i.e. the downcomer is divided into two separated legs and the MFW injection devices are designed to send the main feedwater into the cold leg). The aim of the PIQARE object is to take into account the imperfect operation of the MFW injection devices leading to MFW overflow (i.e. MFW spilling over into the SG hot leg). The model has been derived for a type 73/19 SG (N4 type French nuclear power plant).

This model can then be used if the primary and secondary sides of the steam generator are defined with AXIAL elements. In fact, using this gadget, the code creates two PIQREV objects each of which is connected to one AXIAL element.

5.6.2.1.1 How to proceed?

The component needs to be defined. Use SECASP and CONST to tune the overflow rate. Use RATINI to define the overflow rate during initialisation (see §6.3.2.2 for more information on initialising CATHARE).
This model is enabled / disabled using the OPEN/CLOSE key words.
It should be activated before the initialisation step.
The overflow calculation is automatically activated when the module is connected but it can be stopped and restarted using STOPOVRF / STAROVRF.

5.6.2.1.2 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.6.3 PIQSEB gadget

5.6.3.1.1 Description

This gadget is designed to model **a PWR SEBIM safety valve**.

The PIQSEB gadget is a special kind of valve used in a nuclear power plant. The main difference with the PIQSOUP safety valve is that the opening and closing pressures are different. This gadget takes into account the real operation of the SEBIM type safety valve, modelling the behaviour of the pilot valves and managing the opening and closing sequences.
It consists either of a single safety valve or of a group of two safety valves.

5.6.3.1.2 How to proceed?

A PIQSEB gadget should be defined at a scalar point.
The external conditions have to be defined by the user through CCV (WRITE) (void fraction, temperature, pressure).
This model is enabled / disabled using the OPEN/CLOSE key words.
Failures may be applied to each sub-component.

5.6.3.1.3 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.6.4 SINKRRI gadget

5.6.4.1.1 Description

The SINKRRI gadget is a special sink designed to model the safeguard systems in recycling conditions. By these systems, water is drawn out of the bottom of the containment, cooled through exchangers and re-injected into the containment spray system or into the safety injection system.
A SINKRRI must be defined on a VOLUME element. This volume should be located at the bottom of the containment. It is a normal sink but also includes a description of the safeguard systems.

5.6.4.1.2 How to proceed?

This gadget has to be defined (SINKRRI).
It is enabled / disabled using the OPEN/CLOSE key words.

5.6.4.1.3 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.6.5 SOURIS gadget

5.6.5.1.1 Description

The SOURIS gadget is a special SOURCE designed to model the phase distribution of a break flowrate into the containment, as a function of the safety injection conditions and of the core residual power (relevant only for containment calculations).
A SOURIS must be defined on a VOLUME element. This volume should be located at the top of the containment. It is a common source which includes in addition a description of the “RIS” model.

5.6.5.1.2 How to proceed?

This gadget has to be defined (SOURIS) and needs to be activated to be taken into account (OPEN) and can be deactivated (CLOSE).

5.6.5.1.3 Other sources of information

This gadget is described in detail (physical, numerical and software implementation aspects) in [DOC12].

5.7 Hydraulic link between elements

5.7.1.1.1 Description

The EXHYLINK module is designed to model **an explicit hydraulic link** between two hydraulic objects (element or gadget) that can belong to the same circuit, to two different circuits of a reactor or to two different reactors.

It can be defined between a boundary condition (BCONDIT) or a RUPTURE element and a PIQREV or a SOURCE gadget, or between two injection/extraction gadgets (PIQBREK/SINK/PIQREV/SOURCE).

One of the main aims of this link is to simulate a rupture between the main system of a plant and the reactor containment.

5.7.1.1.2 How to proceed?

The two modules to be connected first have to be defined. Then the link is defined.
The gadget has to be connected (OPEN) before the link is activated (SWITCH ON) (the only exception is the case of a link between a RUPTURE and a PIQREV where the opening of the PIQREV is automatic). The link can be deactivated during the calculation (SWITCH OFF).

5.7.1.1.3 Caution

- When using injection or extraction gadgets with no reversal flow allowed (SINK, SOURCE), the user should be sure that no reversal flow will occur (this will provoke a calculation error).
- The user has to ensure the compatibility of the information exchanged: when the link is activated, the SOURCE energy is given by the linked element as enthalpies. Then, ITLSOU and ITGSOU flags have to be forced to 1 using CCV.
- Be careful not to disconnect a gadget without first having switched off the link.

- As the user is connecting two hydraulic circuits, the same non condensable gases, radio or chemical components have to be defined in both circuits.

5.7.1.1.4 Other sources of information

Refer to [DOC3] (EXHYLINK) for more information on defining characteristics.

5.8 Non condensable gases, Fluids, radio-chemical elements

Besides the main fluid used for PWR or BWR reactors, (steam and liquid water), CATHARE can be used to define other main fluids or other elements in the circuit fluid. This chapter refers to the description of other main fluid definition, transport of non-condensable gases and boron and radio-chemical elements. It is defined for a particular circuit. The transfer between circuits is possible only through a Steam Generator Tube Rupture (SGTR) (see §5.5.6).

5.8.1 Non-condensable gases (NONCOND)

5.8.1.1.1 Description

Transport of one to four non-condensable gases by a circuit can be modelled by CATHARE. These gases are assumed to be in thermal and mechanical equilibrium with steam.

A transport equation is solved for each non-condensable gas. The mass fraction is a principal variable of the system. The equations are implicitly coupled to the hydraulic equations and the six-equation model becomes a 7, 8, 9 or 10-equation model. Non-condensable gas concentration has an impact on the standard calculation because it changes:

- the vapour pressure, which becomes a partial pressure with respect to the gas phase total pressure,
- the condensation correlations (condensation is reduced in the presence of non-condensable gases).

5.8.1.1.2 How to proceed?

The user needs to use the NONCOND key word to define the list of non-condensable gases to use in the calculation.

The user may:

- Use the six non-condensable gases predefined in the CATHARE Code, i.e., NITROGEN, HYDROGEN, HELIUM, OXYGEN, AIR and ARGON. The properties used by the code in the calculation are then predefined by CATHARE: refer to [DOC18] for a complete description of the constitutive relationships of CATHARE non-condensable gases.
- Or define himself the properties of the non-condensable gases. The properties used by the code in calculation should then be given by the user (see [DOC3] NONCOND).

The value of transport properties used during calculation depends on the mixture model chosen by user. By default a molar mass averaged standard model is used for the gas mixture viscosity and conductivity computation, but specific models can be used (see [DOC6].

The NONCOND object name has to be added to the list of the circuit elements (CIRCUIT).

Initial concentrations also have to be given for initialisation step (see §6.3.2.2).

Remark: if the void fraction is almost 1 (liquid residual phase) define TL as $T_{SAT}(P_V)$. If the void fraction is almost 0 (gas residual phase), define TG as $T_{SAT}(P_V)$.

External concentrations also have to be defined by time-dependent laws or by CCV, from boundary conditions (MODEL or BCMOD) or from injection systems (injection gadgets or reversible valves connecting the flow with the outside).

By default, the injection concentrations are set to 0. The gases can be injected in the circuit by the user through, for example, a source, an accumulator or a boundary condition (see ACCU, SOURCE, PIQREV, BCONDIT for more explanations).

Remark: if HYDROGEN has been declared as a non-condensable gas in NONCOND and if there are fuel rods in the circuit, there will be a source term of H_2 and a sink term of H_20 in the AXIAL or THREED element which contains the fuel rods, because of oxidation of the fuel rod cladding (see §5.4.3.1).

5.8.1.1.3 Caution

Except for two circuits implicitly coupled through heat exchangers (the two circuits are considered as one for the solver and they should then have the same number of equations for each element) or two circuits connected with a steam generator tube rupture (SGTR) (in which case it is assumed that the SGTR will be activated and then the fluids of the two circuits will mix), each circuit can carry a different number of non-condensable gases.

NB: in the calculations, care must to be taken with mass fractions and enthalpy while using non-condensable gases.

5.8.1.1.4 Other sources of information

Refer to [DOC2] HOW to CALCULATE using NON-CONDENSABLES GASES?

5.8.2 Radioactive / Radio-chemical elements (RADCHEMI)

5.8.2.1 Main features

5.8.2.1.1 Description

This module is used to follow boron concentration and/or fission components in the primary circuit.

CATHARE includes a mass-continuity equation for a solute moving with the liquid field, for a gas moving with the gas phase and also allows dissolution and evaporation of the component. In case of radioactive products, the decay period is taken into account to calculate the activity.
If the boron mass fraction in a cell is greater than the solubility threshold, boron deposition occurs. If more liquid enters the cell, then two options are available: either any deposited solute instantaneously re-dissolves to maximum concentration, or there is no re-dissolution; the solute is definitively deposited.
For radioactive elements CATHARE calculates an activity and for other chemical components, it calculates a mass fraction.

Time discretisation is at the same level as the hydraulics and ensures a stable solution (implicit in 0D and 1D modules, semi-implicit for 3D module). Transport equations are solved at the end of each time step, after convergence of the hydraulic variables.

The maximum number of radio-chemical components is 12. The calculation is optional.

5.8.2.1.2 How to proceed?

Acquisition of radio-chemical component characteristics must be specified for each circuit using the RADCHEMI key word and then associated with the circuit (CIRCUIT). However, they should be the same in the case of a SGTR definition.

There are 13 radio-chemical components, the characteristics of which are predefined in CATHARE, and it is also possible to enter new user-defined components. For these new components, the user has to provide all characteristic data: chemical or radioactive type, chemical constants for the two-phase treatment, emergency shut down behaviour (see §5.8.2.2).
These properties are common to all the elements of the circuit. Some of them may be changed, in particular elements via WRITE/WRIBA/WCIRCBA (see these directives) or INIBORA.

The element concentration and activity fields have to be initialised (INIBORA). In particular, the element zone used to define the mean core activity for the release peak in case of emergency shut down activation (ZONBAMOY) can be defined. The default zone chosen by the code for this calculation is the entire circuit (including dead zones if they exist).

The calculation has to be launched using the GOBORA key word.

The user can define external injected concentration/activity (ACCU, SOURCE, PIQ***, BCONDIT). The default value is zero concentration for each radio-chemical element.

Some data constants can be modified in certain elements to take into account particular conditions using the INIBORA and/or WRIBA key words.

After using GOBORA, the available user management consists in:

- activating emergency shut down (using the ACTEMIS or SCRAM key word or using CCV (WRIBA key word with FLAG_IEA variable),
- modifying the two-phase treatment constants or the emergency shutdown behaviour using CCV (WRIBA),
- reinitialising activity/concentrations using the INIBORA key word. In this case, the balance calculation should also be reinitialised (INIBIL).

5.8.2.1.3 Caution

If a SGTR gadget has been defined between two circuits, the same radio-chemical components have to be defined in both primary and secondary circuits.

5.8.2.2 How to specify user-defined radio-chemical elements?

A distinction is made between radioactive components and non-radioactive (chemical) elements.

5.8.2.2.1 Radioactive elements

5.8.2.2.1.1 Constants used by all the elements

The type of component is defined by the “ira” flag (ira = 2: gaseous component; ira = 1: non-gaseous component). The following properties of predefined radioactive components are given in the table below: “vie” (half-life), “ka” (weight of one GBq), “henry” (HENRY constant of the gas) and “effi” (efficiency of the Chemical and Volume Control filters). Those four constants have to be specified for every user-defined component.

	Constant	ira	half-life	ka (kg/GBq)	henry (in MPa)	effi
	corresponding CCV	IRA_NB	HFLIFE	KA_CST	HENRY_	RCVEFF
Component	**key-word**					
Krypton 88 (FP)	KRYPTO88	2	78 minutes	9.5 D-13	1.55 D4	1
Xenon 133 (FP)	XENON133	2	5.3 days	1.5 D-10	1.04 D4	1
Iodine 131 (FP)	IODE131	1	8 days	2.2 D-10	1	100
Cesium 138 (FP)	CESIU138	1	30 years	3.1 D-7	1	10
Nitrogen 16 (activation product of water)	AZOTE16	2	7.35 seconds	2.82 D-16	1.55 D4	1. D10

5.8.2.2.1.2 Data used by the element carrying the fuel wall

Radioactive components may be fission products (FP), activation products or radiolysis products. Different constants are to be specified depending on the case.

5.8.2.2.1.3 Activity release in case of clad rupture:

For fission products, the time-dependent change in amount of activity in the core (in GBq) is given by the decay law "loid" as a function of elapsed time after Emergency Shutdown: this deals specifically with the activity contained in the gap between the UO_2 pellets and the fuel rod cladding (modelling hypothesis).

The activity functions used for predefined components are listed below (in GBq):

time (second)	KRYPTON 88	XENON 133	IODINE 131	CESIUM 138
600	0.21 D8	0.87 D8	0.215 D8	0.145 D7
3600	1.32 D7	0.87 D8	0.215 D8	0.145 D7
18000	1.49 D6	0.87 D8	0.215 D8	0.145 D7
86400	0.47 D2	0.84 D8	0.195 D8	0.145 D7
345600	0.	0.61 D8	0.150 D8	0.145 D7
691200	0.	0.36 D8	0.105 D8	0.135 D7

Remark 1: this model was tested for a reactor (one or two circuits) including only one element with a fuel wall. The above data are only used by this element.
Remark 2: the activity of radiolysis products and activation products in the pellet-cladding gap is always equal to zero.

The above data have to be specified for every user-defined component.

5.8.2.2.1.4 Modelling of the Emergency Shutdown activity peak:

Fission products:

At the Emergency Shutdown, a fission product activity peak occurs, which is initialised with two parameters:

- the release rate before Emergency Shutdown (relach).
- the amplification factor of this release rate after Emergency Shutdown (mult).

The IMR flag is equal to zero.

The activity decay after the peak (ratio) can be initiated with two criteria, depending on the IDP flag:

- If IDP = 0, the decay is initiated when the ratio between the primary mean specific activity and its value before Emergency Shutdown, exceeds DACT (which is an input variable of the model).
- If IDP = 1, the decay is initiated when time t is greater than $t_{\text{Emergency Shutdown}}$ + DT_{pic} (DT_{pic} is an input variable of the model).

Radiolysis or activation products:

As there is no activity peak at the Emergency Shutdown for these components, their production decreases in proportion to the neutron flux decay law. IMR is then equal to 1, and the "relach" value represents the proportionality coefficient (instead of the activity release rate). The IDP flag is set to 1, but DT_{pic} (set to 1) is not used, as there is no activity peak. The amplification factor (also set to 1) is not used either.

Thus the relevant constants are: "IMR", "relach" (release rate or proportionality coefficient with respect to neutron flux), "mult" (release rate amplification factor), "IDP" and "ratio" (DACT or DT_{pic}).
Those constants have to be specified for every user-defined component. For predefined components, their values are given in the table below:

Constant	IMR	relach	mult	IDP	ratio
Corresponding CCV	IMR_NB	RELACH	AMPLIF	IDP_NB	RATIO_
Component					
Krypton 88 (FP)	0	1.646 D-1	3.287	0	2.259
Xenon 133 (FP)	0	2.382 D-2	111.3	0	1.898
Iodine 131 (FP)	0	3.359 D-3	479.9	0	15.701
Cesium 138 (FP)	0	1.769 D-5	1.936 D4	0	449.3
Nitrogen 16 (activation product of water)	1	8. D4	1.	1	1

Remark 1: algorithmically, only one “relach” constant is assigned:

- to the release rate (in GBq/s) before Emergency Shutdown, used for fission products (IMR flag equal to 0),
- to the proportionality factor between the activity release rate and the neutron flux, used for radiolysis or activation products (IMR flag equal to 1).

Remark 2: algorithmically, only one “ratio” constant is assigned:

- to the ratio between the value of the primary mean specific activity expected at the end of the peak and the value before Emergency Shutdown, used when IDP = 0,
- to the peak duration, used when IDP = 1.

Remark 3: the model was tested for a reactor (one or two circuits) including only one element with a fuel wall. The above data are only used by this element.

5.8.2.2.2 Non-radioactive (chemical) components

- **Data used by all the elements**

The type of component is defined by the “ira” flag (ira = 3: gaseous component; ira = 0: non-gaseous component). The “ka” constant is always equal to 1. Although the “vie” and “effi” parameters are not used, values should be assigned to them. The “henry” parameter is only used for gaseous components.
The relevant constants of predefined chemical components are given in the table hereafter:

	Constant	ira	vie	ka	henry (in MPa)	effi
	Corresponding CCV	IRA_NB	HFLIFE	KA_CST	HENRY_	RCVEFF
Component	**key word**					
Boron	BORON	0	-	1	-	1.
H+ ions	HPLUS	0	-	1	-	1.
Conductivity	CONDUCTI	0	-	1	-	1.
Sodium Na	SODIUMNA	0	-	1	-	1.
Oxygen O2	OXYGENO2	3	-	1	1.22 D4	1.
Hydrazine N2H4	HYDRAZIN	3	-	1	1.55 D4	1.
Nitrogen N2	AZOTEN2	3	-	1	2.56 D4	1.

Those five constants have to be specified for every user-defined component.

- **Constants used by the element carrying the fuel wall**

5.8.2.2.2.1 Activity release in case of clad rupture:

5.8.2.2.2.2 The activity in the pellet-cladding gap related to chemical components is equal to zero at any time.

5.8.2.2.2.3 Modelling of the Emergency Shutdown activity peak:

No computation of the Emergency Shutdown peak is made, IMR = 0 and IDP = 1 (“relach” = 0, “mult” = 1 and “ratio” = 1) for any chemical component, whether it is predefined or not.

Component	IMR	relach	mult	IDP	ratio
any chemical component	0	0	1	1	1

5.8.2.2.3 Constants with the same meaning for chemical and radioactive components

Those constants are used by all the elements of the reactor (one or two circuits).

The following constants: “ke” (entrainment coefficient through vaporisation): “kc” (entrainment coefficient through condensation), “dilu” (gas into liquid dissolution time constant) and “dega” (gaz stripping time constant), are listed in the table below, for the 13 predefined components.

Constant	ke	kc	dilu	dega
Corresponding CCV	EVAPOR	CONDEN	DISSOL	DEGASA
Component				
Krypton 88	0.1	0.01	5000. D3	5000. D3
Xenon 133	0.1	0.01	5000. D3	5000. D3
Iodine 131	0.1	0.01	1	1
Cesium 138	0.1	0.01	1	1
Nitrogen 16 (activation product of water)	0.1	0.01	5000. D3	5000. D3
Cobalt 60 / Manganese	0.1	0.01	1	1
Boron	0.1	0.01	1	1
H+ ions	0.1	0.01	1	1
Conductivity	0.1	0.01	1	1
Sodium Na	0.1	0.01	1	1
Oxygen O2	0.1	0.01	5000. D3	5000. D3
Hydrazine N2H4	0.1	0.01	5000. D3	5000. D3
Nitrogen N2	0.1	0.01	5000. D3	5000. D3

Those four constants have to be specified for every user-defined component.

For any component, a “loim” law gives the maximum value of the concentration in liquid phase as a function of liquid temperature.
This law must be specified for any user-defined component. The laws for predefined components are:

component	temperature	max concentration (kg/kg)
Boron	100 °C 300 °C	6.35 D-2 27.5 D-2
Other components	0°C 500 °C	1. 1.

5.8.3 Main Fluids (FLUID)

5.8.3.1.1 Description

The code is based on a water 2-fluid 6-equation model including non-condensable gases equations and additional equations for radio-chemical components transport. It allows the coolant circuits of any water reactor to be represented by assembling 1D, 0D and 3D modules.

In the present and future version, new main two-fluid models are and will be available.

5.8.3.1.2 How to proceed?

The main fluid is defined with the FLUID operator and is attached to the reactor during the construction of the circuit (see CIRCUIT operator). The default main fluid is the water. At the present time, a modified water fluid is available and is used for gas cooled reactor calculations. In the future, SODIUM, OXYGEN and HYDROGEN will be implemented in the CATHARE code.

5.8.3.1.3 Caution

In case of 2 circuits, if a SGTR operator is connected during the calculation or if the circuits are implicitly coupled, the FLUID operator has to be the same for both circuits. In other cases, each circuit can deal with a different FLUID.
The acquisition of FLUIDs can be done only one time for two circuits if they are the same but they will have to be declared in the two circuit operators.

5.9 Other facilities

5.9.1 Gas single phase calculation (MONOPHAS)

5.9.1.1.1 Description

The MONOPHAS(E) directive may be used to simulate a single gas-phase flow computation. This functionality is mainly used for gas cooled reactor calculation but can be also used for water or other reactor transients when liquid phase is not present.

5.9.1.1.2 How to proceed?

The MONOPHAS(E) directive has to be used in the *command block*. The default setting is OFF, meaning standard two-phase flow computation. When the keyword GAZ is applied to a circuit, the liquid residual phase of its elements is set in thermal and dynamic equilibrium with the gas phase, i.e. saturation temperature at vapour pressure, gas velocity. The void fraction is set to its maximum value $1.-10^{-6}$ and the volume levels are set to their residual value 10^{-3}. No more liquid mass, energy and momentum balances are computed for the corresponding elements.

The option GAZ may either be activated before any PERMINIT directive (and associated ones as REALC, REALAX…) for use in GOPERM, or anywhere else during the computation after GOPERM.

The MONOPHAS(E) directive can be used several times for a same element or for different elements during a computation. This, for instance, allows the user to first initialize a circuit CIRC1 under gas condition (MONOPHASE circ1 GAZ), then perform a stabilized transient under the same conditions, then re-activate the two phase flow computation (MONOPHASE circ1 OFF) in order to take into enable water injection again.

5.9.1.1.3 Caution

When the option is activated, no water mass and/or energy and impulsion injections are allowed by SOURCE, PIQREV, PIQVANNE, PIQSOUP, PIQSEB, ACCU, BC3x, BC4y or BC5yy or CANDLEs. The use of REFLOOD or CCFL models are then not allowed.

5.9.2 Specific components (COMPONENT)

5.9.2.1.1 Description

The COMPONEN(T) operator enables the user to define technical data describing specific hydraulic and thermal components like heat exchangers. These data will be used to calculate wall friction and/or forced convection exchange coefficients with specific correlations, which are not those used by the standard revision 6.1 of CATHARE.

5.9.2.1.2 How to proceed?

The COMPONEN(T) operator has to be used in the used in *data block* only.
The COMPONEN(T) must be ASSIGNed to an AXIAL module to be taken into account (see ASSIGN directive for AXIAL modules) specifying which law should be used.

5.9.2.1.3 Caution

There are as many syntax as there are different types of components. The hydraulic (of AXIAL type) and its thermal elements (WALL or Heat EXCHANGER) must have been previously defined in the *data block* of the input deck. The hydraulic data (flow section, hydraulic diameter) must be coherent with the geometry set in the COMPONEN**(T)**.

6 DATA FILE SPECIFICATIONS

The previous chapter presented the different elements which may be included in a reactor and their functionalities.

This chapter shows how to build a CATHARE input deck and more exactly its two parts:

- the data block part, which deals with the spatial reactor description,
- the command block (or exec block) part, which deals with the simulated scenario (steady state and time transient).

To be understood by CATHARE pre-processing (definition is given in §4.2), data acquisition has to use a specific language, which is described in the first part of this chapter. As well as syntax rules, the user will find here a summary of the CATHARE commands. The syntax of each of them is developed in the dictionary ([DOC3]).

6.1 CATHARE language

CATHARE user language is based on thermal hydraulic objects. Reading the dictionary, the user will discover the notions of **objects**, **operators** **and** **directives.**

This section defines these notions.

The next section will be devoted to developing the **syntax rules** that the user must respect in the input data deck.

The **structure of an input deck** will then be presented.

A summary of the **operators and directives** is presented (a summary is available in Appendix 14).

A review of the more common errors is also given.

6.1.1 Definition of objects, operands, operators and directives

- An object is a structure of information with a type and a name.

- An object is created by an operator.

- Operands are the objects needed by an operator to perform an operation.

Example:

x = a + b;

+ *is the operator to add two numbers*
a b *are two operands of the real number type*
x *is the result. A new object is created with the name x and the real number type.*

- A directive is a command, which is used to modify an object.

Examples

circprim = CIRCUIT elem1 elem2 elem3;

CIRCUIT is the operator used to define a new circuit. The word “circprim” is the name of the circuit and elem1, elem2, elem3 are the names of the elements included in the circuit. The operands elem1, elem2 and elem3 are not modified. A new object circprim is created.

MESH elem1 mesh1;

The mesh, mesh1, is assigned to the element, elem1. The object, elem1, is modified by the directive.

*The only condition to run an operator or a directive is that the operands **exist** when they are used.
For instance, in the previous instruction:*

MESH elem1 mesh1;

If no mesh type object with the name mesh1 exists, the directive cannot run, and an error message is sent to the user.

6.1.2 Syntax rules

- Only the first 80 characters of a line are taken into account. If a word ends on 80th column and the following word begins on the 1st column, only one word will be read instead of two, because there is no blank character read.

- When an asterisk * is in the first column of a line, the line is ignored. It can be useful to write comments in the input deck.

- A semi colon ";" ends an instruction. If it is omitted the user will receive an error message or CATHARE will jump to the next instruction.

- One or several operators take place in an instruction. An instruction can be written on several lines. Comment lines can exist in the middle of an instruction.

An instruction is read line after line and, in each line, from left to right.
Example:

The following line is filled with four instructions:
P1 = XAXIS 0.; P2 = XAXIS 0.82; P3 = XAXIS 4.12; P4 = XAXIS 5.26;

In the next line, the instruction links 3 SEGMENT operators:
Idownco = P1 SEGMENT 2 P2 COS – 1.D0 SEGMENT 4 P3 COS – 1.D0
SEGMENT 2 P4 COS – 1.D0;

In the example, first the operation P1 SEGMENT 2 P2 COS – 1.D0 is carried out. The result is an object of the 'mesh' type. Then, the second operation SEGMENT 4 P3 COS – 1.D0 is carried out, etc. The final result is the mesh object with the name Idownco.

- The five characters: **blank** ,; , **=** , **(** , **)** are separators.

Consequence: arithmetic operators (+ , -, etc.) must be separated by blanks.

- There is no hierarchy between operators. Operations are performed from left to right; parentheses must be used to handle the hierarchy of arithmetic expressions.

- When an instruction contains parentheses, the operation inside the parentheses is executed first.

Example:

IF ((A EQ B) AND (C < D));

- The language gives a type to each chain of characters it meets on a line.

Example: 50 is an **integer**, 50 or 5.D1 a **double precision real** number, 'fifty' is an **8-character word**, true and false are **logical**.

- When the reader cannot interpret a chain of characters as a number, it searches whether this chain is the name of a typed object already defined by the user. In this case it gives a type and a pointer, otherwise it gives the type character*8.

- A name cannot have more than 8 characters. Only the first 8 characters of an operator or a directive are taken into account by the code.

Remark: In this manual, the key words are printed in upper case letters and the names are printed in lower case letters to make it easier to read the examples. Of course key words and names in the input data decks are in upper case letters.

6.1.3 User advice in the event of a READER error message

Refer to §4.2 to understand what the reader is.

During reader execution, CATHARE lists each instruction before interpreting it. Then what has been read appears in the output listing.
If the reader finds an error in a command, it stops the processing and displays a message explaining the nature of the error.
During reader execution, this can happen inside the command or at the end of reader execution (if the program has to know the relation between several objects to recognise the error). It is generally linked to a syntax error.

Common syntax errors

- Omission of “;” at the end of an instruction
- Omission of “” in the title, or a graphic title
- Wrong name for an operator
- For arithmetic operations as for other operators, bad priority between operations
- Use of the same name for two objects
- Lines with more than 80 characters,
- Use of a non-existent object (spelling mistake)
- Use of a key word as a name of an object
- Names with more than 8 characters
- “*” (for a comment line) not inserted in the first column
- Automatic tabulation character after a carriage return (generated by some text editors)
- Same name used for two BLOCKS (command REPEAT)
- Forgetting non-condensable gas mass fraction in sub-module interaction laws.

NB: the syntax error such as the **omission of “;”** or **several objects with the same name** could generate a meaningless message because CATHARE always tries to interpret a general instruction (from one “;” to the following).

6.1.4 CATHARE operators and directives

6.1.4.1 For input deck basic facilities

In the following, operators are printed in bold characters and directives in normal characters. These are commands both for data and command block.

6.1.4.1.1 Assigning a value to real numbers, integers, logical operators

These operators have no name. The language assigns itself the value and the type.

A = 50;	A is an integer
B = 50;	B is a real number

C = true;   C, D are logical operators
D = false;

6.1.4.1.2 Arithmetic operators

+	:	addition
-	:	subtraction
*	:	multiplication
/		division
**	:	power
EXP	:	exponential
ABSOLUTE	:	gives the absolute value of a number
MIN, MAX	:	minimum, maximum of two values
INTERP	:	interpolation

6.1.4.1.3 Trigonometric operators

SINUS	(Sin)
COSINUS	(Cos)
TANGENT	(Tan)
ARCSINUS	(Arcsin)
ARCCOSIN	(Arccos)
ARCTANGT	(Arctan)

6.1.4.1.4 Logical operators (result = TRUE or FALSE)

EQ	:	equality
NEQ	:	different
>	:	strictly greater than
>EQ	:	greater than or equal to
<	:	less than
<EQ	:	less than or equal to
AND, OR	:	operations on two logical expressions

6.1.4.1.5 Test and loop directives

IF, ELSE, ENDIF	:	conditional execution
REPEAT, QUIT, END	:	execution of loops

6.1.4.1.6 Variable definition

Data block operators:

SEGMENT XAXIS	:	Definition of mesh
REALLIST	:	List of real numbers
LAW	:	Assembly of two REALLIST objects

Command block directives:

CHAR8, INTEGER, DOUBLE	:	Definition of local 8 character chains, integers and double precision real variables

Common operators:

SCALAR, VECTOR	:	Scalar and vector point location in the mesh

6.1.4.1.7 General input deck organisation directives

Data block:

BEGIN DATA	:	start of data block
END DATA	:	end of data block

Command block:

END EXEC	:	end of computation

6.1.4.2 For general management

6.1.4.2.1 Computation management

OPTION, VERBOSE, RESETIME, RESTORE, SAVE, INISIM, MANAGE, FASTSIZE	:	Computation options
INISIM, MANAGE, OPTION	:	Simulator use
OPTION, PERF, PESEE, PRINSTRU	:	Parallel calculation
RESULT	:	Prepare post-processing
SAVE, RESTORE	:	Save / restore plant state for later use

Refer to §6.3.1 for more information.

6.1.4.2.2 Calculation management

TRANSIENT	:	Achieving a time step
NEWDT	:	Advised next time step
NEWTIME	:	Physical time
GODIFF, GOTURB, GOBORA, GOFUEL, GONEUT, GOPERM	:	Launching specific calculation
OPEN/CLOSE, ENABLE/DISABLE, OPENBREK	:	Connecting / disconnecting gadgets
PERMINIT, REINIT, RESETIME, INIBIL, INIBORA, **ZONBAMOY**	:	Initialisation
PERMINIT, GOPERM, REALC, REALVO, REALAX, IMPOSFLOW, NOFLOW, LEVEL, ECHPOWER, TPER3D	:	Steady state initialisation
REINIT, REALVO, REALAX	:	Deadzone reinitialisation
VALUE, VALUFEAU, VALBA, WRITE, WRIBA	:	Obtaining / imposing CATHARE Calculation Variables (CCV)

6.1.4.2.3 Printout management

LIST, VERBOSE (for threed)	:	Definition (topology, mesh, geometry) information
PERIOD, IMPRIME, PRIN3D	:	Physical state information
VERBOSE	:	Debugging information (physical increments, etc.)
MESSAGE, TITLE	:	User messages
BILAN3D, **ZONE**, INIBIL	:	Managing balances
VALUE, **SENSOR**, UTILx, VALUFEAU, VALBA	:	Obtaining calculation variables
PICTG	:	Special output for cladding temperatures
OPENFILE, READHEAD, READVAR, REWIND, WRITHEAD, WRITVAR	:	User's file management

Refer to §6.3.1.4 for more information.

6.1.4.3 For reactor definition

They are defined in the data block. Refer to chapter 5 for more information on the corresponding objects.

CIRCUIT, REACTOR, ZONE	: Modules assembly
AXIAL, VOLUME, THREED, BCONDIT, RUPTURE	: Modules
WALL, WALL3D, FUEL, FUEL3D, FUELCHAR, FUELPLAQ, EXCHANGER	: Thermal sub-modules
CORE, SGCARACT, REFLCHAR, REFLCH3D	: Other thermal sub-modules
TEE, ACCU, CCFL, PUMPCHAR, SGTR, TURBINE, CANDLE, SENSOR, TCOMCHAR **ECHECK, ECVALVE, FLOW-LIMITER, CHECK VALVE,** **CONTROL VALVE, PIQSOUP, PIQVANNE,** **SINK, SOURCE, PIQREV, BREAK, PIQBREK**	: Gadgets
RADCHEMI, NONCOND	: Chemical components
PIQARE, FLOMIXER, PIQSEB, SINKRRI, SOURIS	: Special PWR components
EXHYLINK	: Explicit hydraulic link

6.1.4.4 For module definition and management

Data block directives are used to complete the module definition. In command block, besides directives, many CATHARE Computational Variables (CCV) are available to check and/or change the physical or functional variables of the modules. Refer to [DOC3] (WRITE and VALUE directives).

6.1.4.4.1 Main modules

AXIAL	: Pipe module
GEOM, HYDR, MESH, SINGULAR	: Data block directives
VALVE	: Basic valve modelling
VOLUME,	: Volume module
GEOM, LEVEL, SINGULAR	: Data block directives
Specific models for volumes:	
FLOMIXER, STARMIX, STOPMIX	: Vessel loop mixing
VFILM, STARTVC, STOPVC	: Film condensation
THREED	3D module
CONNECT, GEOM, HYDR, MESH, PHYSCALE, SINGULAR	: Data block directives
Specific models for 3D:	
TPER3D	: Steady state time limit
GODIFF	: Diffusion
TURBULEN, GOTURB	: Turbulent diffusion
DTMASS3D	Step time management from 3D mass error
BCONDIT	: Boundary condition
MODEL	: Data block directive
BCMOD	: Changing boundary condition
RUPTURE	: Double-ended break
ASSIGN	: Directive to define and change the break laws

6.1.4.4.2 Thermal component sub- modules

WALL, **WALL3D**	:	Wall
FLUMOD, ADIABWAL, RADIAT	:	Modifying heat exchange
POWER	:	Modifying generated power
PNRSHAPE/PNRSHAPEX	:	Modifying power profile
EXCHANGER	:	Wall exchangers
ECHPOWER	:	Initialisation of exchanged power
EXCPOWER		Power from gamma radiation
FUEL, **FUEL3D**, **FUELCHAR**,AND, INTEGRATE	:	Rod fuel definition
POWER	:	Modifying generated power
FUELDTMX		Time step management before and after the clad rupture
FUELPLAQ	:	Plane fuel definition
PNRSHAPE/PNRSHAPEX, PRESHAPE/PRESHAPX, XNEULIST/XNEULISX	:	Modifying power profile
GOFUEL	:	Launching thermo-mechanics computation
STOPNEUT		Only for a FUELPLAQ Stop the point kinetics model (CORE) and used the user's fuel power laws
RSETFUEL, STOPFUEL,	:	Special directive for SCARFUEL option
PICTG	:	Post-processing of cladding temperature peaks
MODXNEUT	:	Modifying energy distribution parameter
OXRATE	:	Performing several oxidation calculations
SPALL, SPALLOX	:	Elimination of external oxide layer in fuel thermo-mechanics calculation
ALWMIN	:	Modifying the dry out criterion

6.1.4.4.3 Other thermal sub-modules

SGCARACT also defined in **WALL** (POINTSG key word),	:	Point steam generator
GVPPOWER	:	Initialisation of exchanged power
SGFEED	:	Modifying steam generator supply
CORE also defined in **FUEL** (NEUTRO), **FUEL3D** (NEUTRO), **FUELCHAR** (NEUTRO)	:	Point kinetics
XNEULIST, XNEULISX	:	Definition and activation of profile for energy distribution
GONEUT	:	Launching point kinetics model
SCRAM	:	Defining reactivity due to emergency shutdown
STOPNEUT	:	Only for a FUELPLAQ Stop the point kinetics model (CORE) and used the user's fuel power laws
FUELDTMX	:	Time step management before and after the clad rupture
REFLCHAR, **REFLCH3D**	:	Rewetting, reflooding specifications
REFLOOD	:	Launching reflooding model

6.1.4.4.4 Gadgets

Generally, to be taken into account in the thermal hydraulic computation, these objects not only have to be defined in the data block but also have to be connected in the command block (the directives to connect and disconnect them are given after the operator name).

TEE	:	Tee branch
CONNECT, GEOM, LINK, SINGULAR	:	Data block directives
ENABLE, DISABLE		For the option JETPUMP of the TEE it concerns the spatial meshing of the injectors
CCFL, ENABLE, DISABLE	:	Counter-current flow limitation
ACCU, OPEN, CLOSE	:	Accumulator (automatically connected)
ENABLE, DISABLE		For the pressure drop of the valve hysteresis
BREAK, OPEN, CLOSE	:	Break
CANDLE, OPEN, CLOSE	:	Sink/source of energy
PIQBREK, OPENBREK, CLOSE	:	Break
PIQREV, OPEN, CLOSE	:	Sink/source (mass)
PIQSEB/**PIQSOUP**/**PIQVANNE,** OPEN, CLOSE	:	Valve
PUMPCHAR	:	Pump (automatically connected)
PUMPMOD		Modifying pump characteristics
BLKROTOR, STOPPUMP, STARPUMP	:	Failure modelling
PCAVIT	:	Cavitation modelling
SENSOR	:	Sensor (automatically connected)
SINK, OPEN, CLOSE	:	Sink (mass)
SOURCE, OPEN, CLOSE	:	Source (mass)
SOURCEMOD	:	Modifying feeding laws
SGTR, OPENBREK, CLOSE	:	Steam generator tube rupture
TURBINE	:	Turbine (automatically connected)
MODPCST, MODPNST	:	Changing the kind of definition (power or efficiency)
TCOMCHAR	:	Turbine or compressor (automatically connected)
TCOMMOD	:	Modifying TCOMCHAR characteristics
CHECK VALVE, CONTROL VALVE, OPEN, CLOSE	:	Valve
FLOW LIMITER, OPEN, CLOSE	:	Flow limiter
ECHECK/ **ECVALVE,** OPEN, CLOSE	:	Valve

6.1.4.4.5 Special PWR components

PIQARE	: Steam generator feedwater overflow
STAROVRF, STOPOVRF	: Launch and stop the model
FLOMIXER	: Vessel loop mixing
STARMIX, STOPMIX	: Launch and stop the model
EXHYLINK	: Explicit thermal hydraulic connection between objects
SWITCH	: Launch and stop the model
SINKRRI, OPEN, CLOSE	: RRI sink
SOURIS, OPEN, CLOSE	: RIS overflowing model

6.1.4.4.6 Radio-chemical and boron components

RADCHEMI	: Assembly of radio and chemical components
ZONBAMOY	: Defining a reference zone (assembly of modules)
INIBORA, GOBORA,	: Initialise and launch the transport model for the radio and chemical components
ACTEMIS	: Trigger the activity emission due to the emergency shutdown

6.1.4.5 For stand-alone fuel computation definition and management

Refer to Chapter 7 for more information on stand-alone fuel calculations.

Operators and directives used in preparatory standard computation:

STOREHYD, **HYDCHAN**, RESHCAT	: Preparation of calculation

Specific operators and directives for stand-alone fuel computation:

CATAFUEL	: Stand-alone fuel head structure
HYDIMP	: Hydraulic channel
READHCAT, READHEXT	: Defining the type of thermal hydraulic input data
HYDMOD	: Changing the hydraulic channel
STDYFUEL	: Initialisation of the computation
TRANFUEL	: One time step of stand-alone fuel
COMPHYD	: Add several hydraulic files in another one

6.2 Data block

The data block consists of **the description of the reactor**, that is

- the definition of all module specifications:

  topological (connections between modules),

  geometrical (length, high, diameter, angle, etc.),

  numerical (mesh),

  hydraulic (type of flow, hydraulic diameter, flow section, etc.),

- the definition of all sub-modules and gadgets connected to these main modules,
-
- the choice of chemical elements and non-condensable gas,
- The assembling of circuit(s) and reactor.

To describe the reactor geometry, two concepts are then used. The first is the **JUNCTION** notion and concerns the object element connection. Refer to §5.3 introduction for more information.
The second concept concerns the **WEIGHT** notion. This is used to simplify the data file while avoiding repeating identical parts of the reactor.

6.2.1 The junction notion, the junction head loss coefficients and the momentum equations

Refer to §5.3 for an introduction to the notion of junction.

The singularities on junctions are defined in each module by the directive SINGULAR.
Two head loss coefficients are defined:

- for AXIAL, TEE and THREED elements: on vector nodes in the mesh direction and in the reverse direction,
- for a VOLUME: on a junction for outgoing and incoming velocities.

CAUTION: At junctions, two definitions can be found for the same coefficient and CATHARE does not check the coherency of these data.

Remark: The head loss values can be changed during the computation (in exec block) using the WRITE directive and COEFROT1 and COEFROT2 (AXIAL, TEE, VOLUME) or XFR1x/y/z and XFR2x/y/z (THREED).

The definition used is the one given for the element which writes the momentum equation (M.E) on the junction vector point.
This depends on the kind of connected modules:

→ Writes M.E ?	AXIAL / TEE	VOLUME	THREED	BCONDIT			RUPTURE
connected to ↓				a	b	c	
AXIAL / TEE	1	0	0	2	0	1	1
VOLUME	2	1	2	2	0	1	1
THREED	2	0	1	2	0	1	1
BCONDIT a	0	0	0	-	-	-	-
BCONDIT b	2	2	2	-	-	-	-
BCONDIT c	1	1	1	-	-	-	-
RUPTURE	1	1	1	-	-	-	-

BCONDIT type a is BLIND or SAFETYVA model of a boundary condition, type b is BC4 or BC5A, BC5B, BC5HO model, type c is BC3 or BC5XX or BC5YY or BC5ZZ model.

2 means the module always writes the equation.
1 means the equation is written depending on the liquid velocity sign.
0 means the module never writes the equation.
- means an impossible configuration.

Remark: for a VOLUME module, the reader gives a default value for singular head loss coefficient at a volume junction: **1 for incoming value and 0 for outgoing value**. This definition is replaced by the user-defined value if it exists.

Remark2: When a THREED element is connected to an AXIAL element, since it never writes the momentum equations and to avoid user errors, the CATHARE reader does not allow the user to define a junction head loss coefficient in THREED definition (CONNECT directive).

6.2.2 Weight notion

6.2.2.1 Weight for hydraulic elements

In CATHARE, the notion of weight appears twice:

- in connections between elements,
- in a volume or a 3D element concerning junctions.

The weight of an element is the number of identical elements simulated by this element. For example, for a reactor with three loops, the broken loop is simulated by elements with weight 1, and the intact loops can be described with a single loop and weight 2 in each element.

Inside a capacity or a 3D (VOLUME or THREED) element, the weight notion for each junction is used. It is used to connect several identical elements (or assembly of elements) to the module.

Remark: This is a way of simplifying the model, but has a drawback: the user loses the possibility of representing dissymmetry effects.

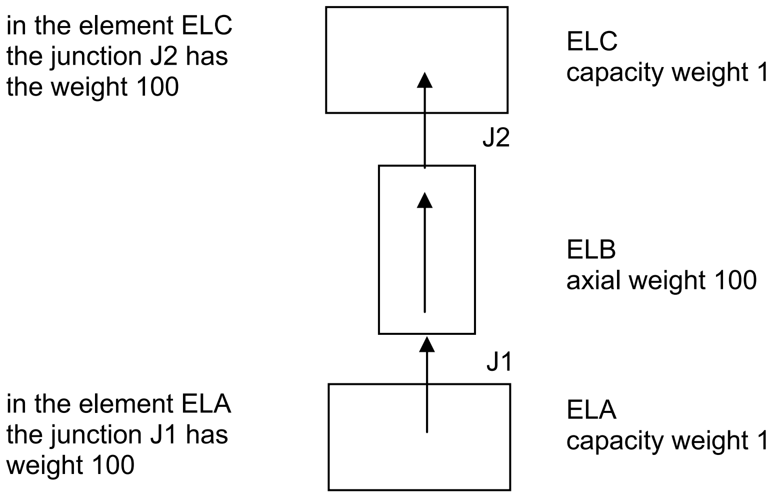
Remark2: It should be borne in mind that the weight has to be taken into account to ensure flow section, gravity and friction perimeter continuity at junctions.

In VOLUME elements the junction data SECT and PERI are those of an elementary junction linked to an elementary axial element:
SECT: section of one elementary axial element
PERI: wall friction perimeter of one axial element
CAUTION: SIZE data should be equal to the total size (because it is used to determine flow distribution in sub-volumes).

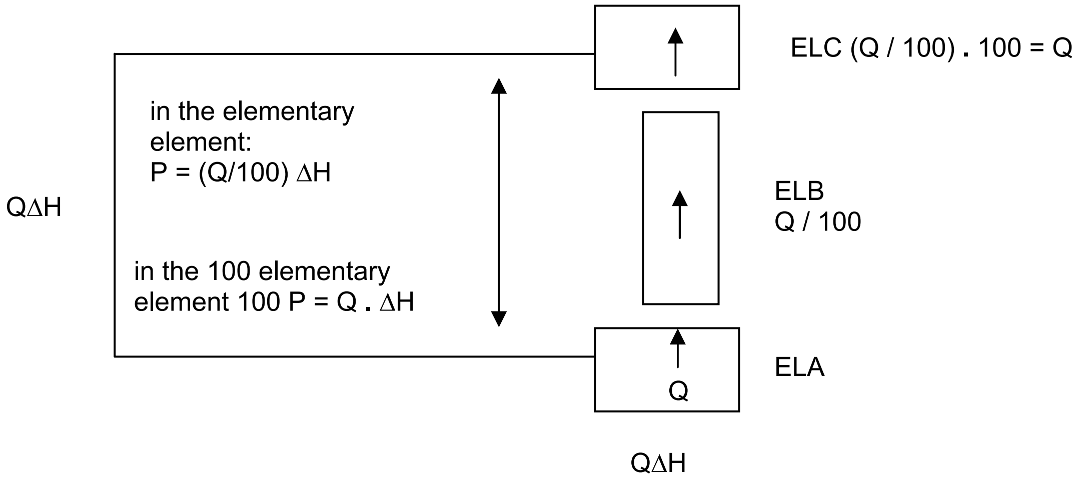
The weight value n is then attributed to the junction connected with n elementary elements modelled by an axial element with weight n.

In the AXIAL element all geometrical data are those of an elementary axial element.
SECT: section of one elementary element,
PERI: wall friction perimeter of one elementary element,
SIZE: size of one elementary element.

Example: Consider three elements: ELA, ELC (VOLUME elements) and ELB (AXIAL element).
J2 and J1 are the junctions:

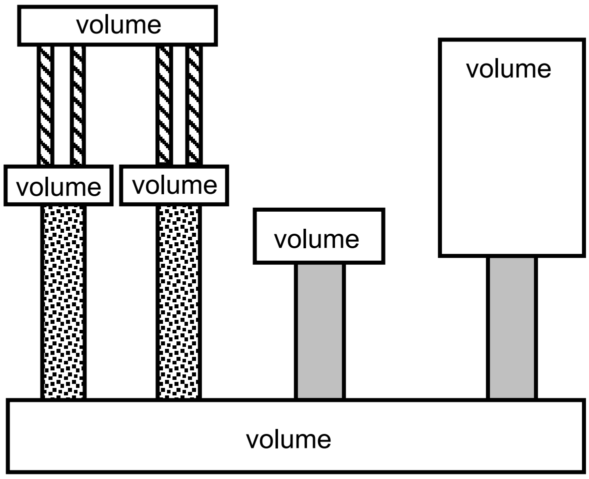


The flowrate and energy balances in steady conditions can be written:

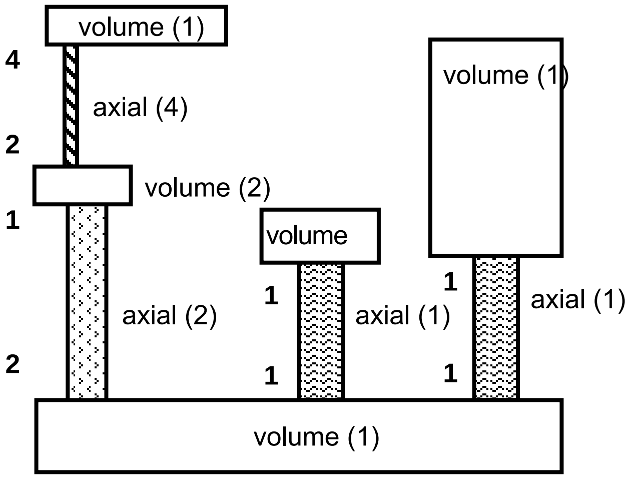


This example shows the specifications of mass and energy flowrates when using the weight notion.

Example: These two figures illustrate how to use the weight notion to simplify a hydraulic circuit description.



Actual Plant



CATHARE schematisation

6.2.2.2 Weight for sub-modules and gadgets

The WEIGHT is a property that is inherited between a module and its connected sub-modules or gadgets.
Example: an AXIAL element is defined with a weight 3. A sink gadget is connected to the third scalar mesh of this axial element. Each of the 3 “real” pipes has a sink connected to its third scalar mesh.
And if this sink is activated, it will be activated for the three pipes. Consequently, there is no way of imposing a different extracted flowrate for one of these three “real” sinks (because in fact for CATHARE there is only one AXIAL element and one SINK gadget).

This is the same for all gadgets and sub-modules.

The TEE is the only gadget or sub-module on which a particular weight may be defined (with respect to the main AXIAL module). This is due to the special correlations used for flowrate distribution and stratification making it possible to define several tee branches connected to the same scalar mesh.

Example: an AXIAL element is defined with a weight 3. A tee-branch gadget with a weight 2 is connected to the third scalar mesh of this axial element. Each of the three “real” pipes has two tee-branches connected on its third scalar mesh.

Remark: Case of an EXCHANGER wall defined between two AXIAL elements not having the same weight – Calculation of the heating perimeter: the user must input the heating perimeter seen from the wall element.

Example: An exchanger is defined between PIPE1 of weight 1 and PIPE2 of weight 500.

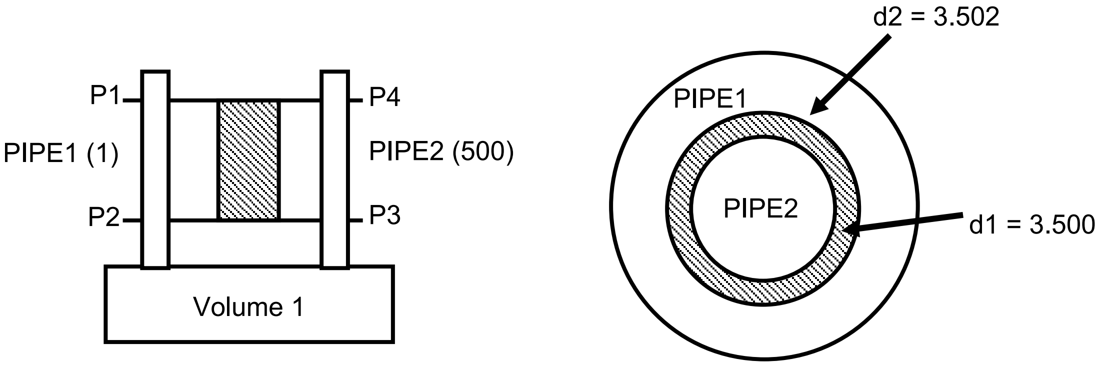


Figure 31: Heat exchanger between elements with different weight

If the user wants PIPE1 to be the secondary element, the syntax will be:

```
ECM1 =     EXCHANGE PIPE1 PIPE2 WALL
           SEGMENT P1 P2 REVERSE P3 P4
           ACIER533 DATA 2 DIAM 3.500  3.501  3.502
           HPERIM CONST (π x d1/500);
```

If the user wants PIPE2 to be the secondary element, the syntax will be:

```
ECM1 =     EXCHANGE PIPE2 PIPE1 WALL
           SEGMENT P3 P4 REVERSE P1 P2
           ACIER533 DATA 2 DIAM 3.500  3.501  3.502
           HPERIM CONST (π x d1);
```

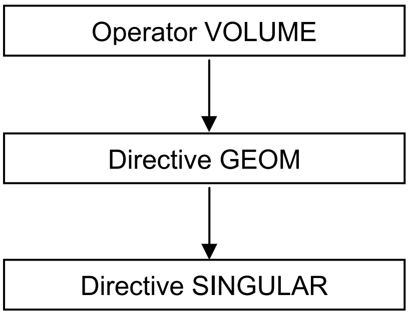
CAUTION: the definition of initial exchanged power (ECHPOWER) will not be the same in both cases.

6.2.3 Building the reactor

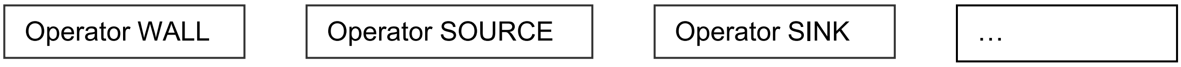
For each circuit in the reactor the following steps are recommended:

Step 1: Each element, one after the other, has to be completely defined, as well as the belonging-to sub-modules (thermal sub-modules, PWR sub-modules or gadgets). The steps for each kind of module are summarised below.

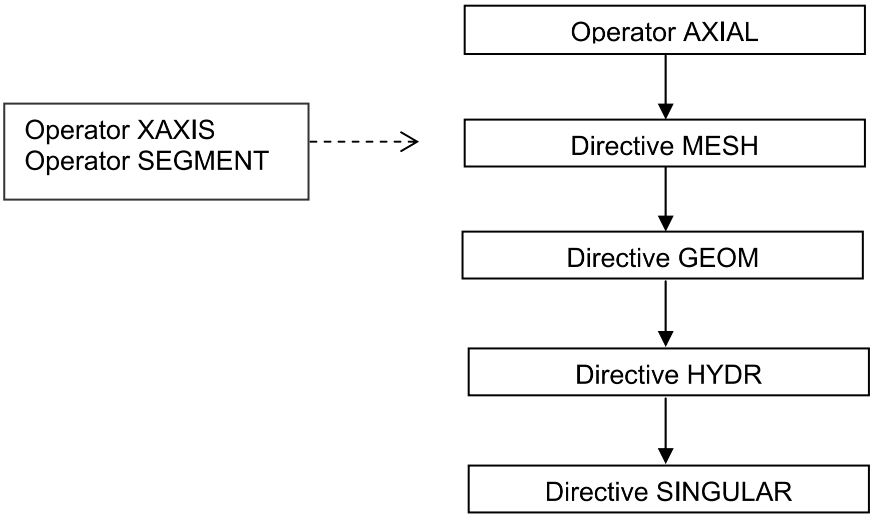
For all the **VOLUME** modules in the circuit:



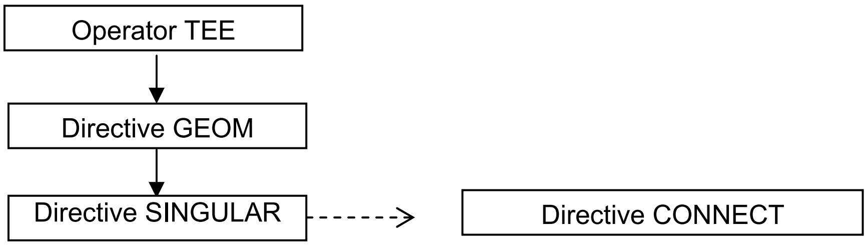
and then sub-modules and/or gadgets associated with this element:



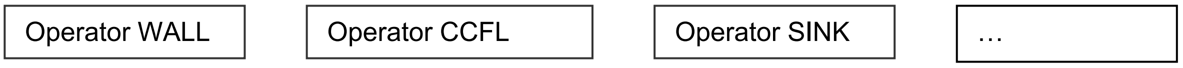
For all the **AXIAL** elements in the circuit:



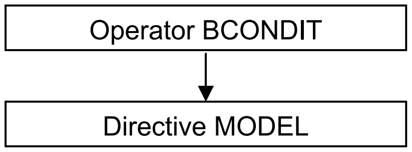
And, if needed, the **TEE** gadgets belonging to it:



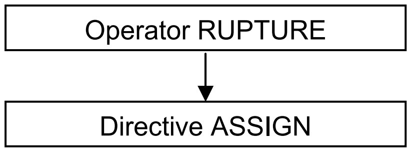
and then sub-modules and/or gadgets associated with this element:



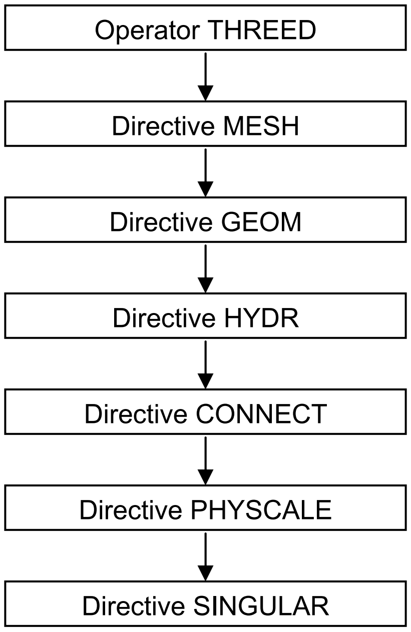
For all the **boundary conditions** in the circuit:



For the double-ended break:



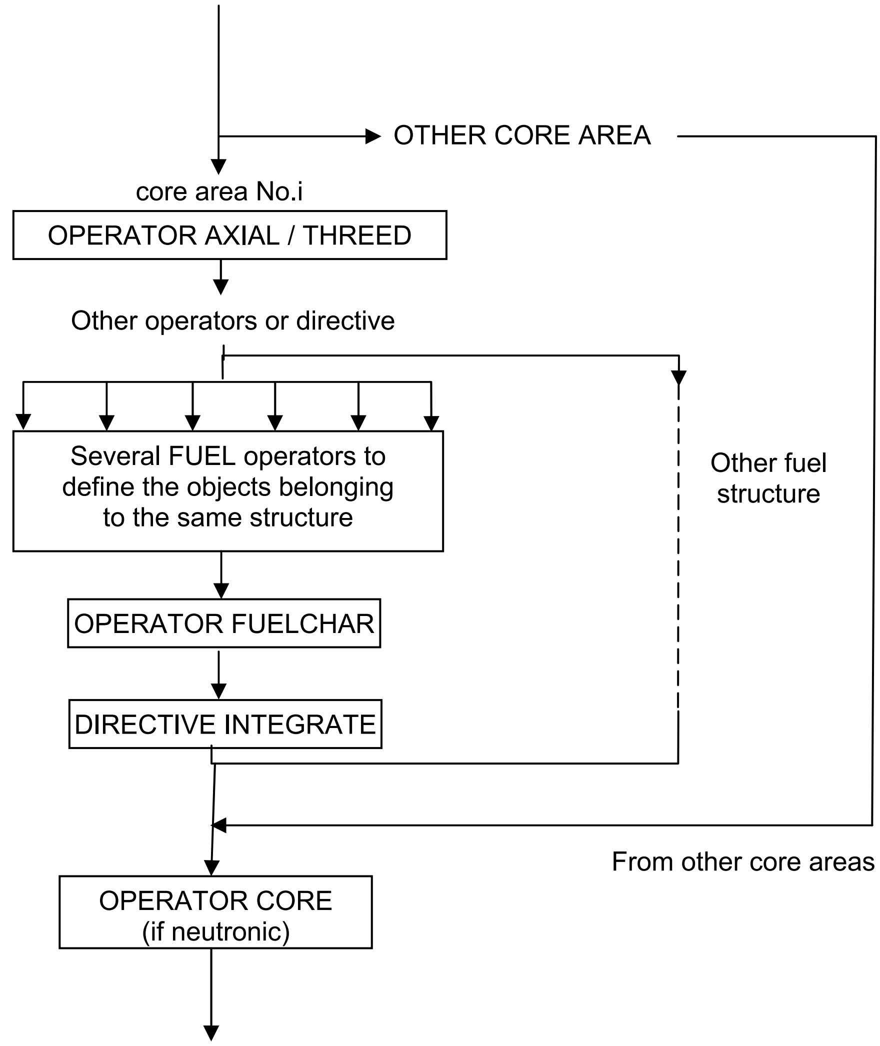
For the 3D module:



and then sub-modules and/or gadgets associated with this element:

Operator WALL
Operator CCFL
…

Remark: Definition of a **core**
The definition of a core in a circuit is all the more complex as its properties have to be defined in several operators:


OTHER CORE AREA
core area No.i
OPERATOR AXIAL / THREED
Other operators or directive
Several FUEL operators to define the objects belonging to the same structure
Other fuel structure
OPERATOR FUELCHAR
DIRECTIVE INTEGRATE
From other core areas
OPERATOR CORE (if neutronic)

Remark: Only main operators and directives are mentioned above, refer to §6.1.4 for the complete list of available operators and directives that may be used.

Step 2: Definition of the components other than water in the circuit with NONCOND or RADCHEMI operator.

Remark: It is advisable for the user to check the definition of several elements with respect to these definitions (forgetting radchemi or non-condensable gas concentrations / mass fraction is a common user error):
- MODEL definition of defined boundary conditions,
- ACCU definition (if NOCLOSE option is used, NITROGEN should be defined as a non-condensable gas).

Steps 1 and 2 have to be repeated as many times as there are circuits in the reactor.

Step 3: Definition of the modules which belong to several CIRCUIT modules: heat exchangers (EXCHANGER, SGTR), special links between elements of different circuits (EXHYLINK).

Step 4: Definition of circuits (CIRCUIT).

Remark: do not forget to include in the circuit element list the NONCOND, RADCHEMI and CORE names if they exist.

Steps 1 to 4 have to be repeated as many times as there are reactors in the data block

Step 5: Definition of the modules which belong to several REACTOR modules: special links between elements of different reactors (EXHYLINK).

Step 6: Definition of reactors (REACTOR).

6.2.4 Common user data block errors – CATHARE data block controls

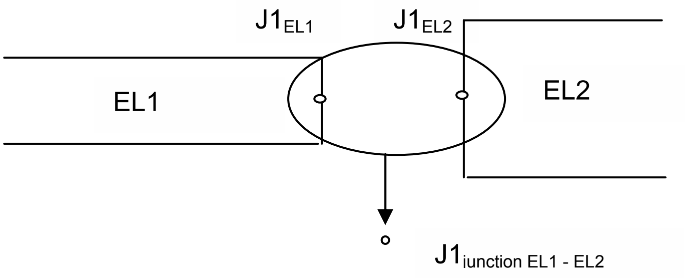
At the beginning of the CATHARE execution, e.g. the second step of the calculation (see §4.2 to understand the different steps of a CATHARE calculation) further checks of the hydraulic circuit definition are made that may induce fatal error messages. These are described below.

6.2.4.1 Assembling control

CATHARE checks that every junction is defined twice in two different elements and that the junction has been defined once as USTREAM and the other time as DSTREAM. This is a way of checking that the circuit is properly '"closed".

6.2.4.2 Junction coherency control

Refer to §5.3 introduction for more information.
The junction vector point is defined in the two connected elements. It should have then the same geometrical parameters on both sides: flow area, gravity, weight and friction perimeter.

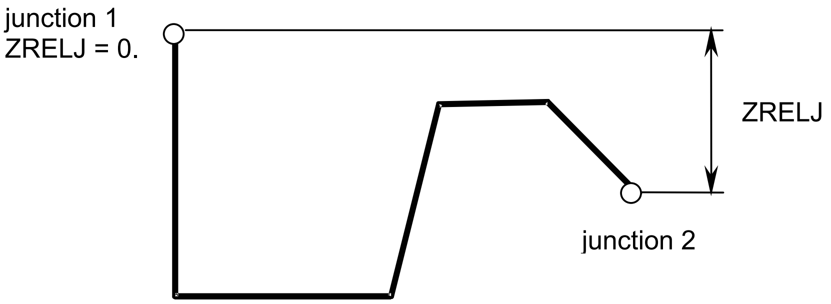


6.2.4.3 Elevation check

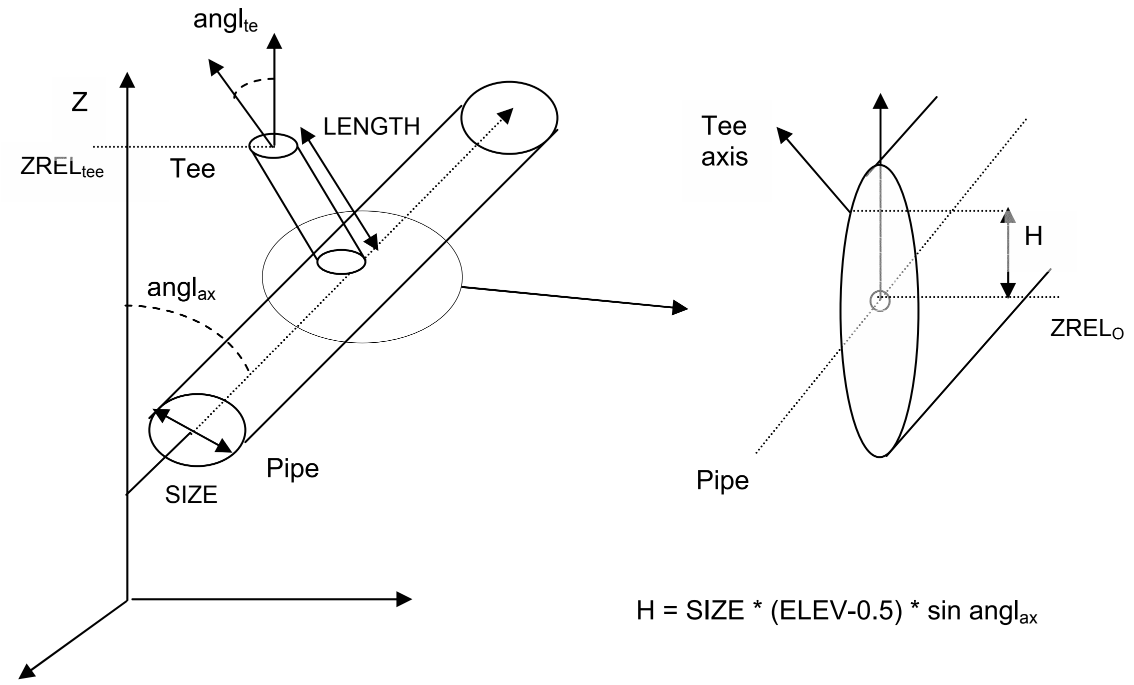
This is the first check made by CATHARE at the beginning of the calculation. Three kinds of information can be found in the standard printout.

6.2.4.3.1 Relative elevations of junctions inside an element

For an AXIAL element, relative elevations (ZRELJ) are given with respect to the first junction



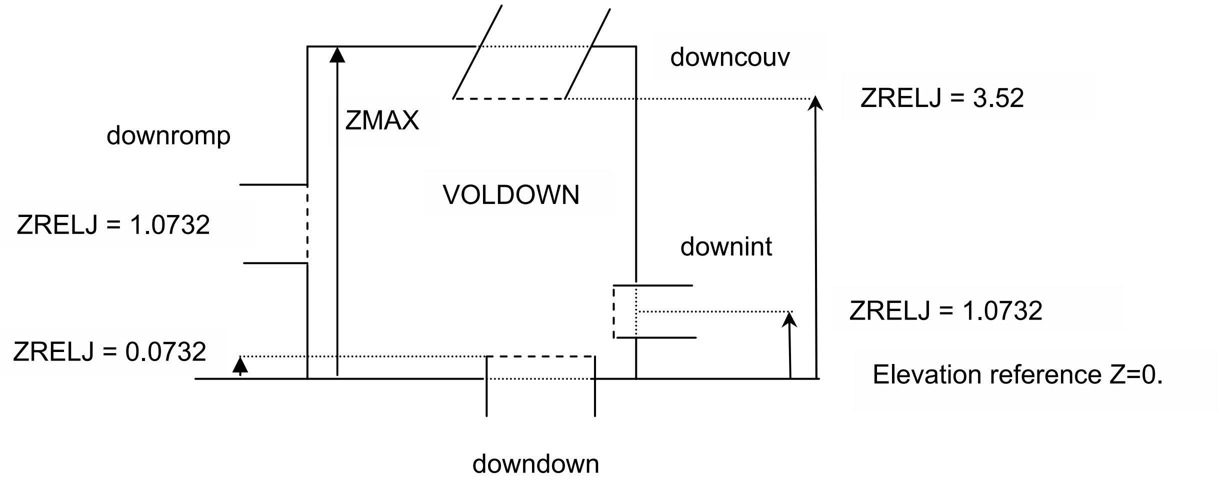
If a TEE branch is connected to this AXIAL element, the relative elevation of the TEE junction is calculated using the geometrical definition of the pipe and the tee branch (GEOM directive).



H = SIZE * (ELEV-0.5) * sin $angl_{ax}$

$ZREL_{tee}$ = $ZREL_O$ + SIZE * (ELEV- 0.5) x sine ($angl_{ax}$) + LENGTH * cosine ($angl_{te}$)

For a VOLUME, relative elevations (ZRELJ) are given with respect to the bottom of the volume.



It takes into account

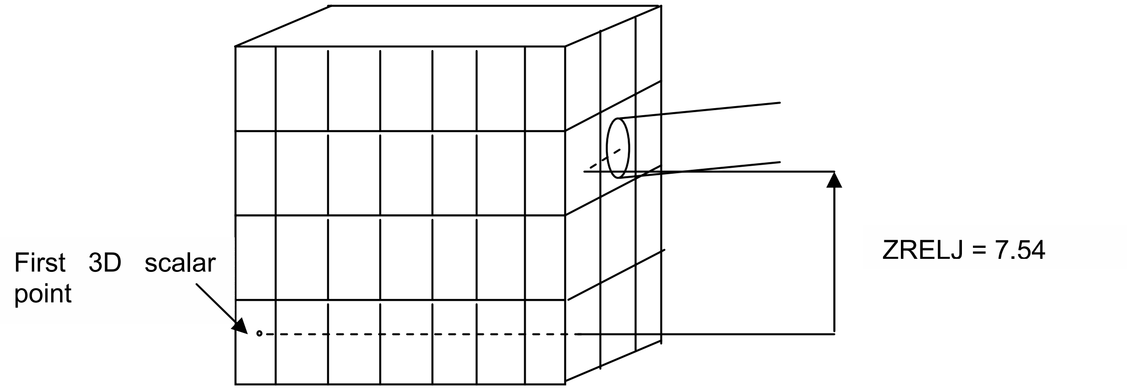
- the penetration length (see LENGTH in GEOM),
- the position of the junction (TOP, BOTTOM, ELEV)
- and its inclination cos (0 for HORIZONTAL, 1 (upward) or –1 (downward) for VERTICAL, cosine of inclination (cos) if INCLINED):

If ELEV < 10^{-5}, then ZRELJ = - LENGTH * cos

I f ELEV > ZMAX - 10^{-5}, then ZRELJ = ZMAX - LENGTH * cos

Else then ZRELJ = ELEV - LENGTH * cos.

For a THREED element, relative elevations (ZRELJ) are given with respect to the first scalar mesh.
Example of a 3D vessel:



In case of a junction which does not correspond to an entire THREED cell face (like in the example above), a junction length can be defined and is taken into account in the elevation calculation (DELTAZ in CONNECT directive). (This is to define a relevant integration length with respect to the control volume used for the junction momentum equations).

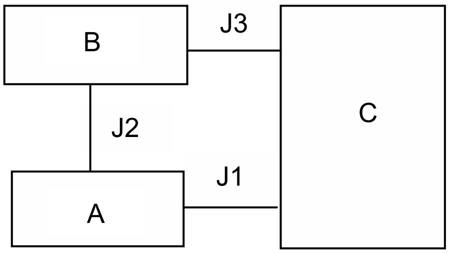
For a BCONDIT or a RUPTURE object, the relative elevation is defined as zero.

6.2.4.3.2 Elevation of junctions in the circuit and elevation of elements in the circuit

The elevation reference is given by the CIRCUIT operator: the reference junction is that between the first and the second element given in the CIRCUIT element list. The default elevation value is 0 for this junction but this can be changed by using the key word ELEV in CIRCUIT operator (see [DOC3] for more information).

Using this reference and using the previously calculated junction relative elevations, CATHARE processes all elements and all junctions in the circuit one by one and can calculate for each junction an absolute elevation (ZABSJ), but also an absolute elevation of the element (that is the absolute elevation of the element reference) (ZELEM).
For each junction J, this calculation can be done with the data of the two different elements A and B which shared the junction and it has to be consistent:
ZELEMa + ZRELJa = ZABSJa = ZABSJb = ZELEMb +ZRELJb

Example:



ZRELJ1c, ZRELJ3c, ZRELJ1a, ZRELJ2a, ZRELJ2b, ZRELJ3b are known (calculated in the first step)
CIRCTOT = CIRCUIT C A B;

1/ Consider the CIRCUIT definition: the first junction of the circuit is J1, the first element to be processed is C
ZABSJ1 = 0
Then ZELEMc = ZABSJ1 – ZRELJ1c
Then **ZABSJ3 = ZELEMc + ZRELJ3c**

2/ Then processing of element A (second element in the circuit definition)
ZELEMa = ZRELJ1a – ZABSJ1
Then ZABSJ2 = ZELEMa + ZRELJ2a

3/ Processing of element B
ZELEMc = ZABSJ2 – ZRELJ2b
Then ZABSJ3 = ZELEMc + ZRELJ3b

4/ The consistency of the J3 junction definition (1/ and 3/) can be checked here.

5/ This calculation is continued until all junctions have been calculated twice, thereby checking each junction definition:
Another calculation is run for the C element to check J2 junction
ZELEMc = ZABSJ3 – ZRELJ3c
ZABSJ2 = ZELEMc + ZRELJ2c
, etc.

A message is sent if the error is greater than 10^{-6} m. The calculation stops if the error is greater than 10^{-4} m.

Remark: If a discrepancy is printed for a junction, it may not be clear which data have to be corrected. The discrepancy may not occur at the considered junction. Another incorrect data element causes two or more error messages. It is recommended to use a drawing of the circuit to find the error.

Remark2: **Further checks must be made by the user** since it cannot be done automatically:

- the cold and hot legs must be at the same elevations, on all the loops,
- the steam generator tubes start and end at the same level,
- symmetrical loops have the same elevations,
- the primary and secondary circuits are linked by SG tubes. The elevation of both should follow the plant data.

6.3 Executable block

This part of the input deck is the place where the calculation is managed. The available CATHARE facilities for general management are listed in the first section.

But the command block consists mainly of the description of **the computation to be carried out**, that is:

- initialisation of the thermal hydraulic state for the entire reactor,
- achievement of the initial steady state,
- transient scenario.

The means available in CATHARE to achieve these goals and the related user recommendations are developed in the second section.

The third section deals with special user facilities (Common user, UTIL masks).

The fourth section aims at gathering common user errors committed in this part of the input deck.

6.3.1 CATHARE general management

All specific operators and directives that can be used to manage the CATHARE calculation are listed in §6.1.4.2.

All the files written or read by CATHARE are listed in Appendix 1.

6.3.1.1 Numerical options

These options are managed with the OPTION directive.

The automatic choice of the CATHARE proposed time step does not guarantee calculation convergence.
Because of the time step resolution (fully implicit for 0D and 1D elements), the limitation of the time step is crucial for rapid transient simulation. This limitation is the user's responsibility using the DTMAX key word. The DTMAX default value is 1000 seconds.
The DTMIN key word is used to define a minimum value for the time step above which the calculation is stopped. This is useful in order not to spend CPU time in case of bad convergence. The DTMIN default value is 1 x 10^{-5} seconds.
Refer to [DOC2] HOW to CONTROL the TIME STEP? for more information and recommendations.
This choice can be made easier by checking mass and energy balances (see [DOC2] How to check the ENERGY and MASS BALANCE?).

Because of the Newton method used, another way to ensure convergence is to use the ITERMIN key word to give the minimum number of iterations to be calculated for each time step. ITERMIN 6 is generally a good choice to ensure proper time solution convergence of a calculation when a doubt remains. A larger value should be avoided because of calculation cost.

The MAXREP key word is used to change the maximum number of time step restarts/divisions during one time step (in case of bad convergence after 10 iterations). The default value is 10. For rapid transients this may be increased up to 20.

6.3.1.2 Physical option

The revision code is identified from a set of constitutive physical relationships. The CATHARE Team has validated three sets of closure laws which are known as revisions 5, 6 and 6.1.

V1.3L version was validated with the revision number 5 (revisions 6 and 6.1 are not available with this version). V2.5 version is qualified with the revision number 6.1. It is the default value for this version. This may be changed to 5 or 6 using the GRIDS key word (OPTION directive).

6.3.1.3 Computing options

6.3.1.3.1 Calculation performance

Using the PERF and PRINSTRU directives, the user can access CPU time performance. The PERF directive is used to activate the performance measurement and PRINSTRU to define the printout frequency of these performance values in PERF (Fortran unit 90) and MAILLAGE (Fortran unit 92) files (standard calculation) or in the standard output (SCAR simulator use). These directives apply to the entire reactor; the printouts will be made for each of reactor circuit.
The calculation steps monitored by these directives are described in detail in the next section.

In the MAILLAGE files, for each element of the circuit printout includes:

- its name,
- its type 'TUYAU 1D' for an AXIAL element, 'VOLUME 0D' for VOLUME, '3D' for THREED, 'BC' for BCONDIT, 'RG' for RUPTURE and 'UNKNOWN' for others,
- CPU time used for its Jacobian calculation (in milliseconds),
- CPU time used for its regeneration calculation (in milliseconds),
- CPU time used for its boron and activity transport calculation (in milliseconds),
- total CPU time used for this element (in milliseconds),

In PERF file, the user will find:

- General information for the circuit:

'NOMBRE TOTAL DE MAILLES ' total number of meshes in the circuit,
'NOMBRE D''ITERATIONS ', cumulative number of calculated iterations,
'NOMBRE DE PAS DE TEMPS ', number of time steps.

- Time information on the Jacobian and elimination steps:

'POSTE CALCUL JACOBIENS + ELIMINATION',
'TEMPS TOTAL ', total time spent on these steps (seconds)',
'TEMPS MOYEN / ITER / MAILLE', average time (seconds) per mesh and iteration.

- Time information on solver steps:

'POSTE CALCUL SOLVER'',
'TEMPS TOTAL ', total time used in these steps (seconds),
' DONT TEMPS SGELIM ', time spent on solver elimination step (included in solver time),
' DONT TEMPS SGREGE ', time used for solver regeneration step (included in solver time) .

- Time information on regeneration steps:

'POSTE CALCUL REGENERATION',
'TEMPS TOTAL ', total time spent on these steps (seconds),
'TEMPS MOYEN / ITER / MAILLE', average time (seconds) per mesh and iteration.

- Time information on updating circuit variable and Jacobian writing of boron and chemical component transport:

'POSTE CALCUL ACTIVITE CHIMIQUE (LFWA)',
'TEMPS TOTAL ', total time spent on these steps (seconds),
'TEMPS MOYEN / PDT ', average time (seconds) per mesh and iteration.

- Time information on boron and chemical component transport calculation:

'POSTE CALCUL ACTIVITE CHIMIQUE (LBAREG)',
'TEMPS TOTAL ', total time spent on these steps (seconds),
'TEMPS MOYEN / PDT ', average time per time step.

- Time information on the variable regeneration of boron and chemical component transport:

'POSTE CALCUL SOLVER DANS CIBASC.F',
'TEMPS TOTAL ', total time spent on this step (seconds),
'TEMPS MOYEN / PDT ', average time per time step.

- Total CPU time information:

' SOMME DES POSTES ', total time spent on the steps above (seconds) T1,
' TEMPS DU TRANSITOIRE', effective CPU time used for the calculation T2,
' TEMPS DIVERS TRANSITOIRE (NON INSTRUMENTE)', T2 –T1,
'CONSTANTES PREDICTION TEMPS CALCUL',
'NOMBRE DE PROCESSEUR:', thread number.

- average time values (milliseconds) per iteration:

'PAR ITERATION:'
'JACOBIENS + ELIMINATION' average time for Jacobian and elimination steps TJ per iteration,
'SGELIM', average time for solver elimination step per iteration,
'SGREGE', average time for solver regeneration step per iteration,
'REGENERATION', average time for regeneration steps TG per iteration,
'DIVERS SOLVEUR', average time for solver (except elimination and regeneration) per iteration,
'TOTAL / ITER ', average total time (Jacobian + elimination + regeneration + solver) per iteration.

- average time values per time step (milliseconds):

' PAR PAS DE TEMPS:'
'ACTIVITE CHIMIQUE (LFWA)', average time spent on updating circuit variables and Jacobian writing of boron and chemical component transport (T1) per time step,
'ACTIVITE CHIMIQUE (LBAREG)', average time spent on boron and chemical component solver (T2) per time step,
'SOLVER DANS CIBASC.F', average total time spent on boron and chemical component regeneration (T3) per time step,
'TOTAL / PAS', total average time spent on T1+T2+T3 per time step.

6.3.1.3.2 Parallel calculation

This section is freely adapted from an article presented at SNA2003 entitled "QUEST FOR THE REAL-TIME FOR THE SAFETY ANALYSIS CODE CATHARE 2 USED IN THE POST-ACCIDENT SIMULATOR SIPA" by A. Ruby (CEA), O. Antoni (CEA), V. Créach (IRSN), Ph. Dufeil (IRSN), Ch. Rose (EDF), F. Iffenecker (EDF).

Parallelism in CATHARE has been developed with a shared-memory model (using OPEN MP) since V1.5b version. Standardised and adapted to the numerical method and to the structure of CATHARE, it enables parallel tasks in 95% of the code with efficient parallel loops on the elements, and an optimised but limited parallelism in the solver.

The modular structure of the code makes it easier to identify zones of independent memory and independent computations that may then be treated simultaneously. In the numerical scheme of CATHARE (Figure 32: Numerical scheme of CATHARE), a time step is solved by means of Newton-Raphson iterations involving four stages, numbered (i) to (iv), which were instrumented in order to measure their CPU time costs on an N4 plant-series computation (1D-0D nodalisation). One iteration consists of three stages (i) to (iii):

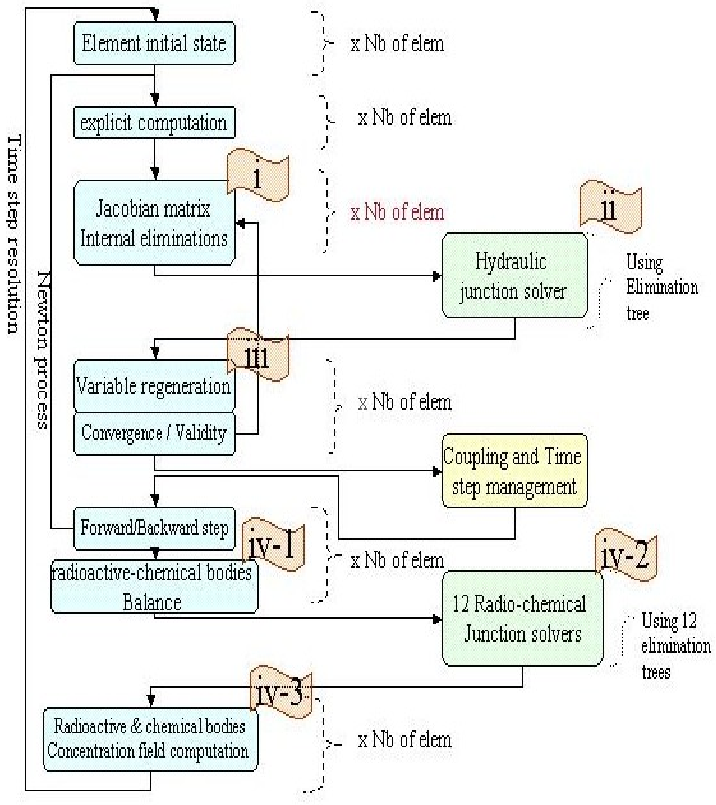


Figure 32: Numerical scheme of CATHARE

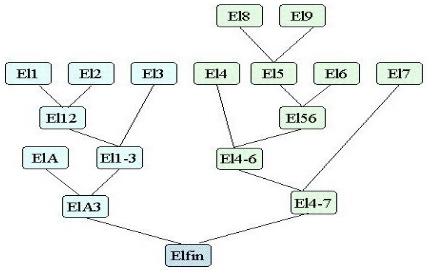


Figure 33: Example of elimination tree, Eli is the Jacobian contribution of element i.

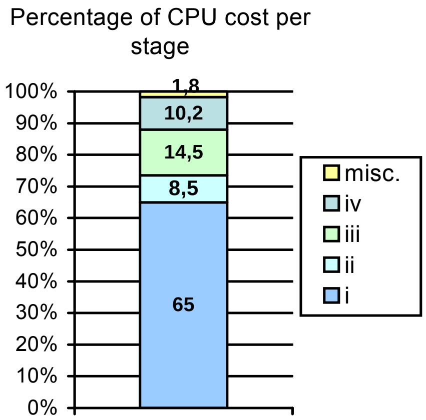


Figure 34: CPU use of each stage of the time step

(i) The building of an elementary Jacobian matrix for each module Ele $_i$: the mass, energy and momentum balance equations of each module are made independent by means of preliminary exchanges of information and pre-elimination of internal variables as functions of junction variables. This step is responsible for 65% of the CPU time.

(ii) The resolution of the global Jacobian matrix made up of the assembly of all the elementary Jacobian matrices (El_i) by a Gauss elimination. This is performed by an iterative algorithm that successively eliminates the blocks (El_i on Figure 33) of the matrix corresponding to a junction common to two elements (e.g.El1 to El2), resulting in the building of a virtual element (e.g.El12). This process continues as long as the non-eliminated junctions of this virtual element are common to at least one other element (e.g. El3A and El47 yielding Elfin). The order used to eliminate the elementary matrices is called an “elimination tree”. It depends on the reactor nodalisation and is optimised by CATHARE for a sequential resolution. As long as an element is not concerned by a junction it remains independent. This step operates on reduced matrices of around 2000 terms and is responsible for 8.5% of CPU time.

(iii) The regeneration and convergence tests for the junction and internal main variables of each module. This step may be achieved simultaneously on every hydraulic module. It is responsible for 14.5% of the CPU time.

(iv) At the end of the time step, the radioactive and chemical elements are computed with the converged values of the thermal-hydraulic variables (mainly temperatures and flowrates). In a sequential run with 12 elements, this step is responsible for 10.2% of the CPU time, involving three sub-steps:

- Each module writes the mass balance equations of the transported elements (iv-1), building the 12 corresponding elementary Jacobian matrices. This may be done simultaneously for every element.

- Then, for each of the 12 elements, the overall linear system is assembled and solved (iv-2).
- The solution values for the transported elements are computed by each module (iv-3).

Three possible types of parallel tasks are compatible with this scheme and with the data distribution in order to speed up the elementary time of parallel computations with respect to that of a sequential computation.
The chosen parallelism level should be specified in the input deck using the PARALLEL key word (OPTION directive).

The first type appears for the thermal-hydraulic system building steps (i), variable regeneration (iii), and radioactive and chemical element mass balance equation (iv-1): each hydraulic module (and its sub-modules and walls) writes independently the two fluid model equations, the associated source terms and the physical properties of the fluid. It also performs Gauss eliminations of Jacobian internal variables. So the hydraulic modules can be gathered in groups, each group being computed by one processor. This parallelism level uses results generated by the PESEE directive (OPTION PARALLEL PAQUET or OPTION PARALLEL ALL).

The second type is related to the resolution of the overall linear system. Instead of the sequential elimination described above, independent groups of elements can simultaneously eliminate the junctions involved in each group in different colors on Figure 33. Each group results in the building of a virtual element (e.g.El3A and El47). All of them still involving common junctions can be grouped and eliminated again, and so on until there remains only one final reduced system that can be solved. Hence, the treatment of parallel independent eliminations is possible. However, the number of parallel tasks is limited by the reactor model, the risks of access memory conflicts and the load balancing over the set of available processors. This type of parallel task needs a predefined tree. Refer to Appendix 11 for explanations on how to build an elimination tree (OPTION PARALLEL SOLVER or OPTION PARALLEL ALL).

The third type deals with the possible parallel resolution of all the radioactive and chemical elements, which are completely independent of each other. Each resolution is performed in sequentially, following the “binary resolution tree” used in step (ii) (OPTION PARALLEL SOLVER or OPTION PARALLEL ALL).

6.3.1.3.3 Convergence studies

Refer to [DOC2] HOW to EXPLAIN a TIME STEP DECREASE? and to [DOC3] (VERBOSE).

6.3.1.3.4 Coupling use

The MANAGE directive can be used to create a time cycle management which allows regular (with respect to the simulated time) communication between CATHARE and any other coupled code.

Case of the CATHARE - EDF-IRSN SIPA simulator coupling:
The INISIM directive has to be used to set the running mode of CATHARE to simulator use.
The MANAGE directive is used to force 0.1s time cycles. Using VERBOSE STAT 2 directive, this allows the post-processing of the per cycle delay between simulated and real time.
The OPTION HISTORY directive is used to store the calculation scenario (with respect to the different directive calls) to allow additional execution of the same calculation without simulator coupling (a CATHARE independent calculation).
The SENSOR objects have been created to display the same information which is available in French EDF power plants.

6.3.1.3.5 CATHARE size

CATHARE original requirements and Fortran ANSI77 compliance resulted in such a data structure with memory allocation done at the beginning of the calculation. The default allocated memory size is given in FAST.H file. It gives the maximum number of real number (double precision format), integer and character variables that can be used during the calculation.
This may be changed in the input deck using the FASTSIZE directive.

Among all the files managed by CATHARE, the binary result file (FORT21) is generally the biggest one (along with the cathar.exe program) but is also of primary importance. Consequently, in order to reduce its size, the default format of real variable storage, in this file, is simple precision; this precision proves to be sufficient for current post-processing purposes. This can be changed using the OPTION NCPUSAV directive: this is needed in case of a binary comparison between two calculations for example. Another simple way to reduce its size is to take care in the choice of storage frequency.

6.3.1.4 CATHARE management

6.3.1.4.1 Printouts

During CATHARE execution some printouts are made automatically, for example the steady state results. During transient computations (= for each TRANSIENT directive or for each time step), standard printouts also print out one line for each circuit: the physical time (TIME), the time step used to reach it (DT), the time step number (NSTEP), the number of iterations made in the time step (ITER), the total and real number of useful iterations made by the code since the beginning of the calculation (EFF. ITER. And T. ITER.). Depending on the machine, the CPU time can also be printed.
Example:
CIRCUIT CIRCB TIME 1.70555D+03 DT 1.000D+02 NSTEP 32 IN 1 ITER.; EFF. ITER. 115 T. ITER. 140 CPU 7.58D-01

Main information
The user can manage more printouts using the following directives which are fully described in [DOC2] (HOW to CONTROL the PRINTOUTS?):

LIST	:	Definition (topology, nodalisation and geometry) information, useful for checking CATHARE pre-processing translation of user input deck.
PERIOD, IMPRIME, PRIN3D	:	Thermal-hydraulic state information
VERBOSE	:	Debugging information (physical increments, etc.)
MESSAGE, TITLE	:	User messages

Remark:
Some of the directives listed above display their information in specific files which are specified in the corresponding [DOC3] definition.

Refer to Appendix 10 for a description of CATHARE transient printouts (automatic and PERIOD-dependent printouts).

Some operators and directives can be useful to get specific information on the calculation, which can then be printed:

Mass and energy balances

- Refer to the User Guide [DOC2] (HOW to CHECK the ENERGY and MASS BALANCE?) for more general information.
- In the PERIOD directive dependent printouts, information is available for mass and energy balances at several levels. This is described in Appendix 10.
- Specific facilities are available to define zones for balance calculations within a circuit (see [DOC2] WHAT is a ZONE?) which will be displayed with the PERIOD directive frequency chosen for the ZONE or the corresponding CIRCUIT:

BILAN3D	:	Allows the printout of balances for selected zones within a THREED element predefined with the MESH directive.
ZONE	:	Allows the definition of zones (combining AXIAL, VOLUME and THREED elements) within a circuit.
INIBIL	:	Allows balances to be initialised during a calculation.

Remark: non-condensable gas mass balances are displayed, as well as activity or mass concentration balances for radio and chemical elements, if they exist.

Calculation scenario

OPTION HISTORY	: Used to generate a file tracing the history of the calculation scenario (all directive calls and user messages).

6.3.1.4.2 Results

The RESULT directive allows the user to store any useful information on every kind of CATHARE object in the so-called result file (the Fortran Unit 21).
Remark: the OPTION NCPUSAV directive is used to choose the format of the real numbers saved in the result file.

Results are stored in binary format. They can be plotted by the CATHARE post-processing tool also called POSTPRO (see §4.2 and [DOC4]).

The directive RESULT initialises the writing operations, which are then regularly updated. The **frequency** has to be defined by the user (in seconds or number of time steps).

This directive can be applied to assembly modules and/or main modules. In its application, it follows the CATHARE structure hierarchy: the frequency chosen for the reactor is applied to all belonging-to modules (and all sub-modules and gadgets belonging to the modules).

There is no default result storage frequency.

6.3.1.4.3 Restart

CATHARE offers the possibility of restarting a calculation from a saved state. This saved state may be created either by CATHARE (in case of a detected error – refer to §6.3.4.1 - or bad convergence) or by the user using the SAVE directive. In both cases, all the information needed (characteristics and physical state of the reactor as well as input deck variables values) is stored in the V25_1.RESTART file, also called restart file. To use these data in a future calculation, the user will have to create a restart input deck in which the save label will be specified in the RESTORE directive.

The SAVE directive is used to save all the objects in the memory at **the calling time**.
Several saves (corresponding to different calling times) can be made on a V25_1.RESTART file as the saves are sequentially written. To differentiate one save from another, the user must use a specific save label. A save label can be used only once. If the same save label n is used twice, at the second call, CATHARE skips the specific label n and writes the save under label n+1. The maximum label number is 999 (this is the label used by CATHARE in case of automatic save).
A maximum number of 30 saves can be performed during one calculation.

Example: Generating a save every 30 seconds of a scenario
In the command block, in a transient loop,

```
IF      (temps > tstore);
        nn = nn+1;
        SAVE nn;
        tstore = tstore + 30.0;
ENDIF;
Description: nn = 1     temps = 0.
             nn = 2     temps = 30.
             nn = 3     temps = 60.
             etc…
NB: do not forget to initialise tstore
```

There is no default save storage frequency.

The RESTORE directive allows the user to recall all the information needed to perform a calculation. Using the RESTORE directive, the user specifies the state to recall.

If no label is used (or 0), the state is assumed to be the initial state, then the V25_1.INIT file is read.
If a label is specified, a search is made for this label in the V25_1.RESTART file and the corresponding reactor state is restored as well as the user input deck local variables.

Refer to [DOC2] HOW TO PERFORM a RESTART? for an explanation on building the restart input deck.

Remark: A common user need is to begin the restart calculation at time = 0. RESETIME is the directive to use to reset time to 0 in a calculation. An instruction such as
Time = 0;
only changes the input deck local variable “time” to 0 but not the CATHARE calculation time, and should thus be avoided.

6.3.1.4.4 Hydraulic storage

Refer to [DOC2] for more information on WHY-WHEN TO USE THE STAND-ALONE FUEL? and to Chapter 7 for more information on stand-alone fuel computation.

To perform a stand-alone fuel computation, the user needs to store the hydraulic state in the channel surrounding the fuel wall. For a fuel wall (FUELCHAR) in an AXIAL element, the hydraulic channel is an AXIAL element itself. For a fuel wall in a THREED element, the user has to define the hydraulic channel which is a vertical column of hydraulic meshes and therefore must use the HYDCHAN operator.

The STOREHYD directive is used to store the hydraulic state for as many hydraulic channels as needed. The storage is done every time step in the Fortran unit 62.
Refer to Appendix 11 for a description of the variables stored by CATHARE in this file.

6.3.1.4.5 User's file management

During CATHARE execution the user can write or read in "personal" files. These files are formatted or not, but they have a specific form. They can a HEAD block (a title, a date, a version of CATHARE calculation, and the names of the columns of values) and the VAR block (columns of values)

Refer to [DOC2] for more information on (HOW TO USE a personal file ?):

OPENFILE : Allows the user opening a personal file (formatted or not).

READHEAD : Allows the user reading the heading in a personal opened file.
READVAR Allows the user reading variables in a personal opened file.

WRITHEAD : Allows the user writing his heading in a personal opened file.
WRITVAR Allows the user writing his variables in a personal opened file.

REWIND : Allows the user coming back to the beginning of a personal file (formatted or not).

6.3.2 CATHARE thermal hydraulic calculation

This chapter deals with the facilities available to model physical states with CATHARE. The first section presents the basic facilities allowing access to/interaction with the physical calculation.

To perform a calculation, the user first has to initialise the reactor state (that is the initialisation of all the state variables for every mesh of the reactor). This step, also called PERMINIT step, is described in the second section.
Then, depending on the difference between the state at the beginning of the transient (reference state) and the initial (PERMINIT) state obtained, the user may need to make adjustments in order to reach the transient starting point. This step, also called stabilised transient, is described in the third section

In the CATHARE language, to run either the first phase or both phases can be improperly called to "perform the steady state".
For an introduction to these two main phases, refer to [DOC2] HOW TO REACH the INITIAL or STEADY STATE?

Finally, general information on thermal-hydraulic calculation management of a CATHARE calculation is given.

6.3.2.1 General information

6.3.2.1.1 Interaction with the physical calculation

Some directives are available to read information on the CATHARE result in order to intervene "in real time" on the calculation to manage interventions depending on a physical state:

- The VALUE directive is used to get overall data (example: liquid mass in a volume, activation of a gadget) as well as precise data (example: pressure at the 2nd scalar mesh of a pipe) for all kinds of elements (main and sub-modules, gadgets, circuit, zone, etc.).
- The VALBA directive does the same as VALUE but for specific information related to radio and chemical components.
- The VALUFEAU directive allows direct access to the steam-water properties calculated and used by CATHARE.
- Besides the module specific directives, the WRITE directive is used to modify CATHARE input data during the computation (the WRIBA directive does the same but for specific information related to radio and chemical components).

CAUTION: some user actions are not compatible with mass and energy conservation and may be the cause of calculation non-convergence or errors (for instance, modifying the pressure within an element).

CAUTION: WRITE and VALUE access procedures are different. The user will generally not be stopped if trying to write on a VALUE-available CATHARE variable but his directive will only be taken into account if this is an INPUT CATHARE variable.

Example: writing the liquid temperature in a hydraulic mesh is possible but is not actually taken into account because it is a secondary variable derived from the main variables (pressure and liquid enthalpy). On the other hand, writing temperatures within a wall structure will be taken into account.

The use of WRITE and VALUE for every kind of CATHARE element is fully described in [DOC3]. The input deck variables thus defined are called CCV, i.e. CATHARE Computational Variables.

6.3.2.1.2 Achieving a time step

The basic instruction to launch a single time step calculation is the TRANSIENT directive.
In this directive, the user has to specify the time step to be used (dt) and the time to be reached if this time step value is used (time) that is the current time added to the proposed time step value:

```
time = time +dt;        (1)
TRANSIENT time dt; (2)
```

Remark: time and dt are user input deck variables. They may have any other names.

This proposed time step value is used first, but if after 10 iterations convergence is not obtained with this value (or in case of a thermal-hydraulic convergence out of CATHARE range), then CATHARE decreases it. Thus, the time step value finally used is not necessary the one given in the TRANSIENT directive.

When convergence is obtained, CATHARE calculates the real time reached taking into account the real time step used to reach convergence (which is smaller or equal to the proposed time step). This value may be read by the user using NEWTIME operator.

Then, depending on how difficult or how easy it was to reach convergence at the former time step, CATHARE calculates and proposes a new time step. The user may access this value using the NEWDT operator.

These two operators (NEWDT and NEWTIME) may or may not be used by the user to update his input deck variables time and dt:

```
time = NEWTIME;         (3)
dt = NEWDT;        (4)
```

For the standard user, there are only a few possibilities:

- Standard use: CATHARE management and time increase:

```
time = time +dt;
TRANSIENT time dt;
time = NEWTIME;
dt = NEWDT;
```

- Achieving a time step at time = 0 (this allows a steady calculation to be performed when time dependent laws have been defined) using a proposed CATHARE time step value: in this case, instruction (1) should not be used:

```
TRANSIENT time dt;
dt = NEWDT;
time = NEWTIME;
```

- Making a personal proposal for the time step value (this is only a proposal and CATHARE may not achieve convergence with this value). Thus, a sequence of instructions such as the following may be used:

```
time = time +dt;
TRANSIENT time dt;
time = NEWTIME;
dt = XXX;
```

or at time = 0.:

```
TRANSIENT time dt;
dt = XXX;
time = NEWTIME;
```

Remark: The first time step value of a calculation must be imposed by the user. A small value (10^{-3} for instance) is generally advisable following the initial state computation (GOPERM). In case of restore from a previous calculation, it is advisable to use the recovered value of NEWDT from the saved state (any other choice is under the user's responsibility).

6.3.2.2 Initialisation

6.3.2.2.1 Goal

The initialisation is a mandatory step for any CATHARE calculation because any time step calculation requires the knowledge of the initial values of all the variables.
To minimise the amount of data to enter and to facilitate this first calculation, this initialisation step aims at reaching a plant steady step.

6.3.2.2.2 Input deck facilities

The initialisation order for each hydraulic circuit is defined with the PERMINIT directive. This defines both water circulating zones and zones where there is no flow using the DEADZONE key word. Refer to [DOC2] HOW to INITIALISE a DEAD ZONE? for more information. Other useful information and examples can be found in [DOC3] (PERMINIT).

Then, the imposed thermal hydraulic state has to be defined. To perform this operation, the code needs to know the values of the main variables for, at least, one junction in each fluid zone of the circuit from which the initialisation will be propagated to the entire zone. This corresponds, in the input deck, to the definition of a real constraint point (REALC operator).

The REALAX and REALVO operators are equivalent to REALC but they must be used only for stagnant fluid zones (DEADZONE).

The IMPOSFLOW directive can be used to define a flowrate distribution between several outgoing junctions of an element (VOLUME or THREED).
The NOFLOW directive can be used to impose an almost zero flowrate at a junction.

The LEVEL directive can be used to impose a level in a VOLUME element.

As each hydraulic circuit is initialised separately, the coupling power through exchangers has to be defined by the user (ECHPOWER directive).

When the definition is complete, the initialisation calculation can be launched using the GOPERM directive.

Auxiliary tools:

- To reduce wall inertia: the thermal inertia of all walls and fuels may be temporarily reduced during this step to facilitate initialisation. For example the ρCp can be reduced by a factor of 100 (OPTION ROCP).
- To eliminate external thermal exchanges: the loss terms may be temporarily set to zero for selected or all walls using the ADIABWAL directive.

6.3.2.2.3 CATHARE calculation

The initialisation procedure is performed in two steps:

- The first step, following the initialisation order given by PERMINIT, calculates in each element the variable values which satisfy the stationary condition. The inlet conditions are propagated along the element taking into account a hydrostatic pressure correction. If there are walls, a conduction calculation will be done taking into account generating power or heat loss flux. All connected gadgets are taken into account during this calculation.
  The calculation is done to obtain a thermal hydraulic stationary state (until time step exceeds 100 s) with the inlet junction calculated pressures being monitored with respect to the imposed values (REALC).

  Knowledge of the thermal-hydraulic variables for all the inlet junctions is thus required. This implies that if the inlet junction conditions are not given by elements that have already been calculated, then additional real constraint points have to be defined.

  This algorithm can be used to calculate all the elements successively by specifying only a reduced number of flow parameters.

  Another easily predictable consequence is that all the imposed conditions may not be obtained by the code (either if they are not coherent or if the code finds another solution than the one expected).
  Remark**:** boundary conditions are not calculated during this step

- The second step allows the user to reach an overall coherent state. This is the stabilised solution of the standard transient calculation of ten time steps for the entire reactor (i.e. a calculation with time set to zero).
  Remark: during this step, contrary to the previous one, the following items are taken into account:

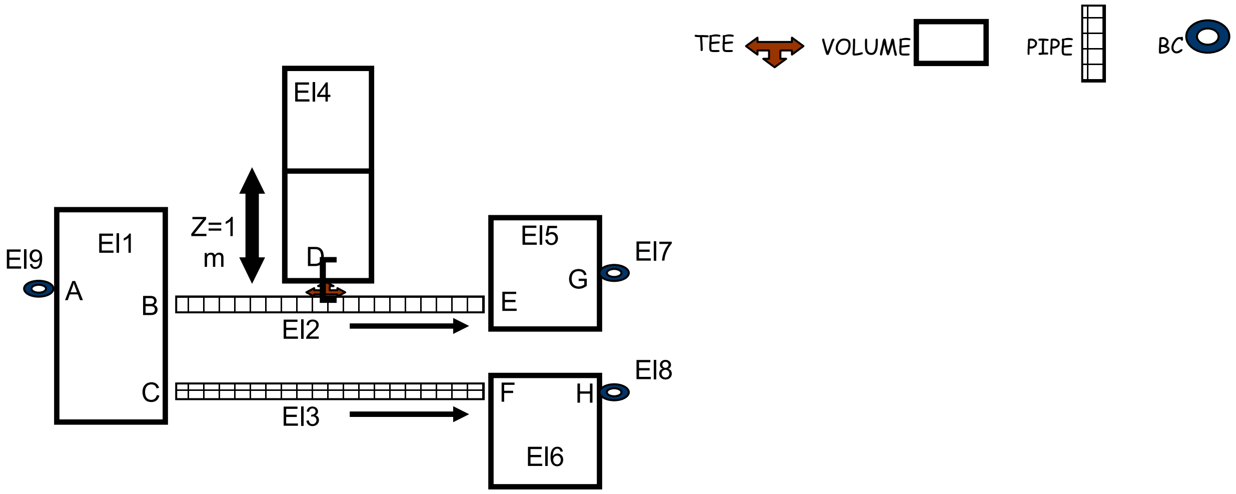
  - boundary conditions – and then their models as defined in the data block,

  - the real exchanged power between circuits, i.e. power calculated by CATHARE based on the physical state of each circuit and not that defined in the ECHPOWER directive.

At this step, the most common user error is the lack of coherency between the results obtained by the first step and the input deck imposed conditions.

6.3.2.2.4 Examples

Example 1: standard circuit



Users must choose:

- the starting point: A
- the calculation order: El1 El2 El3 El4 El5 El6 EL9 El7 El8

Remark: in the PERMINIT directive, the BC are given at the end of the list of elements (because they are not calculated in the first algorithm)

PERMINIT CIRC1 El1 El2 El3 El4 El5 El6 EL9 El7 El8;

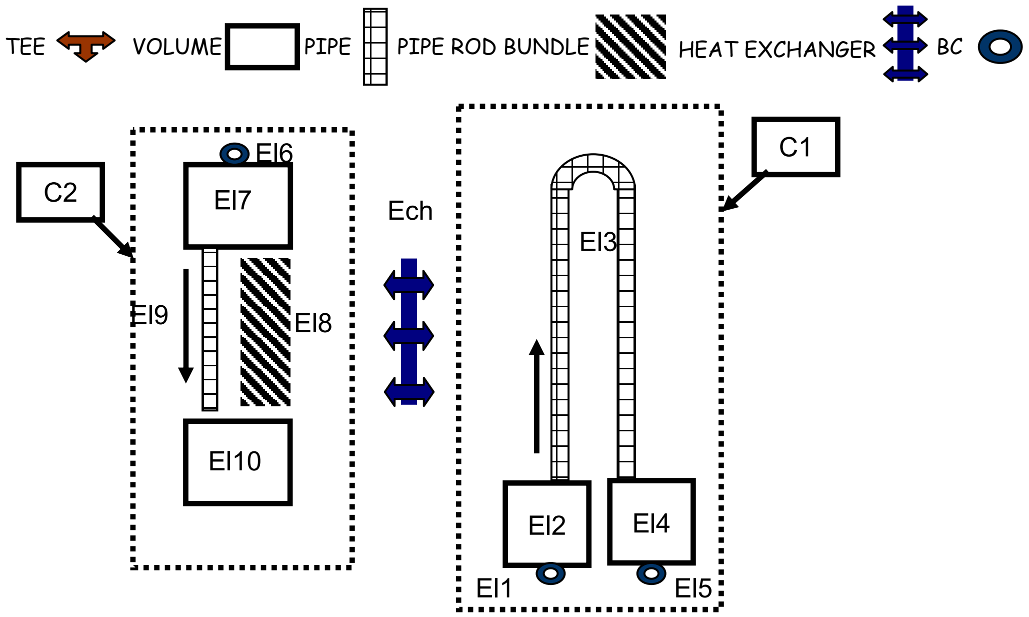
Optionally, other constraints may be imposed:

- flow distribution at the outlet junctions of the volumes:
IMPOSFLOW El1 B 0.500000 C 0.500000

- a no flow junction at D:
NOFLOW TEE D;

- the liquid level in El4:
LEVEL El4 1.D0;

Example 2: Two coupled circuits



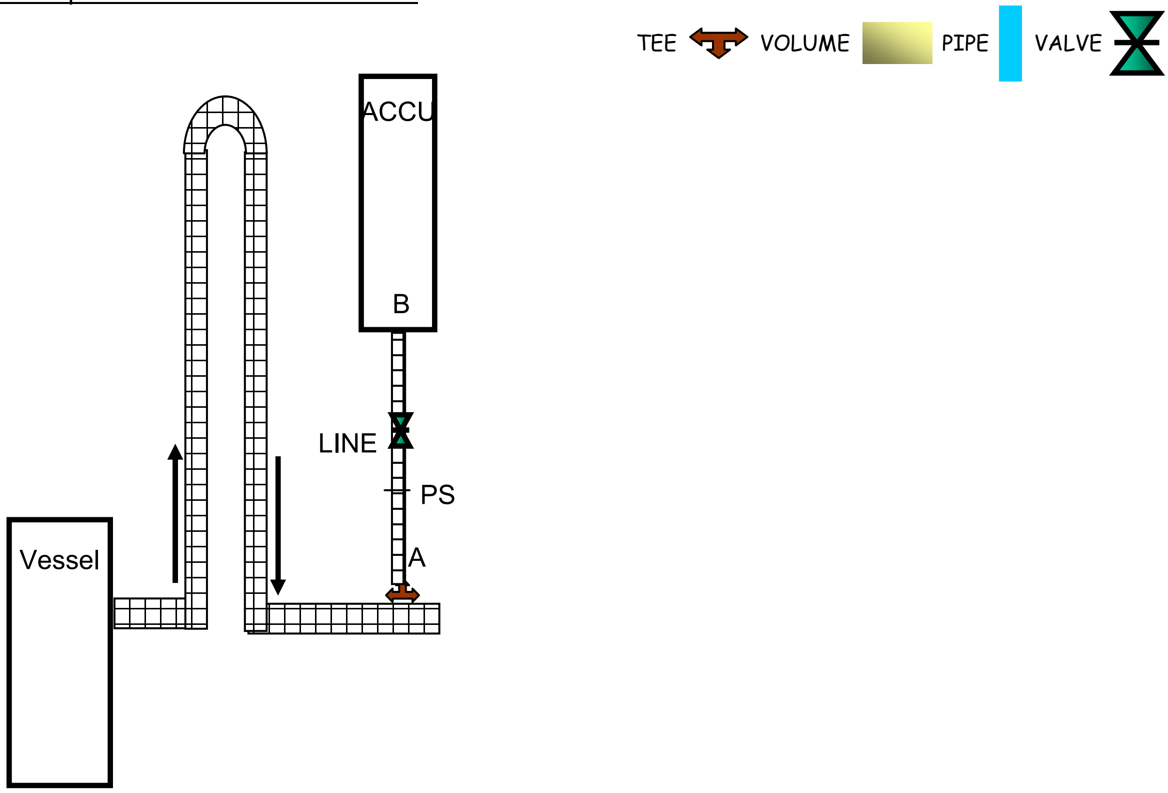
- For each circuit, definition of the calculation order and the corresponding real constraint points:

PERMINIT C1 EL4 EL3 EL2 EL5 EL1;
REALC …;
PERMINIT C2 EL7 EL9 EL10 EL8 EL6;
REALC …;

- Initialisation of the heat exchange power in the exchanger connecting the two circuits:

ECHPOWER Ech XXX;
GOPERM;

Example 3: dead zone initialisation



To initialise the PCS at 155 bars and the accumulator circuit at 40 bars:

- To define the initialisation order and the DEADZONE(s) DZ1 and DZ2:

PERMINIT XXX1 XXX2 … ;
DEADZONE DZ1 ACCU LINE B V1;
DEADZONE DZ2 LINE V1 A;

- To impose the circulating zone real constraint point:

REALC …;

- To impose a NOFLOW junction at A:

NOFLOW A;

- To initialise the dead zones:

DZ1 :
REALVO ACCU
LIQUID (low. sub-vol) HL,HV,X
GAS (upp. Sub-vol) P,HL,HV,X;

DZ2:
PS = SCALAR LINE 4.;
REALAX LINE PS
P,HL,HV,ALFA,X,VL,VV;

GOPERM;

6.3.2.3 Stabilised transient

6.3.2.3.1 Purpose

The purpose of this calculation step is to obtain a system at equilibrium with the required initial conditions (P at pressuriser, flowrate at secondary side, pump rotation speed, etc.).

6.3.2.3.2 Input deck facilities

Usually, this part of the input deck consists of a transient loop with regulations and with outlet condition tests set on a time step value (ex: DT > 100 s) to ensure the stability of the physical state obtained and its agreement with the imposed reference state (tests on physical variable values with respect to imposed values).

This transient block may be performed with time set constant to 0 or not.
Performing this phase with running time may be advantageous in order to post-treat the regulation results in order to check reactor convergence towards the desired state.
Then, the RESETIME directive can be used to reset time to 0 before beginning the real transient scenario.

Several tools may be used to reduce CPU time during this step:

- Reducing wall inertia: the thermal inertia of all walls and fuels may be artificially reduced to facilitate initialisation. For example the ρCp can be reduced by a factor of 100 (OPTION ROCP)
- Use of the PRSZER (VOLUME) key word (pressuriser volume)

The regulation theory and an application example for PWR initialisation are presented in §6.3.2.4.

Some regulation tools can be used but are not actually available in reactor plants or facility plants. For example, it is possible to regulate loop mass flowrates or flow distribution by changing pressure loss coefficients at volume junctions. It is also possible to regulate the temperature difference between a primary and a secondary circuit by introducing a fouling factor in the heat exchanger. In both cases, after convergence, the final values of the pressure loss coefficients or fouling factor have to be checked; realistic values may be used.

The user should check convergence, possibly by writing convergence tests in the input deck. When a stationary state is reached, regulations must be inoperative. No drift should occur after removing regulations.

To check that the state of the circuit becomes stable during the transient with regulations, the user can draw the following curves:

- liquid flowrate in the pump versus time,
- pump rotation speed versus time,
- feedwater flowrate and steam flowrate at the outlet of the cavity versus time; drawn on the same axis-system the two curves should join up when the system is stabilised,
- primary temperature versus time,
- mixture level in the cavity of the steam generator versus time,
- Liquid flowrate of the external source injecting into the cavity versus time. The value of the flowrate should be zero when the system is stabilised.

6.3.2.4 Regulations

Regulations should be written in the input deck for both the following cases:

- to represent actual reactor or experimental regulations,
- to obtain a stabilised initial state.

6.3.2.4.1 General expression

The proportional-integral regulation general formula is:

$$\tau \frac{\partial A}{\partial t} + A = GE = G\ (F_{oper} - F)$$

τ = integration time constant
A = controlling value
t = time
G = gain
E = gap = F_{oper} - F
F = controlled value
F_{oper} = operating point

- The first term of the formula takes into account an integrating effect:

$$\tau \frac{\partial A}{\partial t}$$

τ represents the time needed for an action (represented by a controlling value) to be taken into account by the circuit.

To illustrate this definition, an example is given where the pump rotation speed is used to obtain a stationary state in the circuit. In this case, modifying the rotation speed induces a change in the pressure, and this change is nearly instantaneous in the entire circuit (the pressure propagates at sonic velocity). But the pressure change is not the only factor reacting to the modification. Because of heat exchanges between primary and secondary sides, the state of the circuit will be stable after one particle has completed one "lap" in the primary circuit. In such case, the user should thus take into account the longest "characteristic" time connected to a physical modification.

Here, this value would be obtained by dividing the average velocity of a particle by the total length of the circuit.

- The third term of the formula is proportional to the gap between the controlled value and the operating point. The coefficient G is calculated using the following relation:

$$G = \frac{s}{3}$$

where s = sensitivity

s represents the ratio between a small variation in the controlling value and the corresponding variation in the controlled value. This can be expressed as follows:

$$s \approx \frac{\partial A}{\partial E} \text{ with } E = F_{oper} - F$$

A gain equal to one third of s is then chosen which means that, at each time step, the aim is to retrieve no more than one third of the discrepancy between the controlled value and the operating point.

6.3.2.4.2 Example 1: regulation of pump rotation speed

This example describes regulation of pump rotation speed to impose the flowrate Q_{oper} in the pump. The general formula is used:

$$\tau \frac{\partial\,(\delta\omega)}{\partial t} + \delta\omega = -\ G_p\ \ (Q(t) - Q_{oper})$$

where:

$\delta\omega$ = controlling value (pump rotation speed variation)

Q = controlled value (mass flowrate in the pump)

Q_{oper} = operating point

after discretisation:

$$\left(1+\frac{\tau}{dt}\right)\left(\omega\left(t+dt\right)-\omega\left(t\right)\right)=-\;G_p\;\left(Q\left(t\right)-Q_{oper}\right)$$

$$\boxed{\omega\left(t+dt\right)=\omega\left(t\right)-\frac{G_p}{1+\frac{\tau}{dt}}\;\left(Q(t)-Q_{oper}\right)}$$

dt = time step used for the calculation

How to define G_p?

$$\boxed{G_p=\frac{1}{3}\frac{\partial\omega}{\partial Q}}$$

To determine $\partial\omega/\partial Q$, the definition of pressure losses can be used:

- at the limits of the pump (through the pump),

$$\Delta P=\rho\;g\;H\;(q,\;\omega)\quad\text{(a)}$$

ρ =	density
g =	gravity
H =	pump head function of q and ω
q =	volumetric flowrate (m^3/s)
ω =	pump rotation speed (rd/s)

- at the limits of the pump (through the rest of the circuit).

$$\Delta P=K\,.\,\frac{q^2}{2\rho}\qquad\text{(b)}$$

K =	constant
ρ =	density
q =	volumetric flowrate

The constant K takes into account friction and singular pressure drops which exist in the circuit.

Combining these two relations ((a) and (b)):

$$\Delta P=\rho g\;H\;(q,\;\omega)=K\frac{q^2}{2\rho}$$

Then, by differentiation with respect to ω:

$$\rho g\left[\frac{\partial H}{\partial q}\,x\,\frac{\partial q}{\partial\omega}+\frac{\partial H}{\partial\omega}\right]=\frac{K}{\rho}\frac{\partial q}{\partial\omega}$$

Using the characteristics of the pump, it is then possible to determine $\partial\omega/\partial q$, $\partial\omega/\partial Q$ and hence G_p.

A good approximation of $\partial\omega/\partial q$ can be obtained very easily without considering the pump characteristics.

H is considered to have the following form:

$$H\,(q,\,\omega) = q^2\, f\left(\frac{q}{\omega}\right)$$

Consequently:

$$\rho g\, f\left(\frac{q}{\omega}\right) = \frac{K}{2\rho}$$

Neglecting the density variation, f(q/ω) is constant and q is proportional to ω.
So, if one operating point (q_0, ω_0) is known, it may be considered that $\partial q/\partial\omega = q_0/\omega_0$

We then need to define

$$\text{denom} = 1. + \tau\,/\,dt$$

this value is obtained by dividing the average velocity of a particle by the total length of the circuit,

$\tau = \dfrac{l}{v}$

where:
l = total length of the circuit,
v = average flow velocity in the circuit. This constant represents the average time used by a particle to complete one "lap" of the circuit.
Then the term "denom" is introduced in the formula of the coefficients used in the regulations:

$$\text{denom} = 1. + \tau\,/\,dt$$

τ = circuit constant,
dt = current time step of the calculation.

The "denom" value should be calculated at every time step because it is time-dependent.

The following relations are thus obtained:

$$\omega\,(t + dt) = \omega(t) - \text{coefpomp}\,(Q(t) - Q_{oper})$$

with

$$\text{coefpomp} = \frac{Gp}{1+\dfrac{\tau}{dt}}$$

CATHARE input deck application

To calculate the new rotation speed Ω(t + dt) called "vitrr1", we need to:

- use CCV to get values of (VALUE):
  - pump rotation speed ω(t) called " vitror",
  - flowrate in the axial element where the pump is located ("FROIDE1R') Q(t) called "qpnrr".

- impose as operating point the flowrate in the axial element where the pump is located, i.e. a fixed value for Q_{oper} called "q_{cons}",

- calculate the coefficient "coefpomp":

$$\text{coefpomp} = \frac{Gp}{1+\dfrac{\tau}{dt}} = \frac{Gp}{\text{denom}}$$

in our case, using the characteristics of the pump,

$$\frac{\Delta\varpi}{\Delta Q} \approx 0.03$$

Hence

$$Gp = \frac{1}{3}\frac{\Delta\varpi}{\Delta Q} = 0.01$$

After the new value of the rotation speed has been calculated, it is imposed using the directive PUMPMOD.

Then a transient time step is calculated, and the value of the pump rotation speed $\omega(t)$ is updated for the next time step

Example:

```
*Regulation general parameters
 OPTION ROCP  1.D+3;
 OPTION DTMAX 1.D+0;
  CSTC   = 10.     ;
  TIME   =  0.     ;
  DT     = 1.D-3   ;
  NPASTP = 2000    ;
*
*primary temp. regulation initialisation
[…]
*
*pump rotation speed regul. initialisation
 FLGPU = 0                          ;
 IPOI = VECTOR FROIDEI  9.05        ;
 IPOR = VECTOR FROIDE1R 9.05        ;
 QCONS  = 4.601D3                   ;
 DIFFDP = .20                       ;
 VITROI = VALUE PMPSPEED CARPOMI ;
 VITROR = VALUE PMPSPEED CARPOMR ;
*
*pressuriser regulation initialisation
[…]
*
*REGULATION LOOP
REPEAT BLREGU NPASTP;
*
 DENOM = 1 + (CSTC/DT)          ;
*
*primary temperature regulation
[…]
*
*2nd circuit pressure regul. (punctual
*secondary circuit modelled with SGCARACT)
[…]
*
*secondary flowrate regulation
[…]
*
*pumps rotation speed regulation
 COEFPO = 0.01 / DENOM              ;
 QPNRI = VALUE LIQFLOW FROIDEI  IPOI;
```

```
 DIFFDI = QPNRI - QCONS       ;
 VITRI1 = VITROI - (COEFPO*(QPNRI-QCONS));
 PUMPMOD CARPOMI ROTATION  RADSEC
RELATIVE  REALLIST 0.     1.E6
ROTV      REALLIST VITRI1 VITRI1;
*
 QPNRR = VALUE LIQFLOW FROIDE1R IPOR   ;
 DIFFDR = QPNRR - QCONS   ;
 VITRR1 = VITROR - (COEFPO*(QPNRR-QCONS));
 PUMPMOD CARPOMR ROTATION  RADSEC
RELATIVE  REALLIST 0.     1.E6
ROTV      REALLIST VITRR1 VITRR1 ;
*
*pressuriser regulation
[…]
*
*Time step achievement
 TRANSIENT CIRCTOT TIME DT       ;
 DT=NEWDT                        ;
 TIME=NEWTIME                    ;
*
*secondary circuit level regulation
[…]
*
*primary temp. regul. update and exit test
[…]
*
*pumps regul. update and exit test
 VITROI = VITRI1                 ;
 VITROR = VITRR1                 ;
 IF (ABSOLUTE(DIFFDI)<DIFFDP)    ;
   IF (ABSOLUTE(DIFFDR)<DIFFDP)  ;
     FLGPU = 1                   ;
   ELSE                          ;
     FLGPU = 0                   ;
   ENDIF                         ;
 ENDIF                           ;
*
*pressuriser regul. update and exit test
[…]
*
*
 END BLREGU                      ;
```

6.3.2.4.3 Example 2: Regulation of fouling

Fouling is regulated in order to impose the temperature "T_{oper}" in the primary circuit: the general formula is used:

$$\tau \frac{\partial A}{\partial t} + A = GE = G\,(F_{oper} - F)$$

where:
fouling = controlling value (A)

Fouling is called the multiplication factor of the thermal resistance inside the heat exchanger.

T1 = controlled value (F)
= temperature in the primary circuit

T_{oper} = operating point (F_{oper})

$$\tau \frac{\partial\, \text{fouling}}{\partial t} + \text{fouling} = -\,G_f\ = (T_1 - T_{oper}\,)$$

After discretisation

$$\left(1+\frac{\tau}{dt}\right)\left(\text{fouling}\left(t+dt\right)-\text{fouling}(t)\right)=\text{-}\,G_f\ (T_1-T_{oper})$$

$$\boxed{\text{fouling}(t+dt)=\text{fouling}(t)=-\frac{G_f}{1+\frac{\tau}{dt}}(T_1-T_{oper})}$$

dt = time step used
for the calculation

How to define G_f?

$$\boxed{G_f\approx\frac{1}{3}\frac{\partial\text{fouling}}{\partial T_1}}$$

To determine ∂fouling/∂T_1, the definition of the flux exchanged between primary and secondary circuit is used.

$$\varphi=\frac{T_1-T_2}{R}$$

φ = flux exchanged between primary and secondary (W)
T_1 = primary temperature (°C)
T_2 = secondary temperature (°C)
R = thermal resistance (°C/VV)

$$R = R_1 + f \text{ x } R_p + R_2$$

R_1 = thermal resistance between primary fluid and heat exchanger
R_2 = thermal resistance between secondary fluid and heat exchanger
R_p = thermal resistance inside the heat exchanger
f = fouling factor

$$R_p=\frac{e}{\lambda \text{ x } S_{exch}}$$

e = thickness of the heat exchanger (m)
λ = thermal conductivity of the heat exchanger (W/m/°C)
S_{exch} = exchange surface of the heat exchanger (m²)

Then:

$$\text{fouling}=\frac{\lambda\,.\,S_{exch}}{e}\left[\frac{T_1}{\varphi}-\frac{T_2}{\varphi}-R_1-R_2\right]$$

$$\frac{\partial\text{fouling}}{\partial T_1}=\frac{\lambda\,.\,S_{exch}}{e.\varphi}$$

$$\boxed{G_f=\frac{\lambda\,.\,S_{\ exch}}{3\ e.\varphi}}$$

Finally we obtain

$$\text{fouling } (t + dt) = \text{fouling} - \text{coefencr } (T_1 - T_{oper})$$

with

$$\text{coefencr} = \frac{G_t}{1+\frac{\tau}{dt}}$$

To calculate the new fouling value at (t + dt) called "foul1", we need to:

- use CCV to get the value of the temperature in the primary side T_1 called "tempn",
- impose as operating point the temperature in the primary side, i.e. a fixed value for T_{oper} called "t_{cons}",
- calculate the coefficient called "coefen":

$$\text{coefen} = \frac{G_f}{1+\frac{\tau}{dt}} = \frac{G_f}{\text{denom}}$$

in our case

$$\begin{cases} \lambda = 20\, W/m/°C \\ S_{exch} = 4800\, m^2 \\ e = 1.3\ 10^{-3}\, m \\ \varphi = 900\, MW \end{cases}$$

So

$$G_f = \frac{\lambda \,.\, S_{exch}}{3e \,.\, \varphi} = 2.7\ 10^{-2}$$

When the new fouling value is calculated, the WRITE directive is used to modify the fouling value in the heat exchanger. Then, a transient time step is calculated, and the fouling value at time t is updated for the next time step.

Example:

```
*Regulation general parameters
 OPTION ROCP  1.D+3;
 OPTION DTMAX 1.D+0;
  CSTC   = 10.     ;
  TIME   =  0.     ;
  DT     = 1.D-3   ;
  NPASTP = 2000    ;
*
*primary temp. regulation initialisation
 TCONS  = 322.80;
 DIFFTE = .10   ;
 ENCRAI = 1.0   ;
 ENCRAR = 1.0   ;
*
*pump rotation speed regul. initialisation
 […]
*
*pressuriser regulation initialisation
[…]
*
*REGULATION LOOP
REPEAT BLREGU NPASTP;
*
 DENOM = 1 + (CSTC/DT)         ;
*
*primary temperature regulation
 COEFEN = 2.72D-2 / DENOM        ;
 TEMPN = VALUE LIQTEMP PLENSUP INF;
 ENCR1I = ENCRAI-(COEFEN*(TEMPN-TCONS) );
 ENCR1R = ENCRAR-(COEFEN*(TEMPN-TCONS) );
 WRITE ENCR1I FOULING WALL8   ;
 WRITE ENCR1R FOULING WALLGR  ;
*
*2nd circuit pressure regul. (punctual

*secondary circuit modelled with SGCARACT)
[…]
*
```

```
*secondary flowrate regulation
[…]
*
*pumps rotation speed regulation
[…]
*
*pressuriser regulation
[…]
*
*Time step achievement
 TRANSIENT CIRCTOT TIME DT       ;
 DT=NEWDT                        ;
 TIME=NEWTIME                    ;
*
*secondary circuit level regulation
[…]
*
*primary temp. regul. update and exit test
 ENCRAI = ENCR1I                 ;
 ENCRAR = ENCR1R                 ;
 DIFFTI = TEMPN - TCONS          ;
 IF (ABSOLUTE(DIFFTI)<DIFFTE)    ;
   FLGTP = 1                     ;
 ELSE                            ;
   FLGTP = 0                     ;
 ENDIF                           ;
*
*pumps regul. update and exit test
[…]
*pressuriser regul. update and exit test
[…]
*
 END BLREGU                      ;
```

6.3.2.4.4 Example 3: Regulation of secondary side - equality between feedwater flowrate and steam flowrate

The feedwater flowrate is regulated to impose the steam flowrate in the cavity:

The general formula is used:

$$\tau \frac{\partial A}{\partial t} + A = GE = G\,(F_{oper} - F)$$

where:

δQ_{feed} = controlling value (A)
= feedwater flowrate variation

$O_{feed} - Q_{steam}$ = controlled value (F)
= discrepancy between feedwater flowrate and steam flowrate.

0 = operating point (F_{oper}), the discrepancy ($Q_{feed} - Q_{stearn}$) should be set to zero.

$$\tau \frac{\partial(\delta Q_{feed})}{\partial t} + \delta Q_{feed} = -Ga\,(Q_{feed} - Q_{steam})$$

after discretisation:

$$\left(1+\frac{\tau}{dt}\right)(Q_{feed}(t+dt)-Q_{feed}(t)) = -Ga\,(Q_{feed}-Q_{steam})$$

$$\boxed{Q_{feed}(t+dt) = Q_{feed}(t) \quad -\frac{Ga}{1+\frac{\tau}{dt}}(Q_{feed}\,(t)-Q_{steam})}$$

dt = time step used for the calculation

In this case, ∂A/∂F = 1, then

$$\boxed{Ga = \frac{1}{3}}$$

Finally, we obtain:

$$Q_{feed}\ (t+dt) = Q_{feed}(t) - \text{coefalim}\ \left(Q_{feed}(t) - Q_{steam}(t)\right)$$

with

$$\text{coefalim} = \frac{G_a}{1+\frac{\tau}{dt}}$$

To calculate the new feedwater flowrate q_{feed} (t+dt) called "qaln1r", we need to:

- use CCV to get values of:

- the liquid flowrate $Q_{feed}(t)$ called "qalnr" in the feeding tee element,
- the steam flowrate $Q_{steam}(t)$ called "qvapnr" in the outlet of the steam generator cavity.

- calculate the coefficient "coefalim"

$$\text{coefalim} = \frac{G_a}{1+\frac{\tau}{dt}} = \frac{G_a}{\text{denom}}$$

When the new value of the feedwater flowrate is calculated, the corresponding liquid velocity in the feeding tee element, called "vlxr" is calculated too. This velocity is a function of:

- the section of the junction, called "sectal",
- the liquid density at this point, called "roalr"; the value of the density is obtained using a CCV.

To impose this liquid velocity, the feeding boundary condition is modified by the BCMOD directive on the boundary condition used as inlet on the secondary side. Then a transient time step is calculated, and the value of the feedwater flowrate Q_{feed} (t) is updated for the next time step.

Example:

```
*  regulation block initialisation
*
OPTION   ROCP  1.D6  ;
OPTION  DTMAX  10.0D0;
OPTION  MAXREP 20    ;
*
NPASTP  = 1000              ;
NPASTP1 = 500               ;
NPASTP2 = NPASTP - NPASTP1  ;
DT      = 1.D-3             ;
CSTC    = 10.               ;
CSTCS   = 30.               ;
*
* fouling regulation parameter
[…]
*
* feedwater regulation parameters
SECAL=0.073                ;
ROALR= 842.70              ;
QALNR= VALUE LIQFLOW ENTREI;
*
* cavity level regul. parameters
[…]
*
*regulation block
REPEAT BLREGU NPASTP1        ;
*
DENOM = 1 + (CSTC / DT )     ;
COEFEN = 2.72D-2  /  DENOM   ;
*
*primary temperature and pressure regul.
[…]
*
*secondary pressure flowrate regul.
COEFALIM = 0.3 / DENOM                 ;
QVAPNR = VALUE GASFLOW TUYTEI IPWXCV;
QALNLR = QALNR + (XFFF *(QVAPNR - QALNR));
VLXR = QALNLR / (SECAL * ROALR)      ;
```

```
*
BCMOD   ENTREI BC3A
HL (REALLIST   945.E3 945.E3 945.E3 945.E3)
HG (REALLIST   -1.    -1.    -1.    -1.   )
ALFA(REALLIST  1.D-5  1.D-5  1.D-5  1.D-5 )
VL (REALLIST   VLXR   VLXR   VLXR   VLXR  )
VV (REALLIST   VLXR   VLXR   VLXR   VLXR  )
ABSTIME(REALLIST 0.0  1000.  2000.  9999.);
*
BCMOD    ENTRER     BC3A
HL (REALLIST  945.E3  945.E3 945.E3 945.E3)
HG (REALLIST  -1.     -1.    -1.    -1.   )
ALFA(REALLIST 1.D-5   1.D-5  1.D-5  1.D-5 )
VL  (REALLIST VLXR    VLXR   VLXR   VLXR  )
VV  (REALLIST VLXR    VLXR   VLXR   VLXR  )
ABSTIME(REALLIST 0.0  1000.  2000.  9999.);
*
*secondary circuit level regulation
[…]
*
*time step achievement
TRANSIENT CIRCTOT TEMPS DT  ;
DT    = NEWDT               ;
TEMPS = NEWTIME             ;
*
* regulation parameters actualisation
[…]
    QALNR = QALNLR              ;
*
END BLREGU  ;
*







*
```

6.3.2.4.5 Example 4: Regulation of secondary side - regulation of the level in the steam generator cavity

The level in the steam generator cavity is regulated by introducing an external source/sink in the cavity, the flowrate of which is calculated as a function of the imposed level.

The flowrate of the source is regulated to impose the level "xnivoc" in the cavity.

The general formula is used:

$$\tau \frac{\partial A}{\partial t} + A = GE = G\,(F_{oper} - F)$$

where:

q_{sou} = controlling value (A)
= flowrate of the external source

xnivo = controlled value (F)
= level in the cavity

xnivoc = operating point (F_{oper})

$$\tau \frac{\partial\, q_{sou}}{\partial t} + q_{sou} = -\,G_s\,(xnivo(t) - xnivoc)$$

In this case the integrating effect is neglected so:

$$\boxed{q_{sou}(t) = -\ G_s\ (\text{xnivo}(t) - \text{xnivoc})}$$

How to define G_s?

The sensitivity which G_s depends on is not very easy to estimate. So an approximated formula is used to calculate the source flowrate q_{sou}:

$$q_{sou} = \rho \frac{\vartheta}{\tau_s} = -\rho \frac{S}{\tau_s}(\text{xnivo} - \text{xnivoc})$$

It is assumed that, in order to fill the volume ϑ (of section S and height (xnivoc – xnivo)) a time τ_S is required.

S	= average section of the steam generator cavity
xnivo	= level in the SG cavity
xnivoc	= imposed level in the SG cavity
ρ	= density of the liquid
τ_s	= time required to reach the xnivoc level in the SG cavity.

$$\boxed{G_s = \frac{\rho S}{\tau_s}}$$

Finally we obtain

$$q_{sou}(t) = -\ \text{asou}\ (\text{xnivo}(t) - \text{xnivoc})$$

with

$$\text{asou} = G_s = \frac{\rho S}{\tau_s}$$

To calculate the new flowrate of the external source introduced in the SG cavity, we need to:

- create an external piqrev (source/sink) in the SG cavity using the PIQREV operator ("sousinr"), and impose its external conditions (injection conditions) by CCV. And then activate it with the OPEN directive.
- impose as operating point, the level in the cavity, i.e. a fixed value for xnivoc called "xnivoc".
- calculate the coefficient called "asou" which is a time-dependent coefficient:

- average section of the cavity,

- density of the liquid in the cavity,

- time required to reach the level xnivoc in the cavity.

In our case:

ρ	=	762 kg/m^3
S	=	15 m^2
τ_s	=	30 seconds (a value a little longer than the "integration time constant" used for the other regulations chosen).

When the new flowrate value of the external piqrev in the cavity is calculated, this value should be imposed, using CCV, as the total flowrate of the piqrev, and then a new transient time step is calculated.
When the regulation is stopped, the piqrev should be deactivated using the CLOSE directive.

Example:

```
*regulation block initialisation
*
OPTION   ROCP  1.D6  ;
OPTION  DTMAX  10.0D0;
OPTION  MAXREP 20    ;
*
NPASTP  = 1000           ;
NPASTP1 = 500            ;
NPASTP2 = NPASTP - NPASTP1 ;
DT      = 1.D-3          ;
CSTC    = 10.            ;
CSTCS   = 30.            ;
*
*fouling regulation parameter
[…]
*
* feedwater regulation parameters
[…]
*
*cavity level regul. parameters
XNIVOC = 2.4969        ;
ASOU = 15.*762./CSTCS  ;
WRITE 0. UNDERSAT SOUSINI;
WRITE 0. UNDERSAT SOUSINR;
WRITE  1.D-5  ALFAEXT  SOUSINI;
WRITE  1.D-5  ALFAEXT  SOUSINR;
WRITE 0. OVERHEAT SOUSINI;
WRITE 0. OVERHEAT SOUSINR;
OPEN SOUSINI           ;
OPEN SOUSINR           ;
*
[...]
*
*regulation block
```

```
REPEAT BLREGU NPASTP1        ;
*
DENOM = 1 + (CSTC / DT )     ;
COEFEN = 2.72D-2  /  DENOM   ;
*
*primary temperature and pressure regul.
[...]
*
*secondary pressure flowrate regul.
[...]
*
*secondary circuit level regulation
XNIVOI=VALUE LEVEL CAVITEI  ;
QSOUR=ASOU*(XNIVOC-XNIVOI);
WRITE QSOUR TOTFLOW SOUSINI   ;
*
XNIVOR=VALUE LEVEL CAVITER  ;
QSOUR=ASOU*(XNIVOC-XNIVOR);
WRITE QSOUR TOTFLOW SOUSINR   ;
*
*time step achievement
TRANSIENT CIRCTOT TEMPS DT      ;
DT    = NEWDT               ;
TEMPS = NEWTIME             ;
*
* regulation parameters actualisation
[…]
*
END BLREGU  ;
*
CLOSE SOUSINI             ;
CLOSE SOUSINR             ;
*

*
```

6.3.2.5 Transient scenario

This part of the executable block is used to represent the actual reactor or experiment transients.

Available directives may be sorted into different categories:

6.3.2.5.1 Initialisation

The INIBIL directive is used to reinitialise inventories in a circuit during a calculation: water mass, non-condensable gas mass and energy or boron mass and activity. This directive should be used after imposing values which modify balances (ex: WRITE directive).
The directive INIBIL must be applied to a CIRCUIT, ZONE or REACTOR.

The INIBORA directive should be used to initialise or reinitialise the concentration of a radio-chemical component in a circuit or in a zone. It can also be used to reinitialise these data during a calculation, in which case, the unmentioned variables are not modified (it then becomes a way of stopping the radio-chemical transport calculation). With the INIBORA directive, data can be given to describe a clad rupture. The clad rupture will be simulated using the WRIBA directive.

The RESETIME directive is used to reinitialise the CATHARE calculation time as well as the time input deck local variable to zero.

The REINIT directive is used to reinitialise a DEADZONE (PERMINIT) physical state. It should be used after a REALVO or REALAX directive which defines the new thermal-hydraulic conditions in the dead zone. The user should call the INIBIL directive to reinitialise mass and energy balances after using REINIT.

Remark: REALVO and REALAX directives have to be used at the same points as they were used for the PERMINIT phase.

6.3.2.5.2 Specific models launching

GOBORA is used to launch the radio and chemical transport calculation.

GOFUEL is used to launch the thermo-mechanical calculation for fuel walls (fuel deformation and oxidation, hydrogen release, clad rupture model). Not using it means that only thermal radial conduction is calculated.

GONEUT is used to launch the point kinetics model calculation. Not using it means that no reactivity calculation is made.

The REFLOOD directive is used to activate the rewetting and reflooding model.

GODIFF and GOTURB are used to launch respectively diffusion and turbulent diffusion models for a THREED element.

GOBALLON is used to launch the fuel model ballooning.

6.3.2.5.3 Other trips

Trips available for every kind of element are described in chapter 5 and summarised in Appendix 14.

Many gadgets have to be activated before being taken into account in the calculation or may be deactivated during the calculation. Refer to Appendix 14 for a summary.

More generally, the user can define signals in the input deck to launch special instructions depending on any chosen parameter (calculation time, a specific physical state in an element, etc.).
Example: opening the safety injections 10 seconds after opening the accumulator.
Refer to [DOC2] HOW TO PROGRAM SIGNALS? for more information.

6.3.3 User-programming CATHARE facilities

6.3.3.1 User common

The USER common allows the user to store real numbers (XUSER table) or integers (KUSER table) or character*8 (CUSER).

The user common values are saved (for later restore) like the other state variables when the SAVE directive is used and at the frequency chosen for the REACTOR module for purpose of post-processing of the results (RESULT directive).
Remark: a common user error is to give the CIRCUIT name instead of the REACTOR name in the RESULT directive.

To limit the standard result file size, these common values are not automatically saved (with respect to result post-processing). The user has to activate it at the beginning of the calculation:

- either using the WRITE directive in the input deck,

Example: to save real user variables 1 to 10 values in result file
WRITE XUSER 1 10;

- or in UTILx.f files (see §6.3.3.3).

For example, to use variables 1 to 10 values:

```
DO i= 1, 10
        XUSER(i) = my_value(i) * 2.D0
        CUSER(i) = ‘SET’
ENDDO
```

They have to be used like any of the CCV variables in the input deck:

Filling the common	:	WRITE xret XUSER i1;
		WRITE iret KUSER i1;
		WRITE cret CUSER i3 ;

Reading the values	:	x = VALUE XUSER i1;
		i = VALUE KUSER i2;
		C= VALUE CUSER i3 ;

6.3.3.2 User material properties

Refer to §5.4 and [DOC18] for information on built-in material properties in CATHARE.

If the user wants to use a new material not defined in the CATHARE material library, the material takes the name XXXXXXi and its properties are written by the user in subroutine FWMAXX (or YYYYYYij defined in FWMAYY). In FWMAXX, the user may describe up to six user materials. If more are needed, FWMAYY subroutine allows up to 50 user materials to be described.

These subroutines (FWMAXX or FWMAYY) describe the physical properties (density, heat capacity and conductivity).

A sample is given in CATHARE sources in ~/sources/walcom/walmat as follows:

```
SUBROUTINE FWMAXX (NWMAIL,INDMAT,TWMOY
     1  ,RO,CP,XLAMB,ROCP,*)
C
C_BEGIN_DOCUMENT
      IMPLICIT NONE
90000  FORMAT('  @(#)  FWMAXX.f  +_v25_+  2.1
11/21/03 ')
C
C_DESCRIPTION
C
C*****************************************
C    FWMAXX.F FILE: NUMBERS 28 to 33
C
C   COMPUTATION   UNKNOWN   MATERIAL   PHYSICAL
PROPERTIES:
C
C  THESE COEFFICIENTS ARE AVAIBLE FOR
C        A VARIABLE IN CELSIUS:
C
C*****************************************
C
C_GLOBAL
C
       INCLUDE 'GLOBAL.H'
       INCLUDE 'IOUNIT.H'
C
C_INPUT
C
      INTEGER  NWMAIL,INDMAT(NWMAIL)
      DOUBLE PRECISION TWMOY(NWMAIL)
C
C_OUTPUT
C
      DOUBLE PRECISION RO(NWMAIL),CP(NWMAIL),
     1  XLAMB(NWMAIL),ROCP(NWMAIL),IBID
C
C
C_MODIFIED
C
C_LOCAL
C
      INTEGER I
C---------------------------------------------
----------
C
C-->  LAMBDA AND ROCP IN FMATX
C--> MATERIAU INCONNU XXX  (UNKNOWN MATERIAL)
C---------------------------------------------
----------
C
C   CONSTI  IS  SENT  TO  FWMAXX  TO  IDENTIFY  THE
MATERIAL
C       IN   CASE   OF   SEVERAL   USER-SUPPLIED
MATERIALS
C
C
C_POLYNOMIAL
C
       INCLUDE 'UPOLY.H'
C
C
C TO MODIFY (BEGINNING)
C  XXXXXXXXXXXXXXXX
      IBID=1
      IF (IBID.EQ.1) THEN
       WRITE(6,90000)
       WRITE(6,1000)
1000     FORMAT  (1X  ,/,'UTILIZATION  OF  USER
MATERIAL'
     1   , '8 to 33 '
```

```
     2   ,'AND THE FWMAXX SUBROUTINE '
     3   ,'HAS NOT BEEN MODIFIED '6
         GOTO 9999
      ENDIF
C  XXXXXXXXXXXXXXXX
C TO MODIFY (END)
C--------------------------------
C    CALCULATION OF THE PROPERTIES
C--------------------------------
C
      DO 100 I=1,NWMAIL
C
        IF (INDMAT(I).EQ.28) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.29) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.30) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.31) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.32) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.33) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ENDIF
C
100   CONTINUE
C
8888   RETURN
9999   WRITE (IOSTER,90000)
       RETURN 1
       END
```

First, the line IBID = 1 should be changed to IBID = 0 to avoid CATHARE execution stopping with the following message:

```
UTILIZATION OF USER MATERIAL 28 to 33
 AND THE FWMAXX SUBROUTINE HAS NOT BEEN MODIFIED
 >>>>> FWMAXX <<<<<
```

Then appropriate user material properties should be coded with respect to the actual use of the input deck material.

In this file, i represents an axial wall mesh number and INDMAT(i) gives the kind of material for the mesh.

Six different INDMAT variables are listed to define up to six possible different materials. Thus XXXXXXX1 should be defined in case of INDMAT(i) = 28, XXXXXXX2 in case of INDMAT(i) = 29,…, XXXXXXX6 in case of INDMAT(i) = 33.

Subroutine physical input data are the average temperature in the cell I, TWMOY(i). The user then only has to fill in the blanks with the desired correlations for conductivity (λ), density (ρ) and heat capacity (Cp) as a function of this temperature.

Example:

```
      DO 100 I=1,NWMAIL
C
        IF (INDMAT(I).EQ.28) THEN
C
           XLAMB(I) =.144D0
     1                                         *
((TWMOY(I)+273.15D0)/273.15D0)**.7D0
C
           CP(I)    = 5.194D3
C
           RO(I)                               =
187.64D0/(TWMOY(I)+273.15D0)
C
           ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.29) THEN
C
             XLAMB(I)  = 0.0166D0 *
     1                    (   (TWMOY(I)   +
273.15D0)/273.15D0 )**0.7517D0
C
             CP(I) = 520.3D0
C
             RO(I)       =       1873.54D0/(
TWMOY(I)+273.15D0 )
C
           ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.30) THEN
C
C          XLAMB(I) =
```

```
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.31) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.32) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ELSEIF (INDMAT(I).EQ.33) THEN
C
C          XLAMB(I) =
C          CP(I)    =
C          RO(I)    =
C
C          ROCP(I)  = RO(I)*CP(I)
C
        ENDIF
C
```

6.3.3.3 User directive UTILx

The user can call personal directives with the predefined names: UTIL1, UTIL2, etc., UTIL10.
The corresponding Fortran subroutines have to be defined (written in Fortran) by the user and be located in the current CATHARE executable working directory.

The UTILx directives are called without operands in the input deck:
Instruction “UTIL1;” corresponds to “CALL UTIL1”. No formal arguments are allowed.

For the inexperienced user, it is strongly advised to use only these UTILx directives as personal post or pre-processing facilities with no results sent back to CATHARE memory (i.e. space defined in FAST.H) (except by using the common USER).

Refer to Example 1: Reading experimental values and for examples of use.

If the user needs to access CATHARE internal data, it is strongly recommended to use VALUE, VALUFEAU, VALBA (and WRITE to modify CATHARE internal data) Fortran interfaces.

However, any CATHARE directive can be used in a UTILx subroutine (if the user knows a little about CATHARE data management) and the corresponding Fortran call is described in [DOC3].

6.3.3.3.1 Example 1: Reading experimental values

An UTIL directive can read the data on a file and write them to the user common for subsequent use by the BCMOD directive in the input deck.
In the following example, experimental data are stored in the file EXPDATA (FORT19):

```
TITLE of the experiment
exp_time1
α(exp_time1) QL(exp_time1) QG(exp_time1) TL(exp_time1) TG(exp_time1)
exp_time2
α(exp_time2) QL(exp_time2) QG(exp_time2) TL(exp_time2) TG(exp_time2)
…
exp_timeN
α(exp_timeN) QL(exp_timeN) QG(exp_timeN) TL(exp_timeN) TG(exp_timeN)
```

User CCV XUSER(1),…,XUSER(6) are used to impose BC3E boundary conditions values which result from a data transfer from UTIL1 to the input data deck. They are calculated in the UTIL1 subroutine (simple interpolation) from the experimental data read in the FORT19 file.

```
      SUBROUTINE UTIL1
      IMPLICIT NONE
C-------------------------------------------
C     READ AND DEFINE BOUNDARY CONDITIONS
C-------------------------------------------
C
C XUSER (1): BC INLET VOID FRACTION
C XUSER (2): BC INLET GAZ FLOWRATE
C XUSER (3): BC INLET LIQUID FLOWRATE
C XUSER (4): BC INLET LIQUID TEMPERATURE
C XUSER (5): BC INLET GAZ TEMPERUTRE
C XUSER (6): CATHARE current time
C
C  KUSER (1):  READING FLAG
C
C last read values:
C XUSER (11): experiment VOID FRACTION
C XUSER (12): experiment GAZ FLOWRATE
C XUSER (13): experiment LIQUID FLOWRATE
C XUSER (14): experiment LIQ. TEMPERATURE
C XUSER (15): experiment GAZ TEMPERATURE
C XUSER (16): experiment time
C
C User common included
      INCLUDE  'USER.H'
C global variables
      DOUBLE PRECISION ALFAIN,QGIN,QLIN
     1         ,TLIN,TGIN,TIMECAT,DTCAT
      DOUBLE PRECISION ALFAEXP,QGEXP,QLEXP
     1        ,TLEXP,TGEXP,TIMEEXP
      INTEGER IOPEN, ILINE
      CHARACTER*60 TITLE
C
C Local variables
      DOUBLE PRECISION TIME1,ALFA1,QL1,QG1
     1        ,TL1,TG1
      DOUBLE PRECISION TIME2,ALFA2,QL2,QG2
     1        ,TL2,TG2
      DOUBLE PRECISION DTX
C
      EQUIVALENCE (XUSER(1),ALFAIN)
      EQUIVALENCE (XUSER(2),QGIN)
      EQUIVALENCE (XUSER(3),QLIN)
      EQUIVALENCE (XUSER(4),TLIN)
      EQUIVALENCE (XUSER(5),TGIN)
      EQUIVALENCE (XUSER(6),TIMECAT)
C
      EQUIVALENCE (XUSER(11),ALFAEXP)
      EQUIVALENCE (XUSER(12),QGEXP)
      EQUIVALENCE (XUSER(13),QLEXP)
      EQUIVALENCE (XUSER(14),TLEXP)
      EQUIVALENCE (XUSER(15),TGEXP)
      EQUIVALENCE (XUSER(16),TIMEEXP)
C
      EQUIVALENCE (KUSER(1),IOPEN)
C
C INITIALISATION
C
      IF (IOPEN.EQ.0)  THEN
         OPEN (UNIT=19,FILE='EXPDATA'
     1                ,FORM='FORMATTED')
         READ (19) TITLE
         IOPEN = 1
         TIMEEXP = 0.0
      ENDIF
```

```
C
C restore saved values - last read values
C
      TIME1 = TIMEEXP
      ALFA1 = ALFAEXP
      QG1 = QGEXP
      QL1 = QLEXP
      TG1 = TGEXP
      TL1 = TLEXP
C
C searching TIMECAT in EXPDATA
C
 1    CONTINUE
      READ (19) TIME2
      READ (19) ALFA2, QL2, QG2, TL2, TG2
C
      IF (TIME2.LT.TIMECAT) THEN
C
       TIME1= TIME2
       ALFA1= ALFA2
       QG1  = QG2
       QL1  = QL2
       TG1  = TG2
       TL1  = TL2
       GOTO 1
C
      ELSE
C
C first reading treatment
C
        IF (TIMEEXP.EQ.0.0) THEN
          ALFAIN= ALFA2
          QGIN  = QG2
          QLIN  = QL2
          TGIN  = TG2
          TLIN  = TL2
          GOTO 2
        ENDIF
C
C time1 < cathare time < time2
C
C CATHARE inlet condition calculation
C exp data interpolation
C
       DTX  = (TIMECAT-TIME1)/(TIME2-TIME1)
       ALFAIN = ALFA1 + (ALFA2-ALFA1)*DTX
       QLIN = QL1 + (QL2 - QL1) * DTX
       QGIN = QG1 + (QG2 - QG1) * DTX
       TLIN = TL1 + (TL2 - TL1) * DTX
       TGIN = TG1 + (TG2 - TG1) * DTX
C
      ENDIF
C
C storage of the last read values
C
 2    CONTINUE
      TIMEEXP= TIME2
      ALFAEXP= ALFA2
      QGEXP  = QG2
      QLEXP  = QL2
      TGEXP  = TG2
      TLEXP  = TL2
C
      RETURN
      END
```

6.3.3.3.2 Example 2: CATHARE result post-processing

In this example, UTIL3 is used to read CATHARE calculated variables by means of the corresponding VALUE Fortran call and to store them in an external file (FORT23).

```
SUBROUTINE UTIL3 (*)
      IMPLICIT NONE
C
C_STRUCTURES
      INCLUDE 'FAST.H'
      INCLUDE 'GLOBAL.H'
      INCLUDE 'USER.H'
C
CINPUT
      INTEGER LURESU,IVAR, MVAR, KDEBUT
      PARAMETER(MVAR=20)
      COMMON/IUTIL3/KDEBUT,LURESU
C
      CHARACTER*10    NOMVAR(MVAR)
C
      INTEGER NSTEP
      DOUBLE PRECISION VAR (MVAR)
      DOUBLE PRECISION  VALUE
C
C RETRIEVE the interesting VALUES
      NOMVAR( 1) = '       NPAS'        NSTEP
= KUSER(1)
C
      NOMVAR( 2) = '      TEMPS'       VAR   (
2) = XUSER(1)
C
      NOMVAR( 3) = '     PRESP’
      VAR    (  3)  =  VALUE('PRESSURE','PRESSU
',2,0,IVSTAT) * 1.D-5
      IF (IVSTAT.NE.0) GOTO 9999
C
      NOMVAR( 4) = '      NIVP‘
      VAR    (  4)  =  VALUE('LEVEL    ','PRESSU
',0,0,IVSTAT)
      IF (IVSTAT.NE.0) GOTO 9999C
C  OPENING FILE MANAGEMENT - ONCE
      IF (KDEBUT.NE.1) THEN
         KDEBUT=1
         LURESU=23
         OPEN
(UNIT=LURESU,NAME='FORT22',FORMAT=’FORMATTED’
)
         WRITE(LURESU,1000)           (NOMVAR(
IVAR),IVAR=1, 4)
      ENDIF
C
C WRITING IN FORT23 FILE
      WRITE(LURESU,2000)          NSTEP,(VAR(
IVAR),IVAR=2, 4)
C
8888  RETURN
1000  FORMAT(20A10//)
2000  FORMAT(I10,F10.3,E10.3,20F10.3)
9999  WRITE(6,*)’ERROR IN UTIL1’
      RETURN  1
      END
```

6.3.4 Common user command block errors

6.3.4.1 CATHARE controlled error

CATHARE execution, e.g. the second calculation step (see §4.2 to understand the different steps of the CATHARE calculation), is a succession of user-requested actions.

Checks are made on all actions that may result in calculation errors during execution of the command. When this kind of error occurs, CATHARE execution stops but all information needed to restart is written in the V25_1.RESTART file (label 999). This avoids the loss of calculation results (and making you happy in the event of a long calculation!) (see [DOC2] HOW to PERFORM a RESTART?).

6.3.4.1.1 Key word control

The use of an unknown variable (not listed in the DICO file) in any directive of the command block will stop the calculation. Most of the time this happens when the WRITE or VALUE directives are used.

6.3.4.2 Other common user errors with command block

Many errors cannot be checked by the CATHARE code, for example setting an incoherent physical variable or action on a wrong element.

6.3.4.2.1 Missing initialisations

Depending on the kind of computer used, using non-initialised variables may or may not stop the calculation. If not, the best that can be expected is that the calculation results are sufficiently wrong to cause a CATHARE stop (bad convergence and too small a time step), etc.

Incorrect non-condensable / radio-chemical mass fraction / concentration definition (BCMOD or ASSIGN directive for BCONDIT and RUPTURE elements).

An input deck user variable not initialised before its use (warning messages are printed in a standard print out).

Errors occur when activating a gadget without having defined all the variables needed (opening a SOURCE without having defined the injected flowrate or its temperature, for example).

Most of the time, this happens for objects connecting the hydraulic circuit with the outside and when certain external conditions have not been defined: for reversal PIQxxx gadgets (a PIQBREK which has to overcome a flowrate return, a PIQREV where the injection becomes an extraction, etc.) or for RUPTURE elements.

6.3.4.2.2 WRITE non-accessible variables

In the command block, all existing CCVs (CATHARE Calculation Variables) can in fact be passed to the VALUE and WRITE directives. It is the user's responsibility to make proper use of the CCVs following the Dictionary ([DOC3]) specifications.

6.3.4.2.3 Time-dependent laws

Take care when using ABSOLUTE or RELATIVE key word.

6.3.4.2.4 Error in the time sequence

A user who is not well-informed is advised to use the following time sequence without any inserted instruction:

```
TIME = TIME + DT;
TRANSIENT reactor_name TIME DT;
DT = NEWDT;
TIME = NEWTIME;
```

6.3.4.2.5 Common restart errors

All data needed (including user input deck variables) are automatically saved when using the SAVE directive. The user should then suppress in the restart input deck any spurious initialisation from the initial input deck.

Take care not to repeat any instruction block that was already completed during the first calculation.

Refer to [DOC2] HOW to PERFORM a RESTART?

6.4 Post-processing data specification

In CATHARE code, a post-processing tool is available, also called postpro executable. This program processes the result binary file FORT21 following user postpro input deck directives and creates a formatted file FORT07 suitable for simple graphic representations (XY plots).

The default format of this file is not a column format and thus cannot be easily visualised. But an option (XMGROP) can be used in the user postpro input deck to generate two column formatted files which can be read by most standard software packages.

Refer to [DOC4] for an explanation on:

- How to create a user postpro input deck?
- What are the post-processing accessible variables for every type of CATHARE element?
- What is the format of the binary result file FORT21 and what is the information stored in this file?

Common user postpro input deck errors are:

- Error in files: no result file or wrong name for the graphic input deck
- Wrong element for the required variable (confusion between the name of the FUELCHAR object and the name of the assembly of fuel walls for example).

7 THE FUEL STAND-ALONE COMPUTATION

Refer to [DOC2] for more information on WHY-WHEN TO USE THE STAND-ALONE FUEL?

The stand-alone fuel module developed in CATHARE2 V2.5 provides the major possibilities available in CATHARE2 V1.3L version. However stand-alone fuel calculations simultaneously with a standard CATHARE calculation (option 3 from V1.3L) and a calculation with reflooding from an external fluid file (option 1 with reflooding) have been dropped because of insufficient use.

Since CATHARE2 V1.3L, some new features have been added:

- printouts of the hydraulic file saved during the CATHARE calculation (option 2 cf §7.2);
- possibility of imposing a maximum and a minimum time step with options DTMAX and DTMIN as in a standard calculation;
- possibility of performing a stand-alone fuel calculation from the hydraulics of a THREED element.

7.1 Principle of the stand-alone fuel module

The aim of the stand-alone fuel module is to carry out fuel calculations rapidly to provide information on the behaviour of specific fuel rods. In these calculations it is assumed that the fuel rods do not influence significantly the hydraulic conditions to which they are subjected. These rods are named "fuel rods with imposed hydraulic conditions".

This assumption implies that the fluid balance equations are not calculated and that, the wall-fluid heat flux terms (Φ_{wf}) are written with fixed fluid state variables read in the fluid file and corresponding to the fluid state at time t+Δt. Consequently, at iteration k+1, the equation used to calculate the wall-fluid fluxes at time t+Δt becomes:

$$\Phi_{wf}^{k+1} = \Phi_{wf}^{k}\left(F(t+\Delta t), T_w^k\right) + \left(\frac{\partial \Phi_{wf}^{k}}{\partial T_w}\right)\delta T_w \qquad (1)$$

instead of the standard CATHARE wall-fluid coupling scheme:

$$\Phi_{wf}^{k+1} = \Phi_{wf}^{k}\left(F^k, T_w^k\right) + \left(\frac{\partial \Phi_{wf}^{k}}{\partial T_w}\right)\delta T_w + \left(\frac{\partial \Phi_{wf}^{k}}{\partial F}\right)\delta F \qquad (2)$$

where:

δT_w = fuel variable increment

δF = fluid variable increments

The simplified equation (1) is solved by the NEWTON-RAPHSON iterative method.

7.2 General description

Two different methods are available for reading the imposed hydraulic conditions. They correspond to options 1 and 2 of the CATHARE2 V1.3L version.

Option 1: The fluid data are read from an **external file** obtained by another code or defined by the user. This file contains a simplified fluid data set which can take four different forms allowing four different types of boundary condition;

Option 2: The fluid data are read from a file created by a previous CATHARE2 calculation for which some selected hydraulics (AXIAL element or 1D channel of a THREED element) have been saved using the directive STOREHYD.

7.2.1 First method for a stand-alone fuel calculation (option 1)

7.2.1.1 Description of the different boundary conditions

For option 1, there are four possible ways to impose the fluid hydraulic conditions to a considered fuel rod:

ITYCL = 1 : pressure, fluid temperature, cladding-fluid exchange coefficient are imposed;

ITYCL = 2 : pressure and cladding-fluid flux are imposed;

ITYCL = 4 : fluid properties P, TL, TV, ALFA, VL, VV, Xk (k∈[1;4]) and hydraulic diameter DH are imposed. As in the previous case the calculation of fluid properties and wall-fluid fluxes becomes possible.

7.2.1.2 Description of the external fluid file

Since CATHARE2 V1.3L version, the structure of the file has been entirely respected in order to make it possible to use old external fluid files. This explains why some key words are compulsory but not used by the code.
The key words and values of a given record can be positioned on several lines but with a fixed order. Note that specifying the key word REFLOOD instead of NOREFLOO has no effect on the calculation because **the reflooding is not treated in option 1**.

7.2.1.2.1 External fluid file header

The external fluid file header consists of 3 different records:

record 1	"Title"	

record 2	key word1 key word2 keywork3 nmesh ntime	

record 3	hd_1 hd_2 ... hd_{nmesh} nothing	- if key word2 is VARIABHD - in other cases

Title : title between quotes (maximum length 80 characters)
key word1 : **REFLOOD** or **NOREFLOO** (REFLOOD accepted but not used)
key word2 : **EXCHCOEF** - for the first type of hydraulic conditions (ITYCL = 1)
FLUX - for ITYCL = 2
VARIABHD - for ITYCL = 4.
key word3 : **AXIFLUI** - not used but compulsory
nmesh : number of axial meshes in the data file
ntime : number of time steps
hd_1 hd_2 ... hd_{nmesh} : nmesh real values to define the hydraulic diameters (only for ITYCL = 4)

7.2.1.2.2 Recording at each time step

The file header is followed by **ntime** sets of values. Each set represents the hydraulic conditions at a considered time. The fourth record defining the time is common to all types of boundary condition. The fifth record depends on ITYCL, and then three different forms of record 5 are expected.

- Time record common for all cases:

record 4 time fpower Tf_{bot} Tf_{top}

- Record of imposed fluid conditions:

record 5a (ITYCL=1) [Tf_i P_i H_i $]_{i=1...nmesh}$ itime

record 5b (ITYCL=2) [P_i φ_i $]_{i=1...nmesh}$ itime

record 5c (ITYCL= 4) [P_i TL_i Tv_i $ALFA_i$ VL_i VV_i $Xk_i$$]_{i=1...nmesh}$ itime

time : time (s)
fpower : fraction of power - not used (see STR/LML-EM/91-61, p258 §V.4.3.2 i))
Tf_{bot} : inlet fluid temperature (°C) - not used (no scalar nodes at junctions)
Tf_{top} : outlet fluid temperature (°C) - not used (no scalar nodes at junctions)
Tf_i : fluid temperature in mesh i (°C)
H_i : thermal exchange coefficient in mesh i (W/m^2/°C)
φ_i : wall-fluid thermal flux in mesh i (W/m^2)
P_i : fluid pressure in mesh i (**CAUTION** of the unities: bars if $P_i < 1000$ or Pascal if $P_i \geq 1000$ – same convention as in V1.3)
TL_i TV_i : liquid and gas temperatures in mesh i (°C)
$ALFA_i$: void fraction in mesh i
VL_i VV_i : average liquid and gas velocities in mesh i (m/s)
Xk_i : fraction of non-condensable gas k ($k \in [1;4]$) in mesh i
itime : any integer value

7.2.1.3 Data set for a stand-alone fuel calculation

The main points of the data set for a stand-alone fuel calculation are the following:

- definition of the hydraulic channels (HYDIMP operator),
- definition of fuel rods (FUEL, FUELCHAR operators and INTEGRATE directive),
- definition of the circuit (CATAFUEL operator),
- definition of the command block (READHEXT, STDYFUEL and TRANFUEL directives).

7.2.1.3.1 Definition of imposed hydraulic elements (hydraulic channel)

The HYDIMP element has the same function as the AXIAL element in a standard CATHARE calculation except that its hydraulic characteristics are read in a file created by the user. The HYDIMP element is necessary to associate a fuel rod (a FUELCHAR element) to the hydraulics stored in the external fluid file.
As values in the external hydraulic file are given for a given mesh network, this network should also be defined (XAXIS, SEGMENT) and associated with the HYDIMP element (MESH).

Example 1: Definition of HYDIMP element

```
BEGIN DATA;
...
EDGROD    =      HYDIMP COEURMOY;

PCO15 = XAXIS 0.D0;  PCO18 = XAXIS 0.490D0;
MEDGROD = PCO15 SEGMENT 4 PCO18 COS 1.D0;
MESH EDGROD MEDGROD;
…
```

7.2.1.3.2 Definition of fuel rods

After the definition of the HYDIMP element, the definition of the fuel rod is made as usual except the name of the AXIAL element which is replaced by the HYDIMP element name.

Example 2: Definition of a fuel rod

```
BEGIN DATA;
...
FUEL2C1          =      FUEL EDGROD INTERNAL
                        SEGMENT  PCO15 PCO18 ISO 1 DIAM 0.D0 8.2D-3
                        UO2      IRRAFUEL
                        WZUO     1.0D0      LAW  LAWRAD   GAP  8.D-5
                        OXIDE               EO   1.D-8    EOE  1.D-8
                        CLADDING      ZRCEA          RM   4.465D-3;
CARCBCO2=       FUELCHAR EDGROD
                        GAP      DELTFUEL 1.0D-8   GAPP   0.1D0
                        CRAD     0.06D-3
                        FUELP    INTERNAL
                             POWNEUT  0.     POWRESI   0.
                        LAWNEUT  LAWPOW     LAWRESI   LAWPOW
                GASVOL   VOIDV 0.   EXPANV    1.4D+7
                        BPLUGV   0.         CRACKV    0.
                        POROSV   0.         TPLUGV    0.
                        VOLFRAC  1.D0       POROS          3.96D-2
                        XNEUT         1.0;

INTEGRATE FUEL2C1 NUMBER 1 CARCBCO2;
...
END DATA;
```

7.2.1.3.3 Definition of the CATAFUEL element

The CATAFUEL element is defined by the CATAFUEL operator. The name of the HYDIMP element must be specified as well as the unit number of the external hydraulic file.

Example 3: Definition of the CATAFUEL element

```
BEGIN DATA;
...
CATA       =      CATAFUEL EDGROD FILE 63;
...
END DATA;
```

7.2.1.4 Definition of the command block

The command block must be organised as follows:

- first reading of the external fluid file (READHEXT) before the stand-alone fuel steady state calculation (STDYFUEL);
- a loop for the stabilised transient if necessary (READHEXT, TRANFUEL);
- launching fuel thermo-mechanical computation (GOFUEL);
- a loop for stand-alone fuel transient steps (READHEXT, TRANFUEL).

In example 4, the BLOC1 loop is defined in case of stabilised transient values stored (with time equal to zero) in the external fluid file. This loop is skipped when the time read by READHEXT is greater than 0. The BLOC2 loop will perform the transient steps. The user must be careful with successive calls of READHEXT and TRANFUEL to avoid time step skipping.

The time and the time step values are read from the external fluid file by the READHEXT directive and transferred to the TIME and DT variables declared in the data set. As in a standard calculation, TIME=TOLD+DT. Then, the TRANFUEL directive tries to reach the time proposed by the fluid file. If a fuel calculation convergence problem occurs or if a maximum time step is set to a value lower than DT (by OPTION DTMAX) then a sub-cycling is performed in TRANFUEL in order to reach the time t=TIME in several steps.

Therefore the user must not use the NEWTIME and NEWDT operators to update TIME and DT.

Example 4: Definition of the *command block*

```
READHEXT        CATA TIME DT;
STDYFUEL CATA;
***** STABILISED TRANSIENT **********************************************
READHEXT        CATA TIME DT;
REPEAT BLOC1 10;
        IF (TIME > 0.0); QUIT BLOC1; ENDIF;
        TRANFUEL CATA TIME DT;
        READHEXT        CATA TIME DT;
END BLOC1;
***** TRANSIENT **********************************************************
GOFUEL    CATA;
REPEAT BLOC2 NPAS;
        TRANFUEL CATA TIME DT;
        READHEXT        CATA TIME DT;
END BLOC2;
```

7.2.1.5 Example of input data file

Example 5: A complete input data set for a stand-alone fuel calculation with an external fluid file

```
*TCS EDGAR ESSAIS RAMPE 461A
*******************************
BEGIN DATA;
***************************************************************************
* DEFINITION OF THE HYDIMP ELEMENT
***************************************************************************
PCO15          =      XAXIS 0.D0;PCO18   =  XAXIS 0.490D0;
MEDGROD =      PCO15 SEGMENT 4 PCO18 COS 1.;
EDGROD         =      HYDIMP COEURMOY;
MESH EDGROD MEDGROD;
***************************************************************************
* DEFINITION OF THE FUELCHAR ELEMENT
***************************************************************************
FRUOE          =      0.1D-3;       DEC   =   9.51D-3;THC   =   0.569D-3;
THG        =      ( DEC * 0.5 ) - ( FRUOE + THC );
RMC        =      ( DEC - THC ) * 0.5;
FDUOE          =      FRUOE * 2.;
LAWRAD     =      LAW  'RADIUS'    'POWER'  0.  1.  FRUOE        1.;
LAWPOW     =      LAW  'TIME'               'POWER'  0.  1.  1.D+10        1.;

FUEL2C1        =      FUEL EDGROD INTERNAL
                      SEGMENT  PCO15 PCO18 ISO 1 DIAM 0.D0 FDUOE
                      UO2  IRRAFUEL  WZUO        1.0D0
                      LAW  LAWRAD    GAP  THG
                      OXIDE          EOI   1.D-8         EOE  1.D-8
                      CLADDING  ZRFRA                    RM    RMC;

CARCBCO2=      FUELCHAR EDGROD
                      GAP  DELTFUEL 1.0D-8GAPP     0.1D0
                      CRAD0.06D-3
                      FUELP        INTERNAL  POWNEUT 1.D0   POWRESI 0.D0
                             LAWNEUT LAWPOW     LAWRESI LAWPOW
                      GASVOL     VOIDV 0.    EXPANV      1.4D+7
                      BPLUGV     0.          CRACKV      0.
                      POROSV     0.          TPLUGV      0.
                      VOLFRAC  1.D0          POROS       3.96D-2
                      XNEUT      1.0;

INTEGRATE FUEL2C1 NUMBER 1 CARCBCO2;
```

```
****************************************************************************
* DEFINITION OF THE CATAFUEL ELEMENT
****************************************************************************

CATA        =       CATAFUEL EDGROD FILE 66;
END DATA;

****************************************************************************
* EXEC BLOC
****************************************************************************

DOUBLE  DT  TIME;
INTEGER NPAS;

RESTORE;

***** STEADY STATE *****************************************************

READHEXT        CATA TIME DT;
STDYFUEL  CATA;

***** STABILISED TRANSIENT ***********************************************

READHEXT        CATA TIME DT;
REPEAT BLOC1 10;
             TRANFUEL CATA TIME DT;
END BLOC1;

***** TRANSIENT **********************************************************

GOFUEL      CATA;
PERIOD EDGROD ALL SECOND 5.;
RESULT CATA       SECOND 1.;
OPTION DTMAX   1.;
OPTION ITERMIN 6 ;

NPAS = 9999;
REPEAT BLOC2 NPAS;
             READHEXT        CATA TIME DT;
             TRANFUEL CATA TIME DT;
END BLOC2;

END EXEC;
****************************************************************************
* END
****************************************************************************
```

This input data file makes use of an external fluid file such as example 6 shown in the next section.

Note that the external fluid file is read at several times:

- Once just before the calculation of the steady state for the initialisation of fluid and fuel variables (the fuel temperatures are set equal to the fluid temperatures). The first time of the external fluid file must be 0.
- Once before each time step calculated by TRANFUEL in the BLOC1 loop if the user wants to calculate the stabilised transient. The time read by READHEXT must also be 0 (see example 6).
- Once before each time step calculated by TRANFUEL in BLOC2 loop. The values of the time must then increase and be different from 0. If the number of steps stored in the external fluid file is lower than NPAS (the number of iterations in BLOC2 loop) the calculation simply stops.

Practical advice:

If it is not equal to 1, the number of axial meshes specified in the external fluid file *nmesh* must imperatively be the same as the number of segments defined for the HYDIMP element by the MESH directive.

7.2.1.6 Examples of external fluid file

Example 6: Case ITYCL = 1

```
'TCS EDGAR ESSAIS RAMPE 461A          :        ITYCL = 1'
 NOREFLOO  EXCHCOEF  AXIFLUI 1  16
   0.    1.  352.  352.
                   352.  -50.5   1.E+10   1
   0.    1.  352.  352.
                   352.  -50.5   1.E+10   1
   8.75  1.  384.  384.
                   384.  -50.5   1.E+10   1
  18.5   1.  441.  441.
                   441.  -50.5   1.E+10   1
  28.75  1.  495.  495.
                   495.  -50.5   1.E+10   1
  38.5   1.  555.  555.
                   555.  -50.5   1.E+10   1
  48.5   1.  607.  607.
                   607.  -50.5   1.E+10   1
  58.5   1.  657.  657.
                   657.  -50.5   1.E+10   1
  68.5   1.  714.  714.
                   714.  -50.5   1.E+10   1
  78.5   1.  767.  767.
                   767.  -50.5   1.E+10   1
  88.5   1.  813.  813.
                   813.  -50.5   1.E+10   1
  98.5   1.  857.  857.
                   857.  -50.5   1.E+10   1
 100.0    1.  864.  864.
                   864.  -50.5   1.E+10   1
```

Example 7: Case ITYCL = 2

```
'TEST WITH IMPOSED FLUXES: ITYCL = 2'
  NOREFLOO FLUX AXIFLUI 10 322
    0.0000   0.999999E+00   0.288628E+03   0.339567E+03
  0.1609161D+08  297.25
  0.1607836D+08  313.36
  0.1606500D+08  329.54
  0.1605149D+08  344.08
  0.1603779D+08  349.34
  0.1602386D+08  350.00
  0.1600967D+08  350.16
  0.1599522D+08  349.94
  0.1598056D+08  349.31
  0.1596580D+08  347.49              2
    0.0137   0.995284E+00   0.288514E+03   0.339399E+03
  0.1578822D+08  0.2775101D+06
  0.1578151D+08  0.6486849D+06
  0.1577437D+08  0.9625006D+06
  0.1576675D+08  0.1190276D+07
  0.1575856D+08  0.1371547D+07
  0.1574959D+08  0.1374662D+07
  0.1573979D+08  0.1253529D+07
  0.1572915D+08  0.1021712D+07
  0.1571774D+08  0.6996534D+06
  0.1570567D+08  0.3117226D+06       3

...

   17.0000   0.630558E-01   0.188725E+03   0.192005E+03
  0.1218070D+07  0.2057241D+06
  0.1218066D+07  0.2731334D+05
  0.1218045D+07  0.3036480D+05
  0.1218021D+07  0.3426380D+05
  0.1217988D+07  0.3787859D+05
  0.1217940D+07  0.4181048D+05
  0.1217870D+07  0.4211230D+05
  0.1217778D+07  0.2925798D+05
  0.1217655D+07 -0.1371475D+05
  0.1217546D+07 -0.1229314D+05      322
```

Example 8: Case ITYCL = 4

```
'TEST WITH IMPOSED VARIABLES AND HYDRAULIC DIAMETERS: ITYCL = 4'
 NOREFLOO
 VARIABHD
 AXIFLUI  5  12
1.2034D-2  1.2034D-2  1.2034D-2  1.2034D-2  1.2034D-2
 0.       1.  133.68  133.66
     3.00000  133.68  133.68  0.9999990   1.D-3       1.D-3
     3.00000  133.68  133.68  0.9999990   1.D-3       1.D-3
     3.00000  133.68  133.68  0.9999990   1.D-3       1.D-3
     3.00000  133.68  133.68  0.9999990   1.D-3       1.D-3
     3.00000  133.68  133.68  0.9999990   1.D-3       1.D-3
1
 21.7500   1.  133.68  196.90
     3.00000  133.68  193.22  0.9999990   1.5475D-3   1.5475D-3
     3.00000  133.68  193.40  0.9999990   2.5040D-3   2.5045D-3
     3.00000  133.68  193.40  0.9999990   3.2675D-3   3.2680D-3
     3.00000  133.68  193.40  0.9999989   3.9735D-3   3.9735D-3
     3.00000  133.68  210.67  0.9999990   4.9115D-3   4.9120D-3
2
 59.7500   1.  133.68  304.88
     3.00000  133.68  301.59  0.9999990   1.6045D-3   1.6045D-3
     3.00000  133.68  302.08  0.9999990   2.5485D-3   2.5485D-3
     3.00000  133.68  302.08  0.9999990   3.1810D-3   3.1810D-3
     3.00000  133.68  302.08  0.9999990   3.7645D-3   3.7650D-3
     3.00000  133.68  348.66  0.9999990   4.6180D-3   4.6185D-3
3

...

 223.981  1.    76.17  141.08
     3.22625   78.52  136.17  0.0000100   8.2140D-2   8.2140D-2
     3.21216   80.50  136.02  0.0000100   8.2250D-2   8.2250D-2
     3.20013   82.22  135.89  0.0000100   8.2345D-2   8.2345D-2
     3.18903   83.91  135.77  0.0000100   8.2435D-2   8.2435D-2
     3.17732   86.39  135.64  0.0000100   8.2550D-2   8.2550D-2
12
```

7.2.2 Second method for a stand-alone fuel calculation (Option 2)

This section describes how to create a CATHARE hydraulic file and how to use it later in the calculation of a stand-alone fuel.

For further details on all the following operators and directives specific to the stand-alone fuel calculation, refer to the corresponding pages of [DOC3].

7.2.2.1 Creation of a CATHARE hydraulic file

The CATHARE hydraulic file is built during a standard CATHARE calculation when the STOREHYD directive is specified in the input file. The data set is exactly the same as the common data sets.

Case of a calculation with axial components
After the circuit and reactor definitions and just before the end of the DATA block, the STOREHYD directive must be added followed by the names of the AXIAL elements for which the hydraulic conditions have to be stored. The fluid data are stored after the steady state and then at each step of the stabilised transient and of the transient.

Case of a calculation with a threed component
After the circuit and reactor definitions and just before the end of the DATA block, the 1D hydraulic channels have to be defined by using the HYDCHAN operator. The hydraulic channels defined in such a way can be likened to fictitious axial elements and will become HYDIMP elements in the input data file of the stand-alone fuel calculation. Then, the STOREHYD directive must be added followed by the name of the THREED element and the 1D hydraulic channel names for which the hydraulic conditions have to be stored.

In case of reflooding in the CATHARE hydraulic calculation, all the information dealing with the quench front is also automatically saved and will be automatically taken into account in the stand-alone fuel calculation (no use of REFLOOD directive is needed).

The NEUTRO key word is obsolete since the non-residual and residual powers corresponding to the fuel wall in the hydraulic channel (1D or 3D) are always saved in the hydraulic file. Consequently, using it or not is of no importance.

If the user does not want to store hydraulics at every time step, he can use the RESHCAT directive, the logic and syntax of which are similar to those of the RESULT directive.

It should be noted that several hydraulics may be stored in the same hydraulic file (STOREHYD).

The description of the CATHARE hydraulic file is given in Appendix 11.

Example 9: CATHARE data set for the creation of a CATHARE fluid file – AXIAL components case

```
BEGIN DATA;
...
PIPE1=     AXIAL...;
PIPE2=     AXIAL...;
PIPE3=     AXIAL...;
CORE1      =      CORE       PO  1.4D8   CORENAME       PIPE1  PIPE2  PIPE3     ...;
…
AIR   =    NONCOND  2       HYDROGEN                NITROGEN;
CIRC  =    CIRCUIT    PIPE1 PIPE2 PIPE3 ...  AIR;
CIRCTOT=   REACTOR   CIRC;

STOREHYD PIPE1                PIPE3        NEUTRO;

END DATA;
```

Example 10: CATHARE data set for the creation of a CATHARE fluid file – THREED component case

```
BEGIN DATA;
...
CUVE    =   THREED     ...;

…
FUELCH1    =      FUELCHAR CUVE…;
FUELCH2    =      FUELCHAR CUVE…;
FUELCH3    =      FUELCHAR CUVE…;
…
CORE1          =  CORE       PO  1.4D8   CORENAME       PIPE1  PIPE2  PIPE3     ...;
...
AIR       =  NONCOND  2       HYDROGEN                NITROGEN;

CIRC      =  CIRCUIT            ... PIPE1 PIPE2 PIPE3 ...  AIR;
CIRCTOT    =      REACTOR   CIRC;

HYDCH1     =      HYDCHAN   FUELCH1;
HYDCH2         =  HYDCHAN   FUELCH2;
HYDCH3         =  HYDCHAN   FUELCH3;

STOREHYD          CUVE       HYDCH1     HYDCH2     HYDCH3     NEUTRO;

END DATA;
```

7.2.2.2 Data set for a stand-alone fuel calculation

The main points of the data set for a stand-alone fuel calculation are the following:

- definition of the hydraulic channels (HYDIMP operator),
- definition of fuel rods (FUEL, FUELCHAR operators and INTEGRATE directive),
- definition of the circuit (CATAFUEL operator),
- definition of the command block (READHEXT, STDYFUEL and TRANFUEL directives).

7.2.2.2.1 Definition of imposed hydraulic elements (hydraulic channels)

Case of a calculation with axial components

The HYDIMP element has the same function as the AXIAL element in a standard CATHARE calculation except that its hydraulics is read in a file created by CATHARE (see example 12). The HYDIMP element is necessary to associate a fuel rod (a FUELCHAR element) to the hydraulics stored in the CATHARE fluid file (see example 14). As values in the hydraulic file correspond to the mesh of the axial elements stored by STOREHYD directive during the direct calculation, this mesh should also be defined (XAXIS, SEGMENT) and associated with the HYDIMP element (MESH).

Example 11: Definition of HYDIMP elements – AXIAL component case

```
BEGIN DATA;
...
P10  =    XAXIS     0.;        P11  =    XAXIS     3.6576;
P20  =    XAXIS     0.;        P21  =    XAXIS     3.6576;

MHYD1     =    P10  SEGMENT  20    P11  COS 1.;
MHYD2     =    P20  SEGMENT  20    P21  COS 1.;

HYD1 =    HYDIMP         PIPE1;
HYD2 =    HYDIMP         PIPE2;

MESH           HYD1           MHYD1;
MESH           HYD2           MHYD2;
...
END DATA;
```

Case of a calculation with a threed component

The HYDIMP element has the same function as the axial element in a standard CATHARE calculation except that its hydraulics is read in a file created by CATHARE (see example 13). The HYDIMP element is necessary to associate a fuel rod (a FUELCHAR element) to the hydraulics stored in the CATHARE fluid file (see example 14). As values in the hydraulic file correspond to the mesh of the fictitious axial elements stored by STOREHYD directive during the direct calculation, the mesh corresponding to the hydraulic channel (HYDCHAN) of the direct calculation should be defined (XAXIS, SEGMENT) and associated with the HYDIMP element (MESH).

In the following example 13, HYDCH1 and HYDCH2 are the stored axial channels (HYDCHAN) of a THREED saved in the CATHARE fluid file by a previous CATHARE calculation with the STOREHYD directive (see example 11). The geometries attached to the HYDIMP elements have to coincide with the HYDCHAN ones.

Example 12: Definition of HYDIMP elements – THREED component case

```
BEGIN DATA;
...
P10  =    XAXIS     0.;        P11  =    XAXIS     3.6576;
P20  =    XAXIS     0.;        P21  =    XAXIS     3.6576;

MHYD1     =    P10  SEGMENT  20    P11  COS 1.;
MHYD2     =    P20  SEGMENT  20    P21  COS 1.;

HYD1 =    HYDIMP         HYDCH1 ;
HYD2 =    HYDIMP         HYDCH2    ;

MESH           HYD1           MHYD1;
MESH           HYD2           MHYD2;
...
END DATA;
```

Remarks:

As can be seen in the previous examples 12 and 13, several HYDIMP elements may be defined in the input file, i.e., CATHARE is able to calculate simultaneously several fuel thermal-hydraulic behaviours (if the hydraulics of several hydraulic channels has been stored with the STOREHYD directive during the direct calculation).

Moreover, if the hydraulics of several hydraulic channel has been stored with the STOREHYD directive during the direct calculation, the HYDMOD directive can be used to change the stored hydraulics imposed on a HYDIMP element during the stand-alone fuel calculation (if they have identical meshes). For example, in a LB-LOCA, the HYDIMP element can be attached to the HOTCORE saved hydraulics during the blow down and refill phases, and then to the MEANCORE hydraulics during the reflood phase.

7.2.2.2.2 Definition of fuel rods

After the definition of the HYDIMP elements the definition of the fuel rods is given.
This is a 1D definition (use FUEL operator and not FUEL3D even if the hydraulic file was created considering a fuel in a THREED element because the stand-alone fuel calculation is always conducted on a 1D equivalent element).
Besides, the name of the hydraulic element (1D or 3D) is replaced by the HYDIMP element name.

Example 13: Definition of fuel rods -

```
BEGIN DATA;
...
FUEL1       =     FUEL              HYD1 INTERNAL
                  SEGMENT  P10 P11  ISO 1 DIAM 0.D0 FDUOE ...;

FUEL2       =     FUEL              HYD2 INTERNAL
                  SEGMENT  P20 P21  ISO 1 DIAM 0.D0 FDUOE ...;

FUEL3       =     FUEL              HYD2 INTERNAL
                  SEGMENT  P20 P21  ISO 1 DIAM 0.D0 FDUOE ...;

FUELCH1=  FUELCHAR HYD1 ...;
FUELCH2=  FUELCHAR HYD2 ...;
FUELCH3=  FUELCHAR HYD2 ...;

INTEGRATE         FUEL1             NUMBER 1          FUELCH1;
INTEGRATE         FUEL2             NUMBER 1          FUELCH2;
INTEGRATE         FUEL3             NUMBER 1          FUELCH3;
...
END DATA;
```

Fuel wall over-meshing (CATHARE 2 V2.5 mod4.1 and further versions)

A new functionality has been added in mod4.1 version to allow over-meshing of the fuel wall in the stand-alone fuel calculation with respect to the direct calculation. This enables the user to perform fuel meshing sensitivity studies with direct calculation achieved at low cost on a coarse mesh.

Basically, the user chooses a multiplicative coefficient **n** for the number of meshes. Rules to follow in the stand-alone fuel input deck are as follows (see Figure 35: Stand-alone fuel over-meshing rules):

- the total length of the fuel wall should be the same as in the direct calculation input deck,
- each fuel mesh of the direct calculation input deck has to be represented by **n** fuel meshes in the stand-alone fuel input deck,
- each vector point of the direct calculation fuel mesh has to be kept as a vector point for the stand-alone fuel input deck fuel mesh.

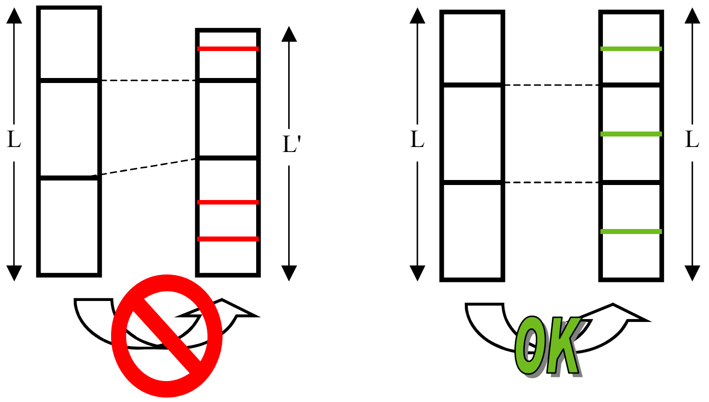


Figure 35: Stand-alone fuel over-meshing rules

The main principle is that hydraulic values given by the direct calculation facing each fuel mesh are copied and used in the stand-alone fuel calculation for **n** fuel meshes (see Figure 36: Stand-alone fuel over-meshing principle).

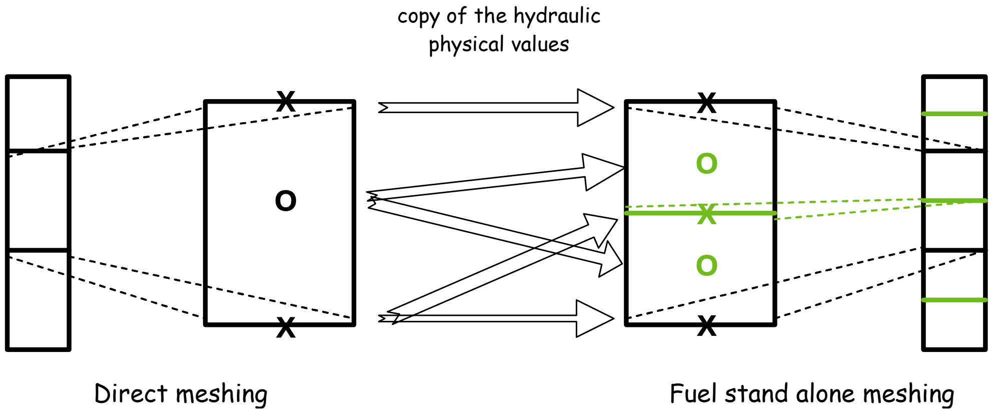


Figure 36: Stand-alone fuel over-meshing principle

<u>Fuel wall power</u> (CATHARE 2 v2.5 mod4.1 and further versions)

In a direct calculation, there are three ways to define the power to be generated by a fuel wall (see §5.4.3):

- internally by defining laws for non-residual and residual powers,
- externally by directly accessing non-residual and residual powers with CCV,
- using the CORE sub-module – in this case, the powers are distributed by the CORE sub-module between all the fuel walls (see §5.4.5.2).

The first and second ways are also available for a stand-alone fuel calculation; but as the third possibility cannot be used in a stand-alone fuel calculation, and to facilitate user management of this kind of computation, fuel non-residual and residual powers are saved (in all cases) in the hydraulic file. This allows the user to re-use powers from the direct calculation, using the CATAFUEL option in the <u>FUELCHAR</u> operator definition.

In this case:

- the user defines a multiplicative coefficient for the non-residual power which will be applied to the stored values from the direct calculation,
- the user defines a nominal value and a time-dependent law for the residual power (it will then be independent of the direct calculation),
- the NEUTRO key word has to be used in the <u>CATAFUEL</u> operator definition.

7.2.2.2.3 Definition of the CATAFUEL element

The CATAFUEL element may be considered as the REACTOR/CIRCUIT of a stand-alone fuel calculation. It is defined using the CATAFUEL operator. It enables several stand-alone fuels to be calculated simultaneously.

Rules of definition:

- The order of HYDIMP elements listed in the CATAFUEL operator must follow the order of storage of their attached hydraulics (AXIAL elements or 1D hydraulic channels (HYDCHAN)) during the original calculation (see STOREHYD directive in examples 10, 11).
- If the CATAFUEL option is used in FUELCHAR definition to reuse the powers stored during direct calculation, the NEUTRO option has to be used in the CATAFUEL definition.
- If the CATHARE fluid file has been created by a calculation with non-condensable gases the same NONCOND element must be defined and specified in the CATAFUEL operator.

Example 14: Definition of the *catafuel* element

```
BEGIN DATA;
...
AIR  =    NONCOND  2     HYDROGEN          NITROGEN;
CATA =    CATAFUEL HYD1 HYD2 NEUTRO   AIR;
...
END DATA;
```

7.2.2.2.4 Definition of the command block

The command block must be organised as follows:

- first reading of the external fluid file (READHEXT) before the stand-alone fuel steady state calculation (STDYFUEL);
- a loop for the stabilised transient if necessary (READHEXT, TRANFUEL);
- launching fuel thermo-mechanical computation (creep) (GOFUEL);
- a loop for stand-alone fuel transient steps (READHEXT, TRANFUEL).

In example 16, the BLOC1 loop is defined in case of a stabilised transient calculated and stored during the original CATHARE calculation. This loop is skipped if the time read by READHCAT is greater than 0. The BLOC2 loop will perform the transient steps. The user must take care with the successive calls of READHCAT and TRANFUEL to avoid time step skipping.

Take care to use the GOFUEL directive before the fuel transient calculation, in order to allow the thermo-mechanical calculation.

The time and time step values are read from the fluid file by the READHCAT directive and transferred to the TIME and DT variables declared in the data set. As in a CATHARE standard calculation, TIME=TOLD+DT. Then the TRANFUEL directive tries to reach the time proposed by the fluid file. If a fuel calculation convergence problem occurs or if a maximum time step is set to a value lower than DT (by the OPTION DTMAX) then a sub-cycling is performed in TRANFUEL in order to reach the time t=TIME in several steps.

Therefore, the user must not use the NEWTIME and NEWDT operators to update TIME and DT.

Example 15: Definition of the *command block*

```
INTEGER    NPAS;
DOUBLE              TIME  DT;
RESTORE;

***** STEADY STATE *****

READHCAT            CATA          TIME  DT;
STDYFUEL            CATA;

***** STABILISED TRANSIENT *****

NPAS          =     9999;

READHCAT            CATA          TIME  DT;

REPEAT BLOC1 NPAS;
        IF (TIME > 0.);     QUIT  BLOC1;      ENDIF;
        TRANFUEL CATA TIME DT;
        READHCAT CATA TIME DT;
END BLOC1;

***** TRANSIENT *****

GOFUEL              CATA;

OPTION              DTMAX         0.1;
PERIOD              CATA          ALL   SECOND          1.0;
RESULT              CATA          SECOND          0.1;

REPEAT BLOC2 NPAS;
        IF (TIME > 30.);    QUIT  BLOC1;      ENDIF;
        TRANFUEL CATA TIME DT;
        READHCAT CATA TIME DT;
END BLOC2;

END EXEC;
```

APPENDIX 1 List of files used by CATHARE

Many FORTRAN units are used or opened by CATHARE. Only a few have an interest for CATHARE standard calculation (see [DOC2] WHAT are the FILES USED by CATHARE?).
The tables below list all the files managed by the CATHARE code.

Symbols used in the following tables:
+++ Debug facility which should not be used to perform a calculation,
(*) Not used in case of a PC calculation,
(**) Only opened if the corresponding directive is used,
< Is followed by the corresponding CATHARE directive,
xx File opened using any free FORTRAN unit.

READER executable

File	Name	Description	Remarks
03	**PILOT.f**	Output file, image of command block	Copy into PILOT.f to be compiled to create CATHAR program
04	**FAST.H**	FORTRAN include file used to allocate space for the variables of the calculation	Default values defined can be changed using the FASTSIZE directive
05	**Name_of_input_deck**	Input data file	
06	**Selected output V25_1.LIST (PC use)**	Standard output	
08	**DICO**	List of CATHARE key words	
14	**V25_1.INIT**	Output file, image of data block	
15	**V25_1.INIT.FORM**	Formatted output file, image of data block	+++ < OPTION SAVE FORMAT; (before END DATA directive;)

POSTPRO executable

File	Name	Description	Remarks
05	**Name_of_input_deck**	Graphic input deck	
06	**V25_1.LIST (PC use) or selected output**	Standard output	
07	**FORT07**	Output file	
08	**DICO**	List of CATHARE key words	
30	**FORT30**	Standard input formatted file	+++
31	**FORT31**	Standard input file	can also be 21 or in the range 60 – 80 according to the user specification
11 **	**FORT11**	Standard experimental result file	< READ EXPERIM can be also in range 60 – 80 and must be then specified by the user
60 **	**FORT60**	file resulting from compression of two unformatted results files	< COMPRESS
61 **	**FORT61**	file resulting from compression of two formatted results files	< COMPRESS
81 **	**FORT81**	Buffer	< COMPRESS

CATHARE executable

File	Name	Description	Remarks
06	**Selected output** **V25_1.LIST (PC use)**	Standard output	
09 **	**V25_1.HIST** **(or V25_1.HIST.FORM)**	List of commands used in PILOT	< OPTION HISTORY
11 **	**JACOB4**	Jacobian device: Jacobian Matrix	< VERBOSE directive
12 **	**JACODX**	Jacobian device: Jacobian Matrix * Increment vector	< VERBOSE directive
14	**V25_1.INIT**	Input file (stack of initial data)	< READER
15	**V25_1.INIT.FORM**	Formatted input file	< READER +++
16 *	**V25_1.RESTART.FORM**	Formatted restart file	+++ < SAVE directive added to OPTION SAVE FORMAT; command line used after RESTORE directive;
20	**FORT20**	Formatted result file	+++ < RESULT directive added to OPTION POSTPRO FORMAT; command line used after RESTORE directive;
21	**FORT21**	Binary result file	< RESULT directive
24	**V25_1.STOP**	User external and manual control upon the calculation continuation	cannot be use for PC calculations STOP or CONT key word is read each time step in this file
26	**V25_1.TRAIN**	Convergence studies + Simulator use	< VERBOSE directive
27	**V25_1.CONVR**	Convergence studies	< VERBOSE directive
28	**V25_1.BRIDE**	Convergence studies using cycle management	< MANAGE directive +++
35	**LISTING3D**	Checking of some 3D variables	< VERBOSE LIST3D directive
42 *	**CATHARE.LOG**	Icare/CATHARE log file	
43 *	**ICARE.LOG**	Icare/CATHARE log file	
44	**V25_1.WHAT**	Version label	Case of HP calculation
50 **	**PICTG**	Cladding maximum temperature in a time interval and other information	< PICTG directive
55	**V25_1.RESTART**	Restart file	< SAVE directive
57	**FORT57**	MSE3D method	Not documented
60 **	**PARAMS**	Pre sensitivity calculation: List of studied responses	< TRANSENS
61 **	**INDEX.i**	Pre sensitivity calculation: List of time step history	< TRANSENS i = circuit number
62	**FORT62**	Imposed hydraulic conditions	Catafuel calculation
63 *	**FORT63**	Imposed hydraulic conditions in formatted file	Catafuel calculation
64	**visu.cir**	File for cathare GUI	< VISUCAT
65	**visu.pla**	File for cathare GUI	VISUCAT option
80+i **	**FORT(80+i)**	Restart file	< OPTION NCPUSAV 1 + SAVE
90 **	**PERF**	Parallel calculation: Summary of overall performance	< OPTION PARALLEL, PESEE, PERF
92 **	**MAILLAGE**	Parallel calculation: CPU time per element	< OPTION PARALLEL, PESEE, PERF
xx **	**V25_1.RESTEST or** **V25_1.RESTEST.FORM**	Image of restart file used to control SAVE/RESTART facility	< RESTEST directive

xx **	**RES.i.j**
	JAC.i.j
		Files used for sensitivity calculation to store results which will be used by SENSIB program	< TRANSENS directive
			i = circuit number
			j = time step number

Simulator use

14	**V25_1.INIT**	Input file (stack of initial data)	

Simulator use in debug mode

14	**V25_1.INIT**	Input file (stack of initial data)	
21	**FORT21**	Result file	
26	**V25_1.TRAIN**	Convergence studies	< VERBOSE
27	**V25_1.CONVR**	Convergence studies	< VERBOSE
28	**V25_1.BRIDE**	Convergence studies using cycle management	< MANAGE directive +++
29	**V25_1.INCIT**	Warning convergence messages	
55	**V25_1.RESTART**	Restart file	< SAVE
09 **	**V25_1.HIST** **(or V25_1.HIST.FORM)**	History of commands used during BLOCK EXEC processing	< OPTION HISTORY

APPENDIX 2 Presentation of the Newton-Raphson iterative method applied to CATHARE

General presentation of the NEWTON-RAPHSON iterative method

Consider the following problem: Solve F[X(t+dt)] = 0 with an initial guess X(t).

If X(t) is not too far from the solution, then the following series converges to the desired solution X(t+dt) (see Figure 37: Non-dimensional example of the Newton Raphson method principle).

- X0 = X(t)
- $F'(X_k)\ dX = -F(X_k)$
- $X_{k+1} = X_k + dX$

Then X(t+dt) = $\lim\limits_{k \longrightarrow \infty} X_k$

Generalised to a set of equations and a set of variables, this gives the CATHARE numerical method.

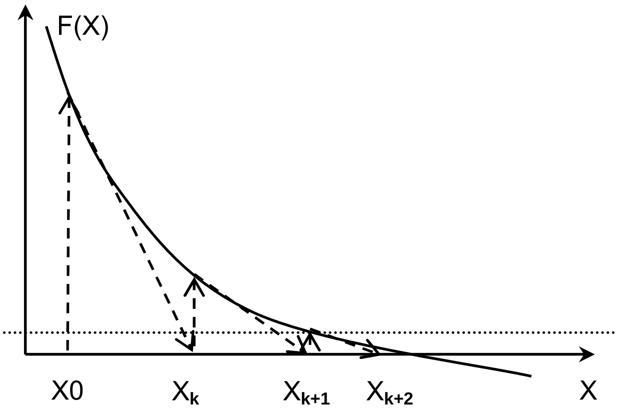
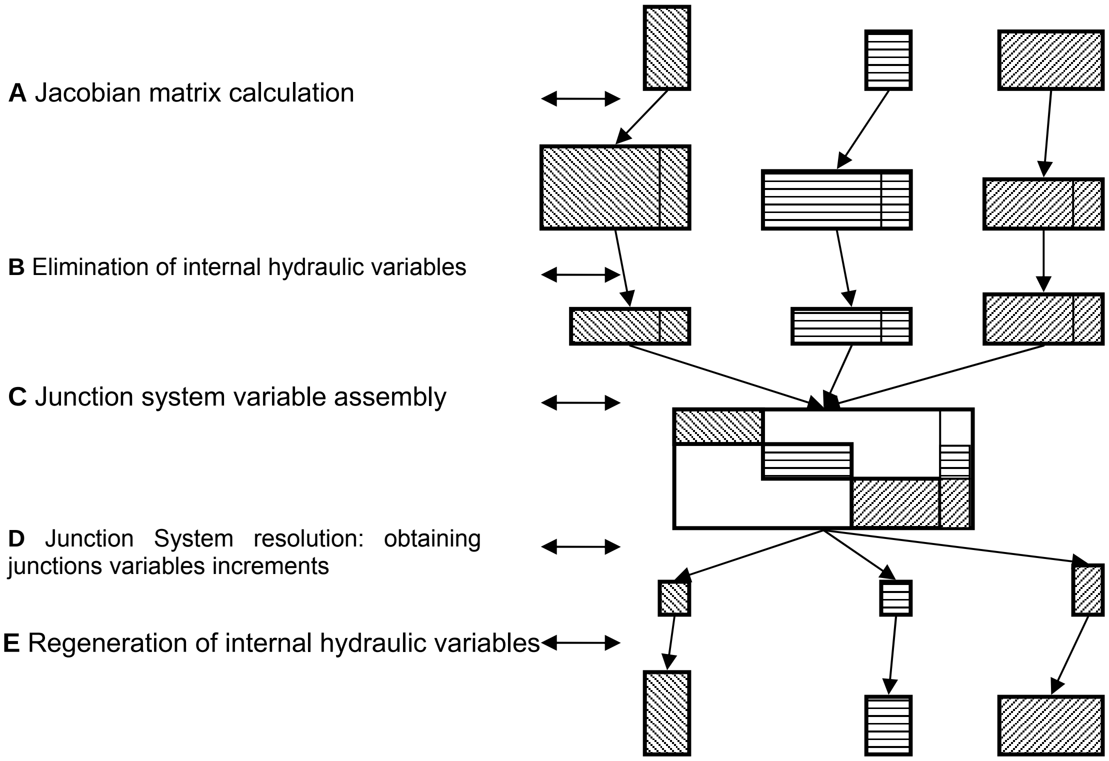


Figure 37: Non-dimensional example of the Newton Raphson method principle

CATHARE calculation steps

The first object represents for each module, the thermo-hydraulic state vector.


A Jacobian matrix calculation
B Elimination of internal hydraulic variables
C Junction system variable assembly
D Junction System resolution: obtaining junctions variables increments
E Regeneration of internal hydraulic variables

Refer to the following pages for more information on steps A, B, C, D and E.

Step A: Writing a Jacobian matrix: the first member is obtained by differentiation of discretised equations with respect to the main variables (see figure below for the example of the pipe).

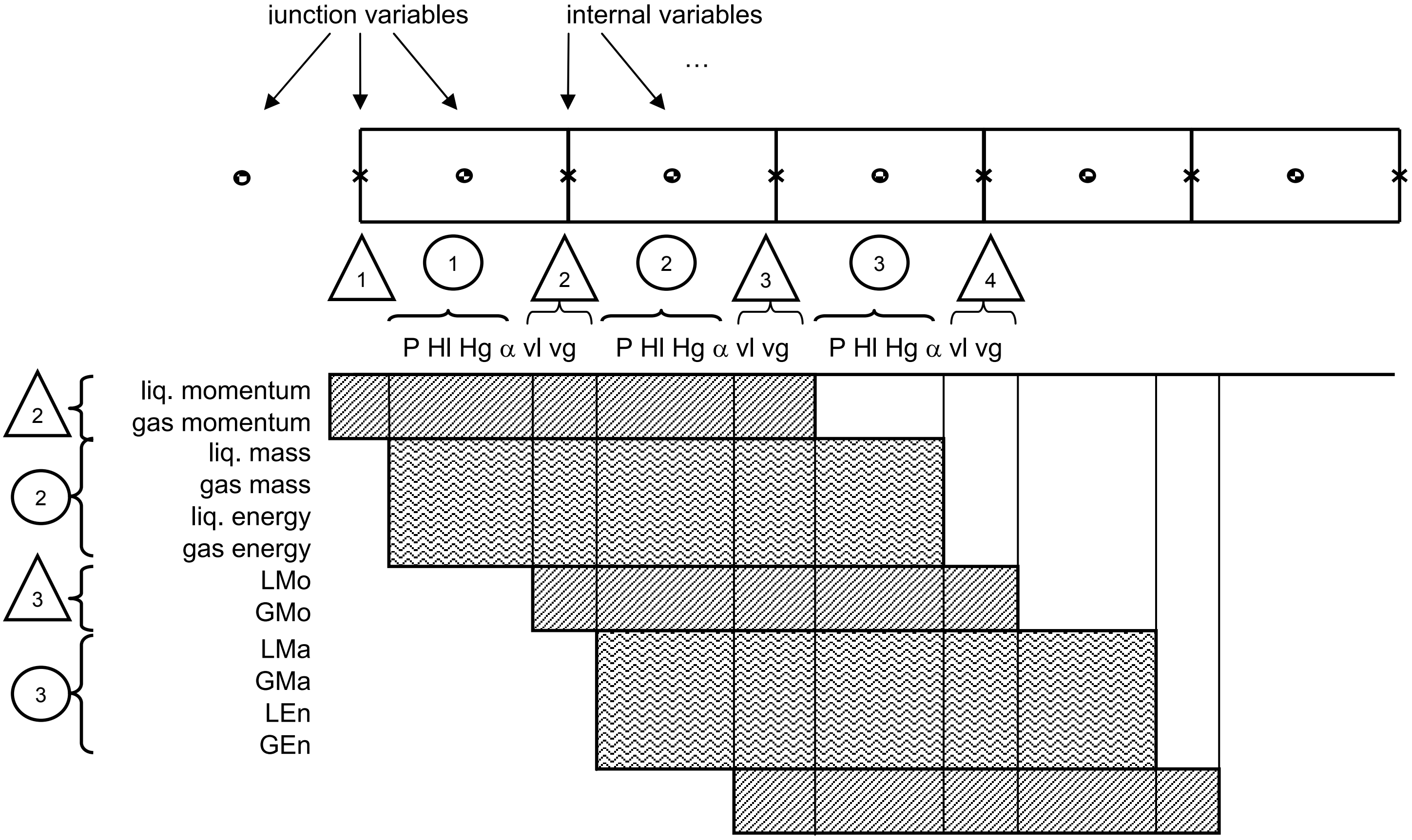


Figure 38: Building the AXIAL Jacobian matrix

Finally one obtains for each element a matrix of the following shape:

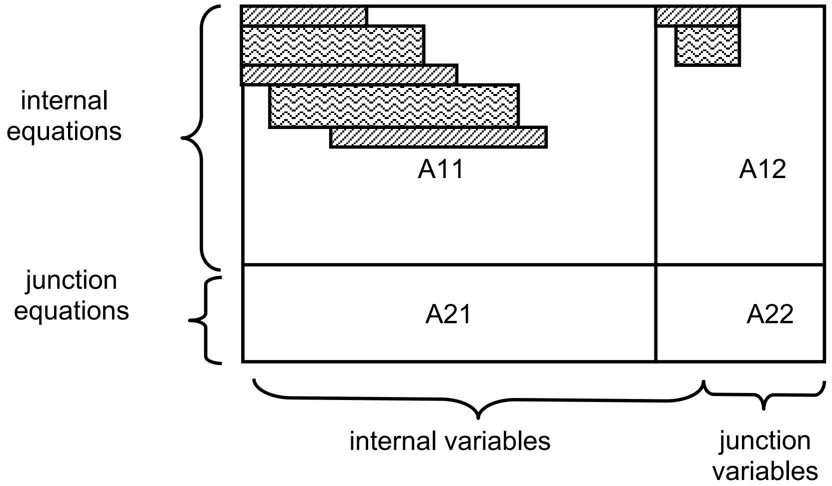


Figure 39: Jacobian matrix composition

with
A11: terms of equations for internal points depending on internal variables
A12: terms of equations for internal points depending on junction variables
A21: terms of equations for junction points depending on internal variables
A22: terms of equations for junctions points depending on junctions variables

Step B: reduction

For each element, the system obtained is reduced using the Gauss method to obtain a reduced system in which the unknowns are the main junction variables.

- System to be solved:(X1, X2) internal and external (at junctions) main variable increments

(B1,B2) second member (source terms)

$$\begin{cases} A11\ X1 + A12\ X2 = B1 \\ A21\ X1 + A22\ X2 = B2 \end{cases}$$

- Thus, internal variable regeneration formula:

$X1 = A11^{-1}\ (B1 – A12\ X2)$

- And the junction system to be solved:

$(-A21\ A11^{-1}\ A12 + A22)\ X2 = B2 - A21\ A11^{-1}\ B1$ which only depends on junction variable increments

Step C: circuit system assembly

All elementary matrixes are combined to form the general system to be solved

Steps D and E:

The system is first reduced using the Gauss method and increments of the main junction variables are calculated and then internal variable increments are calculated.

For each time step, the iterations are stopped when CATHARE considers that the solution is obtained. There are two ways to determine whether the solution (X for F(X)=0) is reached:

- check that F(X) is actually minimised: |F(X)| < e,
- ensure that dX is small enough: dX < e (e is called the convergence criterion).

In the CATHARE code the second way is used: for each time step, Newton-Raphson iterations are stopped as soon as the convergence tests are fulfilled for each module. This occurs in one module when the increments of all internal and external variables become lower than specific values (which are convergence criteria).

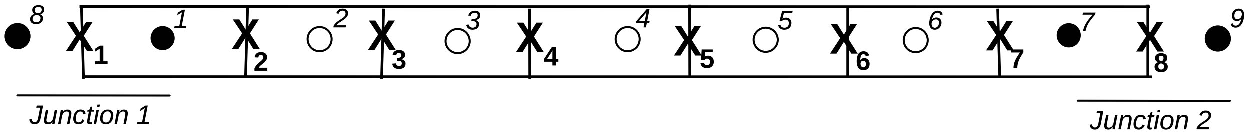
Example of convergence criteria for an AXIAL element:

- pressure: 100 Pa (same for non-condensable partial pressure),
- enthalpy: 100 J/kg,
- temperature (hydraulic and wetted wall): 0.01 °,
- void fraction: 0.01*min(a, (1- a)),
- velocities: 0.01 m/s,
- non condensable gas mass fraction: 0.01.

APPENDIX 3 Mesh definition and numbering in an AXIAL module

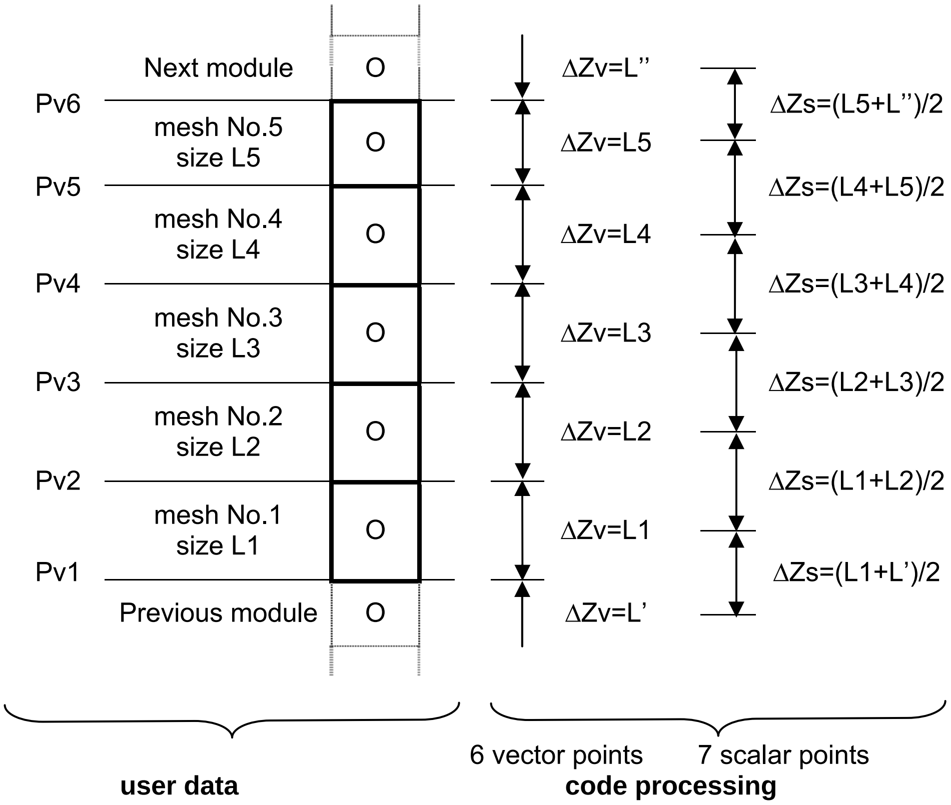
Case of a standard AXIAL module

When a pipe with N meshes is defined, it will have N+1 vector nodes and N+2 scalar nodes.
Numbering increases following the nodalisation definition.



- ● junction scalar point
- ○ AXIAL internal scalar point
- X vector point

The following scheme shows how the vector and scalar mesh lengths, used respectively for vector and scalar equation integration, are defined:



Pv is a vector point (defined by the SEGMENT operator).
ΔZv is the vector length of a mesh.
ΔZs is the scalar length of a mesh.
L' and L'' are imposed by adjacent modules.

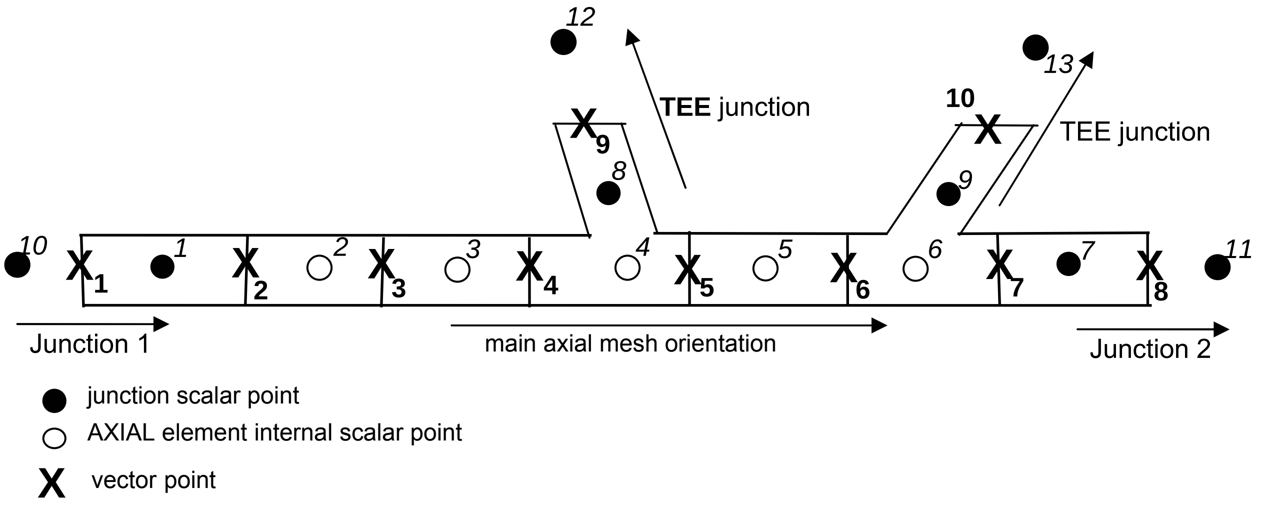
Remark: Velocity and flowrate signs are given with respect to the mesh orientation.

Case of an AXIAL element with tee branch connected (TEE)

When a pipe with N meshes and n tee branches connected is defined, it will have N+1+ n vector nodes and N+2+2n scalar nodes.

The numbering obeys the following rules:

- For scalar points, the internal scalar points are numbered first, followed by pipe (main axial element) junction external scalar points and then tee junction external scalar points.
- For vector points, the main axial vector points are numbered first, followed by the tee junction vector points.



APPENDIX 4 Mesh definition and numbering in a VOLUME module

When a volume with N ports is defined, it will have N vector nodes and 2N+2 scalar nodes (considering the junction external scalar points).

Considering junctions, **the numbering follows the order of junction definitions in the VOLUME operator.**

Example: Consider the following set-up:

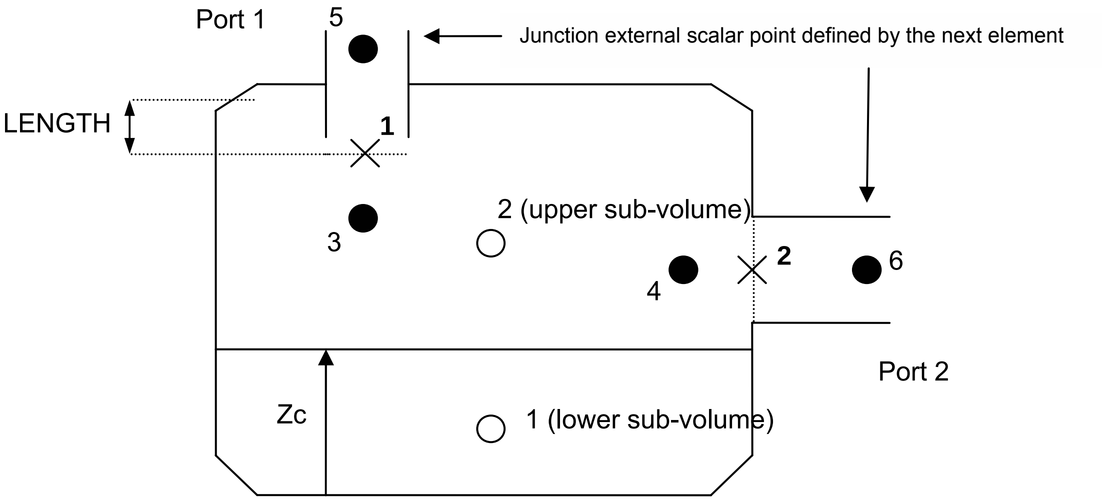
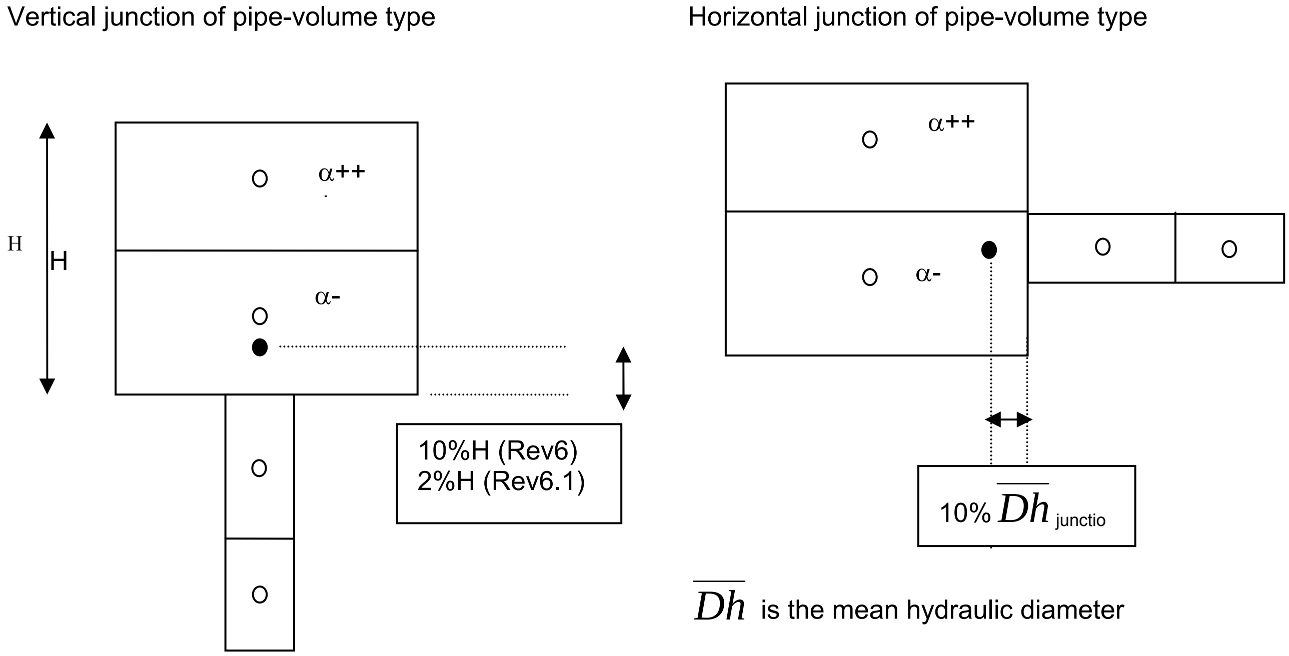


Figure 40: Numbering in a VOLUME element

- The volume is defined as:
  Volum1 = VOLUME port1 USTREAM WEIGHT 2 port2 DSTREAM;
- Scalar point numbering is:
  1 for the lower sub-volume
  2 for the upper sub-volume
  3 for port1 internal scalar point
  4 for port2 internal scalar point
  5 for port1 external scalar point
  6 for port2 external scalar point
- Vector point numbering is
  1 for port1 vector node
  2 for port2 vector node
  (the same as for junction numbering)

Lower and upper sub-volume scalar points are located with respect to the Zc level in the volume.

Junction internal scalar point position depends on the connected element and the junction type (see Figure 41: Position of the junction internal scalar point).



Vertical junction of pipe-volume type
Horizontal junction of pipe-volume type
H
H
α++
α-
10%H (Rev6)
2%H (Rev6.1)
α++
α-
10% $\overline{Dh}$ junctio

$\overline{Dh}$ is the mean hydraulic diameter

Figure 41: Position of the junction internal scalar point

Remark: Velocity and flowrate signs are given positive when the flow goes out of the volume and negative when it enters the volume.

APPENDIX 5 Discretisation and mesh numbering of the THREED module

Numbering of scalar nodes and vector nodes

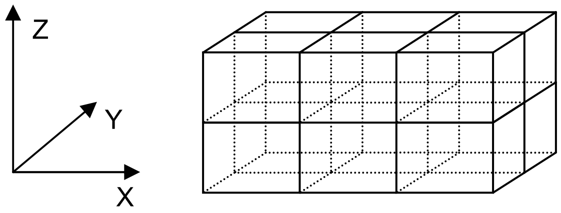
It is of prime importance for users to be able to retrieve the scalar and vector node numbering. It is used in building the input deck, but also in the post-processing of the 3D module. The numbering of scalar and vector nodes is called absolute numbering.

- numbering of **vector nodes** (meshes) also called **faces** (with 3 possible directions [X, Y, Z] or [θ, R, Z] depending on the chosen geometry),

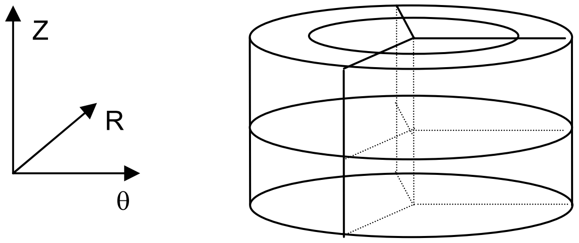
- numbering of **scalar nodes** (meshes)

For this numbering, meshes are always numbered in the same way: Z direction prevails over X (or θ) which prevails over Y (or R).

Consider building a 3D element with two meshes in the Z direction, **3** meshes in the X direction and two meshes in the Y direction. In the following examples, scalar meshes are always drawn.

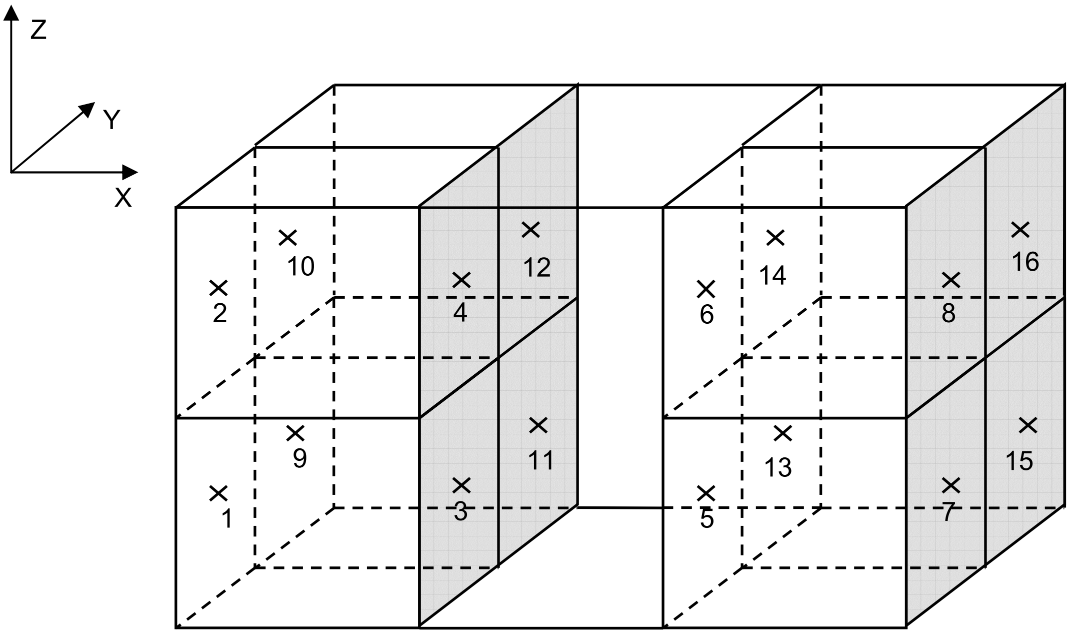


Remark: in cylindrical coordinates:

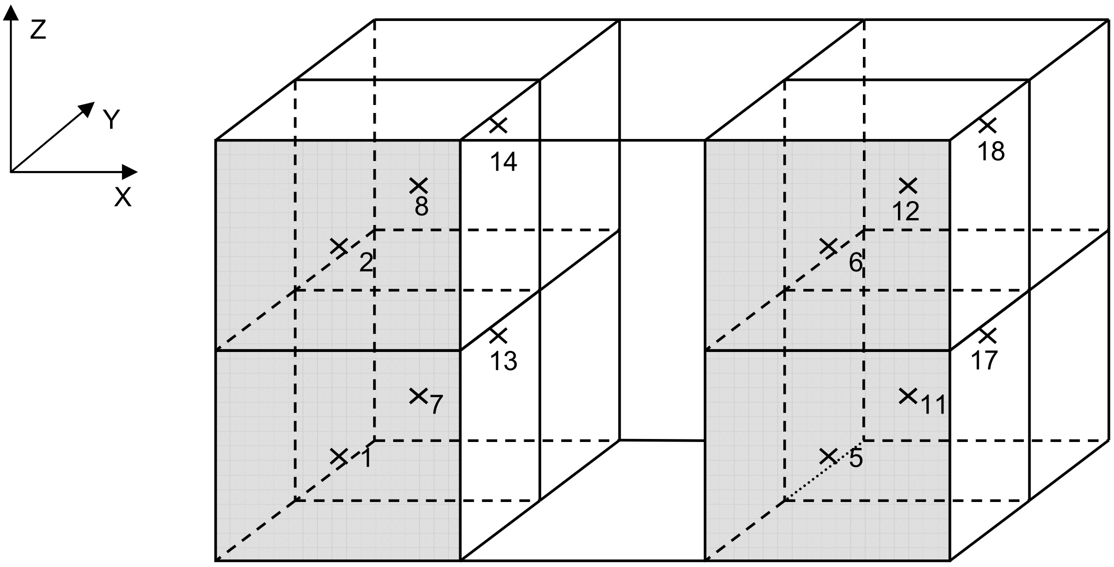


Examples for absolute numbering

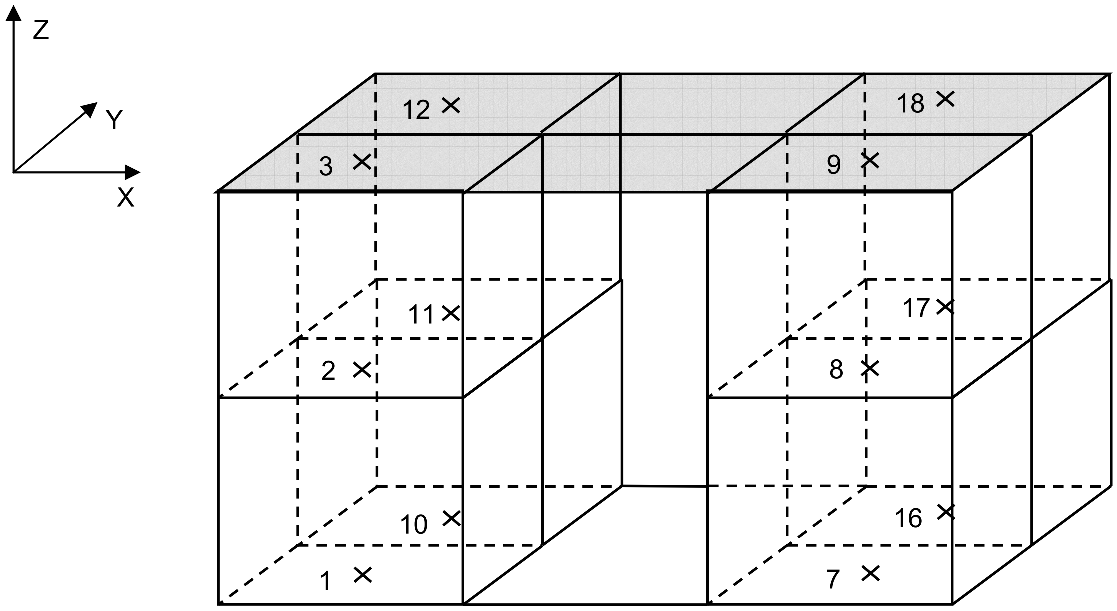
- **Numbering of X faces or X vector nodes**



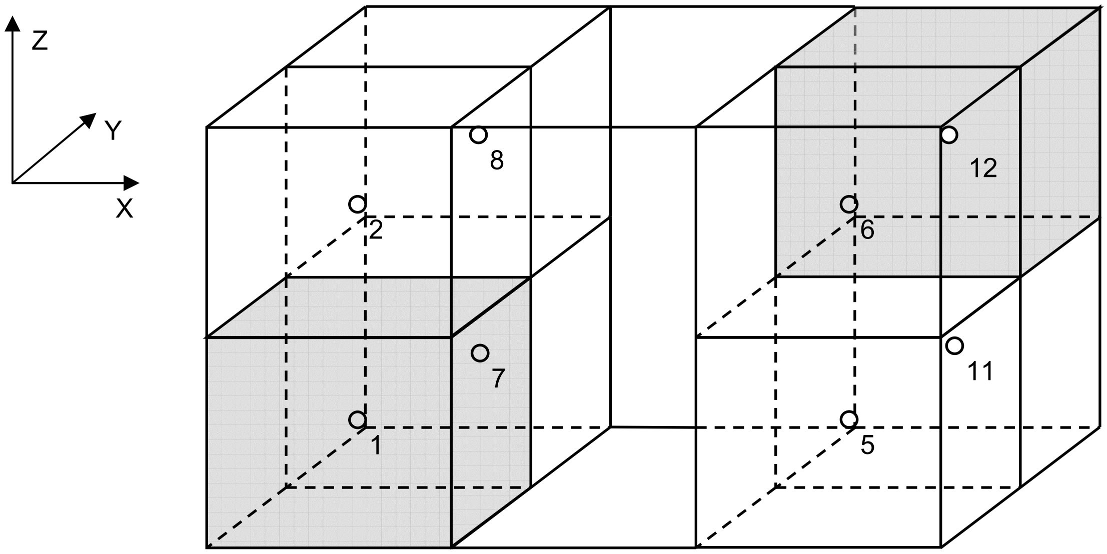
- **Numbering of Y faces or Y vector nodes**



- **Numbering of Z faces or Z vector nodes**



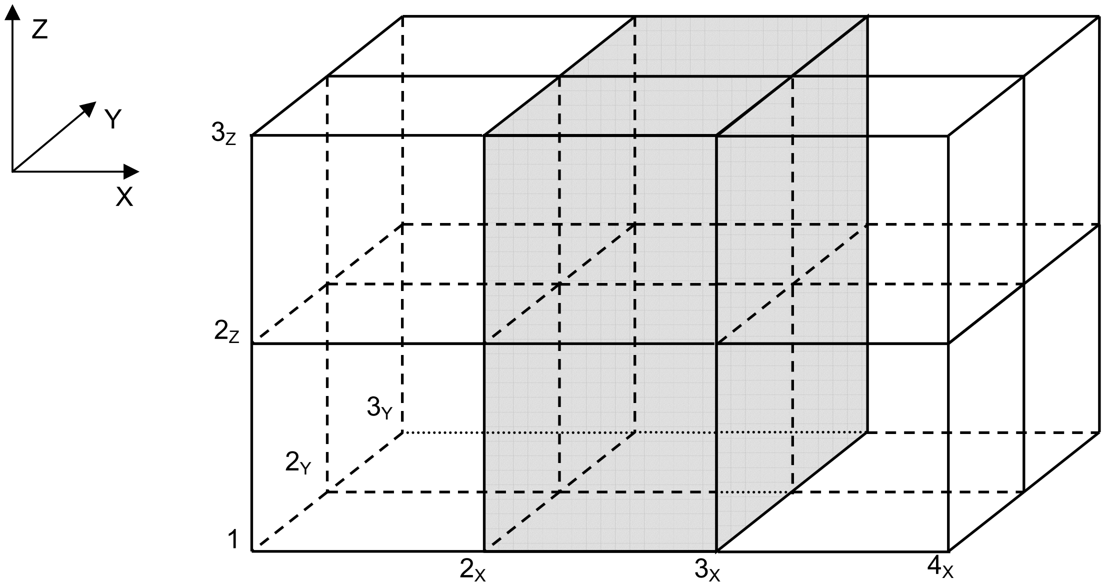
- **Numbering of scalar meshes**



Axis numbering

This numbering is used to define zones in directives related to THREED element definition. Zones defined in the MESH directive can be used in all other THREED directives (GEOM, HYDR, etc.). Vectors or scalar zones are defined the same way, by defining frontier faces which are vector planes in each direction.

- **Scalar zone numbering**

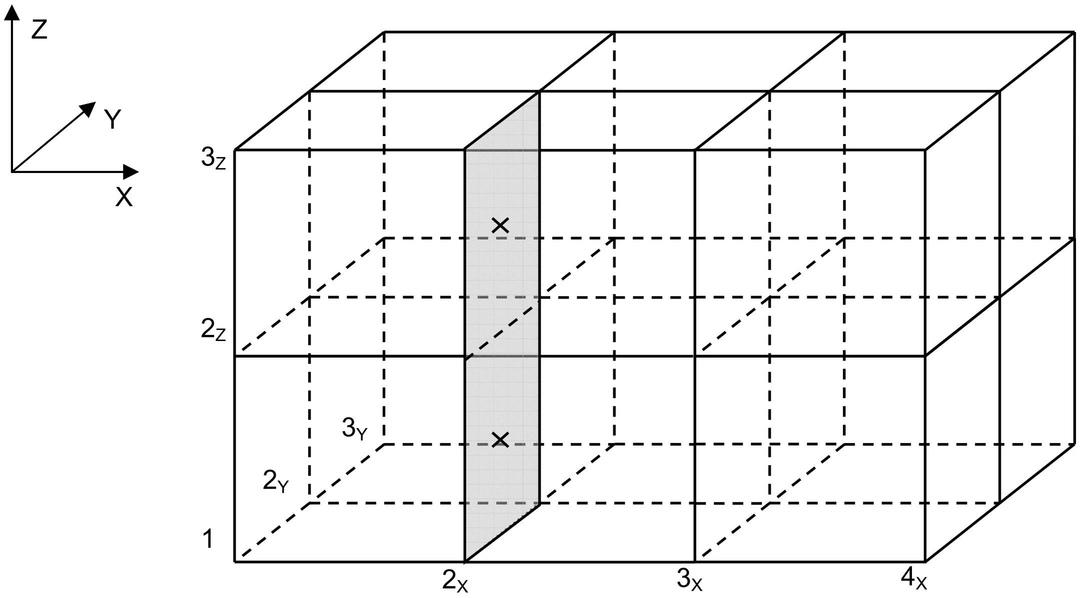


In the X direction, the zone (containing scalar meshes 3, 4, 9 and 10) is delimited by vector plane numbers 2 and 3; in the Y direction by vector plane numbers 1 and 3 and in the Z direction by vector plane numbers 1 and 3 also.

Then this zone will be defined (following X(θ), Y(R), Z order) as:

ZONEDEF name_of_the_zone 2 3 1 3 1 3

- **X vector zone numbering**



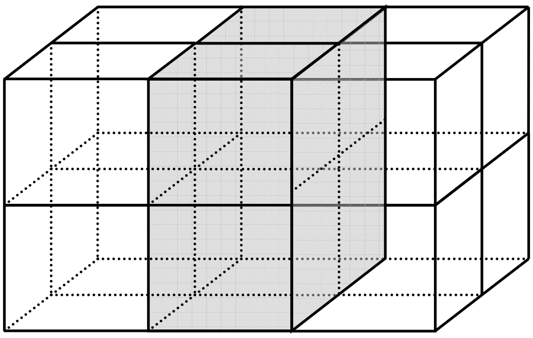
In this case the zone is the assembly of X vector nodes 3 and 4. It is delimited by vector plane 2 in X direction, by vector planes 1 and 2 in Y direction and by vector planes 1 and 3 in Z direction

ZONEDEFX name_of_the_zone 2 2 1 2 1 3

The same logic applies for other Y or Z vector zones.

- **Overall zone definition**

  Consider the previous defined zone: 2 3 1 3 1 3:



Using an overall zone definition four zones are created:

One scalar zone *zone_s* containing the scalar nodes: 3, 4, 9 and 10

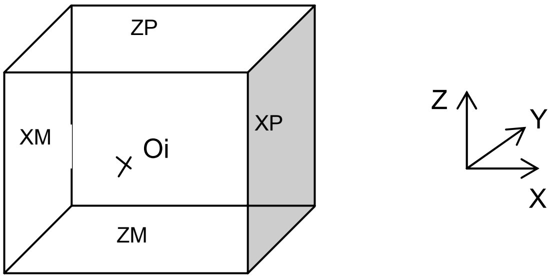
One X vector zone *zone_vx* containing the X vector nodes: 3, 4, 5, 6, 11, 12, 13 and 14

One Y vector zone *zone_vy* containing the Y vector nodes: 3, 4, 9, 10, 15 and 16

One Z vector zone *zone_vz* containing the Z vector nodes: 4, 5, 6, 13, 14 and 15

ZONEDEFG zone_s zone_vx zone_vy zone_vz 2 3 1 3 1 3

- **Mesh cell face**



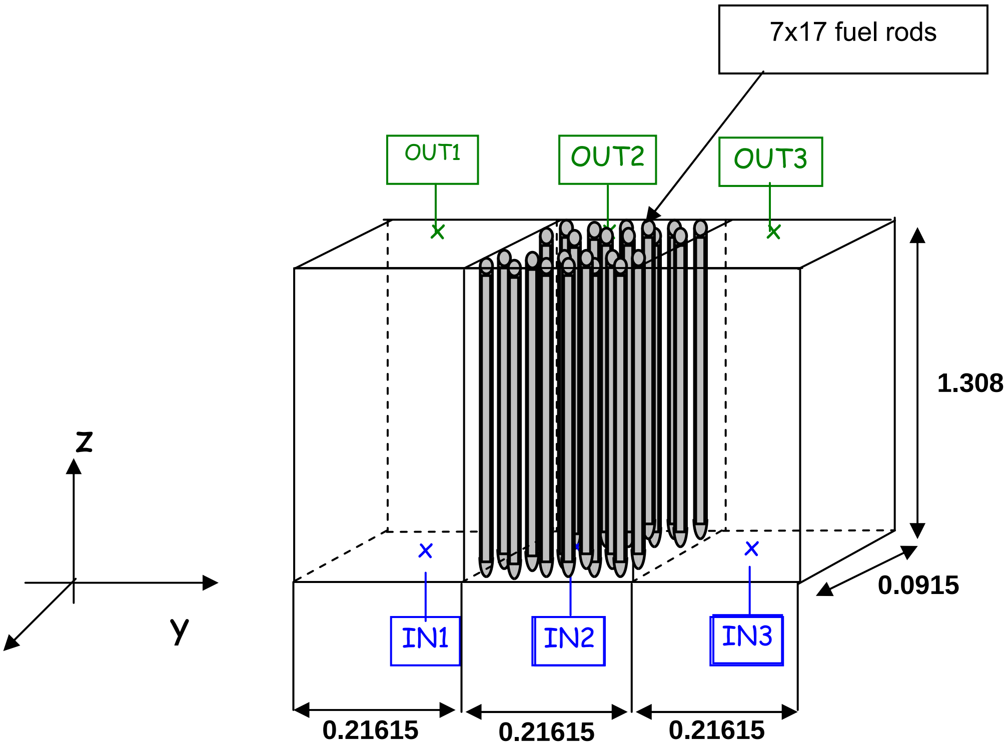
Oi: scalar node

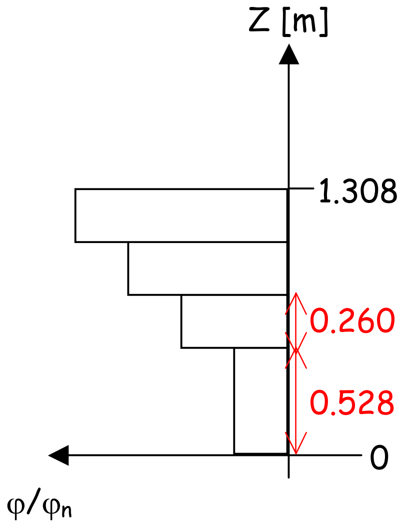
Rectangular CATHARE coordinates (see drawing below)

	XM	defines the junction on the left face X direction for a scalar node,
or	**XP**	defines the junction on the right face X direction for a scalar node,
or	**YM**	defines the junction on the left face Y direction for a scalar node,
or	**YP**	defines the junction on the right face Y direction for a scalar node,
or	**ZM**	defines the junction on the left face Z direction for a scalar node,
or	**ZP**	defines the junction on the right face Z direction for a scalar node

APPENDIX 6 Example of CATHARE THREED element definitions

The aim of this section is to describe step by step the definition of a 3D element. Consider the following 3D element. It is bounded by impermeable walls. The central part of the domain is fitted with a 7x17 heater rod assembly, electrically supplied. The axial distribution of the heat flux is presented below (axial peaking factor versus elevation). On both sides of the central part, the flow domain is an open domain with no structure.





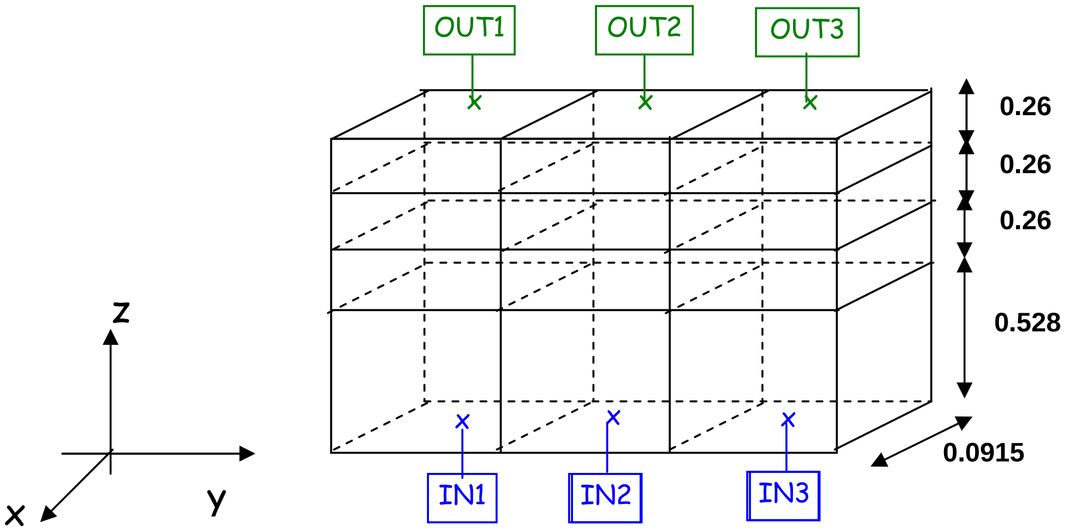
Axial peaking factor vs elevation

THREED

At the beginning, the name of the 3D element as well as the name and the number of the junctions must be defined. As there is only one junction possible per mesh face, three junctions must be used for the inlet and three junctions for the outlet.

PERC3D = **THREED**
IN1 **USTREAM**
IN2 **USTREAM**
IN3 **USTREAM**
OUT1 **DSTREAM**
OUT2 **DSTREAM**
OUT3 **DSTREAM**;

MESH



Rectangular coordinates are well suited to the geometry studied. As the phenomena are mainly 2D, only one mesh is used in the x-direction. The three zones identified in the domain are represented by three meshes in the y-direction. Then the vertical meshing is deduced from the vertical distribution of the heat flux in the heater rods.

MESH PERC3D
  COORDINA RECTANGL
  MESHING X 1
  MESHING Y 3
  MESHING Z 4
  MESHSIZE X LIST_DX 0.0915
  MESHSIZE Y LIST_DY 0.21615 0.21615 0.21615
  MESHSIZE Z LIST_DZ 0.528 0.26 0.26 0.26
  GRAVITY X 0.0
  GRAVITY Y 0.0
  GRAVITY Z –9.81;

CONNECT

The mesh faces where the junctions have to be connected must be defined. The connections are located on the boundary of the 3D element, so they are declared as external.

CONNECT PERC3D
  EXTERNAL BOUNDARY IN1 **ZM** 1 (using absolute numbering, see I.1.2)
Or
  EXTERNAL BOUNDARY IN1 **AZM** 1 2 1 2 1 2 (using axis numbering, see I.1.3)

```
      EXTERNAL BOUNDARY IN2 ZM 5
Or
      EXTERNAL BOUNDARY IN2 AZM 1 2 2 3 1 2
      EXTERNAL BOUNDARY IN3 ZM 9
      EXTERNAL BOUNDARY OUT1 ZP 4
      EXTERNAL BOUNDARY OUT2 ZP 8
      EXTERNAL BOUNDARY OUT3 ZP 12
```

The flow domain is described here by means of the open-closed faces. It may be useful here to remember that the notion of walls in the 3D module has a specific sense. A 3D domain contains solid structures which are described in the 3D module by three different objects:

- impermeable walls, in terms of fluid flow, are described by means of open/closed edges (CONNECT directive),
- internal solid structures are taken into account by means of fluid volume and fluid surface (GEOM directive),
- thermal walls are defined in the WALL3D operator.

Here the input deck uses default values in order to minimise the data volume. Only the open faces are defined explicitly.

```
EDGETYPE
      XEDGE DEFAULT CLOSED
      YEDGE DEFAULT CLOSED
            SEGMENT 5 12 VALUE OPEN
Or
            ZONEDEFY PERCY1 1 2 2 2 1 5
            ZONEDEFY PERCY2 1 2 3 3 1 5
            FYZONE PERCY1 OPEN
            FYZONE PERCY2 OPEN
ZEDGE DEFAULT OPEN;
```

GEOM

This key word enables the definition of fluid volume and area. It is recommended to describe the element in terms of homogeneous zones, which present common characteristics. So, in this example, the lateral zones without any structure represent only one zone. The central part of the domain is then considered in its totality to calculate the corresponding fluid volume.

- Volume of fluid per mesh cell (GEOM VOLUME)

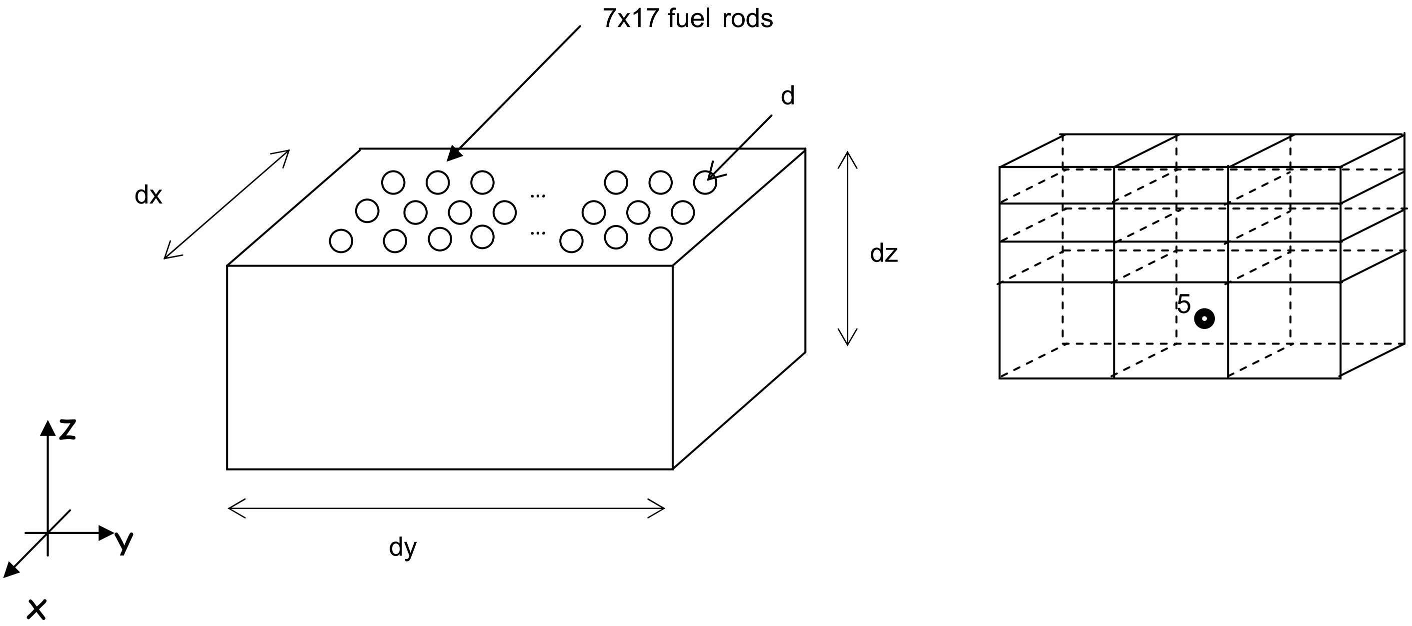
The volume porosity ε_v is defined as follows:

$$\varepsilon_v = \frac{V_{fluid}}{V_{mesh}} = 1 - \frac{V_{structures}}{V_{mesh}}$$

Application to the example:

- fluid volume of meshes free of solid structures:
$V_{fluid} = V_{mesh} = dx*dy*dz$

- fluid volume of mesh cell 5:



$V_{fluid} = dx*dy*dz - (7*17*pi*d^2/4*dz)$
dx = 0.0915 m
dy = 0.21615 m
dz = 0.528 m
d = 0.00945 m

$\Rightarrow V_{fluid} = 0.0060264\ m^3$

GEOM PERC3D
VOLUME LISTVOL
0.010442 0.0051422 0.0051422 0.0051422 0.0060264 0.0029675 0.0029675 0.0029675 . … (12 values)

Another highly recommended possibility is the use of the notion of porosity. All meshes with the heating rods have the same porosity. So they are easier to declare in the input deck. The calculation of volume porosity is detailed below.

- volume porosity of meshes free of solid structures:
$\varepsilon_V = 1.0$

- volume porosity of mesh cell 5:

$\varepsilon_V = V_{fluid} / V_{mesh}$
$= V_{fluid} / (dx*dy*dz)$
$\Rightarrow \varepsilon_V = 0.00603 / 0.01044 = 0.577$

VOLUME GEOMETRI POROS DEFAULT 1.0
SEGMENT 5 8 **VALUE** 0.577

Or (using the zone definition)
ZONEDEF PERC1 1 2 2 3 1 5
VOLUME GEOMETRI POROS DEFAULT 1.0
ZONE PERC1 0.577

- Fluid area per mesh cell face (GEOM AREAEDGE)

The surface porosity ε_s is defined as follows:

$$\varepsilon_s = \frac{S_{fluid}}{S_{mesh\_face}} = 1 - \frac{S_{structures}}{S_{mesh\_face}}$$

Remark:
The surface porosity may be taken equal to the volume porosity in two cases:

- the surface porosity cannot be easily computed,
- the surface porosity is not representative of a mean description of the flow.

Application to the example:

X-faces: no flow, $S_{fluid} = 0$

Z-faces:
- mesh with no solid structures: $S_{fluid} = dy*dx$, $\varepsilon_{SV} = 1.0$
- mesh with fuel rods: $S_{fluid} = dy*dx - (17*7*\pi*d^2/4)$

$$\varepsilon_{sZ} = \frac{S_{fluid}}{dx \cdot dy}$$

Note that $\varepsilon_{sZ} = \varepsilon_v$

Y-faces: In this case, it was decided to use the notion of porosity and to take $\varepsilon_{sY} = \varepsilon_v$. This choice it is possible to have a velocity which is characteristic of the average velocity and does not depend on the mesh network.

AREAEDGE X GEOMETRI POROS DEFAULT 0.0

AREAEDGE Z GEOMETRI POROS DEFAULT 1.0
SEGMENT 6 10 **VALUE** 0.577

AREAEDGE Y GEOMETRI POROS DEFAULT 0.0
SEGMENT 5 12 **VALUE** 0.577;

PHYSCALE
As in the GEOM directive, it is advisable to describe the element in terms of homogeneous zones, which present common characteristics.

- porous domain, i.e. numerous internal structures: the core.
The *volume hydraulic diameter* is computed as

$$D_{hV}(i) = 4\,\frac{V_{fluid}(i)}{S_{friction}(i)}$$

where $V_{fluid}(i)$ is the scalar mesh fluid volume and $S_{friction}(i)$ the friction area inside the scalar mesh
The *edge hydraulic diameter* is computed mesh by mesh as follows, whenever possible:

$$D_{hs}(i) = 4\,\frac{S_{fluid}(i)}{P_{friction}(i)}$$

where $S_{fluid}(i)$ is the flow section and $P_{friction}$ the friction perimeter. When it is not possible, the volume hydraulic diameter is imposed.

- <u>nearly open domain</u>, i.e. some internal structures: the lower plenum, the upper plenum, the upper head.The main idea is to consider the domain as totally homogeneous without any empty mesh (without internal structure). Therefore both hydraulic lengths, volume and edge lengths, are computed considering total quantities: total fluid volume $V_{fluid,tot}$, total friction area $S_{friction,tot}$,

$$D_{hv}(i) = D_{hs}(i) = 4\,\frac{V_{fluid,tot}}{S_{friction,tot}}$$

- <u>specific case of the annular downcomer</u>
Both hydraulic lengths, volume and edge lengths, are sets as being equal to twice the annular downcomer width e:

$D_{hv}(i) = D_{hs}(i) = 2e$

Application to the example:

- Hydraulic scale / mesh cell volume:

- meshes free of solid structures: *dhcell* is taken to be very jigh in order to adapt the closure laws to the case of an open domain.

- volume porosity of the mesh cell 5:

$V_{fluid} = dx*dy*dz – (17*7*\pi*d^2/4*dz)$
$S_{friction} = 17*7*dz*\pi*d + dz*dx + 2*dz*dy$
$\Rightarrow$ dhcell(1) = 0.01127

```
PHYSCALE PERC3D
DHCELL DEFAULT 10000.0
SEGMENT 5 8 VALUE 0.01127
```

- Hydraulic scale / mesh cell face:

<u>X faces</u>: not used (no flow): dhedge = 0
<u>Y faces:</u>
- meshes free of solid structures: dhedge = 10000.0
- *Example:* Y face number 5 (id. for all the other Y faces), the hydraulic scale of the mesh cell face cannot be easily computed, so the volume hydraulic diameter is imposed.

<u>Z faces:</u>
- meshes free of solid structures: dhedge = 10000.0
- *Example:* Z face number 6

$S_{fluid} = dy*dx – (17*7*\pi*d^2/4)$
$P_{friction} = 17*7*\pi*d+2*dy+dx$

$S_{fluid} = 0.01143\ m^2$
$P_{friction} = 4.05668\ m$

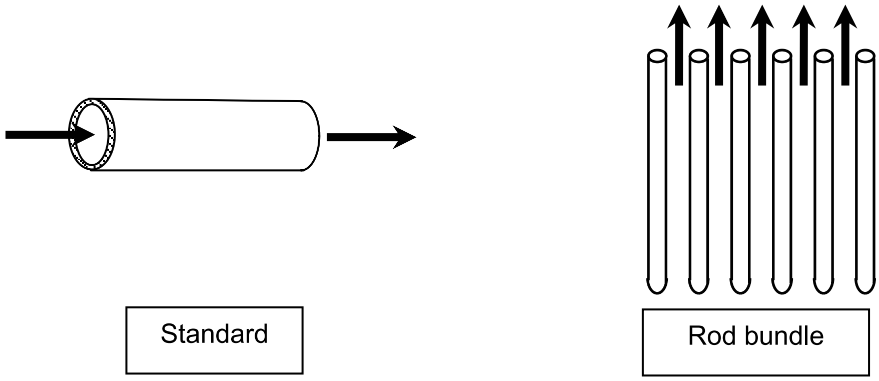
$\Rightarrow$ dhedgeZ(1) = 0.01127

DHEDGE X DEFAULT **0.0**
DHEDGE Y DEFAULT 0.0

```
SEGMENT 5 12 VALUE .01127
DHEDGE Z DEFAULT 10000.
SEGMENT 6 10 VALUE .01127;
```

HYDR

Two types of hydraulics are available, for both scalar (SCAGRID) and vector (VECGRID) correlations:



Standard hydraulics is used in meshes free of structures. The 7x17 heating rod assembly is typically described by the rod bundle hydraulics which is imposed on both scalar and vector nodes. Here again, the use of default values minimises the length of hydraulics definition.

```
HYDR PERC3D
        SCAGRID DEFAULT 0
        ZONE PERC1 1
        VECGRID X DEFAULT 0
        VECGRID Y DEFAULT 0
        FYZONE PERCY1 1
        FYZONE PERCY2 1
        VECGRID Z DEFAULT 0;
        FZZONE PERCZ1 1
```

SINGULAR

In the 3D module, the singular pressure drop coefficients are defined for a given spatial direction. In this example, the singular pressure drop coefficient due to a rod bundle has to be taken into account for the y-direction flow, which can be found in the Idelci'k handbook of singular pressure drop coefficients:
$K_C = 44.9$

```
SINGULAR PERC3D
        PLOSS POSITIVE Y
        DEFAULT 0.
        SEGMENT 5 12 VALUE 44.9
        PLOSS NEGATIVE Y
        DEFAULT 0.
        SEGMENT 5 12 VALUE 44.9
```

WALL3D

The 7x17 heater rod assembly is modelled with a single wall object. The definition of the wall 3D is similar to the 1D case. One difference is the introduction of HSURF, which represents the exchange surface per mesh cell. In this case, the exchange surface is expressed by:

$$HSURF = 7*17*\pi*d*dz$$

The exchange surface is then balanced between the meshes in the following real list:

```
RLHS = reallist  1.87523  0.92341  0.92341  0.92341;
```

```
FUELROD  =  WALL3D PERC3D  INTERNAL CYLINDER
MATERIAL NITRBORE   DATA 1 DIAM 0.           3.55D-3
TOPHET            DATA 1 DIAM 3.55D-3   5.25D-3
NITRBORE          DATA 2 DIAM 5.25D-3   6.94D-3      8.30D-3
INCON800          DATA 2 DIAM 8.30D-3   8.92D-3      9.50D-3
```

The position of the wall 3D is given using the scalar node numbering:

```
SEGMENT 5 8
HSURF RLHS SOURCE  MEDIUM 2 LAW LAWCOLDA VOLPOWER RLVP;
```

APPENDIX 7 Mesh definition and numbering in thermal sub-modules

Case of a standard WALL

Wall defined on an **AXIAL** element: A wall structure is connected to a hydraulic mesh or a segment delimited by two vector nodes of the element mesh. Then, the mesh is predefined axially by axial meshing. The user chooses only radial meshing.

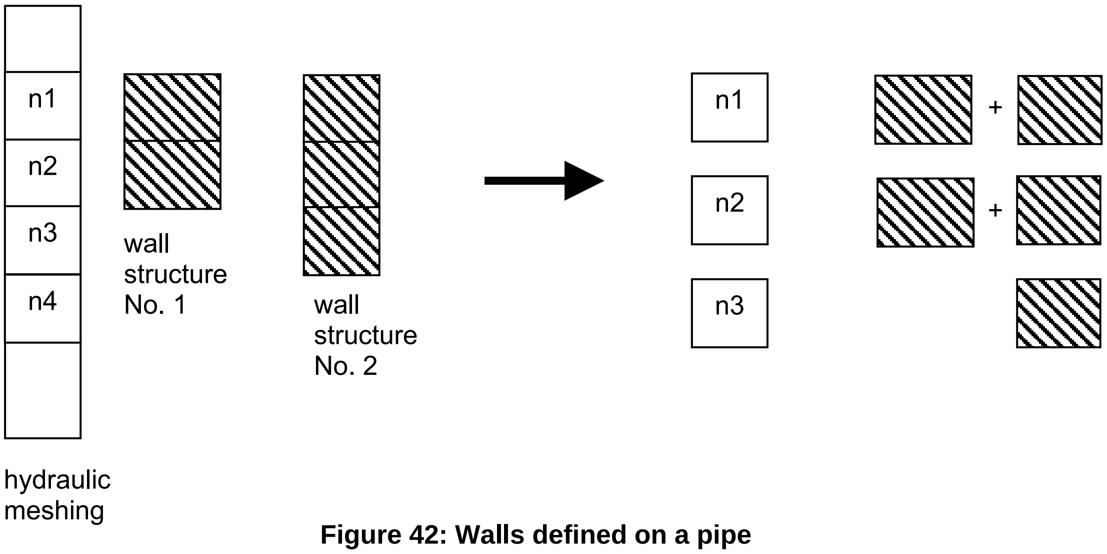


Figure 42: Walls defined on a pipe

The piece of wall connected to a single hydraulic mesh cell is called an "elementary wall". On the example above, two elementary walls are connected to hydraulic meshes n1 and n2, and wall structures No. 1 and 2 are respectively made up of two and 3 elementary walls. During post-processing, results will be read for each elementary wall, defined by the name of the wall structure and the “scalar” point on the wall, defining the axial elevation with respect to the wall bottom.

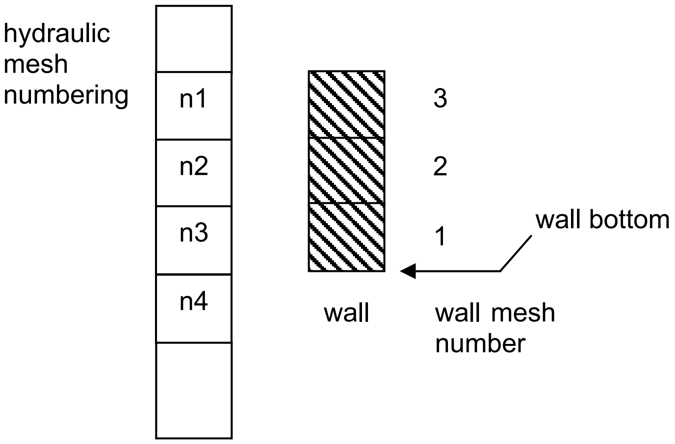


Figure 43: Wall mesh numbering

Wall defined on a **VOLUME** element: each wall structure is divided into two vertical meshes: the lower mesh in front of the lower fluid sub-module, the upper mesh in front of the upper fluid sub-module.
A volume wall structure may be shorter than the total height of the volume and can represent a vessel envelope or internal structure.

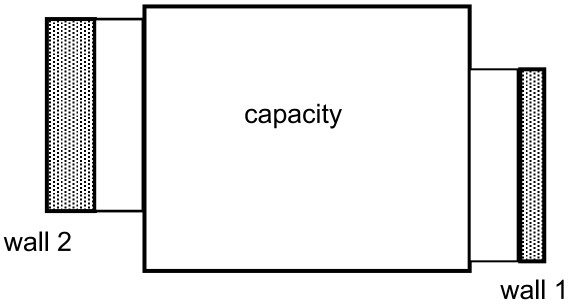


Figure 44: Walls defined on a capacity

One elementary wall will be considered with variables (temperatures, flux, etc.) computed axially at two nodes corresponding to the two scalar nodes of the volume. The key words INF and SUP can then be used to define these two wall meshes.

It should be borne in mind that the value given may not have any sense if the wall has no exchange surface with the sub-volume considered. For example WALLWETT(INF) for wall2 in Figure 45: Volume wall numbering.

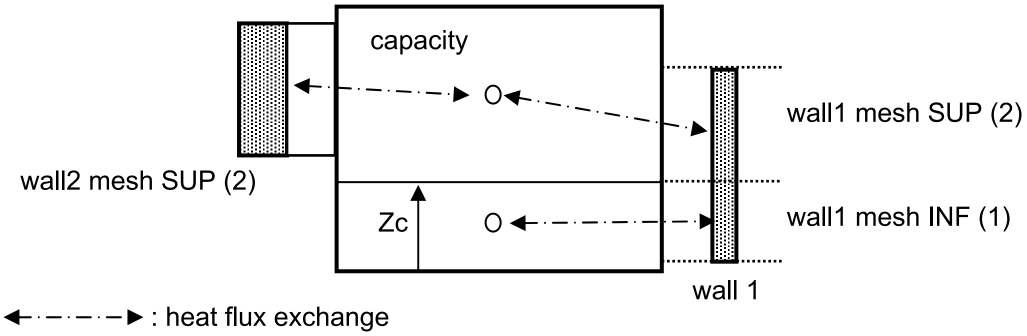
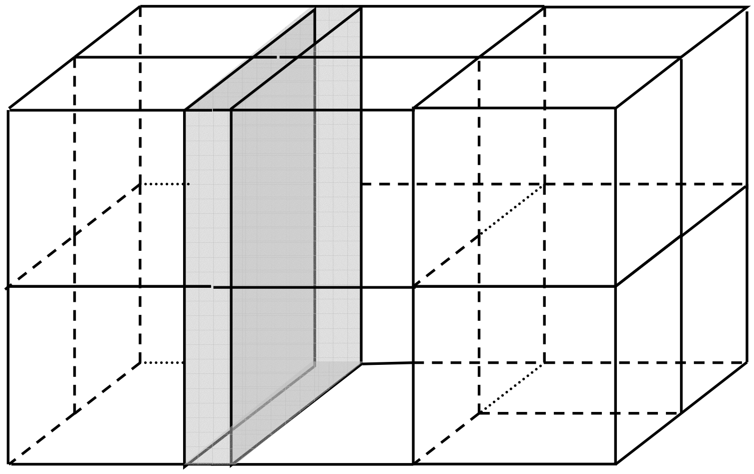


Figure 45: Volume wall numbering

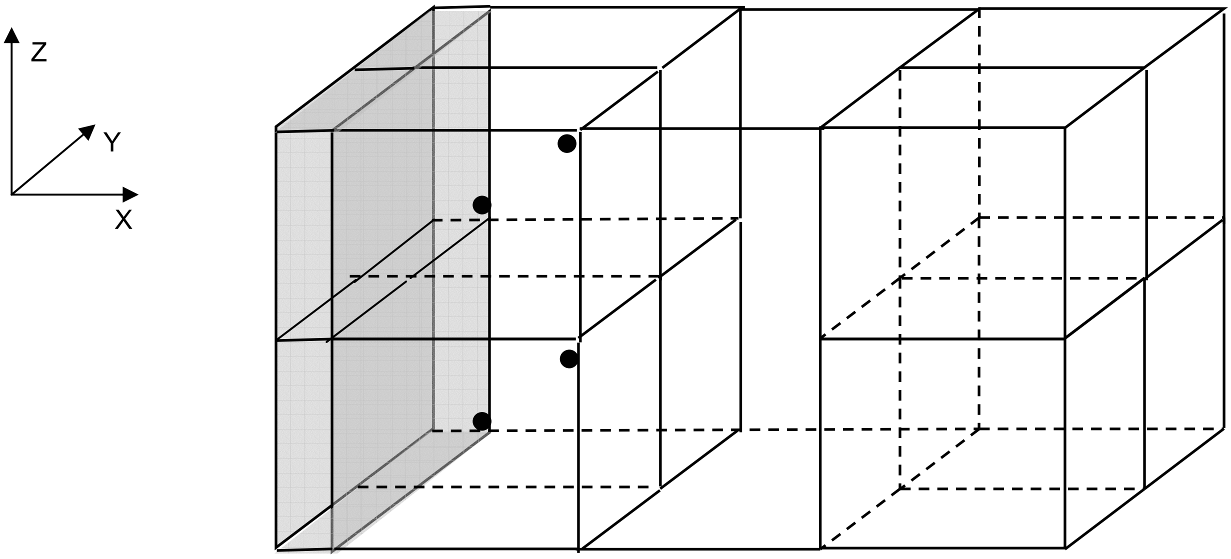
Wall defined in a **THREED** element: the wall structure is defined with respect to the hydraulic mesh of the THREED element on a segment delimited by two scalar nodes of the mesh in a selected direction. It is then made up of as many elementary walls as there are scalar meshes in this direction in the selected segment.

CAUTION: The user should remember that a WALL defined in a THREED element is a thermal structure and not an obstacle to the fluid flow.

There can be only one wall side wetted by the fluid (otherwise the user needs to define an exchanger.
All scalar nodes of the hydraulic mesh on the wall wet side, facing the vector faces corresponding to the selected WALL3D segment, will be taken into account in the exchanged flux calculation.
The following wall should be defined as an exchanger:



The following wall has to be defined as a WALL3D in the X direction and its position is defined by the knowledge of the facing THREED scalar nodes, i.e. by scalar absolute numbering (Refer to Appendix 5 for more information on THREED numbering):



● THREED scalar node exchanging flux with the WALL3D sub-module

```
wall1 = WALL3D threed_name EXTERNAL PLANE
…
SEGMENT X 1 8, etc.;
```

The above declaration in the input deck will define a wall in front of the four and only four scalar meshes 1, 2, 7, 8.

Case of an EXCHANGER

The following rules easily extend to the exchanger in a THREED element case.

As for the WALL in an AXIAL module, the EXCHANGER mesh is predefined by the AXIAL mesh (also called hydraulic mesh).
Consequently, there are two hydraulic meshes in this case (those opposite the EXCHANGER primary and secondary sides). The primary and secondary hydraulic meshes should then correspond.

- An exchanger object is defined on pairs of segments. Each segment is delimited by two vector points which must belong to the mesh of the elements. The segments may include several axial meshes if they are both either vertical or horizontal, and if each primary mesh is located opposite only one secondary mesh.

Example: on Figure 46: Primary and secondary mesh
P1, P2: points of the primary circuit element mesh
S1, S2: points of the secondary circuit element mesh.

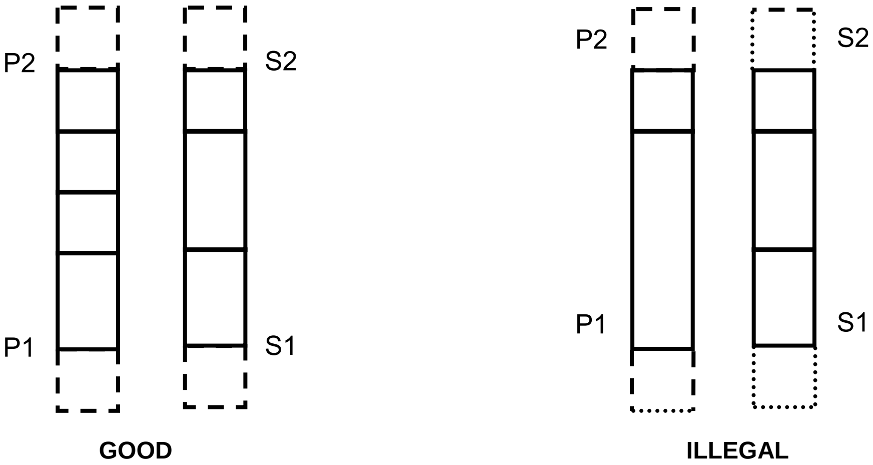


Figure 46: Primary and secondary mesh

The segment P1 P2 must correspond to the segment S1 S2 of the secondary.

- Each secondary axial mesh vector point must have a corresponding primary axial mesh vector point.

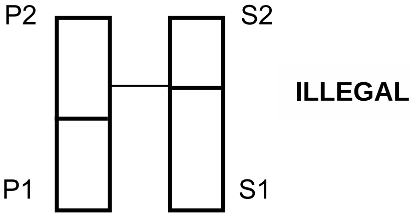


Figure 47: Primary and secondary mesh

- In the case of inclined or non-parallel segments, each segment must not include more than one elementary axial mesh on the secondary side, but may include one or more on the primary side.

P1 P2 $\leftrightarrow$ S1 S2
P2 P3 $\leftrightarrow$ S2 S3

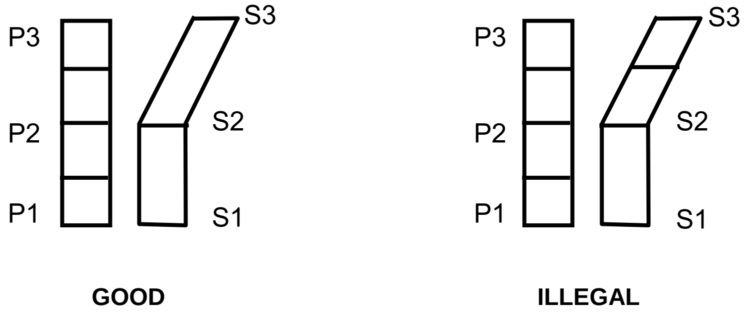


Figure 48: Primary and secondary mesh

The normal flow configurations are as follows:

- standard PWR U-tube steam generator:

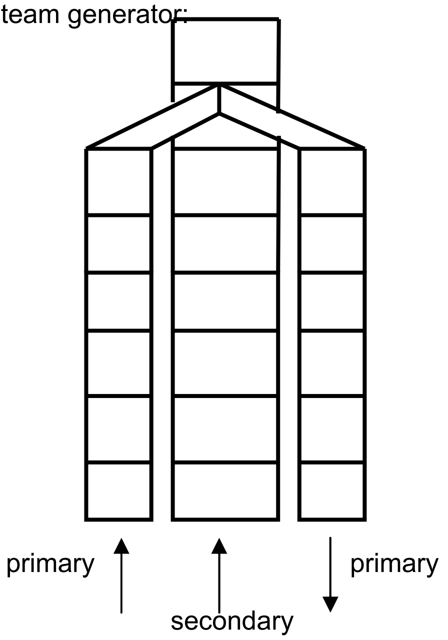


Figure 49: Standard PWR U-tube steam generator

- VVER steam generator:

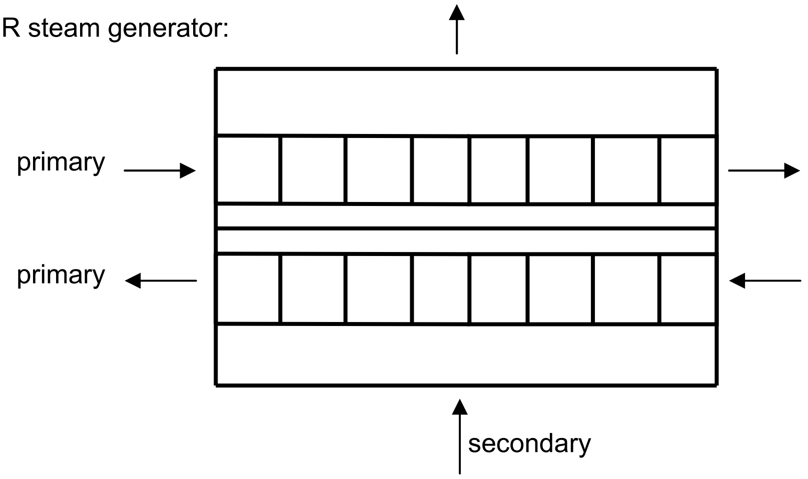
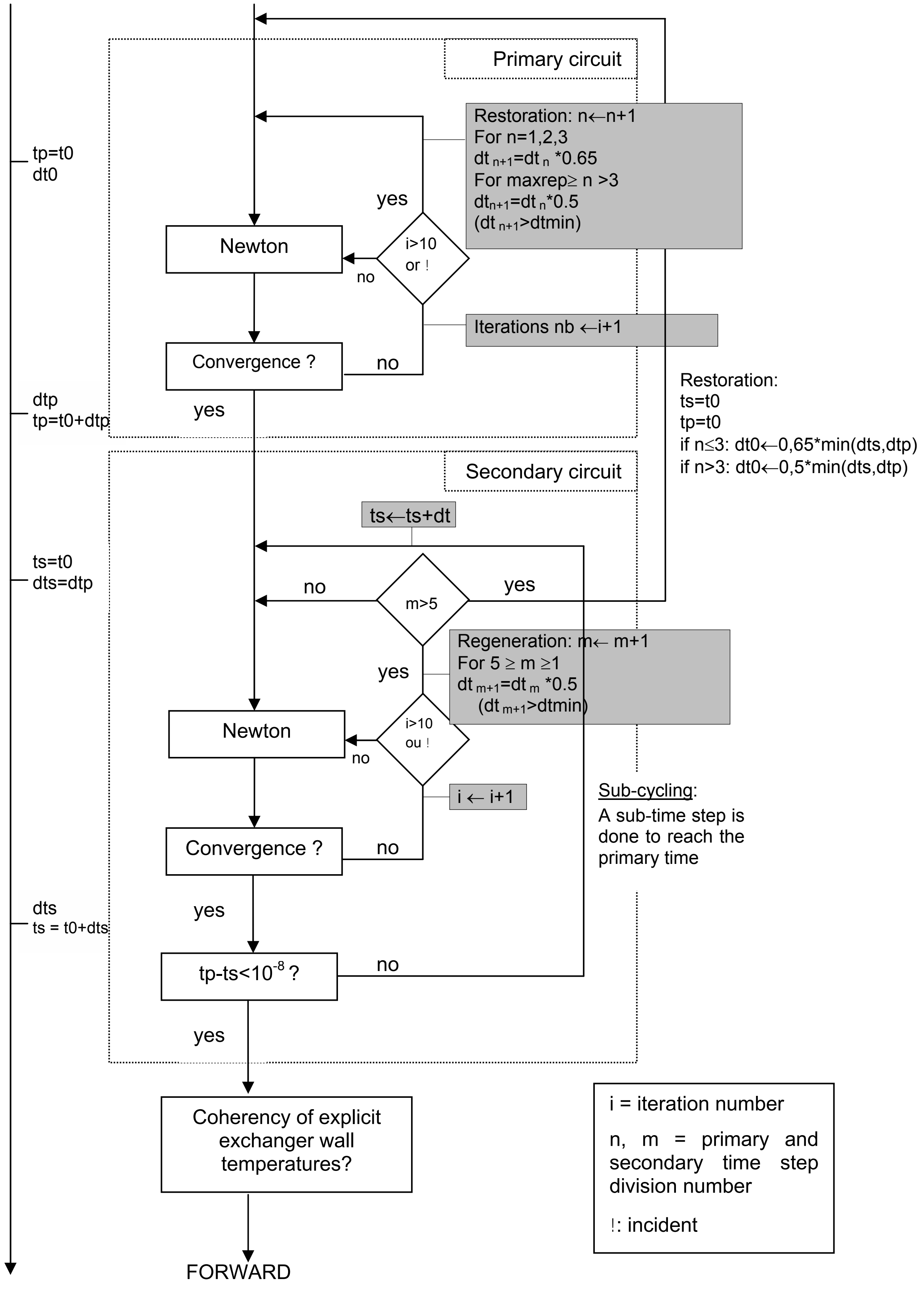


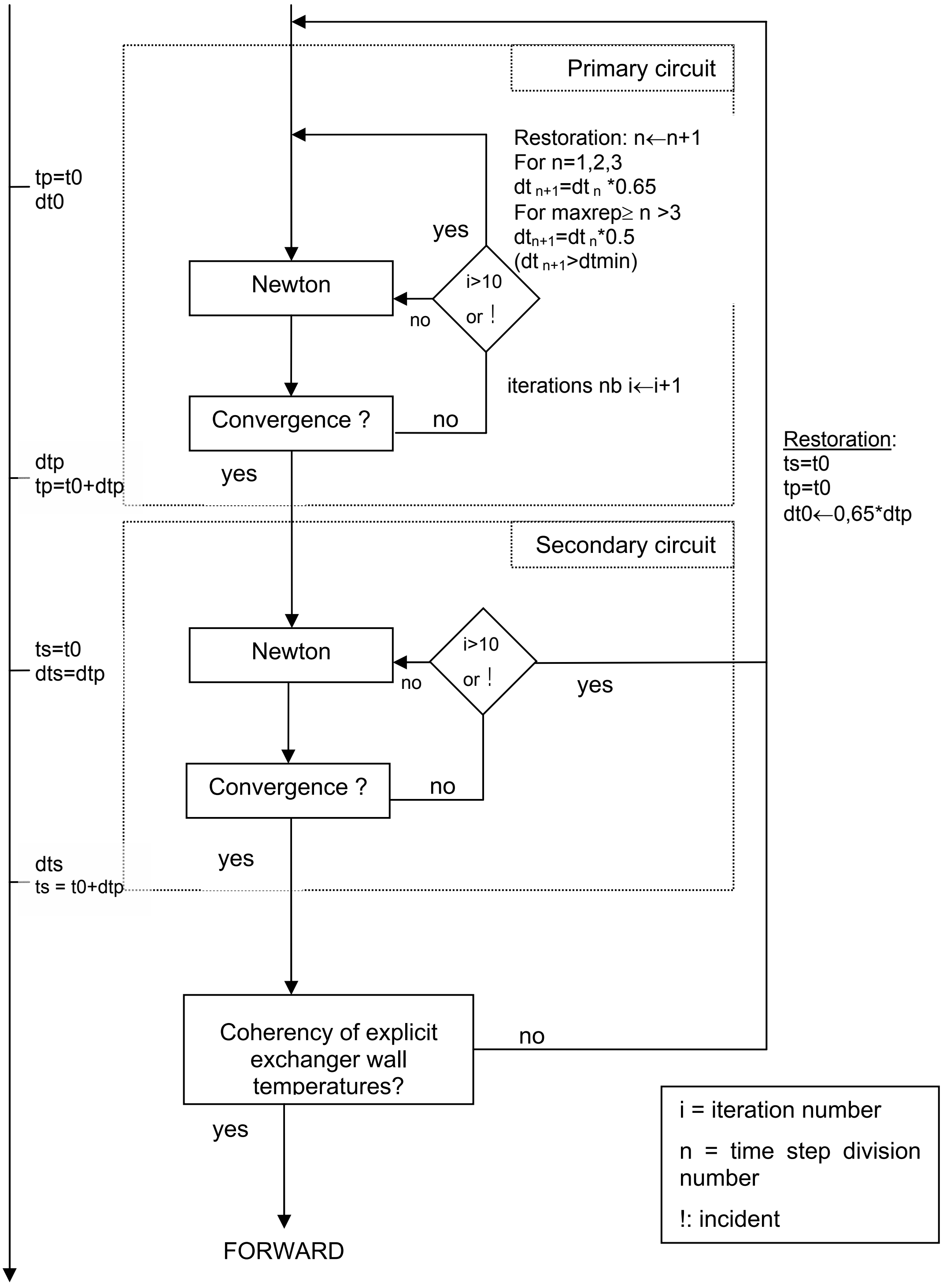
Figure 50: VVER steam generator

APPENDIX 8 Time step management

For a standard TRANSIENT time step:



For a stabilised transient time step:



APPENDIX 9 Core reactivity calculation

Reactivity in the CATHARE point kinetics model is the sum of four terms:

- external reactivity (due to dropping control rods– defined in SCRAM directive),
- reactivity depending on the Doppler effect,
- reactivity due to the moderator effect,
- and, if it has been defined in the circuit, reactivity due to boron.

For reactivity due to boron and moderator effect, there are two possible definitions:

1. In FUELPLAQ only or in FUEL/FUEL3D (NEUTRO key word) for the profile; FUELCHAR (NEUTRO key word) for the multiplicative definition,
2. In CORE operator definition: law definitions.

These are exclusive definitions (if the CORE operator is used the other is ignored).

During calculation, at the end of each time step (after thermal-hydraulic convergence) and for each fuel wall (FUELCHAR or FUELPLAQ), the following calculations are made.

Reactivity due to the Doppler effect calculation

- Calculation of an average volume temperature in the fuel wall for each axial mesh (r is used to define radial meshing):

$$\text{Tvolu (z)} = \frac{\sum_r \frac{dr(z)}{2}\big(T(r,z)+T(r+1,z)\big)}{\sum_r dr(z)}$$

- Depending on the kind of definition used:

1. $$\text{CODOP} = \frac{\sum_z coefd \times coefz(z) \times \sqrt{Tvolu_{kelvin}(z)} \times dz}{\sum_z dz}$$

where COEFD is the global coefficient defined in the FUELCHAR operator and COEFZ is calculated by normalising the profile defined in the FUEL/FUEL3D operator (respectively COEFKD and AZKDOP coefficient for a FUELPLAQ operator).

2. $$\text{CODOP} = \frac{\sum_z \big(A\ln(Tvolu_K(z)) + BTvolu_K(z) + C\,Tvolu_K^2(z)/2 + D\,Tvolu_K^3/3\big)\times dz}{\sum_z dz}$$

where A, B, C et D are the law coefficients defined in the CORE operator .

Reactivity due to the moderator effect calculation

Depending on the kind of definition used:

1. $$\text{COMOD} = \frac{\sum_z - coefm \times coefz(z) \times \big(\alpha(ih)\rho_g(ih) + (1-\alpha(ih))\rho_l(ih)\big)\times dz}{\sum_z dz}$$

where ih is the hydraulic mesh number corresponding to the z wall mesh,

COEFM the global coefficient defined in the FUELCHAR operator,
and COEFZ calculated by normalising the profile defined in the FUEL/FUEL3D operator
(respectively COEFKM and AZKMOD coefficients for a FUELPLAQ operator).

2. COMOD = $$\frac{\sum_z \left(A\ln(f(z)) + Bf(z) + C\, f^2(z)/2 + D\, f^3/3\right)\times dz}{\sum_z dz}$$

where f(z) = $\left(\alpha(ih)\rho_g(ih) + (1-\alpha(ih))\rho_l(ih)\right)$,
ih is the hydraulic mesh number corresponding to the z wall mesh,
and A, B, C et D are the law coefficients defined in the CORE operator.

Reactivity due to the boron effect calculation

The function method is the only one available:

COBOR = $$\frac{\sum_z \left(Af(z) + B\, f^2(z)/2\right)\times dz}{\sum_z dz}$$

where f(z) = $\left[Bore(ih)\right]$ is the bore concentration (ppm),
ih is the hydraulic mesh number corresponding to the z wall mesh,
and A et B are the law coefficients defined the in CORE operator.

Summation for the entire core

These coefficients are calculated for each fuel wall and transmitted to the core module.
Each kind of reactivity is summed for all fuel walls (FUELCHAR/ FUELPLAQ) taking into account their relative influence:

RODOP = $$\sum_{FUELCHARi} COEFQ_i \times CODOP_i$$

ROMOD = $$\sum_{FUELCHARi} COEFQ_i \times COMOD_i$$

ROBOR = $$\sum_{FUELCHARi} COEFQ_i \times COBOR_i$$

where COEFQi is a normalisation coefficient taking into account:
- the COEFPNEU coefficient (defined in FUELCHAR),
- the corresponding wall hydraulic element weight,
- and the fuel wall weight itself (given by NCHANNEL defined in FUELCHAR/FUELPLAQ).

Finally: ROTOT(t) = ROTOT(t=0) + ROEXT(t)
– (ROMOD(t) - ROMOD(t-dt))
– (RODOP(t) - RODOP(t-dt))
– (ROBOR(t) - ROBOR(t-dt))

APPENDIX 10 CATHARE standard printouts

For each element, its name, the time of calculation, the number of spent iterations, and the time step number are given. The factor dividing the wall inertia, possibly related to this element, is also specified. The measurement units (MPa, kg/s, kJ, etc.) are given next to the corresponding values.

TABLE 1: AXIAL module scalar points

I	:	number of the scalar point
Z	:	elevation of the scalar point (m)
P	:	pressure (MPa)
HL	:	liquid enthalpy (kJ/kg)
HLSP	:	saturation liquid enthalpy at pressure P (kJ/kg)
HG	:	gas enthalpy (kJ/kg)
HVSP	:	saturation vapour enthalpy at pressure P (kJ/kg)
AL	:	void fraction
TL	:	liquid temperature (°C)
TG	:	gas temperature (°C)
TSATP	:	saturation temperature at pressure P (°C)
ROL	:	liquid density (kg/m^3)
ROG	:	gas density (kg/m^3)
IQL	:	liquid-interface index value
IQV	:	gas-interface index value

If there is at least one non-condensable gas:

PV	:	partial vapour pressure (MPa)
TSATPV	:	saturation temperature at pressure PV (°C)
Xi	:	volume mass fraction for non-condensable gas $i \in [1;4]$

NB: The first and the last printed scalar points (their elevations and variables) are related to the axial junctions.

TABLE 2: AXIAL module vector points

I	:	vector point number
Z	:	vector point elevation (m)
VL	:	liquid velocity (m/s)
VG	:	gas velocity (m/s)
QL	:	liquid mass flowrate (kg/s)
QG	:	gas mass flowrate (kg/s)
QT	:	total mass flowrate (kg/s)
ITOVI	:	interfacial friction index value

TABLE 3: AXIAL module overall characteristics

- total, spatial, temporal, singular, wall and gravity pressure variations (MPa)
- total, liquid, gas masses (kg) and liquid, gas energies (J).
- water mass, water energy and non-condensable mass (if necessary) balance for the last time step.
- power balance during the last time step for the element and each wall

TABLE 4: Linked TEE branch scalar point

The printouts for the tee module (printouts for offtakes) are made after the printouts of the associated axial module (lateral branch).

I : number of the main axial scalar point where the tee is linked
Z : elevation of this scalar point on the main axial element (m)
COTPI : standardised elevation of the TEE branch (possible values 0, 0.5, 1)
P : pressure (MPa)
HL : liquid enthalpy (kJ/kg)
HLSP : saturation liquid enthalpy at pressure P (kJ/kg)
HG : gas enthalpy (kJ/kg)
HVSP : saturation gas enthalpy at pressure P (kJ/kg)
AL : void fraction
TL : liquid temperature (°C)
TG : gas temperature (°C)
TSATP : saturation temperature at pressure P (°C)
ROL : liquid density (kg/m^3)
ROG : gas density (kg/m^3)

If there is at least one non-condensable gas:
PV : partial vapour pressure (MPa)
TSATPV : saturation temperature at pressure PV (°C)
Xi : volume mass fraction for non-condensable gas i $\in$ [1;4]

TABLE 5: Linked TEE branch vector point

VL : liquid velocity (m/s)
VG : gas velocity (m/s)
QL : liquid mass flowrate (kg/s)
QG : gas mass flowrate (kg/s)
QT : total mass flowrate (kg/s)
ITOVI : interfacial friction index value
ECART : stratification rate in the main axial element

TABLE 6: VOLUME module overall characteristics

- volume (m^3) and its height (m)
- mixture level in the volume (m)
- initial mass, liquid, gas and total masses (kg)
- liquid and gas energies (J)
- exchange between the lower and the upper sub-volume (s/v)

mechanical effect:	fall of droplets	:	< 0	droplets from upper s/v to lower s/v (kg/s)
			> 0	no stratification in the volume
	bubble rise	:	> 0	bubbles from lower s/v to upper s/v (kg/s)
			< 0	no stratification in the volume
thermal effect:	vaporisation	:	> 0	lower sub-volume liquid is vaporised by overheated steam of the upper s/v (kg/s)
			< 0	steam of the upper s/v is condensed due to being sub-cooled
	condensation	:	< 0	condensation of steam of the upper s/v (kg/s)
			> 0	vaporisation of overheated liquid of the lower s/v

- water mass, water energy and non-condensable mass (if necessary) balance for the last time step (kg).
- power balance during the last time step for the element and each wall.

TABLE 7: VOLUME module upper and lower sub-volumes

Values are given for each sub-volume

P	:	pressure (MPa)
HL	:	liquid enthalpy (kJ/kg)
HLSP	:	saturation liquid enthalpy at pressure P (kJ/kg)
HG	:	gas enthalpy (kJ/kg)
HVSP	:	saturation vapour enthalpy at pressure P (kJ/kg)
TL	:	liquid temperature (°C)
TG	:	gas temperature (°C)
TSATP	:	saturation temperature at pressure P (°C)
ALFA	:	void fraction
VOLUME	:	sub-volume capacity (m^3)
MASS	:	mass in sub-volume (kg)
IQL	:	liquid-interface index value
IQV	:	gas-interface index value

If there is at least one non-condensable gas:

PV	:	partial vapour pressure (MPa)
TSATPV	:	saturation temperature at pressure PV (°C)
Xi	:	volume mass fraction for non-condensable gas i ∈ [1;4]

TABLE 8: VOLUME module ports (junctions)

These values are calculated in front of the junction at a distance of 2% (for revision 6.1) of the volume height in the case of a vertical junction. They depend on the volume height.

Ports	:	name of junction
P	:	junction pressure (MPa)
HL	:	junction liquid enthalpy (kJ/kg)
HLSP	:	junction saturation liquid enthalpy at pressure P (kJ/kg)
HG	:	junction gas enthalpy (kJ/kg)
HVSP	:	junction saturation vapour enthalpy at pressure P (kJ/kg)
AL	:	junction void fraction
TL	:	junction liquid temperature (°C)
TG	:	junction gas temperature (°C)
TSATP	:	saturation pressure at P (°C)
ROL	:	liquid density (kg/m^3)
ROG	:	gas density (kg/m^3)
VL	:	junction liquid velocity (m/s)
VG	:	junction gas velocity (m/s)
QL	:	liquid mass flowrate (kg/s)
QG	:	gas mass flowrate (kg/s)
QT	:	total mass flowrate (kg/s)
BETAL	:	liquid flowrate distribution coefficient
BETAG	:	gas flowrate distribution coefficient
BETA	:	influence coefficient of the junction on the lower sub-volume
KLPAT	:	liquid singular pressure loss coefficient
KVPAT	:	vapour singular pressure loss coefficient
DP SING	:	singular pressure loss (delta P, MPa)
ITOIZO	:	interfacial friction index

If there is at least one non-condensable gas:

PV	:	partial vapour pressure (MPa)
TSATPV	:	saturation temperature at pressure PV (°C)
Xi	:	junction volume mass fraction for non-condensable gas i ∈ [1;4]

The PRIN3D directive

This directive is used to define the plane used for printing scalar and vector variables (XZ plane or YZ plane). It allows the user to enable (value 1) /disable (value 0) the printout of the variables of the three-dimensional module.

```
PRIN3D PERC3D PLAN3D YZ GASDENS 0 LIQDENS 0 GASVXM 0 LIQVXM 0;
```

Plane for printing defined by the user

Printing of the variable GASVXM disabled by the user

TABLE 9: THREED module scalar variables

The main scalar variables printed by default in the listing file are: P, Hl, Hg, α, Xi. Additional variables can be printed: ρl, ρg, Tl, Tg, Tsat(Pv), Pv. They have the same meaning and measurement units as those of the AXIAL module.

Plane for printing defined by the user

```
PRESSURE        PLANYZ  N   1
N Z   4    3.01213D+05  3.01210D+05  3.01243D+05
N Z   3    3.03636D+05  3.03649D+05  3.03727D+05
N Z   2    3.06066D+05  3.06121D+05  3.06208D+05
N Z   1    3.09750D+05  3.09897D+05  3.09959D+05
             N Y 1        N Y 2        N Y 3
```

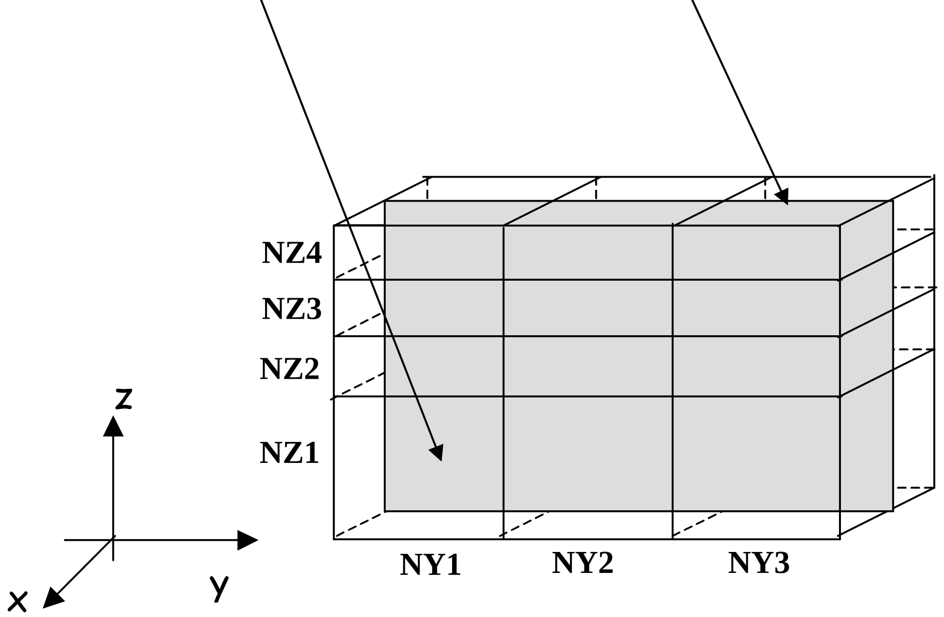


TABLE 10: THREED module vector variables

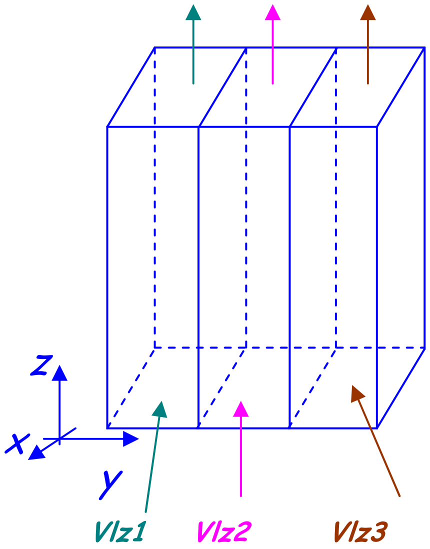
They have the same meaning and measurement units as those of the AXIAL module.

General form of the printout:

```
  VG PLANYZ     1
  Z   5          1.0696D-01  1.3185D-01 -1.0656D-01
Y   4   16  0.0000D+00  3.2126D-03 -3.7984D-02  0.0000D+00
  Z   4          6.3630D-02  1.2684D-01 -1.0656D-01
Y   3   15  0.0000D+00 -2.7026D-02 -5.8368D-02  0.0000D+00
  Z   3          4.4636D-02  1.5863D-01 -1.0676D-01
Y   2   14  0.0000D+00 -5.0991D-02 -6.1261D-02  0.0000D+00
  Z   2          9.0831D-03  1.4468D-01 -1.1157D-01
Y   1   13  0.0000D+00 -8.0941D-02 -5.2098D-02  0.0000D+00
  Z   1          3.0234D-09  5.2852D-09  5.2365D-09
```

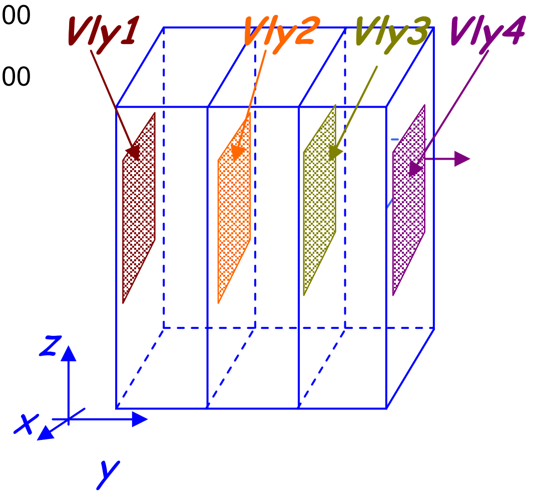
Z-faces

```
  Z   5          1.0696D-01  1.3185D-01 -1.0656D-01
Y   4   16  0.0000D+00  3.2126D-03 -3.7984D-02  0.0000D+00
  Z   4          6.3630D-02  1.2684D-01 -1.0656D-01
Y   3   15  0.0000D+00 -2.7026D-02 -5.8368D-02  0.0000D+00
  Z   3          4.4636D-02  1.5863D-01 -1.0676D-01
Y   2   14  0.0000D+00 -5.0991D-02 -6.1261D-02  0.0000D+00
  Z   2          9.0831D-03  1.4468D-01 -1.1157D-01
Y   1   13  0.0000D+00 -8.0941D-02 -5.2098D-02  0.0000D+00
  Z   1          3.0234D-09  5.2852D-09  5.2365D-09
                 Vlz1        Vlz2        Vlz3
  Vlz
```



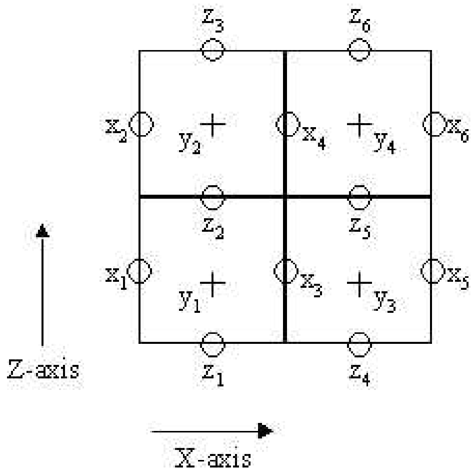
Y-faces

```
  Z   5          1.0696D-01  1.3185D-01 -1.0656D-01
Y   4   16  0.0000D+00  3.2126D-03 -3.7984D-02  0.0000D+00
  Z   4          6.3630D-02  1.2684D-01 -1.0656D-01
Y   3   15  0.0000D+00 -2.7026D-02 -5.8368D-02  0.0000D+00
  Z   3          4.4636D-02  1.5863D-01 -1.0676D-01
Y   2   14  0.0000D+00 -5.0991D-02 -6.1261D-02  0.0000D+00
  Z   2          9.0831D-03  1.4468D-01 -1.1157D-01
Y   1   13  0.0000D+00 -8.0941D-02 -5.2098D-02  0.0000D+00
  Z   1          3.0234D-09  5.2852D-09  5.2365D-09
  Vly     Vly1       Vly2       Vly3      Vly4
```



2D Example

Consider an easy example of a three-D mesh with only two mesh cells along the X-axis and two along the Z-axis. Four scalar values (S_1 to S_4, hidden by the y-vector points), 6 X-vector values (X_1 to X_6), 8 Y-vector (Y_1 to Y_4 for the visible face and Y_5 to Y_8) and 6 Z-vector values (Z_1 to Z_6) have to be defined.



Assuming that the reference plane for printout is XZ, the axes used for the printout are:
- horizontal : X-axis (or Y-axis if YZ is chosen as the reference plane)
- vertical : Z-axis

The scalar values are printed as follows for each slice along the Y-axis, in order to represent the same geometry as above:

```
VARIABLE_NAME REFERENCE_PLANE (here, PLANXZ)        NUMBER_OF_SLICE
NZ   2      S2   S4
NZ   1      S1   S3
```

The vector values are printed for each XZ plane:

```
VARIABLE_NAME REFERENCE_PLANE (here, PLANXZ)        NUMBER_OF_PLAN

                                   ⊗Y- axis
    Z   3               Z3                  Z6
X       2    X2                  X4                   X6
Y       2               Y2                  Y4
    Z   2               Z2                  Z5
X       1    X1                  X3                   X5
Y       1               Y1                  Y3
    Z   1               Z1                  Z4
```

TABLE 11: BCONDIT module

P	junction pressure (MPa)
HL	junction liquid enthalpy (kJ/kg)
HLSP	junction saturation liquid enthalpy at pressure P (kJ/kg)
HG	junction gas enthalpy (kJ/kg)
HVSP	junction saturation vapour enthalpy at pressure P (kJ/kg)
AL	junction void fraction
TL	junction liquid temperature (°C)
TG	junction gas temperature (°C)
TSATP	saturation temperature at pressure P (°C)
ROL	liquid density (kg/m^3)
ROG	gas density (kg/m^3)

VL	junction liquid velocity (m/s)
VG	junction gas velocity (m/s)
QL	liquid flowrate (kg/s)
QG	gas flowrate (kg/s)
QT	total flowrate (kg/s)

If there is at least one non-condensable gas:

PV	partial vapour pressure (MPa)
TSATPV	saturation temperature at pressure PV (°C)
Xi	volume mass fraction for non-condensable gas $i \in [1;4]$

TABLE 12: RUPTURE module

For each variable printed out, two data items are given. The order of these data items corresponds to the order of the junctions in the definition of the rupture module in the input data deck.

P	pressure (MPa)
HL	liquid enthalpy (kJ/kg)
HLSP	saturation liquid enthalpy at pressure P (kJ/kg)
HG	gas enthalpy (kJ/kg)
HVSP	saturation gas enthalpy at pressure P (kJ/kg)
AL	void fraction
TL	liquid temperature (°C)
TG	gas temperature (°C)
TSATP	saturation temperature at pressure P (°C)
ROL	liquid density (kg/m^3)
ROG	gas density (kg/m^3)

If there is at least one non-condensable gas:

PV	partial vapour pressure (MPa)
TSATPV	saturation temperature at pressure PV (°C)
Xi	volume mass fraction for non-condensable gas $i \in [1;4]$

VL	liquid velocity (m/s)
VG	gas velocity (m/s)
QL	liquid mass flowrate (kg/s)
QG	gas mass flowrate (kg/s)
QT	total mass flowrate (kg/s)

TABLE 13: THERMAL STRUCTURE general variables

Power balances are given together for all thermal structures belonging to the same main module, at the end of the main module printouts.

TABLE 14: THERMAL STRUCTURE wall scalar point

I	:	- case of a WALL in an AXIAL or a THREED element: the number of the hydraulic scalar point facing the wall mesh - case of a WALL in a VOLUME element: the sub-volume index facing the wall mesh (1=lower sub-volume; 2=upper sub-volume)
NB	:	number of the scalar point of the wall
IND	:	heat exchange zone index value

W.TWALL	:	wet wall temperature (mesh contacting the fluid) (°C)
D.TWALL	:	dry wall temperature (mesh far from the fluid) (°C)
PHIL	:	wall-liquid heat flux (W/m^2)
PHIV	:	wall-vapour heat flux (W/m^2)
PHIWI	:	wall-interface heat flux (W/m^2)
PHIFC	:	forced convection heat flux (W/m^2)
PHINE	:	nucleate boiling heat flux (W/m^2)
PHICR	:	critical heat flux (W/m^2)
PHIFB	:	film boiling heat flux (W/m^2)
PHIRL	:	liquid radiation heat flux (W/m^2)
PHIRV	:	gas radiation heat flux (W/m^2)
PHIVS	:	supplementary heat flux upstream of quench front for reflooding calculation (W/m^2)

TEXT	:	external temperature for the exchanger (°C)
HEXT	:	heat exchange coefficient with the external side (W/m²/°C)
PHIEXT	:	heat flux towards the external side (W/m^2)

TABLE 15: EXCHANGER

See printout for the wall.

TABLE 16: FUELCHAR or FUELPLAQ

Fuel wall printouts begin with the same information as standard wall (see above)

TABLE 17: FUELCHAR or FUELPLAQ power profile variables

I	:	number of the hydraulic scalar point facing the wall mesh
NB	:	number of the scalar point of the wall
NRPROFIL	:	normalised coefficient of the neutronic (non-residual) power profile
RESPROFIL	:	normalised coefficient of the residual power profile
QNEUT	:	neutronic (non-residual) power injected to the fluid (W)

TABLE 18: FUELCHAR or FUELPLAQ general variables

The amount of the linear power is given as well as its distribution in neutron-to-moderator, neutron-to-fuel, residual and oxidation terms in W/cm. The power distribution coefficient (XNEUT) value is also given.

TABLE 19: FUELCHAR or FUELPLAQ scalar point

NWA	: number of the wall structure
JZ	: number of the scalar point
TUI	: internal UO_2 temperature (°C)
TUE	: external UO_2 temperature (°C)
TIO	: internal oxide temperature (°C)
TIC	: internal clad temperature (°C)
TM	: mean clad temperature (°C)
TEC	: external clad temperature (°C)
TEO	: external oxide temperature (°C)
THIO	: thickness of internal oxide layer (mm)
RM	: mean clad radius (mm)
THEO	: thickness of external oxide layer (mm)
GAP	: width of gap (mm)
CLADDIN	: clad thickness (mm)
RUEXT	: external UO_2 radius (mm)
ALFRAC	: fraction of alpha phase
ALFRMI	: minimum fraction of alpha phase
POWOX	: linear power generated in UO_2 (W/cm)
PHIUC	: linear heat flux from UO_2 to CLAD (W/cm)
POWIO	: linear internal oxidation power (W/cm)
POWEO	: linear external oxidation power (W/cm)
PHICF	: linear heat flux from CLAD to fluid (W/cm)
PHIFL	: part of linear power generated in UO_2 and released into the fluid
conduction	: part of conduction in the flux from UO_2 to cladding (%)
radiation	: part of radiation in the flux from UO_2 to cladding (%)
SIGEL	: true stress in the cladding (bar) (not used)
SIGCR	: engineering stress in the cladding (bar) (not used)
SIGRUP	: clad rupture stress (bar)
DR/RO	: clad strain from the beginning (%)
R-RP	: clad radius variation from previous time step (μm)
DILAT	: part of strain from the beginning due to thermal expansion
CREEP	: part of strain from the beginning due to creep
ELSTR	: part of strain from the beginning due to elastic stress
DRTOT	: total deformation form the beginning

TABLE 20: REFLCHAR or REFLCH3D

PERIOD directive independent printouts

Reflooding events are given in the standard listing when they occur:

- Transition from 1D to 2D physical model of reflooding

Example:

```
 BOTTOM-UP REFLOOD MODULE = EAUC
 1D TO 2D REFLOODING
 FIRST POSSIBLE FLIP-FLOP
```

- Beginning of 2D reflooding
- Example:

```
BOTTOM-UP REFLOOD MODULE EAUC       AT TIME(S):   32.36495
      START OF THE 2D REFLOODING
      USE OF A TEMPERATURE CRITERION
        ZFT       :    0.1532
        TWALL(DO) :   303.9828
        TMFS (DO) :   297.8676
```

ZFT is the quench front level with respect to the bottom of the hydraulic module (m),
TWALL is the wall temperature (°C) downstream (DO) of the quench front level,
TMFS is the minimum film stable temperature (°C) downstream (DO) of the quench front level.

- Ending a wall mesh reflooding

When the quench front level reaches the next wall mesh, the following printout is given:

```
  REFLOOD MODULE =EAUC
 BOTTOM-TOP REFLOOD: NEW SCALAR DOWNSTREAM OF QF:   4;  OLD ONE:   3
 WALL TEMPERATURES IN THE REFLOODED NODE:


  308.4  298.7  284.9  266.5  242.4  219.0  160.6  160.6  147.2
  147.2
```

This line gives the new radial temperature profile for the reflooded wall mesh (in this example, the third mesh cell) interpolated from the refined reflood meshing.

PERIOD directive dependent printouts

Everything including measurement units is clearly written in the listing:

- Case of a 1D quenc**h** front

The following information is given:

- type of reflooding (bottom-up or top-down),
- quench front position (m) and its velocity (m/s),
- temperature (°C), location (m), and index value of the BO (burn out) front,
- temperature (°C), location (m), and index value of the MFS (minimum film stable) front.

- Case of a 2D quench front

- type of reflooding (bottom-up or top-down),
- quench front position (m) and its velocity (m/s),
- CK, the K2 law coefficient,
- total power **emitted to the reflooded element** (W) and the power for boiling (W),
- burn out temperature (°C),
- mass flux at point upstream (kg/m²/s) and the energy flux at point downstream (W/m²),
- quality at the quench front,
- distance between the quench front and the downstream scalar point.

Then power information is given (HEAT RELEASE FROM THE FINE MESH)
For each axial refined mesh of the reflood meshing is given:

- IZP, the number of the reflooding mesh point,
- ZPS, the altitude of the point in the reflooding mesh,
- total heat flow (W),
- heat flux computed at the end of the steady state (W/m²),
- heat flux given to fluid (W/m²).

Then temperatures are given:

```
   TEMPERATURES IN THE 2-D FINE MESH
     1      2      3      4      5      6      7      8      9     10     11     12
  289.8  285.9  274.1  253.9  226.0  192.6  153.6  153.6  150.5  147.6  144.9  144.9
  296.4  292.9  281.8  262.2  233.5  197.5  154.8  154.8  151.5  148.4  145.3  145.3
  305.8  303.0  294.0  277.0  249.2  208.7  157.9  157.9  153.9  150.0  146.3  146.3
  etc.
```

The values given from the left to the right correspond to temperatures from the fuel wall centre to the external side (assuming a cylindrical fuel rod).
The first line is for the first mesh line in the axial direction (the bottom of the reflood mesh for a bottom–up reflood module).

TABLE 21: CORE MODULE

This information is given at the last position of the circuit to which the core module belongs, just before printouts of circuit junctions.

Nominal P. : instantaneous power (W)
Residual P. : residual power generated by the fission products inside the fuel (W)
Effective P. : effective power (W) = nominal power + residual power
Fission P. : power generated by the fission inside the fuel (W)

Reactivity ($, or pcm/BETA):
initial : initial reactivity (t=0 s)
control rods : reactivity introduced by control rod movement (simulated by the SCRAM directive in the input data deck)
void effect : reactivity introduced by coolant density
Doppler effect : reactivity introduced by fuel temperature
boron effect : reactivity introduced by boron poison concentration
total : sum of the five reactivity terms

TABLE 22: RADIOCHEMICAL MODULE

The radiochemical information is printed for each scalar point of the axial, volume, or threed element. For the whole circuit, the mass concentration or activity balance for the last time step and for the whole transient (from initialisation) is given. For zone modules, the balance is given from initialisation.

Chemical components
LIQUID : liquid concentration (ppm)
GAS : gas concentration (ppm)
Radioactive components
LIQUID : activity per mass unit in the liquid phase (GBq/kg)
GAS : activity per mass unit in the gas phase (GBq/kg)

TABLE 23: ZONE

For each zone defined inside a three-D element (BILAN3D), or defined zone module, the following information is printed:

- fluid mixture mass balance
- non-condensable gas mass balance (for each component)
- fluid mixture energy balance
- wall energy balance
- radioactive or chemical activity or mass balances (for each component)

GADGET

No information is printed (except for the state CLOSED) until the gadget is activated.
Information is printed just after the “father” module information.

TABLE 24: SOURCE, SINK, PIQREV, PIQBREK, PIQSEB, PIQSOUP, PIQVANNE, BREAK, SGTR

- mass flowrate (kg/s)
- enthalpy flowrate (J/s)
- if the sink/source is located in a volume, the elevation of the gadget is printed out too.

THE FUEL STAND-ALONE COMPUTATION

TABLE 25: CONTROL VALVE, CHECK VALVE, FLOW LIMITER, ECHECK, ECVALVE

PU	:	stem position
I	:	axial mesh number
CV	:	current valve capacity (unit is compatible with the flowrate given in kg/s and a pressure given in Pa)
KM	:	current singular pressure drop coefficient at the valve
SR	:	reduced valve section (m²)
EQUIV KM	:	equivalent KM
EQUIV SR	:	equivalent SR (m²)
DPVEC	:	current pressure difference at the valve (Pa)

TABLE 26: ACCUMULATOR

The following variables are given:

- pressure (MPa)
- liquid enthalpy (kJ/kg)
- gas volume (m3)
- initial and actual mass of water (kg)
- liquid and gas flows (kg/s)

TABLE 27: TURBINE

FTURB	:	singular pressure drop coefficient
PTURB	:	power extracted from the fluid (W)
DELTA H	:	ideal loss of enthalpy through the turbine (kJ/kg)
VTURB	:	rotation velocity (m/s)
RB1	:	pressure ratio ($P_{downstream}/P_{upstream}$)
DELTA P	:	head loss at the turbine (Pa)

TABLE 28: CANDLE

The mesh number and the exchanged power in Watts are given.

TABLE 29: PUMPCHAR, TCOMCHAR

P. SUCTION	:	pressure at the suction side, Pa
MASS FLOW	:	mass flowrate, kg/s
VOLUME FLOW	:	volume flowrate m^3/s
DENSITY	:	fluid density, kg/s
P. DISCHARGE	:	pressure at the discharge side, Pa
ELEVATION	:	elevation, m.

The meaning of the rest of data follows clearly from the listing.

TABLE 30: SGCARACT

In the listing, most of the data concerning one “small” SG are given under PRINTOUTS FOR STEAM GENERATOR POINT_SG_NAME (PERIOD-dependent printout). The steam flowrate going to “the steam line” is given in the table together with other characteristics of the point SG. The “safety valve” flowrate is given, however, at each time step. For example:

```
SECONDARY SG VALV OPENED IN POINT_SG_NAME
      SECONDARY PRESSURE =    0.291D+07 VALV FLOW =    0.649D-02
```

FEEDWATER ENTHALPY in J/kg;
FEEDWATER TEMP in °C;
TOTAL EXCH. SURF. in m^2;
TOTAL EXCH. POWER in W;
MEAN HEAT FLUX: W/m^2;
THERMAL RESIST. in °C/W;
PRESSURE in Pa;
TEMP. SAT in °C;
SEC. MASS OF WATER in kg;
SEC. MASS OF STEAM in kg;
STEAM FLOW in kg/s;
FEEDWATER FLOW in kg/s;
INIT. WATER FLOW in kg/s

APPENDIX 11 Description of the stand-alone fuel hydraulic file FORT62

FIXED Part

- CATAFUEL CIRCUIT: Nothing

- HYDRAULIC CHANNEL

Subroutine: HISTIHY0.f
Block name in CATHARE Hydraulic File (FORT62): HISTIHY0

Name	Size	Pointer	Type	Definition
NPAT	1	LNVAR	LPOINT	Number of junctions (forced to 2)
NINCON	1	LNVAR	LPOINT	Number of non-condensable gases
NBACON	1	LNVAR	LPOINT	Number of radio-chemical elements (forced to 0)
NBWALL	1	LNVAR	LPOINT	Number of walls (forced to 1)
NREWAL	1	LNVAR	LPOINT	Number of reflooding walls (forced to 1)
NBGADG	1	LNVAR	LPOINT	Number of gadgets (forced to 0)
ICAOPT	1	LICAOPT	LINTEG	Stand-alone fuel calculation option
IOFILE	1	LIOFILE	LINTEG	Stand-alone fuel file address

Subroutine: HISTIHYF.f
Block name in CATHARE Hydraulic File (FORT62): HISTIHYF

Name	Size	Pointer	Type	Definition
HNAMSTOR	1	LHNASTO	LCHARA	Hydraulic name
CNAMSTOR	1	LCNASTO	LCHARA	Associated fuel name
IREWT	1	LIREWT	LINTEG	Reflooding state index

Subroutine: ESTIHY0.f / MSTIHY0.f
Block name in CATHARE Hydraulic File (FORT62): ESTIHY0

Name	Size	Pointer	Type	Definition
NMAIL	1	LNVAR	LPOINT	Number of meshes

Subroutine: ESTIHYF.f / MSTIHYF.f
Block name in CATHARE Hydraulic File (FORT62): ESTIHYF

Name	Size	Pointer	Type	Definition
ITYGRS	NMHYD	LITYGRS	LINTEG	Scalar grid index
DHYDS	NMHYD	LDHYDS (1D) LDH (3D)	LDOUBL	Hydraulic diameter at scalar node
CHFGE	NMHYD	LCHFGE	LDOUBL	Geometrical coefficient for the critical heat flux
VOL	NMHYD	LVOL	LDOUBL	Mesh volume
ZVH	NVHYD	LZV (1D) LZZZV (3D)	LDOUBL	Curvilinear abscissa of vector nodes

NMHYD = number of hydraulic meshes
NVHYD = number of hydraulic vector nodes

VARIABLE Part

- CATAFUEL

Subroutine: ALSTAHYD.f
Block name in CATHARE Hydraulic File (FORT62): ALSTAHYD

Name	Size	Pointer	Type	Definition
TYPSTO	1	LTYPSTO	LINTEG	Storage type
NPDT	1	LNPDT	LINTEG	Number of time steps between two save times
TPRGEN	1	LTGEN	LDOUBL	Save time
TPRGENO	1	LTGENCAO	LDOUBL	Old save time
DTGEN	1	LDTGEN	LDOUBL	Save time step

Remark: Storage type can be either 0, (at each time step) or 1, (user-defined storage frequency in time steps or time).

- HYDRAULIC CHANNEL

Subroutine: ESTAHYD.f / MSTAHYD.f
Block name in CATHARE Hydraulic File (FORT62): ESTAHYD

Name	Size	Pointer	Type	Definition
ISOH2	1	LISOH2	LINTEG	Hydrogen release flag

IWREWT	1	IP(LIWREWT) (1D)
		IP(LIWREWT+IH1D-1) (3D)
			LINTEG	Reflooding activation flag

IDRWE	NMHYD	LIDRWE	LINTEG	Facing reflooded wall flag for hydraulic meshes
IBTFL	1	LIBTFL (1D) IP(LIBTFL+IH1D-1) (3D)	LINTEG	1D or 2D meshing index in case of bottom – top reflooding
KDOB	1	LKDOB (1D) IP(LKDOB+IH1D-1) (3D)	LINTEG	Scalar point just downstream of the quench front in case of bottom – top reflooding
ICHOIB	1	LCHOIB (1D) IP(LCHOIB+IH1D-1) (3D)	LINTEG	Flag to indicate that the cell will be reflooded in case of bottom – top reflooding
ITDFL	1	LITDFL (1D) IP(LITDFL+IH1D-1) (3D)	LINTEG	1D or 2D meshing index in case of top – bottom reflooding
KDOT	1	LKDOT (1D) IP(LKDOT+IH1D-1) (3D)	LINTEG	Scalar point just downstream of the quench front in case of top – bottom reflooding
ICHOIT	1	LCHOIT (1D) IP(LCHOIT+IH1D-1) (3D)	LINTEG	Flag to indicate that the cell will be reflooded in case of top – bottom reflooding
P	NMHYD	LP	LDOUBL	Pressure
HL	NMHYD	LHL	LDOUBL	Liquid enthalpy
HG	NMHYD	LHG	LDOUBL	Gas enthalpy
ALFA	NMHYD	LAG	LDOUBL	Void fraction
Xj (j =1,nincon)	NMHYD	LX(j)	LDOUBL	J^{th} non-condensable mass fraction
VLN	NMHYD	LVLN	LDOUBL	Normalised liquid velocity
VGN	NMHYD	LVGN	LDOUBL	Normalised gas velocity
VLMVG	NMHYD	LVLMVG	LDOUBL	Normalised difference between liquid and gas velocity
SVGMGS	NMHYD	LSVGMGS	LDOUBL	Sign of (V_L*V_G)
ZFTB	1	LZFTB (1D) IP(LZFTB+IH1D-1) (3D)	LDOUBL	Quench front level
EPSVB	1	LEPSVB (1D) IP(LEPSVB +IH1D-1) (3D)	LDOUBL	Weighting factor from quench front level to downstream vector point in case of bottom – top reflooding
DTREFB	1	LDTREFB (1D) IP(LDTREFB+IH1D-1) (3D)	LDOUBL	Reference reflooding DT
ZFTT	1	LZFTT (1D) IP(LZFTT+IH1D-1)	LDOUBL	Quench front level in case of top - bottom reflooding

EPSVT	1	LEPSVT (1D) IP(LEPSVT +IH1D-1) (3D)	LDOUBL	Weighting factor from quench front level to downstream vector point in case of top – bottom reflooding
DTREFT	1	LDTREFT (1D) IP(LDTREFT+IH1D-1) (3D)	LDOUBL	Reference reflooding DT in case of top – bottom reflooding

- fuel wall

Subroutine: COSTAHYD.f
Block name in CATHARE Hydraulic File (FORT62): COSTAHYD

Name	Size	Pointer	Type	Definition
POWNEUT	1	LPOWENR	LDOUBL	Non-residual power
POWRES	1	LPOWRES	LDOUBL	Residual power

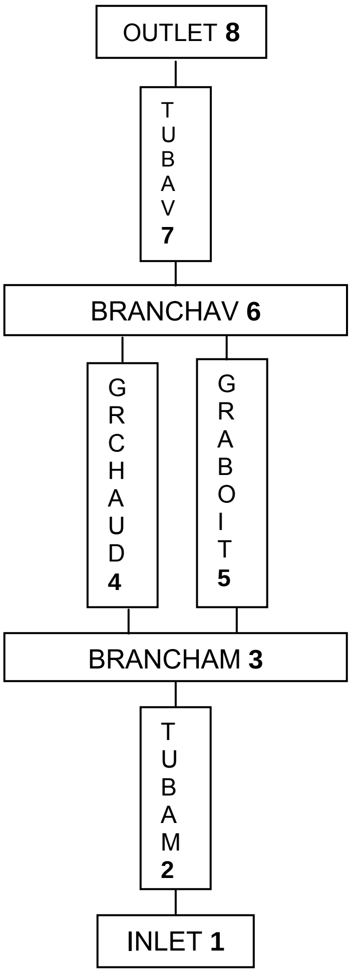
APPENDIX 12 Building the junction elimination tree for a parallel calculation

Description of sequential elimination

In a sequential calculation, the elimination tree is automatically built by the code, based on a Minty-like algorithm. The principle of such an algorithm is as follows:

(1) For all the elements making up a circuit, the junction number of each pair of connected elements (EL1 – EL2) is counted.

Example: Consider the following circuit:



Number	Element	Type
1	INLET	BCONDIT
2	TUBAM	AXIAL
3	BRANCHAM	VOLUME
4	GRCHAUD	AXIAL
5	GRABOIT	AXIAL
6	BRANCHAV	VOLUME
7	TUBAV	AXIAL
8	OUTLET	BCONDIT

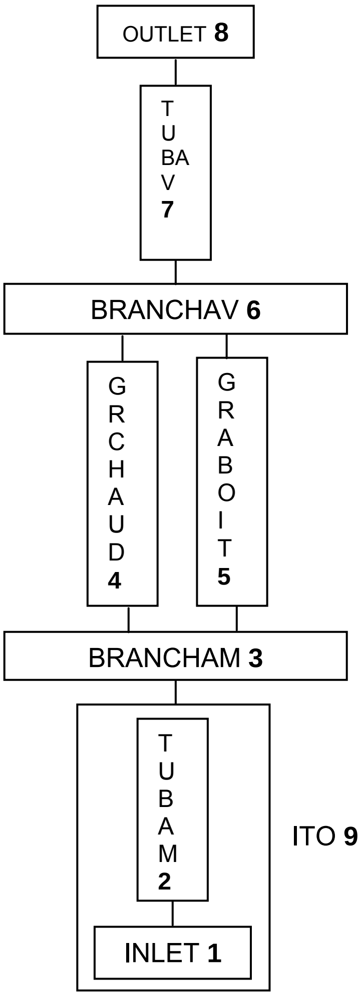
EL1	EL2	Total number of junctions for the pair (EL1-EL2)
1	2	**2**
2	3	**4**
3	4	**4**
3	5	**4**
4	6	**4**
5	6	**4**
6	7	**4**
7	8	**2**

(2) The pair with the smaller number of junctions is selected. Let the pair of elements be called IFROM1 and IFROM2.

Example: Considering the table above, the pair is EL1-EL2 = 1-2 (or 7-8).

(3) The elimination will start with these elements. They can then be considered as one virtual element (ITO). This virtual element can replace IFROM1 and IFROM2 in the connection table.

(4) Operations have to be done again from (1) with this new “circuit” until there are no more junctions.

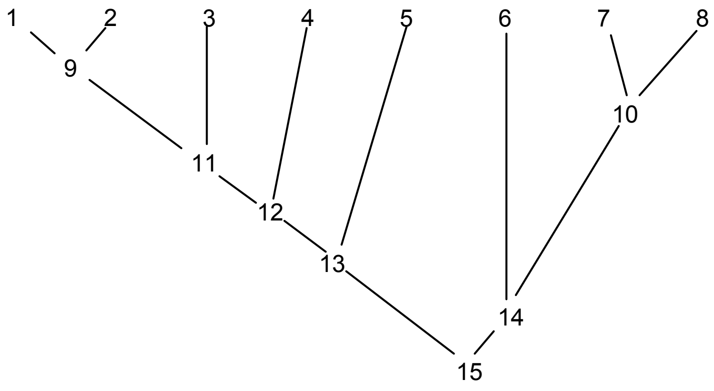


EL1	EL2	Total number of junctions for the pair (EL1-EL2)
9	3	**3**
3	4	**4**
3	5	**4**
4	6	**4**
5	6	**4**
6	7	**4**
7	8	**2**

Example: The second pair is IFROM1-IFROM2 = 7 – 8. Therefore, ITO = 10

EL1	EL2	Total number of junctions
9	3	**3**
3	4	**4**
3	5	**4**
4	6	**4**
5	6	**4**
6	10	**3**

Finally a binary tree of junction eliminations is obtained between elements (1 to 8) or virtual elements (9 to 15).



For a sequential calculation, the elimination tree can be found in the standard output of CATHARE.
Example for the CP1 plant schematisation delivered with the code (6” break accident):

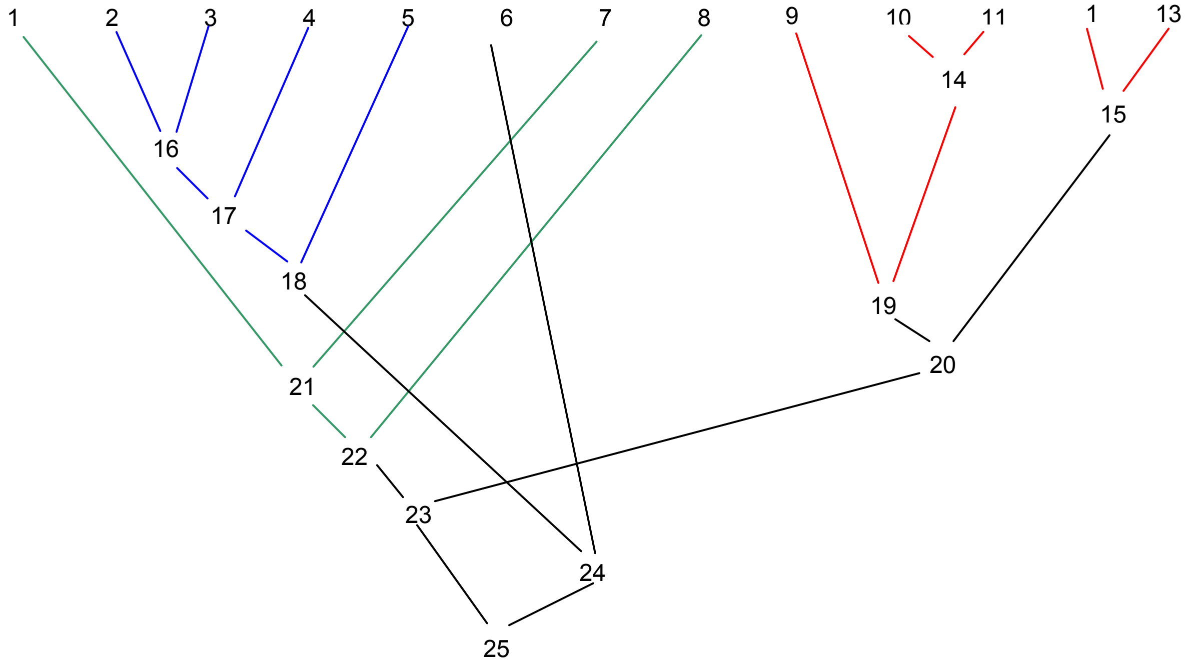
```
            *** ELIMINATION TREE ***
     IFROM1           IFROM2           ITO
10 (SURGEL)      11 (PRESSU  )   14 (VIRTUAL )
12 (PIPBRK  )    13 (BREAK1  )   15 (VIRTUAL )
2 (DOWNCO  )     3 (LWPLEN  )    16 (VIRTUAL )
4 (BYPASS  )     16 (VIRTUAL )   17 (VIRTUAL )
5 (MIDCORE )     17 (VIRTUAL )   18 (VIRTUAL )
9 (STMGENB)      14 (VIRTUAL)    19 (VIRTUAL )
15 (VIRTUAL )    19 (VIRTUAL )   20 (VIRTUAL )
1 (VOLDOWN )     7 (UPHEAD  )    21 (VIRTUAL )
8 (STMGENI )     21 (VIRTUAL )   22 (VIRTUAL )
20 (VIRTUAL )    22 (VIRTUAL )   23 (VIRTUAL )
6 (UPPLEN  )     18 (VIRTUAL )   24 (VIRTUAL )
23 (VIRTUAL )    24 (VIRTUAL )   25 (VIRTUAL )
```

Application to a parallel calculation

For a parallel solver calculation, the objective is to group together appropriate triplets (IFROM1, IFROM2, ITO) on each processor. Each processor will then treat one or several groups of triplets.
The elimination is made in three sequential actions:
Considering N groups of triplets

- N-4 groups of triplets are treated (parallel mode) on the allocated processors,
- The last N-3, N-2 and N-1 groups are then treated (parallel mode – one group for one processor),
- Finally the last group is treated sequentially.

The parallel calculation tree, e.g. elimination group formation, has to be built by the user.
The simplest way is to use the sequential elimination tree to form independent and load-balanced (from the junction number to eliminate point of view) groups knowing the number of processors that will be used.
Example:
Consider the CP1 plant mentioned before. The sequential elimination tree can be drawn from the previous table:



Several independent elimination groups appear in this drawing.

Allowance must be made for the number of junctions to be eliminated for each triplet (see table below).

triplet	Junction number to be eliminated
10 11 14	4
12 13 15	3
2 3 16	5
4 16 17	5
5 17 18	5
9 14 19	6
15 19 20	5
1 7 21	6
8 21 22	6
20 22 23	7
6 18 24	7
23 24 25	10

Knowing that the four last groups will be eliminated in a second step, and considering a 3-processor parallel calculation, a possible group distribution could be as follows:

- First step: distribution of groups on the three processors. In the example, there is one group for each processor but, in a general case, one processor may treat several groups of triplets.

processor 1		
group	triplet	number of junctions
1	10 11 14	4
1	9 14 19	6
1	12 13 15	3
1	15 19 20	5
Total		18

processor 2		
group	triplet	number of junctions
2	2 3 16	5
2	4 16 17	5
2	5 17 18	5
Total		15

processor 3		
group	triplet	number of junctions
3	1 7 21	6
3	8 21 22	6
Total		12

It can be seen that the proposed groups provide poor CPU load balancing. It may be interesting to define more groups in order to obtain a better distribution of the calculation cost (group 1: 10 11 14 and 9 14 19, group 2: 2 3 16 and 4 16 17, group 3, etc.).

- Second step (The last N-3, N-2 and N-1 groups are then processed (parallel mode – one group for one processor). In this example, each group is made up of only one triplet but this is not a general case.

processor 1		
group	triplet	number of junctions
4	20 22 23	7

processor 2		
group	triplet	number of junctions
5	6 18 24	7

processor 3		
group	triplet	number of junctions
6	0	0

Here, the 6th group is empty but necessary to “fill” the third processor.

- Third step (finally, the last group is processed sequentially). In the example given here, this group is made up of only one triplet but this is not a general case.

processor 1		
group	triplet	number of junctions
7	23 24 25	10

Implementation in CATHARE

This information has to be given to the CATHARE code using the SGUSERREAC.f user subroutine (if only one solver is used – one circuit or implicitly coupled circuits) or SGUSERC1.f and SGUSERC2.f (case of explicitly coupled circuits for which one tree for each circuit has to be defined).

Users must specify the elimination groups of triplets in the predefined **TAB** table.

```
C
C_CP1 ARBRE
C
C Goup 1
     DATA (TAB(L,1), L=1,3)  / 10, 11,14/
     DATA (TAB(L,2), L=1,3)  / 9, 14, 19  /
     DATA (TAB(L,3), L=1,3)  / 12, 13, 15 /
     DATA (TAB(L,4), L=1,3)  / 15, 19, 20  /
C Group 2
     DATA (TAB(L,5), L=1,3)  / 2, 3, 16 /
     DATA (TAB(L,6), L=1,3)  / 4, 16, 17 /
     DATA (TAB(L,7), L=1,3)  / 5, 17, 18 /
C Group3
     DATA (TAB(L,8), L=1,3)  / 1, 7, 21 /
     DATA (TAB(L,9), L=1,3)  / 8, 21, 22  /
C Group 4
     DATA (TAB(L,10), L=1,3) / 20, 22, 23  /
C Group 5
     DATA (TAB(L,11), L=1,3) / 6, 18, 24  /
C Group 6
C Group 7
     DATA (TAB(L,12), L=1,3) / 23, 24, 25 /
C
```

The remaining variables have to be initialised. In the example given here, there are 12 triplets:

```
     DO J = 1 ,3
      DO I = 13, NBELIX
         TAB( J,I) = 0
      ENDDO
     ENDDO
```

The group distribution is given in the **NEL** table (dimension is the group number **NGR** variable – here 7) by defining the number of triplets for each group.

```
     DATA NGR   / 7 /
     DATA NEL   / 4, 3, 2, 1, 1, 0, 1 /
```

The elimination distribution on the processors is described in the **NUMG** table (dimension of this table is given by **NELIM** variable = 6):

- Its first value is necessarily 1,
- The second value indicates the number of the last group processed in the first step in the TAB table,
- The third value indicates the first group number (in the TAB table) processed in the second step,
- The fourth value indicates the second group number (in the TAB table) processed in the second step,
- The fifth value indicates the third group number (in the TAB table) processed in the second step,
- The sixth value indicates the last group number that is processed in the third (sequential) step.

In this example, this gives:

```
     DATA NELIM   / 6 /
     DATA NUMG   / 1, 3, 4, 5, 6, 7 /
```

APPENDIX 13 Cross-reference table

Main object cross reference table

Main object	Type	RESULT	PERIOD	LIST	VALUE	WRITE	VALBA	WRIBA	Remarks
REACTOR	Assembly	×	×	×	×				
CIRCUIT	Assembly	×	×	×	×		×	WCIRCBA	
ZONE	Assembly	×	×	×	×		×	WCIRCBA	Balance calculations
JUNCTION	Connection between modules	×			×		×		
AXIAL	Module	×	×	×	×	×	×	×	
VOLUME	Module	×	×	×	×	×	×	×	
THREED	Module	×	×	×	×	×	×	×	
BCONDIT	Module	×	×	×	×	×	×	×	
RUPTURE	Module	×	×	×	×	×	×	×	
CATAFUEL	Assembly	×	×						Fuel stand-alone computation
HYDIMP	1D imposed hydraulic		×						Fuel stand-alone computation

Sub-module - Cross-reference table

Sub-module	Supported by						PERIOD	LIST	VALUE	WRITE	VALBA	WRIBA	Remarks/Physical model
	CIRCUIT	JUNCTION	AXIAL	THREED	VOLUME	WALL							Wall = WALL, WALL3D or FUELPLAQ
ZONE3D				×				×	×		×		Created by BILAN3D for balances
DEADZONE	×												Created by PERMINIT for initialisation
HYDCHAN				×									Operator used for stand-alone fuel computation
WALL			×		×			×	×	×			Thermal conduction
WALL3D				×				×	×	×			Thermal conduction
FUELPLAQ			×					×	×	×			Thermal conduction
FUEL			×										Thermo-mechanics
FUEL3D				×									Thermo-mechanics
FUELCHAR			×	×				×	×	×			Thermal conduction Thermo-mechanics
REFLCHAR			×					×	×				Reflooding - rewetting
REFLCH3D				×				×	×				Reflooding - rewetting
RADCHEMI	×							×					Radio-chemical transport
NONCOND	×							×					Thermal hydraulic and non-condensable gas transport
CORE	×						×	×	×				Point kinetics
SGCARACT						×*		×	×	×			Primary-secondary heat exchange (point model)
EXCHANGER			×	×				×	×	×			Primary-secondary heat exchange

* Not available for FUELPLAQ

Gadget cross-reference table

Gadgets	Supported by						CONNECT	DIS CONNECT	LIST	VALUE	WRITE	VALBA	WRIBA	Remark
	CIRCUIT	JUNCTION	AXIAL	THREED	VOLUME	WALL								Wall = WALL, WALL3D or FUELPLAQ
TEE			×				CONNECT		×	×	axial			Tee branch WRITE is accessible through the axial element
CCFL		V-A 3-A*	×	×			ENABLE	DISABLE	×	axial	axial			CCFL VALUE and WRITE are accessible through the axial element
ACCU			×		×		auto OPEN	CLOSE	×	×	×			Accumulator
BREAK			×				OPEN	CLOSE	×	×		×		Break
CANDLE			×	×	×		OPEN	CLOSE	×	×	×			Energy sink/source
PIQARE			×				OPEN STAROVRF	CLOSE STOPOVRF		×	×	×	×	ARE overflow (model automatically activated when connected)
PIQBREK			×		×		OPENBREK	CLOSE		×	×	×	×	Break
PIQREV			×		×		OPEN	CLOSE		×	×	×	×	Sink/Source
PIQSEB, PIQSOUP, PIQVANNE			×		×		OPEN	CLOSE		×	×	×	×	Valve
PUMPCHAR			×				auto STARPUMP	STOPUMP	×	×	×			Pump (model automatically activated when connected)
SENSOR			×		×	×	auto		×	×				Sensor
SINK			×		×		OPEN	CLOSE	×	×	×	×		Sink
SINKRRI					×		OPEN	CLOSE	×	×	×			Sink with exchangers
SOURCE			×		×		OPEN	CLOSE	×	×	×	×	×	Source
SOURIS					×		OPEN	CLOSE	×	×	×			RIS type source

* V-A for a Volume – Axial junction
3-A for a THREED – Axial junction

Gadgets	Supported by						CONNECT	DIS CONNECT	LIST	VALUE	WRITE	VALBA	WRIBA	Remark
	CIRCUIT	JUNCTION	AXIAL	THREED	VOLUME	WALL								Wall = WALL, WALL3D or FUELPLAQ
SGTR			×				OPENBREK	CLOSE	×	×		×		Steam generator tube rupture
TURBINE			×				auto		×	×	×			Turbine
CHECK VALVE CONTROL VALVE		V-A T-A*	×				OPEN	CLOSE	×	×	×			Valve
FLOW LIMITER		V-A T-A*	×				OPEN	CLOSE	×	×				Flow limiter
ECHECK/ ECVALVE		V-A T-A*	×				OPEN	CLOSE	×	×				Valve
TCOMCHAR			×				auto OPEN	CLOSE		×	×			Gas turbo machine (model automatically activated when connected)
SHAFT			×				auto OPEN	CLOSE		×	×			Gas turbo machine (model automatically activated when connected)

* V-A for a Volume – Axial junction
T-A for a Tee – Axial junction
3-A for a THREED – Axial junction

Link between other objects	Supported by						CONNECT	DIS CONNECT	LIST	VALUE	WRITE	VALBA	WRIBA	Remark
	CIRCUIT	JUNCTION	AXIAL	THREED	VOLUME	WALL								Wall = WALL, WALL3D or FUELPLAQ
FLOMIXER					×		STARMIX	STOPMIX	volume	volume				Vessel loop mixing model VALUE and WRITE are accessible through the volume
EXHYLINK			×[1]	×[1]	×[1]		SWITCH ON	SWITCH OFF						[1] Many gadgets can be connected through this object. Please refer directly to the related pages

APPENDIX 14 Thematic INDEX

Input deck basic facilities	
Mathematic/logical operations	*, **, +, -, /, <, >, <EQ, >EQ, ABSOLUTE, EXP, NEQ, AND, MAX, MIN, OR, INTERP TRIGONOMETRIC functions: SINUS, COSINUS, TANGENT, ARCSINUS, ARCCOSINUS and ARCTANGT (which are respectively the key words to access the SINE, COSINE, TANGENT, ARCSINE, ARCCOSINE, ARCTANGENT trigonometric functios)
Loops, tests, etc.	REPEAT, END, IF, ENDIF, ELSE, QUIT
Variable definition	CHAR8, INTEGER, DOUBLE, REALLIST, LAW, **SEGMENT**, **VECTOR**, **XAXIS**, **SCALAR**
Input deck organisation	BEGIN DATA, END DATA, END EXEC
Computation management	
Computation options	OPTION, VERBOSE, RESETIME, RESTORE, SAVE, INISIM, MANAGE, FASTSIZE
Simulator use	INISIM, MANAGE, OPTION
Parallel calculation	OPTION, PERF, PESEE, PRINSTRU
Calculation management	
Achieving a time step	**NEWDT**, **NEWTIME,** TRANSIENT, TRANFUEL
Launching specific calculation	GODIFF, GOTURB, GOBORA, GOFUEL, GONEUT, GOPERM
Connecting / disconnecting gadget	OPEN/CLOSE, ENABLE/DISABLE, OPENBREK, STARTVC/STOPVC, SWITCH ON/OFF, STARALTR/STOPALTR, STARSHAF/STOPSHAF, STARSURG/STOPSURG
Initialisation	PERMINIT, REINIT, RESETIME, INIBIL, INIBORA, **ZONBAMOY**
Steady state initialisation	PERMINIT, GOPERM, REALC, REALVO, REALAX, IMPOSFLOW, NOFLOW, LEVEL, ECHPOWER, TPER3D
Deadzone reinitialisation	REINIT, REALVO, REALAX
Results	
Output	RESULT, PERIOD, LIST, IMPRIME, MESSAGE, PRIN3D, SAVE, TITLE
Balances	BILAN3D, **ZONE**, INIBIL
Obtaining / imposing values	VALUE, WRITE, SENSOR, UTILx, VALUFEAU, VALBA, WRIBA
User's file management	OPENFILE, READHEAD, READVAR, REWIND, WRITHEAD, WRITVAR

Reactor components	
Reactor assembly	**CIRCUIT**, **REACTOR**, **CORE**, **RADCHEMI**, **NONCOND**
Modules	**RUPTURE**, **THREED**, **VOLUME**, **AXIAL**, **BCONDIT**
Thermo components	**WALL, WALL3D, FUEL, FUEL3D, FUELCHAR, FUELPLAQ, EXCHANGER, REFLCHAR, REFLCH3D**
Gadgets	**TEE**, **ACCU**, **CCFL**, **PUMPCHAR**, **SGCARACT**, **SGTR**, **TURBINE, CANDLE**, **SENSOR,** **ECHECK**, **ECVALVE, FLOW-LIMITER**, **CHECK VALVE**, **CONTROL VALVE**, **PIQSEB**, **PIQSOUP**, **PIQVANNE**, **SINK**, **SOURCE**, **PIQREV, BREAK**, **PIQBREK**, **PIQARE, SHAFT, TCOMCHAR, SOURIS, SINKRRI**
Chemical components	
Other component than water	**NONCOND**, **RADCHEMI**
Boron and activity	**RADCHEMI**, INIBORA, GOBORA, **ZONBAMOY**, WRIBA, VALBA, WCIRCBA, ACTEMIS, SCRAM, **ACCU**
Hydraulic components	
Pipe	**AXIAL**, GEOM, HYDR, MESH, SINGULAR, GOBALLON
Volume	**VOLUME**, GEOM, FLOMIXER, LEVEL, SINGULAR, VFILM
Specific models for volumes	**FLOMIXER**, STARMIX, STOPMIX (Vessel loop mixing) **VFILM**, STARTVC, STOPVC (film condensation)
3D	**THREED**, CONNECT, GEOM, HYDR, MESH, PHYSCALE, SINGULAR, TPER3D
Specific models for 3D	GODIFF (diffusion) TURBULEN, GOTURB (turbulent diffusion) DTMASS3D (time step management from mass error)
Boundary condition	**BCONDIT,** BCMOD, MODEL
Double-ended break	**RUPTURE**, ASSIGN
Thermal components	
Wall	**WALL**, **WALL3D**, FLUMOD, ADIABWAL, RADIAT, POWER, PNRSHAPE/PNRSHAPEX, ALWMIN, AXICOND, **EXWALINK**
Wall exchangers	**EXCHANGER**, ECHPOWER, EXCPOWER
Fuel	**FUEL**, **FUELCHAR**, **FUEL3D**, AND, GOFUEL, INTEGRATE, RSETFUEL, STOPFUEL, FUELDTMX **FUELPLAQ**, POWER, PNRSHAPE/PNRSHAPEX, PRESHAPE/PRESHAPX, XNEULIST/XNEULISX, MODXNEUT, OXRATE, SPALL, SPALLOX, PICTG STOPNEUT
Rewetting, reflooding	**FUELCHAR**, **REFLCHAR**, **REFLCH3D**, REFLOOD

Specific models	
Valve	**ECHECK**, **ECVALVE, FLOW-LIMITER**, **PIQSEB**, **PIQSOUP**, **PIQVANNE**, **VALVE**, **CHECK VALVE**, **CONTROL VALVE**
SEBIM	**PIQSEB**
Primary to secondary heat exchange	**EXCHANGER**, **SGCARACT**
Sink, source	**SINK**, **PIQREV**, **SOURCE**
Overhead limitation	**PIQSOUP**, MODEL(**BCONDIT**), **SINK**(SAFETYVA), **BREAK**(SAFETYVA),
Head loss coefficient	SINGULAR, **VALVE**
Break modelling	**PIQBREK**, **BREAK**, **BCONDIT**(BC4A, BC5), **SGTR**, **RUPTURE**
Point kinetics	**CORE**, SCRAM, GONEUT, **FUEL**(NEUTRO), **FUEL3D**(NEUTRO), **FUELCHAR**(NEUTRO), XNEULIST, XNEULISX, MODXNEUT STOPNEUT(FUELPLAQ)
Vessel loop mixing	**FLOMIXER**, STARMIX, STOPMIX
Tee branch	**TEE**, CONNECT, GEOM, LINK, SINGULAR, ENABLE/DISABLE(JETPUMP)
Counter-current flow limitation	**CCFL**, DISABLE, ENABLE
Pump	**PUMPCHAR**, BLKROTOR, PCAVIT, PUMPMOD, STOPPUMP, STARPUMP
Point steam generator	**SGCARACT**, GVPPOWER, **WALL**(POINTSG), SGFEED
Source	**SOURCE**, SOURCEMOD
Turbine	**TURBINE**, MODPCST, MODPNST
Steam generator feedwater overflow	**PIQARE**, STAROVF, STOPOVRF
Gas turbomachine	**SHAFT, TCOMCHAR,** SHAFTMOD, STARALTR, STARSHAF, STARSURG, STOPALTR, STOPSHAF, STOPSURG**,** TCOMMOD
Sink with exchanger	**SINKRRI**
RIS overflow	**SOURIS**
Explicit coupled modules / gadgets	**EXHYLINK**
Accumulator	ENABLE/DISABLE (VALVE)
Fuel stand-alone computation	
Preparation of calculation	STOREHYD, **HYDCHAN**, RESHCAT
Specific operators and directives	**CATAFUEL**, **HYDIMP**, HYDMOD, READHCAT, READHEXT, STDYFUEL, TRANFUEL, COMPHYD

END OF DOCUMENT