Direction de l'Energie Nucléaire
Direction déléguée aux Activités Nucléaires de Saclay
Département de Modélisation des Systèmes et Structures
Service de Thermohydraulique et de Mécanique des Fluides

Automatisation des cas tests de validation de Trio_U

S. Vandroux
V. Barthel

N° d'identification

Commissariat à l'énergie atomique et aux énergies alternatives
Centre de Saclay – DEN/DANS/DM2S/STMF – Bâtiment 454 – Point courrier 47
91191 Gif-sur-Yvette cedex - France
Tél. : +33 (0)1 69 08 91 92 – Fax : +33 – (0)1 69 08 96 96 – bernard.faydide@cea.fr
Etablissement public à caractère industriel et commercial
R.C.S. PARIS B 775 685 019

NIVEAU DE CONFIDENTIALITÉ				
DO	DR	CCEA	CD	SD
X				

PARTENAIRES/CLIENTS	ACCORD	TYPE D'ACTION
CEA		

RÉFÉRENCES INTERNES CEA			
DIRECTION D'OBJECTIFS	DOMAINE	PROJET	EOTP
DISN	**SIMU**	**Sithy**	**A-SITHY-03-03-04**
JALON	INTITULÉ DU JALON	DÉLAI CONTRACTUEL DE CONFIDENTIALITÉ	CAHIERS DE LABORATOIRE
	Transfert de connaissances		

SUIVI DES VERSIONS			
INDICE	DATE	NATURE DE L'ÉVOLUTION	PAGES, CHAPITRES
A	26/07/2013	Document initial	27, 4
		Document rédigé sur la base de la note technique DEN/CAD/DER/SSTH/LMDL/NT/2009-108/A	

	NOM	FONCTION	VISAS	DATES
RÉDACTEURS	S. VANDROUX V. BARTHEL	Ingénieur Ingénieur		
VÉRIFICATEUR	O. CIONI	Ingénieur		
CONTRÔLE QSE	D. DUMONT	Correspondant Qualité		
APPROBATEUR	O. ANTONI	Chef de Laboratoire		
EMETTEUR	B. FAYDIDE	Chef de Service		

MOTS CLEFS

Validation, base de données, Trio_U, cas test automatisé

RÉSUMÉ / CONCLUSIONS de même niveau de confidentialité que le document

Cette note présente l'organisation de la base de données de validation de Trio_U ainsi que la procédure pour générer de nouveaux cas tests de validation. Elle est basée sur une précédente note datant de 2009 et en constitue une mise à jour.
Cette base de données est composée de deux types de cas test : quelques cas anciens 'faits à la main' avant la version 1.5.5, ainsi que les cas automatisés.
La manière de générer ces cas tests automatisés est présentée tout d'abord. On passe ensuite en revue les différentes procédures automatiques mises en place et qui sont lancées après la mise en gestion de configuration d'un nouveau cas test.
Enfin, le classement des cas tests dans la base de données de validation est ensuite présenté : une liste exhaustive des cas tests rangés suivant une certaine logique.

This document is an update of a former document written in 2009.

The way to generate the new automatic test cases is first explained. A review of the automatic launching procedures after the versioning of a new test case is then described. Last, the classification of the test cases in the validation data base is outlined.

DIFFUSION INITIALE

(Diffusion par email, document DO)

Diffusion interne CEA

Simone VANDROUX,	Auteur	DEN/DANS/DM2S/STMF/LMSF
Valérie BARTHEL,	Auteur	DEN/DANS/DM2S/STMF/LMSF
Olivier CIONI,	Vérificateur	DEN/DANS/DM2S/STMF/LMSF
Olivier ANTONI,	Chef du DM2S/STMF/LMSF	DEN/DANS/DM2S/STMF/LMSF
Bernard FAYDIDE,	Chef du DM2S/STMF	DEN/DANS/DM2S/STMF/DIR
Christian CAVATA,	Chef du DM2S	DEN/DANS/DM2S/DIR
Anne NICOLAS,	Adjoint du chef du DM2S	DEN/DANS/DM2S/DIR
Philippe MONTARNAL,	Chef de projet SITHY	DEN/DANS/DM2S/DIR
Tous les salariés du STMF/LMSF,		DEN/DANS/DM2S/STMF/LMSF
Vincent BERGEAUD,		DEN/DANS/DM2S/STMF/LGLS
Gauthier FAUCHET,		DEN/DANS/DM2S/STMF/LGLS
Didier JAMET,		DEN/DANS/DM2S/STMF/LMEC
Danielle GALO,		DEN/DANS/DM2S/STMF/LATF
Philippe EMONOT,		DEN/DANS/DM2S/STMF/LMES
Frédéric DUCROS,		DEN/DANS/DM2S/STMF

Diffusion externe CEA

Pierre LEDAC,		CS-SI
Marthe ROUX,		CS-SI
Cyril MALOD,		CS-SI

Diffusion résumé

Jacques SEGRÉ,	Assistant Scientifique du DM2S	DEN/DANS/DM2S/DIR
Murielle MARUEJOULS,	Secrétaire du DM2S	DEN/DANS/DM2S/DIR
Xavier AVERTY,	Chef du DM2S/SEMT	DEN/DANS/DM2S/SEMT
Patrick BLANC-TRANCHANT,	Chef du DM2S/SERMA	DEN/DANS/DM2S/SERMA

TABLE OF CONTENT

1.INTRODUCTION......10
2.TEST CASES......10
3.LIST OF TEST CASES......20
4.SUMMARY AND CONCLUSION......24
REFERENCES......25
ANNEX 1: EXAMPLE OF FIRST.AWK FILE......26
ANNEX 2: EXAMPLE OF A PRM FILE (T_PAROI.PRM)......27
ANNEX 3: EXAMPLE OF REPORT FILE (T_PAROI.PDF)......28

List of Tables

Table : List of the available test cases for the laminar thermohydraulic flows

List of Figures

1. INTRODUCTION

The objective of this document is to present the organisation of the Trio_U validation data base, and the procedure to produce new test cases.

The data base is composed of two different kinds of test cases:

- A few 'old' test cases which were run up to the version 1.5.4 of Trio_U. The calculation, the post processing and the corresponding validation report were 'hand-made'. In order to update them, all the post-processing work has to be done again Due to the high CPU time required for the calculation, no automatic procedures were generated for these test cases, but they are kept in the validation base because they are still interesting ones;
- The new test cases run from version 1.5.5, where a procedure enables to run all the calculations corresponding to this test case automatically. The post processing and the validation report are also done automatically. These test cases are very easily updated for each new version of the code. Besides, with this method, it is easy to compare the calculation results of one version of the code with the results of other versions.

The first part of this report describes the procedure used to generate these automatic test cases, how they are classified, and how a new test case is added to the data base.

The second part of the report describes the organisation of the validation data base in different physical problems. It will show how the different types of thermohydraulic cases are organised in a test case list.

2. TEST CASES

This section is devoted to the test cases which are generated automatically. These test cases are generated on Linux PCs or on UNIX machines. The way to generate a new test case is described. The different procedures following the generation of a test case are also explained.

2.1 GENERAL PRINCIPLE OF HOW TO GENERATE AN AUTOMATIC TEST CASE

The objective of the automatic procedure is:

- to run all the calculations corresponding to a given test case;
- to post process the results and generate the needed tables and figures;
- to generate a report file in PDF format which contains all the necessary information for someone willing to have a close look at a given test case;
- to make it easy to compare calculations performed with different versions of the code.

In order to generate an automatic test case, the TRIO_U user must first create a directory devoted to this test case in his work domain.

In this directory, he must create a directory **src** (for source). This directory must contain all the files necessary to create the automatic test case, and nothing more. Among the necessary files, there will be a file with a **.prm** (for parameters) extension. Once this directory filled properly as will be explained below, and once the Trio_U environment initiated, the user will launch the following command to obtain the report:

Run_fiche

2.1.1 Files necessary in the src directory

In order to achieve this, the directory **src** will have to contain several files. In the simplest case, the necessary files will be a data file and the ***file*.prm** which contains the parameters of the automatic report generation.
The other possible files are:

- Files necessary to launch a calculation (for example other data files, meshing files…);
- Files which are not generated automatically but which are in the final report (for example, pictures of test descriptions, experimental data used for comparison…);
- All the optional procedures (prepare, pre_run, post_run) mentioned in section 2.1.2.

2.1.2 Execution of the Run_fiche command line

The execution of the command is described very briefly here. Application examples will be given a bit further.

This command will first copy the files contained in the src directory into another directory named **build**. In this directory **build**,

- The calculations are prepared (generation of directories, data files…) in accordance with the content of a file named **prepare**;
- The calculations listed in the ***file*.prm** are then launched one after the other;
- For each calculation:
    - If necessary, prior to the calculation, some treatment can be needed. This is done according to a **pre_run** file. A few possible examples for the use of pre_run are: uncompress big files, run a prior calculation, for example thanks to an inflow generator to build an inlet profile for boundary condition;
    - The calculation is performed;
    - After each calculation, if necessary, some post processing is done using a **post_run** file;
- Thanks to the ***file*.prm** the different figures or tables which need to be seen are created, if some previous treatment is required before the calculation, a **pre-requis** command is used, and a final PDF report is generated with the text contained in ***file*.prm** and with the figures and tables.

2.1.3 Quick description of Run_fiche command line options

The **Run_fiche** command has several interesting options which enable to generate the final report with more efficiency when a new test case is generated.

- ***-o toto.pdf*** Enables to choose the name of the final report. By default, it will be 'rapport.pdf'.
- ***-not_run*** Enables to launch the generation of the PDF file without launching all the calculations. It is very useful to produce the report file.

- ***-post_run*** Enables to launch the generation of the PDF file without launching all the calculations but launches all the post run automatic commands. It is very useful to correct graphs or tables.
- ***-v 1*** Generates the file test_lu.prm in the build directory. This file contains interesting information, such as the number of the different figures produced in the report file.
- ***-d text*** Only the text in the final report is modified accordingly to the changes of the file.prm.
- ***-d nn*** Only the figure number nn is generated. The final report is not modified.
- ***-dest path_build*** Enables to specify a path for the location of the **build** directory.
- ***-xpdf*** Enables to display directly the PDF report file after the calculation, whatever tool should be used to open the PDF file.
- ***-remote login@computer:PATH*** Enables to launch the file on a remote computer which address must be specified.
- ***-exec binary file*** Very useful in case of remote calculation, as it enables to specify the address of the binary file which should be launched**.**
- ***-parallel_run*** Detects the number of available processors on your computer and launches the different calculations of these prm files on the different processors. The post processing is done only once all the calculations have been performed. This option must not be used if some calculations are dependant of previous ones.
- ***-visit_np np*** Launches visit on np processors when the post processing of the file is finished.
- ***--compare*** Described in section 2.1.4.
- ***-archive*** Described in section 2.5.2.

In order to be informed of new options available in the **Run_fiche** command, user has to subscribe on the following web site:
http://saxifrage:3500/wws/info/trio_u_post_traitement
On this site, the user can also post his questions, and both the question and the answer will be communicated to all the subscribers.

2.1.4 Tips and recommendation to users

During the finalization of his test case, the user will be tempted to modify some files in the **build** directory before running the **Run_fiche** command.
This must be avoided, because when the **Run_fiche** command is launched, the files present in the **src** directory will be copied in the **build** directory. The modified files in the **build** directory will be overwritten; the modifications will be lost and not taken into account.

When launching the **Run_fiche** command with different versions of the code, the **build** directory will be overwritten. If the user wants to save the results of the previous version of the code, he must rename the directory **build** (in build_version1 for example).

Once this is done, using the –compare option of the **Run_fiche** command, the user can then compare the results obtained with his two (or more) versions with the command:
Run_fiche --compare=../build_version1/ --compare=../build_version2/

This will compare the requested (in **file.prm**) meshes, tables and figures of current version, version 1 and version 2 and generate a report file for comparison.
N.B.: All the items are compared, except those for which the 'Origine xxx' indication (in **file.prm**) is specified, where xxx is different from Trio_U.

2.2 AUTOMATIC TEST CASE GENERATION STEP BY STEP

As seen in section 2.1.2, the **Run_fiche** command line has several steps. This section explains more concretely how to generate an automatic test case.

2.2.1 How to use a prepare file

This **prepare** file is a shell script. It is an optional file, but in most test cases, it is necessary to use one. Most of the time, the validation cases need more than one data file because several options are tested. The **prepare** file can thus be used:
-to create sub directories in the build directory,
-to copy the appropriate files in these directories,
-to generate the different data files necessary for the calculation starting from a single data file.

Here is an example of a **prepare** file from the test case 'T_paroi':

```
#!/bin/sh

# Creation of several directories
mkdir -p Nu_impose U_impose Symet

# Appropriate files copied in the directories
for dir in Nu_impose U_impose Symet
do
        cd $dir;   cp ../first.awk ../post_run ../T_paroi.data  . ; cd ..
Done

# Modification of the initial data file in several directories
cd Symet
sed "s/paroi_fixe/symetrie/" T_paroi.data > modifie.data ; mv modifie.data T_paroi.data
cd ../U_impose
```

```
(sed "s/Loi_Paroi_Nu_Impose { nusselt 20.\*Re^(0.)\*Pr^(0.) diam_hydr champ_uniforme 1 2.14e-3 }/Loi_standard_hydr_scalaire/" T_paroi.data | sed "s/{ u_tau Champ_uniforme 1 0. }/{ u_tau Champ_uniforme 1 0.2 }/" > modifie.data ; mv modifie.data T_paroi.data )
cd ..
```

2.2.2 How to use a pre_run or a post_run file

This **pre_run** or **postrun** files are also shell scripts. They are optional files. In most cases, only a **post_run** file is needed. They are useful if some treatment is necessary prior to the calculations or if some post processing is needed after the calculations. Depending on the test case, their content can vary from a few lines to lots of shell, awk or python command lines.

Here is an example of **post_run** file for the expansion_3D_VDF_VEF test case:

```
#!/bin/sh

# Extraction of a curve using a pressure probe
extrait_coupe 3d_vdf_rect SONDE_PRESSION

# Execute a file containing awk command lines
sh ../first.awk 3d_vdf_rect

# Fill a data file which will be used by the file.prm to draw a figure or fill a table
more delta_p.dat >> deltap_courbe.dat
```

An example of a *first.awk* file extracted from the test case *T_paroi* is given in Annex 1.

2.2.3 How to use a prm file

An example of a **prm** file is given in Annex 2 (*T_paroi.prm*), and the corresponding report file is given in Annex 3.
The commands contained in the **prm** file will briefly be commented here with the help of the example *T_paroi.prm*. However, the user will find an exhaustive description of all the tools used in the **prm** files, in the following file:
$TRIO_U_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml
In the bottom of this file, several examples are available, giving both the prm file and the resulting PDF report.

In a **prm** file, a first chapter is filled with several important parameters of the calculation. Different chapters can then be included.

In each chapter, the user can include:

- Tables (**Tableau** or **Table**) defined in the prm file. An interesting option ('Tableau_performance') enables to issue directly the computer performances with one command line.
- Figures (**Figure**) either coming from existing image files or from gnuplot files created by the prm file.
- Visualisations (**Visu**) which can be obtained directly from the lata files generated by the Trio_U calculations.

Latex format can be integrated easily in the description parts of the chapters.

2.2.3.1 Example for a Table for test case *T_paroi*

```
Tableau { Titre "Second calculation results"
    Description "Here, the wall friction velocity is imposed. Thus, the wall temperature gradient is calculated with this u* value. The theorical value of $\Delta$Tw is given thanks to the Kader law. We compare it with the Trio_U results, resumed in the table below."
    Description " "
    nb_colonnes 6
        label Tmean Outlet | Tfluid | Twall (face) | $\Delta$Tw | Twall "equiv" | $\Delta$Tw "equiv"
      Ligne { Legende "Theorical value"
          fichier ./U_impose/theoric.dat }
      Ligne { Legende "Trio_U paroi_fixe"
          fichier ./U_impose/temp.dat
              origine trio_u } }
```

Will be transformed into the following section with table:

5.2 Second calculation results

Here, the wall friction velocity is imposed. Thus, the wall temperature gradient is calculated with this u^* value. The theorical value of ΔTw is given thanks to the Kader law. We compare it with the Trio_U

results, resumed in the table below.

	Tmean Outlet	Tfluid	Twall (face)	ΔTw	Twall equiv	ΔTw equiv
Theorical value	0.1333	-	-	5.6791	-	5.6791
Trio_U paroi_fixe	0.1333	0.4629	0.5377	0.0747	6.1421	5.6791

Figure Example for a table

2.2.3.2 Example for an existing Figure for test case *T_paroi*

Figure *{ Titre "New Boundary conditions"*

Width 8cm

Image ./T_paroi_geo2.png }

Will be transformed, thanks to the image *T_paroi_geo2.png* into the following figure:

5.1 New Boundary conditions

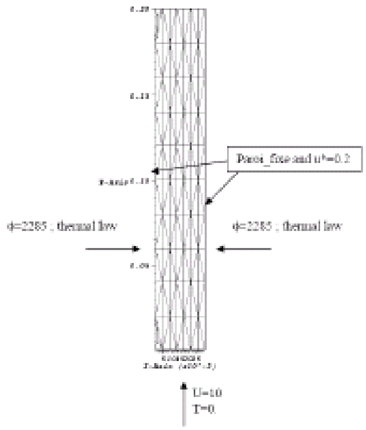


Figure Example for an existing figure

2.2.3.3 Example for a Figure created by the **prm** file for test case *expansion_2D_axi*

```
Figure { Titre "Axial pressure distribution for inlet velocity =1m/s at t=8s"
        Dimension 2
        Description "The graph below shows the pressure distribution along the axis of the pipe"
        Description "The value of P1 is taken for an axial distance equal to zero."
        LabelX "Axial distance (in m)"
        LabelY "Pressure (in kPa)"
        InclureDescCourbes 0
        Courbe { Legende "-axial pressure"
                Origine "Trio_U"
                Version "1.5.5"
                Segment ./2d_vdf_scz_1/2d_vdf_scz sonde_pression
                Style linespoints
                colonnes ($1) ($2) }  }
```

Will be transformed in the following figure:

3.4 Axial pressure distribution for inlet velocity =1m/s at t=8s

The graph below shows the pressure distribution along the axis of the pipe
The value of P1 is taken for an axial distance equal to zero.

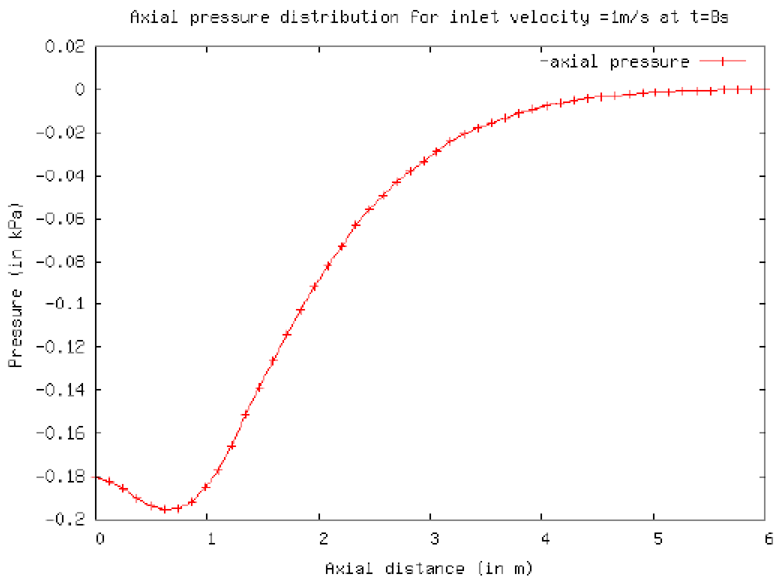


Figure Example for a figure created by the prm file

2.2.3.4 Example for a Visualisation for test case *expansion_2D_axi*

```
Visu { titre "Velocity distribution at t=8s"
    Width 9 cm
    vector ./2d_vdf_scz_1/2d_vdf_scz.lata dom VITESSE ELEM }
```

Will be transformed in the following figure:

3.1 Velocity distribution at t=8s

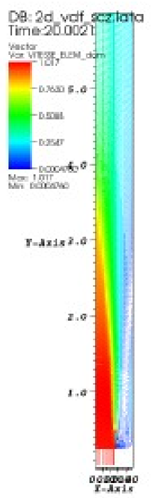


Figure Example for a visualisation

2.2.4 Tips and recommendations to users

Some simple rules have to be respected when generating a new test case.

- All the generated report files in PDF must be written in English. It is however admitted to have comments in French in the scripts.
- In the introduction chapter, the user must specify his name. The name of a 'permanent' member of the Trio_U team who has followed the work must be added if the user only works temporarily in the Trio_U team.
- If the user wants to write a section in his report that will have neither table nor picture, he must use the command **Tableau** or **Table**.
- The user should use as little data sets as possible in his **src** directory. The objective is of course to have just one data set, though it is sometimes not possible or difficult. The reason for this is that in case of a new version of the code, where the syntax of the data set is changed, it will be much quicker to modify just one data set instead of more.
- Once the new test case is generated and ready to be versioned, there should be only one **prm** file left in the **src** directory, in order to avoid confusion.
- The **src** directory should only contain the strict necessary in order to create the test case.

2.3 Organisation of the test cases

Once they exist, these test cases generated on Linux PCs or on UNIX machines must be stored in order to be taken into account in the next release version of the code.

All the existing test cases available for a version of Trio_U can be found in:
$TRIO_U_ROOT/Validation/Rapport_automatiques

This directory is divided into several directories:

- **Validant/Fini** Storage of the finished validating test cases.
- **Validant/pas_fini** Storage of the validating test cases which still need some work to be properly finished (text in the report file, translation in English, test description…).
- **Verification_codage** Storage of the test cases which are designed to verify some parts of the computation.
- **Problemes_en_cours** Storage of the test cases for which problems were found which are not solved yet, but which need to be solved in the future, or least, for which a track has to be kept.

2.4 Versioning of a test case

Once the user is ready to submit his test case to the validation team, he must use the available versioning program (either Clearcase or Git).
The **src** directory and all its containing files or directory must be versioned, either in the **Validant/pas_fini** directory or in the **Verification_codage/pas_fini** directory, depending on the type of test case.

Once warned, the maintenance team will deliver the **src** directory of the new test case with the next version of the code in the $TRIO_U_ROOT/Validation/Rapport_automatiques directory.

The test case can then be submitted to the validation team, which will then approve or suggest potential corrections.

2.5 Following prodedures

Once the test case has been versioned in the code, several procedures described below are applied.

2.5.1 Reference test case

From each test case versioned in the $TRIO_U_ROOT/Validation/Rapport_automatiques directory, **non regression test cases are generated automatically**.

They consist in the data files used in the test cases but with only a few time steps.
They are launched with each new version of the code.

If differences are observed with the previous version, the maintenance team checks if there is an evolution of the syntax in the new version. If it is the case, the necessary syntax is modified in the validation test case. The non-regression test case is generated and launched automatically again and the results are compared to those of the previous version.

If differences are detected for a test case, they will appear on a list of test cases which are to be checked by the validation team. The PDF report can be compared to the previous one by using the "—compare" option of the **Run_fiche** command. Depending on the result, the validation team will decide if the new developments can be integrated in the version or not. If these differences disable the release of the new version of the code, the origins of these differences should then be spotted and corrected before the new release.

2.5.2 Automatic launch of the test cases

All the existing versioned automatic test cases are launched automatically for each version of the code.

The final report of the test cases and an archive file containing the files necessary for the report file (a .tgz extension file obtained by using the –archive option of the **Run_fiche** command) are created. This directory is available to every person of the Trio_U team working on a Linux PC and connected to the intranet.

The new versioned test cases are spotted by an automatic procedure launched every day. Once spotted, the **test_case** directory is launched automatically as soon as possible, according to priority rules defined in this launching procedure.

2.5.3 General report

Once all the test cases calculated, the PDF files can be merged very easily in a global PDF file in order to create a global validation test case report, with an interactive table of contents.

3. LIST OF TEST CASES

Once created and versioned with the procedures seen in the previous chapter, the new test case is classified in a validation data base. Explaining the organisation of this validation data base is the aim of this chapter.

As already mentioned, for the time being, this data base is composed of two different kinds of test cases:

- A few ‘old’ test cases which were run from version 1.4 up to version 1.5.4 of Trio_U with a ‘hand-made’ validation report,
- The new test cases, run from version 1.5.5, using the procedures described in chapter 2.

This data base is available on the DENShare zone of the server CARACAS (S:) at the following address:
S:\300_PROJETS\315_TRIO_U\315b_TRIO-U_Valid\validation

The necessary authorization is required to access this address for the people who do not belong to the Trio_U team.

The data base will be available later on a server in Saclay, probably on the Alba server.

At this address, the user will have access to several “doc” or “pdf” files and several directories.

Each directory contains a validation test case. If it is a hand-made test case, the directory contains different files necessary for the test case (data file, report file…). If it is an automatic test case, the directory contains the report file in PDF format and the archive file (see 2.5.2).
The validation team is in charge of updating the directories with a ‘reference’ report file corresponding to the expected results of the validation test case. The validation team is also in charge to check the results of this test case on further versions of the code.

In order to access the different validation tests more easily, the test cases are gathered in a Word file: *Validation_list_of_cases.doc*, which is described below.

The file Validation_list_of_cases.doc contains the exhaustive list of the validation test cases calculated by Trio_U.

The test cases are classified depending on different aims or problems:

- Numerics: Gives access to the validation test cases designed to test numerical options of the code, for example the convection schemes.
- Laminar flow hydraulic: Gives access to hydraulic laminar flow problems.
- Laminar flow thermohydraulic: Gives access to thermohydraulic laminar flow problems.
- RANS modelling hydraulic: Mixing length or k epsilon turbulent modelling of hydraulic flow problems.
- RANS modelling thermohydraulic: For mixing length or k epsilon turbulent modelling of thermohydraulic flow problems.

- LES modelling hydraulic: Large Eddy Simulation in hydraulic problems.
- LES modelling thermohydraulic: Large Eddy Simulation in thermohydraulic problems.
- Front tracking: Problems using the front tracking option.
- Radiation: Problems taking radiation into account.
- Porous media: Porous media problems.
- Pure or coupled conduction: Problems with wall conduction calculation.

Each problem appears only once in one of these designations. For each designation, a table gives the list of the available test cases.

An example is given below for a part of the thermohydraulic laminar flows.

Title	Modelled problem	Tested functionnality	Code version	Discr.	Author	Tested against
Oscillating_Flow	Thermal-hydraulic flow in a cavity	Laminar 2D flow in an enclosure	1.6.2	VDF VEFP1B	E.Moreau (S.Vandroux)	Behnia & Vahl Davis numerical results
T_paroi	Thermal hydraulic flow in channel Wall temperature verification	Laminar 2D flow in channel	1.5.6	VEFP1B	V.Barthel	Analytic
Nusselt_Correlation_2D	1D Heated flow in a channel	Thermal wall function "Loi_Paroi_Nu_Impose"	1.5.6	VDF VEFP1B	V.Barthel O. Cioni	Analytical Correlations
Laminar_Flow_Vertical_Plate	Incompressible or QC laminar flow on a 2D vertical heated plate	Heat exchange, mixed or natural convection: Incompressible or QC model	1.5.5	VDF VEFP1B	S.Vandroux G Garnier (V. Barthel)	Correlations
INEEL_VDF_QC_1D_2D	Heated Plane Channel QC	Laminar flow with heat exchange, QC model	1.6.2	VDF	E.Moreau (S.Vandroux)	Analytic
Therm_stratif_water_tank	Thermal stratification in heated tank	Laminar flow, free convection	1.5.7	VDF, VEFP1B	R.Bagul, A.Borgohain, S.Vandroux	Experiment

Table : List of the available test cases for the laminar thermohydraulic flows

As can be seen, the first column of the table contains the name of the test case. This name is an hypertext link. A mouse click on this link opens the directory corresponding to the test case.
The links have three different colours.

- The blue colour corresponds to the validating automatic test.
- The green colour corresponds to the automatic test cases which are validating, but which still need some improvements (corresponds to Validant/pas_fini in section 2.3).
- The blue colour corresponds to the validating test cases which are not generated automatically.

The second column contains a quick description of the modelled problem.

The third column contains a quick description of the tested functionalities.

The fourth column specifies the code version at which this test case has been created. For the automatic test cases, the code version is also mentioned in the report file.

The fifth column specifies the author of the test case. If it has been done by someone staying only temporarily, the name of the permanent person 'in charge' should be specified.

The last column enables to see with what kind of data the test cases is validated (experiment, theory, correlations, analytic calculations, other CFD calculations…).

The validation team is in charge of updating this file.

The classification of the test cases in the file Validation_list_of_cases.doc enables to have a quick overview of the existing test cases.

4. SUMMARY AND CONCLUSION

This report presents the organization of the Trio_U validation data base. This data base is composed of two different kinds of test cases: ‘a few old hand-made’ test cases for the versions before 1.5.5 of Trio_U, and the new automatic test cases.

A user can generate an automatic validation test case. For this, he has to work on a Linux PC or a UNIX machine. To create automatic test case, he uses the tools which are released with the Trio_U version.
Once the test case is finished, it must be submitted to the Trio_U validation team. Once it is approved, the user must do the versioning of his test case. This enables the other users to access to the **src** directory of his test case. It is released with each new version of the code.
After the release of each new version of Trio_U, the test case is launched automatically. The resulting files are available to every person of the Trio_U team working on a Linux PC and connected to the intranet. The test case is also used to generate a non-regression test case, which will partly condition the release of a new version of the code.

The full validation data base is available by the Trio_U team and the people having the required permission on the DENShare zone of the server CARACAS. This zone is updated by the validation team and contains:

- An exhaustive list of the validation test cases calculated by Trio_U classified with a certain logic.
- For each test case, it contains both the report and either an archive file (automatic test cases) or several useful files (handmade test cases).

In order to improve the validation state of the code, the test cases to come, will be developed in the automatic form.

References

Description of the Run command:
$TRIO_U_ROOT/Validation/Outils/Genere_courbe/Doc_utilisation

Information on the new developments concerning the Run command.
http://saxifrage:3500/wws/info/trio_u_post_traitement

Description of all the tools used in the prm files :
$TRIO_U_ROOT/Validation/Outils/Genere_courbe/doc/manuel.xhtml

Available existing test cases:
$TRIO_U_ROOT/Validation/Rapport_automatiques

Data base of the available test cases
DENShare zone of the server CARACAS (S:):
\\CARACAS\DENShare\300_PROJETS\315_TRIO_U\315b_TRIO-U_Valid\validation_new

Annex 1: Example of first.awk file

```
#!/bin/sh
# Post processing for the 2D channel: data written temp.dat
tps_init_moyen=0
LC_NUMERIC=C

# Necessary data read in data file
mu=`grep "mu" $1.data| awk '{print $4}' | head -1`
rho=`grep "rho" $1.data| awk '{print $4}' | head -1`
lambda=`grep "lambda" $1.data| awk '{print $4}' | head -1`
Cp=`grep "Cp" $1.data| awk '{print $4}' | head -1`

echo $mu > ../mu.dat
echo $rho > ../rho.dat
echo $Cp > ../Cp.dat
echo $lambda > ../lambda.dat

Prandtl=`awk "BEGIN { print $mu*$Cp/$lambda }" `

# Calculation of theoretical  Tmoyen and DeltaTw
X=`grep -i "longueur" $1.data| awk '{print $2}' | head -1`
Y=`grep -i "longueur" $1.data| awk '{print $3}' | head -1`
NX=`grep -i "Nombre_de_Noeuds" $1.data| awk '{print $2}' | head -1`
D=`awk "BEGIN { print $X/($NX-1)/2.} "`

phi=`grep "flux_impose" $1.data| awk '{print $5}' | head -1`

V=`grep "vitesse_imposee" $1.data| awk '{print $6}' | head -1`
Tmoyen=`awk "BEGIN {print 2*$phi*$Y/($rho*$Cp*$V*$X)}" `
tail -3 $1_pbf_Nusselt.face > fin_nusselt

# Calculation of theoretical temperature gradient
```

```
# If imposed Nu, direct calculation, else, use of Kader
logi=`grep -i "Loi_Paroi_Nu_Impose" $1.data | head -1`
if [ logi == "" ]
then
  # reading of u* which should be different frome 0, Calculation of  y+andt T+ with Kader
  uetoi=`grep "u_tau" $1.data| awk '{print $7}' | head -1`
  yplus=`awk "BEGIN {print $D*$uetoi*$rho/$mu}" `
  beta=`awk "BEGIN {print (3.85*$Prandtl^(1/3)-1.3)^2+2.12*log($Prandtl)}" `
  gamma=`awk "BEGIN {print 0.01*($Prandtl*$yplus)^4/(1+5*$yplus*$Prandtl^3)}" `
   tplus=`awk  "BEGIN  {print  $Prandtl*$yplus*exp(-$gamma)+(2.12*log(1+$yplus)+$beta)*exp(-1/$gamma)}" `
  # DeltaTw=tplus*t_tau is deduced
  t_tau=`awk "BEGIN {print $phi/($uetoi*$Cp*$rho)}" `
  DeltaTw=`awk "BEGIN {print $t_tau*$tplus}" `
else
  Nu=`more fin_nusselt| awk '{print $7}' | head -1`
  DeltaTw=`awk "BEGIN {print $phi*$D/($lambda*$Nu)}" `
fi
awk -v Tmoy=$Tmoyen -v dT=$DeltaTw 'BEGIN {printf ("%.4f %s %s %.4f %s %.4f\n",Tmoy,"-","-",dT,"-",dT)}'  > theoric.dat


#reading of Tmoyen Trio_U
TSmoyen=`tail -1 Tmoyen_sortie | awk '{print $2}' | head -1 `


#reading of Tfluide et Tparoi Trio_U
Tfluide=`more fin_nusselt| awk '{print $11}' | head -1`
Tparoi=`more fin_nusselt| awk '{print $13}' | head -1`
Tparoi_eq=`more fin_nusselt| awk '{print $15}' | head -1`


#Calculation of thes deltaT Trio_U at the wall
deltaT=`awk "BEGIN {print $Tparoi-$Tfluide}" `
deltaT_eq=`awk "BEGIN {print $Tparoi_eq-$Tfluide}" `
```

#filling of the result file temp.dat

awk -v Tmoy=$TSmoyen -v Tf=$Tfluide -v Tp=$Tparoi -v Tp_eq=$Tparoi_eq -v dT=$deltaT -v dT_eq=$deltaT_eq 'BEGIN {printf ("%.4f %.4f %.4f %.4f %.4f %.4f\n",Tmoy,Tf,Tp,dT,Tp_eq,dT_eq)}' > temp.dat

exit

Annex 2: example of a prm file (*T_paroi.prm*)

Parametres *{*

Titre "Wall temperature verification in VEF discretisation with Neumann conditions"

Description "Calculations with imposed u, and Nusselt number imposed or free"*

Auteur "V.B."

CasTest "Nu_impose" "T_paroi.data" data file printed in report file

CasTest "Symet" "T_paroi.data"

CasTest "U_impose" "T_paroi.data"

VersionTrio_U "1.5.6_beta"

}

Chapitre *{*

Titre "Model description"

Figure *{*

Titre "Geometry, Mesh and Boundary Conditions"

Description "Dimensions : h=200 mm, L=30 mm "

Description "7 x 10 rectangular mesh cut with "'trianguler' option"

Description "Velocity inlet = 10 m/s ; Temperature inlet = 0°C"

Description "Lateral Heat flux = 2285 W/m2"

Description "Outlet, pressure = 0"

Width 8cm

Image ./T_paroi_geo1.png

}

Tableau *{*

Titre "Physical properties"

Description "Fluid domain: Helium"

Description "No Gravity effect"

nb_colonnes 1

Label Valeur

```
            Ligne {
                Legende "$\rho$ en kg/m$³$"
                  fichier ./rho.dat
            }
            Ligne {
                Legende "$\mu$ en N/m$²$/s"
                  fichier ./mu.dat
            }
            Ligne {
                Legende "$\lambda$ en W/m/K"
                  fichier ./lambda.dat
            }
            Ligne {
                Legende "Cp en J/kg/K"
                  fichier ./Cp.dat
            }
        }
}

Chapitre {
        Titre "Numerical Results"
Tableau {
        Titre "Analytical solutions for two calculation conditions"
        Description "The mean outlet temperature is deduced from global energy balance :"
        Description "\latex_($\rho\:Cp\:U\Delta Tes = \Phi,\: so \:\Delta Tes = 0.1\:K$\latex_) where $\Delta$Tes is the temperature difference between inlet and outlet."
        Description ""
        Description "The Nusselt number in Trio_U is a local heat exchange coefficient. Knowing this Nu number, we can deduce the theorical temperature gradient at the wall, i.e. the difference between wall and first fluid point temperatures :"
        Description "\latex_($\Delta Tw = \Phi\:d(\lambda\:Nu)$\latex_)where $\Delta$ Tw is this wall temperature difference, it will be taken at the channel outlet."
```

Description "In our case, the distance between these two points is d = 2.14e$^-$^{3}m"

Description "When we impose a thermal wall law (Kader), the local gradient, and so the Nusselt number, is calculated with this law"

}

}

***Chapitre** {*

Titre "First calculation : u=0 and Nu=20"*

***Tableau** {*

Titre "Comparisons of mean and wall temperature value at the outlet"

Description "As defined before, the theorical ΔTw can be known. We show in the following table, the different values calculated by Trio_U and compare with the analytical solution."

Description "The first ΔTw value corresponds to the difference between the first fluid temperature value and the face temperature at the wall. Thus it is disturbed by the convection term discretization."

Description "The second ΔTw "equiv" value corresponds to the difference between the first fluid temperature value and the equivalent wall temperature calculated with the local Nusselt number in Trio_U. "

Description " "

Description "The same calculation has be done with two different boundary conditions at the wall : 'symetrie' or 'paroi_fixe' to see the effect of non-tangential velocities at the edge"

Description " "

nb_colonnes 6

label Tmean Outlet | Tfluid | Twall (face) | ΔTw | Twall "equiv" | ΔTw "equiv"

Ligne {

Legende "Theorical value"

fichier ./Nu_impose/theoric.dat

}

Ligne {

Legende "Trio_U paroi_fixe"

fichier ./Nu_impose/temp.dat

origine trio_u

}

Ligne {

Legende "Trio_U symet"

fichier ./Symet/temp.dat

origine trio_u

}

}

}

***Chapitre** {*

Titre "Second calculation : u=0.2 and thermal law function"*

***Figure** {*

Titre "New Boundary conditions"

Width 8cm

Image ./T_paroi_geo2.png

}

***Tableau** {*

Titre "Second calculation results"

Description "Here, the wall friction velocity is imposed. Thus, the wall temperature gradient is calculated with this u value. The theorical value of ΔTw is given thanks to the Kader law. We compare it with the Trio_U results, resumed in the table below."*

Description " "

nb_colonnes 6

label Tmean Outlet | Tfluid | Twall (face) | ΔTw | Twall "equiv" | ΔTw "equiv"

Ligne {

Legende "Theorical value"

fichier ./U_impose/theoric.dat

}

Ligne {

Legende "Trio_U paroi_fixe"

fichier ./U_impose/temp.dat

origine trio_u

}

}

***Tableau** {*

Titre "Conclusions"

Description "We can note than the temperature gradient calculated with the Trio_U face temperature (*) is very far from the analytical one. This is due to the convection terms in the VEF discretization method."

Description "Because of this convection effect, the modification of the local Nusselt number as no visible effect on the wall temperature."

Description "When we take the wall "equivalent" temperature calculated with the thermal wall law, or the local Nusselt number (*), the ΔTw_eq is perfectly correct."

Description "We can assure that the total wall heat is brougth to the fluid since the mean temperature at the channel outlet is correct."

Description " "

Description "(*): this temperature values can be found in the "T_paroi_pbf_Nusselt.face" output file."

}

***tableau_performance** {*

Titre "Computing performance"

}

}

Annex 3: example of report file (*T_paroi.pdf*)

2 MODEL DESCRIPTION

Wall temperature verification in VEF discretisation with Neumann conditions

1 Introduction

Validation made by : V.B..
Report generated 19/01/2009.

1.1 Description

Calculations with imposed u*, and Nusselt number imposed or free

1.2 Parameters Trio_U

- Version Trio_U : 1.5.6_beta
- Version Trio_U from out: /work/triou/Validation/Trio_U_mpi_opt (1.5.6_beta)

1.3 Test cases

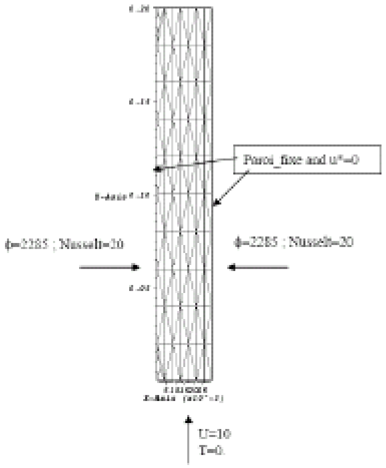
- Nu_impose/T_paroi.data : *impression du jeu de données en fin de fichier*
- Symet/T_paroi.data :
- U_impose/T_paroi.data :

2 Model description

2.1 Geometry, Mesh and Boundary Conditions

Dimensions : h=200 mm, L=30 mm
7 x 10 rectangular mesh cut with 'trianguler' option
Velocity inlet = 10 m/s ; Temperature inlet = 0 C
Lateral Heat flux = 2285 W/m2
Outlet, pressure = 0

Wall temperature verification in VEF discretisation with Neumann conditions 1



2.2 Physical properties

Fluid domain: Helium
No Gravity effect

	Valeur
ρ en kg/m^3	4.4
μ en $N/m^2/s$	4.4e-05
λ en $W/m/K$	0.34
Cp en $J/kg/K$	5193.0

3 Numerical Results

3.1 Analytical solutions for two calculation conditions

The mean outlet temperature is deduced from global energy balance :
$\rho\, Cp\, U \Delta Tes = \Phi$, so $\Delta Tes = 0.1\ K$ where ΔTes is the temperature difference between inlet and outlet.

The Nusselt number in Trio_U is a local heat exchange coefficient. Knowing this Nu number, we can deduce the theorical temperature gradient at the wall, i.e. the difference between wall and first fluid point temperatures :
$\Delta Tw = \Phi\, d (\lambda\, Nu)$where Δ Tw is this wall temperature difference, it will be taken at the channel outlet.
In our case, the distance between these two points is d = $2.14e^{-3}$m
When we impose a thermal wall law (Kader), the local gradient, and so the Nusselt number, is calculated with this law

4 First calculation : u*=0 and Nu=20

4.1 Comparisons of mean and wall temperature value at the outlet

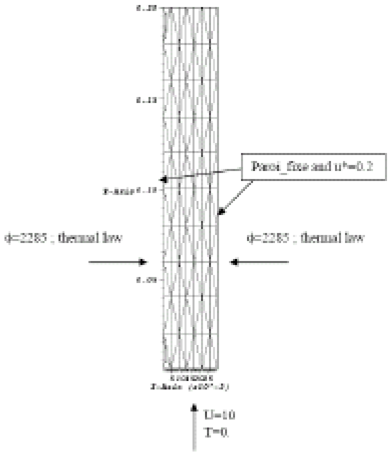
As defined before, the theorical ΔTw can be known. We show in the following table, the different values calculated by Trio_U and compare with the analytical solution.
The first ΔTw value corresponds to the difference between the first fluid temperature value and the face temperature at the wall. Thus it is disturbed by the convection term discretization.
The second ΔTw equiv value corresponds to the difference between the first fluid temperature value and the equivalent wall temperature calculated with the local Nusselt number in Trio_U.

The same calculation has be done with two different boundary conditions at the wall : 'symetrie' or 'paroi_fixe' to see the effect of non-tangential velocities at the edge

	Tmean Outlet	Tfluid	Twall (face)	ΔTw	Twall equiv	ΔTw equiv
Theorical value	0.1333	-	-	0.7201	-	0.7201
Trio_U paroi_fixe	0.1333	0.4629	0.5377	0.0747	1.182	0.7191
Trio_U symet	0.1333	0.4624	0.5375	0.0751	1.1815	0.7191

5 Second calculation : u*=0.2 and thermal law function

5.1 New Boundary conditions



5.2 Second calculation results

Here, the wall friction velocity is imposed. Thus, the wall temperature gradient is calculated with this u* value. The theorical value of ΔTw is given thanks to the Kader law. We compare it with the Trio_U