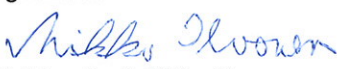




TRICOT

# PORFLO development, applications and plans in 2008-2009

Authors: Mikko Ilvonen, Ville Hovi, Pasi Inkinen

Confidentiality: Public

Report's title PORFLO development, applications and plans in 2008-2009	
Customer, contact person, address SAFIR2010 Research Programme (VYR,VTT)	Order reference
Project name TRICOT/SAFIR2010	Project number/Short name 32754
Author(s) Mikko Ilvonen, Ville Hovi, Pasi Inkinen	Pages 28/
Keywords Porous media model, two-phase flow	Report identification code VTT-R-01414-10
Summary  This report describes the development done in the PORFLO code and the improvements made for the usability of the code under the auspices of TRICOT/SAFIR2010 project in 2008-2009. Some results of the applications dealt with during this time period, the BFBT benchmark and the horizontal steam generator of Loviisa VVER-440 plant, are presented. Finally, plans for future development and applications are discussed.	
Confidentiality	Public
Espoo 17.2.2010	
Signatures 	Signatures 
Written by Mikko Ilvonen Team leader	Reviewed by Elina Syrjälähti Project Manager
	Signatures 
	Accepted by Timo Vanttola Technology Manager
VTT's contact address VTT, PL 1000, FIN-02044 VTT	
Distribution (customer and VTT) SAFIR2010 Reference group 3	
<i>The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.</i>	

Report's title PORFLO development, applications and plans in 2008-2009		
Customer, contact person, address SAFIR2010 Research Programme (VYR,VTT)		Order reference
Project name TRICOT/SAFIR2010		Project number/Short name 32754
Author(s) Mikko Ilvonen, Ville Hovi, Pasi Inkinen		Pages 28/
Keywords Porous media model, two-phase flow		Report identification code VTT-R-01414-10
Summary  This report describes the development done in the PORFLO code and the improvements made for the usability of the code under the auspices of TRICOT/SAFIR2010 project in 2008-2009. Some results of the applications dealt with during this time period, the BFBT benchmark and the horizontal steam generator of Loviisa VVER-440 plant, are presented. Finally, plans for future development and applications are discussed.		
Confidentiality	Public	
Espoo 17.2.2010		
Signatures	Signatures	Signatures
Written by Mikko Ilvonen Team leader	Reviewed by Elina Syrjälähti Project Manager	Accepted by Timo Vanttola Technology Manager
VTT's contact address VTT, PL 1000, FIN-02044 VTT		
Distribution (customer and VTT) SAFIR2010 Reference group 3		
<i>The use of the name of the VTT Technical Research Centre of Finland (VTT) in advertising or publication in part of this report is only permissible with written authorisation from the VTT Technical Research Centre of Finland.</i>		

## Contents

1	Introduction.....	3
2	Development .....	3
2.1	Code architecture.....	4
2.2	Momentum source terms .....	4
2.2.1	Interphase drag.....	5
2.3	Solver development .....	6
2.4	Scalar transport.....	8
2.5	Turbulence model .....	8
3	Usability.....	10
3.1	Parallelization plans .....	10
3.2	Post-processing .....	12
3.2.1	Matlab post-processing script .....	12
3.2.2	3D post-processing .....	12
3.3	Monitoring of residuals.....	15
4	Applications .....	17
4.1	BFBT benchmark.....	17
4.2	VVER Steam Generator.....	20
5	Planned applications .....	23
5.1	COOLOCE simulations .....	23
5.2	PSBT benchmark.....	24
5.3	EPR reactor core .....	25
5.4	Connection with Smabre.....	26
	References .....	27

## 1 Introduction

PORFLO is a 3D flow solver, based on porous media modelling, for two-phase flow of water and steam. The current state of the code is briefly described here.

### **PORFLO: Essential features in a nutshell (as of January 2010)**

- 3D solution of two-phase flow of water and steam
- Six-equation model in the usual sense (mass, energy and momentum conservation of water and steam)
- Cartesian staggered grid (velocities expressed at faces of pressure cells)
- Regarding the flow solution, geometry is described only as porosity (fraction of fluid volume of cell volume), but in other physical models (like heat transfer) any additional knowledge of the geometry may be utilized.
- Solution of common pressure and phase velocities is based on variants of the iterative SIMPLE algorithm
- Several iterative methods for solution of linear systems of equations
- Number of grid cells: quite usual is 100 000; even millions have been used but then calculation times may be several weeks, depending on other factors.
- First turbulence model (k-epsilon) is only in the testing phase; source terms for porous zones remain a subject of future work.
- Steam tables should be implemented in a proper way.
- Heat transfer and other correlations need a lot of development (application specific).
- Visualization of results has been done by Matlab scripts; now an easy-to-use tool, based on OGRE 3D graphics library and Qt toolkit, is being developed.

As a general rule it can be said that the PORFLO simulation of a certain new application will only be successful if appropriate closure laws (usually experimentally based correlations) for the right-hand side source terms of the basic conservation equations are found and if the equations can then still be solved in a converging manner. This situation is much the same as for the ‘usual’ CFD approach (without porous medium assumption).

## 2 Development

The development done in the PORFLO code during 2008-2009 roughly consists of general code development, concentrating more on the six-equation model of PORFLO, and application specific model development, the closure laws needed for the right-hand sides of various conservation equations. These development efforts are described in the following subsections.

## 2.1 Code architecture

The general architectural strategy in the development of the six-equation model has been to prevent version forking and to keep PORFLO as one single code version, which still contains even the older solution methods (five-equation approach by Jaakko Miettinen, with various linear system solution methods). Method of solution can be chosen by appropriate code parameters. This strategy was devised in order to check calculation results of the different methods against each other, and to remain on the ‘safe side’ even when new methods turned out to have serious problems. Eventually in the future, it is possible to clear the code of the obsolete methods, provided that sufficient experience and certainty of superiority of the new methods is accumulated.

A lot of attention is constantly paid to keeping the code clear, readable and easily understandable. This comprises structure, style of writing, choice of variable and other names, as well as good comment lines. Comment lines describing even the latest developments and changes were immediately added in the code.

For new applications, possibly to be calculated by new staff, the readability of the source code is being enhanced, especially to facilitate quicker finding of critically important places where to do changes, as everything is not contained in the input files (and to provide that would be a substantial amount of work).

In the code architecture, some rearrangements were done in late 2009 in order to incorporate more easily various applications. As PORFLO does not have a universal user interface suitable for any kind of application, many application-specific things are ‘hard-coded’ in the source code. This is not uncommon as such; for example, the well-known open-source CFD code OpenFoam almost always necessitates some source code editing by the user. The purpose in PORFLO code architecture is to keep one code version, in which application-specific parts are clearly isolated and marked, in the case that they cannot be specified in the input files only.

## 2.2 Momentum source terms

Interphasial friction as well as tubes-water and tubes-steam friction (for horizontal tubes like in a horizontal steam generator) were formulated according to (Simovic et al. 2007).

### 2.2.1 Interphase drag

Calculation of the interphase drag force is crucial, since it dictates the velocity difference between the phases, also known as phase separation. For dispersed vapor bubbles in liquid the interfacial drag force is of the form

$$\mathbf{F}_{id} = \frac{3}{4} \alpha \rho_l \frac{C_D}{D_B} |\mathbf{u}_g - \mathbf{u}_l| (\mathbf{u}_g - \mathbf{u}_l) \quad (2.1)$$

where  $C_D$  interfacial drag coefficient [-],  
 $D_B$  bubble diameter [m] and  
 $\mathbf{F}_{id}$  interphase drag force per unit volume [N/m<sup>3</sup>].

The correlations for the ratio of interfacial drag coefficient and bubble diameter are taken according to the presentation of Simovic, Ocokoljic and Stevanovic (2007). The correlation proposed consists of two parts. The first part, valid for bubbly flow regime ( $\alpha < 0.3$ ), has been adopted from Ishii and Zuber (1979), in which the authors have introduced a minor modification: the original correlation is multiplied by a factor 0.4, to make a better fit for their experimental data, which was bubbly flow over a horizontal tube bundle.

$$\frac{C_D}{D_B} = 0.267 \left( \frac{g \Delta \rho}{\sigma} \right)^{\frac{1}{2}} \left[ \frac{1 + 17.67 f(\alpha)^{\frac{6}{7}}}{18.67 f(\alpha)} \right]^2 \quad (2.2)$$

where  $g$  acceleration of gravity [m/s<sup>2</sup>],  
 $\Delta \rho$  the difference between liquid and vapor density [kg/m<sup>3</sup>],  
 $\sigma$  surface tension [N/m] and  
 $f(\alpha)$  the dependence on void fraction as follows

$$f(\alpha) = (1 - \alpha)^{\frac{3}{2}} \quad (2.3)$$

The second part of the correlation for churn-turbulent flows, ( $\alpha > 0.3$ ), is given through

$$\frac{C_D}{D_B} = 1.487 \left( \frac{g \Delta \rho}{\sigma} \right)^{\frac{1}{2}} (1 - \alpha)^3 (1 - 0.75\alpha)^2 \quad (2.4)$$

where according to Simovic, Ocokoljic and Stevanovic the dependence on void fraction has the same form as the CATHARE code. The dependence of the ratio of interphase drag coefficient and bubble diameter on void fraction is presented in Figure 2.1.

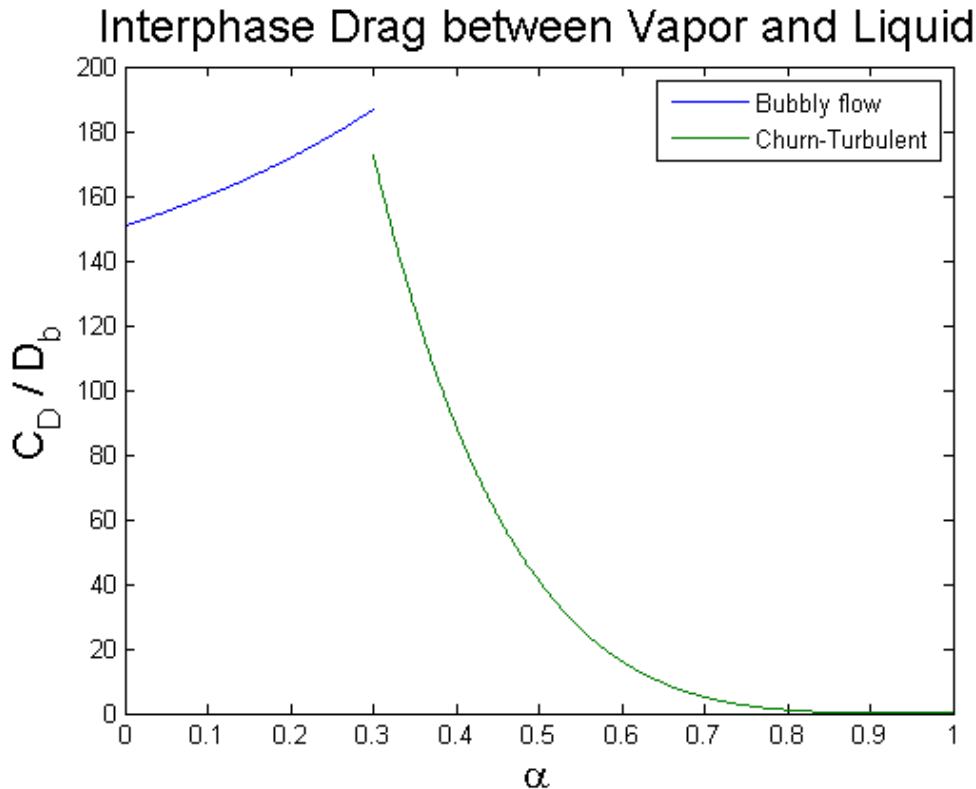


Figure 2.1: Interphase friction.

It is seen from Figure 2.1 that the lines do not actually join. This lead to oscillations inside the SIMPLE iteration, which in turn caused the SIMPLE iterations to stagnate, when the choice between the two parts of the correlation was made based on the implicitly calculated void fraction (at the end of the time step). As a solution to this the choice between the two parts of the correlation is now made based on the local void fraction at the beginning of the time step, which doesn't change between iterations. Another remedy would be to smooth the transition between the two parts of the correlation, and then the implicitly calculated void fraction could be used as the deciding parameter. The latter option would perhaps be less prone to oscillations (from one time step to the next), but would require additional coding effort.

## 2.3 Solver development

PORFLO-related work in TRICOT in early 2008 was concentrated on fine-tuning the SIMPLE pressure correction algorithm variants that were implemented and adapted to the five-equation two-phase model of PORFLO. It turned out that volume conservation is a more suitable starting point to solve pressure corrections than mass conservation. A Master of Science in Technology thesis "Calculations of Boiling Two-Phase Flow Using a Porous Media Model" (Ville Hovi, LUT, 112 p.) was completed in May 2008; the SIMPLE solution approach in PORFLO is thoroughly described. Effect of parameters of the SIMPLE algorithm was systematically assessed. Simulations of BWR fuel bundle boiling flow were performed for a 2x2 rods



simplified bundle. Void fractions, velocities and temperatures were plotted as 2D at several levels along the length of the bundle. As some questionable details remained in the simulated flow, the suitability of five-equation (vs. six-equation) model for 3D two-phase flow was critically considered.

First steps towards the implementation of a six-equation model, the phase-coupled SIMPLE algorithm, were taken by adding separate momentum equations, and separate terms into the pressure correction equations, for both water and steam. Then, new solvability problems emerged as a result of the extremely small void fractions that exist in the lower part of the fuel bundle and in the beginning of the simulations.

The problems with convergence of the six-equation solution encountered in the beginning have apparently been solved. They were mostly related to the practical code-level handling of cell-wise void fractions. The appearance of even very small void fractions could make the SIMPLE-based iteration diverge. In some cases, the iteration might get stuck oscillating between two states. The main ingredient of solvability turned out to be taking the phase-specific flow area between cells (i.e. the area 'pushed' by the pressure gradient) as harmonic or volume average of the cells, instead of the upstream value.

Better foundations for solvability were further laid by coding new iterative linear solvers, including the BiCG (Bi-Conjugate Gradient) method, and some preconditioning methods, like the basic Jacobi (scaling of the diagonal) and SSOR (Symmetric Successive Over-Relaxation) preconditioning. The computation time was reduced almost by an order of magnitude, thanks to these developments. However, it is still prohibitively large if the mesh is dense and numerous different cases are to be simulated. Approximately half the time is spent on calculating the pressure corrections.

The effect of non-conservative formulation of the momentum equations on convergence of the solution algorithm was studied. Quite significant improvements (in some cases over 50 % reductions in the number of iterations) were obtained using the non-conservative formulation instead of the conservative one.

After the development efforts, it is at present easily possible to use the momentum conservation equation either in primitive or in conservative form. Additionally, solution of volume fractions

was improved by solving volumes of both phases, and only then expressing as volume fractions. Furthermore, a new separate iteration was added for solving of heat transfers and temperatures.

In connection with developing the k-epsilon model, problems related to the flow solver were discovered. It seems that the numerical algorithm of solution may produce artificial (non-physical) oscillations of cross-flows in cases where only homogeneous flow in the main flow direction should emerge as a solution. The oscillations are of very low magnitude and would not cause any harm as such, but unfortunately they seem to have a very pronounced effect on the calculated turbulent kinetic energy. Currently it seems that some work is needed once again with flow solver. This was not originally planned for year 2009.

A correction was also made to the relaxation scheme in the SIMPLE algorithm. After pressure corrections, the velocity corrections should be calculated from the 'clean' (unrelaxed) pressure field.

## 2.4 Scalar transport

A general-purpose scheme has been created for scalar transport. This was inspired by the need to calculate transport of the turbulent kinetic energy  $k$  in the turbulence model. The transported scalar may be e.g. the concentration of any species solved in the water. It is now easy to add new transported scalars as the user wishes. It is possible to choose between conservative and primitive form of the transport equation. Some tests of numerical diffusion are being performed.

One of the most important uses of the scalar transport scheme is the transport and spreading of boric acid concentration in PWR, especially when calculating transients at the beginning of fuel cycle. It will be necessary to account for higher density of boron-containing water.

## 2.5 Turbulence model

Development of a turbulence model was begun in fall 2009. The most widely used of the two-equation models, the  $k$ - $\epsilon$  model developed by Launder and Spalding (1974), was chosen. From the perspective of the entire solution process this adds two scalar fields to the list of solved variables: these are the turbulence kinetic energy per unit volume ( $k$ ) and its dissipation rate ( $\epsilon$ ). As a first approach the so called mixture model was chosen, in which the turbulence kinetic energy and turbulence dissipation rate are solved for the mixture of the two phases. The transport equations for turbulence kinetic energy and turbulence dissipation rate

( $k$  and  $\varepsilon$ ), along with the corresponding production terms, are solved in a slightly different form compared to Launder and Spalding (1974), for the reason that the original transport equations do not include density and were developed for finite-difference methods, the latter affecting the production terms in particular. Apart from these modifications the basic principles, model assumptions and constants of the original model still apply. The model constants, which were arrived at after extensive studies (Launder and Spalding, 1974), are given in Table 2.1.

Table 2.1: The model constants of the  $k$ - $\varepsilon$  model.

$C_\mu$	$C_1$	$C_2$	$\sigma_k$	$\sigma_\varepsilon$
0.09	1.44	1.92	1.0	1.3

The transport equations of turbulence kinetic energy and its dissipation rate, in the form that they are solved in PORFLO, are given through

$$\frac{\partial(\rho_m k_m)}{\partial t} + \nabla \cdot (\rho_m k_m \mathbf{u}_m) = \nabla \cdot \left[ \left( \mu_m + \frac{\mu_t}{\sigma_k} \right) \nabla k \right] + G_k - \rho_m \varepsilon \quad (2.5)$$

$$\frac{\partial(\rho_m \varepsilon_m)}{\partial t} + \nabla \cdot (\rho_m \varepsilon_m \mathbf{u}_m) = \nabla \cdot \left[ \left( \mu_m + \frac{\mu_t}{\sigma_\varepsilon} \right) \nabla \varepsilon \right] + C_1 \frac{\varepsilon}{k} G_k - C_2 \frac{\varepsilon^2}{k} \quad (2.6)$$

where

$C_1, C_2$	model constants [-],
$G_k$	turbulence production term [kg/ms <sup>3</sup> ],
$k_m$	mixture turbulence kinetic energy per unit volume [m <sup>2</sup> /s <sup>2</sup> ],
$t$	time [s],
$\mathbf{u}_m$	mixture velocity (vector) [m/s],
$\varepsilon$	dissipation rate of mixture turbulence energy [m <sup>2</sup> /s <sup>3</sup> ],
$\mu_m$	mixture dynamic viscosity [Ns/m <sup>2</sup> ],
$\mu_t$	mixture turbulent viscosity [Ns/m <sup>2</sup> ],
$\rho_m$	mixture density [kg/m <sup>3</sup> ],
$\sigma_k$	turbulent Prandtl number for $k$ [-] and
$\sigma_\varepsilon$	turbulent Prandtl number for $\varepsilon$ [-].

The turbulence production term  $G_k$  is given through

$$G_k = \mu_t \sum_{i=1}^3 \sum_{j=1}^3 \left[ \left( \frac{\partial \mathbf{u}_i}{\partial x_j} + \frac{\partial \mathbf{u}_j}{\partial x_i} \right) \frac{\partial \mathbf{u}_i}{\partial x_j} \right] \quad (2.7)$$

The  $k$ - $\varepsilon$  turbulence model has been implemented in PORFLO, but further work is still needed. While testing the model, spurious oscillations were produced throughout the computation

domain in both the turbulence energy and its dissipation fields. The source of these oscillations was traced back to oscillations in the solved velocity field. Though the oscillations in the computed velocity field were minor, and would not cause any concern as such, the effect on the turbulence energy field was surprisingly pronounced.

For the time being, the model has appropriate source terms at known walls only, not in porous medium; it will be a task of year 2010 to add them to the model. These source terms for porous zones will be developed following the guidelines given in (Serre & Bestion 2005). In either case (open medium or porous volume average) the most difficult part of the development are the right-hand-side source terms of the equations. Once solved, the turbulent kinetic energy will affect many processes, most importantly the flow field through turbulent viscosity, but also enhancement of heat transfer, mixing of void fraction and scalar concentrations etc. Currently the predicted values of  $k$  are being initially checked, but that is difficult before their coupling to other processes, because of the feedback effect.

### **3 Usability**

The various efforts made to improve the usability of the code are discussed in the following subsections.

#### **3.1 Parallelization plans**

In October 2008, a new powerful Linux cluster ‘Ahjo’ (5 computers, each double-CPU, each CPU a quad-core Intel Xeon) was installed at VTT. After some initial difficulties with compilation, PORFLO has since then been run on the cluster, but still in a serial fashion. Even this immediately facilitates performing several slightly perturbed simulations at the same time.

Some very initial investigations in MPI parallelization have been done. The Ahjo cluster has OpenMPI, the open source code implementation of the MPI (Message Passing Interface) standard, which is developed with the objective of being the best of all MPI implementations. The library of OpenMPI calls is vast and can serve many purposes of parallel applications, but only a small subset are necessarily needed to create a parallel-running simulation code.

The following potential levels of parallelization have been initially discussed, ranging from low-level (fine-grained) to high-level (coarse-grained):

1. Parallelization of big basic multiplications, e.g. in the linear solvers. For example, the multiplication of vector by a big sparse matrix is straightforward to perform by dividing the matrix into ‘stripes’. Each processor core can do its own stripe. In worst case, communication time may be comparable to computation time. This kind of matrix-vector operations are common in unpreconditioned linear solvers, but preconditioning, like used in PORFLO, makes the situation more complicated and parallelization is not that straightforward any more. However, it is an open question, whether a parallel unpreconditioned linear solution could compute faster than a serial but preconditioned one.
2. Parallelization of the whole iterative solution of a linear system of equations. Thinking of the usual band-diagonal matrix (with seven non-zero diagonals) resulting from a 3D domain equation system, ‘stripes’ of consecutive equations could be allocated to different processors for iterative solution. Resembling the old ADI (Alternating Direction Implicit) method, the unknown variables outside the current processor’s block would be ‘frozen’ as constants. However, values of these constants would be updated between iteration rounds by inter-processor communication.
3. Running of several complete PORFLO processes in parallel. The advantage of this kind of highest-level parallelization by domain decomposition would be that all operations (the complete code) would run in parallel, albeit that some small identical operations would unnecessarily be done by all processes. When solving a time step iteratively, the processes would communicate about values at their common boundary after every iteration round. In Apros development there are plans to split the system of a simulated plant among several complete Apros processes in the future; however, in the case of a system code, the boundaries are at least smaller than in 3D, even when there are a huge number variables to communicate.

There are many inevitable drawbacks in parallel programming. Among the worst in the development phase is the difficulty of debugging. For example, when the code hangs, the origin of the problem could be in any one of the processes, which all have a different state. The calculated results will usually be at least slightly different from a serial and parallel run. Another difficulty, without very good solutions at the moment, is the problem of output coming from several processes at the same time. This means that e.g. one time instant will have to be picked out of several output files.

Possibly the best way to start parallel tests would be to use a ready-made library, like the publicly available open source code ScaLAPACK, which can be used on any computer supporting MPI.

## 3.2 Post-processing

Two post-processing tools were developed to facilitate debugging of the code. One of them is a Matlab-based script for creating 2D contour and vector plots. The other tool, based on the OGRE 3D graphics library and the QT user interface toolkit, is capable of both 2D and 3D operation.

### 3.2.1 Matlab post-processing script

This is a simple Matlab script meant for quickly obtaining 2D cross-sections perpendicular to the Cartesian coordinates of a chosen variable in PORFLO. The script works by first reading the variable defined by the user from a file that PORFLO automatically generates after each successful time step (restart file) into a 3D array. Then the chosen cross-sections are formed and plotted. The plotted figures can then be modified, if necessary, using the tools available in Matlab and saved to an appropriate picture format.

Attempts have been made to automate the script as much as possible; therefore the user defined options are gathered at the beginning of the script. These options include the selection of the plotted variable, selection of the plotting planes (one for each direction), resolution of interpolation, interpolation mode (no interpolation, linear and cubic interpolation), resolution of the vector plots (how many vectors are plotted in each direction) and scaling factors for controlling the length of the plotted vectors.

### 3.2.2 3D post-processing

StarNode (Scientific Tool for Analyzing and Rendering Nodal Objects and Data Elements) is a 3D visualization tool under development for 3D simulation codes, including PORFLO. StarNode has been written in C++ programming language, and the program has basic functionality for analyzing both 2D and 3D nodal data. Currently, PORFLO data can be read directly from an output file used for simulation restarts.

The 3D visualization implementation has been developed using OGRE (Object-oriented Graphics Rendering Engine), which is a cross-platform 3D rendering engine to facilitate the

development of modern hardware-accelerated 3D graphics applications in DirectX and OpenGL programming environments. By using a hardware-accelerated 3D rendering engine and advanced rendering methods (e.g. batching), it would theoretically be possible to render millions of nodes simultaneously in real time with modern graphics processing units. However, the current implementation of StarNode is capable of rendering several hundreds of thousands of nodes due to memory constraints.

Qt development framework (QT) has been used in the development of StarNode to provide a simple and straightforward graphical user interface for analyzing 3D data. An example view of the user interface is presented in Figure 3.1. The graphical layout is divided into model explorer, properties editor and rendering area. Model explorer presents the model structure in hierarchical view. Properties editor is used to edit different settings of the model and objects. The current view with the applied settings is presented in rendering area, in which the viewpoint can be freely moved and rotated around the object space.

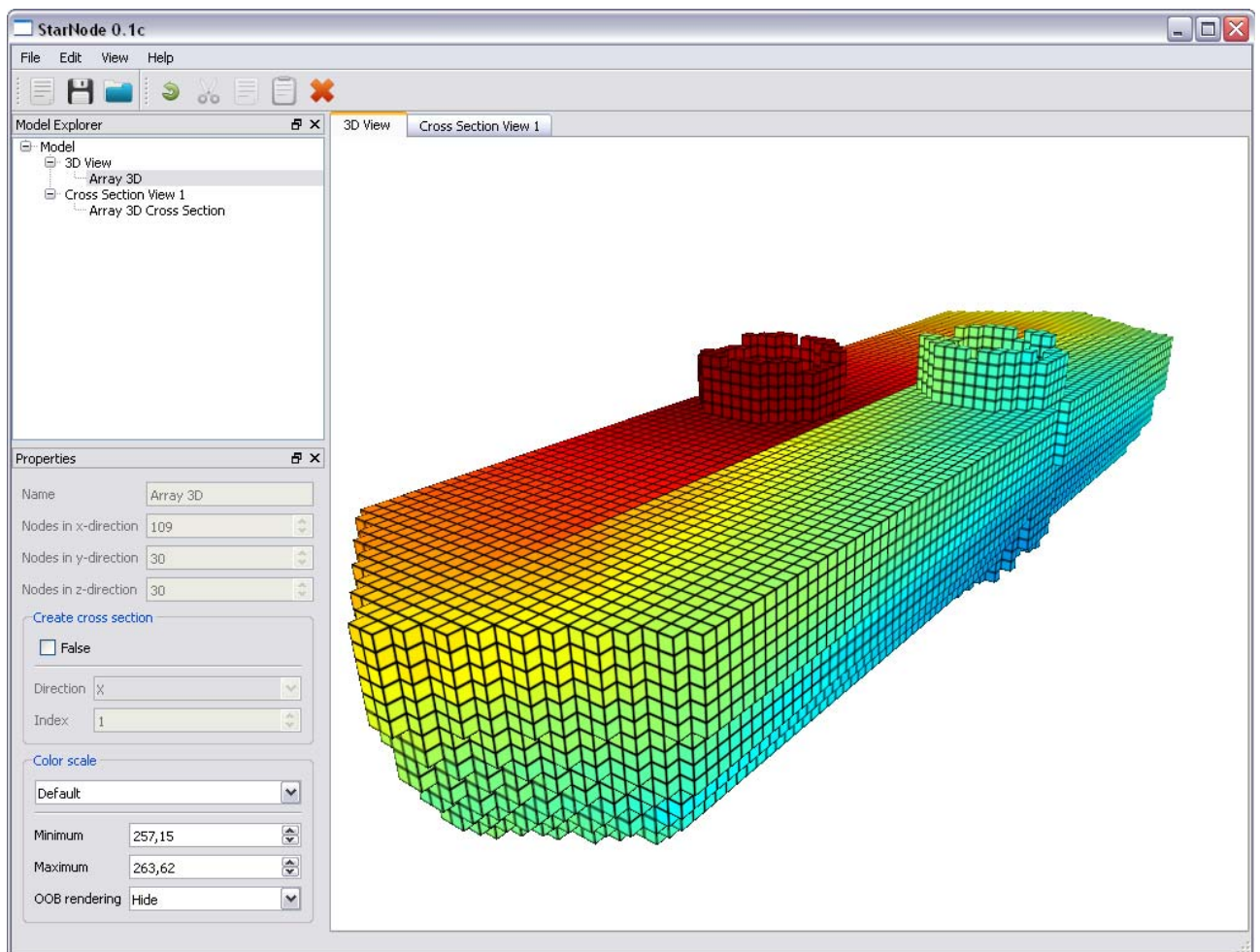


Figure 3.1: Example view of StarNode user interface.

StarNode model is structured to enable diverse analysis possibilities of multiple data sets simultaneously. For rendering purposes, model is divided into views, and each view can have multiple objects, such as node groups and text elements. Node groups can be viewed in their entirety or as a cross section in a chosen direction. In addition, cross sections in 2D views can be saved as SVG (Scalable Vector Graphics) files. Example images rendered in StarNode are presented in Figure 3.2. Text elements can be defined to show input data or values of the selected properties. Color scales are defined on a per-object basis, and multiple objects can be defined to use the same scale. Three different options have been implemented for rendering values outside the scale: clamp, hide and highlight.

Current version of StarNode (0.1c) has two general rendering options. In default mode, nodes are rendered in a conventional manner, i.e. nodes closest to the viewpoint are rendered topmost. Another visualization option - maximum (or minimum) intensity projection - can be used for more advanced analysis of 3D data. In this mode, nodes with the highest (or the lowest) values are rendered topmost. The method is especially useful for analyzing distributions with clearly distinguishable regions, e.g. temperature distribution during the insertion of cold feedwater into a system.

Future development plans of StarNode include animation generation, support for higher number of nodes, option for 3D vector graphics exporting and cross-platform support. Other additional features will be a more general input format and customizable data import option to provide better customizability of input data structure. Furthermore, more versatile editor options for the graphical user interface will be developed.



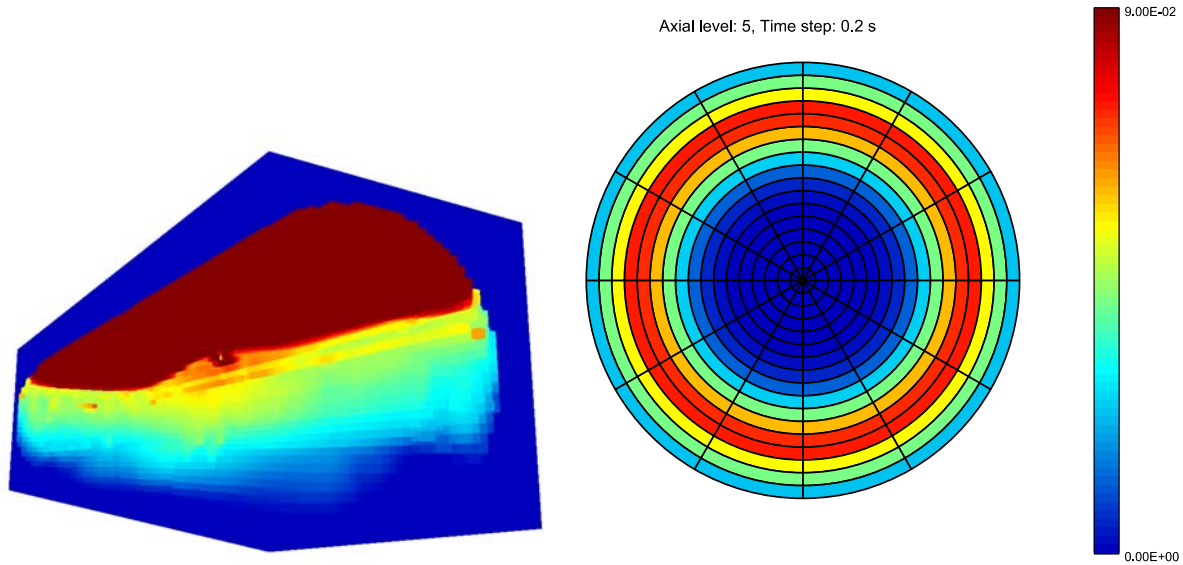


Figure 3.2: Example images rendered in StarNode.

### 3.3 Monitoring of residuals

Real-time monitoring of solution residuals was made possible for both Windows and Linux platforms. A file that contains the residual data is output at the end of each iteration from PORFLO. Two scripts, a Matlab script and a gnuplot script, were made which read the data file and plot the residuals to the screen. The Matlab script is intended to be used in Windows and the gnuplot script in Linux platforms. The number of iterations to be plotted, in other words the range of the x-axis in the residual plot, can be adjusted by a parameter set in PORFLO. Screenshots of the residual windows are presented in Figures 3.3 and 3.4 for the Matlab and gnuplot scripts, respectively. The variables that are plotted are: 1) 'Vol' scaled 1-norm of the RHS of the pressure correction equation (mass imbalance), 2) 'U<sub>l</sub>' scaled norm of the residual of x-directional liquid momentum equation, 3) 'V<sub>l</sub>' scaled norm of the residual of y-directional liquid momentum equation, 4) 'W<sub>l</sub>' scaled norm of the residual of z-directional liquid momentum equation, 5) 'U<sub>g</sub>' scaled norm of the residual of x-directional vapor momentum equation, 6) 'V<sub>g</sub>' scaled norm of the residual of y-directional vapor momentum equation, 7) 'W<sub>g</sub>' scaled norm of the residual of z-directional vapor momentum equation, 8) 'T<sub>l</sub>' scaled norm of the residual of liquid energy equation and 9) 'T<sub>g</sub>' scaled norm of the residual of vapor energy equation.

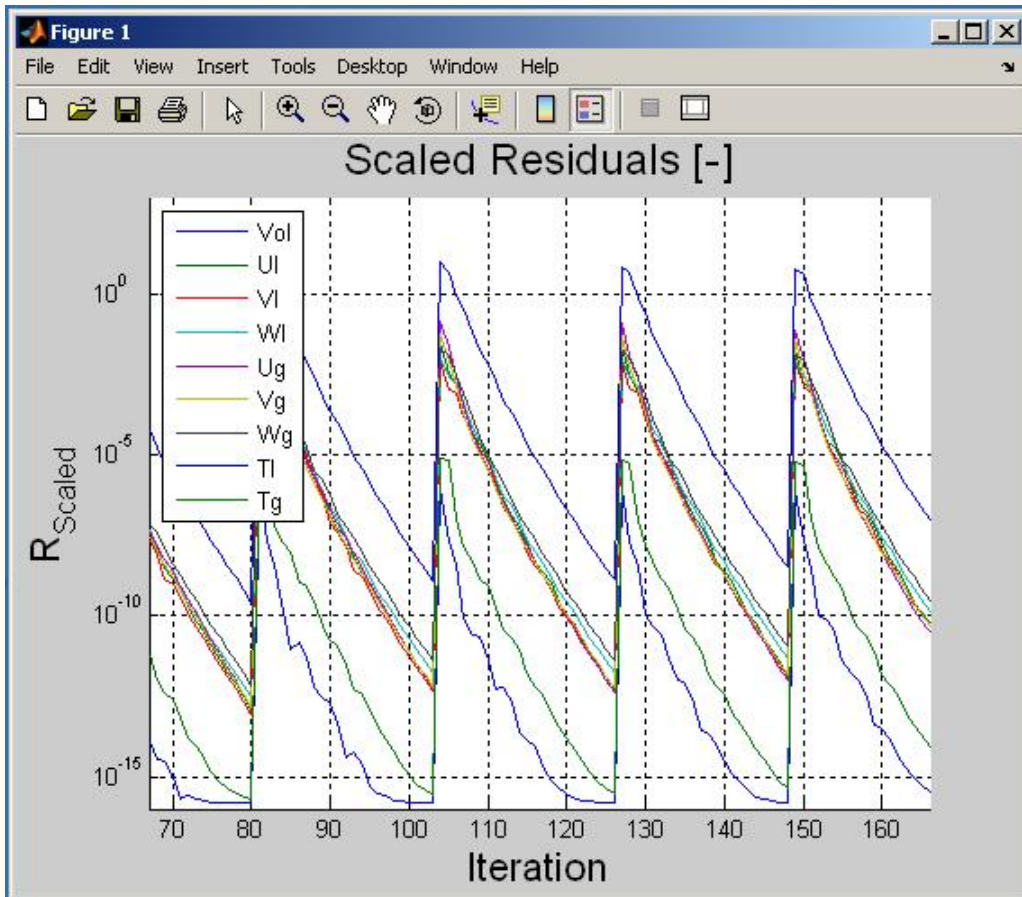


Figure 3.3: Matlab residual window.

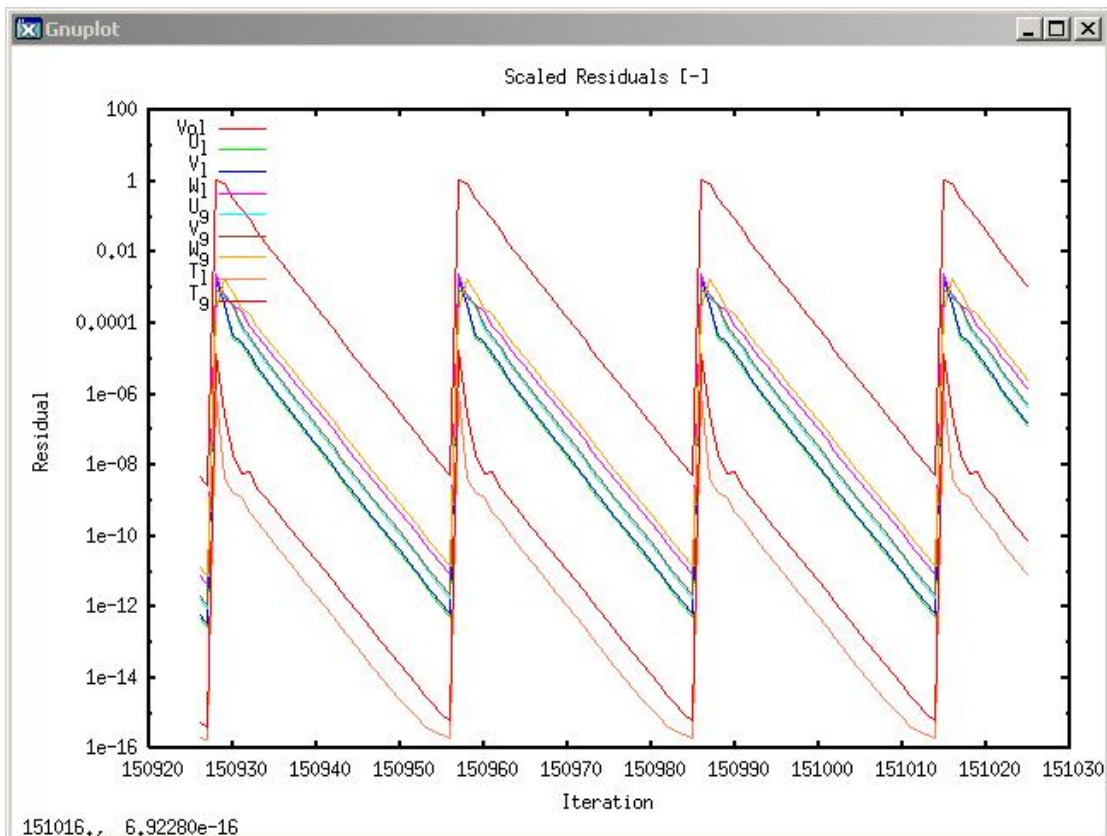


Figure 3.4: gnuplot residual window.

## 4 Applications

Past and present applications of PORFLO are briefly summarized here before discussion of the present applications.

### 1) Past (by the original author Jaakko Miettinen)

- Particle bed dryout simulations
- Isolation condenser

### 2) Present

- BFBT benchmark (BWR fuel bundle)
- Loviisa VVER horizontal steam generator secondary side

Most appropriate applications for a porous medium model are components having complex geometry such that it is not possible to use a computational grid fitted to the structures – partially because the practical work of meshing the geometry would be very time-consuming, and partially because the resulting mesh would have a prohibitively large number of cells, at least leading to unacceptable computation, if not also other problems in the solution. Fuel bundles are, regarding their geometrical complexity, somewhat on the verge between porous medium approach and structure-fitted approach. The latter are increasingly appearing in publications, as available computing powers are increasing.

### 4.1 BFBT benchmark

After extensive debugging of the newly developed code in PORFLO, the first stable simulations of the BFBT ‘simplified test bundle’ (2x2 fuel rods) were achieved in September / October 2008. A report containing BFBT simulation results was written and was presented at the SAFIR2010 Mid-Term seminar: M. Ilvonen & V. Hovi: Status of PORFLO code development (TRICOT Special Report). See (Ilvonen and Hovi 2009).

Preliminary simulations of one of the NUPEC (**N**uclear **P**ower **E**ngineering Corporation) BFBT (**B**WR **F**ull-size **F**ine-mesh **B**undle **T**ests) benchmark steady-state exercises were performed on a BWR (8×8) fuel bundle with a central water rod using the five-equation SIMPLE algorithm of PORFLO, as a transient simulation. System pressure was set at 6 MPa, total heating power at 3.52 MW, inlet mass flux at 1500 kg/m<sup>2</sup>s and inlet enthalpy at 1100 kJ/kg. The domain was split using a nodalization of (83×83×36), which amounts to

approximately 250 000 nodes. The steady state results of the five-equation model, in Figures 4.1 and 4.2, were obtained at the end of a 9.5 s transient. The computation time spent on a single core of a 2.992 GHz quad-core processor was 25 days. The friction was set uniform over the whole domain during the preliminary simulations. The results of the same simulation using the six-equation model are presented in Figure 4.3.

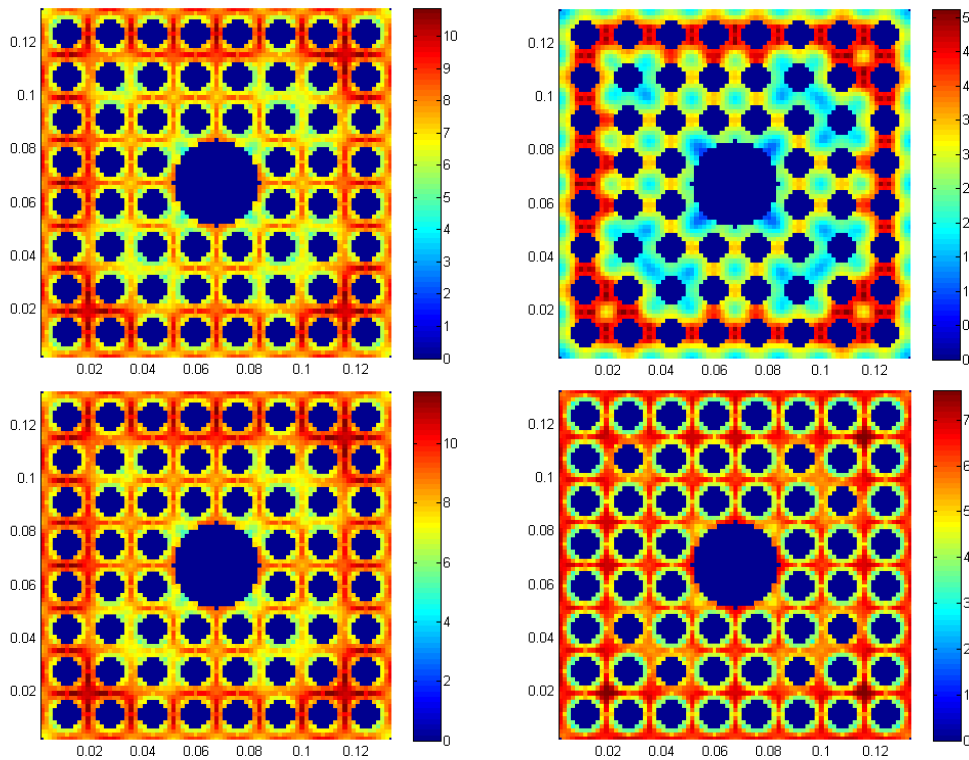


Figure 4.1: Vertical velocities (five-equation model) at outlet [m/s]: mixture velocity (upper left), slip velocity (upper right), vapor velocity (lower left) and liquid velocity (lower right).

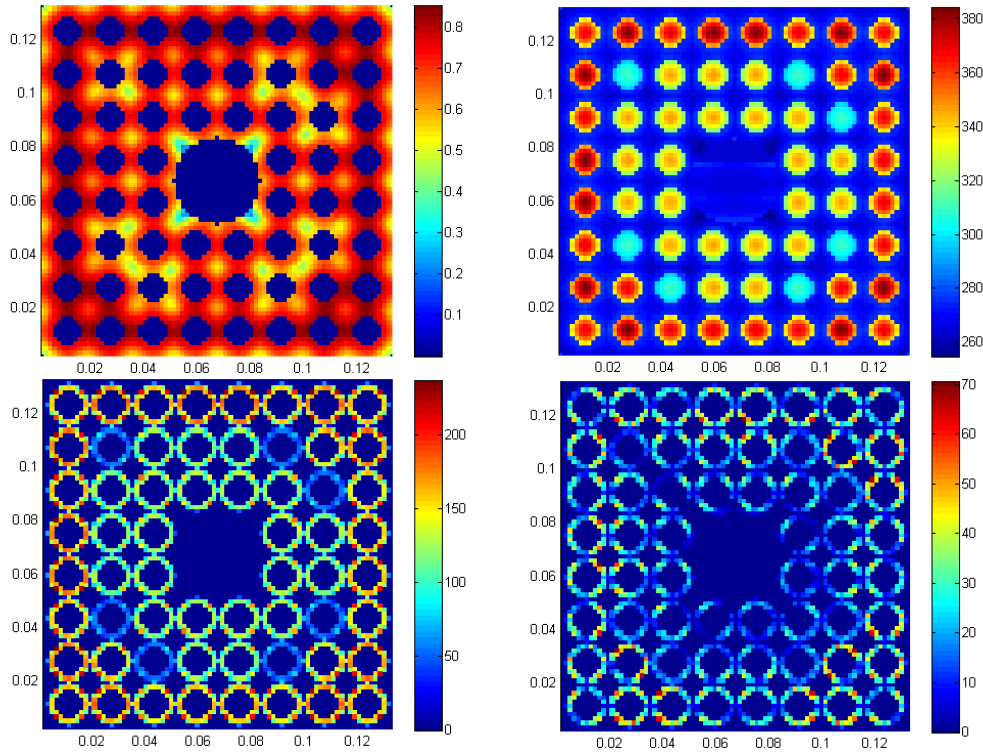


Figure 4.2: Contours of various parameters at outlet (five-equation model): void fraction [-] (upper left), temperature [°C] (upper right), volumetric heat flux for vapor generation [W/m<sup>3</sup>] (lower left) and volumetric heat flux for liquid heating [W/m<sup>3</sup>] (lower right).

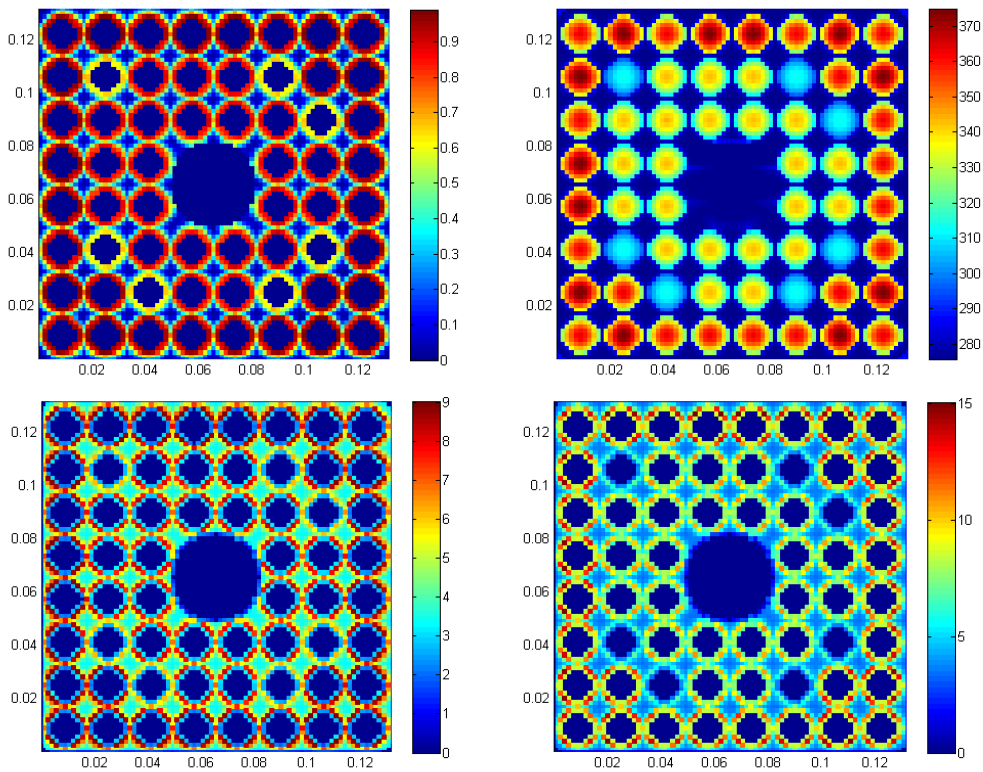


Figure 4.3: Contours of various parameters at outlet (six-equation model): void fraction [-] (upper left), temperature [°C] (upper right), vertical liquid velocity [m/s] (lower left) and vertical vapor velocity [m/s] (lower right).

The results of the six-equation model, in Figure 4.3, differ significantly from the corresponding five-equation results, in Figures 4.1 and 4.2; the most striking difference is the distribution of void fraction. The two void fraction distributions are similar in the sense that local maximums of void fraction are located on the surfaces of the fuel pins whereas local minimums are located at the centerlines of the subchannels, but the gradient of void fraction normal to the fuel pins (the rate in which void fraction changes while moving from the surface of a fuel pin towards the centerline of the subchannel) is radically different. It must be noted that the six-equation model implemented in PORFLO, at the time the simulations were done, lacks mechanisms, such as the drift-flux model inherent in the five-equation model, that convey vapor from the surfaces of the fuel pins towards the centerlines of the subchannels. These mechanisms (turbulence and possibly lift force) are to be developed in the near-future to obtain more realistic spreading of vapor throughout the subchannels.

The BFBT benchmark application was not too actively developed further, but some new considerations were given to its computational grid: It was decided to concentrate in 2010 on subchannel-scale predictions, which is more appropriate in the porous medium framework than the fine-grid ‘quasi-CFD’ simulations done so far.

## 4.2 VVER Steam Generator

The main application on which specific development efforts were concentrated in 2009 was the VVER horizontal steam generator. Steady-state at full power level (approx. 250 MW per SG) was reached after a 40 second transient simulation using PORFLO. The results have been compared with those of Fluent and generally the agreement is quite good.

A rather detailed geometrical description of the Loviisa steam generator was added, with the flexibility of using arbitrary Cartesian grids. Simulations of steam generator secondary side were run with as many as 5 million cells. Results show void fraction, temperatures, flow fields etc.

A rough comparison of PORFLO simulation results to Fluent simulations, conducted in SGEN project (Rämä 2009), are presented in this section. For easier comparison, void fraction distributions are plotted on yz-plane in Figure 4.4 and on xz-plane in Figure 4.5 for both Fluent and PORFLO simulations.

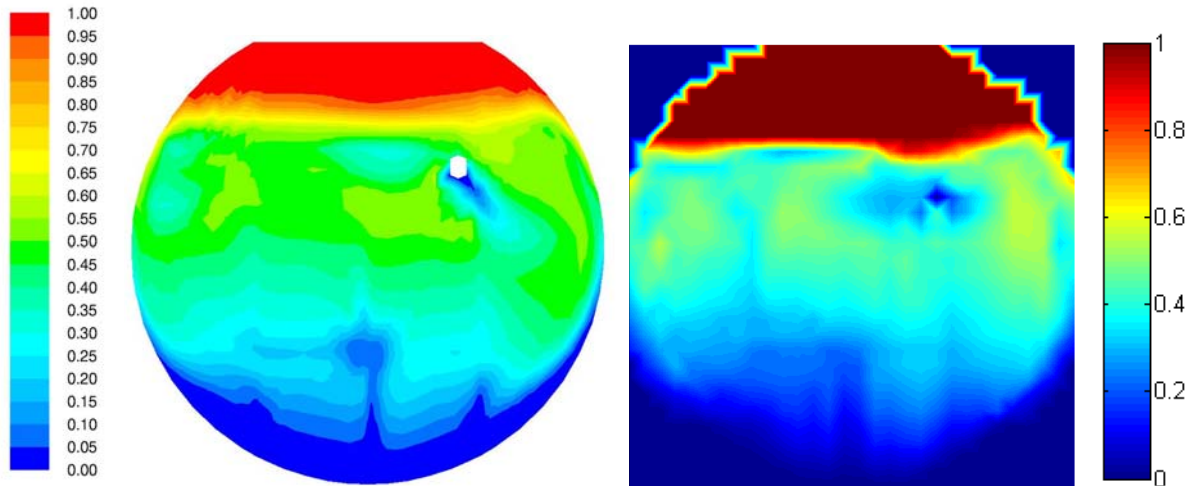


Figure 4.4: Comparison of void fraction distributions on cross-section  $x = 2.83$  m between Fluent (left) and PORFLO (right).

The general outlook of the two void fraction distributions on cross-section  $x = 2.83$ , in Figure 4.4, is similar; smaller void fractions are located near the bottom, near the feedwater injection line, at the gaps between the tube sections, especially in the middle, and at the outer shell of the steam generator. One of the differences is the direction of the feedwater flow near the injection line; Fluent simulations suggest that the feedwater flow (at this cross-section at least) is directed downwards into the section occupied by the primary tubes, whereas PORFLO simulations would suggest that the colder feedwater more or less stays on top of the tube bundles and moves to the middle of the steam generator. This can be explained in part by the differences in the boundary conditions between the two cases; in Fluent simulations the feed water is injected downwards with a source term for the liquid momentum, whereas in PORFLO the liquid is injected as a mass source (into the appropriate nodes) without a source term for momentum. Appropriate momentum sources could be included in PORFLO as well, but in the absence of detailed data concerning the liquid velocity at the nozzles of the feedwater injection line, the momentum source terms were neglected all together. Another difference is that the shape of the water level is different. There is a distinct dip in the water level predicted by PORFLO directly above the feedwater injection line. These two differences between the results may be linked; in Fluent simulations, as discussed above, the feedwater flow is directed further downwards, whereas in PORFLO simulations the cold feedwater is located directly below the dip in the surface level. The existence of cold feedwater near the surface may have a profound effect on the surface level at that point through increased condensation. At this point, however, it must be noted that on such coarse grids (grids with

relatively similar coarseness were used in both cases) it is difficult to capture the surface level accurately.

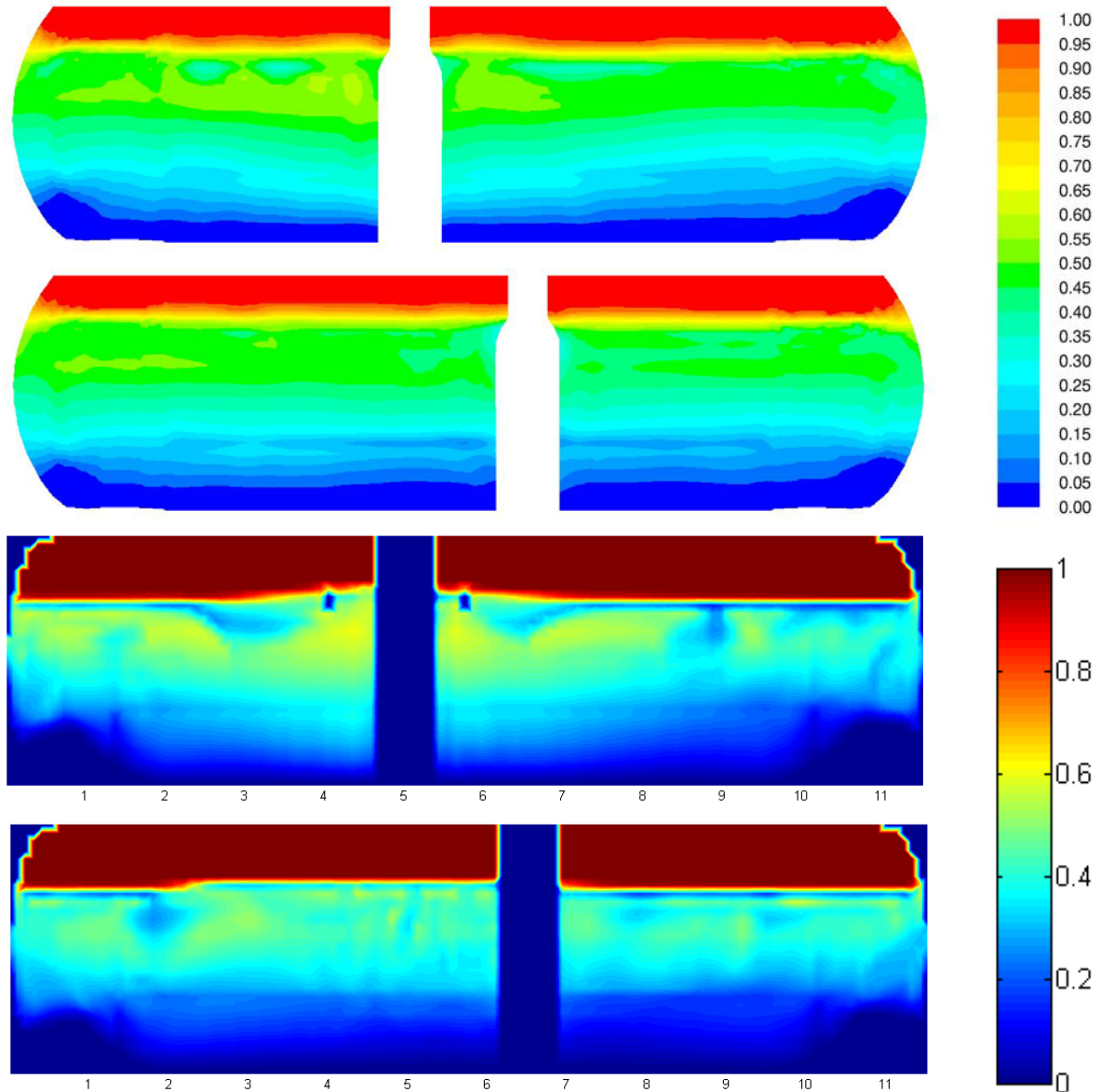


Figure 4.5: Comparison of void fraction distributions on cross-sections  $y = 2.0$  m (upper) and  $y = 1.21$  m (lower) between Fluent (on top) and PORFLO (below).

The void fraction distributions on cross-sections  $y = 2.0$  and  $y = 1.21$  m, in Figure 4.5, are more dissimilar than the cross-sections on  $yz$ -plane, in Figure 4.4. The “broad strokes” are similar in both cases; higher void fractions appear on the hot side (upper of the two pictures), especially near the hot collector, compared to the cold side and void fraction is increased when moving up from the bottom of the steam generator. The details that can be seen in both cases, however, are few. Generally speaking the distribution seems to be more uniform in the Fluent simulations than what is seen in PORFLO results. The numerous small pockets of



higher void fractions, seen in the lower of the two PORFLO plots in Figure 4.5, stand out in particular.

## 5 Planned applications

Plans of near-future (starting in 2010) applications of PORFLO are briefly summarized here.

- Particle bed dryout simulations (COOLOCE, a new experimental apparatus with conically-shaped pile of spherical particles)
- PSBT benchmark (PWR fuel bundle)
- EPR reactor core
- Connection with the system code Smabre

Planning work and preparations for the applications to be simulated in 2010 were partially started already in 2009.

### 5.1 COOLOCE simulations

Some initial plans were made for a new (or actually renewed) application of the code: study of the coolability of a pile of debris which consists of spherical metallic particles in a water pool, representing fuel particles for better-known geometry.

COOLOCE is an experimental apparatus, being assembled in another project (HYBCIS-2), which will be used to study the coolability of a pile of spherical particles by pool water. The particles are considered coolable as long as they are wetted and heat transfer from them is by boiling the water. The particles are made of a ceramic material and they are held by a wire-netting in a conically-shaped pile. Their diameter is less than 1 mm, which makes the application suitable especially for a porous medium model. They represent particles formed of molten fuel in a real severe reactor accident, but in the experiment, they are heated by vertical pin-shaped electrical resistors. When water and steam are at saturation, temperature of the particles (and thus their ability to boil water) is determined by heat conduction from the resistors to the particles and further from one particle to another.

First step in this application of PORFLO will be to set up the geometry and boundary conditions and to check whether the solution converges, without crashing of the code, at least with some initial heat transfer, friction and other correlations. The second and possibly more difficult step will be to find, implement and test correlations which are really physically appropriate for the case. The ultimate aim of the simulation is to predict correctly the time and

location of a dry (vapor-only) zone as observed by temperature measurements in the experiment. It will also be possible to compare PORFLO results with those of a German code, MEWA, which is used for simulations in the HYBCIS-2 project.

## 5.2 PSBT benchmark

The PSBT (PWR Subchannel and Bundle Tests) benchmark is the successor of the BFBT (BWR Full-size fine-mesh Bundle Test) benchmark and has many similarities with it. The original measurements were made by the same Japanese company (NUPEC, or Nuclear Power Engineering Corporation) and the benchmark exercises are arranged partly by the same people under OECD/NEA. During the BFBT benchmark in the previous years, PORFLO was not capable of fuel bundle simulations in the beginning and was constantly developed, until only at the end of the benchmark project some results could have been submitted. Now at least the starting point is already at far more advanced level.

According to the definition of the benchmark, the measurements available to PSBT participants include

### A. Void fraction measurements data

1. Steady-state void fraction in subchannel by CT measurement
2. Steady-state void distribution image in subchannel by CT measurement
3. Steady-state void fraction in rod bundle by chordal measurement
4. Steady-state void distribution image in rod bundle by chordal measurement
5. Transient void fraction in rod bundle

### B. DNB measurements data

1. Steady-state DNB data in rod bundle
2. Steady-state DNB detected location in rod bundle
3. Steady-state fluid temperature distribution in rod bundle
4. Transient DNB data in rod bundle

In much the same way as the BFBT benchmark, even the PSBT benchmark is divided into phases and exercises:

#### Phase I: Void distribution benchmark

- Exercise I-1: Steady-state single subchannel benchmark
- Exercise I-2: Steady-state bundle benchmark
- Exercise I-3: Transient bundle benchmark

#### Phase II: DNB benchmark

- Exercise II-1: Steady-state fluid temperature benchmark
- Exercise II-2: Steady-state DNB benchmark
- Exercise II-3: Transient DNB benchmark

With PORFLO, it is planned to concentrate on the subchannel resolution in the simulations, as opposed to the work done for the BFBT benchmark, where experimental simulations were done with much finer grid resolution. Subchannel resolution is the most appropriate for a porous medium approach, as the ultimate goal is to have a much smaller computational grid and much shorter CPU times than with usual CFD. Choice of the grid determines which phenomena are to be modeled (and resolved in the grid resolution) and which are e.g. implied by the correlations used.

### 5.3 EPR reactor core

For the first time in Finland, the Olkiluoto-3 EPR will have an open reactor core, i.e. no channel boxes enclosing the fuel elements, as is the case in Olkiluoto-1 and 2 BWRs, but also in Loviisa VVER plants. Furthermore, all the other three PWR candidates for the FIN-6 plant (AES2006, APR1400 and APWR) have an open core. Then, the interesting question is, to what extent the flow in each fuel element proceeds straight upwards and to what extent there is transverse mixing between the subchannels and fuel elements. More mixing will mean less probability of a local ‘hot spot’; therefore, it is very important to simulate the 3D flow field in a manner as mechanistic as possible, without resorting to artificial mixing terms. The transverse mixing is also important in such cases as ‘cold tongue’ and boron dilution. Generally, in any case where part of the reactor core (in the radial direction) is in a state other than the rest, it is important to have a proper 3D thermal-hydraulic model.

This is the background why one of the new PORFLO applications in 2010 will be the EPR reactor core with its 241 fuel elements and 89 control elements. In normal operation, the void fractions are very small (subcooled boiling near heat surface of the rods). PORFLO can accept geometrical information with as many details as available for this project, but as it uses the porous medium approach, the representation during simulation will be in the resolution of the grid used. In comparison to the app. 5500 primary tubes of the VVER steam generator (previously done application of PORFLO), the EPR core has app. 64 000 fuel rods (241 fuel elements, each of which has 265 rods in a 17x17 geometry). To have a reasonable number of grid cells, one fuel element could still be divided into a grid of 3x3 cells in the radial direction, each cell having app. 30 fuel rods. Then there would be more resolution available for the flow field solution than on a purely assembly-wise basis. Assuming almost cubical

grid cells and only the core as domain, the total number of cells in the simulation would be app. 150 000. The final objective is to have all the internal volume of the reactor pressure vessel (RPV) as the calculation domain (with only small couplings to the 1D circuit), but due to complexity of the geometry, some initial tests could be done with the core portion only, with suitable assumptions. It is hard to find validation data for such calculations; in France, which has 58 nuclear power plants, all PWR, some calculations e.g. with the FLICA-4 code, many times connected with the system code Cathare, have been published.

## 5.4 Connection with Smabre

Worldwide there has been a trend towards introducing 3D modules to traditional 1D system codes, in many cases by coupling two existing codes, one 1D and the other 3D, to work together, like the French Cathare-Flica-4 or Cathare-NeptuneCFD couplings or the ATHLET-CFX coupling by the German GRS. In Finland, Apros is being enhanced with a 3D flow solution module. For PORFLO, there has been a plan to study the possibility of coupling with the Smabre system code (and eventually the whole TRAB-3D-Smabre coupled code for system/core TH and neutronics), which was developed by Jaakko Miettinen, the author of the original PORFLO.

This plan will be put into practice in 2010 by first testing a simple one-directional coupling scheme in which boundary conditions from Smabre calculation are given to PORFLO, which will then calculate its own 3D version of the phenomena that take place inside a 3D component. This could be done as simply as transferring Smabre results to PORFLO through a file, but for the sake of later development (two-way coupling with common iteration) it is better to make a common 'master' code which calls the two models as subroutines. As the 3D application of the coupled simulation, it is suggested to use EPR reactor core or pressure vessel as described above.

The future objective, after 2010, is to have a two-way coupling of Smabre and PORFLO in such a way that the 3D part will be calculated by PORFLO only and the 1D circuitry by Smabre only. This would be best done by a common iteration loop in the master code, which calls the appropriate sections of both codes to iteratively once again calculate the same time step until convergence (convergence of both codes separately and at the same time convergence of common variables on the boundaries) is reached. Another theoretical possibility is to run the two codes as separate processes and do the necessary communication through MPI calls, resulting in at least a clearly defined interface. At the start of each

iteration, the two codes would get their boundary conditions from the other one in a ‘cross-referencing’ manner: Each one gets at its inlet boundary the mass inflow rate from the other code, and at the outlet boundary, the pressure as calculated by the other code. Only practical testing will find out whether convergence (as defined above) will usually be reached or not. Possibly there will be problems in situations where the direction of flow is reversed.

A potential limitation of the coupled code is the fact that Smabre does not perform iterative solution of a time step, but rather a once-through solution, the mass error of which is then ‘pushed’ forwards to be corrected during the next time step. This behavior is known to lead to oscillations in fast transients. So it would be best to run Smabre to a stabilized state (using description of the whole circuit) before spending a lot of CPU effort with the coupled PORFLO 3D calculation, or then modify Smabre to do iterative corrections during one time step.

In the coupled code, each original code would still use its input as before or almost as before; it would be up to the user to verify the compatibility of the two input sets. Each code would also keep its internal variables as before, but they should save / share their values between successive calls to initialization or calculation step, which can be accomplished e.g. by packing them into an f90 module.

## References

Hovi, Ville and Ilvonen, Mikko, 2010. PORFLO Simulations of Loviisa Horizontal Steam Generator. VTT Research report, VTT-R-01406-10. 34 p.

Ilvonen, M. and Hovi, V., The porous medium model PORFLO for 3D two-phase flow and its application to BWR fuel bundle simulations, In: SAFIR2010, The Finnish research programme on nuclear power plant safety 2007-2010, Interim report, VTT Research Notes 2466, Finland, 2009

Inoue, A., Kurosu, T., Aoki, T., Yagi, M., Mitsutake, T. and Morooka, S., Void Fraction Distribution in BWR Fuel Assembly and Evaluation of Subchannel Code. Journal of Nuclear Science and Technology. Vol. 32, part 7. pp. 629-640. (July 1995) ISSN 0022-3131

Ishii, M. and Zuber, N., 1979. Drag coefficient and relative velocity in bubbly, droplet or particulate flows. American Institute of Chemical Engineers Journal, Vol. 25, Issue 5, pp. 843-855

Lauder, B. E. and Spalding, D. B., 1974. The Numerical Computation of Turbulent Flows. Computer Methods in Applied Mechanics and Engineering, Vol. 3, pp. 269-289.

OGRE – Open Source 3D Graphics Rendering Engine  
<http://www.ogre3d.org/> (as in 5.2.2010)

Qt – A Cross-platform application and UI framework  
<http://qt.nokia.com/> (as in 5.2.2010)

Rämä, Tommi, 2009. Simulation of the Horizontal VVER-440 Steam Generator. Safir2010 report, (FNS-TERMO-144). 22 p.

Simovic, Z. R., Ocokoljic S. and Stevanovic V. D., 2007. Interfacial friction correlations for the two-phase flow across tube bundle. International Journal of Multiphase Flow, Vol. 33, pp. 217-226

Serre, G. and Bestion, D., October 2-6, 2005. Progress in improving two-fluid model in system code using turbulence and interfacial area equations, Proceedings of the 11<sup>th</sup> Int. Top. Meeting on Nuclear Thermal-Hydraulics (NURETH-11) Avignon, France.